

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGEEL**

DIEGO BUCHINGER

**AJUSTE DE MÚLTIPLAS SUPERFÍCIES B-SPLINE UTILIZANDO PRINCÍPIOS DE
RETALHOS DE GREGORY E ESTRATÉGIA DE DIVISÃO E CONQUISTA**

JOINVILLE

2022

DIEGO BUCHINGER

**AJUSTE DE MÚLTIPLAS SUPERFÍCIES B-SPLINE UTILIZANDO PRINCÍPIOS DE
RETALHOS DE GREGORY E ESTRATÉGIA DE DIVISÃO E CONQUISTA**

Tese apresentada como requisito parcial para obtenção do título de doutor em Engenharia Elétrica pelo Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Tecnológicas – CCT, da Universidade do Estado de Santa Catarina – UDESC.

Orientador: Roberto Silvio Ubertino Rosso Jr

**JOINVILLE
2022**

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Setorial do CCT/UDESC,
com os dados fornecidos pelo(a) autor(a)**

Buchinger, Diego
Ajuste de múltiplas superfícies b-spline utilizando princípios de
retalhos de Gregory e estratégia de divisão e conquista / Diego
Buchinger. -- 2022.
180 p.

Orientador: Roberto Silvio Ubertino Rosso Junior
Tese (doutorado) -- Universidade do Estado de Santa Catarina,
Centro de Ciências Tecnológicas, Programa de Pós-Graduação em
Engenharia Elétrica, Joinville, 2022.

1. Ajuste de superfícies. 2. Reconstrução de superfícies. 3.
B-Splines. 4. Algoritmo de divisão e conquista. 5. Retalhos de
Gregory. I. Rosso Junior, Roberto Silvio Ubertino. II. Universidade
do Estado de Santa Catarina, Centro de Ciências Tecnológicas,
Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

DIEGO BUCHINGER

AJUSTE DE MÚLTIPLAS SUPERFÍCIES B-SPLINE UTILIZANDO PRINCÍPIOS DE RETALHOS DE GREGORY E ESTRATÉGIA DE DIVISÃO E CONQUISTA

Tese apresentada como requisito parcial para obtenção do título de doutor em Engenharia Elétrica pelo Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Tecnológicas – CCT, da Universidade do Estado de Santa Catarina – UDESC.

Orientador: Roberto Silvio Ubertino Rosso Jr

BANCA EXAMINADORA

Prof. Dr. Roberto Silvio Ubertino Rosso Junior
Universidade do Estado de Santa Catarina

Membros:

Profa. Dra. Wu Shin-Ting
Faculdade de Engenharia Elétrica e de Computação da Unicamp

Prof. Dr. Marcos de Salles Tsuzuki
Escola Politécnica da Universidade de São Paulo

Prof. Dr. André Tavares da Silva
Universidade do Estado de Santa Catarina

Prof. Dr. Marcelo da Silva Hounsell
Universidade do Estado de Santa Catarina

Joinville, 29 de junho de 2022.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelo dom da vida e por este vasto, vislumbrante, singular e surpreendente universo em que vivemos.

Agradeço também toda minha família e amigos pela compreensão nos meus momentos de ausência e isolamento durante a elaboração e escrita deste projeto. Agradeço pelo afeto, pela atenção, pelo companheirismo e pelo incentivo durante todo o processo.

Gostaria de expressar ainda meus agradecimentos a todos os meus professores, do ensino básico ao ensino superior, que me ajudaram a obter o conhecimento necessário para a elaboração desta pesquisa e de todos os outros trabalhos, incentivando a curiosidade e a busca pelo conhecimento. Em especial, agradeço meu orientador de doutorado, Prof. Dr. Roberto Silvio Ubertino Rosso Junior, pela sua disposição e ajuda durante o processo desta pesquisa, assim como meu orientador de mestrado e graduação, Prof. Dr. Marcelo da Silva Hounsell, por me "iniciar" e direcionar neste caminho científico.

Não poderia deixar de agradecer também a todos os membros da banca, pelo seu tempo investido na leitura e discussão dos temas tratados nesta tese, pelas suas sugestões e críticas construtivas que colaboraram no aprimoramento deste trabalho como um todo.

Aproveito ainda este espaço para expressar meus agradecimentos à Universidade do Estado de Santa Catarina que propiciou a oportunidade de uma graduação e pós-graduação. Em especial, agradeço ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEEL) e ao Grupo de Automação de Sistemas e Robótica (GASR), por disponibilizar um espaço no Laboratório de Automação da Manufatura (LAMAN) e possibilitar a realização desta pesquisa com viés mesclado entre as áreas de engenharia elétrica e computação.

Por fim, gostaria ainda de expressar meu agradecimento a todas as pessoas que de alguma forma contribuíram para este trabalho – colegas da pós-graduação, amigos, pessoas desconhecidas ou mesmo anônimas – por meio de conversas, discussões ou publicações no meio digital. Agradeço inclusive àquelas pessoas que tiveram alguma contribuição indireta, mas que por meio de alguma influência não proposital acabaram contribuindo para a elaboração e execução desta pesquisa e trabalho.

*“There is nothing like looking, if you want to
find something. You certainly usually find
something, if you look, but it is not always quite
the something you were after.”*

(John Ronald Reuel Tolkien)

RESUMO

O ajuste ou reconstrução de superfícies é ainda um problema sem uma solução definitiva, o qual atrai atenção pela sua diversidade de aplicações, tais como *design* auxiliado por computador, manufatura auxiliada por computador e engenharia reversa. Superfícies paramétricas como B-Splines são amplamente utilizadas nesta tarefa, otimizando seus parâmetros de forma e localização com base na minimização do erro de ajuste. A quantidade de pontos de controle, os quais regem a superfície paramétrica, é um dos principais parâmetros relacionados com a qualidade e simplicidade do resultado. Como estimar tal valor é difícil, muitos métodos já vêm utilizando abordagens adaptativas para encontrar uma faixa de valores adequados para este parâmetro. Nesta direção, este trabalho apresenta um novo método de ajuste de múltiplas superfícies B-Splines, o qual escolhe de modo adaptativo a quantidade de pontos de controle, baseado em um algoritmo de divisão e conquista com utilização de um limiar de erro tolerável. Quatro procedimentos primários compõem o método: uma etapa de divisão e conquista, que subdivide e ajusta iterativamente os subconjuntos da nuvem de pontos; reparametrização, que tenta melhorar os parâmetros de localização dos retalhos de superfícies gerados; simplificação, que busca reduzir a quantidade de pontos de controle empregados; e a combinação, a qual conecta os retalhos com continuidade C^1 , utilizando os princípios das superfícies de Gregory. Diversos experimentos foram realizados com nuvens de pontos organizadas e não organizadas para verificar a viabilidade, eficiência e eficácia do método proposto. Foram testados e comparados também três métodos de particionamento da nuvem de pontos – bisseção, ponto de maior curvatura e ponto de menor curvatura. O método desenvolvido se mostrou capaz de ajustar superfícies de forma eficiente, gerando resultados com qualidade e simplicidade. Assim, o método proposto não exige parâmetros adicionais, escolhe automaticamente a quantidade de pontos de controle, particiona a nuvem de pontos sem auxílio manual e pode ser utilizado mesmo para nuvens de pontos com grande quantidade de amostras, casos nos quais o método pode ser ainda mais rápido do que um ajuste por uma única superfície com a mesma quantidade de pontos de controle.

Palavras-chave: Ajuste de superfícies. Reconstrução de superfícies. B-Splines. Algoritmo de divisão e conquista. Retalhos de Gregory

ABSTRACT

Surface fitting or surface reconstruction is still an open problem without an ultimate solution, which attracts attention from many applications, such as computer-aided design, computer-aided manufacturing, and reverse engineering. Parametric surfaces such as B-Splines are widely used in this complex task, optimizing their shape and localization parameters based on fitting-error minimization. The number of control points, which rules the parametric surface, is one of the prime parameters related to fitting quality and simplicity. As guessing such a value is difficult, many methods are currently using adaptive approaches to find a range of suitable values. This work presents a new multiple B-Spline surface fitting method, which adaptively chooses the number of control points based on a divide-and-conquer algorithm using a tolerable error threshold. The method is composed of four primary procedures: a divide-and-conquer step, which iteratively subdivides and fits patches on point cloud subsets; reparameterization, which tries to improve the patches' localization parameters; simplification, which tries to reduce the number of control points for each patch; and the combination, which connects the patches together with continuity C^1 , using Gregory's patch principles. Several experiments were performed with organized and unorganized point clouds to verify the feasibility, efficiency, and effectiveness of the proposed method. Three point cloud partitioning methods – bisection, greatest curvature point, and smallest curvature point – were also tested and compared. The developed method proved to be able to fit surfaces efficiently, generating results with quality and simplicity. Thus, the developed method does not require any additional parameters; it automatically chooses the number of control points; and partitions the point cloud without manual assistance. The method can be used even for point clouds with a large number of samples, in which case it can be even faster than a single-surface fit with the same number of control points.

Keywords: Surface fitting. Surface reconstruction. B-Splines. Divide and conquer algorithm. Gregory patches.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração dos vetores tangente e normal em relação a um ponto p de uma curva.	28
Figura 2 – Ilustração do plano tangente, dos planos de curvaturas principais e do vetor normal em relação a um ponto de uma superfície de sela.	29
Figura 3 – Graus de continuidade geométrica entre retalhos de superfície	31
Figura 4 – Exemplo de uma superfície paramétrica e seus componentes	33
Figura 5 – Exemplo da distribuição de influência das funções de mistura de uma curva B-spline em relação aos valores paramétricos normalizados.	35
Figura 6 – Exemplo de uma operação de inserção de nó.	38
Figura 7 – Exemplo de um retalho quadrilátero de Gregory como uma superfície Bézier bicúbica.	42
Figura 8 – Exemplo de um retalho quadrilátero de Gregory sob a forma de uma superfície de Bézier biquintica.	43
Figura 9 – Representação dos principais elementos no processo de ajuste de superfícies paramétricas	45
Figura 10 – Classificação de nuvens de pontos de acordo com a distribuição das amostras	50
Figura 11 – Características usuais nas nuvens de pontos – representação 2D	51
Figura 12 – Exemplo de parametrização de superfície com mapeamento um-para-um . .	56
Figura 13 – Fluxograma de alto nível do método proposto para o ajuste de múltiplas curvas	89
Figura 14 – Exemplo de funcionamento do método proposto para o ajuste de múltiplas curvas sobre o <i>benchmark semicircle</i>	92
Figura 15 – Resultados obtidos com o ajuste de múltiplas curvas	94
Figura 16 – Comparação de tempo e erro de ajuste entre o método tradicional de ajuste de curvas e o método proposto para o ajuste de múltiplas curvas para a nuvem de pontos do contorno do continente americano	95
Figura 17 – Exemplo de um retalho quadrilátero de Gregory como uma superfície B-Spline bicúbica	96
Figura 18 – Visão geral do método de ajuste por múltiplas superfícies – macro procedimentos	101
Figura 19 – Análise do tempo de ajuste de superfície utilizando o método tradicional . .	102
Figura 20 – Demonstração de uma sequência iterativa de procedimentos de conquista e divisão	106
Figura 21 – Exemplo de divisão realizada pelos métodos de particionamento implementados	109
Figura 22 – Demonstração de considerações sobre operações de mesclagem com superfícies	111
Figura 23 – Exemplo do processo de redução	112
Figura 24 – Possíveis casos de ajuste múltiplo de quinas	114
Figura 25 – Ilustração de casos de ajustes múltiplos de bordas	117

Figura 26 – Nuvem de pontos e resultados obtidos para o ajuste sobre o <i>benchmark Shell</i>	125
Figura 27 – Mapas de calor dos resultados obtidos para o ajuste sobre o <i>benchmark Shell</i>	126
Figura 28 – Resultados quantitativos do ajuste do <i>benchmark Shell</i>	127
Figura 29 – Nuvem de pontos e resultados obtidos para o ajuste sobre o <i>benchmark Horn</i>	128
Figura 30 – Mapas de calor dos resultados obtidos para o ajuste sobre o <i>benchmark Horn</i>	129
Figura 31 – Resultados quantitativos do ajuste do <i>benchmark Horn</i>	130
Figura 32 – Nuvem de pontos e resultados obtidos para o ajuste sobre o <i>benchmark Crescent</i>	131
Figura 33 – Mapas de calor dos resultados obtidos para o ajuste sobre o <i>benchmark Crescent</i>	132
Figura 34 – Resultados quantitativos do ajuste do <i>benchmark Crescent</i>	133
Figura 35 – Nuvem de pontos e resultados obtidos para o ajuste sobre o <i>benchmark TearDrop</i>	134
Figura 36 – Mapas de calor dos resultados obtidos para o ajuste sobre o <i>benchmark TearDrop</i>	135
Figura 37 – Resultados quantitativos do ajuste do <i>benchmark TearDrop</i>	136
Figura 38 – Nuvem de pontos e resultados obtidos para o ajuste sobre o <i>benchmark Pisot Triaxial</i>	137
Figura 39 – Mapas de calor dos resultados obtidos para o ajuste sobre o <i>benchmark Pisot Triaxial</i>	138
Figura 40 – Resultados quantitativos do ajuste do <i>benchmark Pisot Triaxial</i>	139
Figura 41 – Nuvem de pontos e resultados obtidos para o ajuste sobre o <i>benchmark Elliptic Torus</i>	141
Figura 42 – Mapas de calor dos resultados obtidos para o ajuste sobre o <i>benchmark Elliptic Torus</i>	141
Figura 43 – Resultados quantitativos do ajuste do <i>benchmark Elliptic Torus</i>	142
Figura 44 – Nuvem de pontos e resultados obtidos para o ajuste sobre Equação A	143
Figura 45 – Mapas de calor dos resultados obtidos para o ajuste sobre Equação A	144
Figura 46 – Resultados quantitativos do ajuste sobre a nuvem de pontos gerada pela Equação A	144
Figura 47 – Nuvem de pontos e resultados obtidos para o ajuste sobre Equação B	146
Figura 48 – Mapas de calor dos resultados obtidos para o ajuste sobre Equação B	146
Figura 49 – Resultados quantitativos do ajuste sobre a nuvem de pontos gerada pela Equação B	147
Figura 50 – Exemplo de reamostragem inicial da nuvem de pontos	149
Figura 51 – Nuvem de pontos, parametrização e resultado obtido para o ajuste sobre a superfície que retrata a cabeça de Max Planck	151
Figura 52 – Mapa de calor do resultado obtido para o ajuste sobre a superfície que retrata a cabeça de Max Planck	152
Figura 53 – Nuvem de pontos, parametrização e resultado obtido para o ajuste sobre a superfície Modelo de Orelha	153

Figura 54 – Mapa de calor do resultado obtido para o ajuste sobre a superfície Modelo de Orelha	154
Figura 55 – Nuvem de pontos, parametrização e resultado obtido para o ajuste sobre a superfície <i>FanDisk</i>	155
Figura 56 – Mapa de calor do resultado obtido para o ajuste sobre a superfície <i>FanDisk</i> .	156
Figura 57 – Comparação de tempo e erro de ajuste entre a abordagem tradicional e a abordagem de ajuste por múltiplas superfícies	157

LISTA DE TABELAS

Tabela 1 – Tabela comparativa entre características e recursos de softwares de reconstrução.	78
Tabela 2 – Síntese de trabalhos envolvendo ajuste de curvas paramétricas	80
Tabela 3 – Síntese de trabalhos envolvendo ajuste de superfícies paramétricas	82
Tabela 4 – Lista de parâmetros para o método de ajuste por múltiplas curvas	90
Tabela 5 – Lista de parâmetros para o método de ajuste por múltiplas superfícies	99

LISTA DE ABREVIATURAS E SIGLAS

AIC	<i>Akaike Information Criterion</i>
BFGS	<i>Broyden-Fletcher-Goldfarb-Shanno</i>
BIC	<i>Bayesian Information Criterion</i>
CAD	<i>Computer-Aided Design</i>
CAM	<i>Computer-Aided Manufacturing</i>
GPU	<i>Graphics Processing Unit</i>
ISE	<i>Integral Square Error</i>
ME	<i>Maximum Error</i>
MSE	<i>Mean Square Error</i>
NURBS	<i>Non-Uniform B-Splines</i>
PBS	Partição por Bisseção
PDM	<i>Point Distance Minimization</i>
PMaC	Partição por Maior Curvatura
PMeC	Partição por Menor Curvatura
RMSE	<i>Root-Mean-Square Error</i>
SDM	<i>Squared Distance Minimization</i>
SVD	<i>Singular Value Decomposition</i>
TDM	<i>Tangent Distance Minimization</i> .
T-Spline	<i>Tensor Product B-Spline Surfaces with T-Junctions</i>

LISTA DE SÍMBOLOS

$\lfloor x \rfloor$	Arredondamento matemático para o valor inteiro mais próximo de x
$\lceil x \rceil$	Arredondamento matemático para o maior valor inteiro que é menor ou igual ao valor de x
$\lceil x \rceil$	Arredondamento matemático para o menor valor inteiro que é maior ou igual ao valor de x
$\ p - q\ $	Distância euclidiana entre os pontos p e q
$B_{i,d}$	Função de mistura de uma superfície Bézier ou B-Spline de grau d no vão i
C^i	Continuidade paramétrica de grau i entre curvas ou superfícies
$c^{[a]}$	Quantidade de colunas de uma matriz de pontos de controle referente a uma superfície a
$d_E(p, q)$	Distância euclidiana entre os pontos p e q
d	Grau de uma curva ou superfície paramétrica
d_u	Grau de uma superfície paramétrica na direção do parâmetro u
d_v	Grau de uma superfície paramétrica na direção do parâmetro v
G^i	Continuidade geométrica de grau i entre curvas ou superfícies
k	Ordem de uma curva ou superfície paramétrica
k_u	Ordem de uma superfície paramétrica na direção do parâmetro u
k_v	Ordem de uma superfície paramétrica na direção do parâmetro v
$\ell^{[a]}$	Quantidade de linhas de uma matriz de pontos de controle referente a uma superfície a
\mathbf{N}	Nuvem de pontos
$\mathbf{N}^{[i]}$	Subconjunto i de uma nuvem de pontos \mathbf{N}
N_i	i -ésima amostra de uma nuvem de pontos \mathbf{N}
n	Tamanho (quantidade de pontos) na nuvem de pontos, i.e. $ \mathbf{N} $
m	Quantidade de nós de uma curva paramétrica: $m = \mathbf{T} $
m_u	Quantidade de nós de uma superfície paramétrica na direção do parâmetro u ($m_u = \mathbf{T}_u $)
m_v	Quantidade de nós de uma superfície paramétrica na direção do parâmetro v ($m_v = \mathbf{T}_v $)
\mathbf{P}	Vetor ou matriz de pontos de controle de uma curva ou superfície paramétrica
$\mathbf{P}^{[i]}$	Pontos de controle de uma superfície $s_{[i]}$

$ \mathbf{P}^{[i]} $	Quantidade de pontos de controle de uma superfície $s_{[i]}$
P_D	Pontos de controle diagonais às quinas de uma superfície paramétrica
$P_{i,j}$	Ponto de controle na i -ésima linha e j -ésima coluna de uma superfície paramétrica
$P_{i,j,k}$	Ponto de controle de orientado na i -ésima linha, j -ésima coluna e relacionado com o parâmetro k de uma superfície paramétrica com princípios de Gregory.
p	Quantidade total de pontos de controle em uma curva ou superfície paramétrica
p_u	Índice da última linha na matriz de controle de uma superfície paramétrica
p_v	Índice da última coluna na matriz de controle de uma superfície paramétrica
$s_{[i]}$	Superfície paramétrica que ajusta uma nuvem de pontos i
$s(u, v)$	Equação paramétrica de uma superfície
\mathbf{T}	Vetor de nós de uma curva ou superfície paramétrica
t_i	i -ésimo nó de uma curva paramétrica
$t_{u,(i)}$	i -ésimo nó de uma superfície paramétrica na direção do parâmetro u
$t_{v,(i)}$	i -ésimo nó de uma superfície paramétrica na direção do parâmetro v
\mathbf{U}	Vetor de parâmetros de localização (parametrização de uma curva ou superfície paramétrica), sendo: $\mathbf{U} = \tau$
\mathbf{W}	Vetor de pesos de uma curva ou superfície paramétrica
$w_{i,j}$	Ponderador ou peso sobre o ponto de controle da i -ésima linha e j -ésima coluna de uma superfície paramétrica
Γ	Parâmetros de forma de uma curva ou superfície paramétrica (e.g. d , P e T)
γ	Métrica de erro de ajuste (ex: ISE, RMSE e ME)
δ	Fator utilizado em conjunto com outras variáveis para definir a quantidade de amostras a serem tomadas uniformemente a partir de uma curva ou superfície paramétrica
ε_i	Erro de ajuste residual em relação à i -ésima amostra de uma nuvem de pontos
$\varepsilon_{tol}^{[\gamma]}$	Erro máximo tolerável em um ajuste utilizando a métrica γ
$\varepsilon^{[\gamma]}$	Erro de ajuste de superfície utilizando a métrica γ
ζ	Quantidade máxima de amostras para a realização de uma operação de ajuste de superfície

η	Quantidade de pontos em um subconjunto de N ($\eta \leq n$)
λ	Fator utilizado como critério de decisão, em conjunto com o erro de ajuste, para a escolha adaptativa entre continuidades de junção C^0 e C^1
ρ	Quantidade de pontos de controle utilizada em um ajuste
ρ_{max}	Quantidade máxima de pontos de controle admitida em um ajuste
σ	Quantidade de pontos de controle otimizados na região de junção entre duas curvas paramétricas
τ	Vetor de parâmetros de localização (parametrização de uma curva ou superfície paramétrica)
$\tau^{[i]}$	Vetor de parâmetros de localização referentes a uma nuvem de pontos i
χ	Método de particionamento de nuvem de pontos
ψ	Quantidade de pontos de controle utilizados inicialmente para cada superfície paramétrica
$\psi_{[x]}$	Quantidade de pontos de controle utilizados no ajuste de uma superfície paramétrica X
ω	Fator de redução de pontos de controle

SUMÁRIO

1	INTRODUÇÃO	19
1.1	PROBLEMA	21
1.2	OBJETIVOS	22
1.2.1	Objetivo Geral	22
1.2.2	Objetivos Específicos	23
1.3	ESCOPO	23
1.4	METODOLOGIA	24
1.5	ESTRUTURA TEXTUAL	25
2	FUNDAMENTAÇÃO TEÓRICA	26
2.1	FUNDAMENTOS GEOMÉTRICOS	26
2.1.1	Funções de Distâncias	26
2.1.2	Vetores	27
2.1.3	Medidas de Curvatura	28
2.1.4	Continuidade em Curvas e Superfícies	30
2.2	CURVAS E SUPERFÍCIES PARAMÉTRICAS	31
2.2.1	Funções de mistura	33
2.2.2	Definição de Superfícies de Bézier e B-Spline	36
2.2.3	Inserção e Refinamento de Nós	37
2.3	JUNÇÃO DE SUPERFÍCIES PARAMÉTRICAS	38
2.3.1	O Problema da Compatibilidade de Torção e os Retalhos de Superfície de Gregory	41
2.3.2	Junções Integrais e Parciais	43
2.4	AJUSTE DE SUPERFÍCIES PARAMÉTRICAS	44
2.4.1	Métricas de Erro	48
2.4.2	Características das Nuvens de Pontos	50
2.4.3	Benchmarks no Ajuste de Superfícies	52
2.4.4	Considerações sobre Parametrização	54
2.4.5	Considerações sobre Ajuste	57
2.4.6	Premissas dos Algoritmos de Ajuste	59
2.5	DISCUSSÃO DO CAPÍTULO	61
3	REVISÃO DE LITERATURA	63
3.1	ALGORITMOS DE AJUSTE DE CURVAS E SUPERFÍCIES	63
3.1.1	Ajuste de Curvas com Parametrização	64
3.1.2	Ajuste de Superfícies com Parametrização	67
3.1.2.1	<i>Métodos Determinísticos</i>	67

3.1.2.2	<i>Métodos Estocásticos</i>	72
3.1.3	Ajuste de Curvas e Superfícies sem Parametrização	74
3.2	FERRAMENTAS E BIBLIOTECAS PARA AJUSTE DE SUPERFÍCIE	76
3.3	DISCUSSÃO DO CAPÍTULO	79
4	MÉTODO PROPOSTO	87
4.1	MÉTODO DE DIVISÃO E CONQUISTA NO AJUSTE DE MÚLTIPLAS CURVAS	88
4.1.1	Exemplo do funcionamento do método	91
4.1.2	Experimentos e Considerações sobre Eficiência	93
4.2	MÉTODO DE DIVISÃO E CONQUISTA NO AJUSTE POR MÚLTIPLAS SUPERFÍCIES	95
4.2.1	Superfícies B-Spline com Princípios de Retalhos de Gregory	95
4.2.2	Definição do Problema	97
4.2.3	Visão Geral do Método	98
4.2.4	Procedimento de Conquista	102
4.2.5	Procedimento de Divisão	104
4.2.5.1	<i>Investida com Particionamento Local e Junções T</i>	104
4.2.5.2	<i>Métodos de Particionamento sem Junções T</i>	105
4.2.5.3	<i>Métodos de Particionamento</i>	107
4.2.6	Procedimento de Reparametrização	108
4.2.7	Procedimento de Simplificação	110
4.2.8	Procedimento de Combinação	113
4.2.8.1	<i>Ajuste Múltiplo de Quinas</i>	114
4.2.8.2	<i>Ajuste Múltiplo de Bordas</i>	116
4.2.9	Mecanismos de Aceleração	118
4.2.9.1	<i>Explosão da Nuvem de Pontos</i>	118
4.2.9.2	<i>Programação Dinâmica</i>	119
4.2.9.3	<i>Cálculo de Erro de Ajuste com Limite de Erro</i>	119
4.2.9.4	<i>Redução de Sistemas Lineares em Junções</i>	119
4.3	CONSIDERAÇÕES	120
5	EXPERIMENTOS E RESULTADOS	122
5.1	EXPERIMENTOS COM NUVENS ORGANIZADAS	123
5.1.1	Benchmark Shell	125
5.1.2	Benchmark Horn	128
5.1.3	Benchmark Crescent	130
5.1.4	Benchmark TearDrop	134
5.1.5	Benchmark Pisot Triaxial	136

5.1.6	<i>Benchmark Elliptic Torus</i>	140
5.1.7	Superfícies Explícitas	142
5.2	EXPERIMENTOS COM NUVENS NÃO ORGANIZADAS	148
5.2.1	Superfície Max Planck	150
5.2.2	Superfície Modelo de Orelha	152
5.2.3	Superfície <i>FanDisk</i>	154
5.3	COMPARAÇÕES COM O MÉTODO DE AJUSTE PROPOSTO	156
5.4	CONSIDERAÇÕES	158
6	CONCLUSÃO	161
	REFERÊNCIAS	165
	APÊNDICE A – SÍNTESE DA REVISÃO DE FERRAMENTAS E BI-	
	BLIOTECAS PARA RECONSTRUÇÃO	174

1 INTRODUÇÃO

A aproximação ou ajuste de curvas e superfícies sob um conjunto de pontos discretos – uma nuvem de pontos – é um tema central na área de design auxiliado por computador (CAD – *Computer-Aided Design*), na qual se estudam formas robustas e eficientes de modelagem geométrica (STROUD, 2006; IGLESIAS; GÁLVEZ, 2014). Apesar da sua conexão com o CAD, o problema de ajuste também possui aplicações multidisciplinares, tendo participação significativa em estatística e principalmente em áreas envolvendo computação para aplicações industriais, como na área de manufatura auxiliada por computador (CAM – *Computer-Aided Manufacturing*), engenharia reversa, prototipagem rápida, robótica, visão computacional, reconhecimento de padrões e processamento de imagens (POTTMANN; LEOPOLDSEDER; HOFER, 2002; AFSHAR; SORYANI; RAHMANI, 2011; ČURKOVIĆ; ČURKOVIĆ; VUČINA, 2018). Devido a esta ampla atenção e utilização de métodos geométricos para o ajuste de curvas e superfícies, existe uma vasta literatura que abrange esta temática (POTTMANN et al., 2005; JAVIDRAD, 2012; RODRÍGUEZ, 2017).

A necessidade por prototipagem rápida nas indústrias e o desenvolvimento e barateamento de tecnologias de digitalização contribuíram para o aumento da notoriedade do processo de ajuste de superfícies com finalidade em reconstrução de objetos (BERGER et al., 2017). Na engenharia reversa, por exemplo, pode-se modelar, visualizar, analisar e testar superfícies ajustadas de objetos antes mesmo de manufaturá-los. Isto permite melhor qualidade e eficiência de design, análise e manufatura de produtos (VÁRADY; MARTIN; COX, 1997; BÉNIÈRE et al., 2013).

As operações de ajuste podem ser classificadas em abordagens de interpolação ou aproximação. Para muitas aplicações, um ajuste por aproximação costuma ser mais atrativo, especialmente quando a nuvem de pontos apresenta problemas como ausência de dados, ruído ou *outliers* (BERGER et al., 2017). Já com relação às diversas formas de ajuste possíveis, a utilização de modelos paramétricos recebe significativa atenção nas pesquisas devido às suas características e propriedades (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012; YOSHIMOTO; HARADA; YOSHIMOTO, 2003). Estes modelos costumam ser representados por meio de parâmetros de forma (Γ) e localização (τ). Os principais modelos utilizados são as curvas ou superfícies de Bézier e B-Spline (ZHANG et al., 2016; IGLESIAS; GÁLVEZ; COLLANTES, 2018), as quais são definidas por dois componentes principais: as funções de mistura (*blending functions*) e um polígono ou poliedro de controle – definidos por um conjunto de ρ pontos de controle $\mathbf{P} = [P_0, \dots, P_{\rho-1}]^T$ (PIEGL; TILLER, 1997; POTTMANN; LEOPOLDSEDER; HOFER, 2002; MORTENSON, 2006; ZHENG et al., 2012; BERTOLINO et al., 2021). Se por um lado modelos de domínio mais abrangente são capazes de representar mais geometrias, por outro, a otimização do seu maior número de variáveis torna o problema de ajuste mais complicado e custoso.

Algumas abordagens de ajuste costumam utilizar uma formulação de mínimos quadrados, buscando a minimização do erro entre os pontos da nuvem e o modelo de ajuste resultante (POTTMANN; LEOPOLDSEDER; HOFER, 2002; JAVIDRAD, 2012). Também é desejável

que o modelo resultante seja conciso, utilizando a menor quantidade possível de variáveis. Além disso, como nem todas as variáveis que definem os modelos paramétricos possuem uma relação linear com o erro residual, pode-se atacar este problema de duas maneiras: a primeira é fixar valores pré-definidos a todos os elementos que não possuem uma relação linear e resolver o problema como um sistema de equações lineares; a segunda é resolver o problema como um sistema de equações não lineares, fixando eventualmente alguns dos valores das variáveis envolvidas. Neste trabalho, denominaremos como ajuste tradicional de superfície, o ajuste no qual uma única superfície paramétrica é utilizada e todas as suas variáveis que não possuem uma relação linear com o erro são pré-definidas com valores típicos e o problema é resolvido como um sistema de equações lineares.

Quando são utilizadas nuvens de pontos com um número reduzido de amostras, o problema de minimização de erro em ajustes paramétricos pode ser resolvido de forma rápida pelos métodos tradicionais que utilizam uma única superfície. Por outro lado, a resolução com base nestes mesmos métodos pode demandar muitos minutos ou até mesmo muitas horas caso sejam utilizadas instâncias com uma quantidade abundante de amostras (BARHAK; FISCHER, 2001; KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). Outro complicador é a escolha da quantidade de variáveis de controle a serem utilizadas, a qual costuma ser escolhida manualmente, com base empírica em uma abordagem de tentativa e erro, mas que também pode ser escolhida através de um processo automatizado.

Apesar dos ajustes de curvas e superfícies paramétricas apresentarem características semelhantes, o ajuste de superfícies é uma tarefa mais complexa (PIEGL; TILLER, 1997). Um exemplo desta dificuldade adicional no ajuste de superfícies pode ser ilustrado pela impossibilidade de realizar um ajuste de um único retalho de superfície Bézier ou B-Spline sobre determinadas topologias de objetos, sendo necessária a utilização de múltiplos retalhos (MORTENSON, 2006). De acordo com Ćurković, Ćurković e Vučina (2018), este tipo de abordagem de ajuste com múltiplos retalhos é, atualmente, a forma dominante em se descrever e representar as formas de objetos. Entretanto, ao adicionar múltiplos retalhos de superfícies, cria-se uma necessidade de junção entre eles, preferencialmente com a obtenção de certo grau de continuidade e evitando a criação de características geométricas indevidas ou indesejadas.

Considerando as características e complicadores envolvidos no problema de ajuste de superfícies, o uso de métodos de ajuste que utilizam uma estratégia de divisão e conquista para resolvê-los pode ser um caminho promissor e eficiente. Diversos algoritmos baseados em divisão e conquista já são utilizados para a resolução eficiente de problemas como: busca (busca binária), ordenação (*mergesort* e *quicksort*), multiplicação de matrizes (algoritmo de Strassen), multiplicação de números inteiros grandes (algoritmo de Karatsuba) e transformada rápida de Fourier (Cooley-Tukey FFT) (CORMEN et al., 2009). Ademais, a possibilidade e disponibilidade de uso de múltiplos núcleos de processamento nas arquiteturas computacionais atuais, combinada com o potencial de paralelização apresentado pelos algoritmos que seguem tal estratégia, podem permitir uma melhora significativa no tempo de execução.

1.1 PROBLEMA

Considere uma nuvem de pontos $\mathbf{N} = [N_1, \dots, N_n]^T$ composta por n amostras de uma superfície, possivelmente pré-processadas com operações como registro (junção e alinhamento dos dados oriundos de múltiplos processos de digitalização), e remoção ou redução de ruído e *outliers*. O ajuste de superfícies paramétricas pode ser expresso como um problema de minimização de distância entre pontos, baseado em uma otimização de mínimos quadrados. O objetivo é calcular e ajustar uma superfície paramétrica $s(u, v)$ sobre N , minimizando cada erro residual ε_i entre os pontos da nuvem e seus pontos correspondentes na superfície ajustada. Assim, o problema pode ser formalizado como (IGLESIAS et al., 2018; MAO et al., 2018):

$$\min \left(\sum_{i=1}^n \varepsilon_i \right) = \min_{\Gamma, \tau} \left(\sum_{i=1}^n \|N_i - s(u_i, v_i)\|^2 \right)$$

onde $\|\cdot\|$ representa a função de distância euclidiana e a minimização ocorre através dos conjuntos de parâmetros Γ e τ relacionados a superfície. Os parâmetros de forma Γ – como graus da superfície, funções de mistura, quantidade e posição dos pontos de controle – estão implícitos na notação $s(u_i, v_i)$, enquanto que os parâmetros de localização τ são denotados pelos valores paramétricos associados aos pontos de N : $\tau = [(u_1, v_1), \dots, (u_n, v_n)]^T$.

Apesar do objetivo do problema ser a minimização do erro, a obtenção de um erro de ajuste total igual a zero representa um caso de interpolação da superfície resultante sobre os pontos da nuvem. Assim, buscando evitar a interpolação dos pontos, é comum a adição de termos de suavização na formulação de mínimos quadrados, ou a utilização de uma tolerância pré-estabelecida como um erro máximo tolerável (ε_{tol}) (YANG; WANG; SUN, 2004; WANG; POTTMANN; LIU, 2006; KRAGIĆ; ČURKOVIĆ; VUČINA, 2018).

As principais métricas de erro de ajuste também se baseiam no erro residual, tomado como a distância euclidiana entre a i -ésima amostra da nuvem de pontos e seu ponto correspondente na superfície paramétrica, de acordo com a parametrização utilizada. Elas são baseadas na soma (*Integral Square Error* – ISE), média (*Mean Square Error* – MSE; e *Root-Mean-Square Error* – RMSE) ou no maior valor entre os erros residuais (*Maximum Error* – ME) (EBRAHIMI; LOGHMANI, 2018; DUNG; TJAHIJOWIDODO, 2017; IGLESIAS; GÁLVEZ; LOUCERA, 2015). Embora representem apenas uma aproximação da diferença entre a nuvem de pontos e o modelo paramétrico ajustado, tais valores fornecem meios para a realização de comparações quantitativas que podem ser complementadas com uma comparação visual subjetiva (BERGER et al., 2013).

Além da minimização do erro de ajuste, a redução da quantidade de elementos que compõem o conjunto de variáveis Γ também pode ser considerada na otimização como um segundo objetivo (YANG; WANG; SUN, 2004). A quantidade de variáveis de representação da superfície desempenha um papel fundamental no balanceamento entre qualidade de ajuste e sua simplicidade (HASEGAWA; TORMENA; PARPINELLI, 2014). A escolha da quantidade de variáveis para representar a superfície resultante, no entanto, não é uma tarefa simples. A escolha

manual deste valor pode se tornar uma tarefa difícil e tediosa, exigindo inúmeras tentativas de ajuste até obter o resultado desejado. A quantidade destas variáveis também é específica para cada situação, dependendo das características do objeto que está sendo ajustado. Com isto, diversas pesquisas (EBRAHIMI; LOGHMANI, 2017; YOSHIMOTO; HARADA; YOSHIMOTO, 2003; PARK; LEE, 2007; ZHENG et al., 2012; BERTOLINO et al., 2021) estão atacando este problema com uma abordagem de escolha adaptativa para este valor.

Como as nuvens de pontos relacionadas ao problema de ajuste de superfícies contém usualmente milhares ou até mesmo milhões de pontos, alguns métodos podem se tornar inviáveis para aplicações que demandam um menor tempo de resposta. Métodos de ajuste que otimizam variáveis não lineares, ou até mesmo a solução de ajuste tradicional, podem demandar um tempo de computação significativo (FENG; TAGUCHI, 2017). Com o intuito de contornar este problema, pode-se fazer uso de soluções baseadas em ajustes lineares por meio de múltiplos retalhos de superfície. Ao fazer isto, no entanto, surgem novos complicadores, como a escolha do particionamento da nuvem de pontos em subgrupos e a realização da junção das superfícies com algum nível de continuidade (BERTOLINO et al., 2021). Uma alternativa para viabilizar a obtenção de continuidade entre diversos retalhos de superfícies é a utilização dos princípios dos retalhos de Gregory (GREGORY, 1974; CHIYOKURA; KIMURA, 1983).

Considerando que algoritmos baseados em uma estratégia de divisão e conquista são utilizados para diversos problemas computacionais, pode-se investigar se esta técnica pode ser utilizada no ajuste de superfícies paramétricas de forma eficaz – reconstruindo superfícies de boa qualidade – e eficiente – reconstruindo superfícies tão rápido quanto, ou mais rápido que outros métodos de ajuste equivalentes. O problema de ajuste se encaixa conceitualmente neste tipo de técnica, sendo que uma nuvem de pontos que representa uma superfície pode ser iterativamente dividida em subconjuntos de pontos menores, os quais podem ser ajustados individualmente, gerando diversos retalhos de superfície, e posteriormente combinados – utilizando os princípios dos retalhos de Gregory – em uma única superfície que reconstrói o objeto de interesse original. Além disto, este tipo de método pode ser combinado com diversos critérios (por exemplo, uma medida de erro de ajuste) para definir automaticamente uma quantidade sub-ótima de pontos de controle.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um método com base em uma estratégia de divisão e conquista para o ajuste de múltiplas superfícies B-Spline conectadas com continuidade C^1 , utilizando uma escolha adaptativa da quantidade de pontos de controle, validando e comparando a qualidade, simplicidade e eficiência do ajuste.

1.2.2 Objetivos Específicos

Como objetivos específicos, tem-se:

- a) Investigar estratégias e técnicas utilizadas no ajuste de curvas e superfícies paramétricas;
- b) Desenvolver um método de ajuste baseado na estratégia de divisão e conquista, que viabilize o uso de múltiplos retalhos de superfícies com princípios de retalhos de Gregory, buscando por soluções sub-ótimas com base em limites pré-estabelecidos de erro;
- c) Testar, analisar e comparar a influência de três abordagens distintas de métodos de particionamento da nuvem de pontos em relação aos principais procedimento do método e aos resultados obtidos;
- d) Comparar os resultados obtidos pelo método proposto com resultados encontrados na literatura;
- e) Comparar os resultados de métricas de erro e tempo de computação do método proposto com o método tradicional de ajuste de superfície para uma nuvem de pontos com grande quantidade de amostras.

1.3 ESCOPO

Como argumentado por Pottmann et al. (2005), e Afshar, Soryani e Rahmani (2011), o estudo do problema de ajuste por meio de uma perspectiva de otimização pode trazer mais benefícios do que uma investigação voltada a uma aplicação específica. Ao elaborar métodos genéricos, estes podem ser utilizados para resolver uma ampla variedade de problemas de aproximação de superfícies com poucas ou até mesmo nenhuma alteração. Todavia, em face da amplitude do problema de ajuste e sua vinculação com etapas prévias como obtenção e filtragem dos dados, limitações de escopo foram aplicadas para este trabalho.

Primeiramente, o intuito deste trabalho não é desbravar problemas e questões relativas à operação de registro, logo, são utilizadas apenas nuvens de pontos que possam ser, ou já estejam, alinhadas como uma única nuvem de pontos. De modo semelhante, este trabalho não busca explorar novas formas de computar uma ordenação ou parametrização para uma nuvem de pontos, sendo que, nos casos em que uma ordem ou organização dos pontos não for especificada, são empregados algoritmos amplamente discutidos na literatura a fim de gerar esta informação.

Apresenta-se neste trabalho uma abordagem de ajuste baseada em múltiplos retalhos de superfície B-Splines compostos por quatro bordas, que é o modelo predominante nos trabalhos de reconstrução de objetos. São utilizados os princípios dos retalhos de Gregory para viabilizar a junção com continuidade paramétrica entre os retalhos. Apesar do foco deste estudo estar direcionado ao ajuste de superfícies paramétricas, ideias e técnicas empregadas no ajuste de curvas foram utilizadas como base ou inspiração. Assim, os trabalhos de ajuste de curvas também foram incluídos na revisão de literatura apresentada.

Os resultados obtidos pelo método proposto neste trabalho foram avaliados e comparados em relação ao seu erro de ajuste (qualidade) e quantidade de variáveis necessárias para a representação (simplicidade ou compressão), utilizando as principais métricas da literatura. O tempo de execução também foi mensurado para fins de análise e comparação entre as etapas do método proposto e o tempo de execução do método tradicional. Todavia, a comparação direta com outros resultados apresentados na literatura não pode ser validada com isonomia, visto que o ambiente computacional de testes não é o mesmo.

Como o ajuste de superfícies costuma envolver a utilização de nuvens de pontos com uma quantidade expressiva de amostras, a análise de complexidade de algoritmos é relevante neste contexto. Considerações sobre este assunto são utilizadas para fundamentar o método proposto mas, não faz parte do escopo deste trabalho realizar uma análise de complexidade do método apresentado e dos demais métodos encontrados na literatura. Além disso, apesar de uma das motivações do método proposto ser a possibilidade de execução com processamento paralelo, não faz parte do escopo deste trabalho apresentar uma versão multiprocessada deste método.

1.4 METODOLOGIA

Inicialmente foram utilizadas pesquisas bibliográficas para as etapas iniciais e planejamento deste projeto, com o intuito de conhecer o estado da arte nos problemas de ajuste de curvas e superfícies, assim como no problema mais amplo de reconstrução. O mesmo tipo de pesquisa também foi utilizado para identificar *benchmarks* utilizados na literatura, assim como métricas quantitativas de avaliação referentes à qualidade de ajuste. Na sequência foi realizada uma pesquisa exploratória, buscando por uma estratégia de resolução adequada para o problema de ajuste, considerando as características almejadas, as contribuições e abordagens adotadas pelos trabalhos encontrados no levantamento bibliográfico. Considerando as propriedades e a existência de diversos algoritmos eficientes baseados em uma estratégia de divisão e conquista, adotou-se tal estratégia para nortear o desenvolvimento de uma solução. Foi elaborado primeiramente um método de ajuste por múltiplas curvas paramétricas, o qual foi validado e testado previamente, e então adaptado para o problema de ajuste de superfícies, o qual é apresentado neste trabalho.

Com o método proposto devidamente implementado, foi realizada uma pesquisa experimental com uma abordagem quali-quantitativa. O algoritmo proposto foi testado e comparado com outros métodos da literatura utilizando as métricas e os *benchmarks* apropriados que foram identificados na pesquisa bibliográfica. São analisados e discutidos também o comportamento e as propriedades observadas no algoritmo proposto, considerando as diferentes propriedades geométricas das superfícies ajustadas.

1.5 ESTRUTURA TEXTUAL

A fim de apresentar a pesquisa realizada, o texto deste trabalho foi organizado em seis capítulos da seguinte maneira. O segundo capítulo apresenta a fundamentação teórica das temáticas envolvidas, englobando aspectos geométricos, os principais modelos paramétricos de superfícies, além de questões de ajuste e de representação de superfícies. No terceiro capítulo é apresentada uma revisão do estado da arte, englobando trabalhos voltados para o ajuste superfícies e curvas paramétricas, além de um levantamento de bibliotecas e ferramentas voltadas para o ajuste de superfícies. O quarto capítulo apresenta o método proposto, trazendo uma discussão sobre suas características e detalhes de funcionamento. No quinto capítulo são apresentados os resultados obtidos, assim como comparações entre o método proposto, o método tradicional de ajuste de superfícies e métodos da literatura. No sexto capítulo são apresentadas as considerações finais do projeto, seguidas pelas referências.

2 FUNDAMENTAÇÃO TEÓRICA

O ajuste de curvas e superfícies paramétricas é um processo que pode ser explorado de diversas maneiras. Grande parte dele é fundamentado e envolve aspectos geométricos e topológicos, sendo que diversos termos e conceitos desta área de estudo são utilizados na literatura científica. Em face destes fundamentos e relações, as seções deste capítulo tem como intuito apresentar as terminologias, conceitos e notações que estão associados ou que serão utilizados neste trabalho. São apresentados os conceitos básicos referentes aos aspectos geométricos, formalização e detalhes sobre as superfícies paramétricas utilizadas neste trabalho e os fundamentos do processo de ajuste de curvas e superfícies.

2.1 FUNDAMENTOS GEOMÉTRICOS

Existem dois focos principais no estudo de ajuste de curvas e superfícies. Um deles tange as questões geométricas atreladas ao posicionamento espacial e a forma do objeto, i.e. adquirir uma curva ou superfície que aproxime da melhor forma possível o objeto de estudo; e outro tange os aspectos topológicos atrelados à estrutura e conexão do objeto, i.e. adquirir uma curva ou superfície que aproxime da melhor forma possível a topologia do objeto de estudo. Apesar de relacionados, um método de ajuste geralmente se relaciona mais estreitamente com um deles. Entre eles, o aspecto geométrico costuma ser o objetivo mais almejado pelas pesquisas práticas (BOISSONNAT; CHAZAL; YVINEC, 2017). Nas próximas subseções são apresentados alguns aspectos geométricos utilizados neste trabalho.

2.1.1 Funções de Distâncias

Uma função ou métrica de distância d_Δ sobre um conjunto Y é um mapeamento entre pares de elementos em números não negativos que representam a sua distância relativa, i.e. $d_\Delta : Y \times Y \rightarrow [0, +\infty)$. Considerando três pontos, p , q e r , tal função deve satisfazer as seguintes relações básicas (BOISSONNAT; CHAZAL; YVINEC, 2017; SAMET, 1984):

1. Imagem em \mathbb{R}^+ : $d_\Delta(p, q) \geq 0$ e $d_\Delta(p, q) = 0$ se e somente se $p = q$;
2. Simetria: $d_\Delta(p, q) = d_\Delta(q, p)$;
3. Desigualdade triangular: $d_\Delta(p, r) \leq d_\Delta(p, q) + d_\Delta(q, r)$;

Entre o conjunto de métricas de distâncias, algumas se destacam em relação ao seu uso. Considerando um espaço k -dimensional, no qual dois ponto p e q são definidos por k valores cada: (p_1, \dots, p_k) e (q_1, \dots, q_k) , a métrica de distância euclidiana entre estes dois pontos é definida como (BOISSONNAT; CHAZAL; YVINEC, 2017; SAMET, 1984):

$$d_E(p, q) = \|p - q\| = \sqrt{\sum_{i=1}^k (p_i - q_i)^2} \quad (1)$$

2.1.2 Vetores

Um vetor pode ser entendido como um conjunto ordenado de números que denotam uma flecha ou uma linha finita, através dos quais são expressadas propriedades de direção, sentido e magnitude (ou comprimento) (LEON, 2018). Na literatura em inglês, no entanto, se usa apenas os termos de direção e magnitude, o que foi adotado na escrita deste texto. Considerando um espaço tridimensional, a forma tradicional de sua representação é:

$$v = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$$

onde \mathbf{i} , \mathbf{j} e \mathbf{k} representam os vetores de base (MORTENSON, 2006). Por simplificação, quando os vetores de base são bem definidos, um vetor é representado com a seguinte notação:

$$v = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$$

A magnitude, comprimento ou norma de um vetor é um valor real denotado como $\|\vec{p}\| = \sqrt{\vec{p} \cdot \vec{p}}$, onde o operador \cdot representa o produto escalar (*dot product*). Considerando dois vetores, \vec{p} e \vec{q} , ambos compostos por n valores, o produto escalar é um valor definido como (GRAY; ABBENA; SALAMON, 2006; LEON, 2018):

$$\vec{p} \cdot \vec{q} = \sum_{i=1}^n p_i q_i \quad (2)$$

Vetores que possuem magnitude igual a 1 são denominados como vetores unitários. Note que qualquer vetor não nulo possui um vetor unitário que aponta para a mesma direção e sentido. O ângulo θ entre dois vetores \vec{p} e \vec{q} pode ser calculado como (GRAY; ABBENA; SALAMON, 2006; LEON, 2018):

$$\cos \theta = \frac{\vec{p} \cdot \vec{q}}{\|\vec{p}\| \|\vec{q}\|} \quad (3)$$

O operador de produto escalar pode ser utilizado para determinar a perpendicularidade entre dois vetores: se $\vec{p} \cdot \vec{q} = 0$ então \vec{p} é perpendicular a \vec{q} (MORTENSON, 2006). Outro operador utilizado entre vetores é o produto vetorial (*cross product* ou *vector product*), denotado pelo símbolo \times . O resultado de um produto vetorial é outro vetor com a mesma quantidade de elementos e que é perpendicular aos outros dois vetores. Considerando dois vetores \vec{p} e \vec{q} em \mathbb{R}^3 , o produto vetorial é definido como (GRAY; ABBENA; SALAMON, 2006):

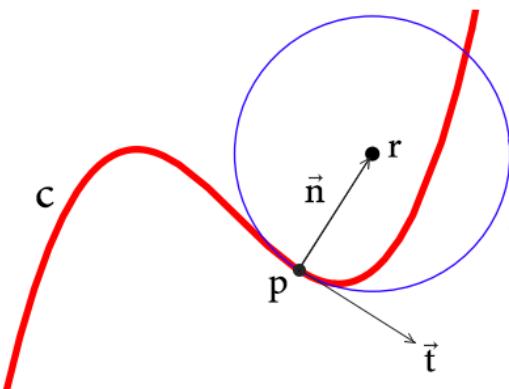
$$\vec{p} \times \vec{q} = \det \begin{bmatrix} i & j & k \\ p_1 & p_2 & p_3 \\ q_1 & q_2 & q_3 \end{bmatrix} \quad (4)$$

Este operador pode ser utilizado para verificar a propriedade de paralelismo entre dois vetores: se $\vec{p} \times \vec{q} = 0$ então \vec{p} é paralelo a \vec{q} (MORTENSON, 2006).

Um vetor tangente pode ser identificado como uma flecha com origem em um ponto p e outra extremidade em $p + v$. Ao considerar curvas tridimensionais, um vetor é denominado

tangente se ele é tangente a um determinado ponto p da curva. De modo semelhante, um vetor é dito normal em relação ao mesmo ponto p se ele está contido no plano normal, um plano que é perpendicular ao vetor tangente deste mesmo ponto. Dentre os vetores normais, um vetor é denominado como vetor normal principal, pois sua direção aponta para o centro de curvatura da curva em relação ao ponto p e sua magnitude pode ser tomada como uma medida de curvatura (GRAY; ABBENA; SALAMON, 2006; MORTENSON, 2006). A Figura 1 ilustra um exemplo de vetor tangente \vec{t} e de um vetor normal principal \vec{n} em relação a um ponto p de uma curva bidimensional c , destacando também o centro de curvatura r deste ponto e seu círculo osculador.

Figura 1 – Ilustração dos vetores tangente e normal em relação a um ponto p de uma curva.



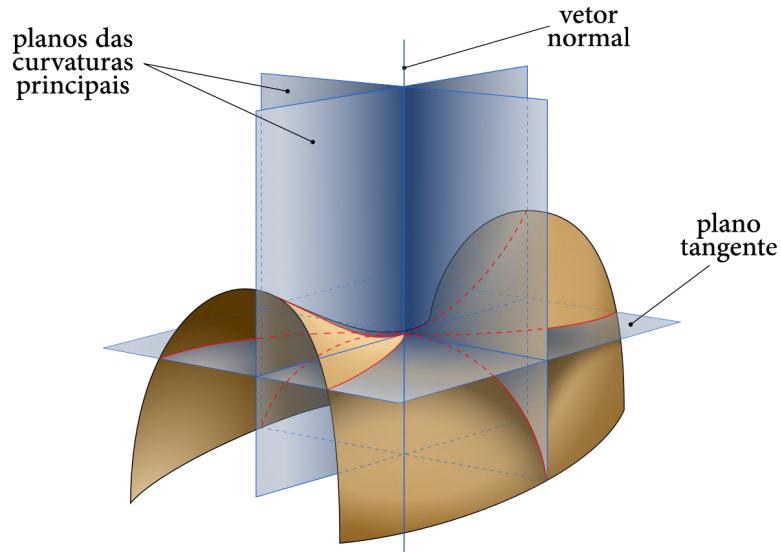
Fonte: Adaptado de Gray, Abbena e Salamon (2006)

2.1.3 Medidas de Curvatura

Considere uma superfície tridimensional s e um ponto p qualquer desta superfície. Diferentemente do caso das curvas, cada ponto p de s possui um plano tangente. Com base neste plano, infinitos planos normais – os quais são ortogonais ao plano tangente – podem ser gerados, mas apenas dois deles possuem os valores máximo e mínimo para a curvatura normal resultante. Estes dois planos normais especiais são denotados como planos de curvatura principal e são ortogonais entre si. Os valores máximo e mínimo de curvatura são denominados de curvaturas principais (κ_1 e κ_2) e os vetores unitários relacionados a estas curvaturas são chamados de vetores principais (\vec{k}_1 e \vec{k}_2). A Figura 2 apresenta estes elementos sobre um ponto central da superfície de sela.

Duas medidas de curvatura bem conhecidas são definidas com base nos valores de curvatura principal: a curvatura Gaussiana (K), que é uma medida intrínseca definida pelo produto das curvaturas principais, i.e. $K = \kappa_1 \cdot \kappa_2$; e a curvatura média (H), que é uma medida extrínseca de curvatura definida pela média das curvaturas principais, i.e. $H = (\kappa_1 + \kappa_2)/2$ (GRAY; ABBENA; SALAMON, 2006; BOTSCHE et al., 2010). Estas medidas de curvatura estão relacionadas ao campo de geometria diferencial, no qual as superfícies são definidas por equações que são diferenciáveis. Em computação gráfica, no entanto, é comum a utilização de

Figura 2 – Ilustração do plano tangente, dos planos de curvaturas principais e do vetor normal em relação a um ponto de uma superfície de sela.



Fonte: Adaptado de GABA (2006)

representação por malhas triangulares. Uma malha de triângulos é uma coleção de triângulos sem uma estrutura matemática particular, na qual cada triângulo define um pedaço de uma superfície. A malha costuma ser representada por um conjunto de vértices e um conjunto de faces triangulares que conectam estes vértices (BOTSCHE et al., 2010).

Definições de operadores diferenciais discretos são utilizadas como alternativas para calcular curvatura em malhas triangulares, considerando que estas malhas podem ser interpretadas como uma aproximação linear por partes de uma superfície suave. Diversas abordagens já foram propostas na literatura, sendo comum a utilização da discretização do operador de Laplace-Beltrami (Δ_s) para este propósito (BOTSCHE et al., 2010).

Miyazaki et al. (2021) apresenta um método para uma aproximação discreta de curvatura para uma malha triangular que utiliza o operador de Laplace-Beltrami juntamente das informações das normais dos triângulos e dos vértices. A normal de um triângulo T qualquer ($n(T)$), da malha de triângulos de s , pode ser calculado como:

$$n(T) = \frac{(p_j - p_i) \times (p_k - p_i)}{\|(p_j - p_i) \times (p_k - p_i)\|} \quad (5)$$

onde p_i , p_j e p_k representam os vértices do triângulo T . Considerando que a área do triângulo T pode ser calculada como $A_T = \|(p_j - p_i) \times (p_k - p_i)\|$, a normal de um determinado vértice p_i pode ser aproximada através da área e normal dos triângulos que possuem o ponto p_i como um de seus vértices (esses triângulos são denotados por T_{p_i}):

$$n(p_i) = \frac{\sum_{T \in T_{p_i}} A_T \cdot n(T)}{\|\sum_{T \in T_{p_i}} A_T \cdot n(T)\|} \quad (6)$$

Já o operador de Laplace-Beltrami de um vértice qualquer p_i pode ser calculado através da

discretização uniforme do operador de Laplace-Beltrami (BOTSCHE et al., 2010):

$$\Delta_s(p_i) = \frac{\sum_{p_j \in E_1(p_i)} (p_j - p_i)}{|E_1(p_i)|} \quad (7)$$

onde $E_1(p_i)$ representa o conjunto de vértices com uma aresta de distância de p_i , i.e. os demais vértices dos triângulos T_{p_i} , e $|E_1(p_i)|$ representa a quantidade destes vértices. Por fim, a aproximação discreta da curvatura normal média em um vértice p_i é calculada como:

$$curv(p_i) = n(p_i) \cdot \Delta_s(p_i) \quad (8)$$

Os valores obtidos para cada vértice devem ser normalizados (MIYAZAKI et al., 2021).

2.1.4 Continuidade em Curvas e Superfícies

O uso de um único retalho de superfície muitas vezes não é suficiente para representar adequadamente formas geométricas complexas. Uma alternativa é representar tais formas através da combinação de múltiplos retalhos de superfícies de menor complexidade. Nesta abordagem, entretanto, deve-se levar em consideração a harmonia desejada nas uniões feitas, isto é, o grau de continuidade nos pontos de junção. Esta continuidade pode ser considerada por critérios geométricos ou paramétricos (PIEGL; TILLER, 1997; MORTENSON, 2006; STROUD, 2006).

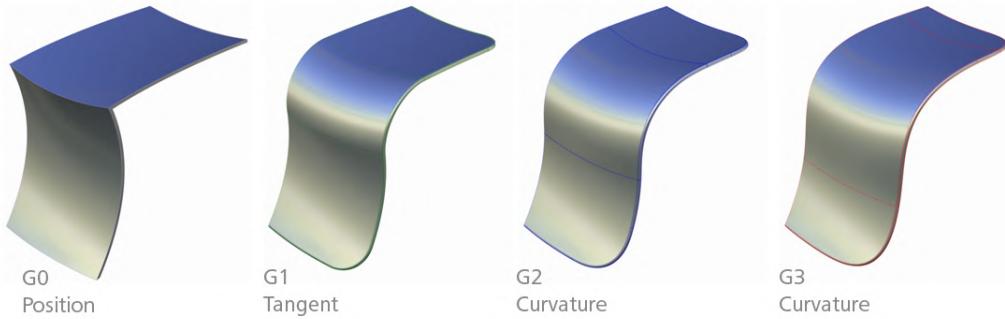
Considerando um cenário tridimensional, diz-se que dois retalhos de superfície possuem continuidade geométrica G^{-1} ou paramétrica C^{-1} caso eles estejam desconexos, sem uma extremidade em comum. Caso contrário, se os retalhos de superfície apresentarem uma extremidade em comum, garantindo que os fragmentos não sejam descontínuos, diz-se que estes retalhos possuem continuidade geométrica G^0 ou paramétrica C^0 . Esta também é denotada por continuidade posicional (STROUD, 2006).

Já para que uma junção entre retalhos de superfície tenha continuidade geométrica G^1 é necessário que, além da restrição de G^0 , as tangentes em cada ponto desses extremos em comum apontem para a mesma direção. Já para uma continuidade paramétrica C^1 , além de apontar para a mesma direção, as tangentes destes extremos devem ter a mesma norma – i.e. as primeiras derivadas nestes extremos devem ser iguais. Este tipo de continuidade é denotado também por continuidade tangencial (MORTENSON, 2006; STROUD, 2006).

Com C^1 , a unificação de retalhos de superfície já possui um aspecto suave, todavia, para algumas aplicações podem ser necessárias restrições ainda mais rígidas. Assim, para que uma unificação de superfícies tenha continuidade geométrica G^2 , além de mesma direção de tangente (G^1), os extremos em comum precisam ter o mesmo raio de curvatura. Já para obter continuidade paramétrica C^2 , exige-se também que as derivadas de segunda ordem nestes extremos tenham a mesma norma. Isso garante maior suavidade e uma curvatura homogênea na área de junção dos fragmentos de superfície. Tal grau de continuidade também é denotado por continuidade de curvatura (STROUD, 2006). De forma semelhante, outros graus superiores de continuidade exigem condições cada vez mais restritivas nas áreas próximas a junção. O terceiro grau de

continuidade por exemplo, G^3 e C^3 , são relativos a continuidade de um maior nível de curvatura, denotado também por torção (PIEGL; TILLER, 1997; MORTENSON, 2006; STROUD, 2006). A Figura 3 ilustra exemplos de junções com continuidade geométrica entre superfícies.

Figura 3 – Graus de continuidade geométrica entre retalhos de superfície



Fonte: AUTODESK (2017)

2.2 CURVAS E SUPERFÍCIES PARAMÉTRICAS

Curvas e superfícies podem ser representadas de forma exata ou aproximada por diversos modos, sendo que cada um deles possui suas características e possíveis limitações. Equações explícitas (e.g. $y = f(x, z)$), nas quais existe uma ou mais variáveis dependentes, possuem limitações para representar superfícies fechadas, por exemplo. Tal limitação pode ser evitada com o uso de equações implícitas (e.g. $f(x, y, z) = 0$), nas quais todas as variáveis são independentes entre si. Todavia, tanto equações explícitas quanto implícitas não são equações intrínsecas, i.e. elas são dependentes a algum ponto de referência ou eixo, sendo que a escolha do sistema de coordenadas pode afetar a facilidade na representação de curvas e superfícies. Equações paramétricas (e.g. $s(u, v) = [x(u, v), y(u, v), z(u, v)]$), por sua vez, utilizam um espaço paramétrico e são equações intrínsecas (MORTENSON, 2006). Ambas as formas implícita e paramétrica de equações possuem suas vantagens complementares em relação à operações geométricas, sendo possível converter uma representação na outra na maioria dos casos (ÜNSALAN; ERÇIL, 1999).

Uma vez que a descrição e ajuste dos coeficientes algébricos de uma equação de curva ou superfície não é uma das formas mais intuitivas de entender sua geometria, os modelos paramétricos de Hermite e Bézier propuseram a utilização de elementos de controle, como pontos e vetores, para definir curvas e superfícies de modo geométrico. Estes elementos de controle são ponderados e utilizados por funções de mistura (*blending function*) – também denominadas de funções de base (*basis function*) – as quais geram efetivamente o traçado da geometria (PIEGL; TILLER, 1997; MORTENSON, 2006). O conjunto de elementos que influenciam as funções de mistura é descrito como parâmetros de forma (Γ).

Uma superfície paramétrica pode ser definida por uma função vetorial de dois parâmetros, $s(u, v)$, representando o mapeamento ou deformação de um plano paramétrico uv sobre o espaço tridimensional euclidiano. Dentre os diversos esquemas possíveis para se representar superfícies, o produto tensorial é um dos mais amplamente utilizados em aplicações de modelagem geométrica. São utilizados os dois parâmetros u e v do espaço paramétrico, um conjunto de elementos geométricos de controle e funções de mistura bivariadas (PIEGL; TILLER, 1997).

Os parâmetros u e v , utilizados na representação por produto tensorial, definem o plano de base da superfície. Já os elementos de controle, também denominados de pontos de controle, são organizados topologicamente em uma malha bidirecional de tamanho $|\mathbf{P}_u| \cdot |\mathbf{P}_v|$, usualmente representada por uma matriz, onde $|\mathbf{P}_u|$ representa a quantidade de linhas e $|\mathbf{P}_v|$ representa a quantidade de colunas. Assim, os pontos de controle \mathbf{P} de uma superfície paramétrica $s(u, v)$ podem ser definidos como: $\mathbf{P} = \{P_{i,j} \mid i \in \mathbb{N}, i < |\mathbf{P}_u|, j \in \mathbb{N}, j < |\mathbf{P}_v|\}$, sendo $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ as coordenadas geométricas do ponto de controle da i -ésima linha e j -ésima coluna da matriz de controle. De acordo com a definição apresentada, os pontos de controle tem sua indexação iniciada pelo número zero. Para simplificar a representação da última linha e última coluna de pontos de controle são utilizados os termos p_u e p_v , respectivamente. Deste modo, tem-se que: $|\mathbf{P}_u| = p_u + 1$, $|\mathbf{P}_v| = p_v + 1$ e $\mathbf{P} = \{P_{0,0}, \dots, P_{0,p_v}, P_{1,0}, \dots, P_{1,p_v}, \dots, P_{p_u,0}, \dots, P_{p_u,p_v}\}$. Devido às suas características, o conjunto de pontos de controle também é denominado como poliedro de controle.

As funções de mistura bivariadas utilizadas na representação de superfícies paramétricas são construídas como produtos de dois conjuntos de funções de mistura univariadas: $f_i(u)$ e $g_j(v)$, ponderando a influência dos pontos de controle sobre as funções de mistura (PIEGL; TILLER, 1997). Assim, uma superfície definida por produto tensorial pode ser escrita como:

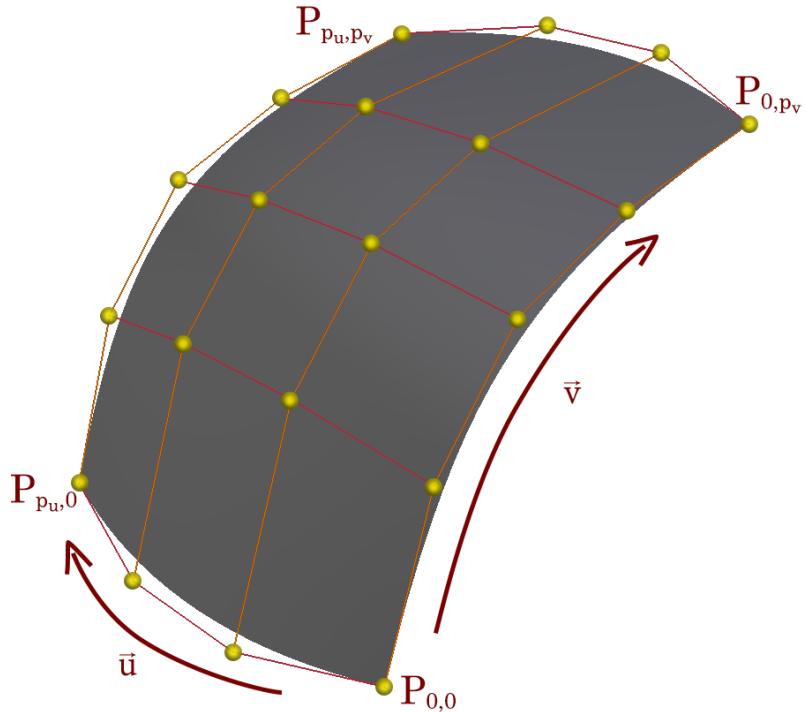
$$s(u, v) = (x(u, v), y(u, v), z(u, v)) = \sum_{i=0}^{p_u} \sum_{j=0}^{p_v} (P_{i,j} \cdot f_i(u) \cdot g_j(v))$$

onde: $\begin{cases} P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ 0 \leq u, v \leq 1 \end{cases}$

(9)

A Figura 4 ilustra um exemplo de superfície paramétrica na qual são evidenciados alguns de seus elementos principais. O retalho de superfície é influenciado pelas esferas amarelas, que denotam os pontos de controle, e delimitado pelos parâmetros normalizados u e v . As linhas de cor laranja e vermelho, que ligam os pontos de controle, indicam o poliedro ou matriz de controle, evidenciando também as relações de vizinhança entre eles. Já a indicação dos vetores direcionais (\vec{u} e \vec{v}) expressa o uso de dois conjuntos de funções de mistura distintos para cada direção dos pontos de controle e distingue a direção das linhas e colunas do polígono de controle.

Figura 4 – Exemplo de uma superfície paramétrica e seus componentes



Fonte: Elaborado pelo autor (2022).

2.2.1 Funções de mistura

As funções de mistura bivariadas utilizadas na representação de superfícies paramétricas podem empregar as mesmas funções de mistura univariadas das representações de curvas paramétricas. A partir desta possibilidade surge a relação entre as definições de curvas e superfícies paramétricas. As duas principais funções de mistura univariadas utilizadas em superfícies paramétricas são provenientes das definições das curvas de Bézier e B-Spline.

As curvas de Bézier empregam a família de polinômios de Bernstein ($B_{i,d}(u)$), que possui a característica de não realizar a interpolação dos pontos de controle, com exceção dos extremos, que são representados pelo primeiro e último ponto de controle. O grau d da curva é determinado pela quantidade de pontos de controle, sendo definido como: $d = p$, para uma curva constituída por $p + 1$ pontos de controle (PIEGL; TILLER, 1997; MORTENSON, 2006). A família de polinômios de Bernstein é definida como:

$$B_{i,d}(u) = \binom{d}{i} u^i (1-u)^{d-i} \quad (10)$$

Devido a definição da família de polinômios de Bernstein, as curvas de Bézier possuem a característica de controle global. Assim, mudanças em qualquer elemento de controle da curva afetam todo o segmento de curva, com maior ou menor intensidade. De modo semelhante, os polinômios de Bernstein podem ser utilizados para representar superfícies com a mesma característica de controle global. Esta forma de representação pode ser suficientemente boa para curvas e superfícies simples. No entanto, pode se tornar numericamente instável e computacionalmente

menos eficiente para representar formas complexas que exigem a utilização de muitos elementos de controle, o que produz um polinômio de alto grau (PIEGL; TILLER, 1997).

Uma maneira de contornar o problema do controle global e do vínculo entre a quantidade de pontos de controle e o grau do polinômio resultante é utilizar splines. Na matemática, as splines são entendidas como funções determinadas por polinômios que são definidos por partes. Na área de computação gráfica, costuma-se estender este conceito a fim de definir curvas ou superfícies compostas, que são constituídas da junção de curvas ou superfícies paramétricas individuais, nas quais a continuidade é controlada nas regiões de junção (MORTENSON, 2006).

Neste contexto, as B-Spline, ou *Basis Spline*, foram criadas como uma generalização das curvas de Bézier, sendo uma forma simplificada de expressar por uma única equação paramétrica a criação e junção de diversos segmentos de curvas, de um determinado grau, em uma curva contínua. Com estas características, uma B-Spline permite o controle local sobre a curva, onde alterações em um ponto de controle não afetam necessariamente toda a curva.

Curvas B-Splines são definidas de modo similar a uma curva de Bézier, utilizando pontos de controle e funções de mistura. Estas curvas são compostas por $p + 1$ pontos de controle e um grau d pré-determinado, o qual está associado à ordem da curva k , sendo $k = d + 1$ e tendo como restrição $d \leq p$. Além disso, B-Splines utilizam uma função base distinta, a qual costuma ser denotada por $N_{i,d}$ na literatura. No entanto, para evitar conflito com a notação utilizada na representação das amostras da nuvem de pontos, optou-se por utilizar a mesma notação já apresentada $B_{i,d}$. Estas funções de mistura fazem uso de um componente denominado de vetor de nós (*knot vector*) \mathbf{T} , constituído por m valores, i.e. $|\mathbf{T}| = m$, onde $m = p + k + 1$. Os nós são utilizados para ajustar as funções de mistura implicitamente, definindo a influência relativa dos pontos de controle sobre a curva (PIEGL; TILLER, 1997; MORTENSON, 2006).

O vetor de nós compõe o conjunto de parâmetros de forma (Γ) das B-Splines e é composto por valores reais que não podem ser decrescentes. A sua sequência de valores pode ser classificada em: periódica – cada valor é incrementado por um valor fixo em relação ao nó anterior – não periódica – os k primeiros nós possuem os mesmos valores assim como os k últimos nós – e não uniforme – os nós não são igualmente espaçados e podem ter qualquer composição não decrescente. O termo uniforme aberto, ou *clamped*, também pode ser encontrado na literatura denotando um vetor de nós que é uniforme, mas possui o mesmo valor para os k nós de cada extremidade. Fazendo uso destes elementos, as funções de mistura univariada das B-Splines é dada pela equação (PIEGL; TILLER, 1997; MORTENSON, 2006):

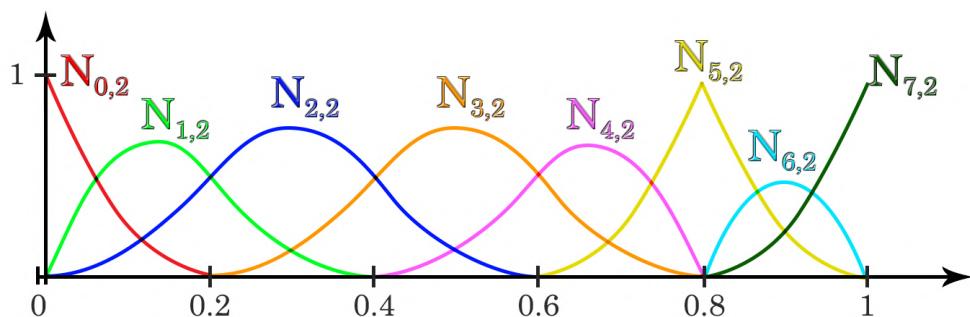
$$B_{i,d}(u) = \frac{(u - t_i)}{t_{i+d} - t_i} B_{i,d-1}(u) + \frac{(t_{i+d+1} - u)}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(u) \quad (11)$$

$\mathbf{T} = [t_0; t_1; \dots; t_{p+k-1}; t_{p+k}]$ Onde : $B_{i,0}(u) = 1$ se $t_i \leq u < t_{i+1}$
 sendo : $t_{i+1} \geq t_i$ $B_{i,0}(u) = 0$ caso contrário

As funções de mistura das B-Splines podem ser escritas de diversas formas, sendo denotada na Equação 11 pela fórmula recursiva de Cox-deBoor. Para esta função base podem ocorrer situações em que o denominador seja igual a zero, casos nos quais deve-se considerar

$x/0 \equiv 0$. Assim, é possível perceber que as funções de mistura são as mesmas para todos os pontos da curva, mas apenas algumas – quantidade relacionada ao grau e ordem da curva – são não-nulas e influenciam na curvatura em um determinado ponto paramétrico, garantindo a propriedade de controle local. Em outras palavras, cada ponto de controle possui uma influência local sobre a curva resultante, que é diretamente impactada e relacionada a um subconjunto do vetor de nós. A Figura 5 mostra um exemplo da variação de influência das funções de mistura de uma curva B-Spline de segundo grau ($k = 3$) com uso do seguinte vetor de nós não uniforme $\mathbf{T} = [0; 0; 0; 0,2; 0,4; 0,6; 0,8; 0,8; 1; 1; 1]$, em relação aos diferentes valores paramétricos considerados. O eixo vertical denota a influência da função de mistura (quanto maior o valor, maior a influência) e o eixo horizontal denota os valores paramétricos considerados. Note ainda que nesta figura utilizou-se a notação padrão para as funções de mistura de B-Splines, as quais são representadas pelo símbolo $N_{i,j}$ ao invés de $B_{i,j}$.

Figura 5 – Exemplo da distribuição de influência das funções de mistura de uma curva B-spline em relação aos valores paramétricos normalizados.



Fonte: Adaptado de Piegl e Tiller (1997)

O intervalo entre os valores de dois nós consecutivos do vetor de nós é denominado de extensão de nó (denominado como *knot span* na literatura) e se utiliza o termo multiplicidade de nós para indicar a quantidade de nós consecutivos de mesmo valor, i.e. com uma extensão de nó igual a zero. As B-Splines são infinitamente suaves no interior de um intervalo de nós diferente de zero. Já a continuidade sobre um nó está diretamente associada à composição do vetor de nós e ao grau da curva, sendo que um trecho de segmento de curva com grau d e multiplicidade de nós r apresentará pelo menos uma continuidade local C^{d-r} (XIE; QIN, 2001).

As funções de mistura univariadas das B-Splines apresentam a característica de serem invariantes sob transformações afins (translação, rotação, escalonamento e cisalhamento). Isto é, computar os pontos em uma curva B-Spline e então aplicar uma transformação afim, produz o mesmo resultado que aplicar a transformação afim sobre os pontos de controle e então computar os pontos desta curva. Além disso, também apresentam a característica de serem invariantes sob transformação de parâmetros – e.g. o espaço paramétrico não precisa se restringir ao intervalo unitário $[0, 1]$, podendo assumir qualquer intervalo $[a, b]$ (MORTENSON, 2006).

Outra propriedade das funções de mistura univariadas das B-Splines é a partição da

unidade, através da qual cada ponto da curva, relacionado a um ponto paramétrico, é representado por uma combinação unitária da influência dos pontos de controle. Isto é, $\sum_{i=0}^p B_{i,d}(u) = 1$ para qualquer u ($t_0 \leq u \leq t_{p+k}$). Para uma enumeração mais detalhada das propriedades destas funções de mistura e das B-Splines, recomenda-se a leitura do material de Piegl e Tiller (1997).

Além das funções de mistura univariadas empregadas nas curvas de Bézier e B-Splines, existem também outras variações que são utilizadas em curvas paramétricas com características semelhantes, mas que incluem outras propriedades específicas, como as Beta Splines, Catmull–Rom Splines e P-Splines, para citar algumas. As funções de mistura das Beta Splines é definida sobre um vetor de nós uniforme e adiciona dois parâmetros extras: β_1 e β_2 , que representam as propriedades de tendência e tensão, respectivamente. Alterar o valor de β_1 afeta os vetores tangentes dos pontos de controle, ao passo que alterar o valor de β_2 afeta a proximidade da curva em relação ao polígono de controle (MORTENSON, 2006). As funções de mistura das Catmull–Rom Splines, por sua vez, possuem a propriedade de interpolarem todos os seus pontos de controle da curva paramétrica com suavidade (CATMULL; ROM, 1974). Já as funções de mistura das P-Splines, ou *Penalized B-Splines*, provindas de estudos na área de regressão, trazem a ideia do uso de uma função de penalidade que visa evitar o sobre-ajuste (*overfitting*) da curva gerada sobre a nuvem de pontos (EILERS; MARX, 1996).

2.2.2 Definição de Superfícies de Bézier e B-Spline

Como consequência da representação de superfícies por produto tensorial através da composição de dois conjuntos de funções de mistura univariadas, uma relacionada ao valor paramétrico u e outra relacionada ao valor paramétrico v , os parâmetros de forma (Γ) são duplicados. Isto é, cada conjunto de funções de mistura apresenta seu próprio grau (d_u e d_v) e ordem (k_u e k_v) de curvatura, assim como um vetor de nós próprio (\mathbf{T}_u e \mathbf{T}_v), quando este é utilizado. As restrições sobre estes elementos são as mesmas aplicadas sobre as funções de mistura univariadas individualmente: $d_u \leq p_u$, $d_v \leq p_v$, $k_u = d_u + 1$, $k_v = d_v + 1$, $|\mathbf{T}_u| = m_u$ onde $m_u = p_u + k_u + 1$, e $|\mathbf{T}_v| = m_v$ onde $m_v = p_v + k_v + 1$. Ademais, o i -ésimo elemento dos vetores de nós t_u e t_v é representado neste trabalho, respectivamente, como: $t_{u,(i)}$ e $t_{v,(i)}$, sendo que a restrição de nós em ordem não decrescentes se aplica a ambos os vetores de nós:

$$t_{u,(i+1)} \geq t_{u,(i)} \text{ e } t_{v,(i+1)} \geq t_{v,(i)}.$$

Com base nas funções de mistura univariadas apresentadas, pode-se definir as superfícies paramétricas de Bézier e B-Spline como (PIEGL; TILLER, 1997):

$$s(u, v) = \sum_{i=0}^{p_u} \sum_{j=0}^{p_v} (B_{i, d_u}(u) \cdot B_{j, d_v}(v) \cdot P_{i,j}) \quad (12)$$

Estas superfícies paramétricas estão em sua forma não racional e por isso apresentam algumas limitações de representação ao serem utilizadas individualmente (MORTENSON, 2006). Entretanto, estas mesmas superfícies podem ser representadas também através de polinômios racionais, de forma semelhante a suas estruturas originais. As splines racionais podem ser entendidas como

uma projeção perspectiva de um produto tensorial 4D. De modo alternativo, pode-se considerar a consequência direta desta projeção, que é a utilização de um ponderador ou peso $w_{i,j}$ sobre cada ponto de controle $P_{i,j}$ (PIEGL; TILLER, 1997; MORTENSON, 2006). Assim, uma superfície B-Spline racional pode ser expressa como:

$$s(u, v) = \frac{\sum_{i=0}^{P_u} \sum_{j=0}^{P_v} (w_{i,j} \cdot P_{i,j} \cdot B_{i, d_u}(u) \cdot B_{j, d_v}(v))}{\sum_{i=0}^{P_u} \sum_{j=0}^{P_v} (w_{i,j} \cdot B_{i, d_u}(u) \cdot B_{j, d_v}(v))} \quad (13)$$

As superfícies racionais possuem a característica de invariância às transformações afins e também às projeções paralela e de perspectiva. Tanto as superfícies de Bézier quanto as superfícies B-Splines com vetor de nós não periódicos aberto (*clamped*), sejam elas racionais ou não, apresentam a característica de interpolar em qualquer condição os quatro pontos extremos do poliedro de controle. A Figura 4 denota estes pontos como $P_{0,0}$, $P_{p_u,0}$, P_{0,p_v} e P_{p_u,p_v} .

Destaca-se em particular o modelo de superfície B-Spline racional com utilização de vetor de nós não-uniforme (*Non-Uniform Rational B-Spline*), abreviado como NURBS (PIEGL; TILLER, 1997; MORTENSON, 2006). Sendo composta por um conjunto de tamanho adaptável de pontos de controle, pesos, conjunto de nós e um grau para cada conjunto de funções de mistura univariadas, as NURBS são um dos modelos paramétricos mais versáteis para representação de superfícies. Entretanto, exatamente o mesmo fato de ser definida por uma quantidade considerável de parâmetros constitui uma de suas principais desvantagens, que é a complexidade em se definir apropriadamente tais parâmetros de modo a representar uma superfície desejada (COSTA; MONTEMURRO; PAILHÈS, 2018).

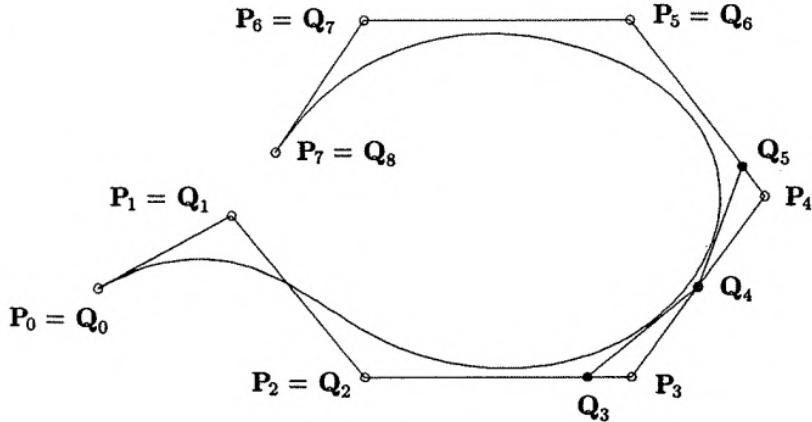
2.2.3 Inserção e Refinamento de Nós

As curvas e superfícies paramétricas de B-Spline viabilizam a realização de operações de adição de novos pontos de controle – e por consequência a mesma quantidade de elementos no vetor de nós – mantendo a mesma geometria. Tal tipo de operação possui seus benefícios, entre eles, o aumento de flexibilidade no controle da geometria, útil para aplicações de design interativo (PIEGL; TILLER, 1997). A nomenclatura utilizada diferencia a operação de adição de um único ponto de controle como operação de inserção de nó, da operação de adição de múltiplos pontos de controle, a qual é denominada como refinamento de nós.

Devido as propriedades das curvas de Bézier e B-Splines, durante uma operação de inserção de nó é necessário apenas o ajuste da posição de, no máximo, d pontos de controle, incluindo o novo elemento adicionado (PIEGL; TILLER, 1997). A Figura 6 ilustra um exemplo de inserção de nó em uma curva B-Spline cúbica, composta por oito pontos de controle $\mathbf{P} = \{P_0, \dots, P_7\}$, destacando a curva resultante e o polígono de controle. A nova configuração de pontos de controle é denotada por $\mathbf{Q} = \{Q_0, \dots, Q_8\}$ e apenas os pontos de controle P_3 e P_4 foram alterados, além da inclusão do novo ponto de controle Q_4 .

A adição de múltiplos nós pode ser realizada através de inserção de nós consecutivas, no entanto, esta estratégia costuma ser diferenciada da operação de refinamento de nós porque

Figura 6 – Exemplo de uma operação de inserção de nó.



Fonte: Piegl e Tiller (1997)

existem métodos mais eficientes para esta finalidade. Mais detalhes este procedimento e os algoritmos utilizados são apresentados por Piegl e Tiller (1997). No caso de operações de inserção ou refinamento de nós em superfícies de Bézier e B-Spline, o procedimento é similar ao caso de curvas, no entanto, se faz necessária a adição de uma ou múltiplas linhas ou colunas completas de pontos de controle, devido a sua estrutura matricial rígida (PIEGL; TILLER, 1997). Assim, ambas as operações não fornecem um meio de realizar um refinamento local da superfície, exigindo a adição de pontos de controle com propósito meramente estrutural.

2.3 JUNÇÃO DE SUPERFÍCIES PARAMÉTRICAS

Nem todos os objetos podem ser representados adequadamente com uma única superfície paramétrica Bézier ou B-Spline. Determinadas superfícies de gênero (ou *genus*) arbitrário só podem ser representadas adequadamente através da junção de superfícies (SEDERBERG et al., 1998; BERTOLINO et al., 2021). Todavia, tal junção deve levar em consideração os aspectos de continuidade nas bordas envolvidas, isto é, em relação a uma direção relacionada com os valores paramétricos (u e v) de cada retalho. Usualmente estas bordas são unidas de forma integral, isto é, as bordas de contato coincidem – ou são forçadas a coincidir – totalmente.

Considerando o caso da junção por uma borda integral, uma junção simples, com continuidade C^0 , é obtida nas superfícies não-racionais de Bézier ou B-Spline (com vetor de nós aberto) fazendo coincidir os pontos de controle de uma linha ou coluna dos extremos da matriz de controle, referente à borda que será compartilhada. Com isso, obtém-se uma junção entre duas superfícies mas, sem garantia de suavidade. Para garantir alguma suavidade, deve-se buscar por níveis de continuidade geométrica ou paramétrica maiores. De modo geral, uma continuidade paramétrica C^i é obtida através da igualdade entre as i -ésimas derivadas parciais das equações paramétrica envolvidas. Estas derivadas parciais são calculadas com base na direção do valor paramétrico associado à junção entre as superfícies (PIEGL; TILLER, 1997).

Assim, para obter uma junção com continuidade C^1 , é necessário aplicar a derivação parcial no sentido em que a ligação dos retalhos de superfícies irá ocorrer. Deste modo, existem duas possíveis equações que podem ser utilizadas dependendo da extremidade do retalho que será unida. As derivadas parciais costumam ser denotadas por: $\frac{\partial}{\partial u} s(u, v) = s^{(1,0)}(u, v) = s_u(u, v)$ e $\frac{\partial}{\partial v} s(u, v) = s^{(0,1)}(u, v) = s_v(u, v)$. Além das derivadas parciais em relação a uma das direções da superfície, Piegl e Tiller (1997) mostram ainda a derivada parcial dupla (relacionada à torção) aplicada a ambos os valores paramétricos u e v , que corresponde à variação combinada de u e v no retalho de superfície. Esta derivada pode ser utilizada para garantir a continuidade diagonal dos cantos de retalhos de superfícies unidos, sendo denotada por: $\frac{\partial}{\partial u \partial v} s(u, v) = s^{(1,1)}(u, v) = s_{uv}(u, v)$.

A fim de diferenciar as ordens de derivação parcial nas duas direções, indicadas por símbolos sobrescritos envolvidos por parênteses, foram utilizados os símbolos κ e ι , onde o primeiro representa a ordem da derivada parcial em relação ao parâmetro u e o segundo representa a ordem da derivada parcial em relação ao parâmetro v . Assim, as derivadas parciais de superfícies paramétricas não racionais, podem ser escritas como (PIEGL; TILLER, 1997):

$$s_u(u, v) = \sum_{i=0}^{p_u-1} \sum_{j=0}^{p_v} \left(f_{i, p_{u-1}}(u) \cdot g_{j, p_v}(v) \cdot P_{i,j}^{(1,0)} \right) \quad (14)$$

$$s_v(u, v) = \sum_{i=0}^{p_u} \sum_{j=0}^{p_v-1} \left(f_{i, p_u}(u) \cdot g_{j, p_{v-1}}(v) \cdot P_{i,j}^{(0,1)} \right) \quad (15)$$

$$s_{uv}(u, v) = \sum_{i=0}^{p_u-1} \sum_{j=0}^{p_v-1} \left(f_{i, p_{u-1}}(u) \cdot g_{j, p_{v-1}}(v) \cdot P_{i,j}^{(1,1)} \right) \quad (16)$$

onde:

$$P_{i,j}^{(\kappa, \iota)} = \begin{cases} P_{i,j} & \text{se } \kappa = \iota = 0 \\ (p - \kappa + 1) \frac{P_{i+1,j}^{(\kappa-1, \iota)} - P_{i,j}^{(\kappa-1, \iota)}}{t_{u(i+d_u+1)} - t_{u(i+1)}} & \text{se } \kappa > 0 \\ (q - \iota + 1) \frac{P_{i,j+1}^{(\kappa, \iota-1)} - P_{i,j}^{(\kappa, \iota-1)}}{t_{v(j+d_v+1)} - t_{v(j+1)}} & \text{se } \iota > 0 \end{cases} \quad (17)$$

Considerando superfícies B-Splines não racionais, com suas funções de mistura, restrições, considerações e notações já mencionadas, suas derivadas parciais de primeira ordem s_u e s_v , assim como sua derivada s_{uv} , podem ser escritas, respectivamente, da seguinte forma (PIEGL; TILLER, 1997):

$$s_u(u, v) = d_u \sum_{i=0}^{p_u-1} \sum_{j=0}^{p_v} \left(B_{i, d_u-1}(u) \cdot B_{j, d_v}(v) \cdot \frac{(P_{i+1,j} - P_{i,j})}{t_{u(i+d_u+1)} - t_{u(i+1)}} \right) \quad (18)$$

$$s_v(u, v) = d_v \sum_{i=0}^{p_u} \sum_{j=0}^{p_v-1} \left(B_{i, d_u}(u) \cdot B_{j, d_v-1}(v) \cdot \frac{(P_{i,j+1} - P_{i,j})}{t_{v(j+d_v+1)} - t_{v(j+1)}} \right) \quad (19)$$

$$s_{uv}(u, v) = d_u d_v \sum_{i=0}^{p_u-1} \sum_{j=0}^{p_v-1} (B_{i, d_u-1}(u) \cdot B_{j, d_v-1}(v) \cdot C_{(i, j)}) \quad (20)$$

onde: $C_{(i, j)} = \frac{P_{i+1, j+1} - P_{i, j+1} - P_{i+1, j} + P_{i, j}}{(t_{u(i+d_u+1)} - t_{u(i+1)}) \cdot (t_{v(j+d_v+1)} - t_{v(j+1)})}$

Pode-se perceber que a derivada de uma superfície paramétrica é outra superfície paramétrica de menor grau relativo à orientação da derivação. De modo similar, as derivadas parciais de primeira ordem das superfícies apresentadas também podem ser analisadas como infinitas curvas paramétricas – fixando-se um dos parâmetros (e.g. $u = 0$), pode-se criar uma curva paramétrica para cada valor válido do outro parâmetro (e.g. $v \in [0, 1]$). A Equação 21 expressa isto para a derivada parcial de uma superfície B-Spline na direção do parâmetro u onde, através da organização dos elementos, encontra-se a derivada parcial de uma curva paramétrica $C_j(u)$ para cada índice de coluna (j) da matriz de pontos de controle. O mesmo processo pode ser aplicado para a derivada parcial na orientação v .

$$S_u(u, v) = \sum_{j=0}^{p_v} \left(B_{j, d_v}(v) \left(\frac{\partial}{\partial u} \sum_{i=0}^{p_u} \left(B_{i, d_u}(u) P_{i,j} \right) \right) \right) = \sum_{j=0}^{p_v} \left(B_{j, d_v}(v) \left(\frac{\partial}{\partial u} C_j(u) \right) \right) \quad (21)$$

Considerando o objetivo de continuidade na junção de superfícies paramétricas, deve-se analisar as derivadas parciais nas suas bordas, i.e. $s_u(0, v)$, $s_u(1, v)$, $s_v(u, 0)$ e $s_v(u, 1)$. Com a designação de um valor fixo para o parâmetro relacionado à direção da derivada parcial, pode-se perceber – especialmente pela combinação da Equação 14 e da Equação 17 – que há influência de apenas um subconjunto de duas linhas da matriz de pontos de controle, no caso da derivada no sentido do parâmetro u ; ou duas colunas da matriz de controle, no caso da derivada no sentido do parâmetro v .

As quinas das superfícies paramétricas são casos particulares da borda, sendo que em superfícies de Bézier e B-Spline com vetor de nós aberto, as derivadas parciais de primeira ordem nestas posições são influenciadas por apenas dois pontos de controle (PIEGL; TILLER, 1997). As equações a seguir apresentam estes casos especiais:

$$\begin{aligned} s_u(0, 0) &= \frac{d_u}{t_{d_u+1}} (P_{1, 0} - P_{0, 0}) & s_u(1, 1) &= \frac{d_u}{(1 - t_{\omega_u})} (P_{p_u, p_v} - P_{p_u-1, p_v}) \\ s_u(0, 1) &= \frac{d_u}{t_{d_u+1}} (P_{1, p_v} - P_{0, p_v}) & s_u(1, 0) &= \frac{d_u}{(1 - t_{\omega_u})} (P_{p_u, 0} - P_{p_u-1, 0}) \\ s_v(0, 0) &= \frac{d_v}{t_{d_v+1}} (P_{0, 1} - P_{0, 0}) & s_v(1, 1) &= \frac{d_v}{(1 - t_{\omega_v})} (P_{p_u, p_v} - P_{p_u, p_v-1}) \quad (22) \\ s_v(1, 0) &= \frac{d_v}{t_{d_v+1}} (P_{p_u, 1} - P_{p_u, 0}) & s_v(0, 1) &= \frac{d_v}{(1 - t_{\omega_v})} (P_{0, p_v} - P_{0, p_v-1}) \end{aligned}$$

onde: $\omega_u = m_u - d_u - 1$ $\omega_v = m_v - d_v - 1$

A derivada parcial para os demais pontos da borda da superfície depende de mais pontos de controle da região de vizinhança. Isto ocorre devido ao comportamento das funções de mistura univariadas que estão relacionadas com o parâmetro não fixado, no qual a propriedade da partição de unidade é fragmentada em mais de um valor. Isto é, as funções de mistura não apresentam um único valor unitário para um determinado índice, mas diversos valores decimais para um grupo de índices e, a soma desses valores decimais constituem a unidade. Assim, a posição dos pontos de controle pode ser calculada através da resolução de um sistema de equações lineares de modo a garantir as propriedades de continuidade C^1 . Os níveis de continuidade C^2 , e outros mais rigorosos, exigem derivações parciais de maior ordem. Isto implica no fato de que mais pontos de controle possuem alguma influência sobre a continuidade e que as expressões geradas se tornam mais complexas e extensas (PIEGL; TILLER, 1997).

2.3.1 O Problema da Compatibilidade de Torção e os Retalhos de Superfície de Gregory

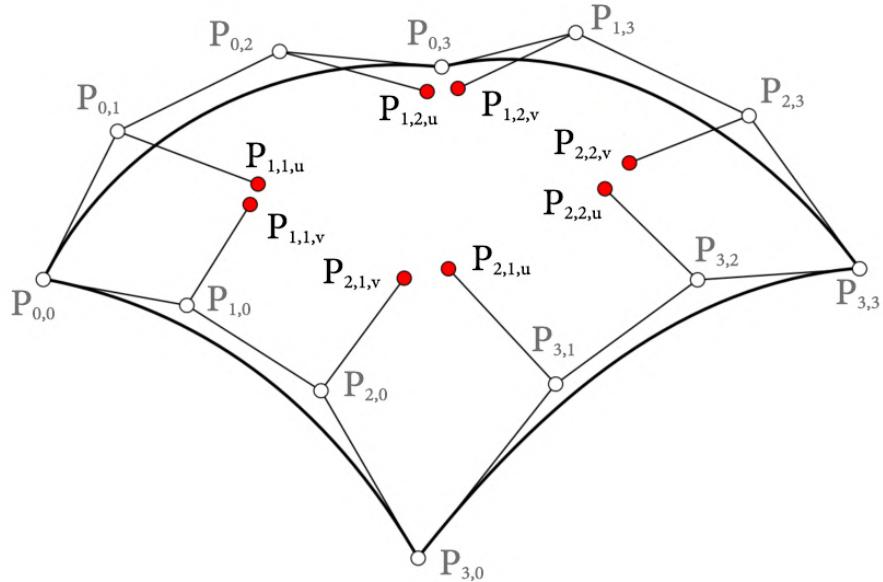
Apesar das fórmulas de derivadas apresentadas, algumas dificuldades adicionais surgem ao realizar a união de múltiplos retalhos de superfícies paramétricas com continuidade. As derivadas parciais – ao longo das linhas coordenadas de u ou de v – nas arestas de borda envolvidas em uma união são utilizadas para o ajuste dos pontos de controle envolvidos na junção com continuidade. No entanto, ao tomar duas uniões, por exemplo, uma em cada direção base de uma das superfícies – uma em u e outra em v – ocorre uma dependência entre duas derivadas direcionais: a derivada parcial misturada $\frac{\partial}{\partial u \partial v} s(u, v)$ ou $\frac{\partial}{\partial v \partial u} s(u, v)$. Considerando uma união com continuidade C^1 , esta dependência delimita a posição de um ponto de controle sob duas igualdades distintas, uma em relação à direção do parâmetro u e outra em direção ao parâmetro v . Esta dependência dupla, todavia, nem sempre pode ser coincidida para as superfícies envolvidas. Este problema é denominado de inconsistência de vértice ou problema da compatibilidade de torção (CHIYOKURA; KIMURA, 1983; HETTINGA; KOSINKA, 2008).

Uma modificação sugerida por Gregory (1974) e posteriormente generalizada e adaptada para retalhos de Bézier por Chiyokura e Kimura (1983) surgiu como uma alternativa para resolver o problema de compatibilidade de torção (HETTINGA; KOSINKA, 2008). O retalho de Gregory, como ficou conhecido, não exige a correspondência entre as derivadas parciais misturadas nas quinas das superfícies que estão sendo unidas, removendo a dependência entre as derivadas direcionais (LOOP et al., 2009). Na proposta deste modelo de retalhos, uma superfície Bézier bicúbica ($d_u = d_v = 3$) é definida normalmente, com exceção dos pontos de controle $P_{1,1}$, $P_{2,1}$, $P_{1,2}$ e $P_{2,2}$, os quais são definidos como pares. Em cada um destes pares, um valor geométrico está vinculado ao parâmetro u e outro ao parâmetro v . Neste trabalho, denominaremos de forma simplificada estes pontos como pontos de Gregory.

A Figura 7 ilustra um exemplo de um retalho bicúbico de Gregory, na qual os pontos de Gregory são destacados, assim como sua relação com um dos parâmetros, podendo ser entendidos como pontos de controle direcionais. Estes pontos são utilizados juntamente de funções de mistura racionais para definir os pontos de controle de uma superfície Bézier na

diagonal das quinas da superfície, isto é: $P_D = \{P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}\}$. Neste caso, os pontos de controle em P_D são definidos pelas seguintes funções (HETTINGA; KOSINKA, 2008):

Figura 7 – Exemplo de um retalho quadrilátero de Gregory como uma superfície Bézier bicúbica.



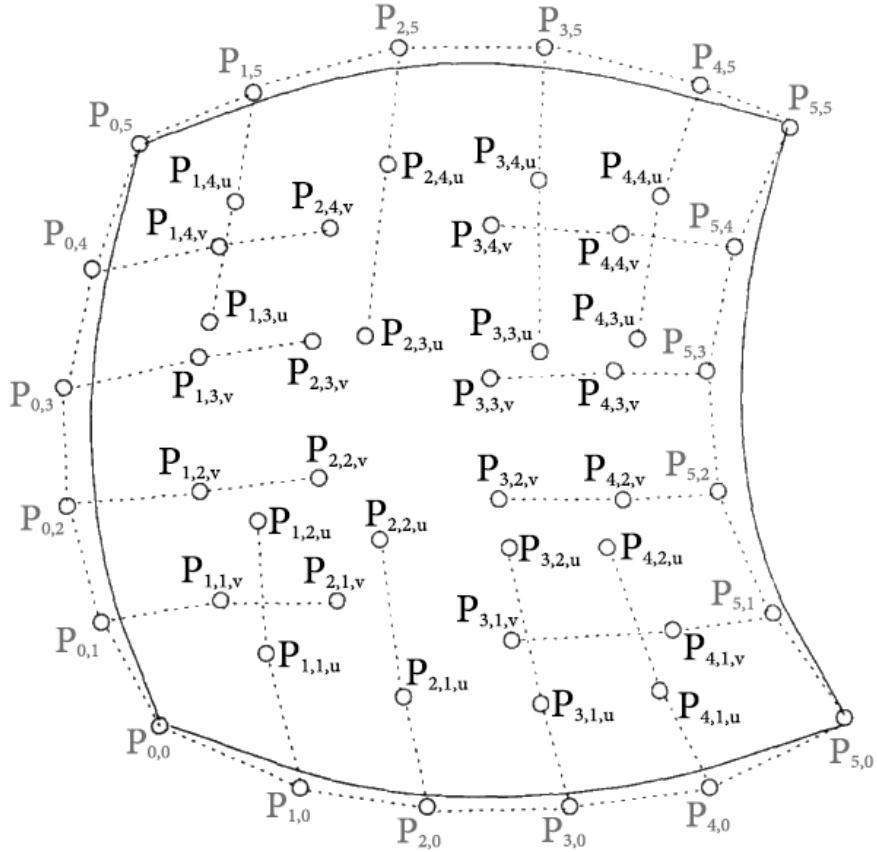
Fonte: Adaptado de Hettinga e Kosinka (2008)

$$\begin{aligned} P_{1,1} &= \frac{v P_{1,1,u} + u P_{1,1,v}}{u+v} & P_{2,1} &= \frac{(1-v) P_{2,1,u} + u P_{2,1,v}}{u+(1-v)} \\ P_{1,2} &= \frac{v P_{1,2,u} + (1-u) P_{1,2,v}}{(1-u)+v} & P_{2,2} &= \frac{(1-v) P_{1,1,u} + (1-u) P_{1,1,v}}{(1-u)+(1-v)} \end{aligned} \quad (23)$$

Através desta forma de representação, os pontos de controle podem ser entendidos como sendo móveis, apresentando uma posição geométrica específica – possivelmente diferente – para cada ponto do domínio paramétrico (LOOP et al., 2009). Com a utilização desta representação, os pares que determinam os pontos de controle em P_D podem ser posicionados de tal modo que uma continuidade C^1 seja obtida entre as superfícies resultantes (HETTINGA; KOSINKA, 2008).

De modo semelhante, uma superfície de Bézier biquística ($d_u = d_v = 5$) pode ser utilizada com as mesmas estratégias a fim de possibilitar junções entre retalhos com uma continuidade C^2 (HERMANN, 1996). A superfície proposta, no entanto, amplia a quantidade de pontos de controle representados por pares. A Figura 8 ilustra um exemplo desta superfície destacando os elementos da matriz de controle. Assim, para este tipo de retalho de superfície existem 16 pares de pontos de controle direcionais que são utilizados em conjunto de uma nova função de mistura racional para definir os pontos de controle regulares da superfície de Bézier, i.e. $P_{i,j}$, onde $1 < i, j \leq 4$. A Equação 24 apresenta esta nova função (HERMANN, 1996):

Figura 8 – Exemplo de um retalho quadrilátero de Gregory sob a forma de uma superfície de Bézier biquântica.



Fonte: Adaptado de Hermann (1996)

$$P_{i,j}(u, v) = \begin{cases} \frac{u^2 \cdot P_{i,j,u} + v^2 \cdot P_{i,j,v}}{u^2 + v^2} & \text{se } (i \in [1, 2]; j \in [1, 2]), \\ \frac{(1-u)^2 \cdot P_{i,j,u} + v^2 \cdot P_{i,j,v}}{(1-u)^2 + v^2} & \text{se } (i \in [3, 4]; j \in [1, 2]), \\ \frac{u^2 \cdot P_{i,j,u} + (1-v)^2 \cdot P_{i,j,v}}{u^2 + (1-v)^2} & \text{se } (i \in [1, 2]; j \in [3, 4]), \\ \frac{(1-u)^2 \cdot P_{i,j,u} + (1-v)^2 \cdot P_{i,j,v}}{(1-u)^2 + (1-v)^2} & \text{se } (i \in [1, 2]; j \in [3, 4]) \end{cases} \quad (24)$$

2.3.2 Junções Integrais e Parciais

As junções entre superfícies paramétricas apresentadas ocorrerem de forma integral, isto é, as bordas envolvidas na união coincidem totalmente. Além disso, os retalhos costumam ser unidos com a utilização dos mesmos parâmetros de forma (Γ). Caso estes valores não sejam compatibilizados ou a junção seja realizada apenas com parte do segmento de borda, é possível que surjam fendas entre as bordas de junção. Visualmente, dependendo de sua intensidade, tais defeitos podem não ser muito significativos por se tratar de uma aproximação de um

objeto. Todavia, este defeito gera uma alteração topológica (criação de um furo) a qual pode ser inapropriada para determinadas aplicações, como simulações ou impressão 3D.

Caso as quantidades de pontos de controle das bordas envolvidas na junção não tenham correspondência, pode-se utilizar operações de inserção de nós ou refinamento de nós (PIEGL; TILLER, 1997). No entanto, os modelos de superfícies Bézier e B-Spline possuem uma estrutura rígida e não permitem a adição de um único ponto de controle, sendo necessário adicionar linhas ou colunas inteiras por necessidade meramente estrutural. Por conta destes impasses, outros modelos de superfícies paramétricas foram sugeridos na literatura, almejando – dentre outros objetivos – adaptações ou modelos que permitem um refinamento local mais eficiente (FORSEY; BARTELS, 1988; SEDERBERG et al., 2003).

Dentre os novos modelos propostos na literatura, como as B-Splines Hierárquicas (FORSEY; BARTELS, 1988) e as B-Splines Hierárquicas Truncadas (GIANNELLI; JÜTTLER; SPELEERS, 2012), as quais permitem a sobreposição de uma cobertura mais refinada sobre uma superfície, destaca-se o modelo de T-Splines – *Tensor Product B-Spline Surfaces with T-junctions* (SEDERBERG et al., 2003). As T-Splines possuem uma estrutura menos rígida de pontos de controle, que viabiliza o mesmo poder de representação de NURBS, mas pode ser mais compacta do que este modelo, possibilitando as junções parciais, denominadas também como junções do tipo T (SEDERBERG et al., 2003).

Se por um lado as T-Splines podem representar o mesmo que NURBS de forma mais compacta, por outro, sua definição global requer o seu recálculo completo a cada alteração no conjunto de pontos de controle, justamente pela flexibilização da sua estrutura. Além disso, diferente das demais propostas, as T-Splines foram patenteadas (número 7.274.364) em 2007, sendo que tecnologias relacionadas ao seu uso foram inicialmente comercializadas pela T-Splines Inc. Em 2011 porém, os direitos de comercialização foram adquiridos pela empresa Autodesk, a qual comercializava esta tecnologia através da ferramenta Rhinoceros – que nasceu como um *plug-in* para o software AutoCAD e posteriormente se tornou um software independente – e passou a ser comercializada de forma integrada na ferramenta Autodesk Fusion 360 (AUTODESK, 2020). Como o vencimento desta patente deve ocorrer apenas em 2024 (SEDERBERG, 2003), tal tecnologia possui limitações de uso e distribuição, as quais devem ser consideradas.

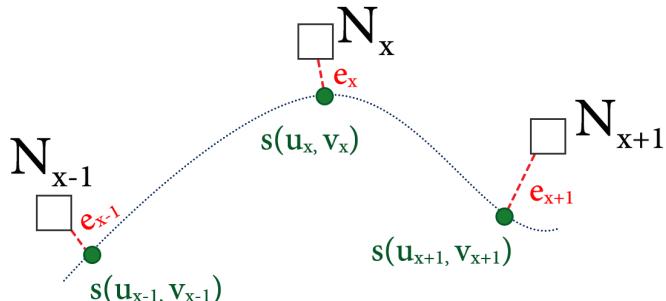
2.4 AJUSTE DE SUPERFÍCIES PARAMÉTRICAS

O ajuste de curvas e superfícies paramétricas é um problema clássico, com vasta literatura em CAD, mas que também se estende a diversos outros domínios como computação gráfica, CAM, modelagem geométrica, realidade virtual, visualização de dados e compressão de dados, para citar alguns (HASEGAWA; TORMENA; PARPINELLI, 2014; COSTA; MONTEMURRO; PAILHÈS, 2018; KRAGIĆ; ČURKOVIĆ; VUČINA, 2018). O objetivo no ajuste de superfícies paramétricas é obter uma superfície que aproxima um conjunto de dados que representa algum objeto ou forma, buscando minimizar o erro de representação e, ao mesmo tempo, ser

representada da forma mais compacta possível (JAVIDRAD, 2012; COSTA; MONTEMURRO; PAILHÈS, 2018).

Dada uma nuvem de pontos $\mathbf{N} = \{N_1, \dots, N_n\}$, composta por n pontos, que representam as amostras de uma superfície de interesse – com possíveis erros – deseja-se encontrar uma superfície paramétrica $s(u, v)$ que passe próxima aos pontos de \mathbf{N} , minimizando o erro quadrático em relação a eles. Para cada ponto N_i de \mathbf{N} , denotados como pontos alvo, deve existir um valor de parâmetro (u_i, v_i) , denotado como parâmetro de localização, que gera a menor distância entre o ponto N_i e o ponto da superfície paramétrica $s(u_i, v_i)$, sendo este último rotulado como *foot point* (JAVIDRAD, 2012; EBRAHIMI; LOGHMANI, 2017). Já a distância entre um ponto alvo e seu respectivo *foot point* é denominada como distância ou erro paramétrico. A Figura 9 ilustra um exemplo de ajuste destacando os elementos mencionados, na qual os pontos alvo são representados pelos quadrados N_{x-1} , N_x e N_{x+1} , os parâmetros de localização são denotados pelos pares (u_{x-1}, v_{x-1}) , (u_x, v_x) e (u_{x+1}, v_{x+1}) , os *foot points* são representados pelos círculos $s(u_{x-1}, v_{x-1})$, $s(u_x, v_x)$ e $s(u_{x+1}, v_{x+1})$, e os erros paramétricos são denotados pelas linhas tracejadas vermelhas e_{x-1} , e_x e e_{x+1} .

Figura 9 – Representação dos principais elementos no processo de ajuste de superfícies paramétricas



Fonte: Elaborado pelo autor (2022).

Considerando o erro residual de ajuste da superfície como a soma das distâncias ou erros paramétricos, o problema de ajuste pode ser formulado como:

$$\min \left(\sum_{i=1}^n e^2 \right) = \min_{\Gamma, \tau} \left(\sum_{i=1}^n \|s_{\Gamma, \tau}(u_i, v_i) - N_i\|^2 \right) \quad (25)$$

A Equação 25 mostra que a minimização ocorre através de dois conjuntos de variáveis: Γ e τ . O primeiro deles representa o conjunto de parâmetros de forma da superfície paramétrica – i.e. seu grau, ordem, função de base, quantidade e posição dos pontos de controle, entre outros – ao passo que a segunda representa o conjunto de valores paramétricos associados aos pontos da nuvem \mathbf{N} , i.e. o conjunto de parâmetros de localização.

De modo genérico, isto é, sem fixar algum valor para as variáveis da equação, o ajuste de curvas e superfícies é um problema de otimização não linear de mínimos quadrados, pois a influência dos valores do vetor de nós (no caso de B-Splines), dos pesos associados a cada ponto

de controle (no caso das superfícies racionais) e dos valores de parametrização, por exemplo, não apresentam uma relação linear com o erro. Este tipo de problema costuma apresentar diversos mínimos locais, sendo um problema que dificilmente é resolvido de forma ótima em tempo aceitável (SHAO; ZHOU, 1996; MADSEN; NIELSEN; TINGLEFF, 2004).

Se por um lado otimizar todas as variáveis atreladas torna o problema difícil de ser resolvido de forma ótima, por outro lado, com a fixação de algumas das variáveis – uso de um vetor de nós de composição e tamanho constante e um determinado conjunto de valores para a parametrização, por exemplo – o problema de ajuste de superfícies pode ser simplificado a um problema de otimização linear de mínimos quadrados. Considerando uma B-Spline não racional paramétrica, pode-se expressar o problema de minimização como:

$$\min_{\mathbf{P}} \left(\sum_{h=1}^n \left\| \left(\sum_{i=0}^{p_u} \sum_{j=0}^{p_v} B_{i, d_u}(u_h) \cdot B_{j, d_v}(v_h) \cdot P_{i, j} \right) - N_h \right\|^2 \right) \quad (26)$$

Como a única variável de otimização é o conjunto de pontos de controle (\mathbf{P}), pode-se minimizar tal equação fixando como zero o valor das derivadas parciais em relação a cada ponto de controle, i.e. $\frac{\partial e^2}{\partial P_{0,0}} = 0, \dots, \frac{\partial e^2}{\partial P_{0,p_v}} = 0, \frac{\partial e^2}{\partial P_{1,0}} = 0, \dots, \frac{\partial e^2}{\partial P_{p_u,p_v}} = 0$, e através disto obter um sistema linear com $(p_u + 1) \cdot (p_v + 1)$ equações (SHAO; ZHOU, 1996).

Outra maneira de estabelecer a mesma minimização pode ser feita através de uma equação matricial. Considerando \mathbf{Q} como o vetor de pontos de controle da superfície paramétrica proveniente do achatamento da matriz de controle, isto é:

$$\mathbf{Q} = [P_{0,0}, \dots, P_{0,p_v}, P_{1,0}, \dots, P_{1,p_v}, \dots, P_{p_u,0}, \dots, P_{p_u,p_v}]^\top = [Q_0, \dots, Q_q]^\top$$

onde: $q = (p_u + 1) \cdot (p_v + 1) - 1$

e tomando \mathbf{E} como o vetor de erros ou resíduos do somatório, \mathbf{N} como o vetor de pontos da nuvem e \mathbf{B} como uma matriz de elementos das funções de mistura da superfície, pode-se reescrever a expressão de erro da seguinte forma:

$$S = \left(\sum_{i=1}^n e^2 \right) = \mathbf{E}^\top \cdot \mathbf{E} = (\mathbf{N} - \mathbf{B} \cdot \mathbf{Q})^\top \cdot (\mathbf{N} - \mathbf{B} \cdot \mathbf{Q}) \quad (27)$$

$$\mathbf{E} = [e_1, \dots, e_n]^\top$$

$$\mathbf{N} = [N_1, \dots, N_n]^\top$$

$$\mathbf{B} = \begin{bmatrix} B_{0, d_u}(u_1) \cdot B_{0, d_v}(v_1) & B_{0, d_u}(u_1) \cdot B_{1, d_v}(v_1) & \dots & B_{p_u, d_u}(u_1) \cdot B_{p_v, d_v}(v_1) \\ B_{0, d_u}(u_2) \cdot B_{0, d_v}(v_2) & B_{0, d_u}(u_2) \cdot B_{1, d_v}(v_2) & \dots & B_{p_u, d_u}(u_2) \cdot B_{p_v, d_v}(v_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_{0, d_u}(u_n) \cdot B_{0, d_v}(v_n) & B_{0, d_u}(u_n) \cdot B_{1, d_v}(v_n) & \dots & B_{p_u, d_u}(u_n) \cdot B_{p_v, d_v}(v_n) \end{bmatrix}$$

Note que na matriz \mathbf{B} cada linha possui os valores das funções de mistura com relação a um respectivo parâmetro, sendo que cada linha é composta por um parâmetro de localização distinto associado aos pontos da nuvem. Como o objetivo é a minimização de S e os pontos de controle

são a única variável de otimização, pode-se fixar como zero o valor da derivada parcial com relação a este vetor \mathbf{Q} , obtendo a expressão matricial (GOLUB, 1965; SHAO; ZHOU, 1996; IGLESIAS et al., 2018):

$$\frac{\partial S}{\partial \mathbf{Q}} = 0 \quad \therefore \quad \mathbf{B}^\top \mathbf{B} \mathbf{Q} = \mathbf{B}^\top \mathbf{N} \quad \therefore \quad \mathbf{Q} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{N} \quad (28)$$

Em ambos os casos de desenvolvimento do problema de minimização, obtém-se um sistema linear. Caso $n > p$, pode-se obter um sistema linear determinado e utilizar algum método de resolução de sistemas lineares – por exemplo, métodos newtonianos, decomposição LU (*Lower-Up*) ou decomposição em valores singulares (*Singular Value Decomposition - SVD*) – para encontrar os valores posicionais ótimos para os pontos de controle de modo a minimizar o erro de ajuste. Assim, a resolução do sistema linear pode ser efetuada com uma complexidade de tempo $\mathcal{O}(np^2)$, onde $\rho = |P| = |Q|$ (GOLUB; LOAN, 1996; EBRAHIMI; LOGHMANI, 2018). Para mais detalhes sobre análise de complexidade, o leitor pode consultar o livro de Cormen et al. (2009). Existe ainda a possibilidade de ocorrência de sistemas lineares mal-condicionados, o que pode implicar em problemas de precisão na computação de soluções (GOLUB, 1965).

Ao utilizar apenas os pontos de controle como variáveis de otimização, garante-se apenas uma solução ótima considerando os valores fixados para as demais variáveis, como o vetor de nós e parametrização. Isto não implica que a solução encontrada seja um ótimo global para o problema, uma vez que a escolha de valores diferentes para as demais componentes da superfície podem conduzir a soluções ainda melhores. Apesar desta simplificação não garantir uma solução ótima global, pode-se iterativamente adaptar as demais variáveis do problema e resolver um sistema linear a cada iteração (SHAO; ZHOU, 1996; AGUILAR et al., 2018).

Embora o objetivo seja a minimização da função de erro apresentada, deve-se levar em consideração que, devido às características usuais de imprecisão das nuvens de pontos, ao minimizar o erro a valores significativamente baixos, pode-se gerar superfícies com oscilações indesejadas, que interpolam os pontos da nuvem. A fim de evitar este tipo de resultado, pode-se utilizar um termo de carenagem, suavização ou regularização, denotado por $F_{fairing}$, para compor a função de erro como um elemento de compensação. Essa função costuma ser usada com a conjectura de que a superfície é contínua, ponderando entre os aspectos de tangência e curvatura da superfície paramétrica (YANG; WANG; SUN, 2004; JAVIDRAD, 2012; EBRAHIMI; LOGHMANI, 2017). A Equação 29 apresenta uma forma de quantificar o erro com este termo de suavização (ZHENG et al., 2012; EBRAHIMI; LOGHMANI, 2017), expondo também uma função de exemplo para este termo, a qual foi utilizada por Ebrahimi e Loghmani (2018).

$$\min \left(\sum_{i=1}^n e^2 \right) = \min_{\Gamma, \tau} \left(\sum_{i=1}^n \|s_\Gamma(u_i, v_i) - N_i\|^2 + F_{fairing} \right) \quad (29)$$

$$F_{fairing} = \iint (s_{uu}^2 + 2s_{uv}^2 + s_{vv}^2) du dv$$

Outra forma de tentar evitar o problema da interpolação dos pontos da nuvem é a utilização de um valor predefinido de erro tolerável referente a alguma métrica de erro qualquer

(GÁLVEZ; IGLESIAS; PUIG-PEY, 2012). Assim, este limiar não precisa estar atrelado à função de minimização, podendo ser comparado com a média de erros paramétricos ou o maior erro paramétrico, por exemplo. Esta alternativa costuma ser utilizada em trabalhos que iterativamente alteram a quantidade de pontos de controle (YANG; WANG; SUN, 2004; MAO et al., 2018; SHANG et al., 2018).

Do ponto de vista de minimização da Equação 25 ou da Equação 26, o ajuste de superfícies costuma ser classificado em três tipos (ZHENG et al., 2012; JAVIDRAD, 2012; EBRAHIMI; LOGHMANI, 2017; EBRAHIMI; LOGHMANI, 2018): minimização de distância de pontos (*Point Distance Minimization* - PDM), minimização de distância tangencial (*Tangent Distance Minimization* - TDM) e minimização de distância quadrática (*Squared Distance Minimization* - SDM). A abordagem de PDM – utilizada principalmente nas áreas de computação gráfica e CAD – é fundamentada na utilização das distâncias quadráticas entre os pontos N_i de uma nuvem de pontos e os seus respectivos *foot points* $s(u_i, v_i)$. Os métodos TDM – utilizados principalmente pela comunidade de visão computacional – são similares, mas se baseiam no uso das distâncias quadráticas entre os pontos N_i e as linhas tangentes sob os respectivos *foot point* $s(u_i, v_i)$. Por fim, os métodos SDM são baseados na curvatura, sendo utilizada uma aproximação quadrática das distâncias quadráticas entre os pontos N_i e uma superfície de ajuste.

2.4.1 Métricas de Erro

Com o intuito de avaliar o resultado de um ajuste de superfície, uma ou mais métricas podem ser utilizadas para quantificar algum critério da representação da geometria resultante. Em problemas de aspecto contínuo, o índice integral do erro quadrático (*Integral Square Error* – ISE) é utilizado como um meio de quantificar o erro de ajuste. No contexto discreto, tal índice costuma ser representado como o somatório das distâncias quadráticas entre as amostras N_i de uma nuvem de pontos \mathbf{N} , composta por n elementos, e os seus pontos associados no modelo paramétrico ajustado $s(u_i, v_i)$, sendo o par (u_i, v_i) o parâmetro da superfície relacionado à i -ésima amostra da nuvem (AFSHAR; SORYANI; RAHMANI, 2011):

$$\text{ISE}(\mathbf{N}, s) = \sum_{i=1}^n (\|N_i - s(u_i, v_i)\|)^2 \quad (30)$$

Uma característica do índice ISE, neste contexto discreto, é que o tamanho da nuvem de pontos, isto é, a quantidade de pontos que a constitui, influencia diretamente em seu valor. Caso isto não seja desejável, existem as alternativas do índice de erro quadrático médio (*Mean-Square Error* – MSE) e o índice de raiz quadrada do erro quadrático médio (*Root-Mean-Square Error* – RMSE), as quais ponderam a quantidade de pontos na nuvem analisada (GÁLVEZ; IGLESIAS, 2013; DUNG; TJAHIJOWIDODO, 2017). Estas métricas são calculadas como:

$$\text{MSE}(\mathbf{N}, s) = \frac{\text{ISE}(\mathbf{N}, s)}{n} \quad \text{RMSE}(\mathbf{N}, s) = \sqrt{\text{MSE}(\mathbf{N}, s)} \quad (31)$$

Além dos erros integrais ou médios, outro índice de mensuração utilizado é o erro máximo (*Maximum Error* – ME), o qual destaca o maior erro paramétrico entre o resultado e a nuvem de

pontos (DUNG; TJAHHOWIDODO, 2017).

$$\text{ME}(\mathbf{N}, s) = \max_{i \in [1, n]} \|N_i - s(u_i, v_i)\| \quad (32)$$

Ressalta-se que os quatro índices apresentados: ISE, MSE, RMSE e ME, não são diretamente comparáveis entre si devido a diferente natureza de suas grandezas. Além disso, a comparação através de um mesmo índice relacionado a diferentes nuvens, também não pode ser realizada, pois tais índices são dependentes de escala. Estas métricas também desconsideram a influência da simplicidade de representação, na qual a quantidade de pontos de controle utilizados não possui influência nestas métricas. Deste modo, melhores métricas de erro podem ser alcançadas à custa de uma maior quantidade de variáveis.

Com o intuito de ponderar a quantidade de variáveis envolvidas, as métricas *Akaike Information Criterion* (AIC) e *Bayesian Information Criterion* (BIC) também são utilizadas na literatura no contexto de ajuste de curvas e superfícies paramétricas. Ambos são critérios teóricos utilizados na busca por um melhor modelo de aproximação para dados verdadeiros, ponderando entre fidelidade e simplicidade. No ajuste de curvas e superfícies, estas métricas são formalizadas como (YOSHIMOTO; MORIYAMA; HARADA, 1999; GÁLVEZ; IGLESIAS, 2013):

$$\text{AIC} = n \log_e(E) + 2\kappa \quad \text{BIC} = n \log_e(E) + \kappa \log_e(n) \quad (33)$$

onde n representa o número de amostras da nuvem de pontos, E representa o erro de ajuste e κ representa a quantidade de parâmetros de otimização utilizada.

Outra forma comum de avaliação de qualidade dos resultados obtidos, porém subjetiva, é a comparação visual. Através deste tipo de inspeção pode-se detectar problemas como sobreposição de superfície, problemas de topologia e detectar mais facilmente quais características da superfície não foram corretamente ajustadas, especialmente se existe um modelo real a ser comparado (BERGER et al., 2013). Mapas de calor também são utilizados como um meio de observar visualmente a distribuição dos erros locais de ajuste (KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018; ĆURKOVIĆ; ĆURKOVIĆ; VUČINA, 2018). Esta técnica de visualização de dados pode simplificar a identificação dos principais problemas na superfície ajustada.

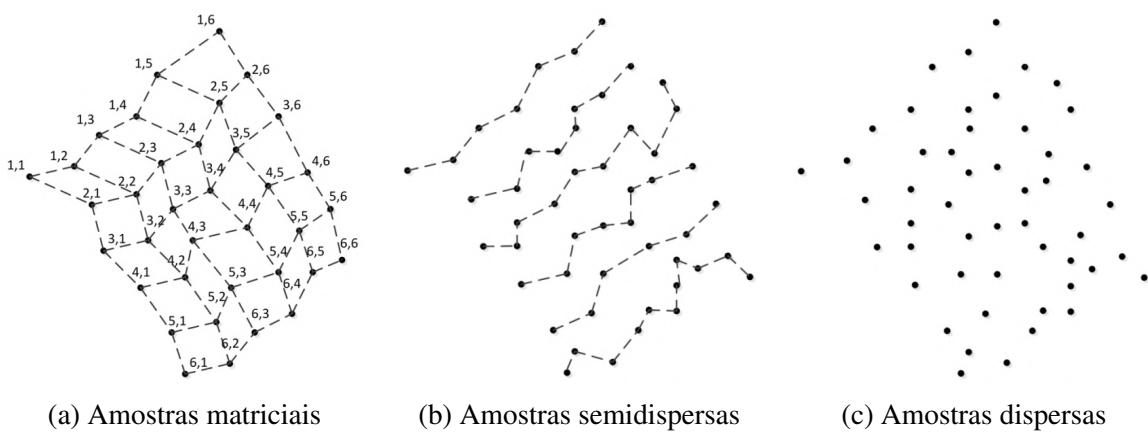
Além das métricas tradicionais para a avaliação do ajuste, existem também outras técnicas que ainda não são amplamente utilizadas na literatura. Cignoni, Rocchini e Scopigno (1998), por exemplo, apresentam um método (*Metro*) para calcular a diferença entre malhas triangulares, através de uma abordagem de Monte Carlo. O *Metro* gera amostras proporcionais às áreas da malha triangular da superfície e calcula as menores distâncias destas amostras até o objeto de referência. Assim, pode ser utilizado para estimar o erro no ajuste de superfícies paramétricas, desde que exista um modelo virtual de referência do objeto digitalizado e não apenas sua nuvem de pontos. De modo semelhante, mas voltado a superfícies suaves, Berger et al. (2013) apresentam um método com a mesma estratégia para comparar superfícies paramétricas com superfícies implícitas. A partir do mapeamento das menores distâncias são calculadas quatro medidas: a distância de Hausdorff, a distância média, e seus análogos em relação à diferença entre as normais dos pontos mapeados.

2.4.2 Características das Nuvens de Pontos

No ajuste de curvas ou superfícies é fundamental considerar quais tipos de informação estão disponíveis na nuvem de pontos. No mínimo, as amostras devem possuir os dados geométricos, mas outros dados também podem acompanhar estes valores. Por exemplo, cada ponto pode vir acompanhado de um vetor normal, que pode ser útil na inferência de curvatura, ou um valor referente à cor do ponto, que pode ser útil para o processo de registro e geração de textura. Informações do dispositivo de digitalização utilizado também podem acompanhar uma nuvem de pontos, através das quais se pode estimar e prever possíveis problemas característicos no processo de captura (FRÉDÉRIC; JOACHIM, 2004; BERGER et al., 2017).

Outra informação relevante, que é inclusive utilizada como um parâmetro de classificação, é a sequência ou vizinhança dos pontos. Uma nuvem de pontos é dita organizada se existe alguma informação de vizinhança entre os pontos e é denominada como não organizada caso não apresente relação alguma de ordem ou organização. As nuvens organizadas podem ser distinguidas ainda em duas categorias de acordo com a forma em que os dados estão dispostos. Eles podem estar organizados em linhas e colunas, na forma de uma matriz, ou eles podem estar semidispersos (*quasi-scattered*), existindo apenas o dado da ordem na leitura dos pontos que pode ser utilizado para agrupá-los em linhas. Nuvens de pontos semidispersas costumam ser produzidas por dispositivos de digitalização que fazem a coleta de amostras em linhas de leitura, isto é, por seções transversais (JAVIDRAD, 2012). Este tipo de nuvem de pontos é comum em problemas envolvendo ramos da medicina e engenharia biomédica (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012). Devido a sua estrutura, nuvens organizadas podem ser mais facilmente ajustadas por superfícies, uma vez que já existe alguma informação intrínseca de vizinhança (RUSU; COUSINS, 2011; JAVIDRAD, 2012; MAO et al., 2018). A Figura 10 ilustra este tipo de classificação entre as nuvens de pontos, na qual as linhas tracejadas indicam a relação de um dado de vizinhança entre duas amostras.

Figura 10 – Classificação de nuvens de pontos de acordo com a distribuição das amostras

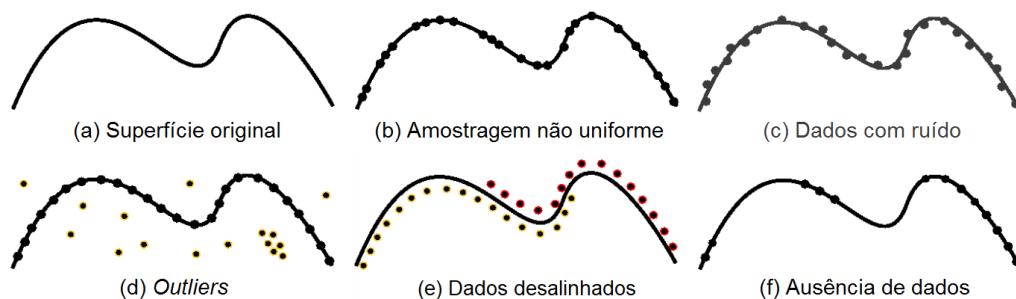


Fonte: Adaptado de Mao et al. (2018)

Existem características e problemas que são comuns aos processos de digitalização e

geração de nuvem de pontos, em maior ou menor proporção, sendo possível antecipar que dados errados implicam na geração de resultados errados. Entender os principais problemas das nuvens de pontos é importante para que filtros e algoritmos de ajuste possam combater as imperfeições previstas, minimizando os erros no resultado final (BERGER et al., 2017). As imperfeições mais comuns nas nuvens de pontos adquiridas, quanto ao aspecto geométrico e decorrentes dos problemas práticos na sua coleta, são representadas em 2D na Figura 11, e são as seguintes (VÁRADY; MARTIN; COX, 1997; BERGER et al., 2017):

Figura 11 – Características usuais nas nuvens de pontos – representação 2D



Fonte: Adaptado de Berger et al. (2017)

- Amostragem não uniforme: caracterizada pela existência de má distribuição dos pontos na nuvem – regiões muito densas e regiões com poucas amostras. A densidade da amostragem é importante para uma boa reconstrução das geometrias locais de uma superfície, formando uma vizinhança de pontos bem definida. Pode ser tratado através da redução de amostragem onde há muitos pontos e do aumento de amostragem onde existem poucos pontos (ALEXA et al., 2003);
- Ruído: caracterizado por pontos que não representam fielmente as superfícies do objeto real, mas estão muito próximos. Pode ocorrer tanto em relação ao aspecto posicional, quanto ao vetor normal dos pontos da nuvem (KAZHDAN; BOLITHO; HOPPE, 2006). Costuma-se tratar este problema através de filtros, ajustes por aproximação ou através de técnicas de suavização da superfície resultante;
- Outliers*: problema caracterizado por pontos que não representam e estão longe das superfícies do objeto real. Geralmente possuem densidade menor em relação às amostras regulares. Este tipo de amostragem costuma ser descartado com a utilização de técnicas que fazem uso da relação entre funções de distância e probabilidade de medidas antes do ajuste de superfícies (CIGNONI et al., 2008; BOISSONNAT; CHAZAL; YVINEC, 2017). Técnicas de ajuste por aproximação ajudam a amenizar os reflexos deste tipo de problema na superfície gerada;
- Dados desalinhados: problema caracterizado por pontos deslocados sobre as superfícies do objeto real. Ocorre principalmente por imperfeições no processo de registro mas, também podem ocorrer devido a erros de calibração no caso de digitalização de objetos reais (ZHANG; XU, 2018);

- e) Ausência de dados: caracterizada pela inexistência de amostras em determinadas regiões, criando uma lacuna entre uma sequência de pontos. Pode-se tentar inferir pontos fictícios adicionais para essas regiões, interpolando entre os dados geometricamente mais próximos, mas este é um dos problemas mais complicados a se tratar, exatamente pela característica de ausência de informação (BERGER et al., 2014; ZOLLHÖFER et al., 2018).

A classe de algoritmos aproximadores costuma apresentar maior tolerância a dados ruidosos, podendo gerar resultados satisfatórios mesmo na presença deste problema (BERGER et al., 2017; BOISSONNAT; CHAZAL; YVINEC, 2017). Este tipo de métodos apresenta também um maior grau de flexibilidade de reconstrução, produzindo eventualmente resultados satisfatórios, mesmo com a existência de outros problemas (até certo nível) como eventual falta de uniformidade e incompletude (BERGER et al., 2013).

2.4.3 Benchmarks no Ajuste de Superfícies

Por vezes, algoritmos e métodos computacionais podem ter sua eficiência garantida e comprovada matematicamente. No caso de problemas de reconstrução de curvas e superfícies, entretanto, isso geralmente não é possível devido aos problemas e as características do processo. Para estes casos e em situações similares, a avaliação de qualidade dos resultados de um algoritmo é usualmente verificada através de estudos empíricos com *benchmarks* e métricas bem definidas e utilizadas na literatura. Deste modo é possível testar a eficiência e eficácia dos algoritmos, tentando entender melhor o desempenho relativo, suas vantagens e desvantagens (ZHENG et al., 2012). Apesar de não atestarem a eficiência geral do algoritmo testado, através da realização de experimentos é possível verificar o comportamento e características do algoritmo para determinados casos e contextos.

Os *benchmarks* de ajuste de superfícies podem ser classificados através da natureza da nuvem de pontos utilizada, a qual pode ser real ou sintética. As nuvens de pontos reais são capturadas através de algum dispositivo ou procedimento de digitalização. Com a vasta e diversa demanda por digitalização 3D, existem repositórios públicos que oferecem diferentes tipos de dados, como por exemplo o repositório de escaneamento 3D de Stanford¹, que disponibiliza modelos densos e de complexidade diversificada e o arquivo digital de Michelangelo² (LEVOY et al., 2000), um repositório com digitalizações de alta resolução das obras de Michelangelo. Com a maior disponibilidade e acessibilidade de dispositivos de digitalização RGB-D, os quais capturam imagens com informação de profundidade de cada pixel, repositórios compostos por este tipo de digitalização também já estão disponíveis. Para citar alguns exemplos, um repositório é indicado por Choi et al. (2016)³ (mais de 10.000 digitalizações de objetos do cotidiano) e

¹ The Stanford 3D Scanning Repository homepage: <<http://graphics.stanford.edu/data/3Dscanrep/>>

² The Digital Michelangelo Project Archive homepage: <<http://graphics.stanford.edu/data/dmich-public/>>

³ A Large Dataset of Object Scans homepage: <<http://redwood-data.org/3dscan/index.html>>

outro por Lai et al. (2011)⁴ (300 objetos digitalizados em ambiente controlado) e o repositório BigBIRD⁵ indicado por (SINGH et al., 2014) (125 objetos com fotos de alta resolução, nuvens de pontos RGB-D individuais e registradas, malhas de reconstrução e mais).

Apesar da criação de nuvens de pontos reais estar mais acessível atualmente, para fins de estudo comparativo é importante a utilização de modelos bem conhecidos da área. Neste sentido, existem repositórios que fornecem conjuntos de nuvens de pontos que podem ser utilizados para propósitos de avaliação de reconstrução. Entretanto, tais repositórios não costumam oferecer os modelos exatos (verdadeiros) dos objetos digitalizados, pois tratam-se de objetos do mundo real e a mensuração de suas superfícies pode ser complicada e passível de erros de medição, dificultando ou afetando a avaliação da qualidade de digitalização.

Devido aos problemas e dificuldades advindos do escaneamento de objetos reais, a utilização de superfícies e nuvem de pontos sintéticas é comum na realização de experimentos na área de ajuste (LEE, 2002; GÁLVEZ; IGLESIAS; PUIG-PEY, 2012; BERGER et al., 2013). Deste modo, é possível definir um objeto virtual, com todas as suas medidas conhecidas e avaliar o resultado de ajuste com precisão. Nestes casos, a nuvem de pontos é obtida através da superfície original, a qual usualmente é representada por meio de equações implícitas e paramétricas, ou através de uma malha triangular.

Nuvens de pontos geradas por equações podem amostrar superfícies suaves de maneira precisa, podendo representar diversos aspectos geométricos. Nuvens geradas desta forma podem ser relativamente simples (LEE, 2002; MAO et al., 2018), mas também podem representar formas mais rebuscadas (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012; MAO et al., 2018). Estas nuvens de pontos costumam representar um único objeto, relativamente pequeno e com pouca variação de detalhes geométricos. Em contrapartida, nuvens de pontos geradas a partir dos vértices de malhas triangulares podem representar praticamente qualquer tipo de objeto de maneira mais simples do que equações. A discretização de objetos em uma malha de triângulos, todavia, possui limitações na representação de aspectos suaves. Diversos trabalhos (KINERI et al., 2012; WANG; ZHENG, 2013; ZHANG et al., 2016; BERTOLINO et al., 2021) utilizam este tipo de nuvem de pontos.

Existem também estudos dirigidos à geração de nuvens sintéticas, através de amostragem ou simulação do processo de captura. Kazhdan (2005), Manson, Petrova e Schaefer (2008) e Sübmuth, Meyer e Greiner (2010), por exemplo, utilizam modelos com malha triangular como referência e geram nuvens de pontos sintéticas através da aplicação de amostragem aleatória de pontos sobre as superfícies dos triângulos. Buscando por uma semelhança mais próxima à digitalização real, uma superfície sintética é submetida a um processo de digitalização virtual, gerando uma nuvem sintética que pode conter inclusive os problemas típicos de digitalização. Hoppe et al. (1992) e Haar et al. (2005) produzem nuvens de pontos sintéticas a partir de malhas triangulares utilizando algoritmos de *ray tracing* ou *z-buffering* sobre a malha, simulando o processo de digitalização com dispositivos baseados em visão, laser ou tecnologias similares.

⁴ *RGB-D Object Dataset homepage:* <<http://rgbd-dataset.cs.washington.edu/index.html>>

⁵ *Big Berkeley Instance Recognition Dataset homepage:* <<http://rll.eecs.berkeley.edu/bigbird/>>

Essa abordagem produz dados mais realísticos, podendo ser utilizada para testar e avaliar a qualidade de algoritmos de reconstrução e de registro.

Se por um lado nuvens de pontos obtidas por um processo de digitalização real são interessantes para avaliar o comportamento e as capacidades de um algoritmo, por outro lado, a utilização de nuvens sintéticas permite a realização de avaliação de qualidade da reconstrução de forma mais simples e precisa. Ambos os tipos de dados são utilizados na literatura, sendo que os dados sintéticos costumam ser mais utilizados nas comparações entre métodos.

2.4.4 Considerações sobre Parametrização

A parametrização é o processo em que se mapeia uma nuvem de pontos – no espaço geométrico – a um conjunto de valores escalares – no espaço de parâmetros – por meio de um conjunto de funções. Estes valores escalares, ou valores paramétricos, definem a organização das amostras da nuvem de pontos e podem ser entendidos como o tempo que se leva para o deslocamento da origem da curva ou superfície até o ponto determinado. Nesta abordagem, considerando o ajuste de superfície paramétrica, cada ponto N_i de uma nuvem de pontos \mathbf{N} é associado a um par de valores paramétricos (u_i, v_i) no vetor de parâmetros de localização τ (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012; KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). Assim, o problema de parametrização está relacionado aos dados de organização da nuvem, sendo que no caso de nuvens não organizadas, deve-se determinar pelo menos uma relação de vizinhança entre as amostras.

Nos casos de nuvens de pontos organizadas ou quando é possível calcular uma organização, três métodos primários de parametrização costumam ser mencionados e utilizados na literatura: uniforme, comprimento de corda e centrípeto. O método uniforme é o mais simples entre os três, atribuindo valores igualmente espaçados ao vetor de parâmetros de localização τ . Esta estratégia costuma não produzir resultados bons, uma vez que as nuvens de pontos costumam apresentar dados com espaçamento variado. Já os métodos de comprimento de corda e centrípeto utilizam a distância entre os pontos da nuvem como referência para o espaçamento dado aos valores de τ , sendo que ambos apresentam resultados similares (PIEGL; TILLER, 1997; HASEGAWA; TORMENA; PARPINELLI, 2014; DUNG; TJAHIJOWIDODO, 2017). A Equação 34, proposta por Lee (1989), generaliza estes métodos de parametrização para o caso de curvas paramétricas atribuindo valores normalizados no intervalo $[0, 1]$ ao vetor τ , sendo que n representa o número de pontos da nuvem e N_i representa o i -ésimo ponto da nuvem:

$$\begin{aligned} u_1 &= 0 \\ u_k &= u_{k-1} + \frac{\|N_i - N_{i-1}\|^\xi}{\sum_{j=2}^n \|N_j - N_{j-1}\|^\xi} \quad , 1 \leq k \leq m \end{aligned} \tag{34}$$

Através da Equação 34, pode-se obter o modelo uniforme utilizando $\xi = 0$, o modelo de comprimento de corda utilizando $\xi = 1$ ou o modelo centrípeto utilizando $\xi = 0,5$. Esta mesma equação pode ser estendida para o caso de superfícies paramétricas, aplicando-a isoladamente para os valores de u e v no vetor de parametrização τ .

O uso isolado dos métodos primários de parametrização, entretanto, não costuma ser apropriado para nuvens de pontos com características complexas (BARHAK; FISCHER, 2001). Nesse sentido, existem trabalhos que propõem o uso de abordagens iterativas, nas quais inicia-se o vetor τ com uma estimativa de parâmetros e tenta-se reduzir o erro de aproximação (Equação 25) a cada iteração do algoritmo. Este tipo de operação é denominado como reparametrização, já que a associação dos valores paramétricos com as amostras da nuvem de pontos é alterada. Note que a reparametrização em si não altera a geometria da curva ou superfície, sendo necessário um novo ajuste por minimização, por exemplo, para efetivamente alterar a geometria (HASEGAWA; TORMENA; PARPINELLI, 2014; EBRAHIMI; LOGHMANI, 2017; EBRAHIMI; LOGHMANI, 2018; UEDA et al., 2020).

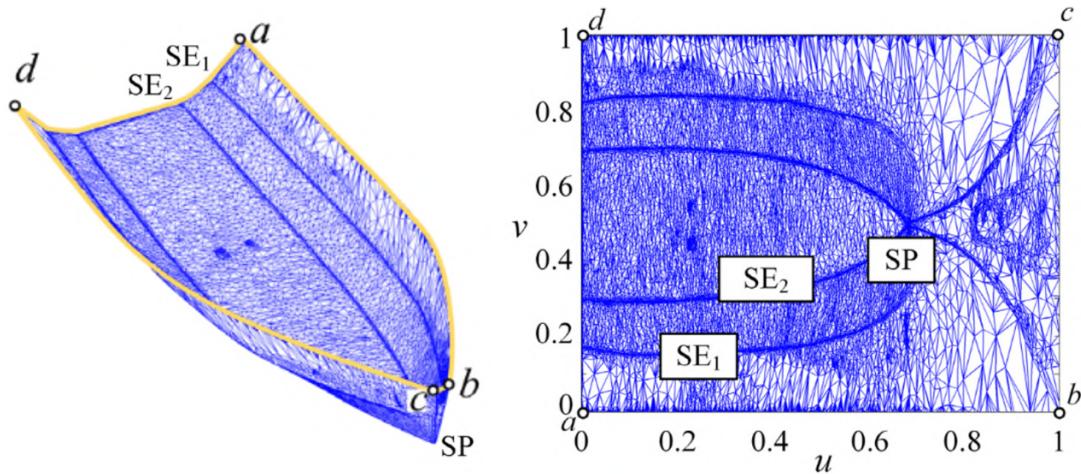
As abordagens iterativas que empregam reparametrização costumam utilizar estratégias baseadas em gradiente, heurísticas ou meta-heurísticas. Os métodos baseados em gradientes podem apresentar uma rápida convergência, mas requerem uma estimativa inicial para o conjunto de parâmetros τ e, dependendo do valor utilizado, podem apresentar problemas na convergência, ficando presos em ótimos locais. Por outro lado, heurísticas e meta-heurísticas costumam explorar mais o espaço de busca, tendo maior probabilidade de convergência a mínimos globais. Todavia, estes métodos, especialmente as meta-heurísticas, costumam requerer maior número de iterações e tempo de execução, sendo computacionalmente mais custosos (RODRÍGUEZ, 2017; COSTA; MONTEMURRO; PAILHÈS, 2018; DUNG; TJAHIJOWIDODO, 2017; ĆURKOVIĆ; ĆURKOVIĆ; VUČINA, 2018).

Quando a nuvem de pontos não é organizada e não é possível distribuir seus elementos de forma matricial, é necessário determinar pelo menos uma relação de vizinhança entre as amostras da nuvem. Isto pode ser feito através de um procedimento de triangulação das amostras, gerando uma ou mais malhas triangulares. Não faz parte do escopo deste trabalho estudar as formas de triangulação de uma nuvem de pontos mas, destacam-se dois algoritmos tradicionais da literatura: o algoritmo de *alpha shapes* (ou α -shapes), um dos primeiros métodos utilizados na área de reconstrução de superfícies (EDELSBRUNNER; KIRKPATRICK; SEIDEL, 1983), e o algoritmo de pivotamento de bola – *Ball-Pivoting Algorithm* – que é conceitualmente simples e eficiente (BERNARDINI et al., 1999).

As técnicas de parametrização de superfícies com base em uma malha triangular têm suas raízes na área de computação gráfica, especialmente nas aplicações de mapeamento de textura sobre um modelo 3D (REMACLE et al., 2010). Tendo uma malha triangular, na qual os vértices são os pontos da nuvem N , o processo de parametrização pode ser realizado como um mapeamento um-para-um, no qual valores paramétricos únicos são atribuídos a cada ponto de N . No caso de superfícies, o mapeamento deve ocorrer sobre \mathbb{R}^2 pois, cada elemento (u_i, v_i) do conjunto de parâmetros é constituído por uma tupla de dois elementos, e deve-se ainda manter um aspecto quadrilateral devido às propriedades das funções de base das superfícies paramétricas (FLOATER; HORMANN, 2005; REMACLE et al., 2010). Isto é, no caso da parametrização de superfícies é necessário definir os pontos de borda, sendo que esta escolha pode influenciar

significativamente no processo (KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). A Figura 12 mostra um exemplo do mapeamento de uma nuvem de pontos triangulada em um plano que representa o espaço paramétrico unitário. Foram sinalizadas sete referências – os pontos a , b , c e d , as bordas de ângulo agudo SE_1 e SE_2 , e a cúspide SP – para facilitar a correspondência entre a nuvem de pontos e os pontos no espaço de parâmetros.

Figura 12 – Exemplo de parametrização de superfície com mapeamento um-para-um



Fonte: Kragić, Ćurković e Vučina (2018)

O procedimento de mapeamento pode ser único e direto, projetando os pontos em um vetor ou plano paramétrico, ou pode ser realizado de maneira indireta, sendo realizado inicialmente um mapeamento para um espaço k -dimensional com uma topologia qualquer – como esferas ou hiper-cubos – e posteriormente efetuando um novo mapeamento para as variáveis paramétricas da superfície (KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). Indiferentemente desta escolha, durante a associatividade da malha geométrica com o espaço de parâmetros, as técnicas de parametrização de superfícies por mapeamento quase sempre geram distorções nos ângulos e nas áreas dos triângulos. Isso ocorre porque mapeamentos isométricos – i.e. que preservam as distâncias entre os pontos – só ocorrem em casos específicos, quando é possível preservar tanto os ângulos como as áreas triangulares envolvidas. Por consequência, a topologia da curva ou superfície reconstruída pode não ser preservada durante o processo de parametrização (BARHAK; FISCHER, 2001; DESBRUN; MEYER; ALLIEZ, 2003; FLOATER; HORMANN, 2005; LAI; HU; POTTMANN, 2006).

Como um mapeamento isométrico pode não ser viável, os algoritmos de parametrização de superfícies buscam então objetivos intermediários. Uma projeção conforme (*conformal map*) procura preservar os ângulos originais – i.e. minimizar a sua distorção – à custa das áreas dos triângulos. Já uma projeção equiareal, também denominada de homolográfica ou *authalic*, procura preservar as áreas originais dos triângulos à custa de seus ângulos. Outros métodos podem optar ainda por um balanceamento entre estes dois aspectos (DESBRUN; MEYER; ALLIEZ, 2003; FLOATER; HORMANN, 2005).

Outros algoritmos de parametrização utilizam abordagens envolvendo a minimização de energia. Nestes métodos, uma borda deve ser fornecida de alguma forma – seja manualmente escolhida ou através de algum algoritmo – e a solução é calculada com base na resolução de sistemas de equações lineares (DESBRUN; MEYER; ALLIEZ, 2003; KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). O método *spring-mesh mapping*, por exemplo, transforma a malha da nuvem de pontos em uma malha de molas e gera uma solução de parametrização através da tentativa de alcançar as posições de equilíbrio para cada ponto, minimizando a energia do sistema (KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). O método de coordenadas de valor médio, ou *Mean Value Coordinates* (FLOATER, 2003; FLOATER; KóS; REIMERS, 2005) otimiza cada vértice interno do espaço de parâmetros como uma combinação convexa dos seus vértices vizinhos utilizando o teorema de valor médio ou teorema de Lagrange. Existem também as abordagens de mapeamento harmônico que são baseadas na minimização da energia de Dirichlet, minimizando a deformação da malha (REMACLE et al., 2010; KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018). Os principais métodos de mapeamento utilizados como parametrização, no entanto, apresentam a significativa limitação de que só podem ser aplicados para superfícies que podem ser decompostas em retalhos sem dobras – isto é, uma planificação sem sobreposição. Para os casos nos quais isto não é possível, a utilização de múltiplos retalhos pode ser uma solução (BERTOLINO et al., 2021).

2.4.5 Considerações sobre Ajuste

O problema de ajuste de superfícies paramétricas é um difícil problema de otimização não linear, contínuo, multivariável e multimodal. É dito ser contínuo por apresentar a característica de que uma alteração nos valores de parâmetros ou na posição dos pontos de controle afeta de maneira proporcional o resultado obtido, não havendo um salto descontínuo nestes casos. É dito ser multivariável pois usualmente envolve um número considerável de variáveis desconhecidas. Por fim, é dito multimodal devido à característica da função objetivo – a minimização do erro entre a superfície gerada e os pontos da nuvem – que pode apresentar múltiplas soluções locais e globais (GÁLVEZ; IGLESIAS, 2013)

De forma geral, as características visadas durante o processo de reconstrução de superfície costumam ser: maior velocidade, baixo consumo de memória, uso de aproximação do modelo paramétrico a partir dos dados ao invés de interpolações, representação fiel de características como detalhes pontiagudos ou suavidade, e robustez contra as possíveis imperfeições durante a aquisição da nuvem, como oclusão e baixa amostragem (OHTAKE et al., 2003).

Os métodos de reconstrução de superfícies encontrados na literatura podem ser subdivididos em duas macro abordagens de resolução. Em uma, a nuvem de pontos é ajustada por uma ou múltiplas superfícies paramétricas, ao passo que na outra, múltiplas curvas paramétricas são ajustadas e posteriormente combinadas a fim de gerar a superfície resultante. Esta última macro abordagem é denominada de *Surface Skinning* e é utilizada em nuvens de pontos organizadas em matriz ou semidispersas e que representam objetos com uma topologia que possa ser representada desta forma (LI et al., 2015; MAO et al., 2018).

Independentemente da macro abordagem adotada, existem técnicas de ajuste que requerem a parametrização das amostras da nuvem de pontos e outras que a omitem. As técnicas sem parametrização utilizam a estratégia de mover iterativamente uma curva ou superfície em direção aos pontos da nuvem, a partir de uma aproximação inicial, a qual é fornecida pelo usuário ou computada através de algum método (POTTMANN; LEOPOLDSEDER; HOFER, 2002; YANG; WANG; SUN, 2004). A movimentação iterativa do modelo geométrico ocorre através de um procedimento de otimização baseado na minimização da Equação 25, adaptando a localização e quantidade de pontos de controle. A velocidade de convergência deste tipo de método está relacionada ao número e a posição inicial definida para os pontos de controle, sendo necessário que a forma inicial seja semelhante à forma objetivo (POTTMANN et al., 2005; JAVIDRAD, 2012; EBRAHIMI; LOGHMANI, 2018; KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018).

Ainda não existe um método padrão para especificar uma estimativa inicial para o número de pontos de controle a ser utilizados. Assim, a definição deste valor costuma ser realizada de forma empírica, como um parâmetro de entrada (EBRAHIMI; LOGHMANI, 2018). A escolha manual desta quantidade e até mesmo outras variáveis envolvidas pode ser difícil e tediosa, exigindo várias tentativas até se obter um resultado desejável. Um número adequado de pontos de controle é específico para cada caso, dependendo da quantidade e do aspecto das amostras da nuvem de pontos. Este valor desempenha um papel fundamental na relação entre qualidade e simplicidade de um ajuste de superfície (HASEGAWA; TORMENA; PARPINELLI, 2014). Considerando estes aspectos, uma escolha adaptativa deste valor é recomendada, permitindo maior flexibilidade na representação das formas (GÁLVEZ; IGLESIAS, 2013; COSTA; MONTEMURRO; PAILHÈS, 2018; EBRAHIMI; LOGHMANI, 2018)

A definição iterativa da quantidade de pontos de controle costuma ser feita por heurísticas, meta-heurísticas, ou com base em operações de inserção ou remoção de nós (WANG; ZHENG, 2013; BERTOLINO et al., 2021). Com este tipo de abordagem, os resultados podem melhorar significativamente, porém, agrega-se a utilização de termos ou parâmetros adicionais – como medida de tolerância de erro, fator de suavização e funções de custo, por exemplo – os quais geralmente são definidos de forma subjetiva. Além disso, a estratégia de inserção ou remoção de nós possui a inconveniente alteração de dimensionalidade do problema a cada iteração, impossibilitando o uso de métodos de otimização baseados em gradiente (GÁLVEZ; IGLESIAS, 2013; ĆURKOVIĆ; ĆURKOVIĆ; VUČINA, 2018).

Aguilar et al. (2018) sugerem uma distinção em quatro categorias para os métodos de definição iterativa da quantidade de pontos de controle:

- a) Iterativos globais: realizam a geração iterativa e recursiva de curvas ou superfícies paramétricas baseadas no objetivo de reduzir uma métrica geral de erro estabelecida. Devido a geração de curva e cálculo de erro a cada iteração, podem ser custosos;
- b) Iterativos locais: realizam computação iterativa dos atributos da curva ou superfície em subconjuntos de dados restritos, sem gerar a forma paramétrica completa a cada iteração e, por isso, podem ser significativamente menos custosos do que os globais;

- c) Heurísticas e meta-heurísticas: utilizam regras definidas pelo usuário que estabelece o processo de escolha da quantidade de pontos de controle. O valor também pode ser otimizado pelos próprios métodos junto das demais variáveis de otimização;
- d) Guiados por curvatura: métodos que dependem e utilizam dados de curvatura para alocar a quantidade de pontos de controle, assumindo que regiões de maior curvatura devem receber mais nós do que regiões com pouca variação de curvatura.

Os métodos que computam e estimam a quantidade de pontos de controle podem se basear em limiares de erro aceitáveis, sendo que a quantidade de nós é fixada quando o erro entre um segmento de curva gerado e os pontos da nuvem associados é menor do que o limiar escolhido. A recomendação é utilizar um limiar de erro próximo ao valor do nível de ruído incidido sobre o processo de aquisição da nuvem de pontos, sendo que, um valor de limiar muito abaixo ao ruído da nuvem gerará pontos de controle em excesso e, um valor de limiar muito acima ao ruído da nuvem ocasionará a supressão de pontos de controle (DUNG; TJAHWIDODO, 2017). Deste modo, a escolha do limiar de erro aceitável está diretamente associada à relação de qualidade e simplicidade do modelo paramétrico resultante.

Além de considerações em relação à definição da quantidade de pontos de controle, outro aspecto relevante ao ajuste de curvas e superfícies paramétricas é a escolha da quantidade destes modelos paramétricos a serem utilizados, sendo distinguidas em abordagens que utilizam um único elemento ou múltiplos elementos. Por um lado, ao utilizar uma abordagem com uma única superfície paramétrica, pode ser difícil ou até mesmo inviável representar determinadas formas dependendo da sua topologias. Por outro lado, ao utilizar múltiplas curvas ou superfícies, existe maior flexibilidade na representação de geometrias e topologias mas, em contrapartida, é necessário realizar junções entre elas a fim de garantir algum nível de continuidade geométrica ou paramétrica. A estratégia de usar múltiplas superfícies também pode ser ideal para geometrias com partes simétricas, cantos agudos e outras regiões que podem ser simplificadas com particionamento (FLOATER; HORMANN, 2005; ĆURKOVIĆ; ĆURKOVIĆ; VUČINA, 2018). O uso de múltiplas superfícies, aliado à escolha adaptativa da quantidade de pontos de controle, pode ser utilizada para controlar e direcionar a concentração dos pontos de controle em regiões de maior curvatura como pontos de torção, bordas afiadas, cumes e vales, regiões que costumam requerer maior refinamento (JAVIDRAD, 2012; BERTOLINO et al., 2021).

2.4.6 Premissas dos Algoritmos de Ajuste

Os algoritmos de ajuste podem ser classificados de diversas maneiras. Uma forma usual de classificá-los é baseada na característica de determinismo. Neste sentido, abordagens de reconstrução podem ser agrupadas em algoritmos determinísticos, os quais sempre retornam um mesmo resultado para uma mesma entrada, e algoritmos estocásticos, os quais possuem alguma aleatoriedade no processo, o que conduz a resultados possivelmente diferentes a cada execução.

Berger et al. (2017) apresentam observações sobre sete suposições principais assumidas pelos métodos de ajuste de superfícies. Estas suposições podem ser utilizadas isoladamente

ou em conjunto. As três primeiras suposições e com vasta literatura utilizam a premissa da necessidade de suavização, que pressupõe que a nuvem de pontos representa um objeto com superfície suave, com existência de amostras ruidosas. Esta premissa de suavização é classificada em local, global e por partes.

A abordagem de suavização local procura gerar superfícies suaves próximas aos pontos detectados, suavizando as partes dos objetos, mas não necessariamente o todo. A utilização de método de mínimos quadrados móveis para tentar aproximar os pontos da nuvem através de polinômios, equações implícitas e funções de base radial são exemplos deste tipo de abordagem (ALEXA et al., 2003). Os métodos de suavização global, por sua vez, pressupõem também que a nuvem de pontos do objeto de interesse está completa, buscando gerar suavização mais expressiva e em escala maiores, reconstruindo um único objeto com superfícies herméticas (KAZHDAN; BOLITHO; HOPPE, 2006). Funções de base radial, funções indicadoras e métodos de segmentação volumétrica são representantes desta classe de métodos (BERGER et al., 2017). Por fim, os métodos de suavização por partes pressupõem a existência de regiões com características agudas, buscando identificar tais características e evitar a suavização destas. Estes métodos geralmente são específicos para objetos que possuem tais características. Métodos baseadas em particionamento, campos de normais (AVRON et al., 2010) e dicionário de aprendizagem são exemplos de técnicas que utilizam esta abordagem (BERGER et al., 2017).

O segundo tipo de pressuposto principal indicado por Berger et al. (2017) é a premissa de visibilidade. Ela indica que os métodos pressupõem algumas características sobre o espaço externo aos objetos de interesse e como elas podem auxiliar no tratamento das imperfeições das nuvens de pontos. Esta abordagem pode ser adequada para a reconstrução de um objeto individual hermético que apresenta ruído. Também é indicada para configurações interativas de reconstrução, nas quais os dispositivos são mantidos próximos da superfície do objeto (ZACH; POCK; BISCHOF, 2007; BERGER et al., 2017).

O terceiro tipo de pressuposto principal é a premissa de suavização volumétrica, a qual impõe a suavização em relação às variações de volume do objeto de interesse, admitindo que não ocorreu uma aquisição bem distribuída da nuvem de pontos. Esta premissa pode contribuir no ajuste de nuvens de pontos com ausência de dados significativa (TAGLIASACCHI; ZHANG; COHEN-OR, 2009; BERGER et al., 2017).

O quarto tipo de pressuposto principal é a premissa de que a geometria do objeto de interesse, amostrado pela nuvem de pontos, pode ser gerado por um conjunto de formas geométricas simples, como paralelepípedos, esferas, cilindros e planos. Tal proposição é plausível para diversas situações, visto que os objetos manufaturados são construídos por peças que possuem semelhança com alguma forma geométrica simples (LI et al., 2011; LANKVELD; KREVELD; VELTKAMP, 2013).

O quinto tipo de pressuposto principal é a premissa de que a nuvem de pontos representa um objeto ou um ambiente com algum nível de regularidade na sua composição. Por exemplo, as fachadas de prédio com a mesma estrutura, ou salas de aula com várias mesas e cadeiras

do mesmo modelo. Técnicas que utilizam esta abordagem podem admitir ausência de dados, replicando as regularidades detectadas para estes espaços, sendo úteis para tratar escaneamentos desalinhados (LI et al., 2011).

O sexto tipo de pressuposto principal é a premissa de que um ou múltiplos objetos de interesse são amostrados na nuvem de pontos e podem ser reconhecidos através de padrões. Presume-se que eles pertencem a uma classe de objetos que estão armazenados em uma base de dados de modelos 3D. Dessa maneira, mesmo que a nuvem de pontos apresente ruídos ou ausência de dados, é possível procurar e utilizar um modelo da base de dados que melhor corresponde à nuvem de pontos (PAULY et al., 2005; BRADLEY; NOWROUZEZAHRAI; BEARDSLEY, 2013). Logo, os resultados desta abordagem podem ser tão bons quanto os dados armazenados, o que ressalta o seu poder de reconstrução, mas também a importância da aquisição e armazenamento de uma grande quantidade de dados (BERGER et al., 2017).

Por fim, o sétimo tipo de pressuposto principal é a presunção de que o usuário pode auxiliar no processo de reconstrução de superfícies. Métodos que utilizam esta premissa esperam que o usuário informe algumas pistas que auxiliem no processo de reconstrução. Esta interação pode se estender desde a indicação de parâmetros de entrada, como bordas de limite do objeto, ou a topologia do objeto, até uma participação ativa, indicando regularidades, repetições de objetos e suas partes, ou selecionar grupos de pontos e apontar seus problemas, por exemplo (YAN et al., 2014; YIN et al., 2014; BERGER et al., 2017).

2.5 DISCUSSÃO DO CAPÍTULO

Foram apresentadas neste capítulo algumas noções de aspectos geométricos relevantes ao problema de ajuste de superfícies, assim como modelos de representação de curvas e superfícies paramétricas que podem ser utilizados. Também foi apresentada a definição matemática do problema de ajuste de superfícies como um processo de minimização de erro, destacando os principais aspectos envolvidos e as principais suposições utilizadas pelos métodos de resolução encontrados na literatura.

De acordo com Dung e Tjahjowidodo (2017), em um primeiro momento, o principal objetivo da utilização de métodos de ajuste de superfícies paramétricas era encontrar melhores maneiras de representar, de forma suave, um conjunto de dados ou funções complexas. Já com as recentes pesquisas em aplicações de engenharia reversa e outras áreas, passou-se a estudar não somente como representar objetos de forma suave, mas também como representar aspectos não triviais, como descontinuidades, torções, pontas e quinas, por exemplo.

O ajuste de curvas ou superfícies paramétricas sob uma nuvem de pontos usualmente é tratado como um problema otimização, no qual a minimização de erro costuma ser atrelada à resolução de sistemas de equações lineares ou não lineares. Estes costumam ser resolvidos com abordagens determinísticas, heurísticas ou técnicas de inteligência artificial, sendo que uma solução ótima para o problema não é garantida (IGLESIAS; GÁLVEZ, 2014). Além

disso, diversas outras particularidades devem ser consideradas, como a definição das variáveis do problema, a eventual parametrização, método de resolução e a decisão de outros valores eventualmente fixados.

Dentre os parâmetros dos modelos paramétricos, a definição da quantidade de pontos de controle, ou nós, é um dos principais elementos que influenciam na qualidade e simplicidade do modelo resultante (HASEGAWA; TORMENA; PARPINELLI, 2014). A otimização deste valor evita a necessidade de tentativa e erro na sua escolha, mas exige algum critério de parada, sendo comum a utilização de um termo de suavização (KRAGIĆ; ĆURKOVIĆ; VUČINA, 2018) ou um limiar de erro máximo tolerável de ajuste (MAO et al., 2018). Apesar das dificuldades envolvidas no processo, o ajuste de múltiplas superfícies paramétricas vem sendo utilizado como uma forma mais flexível para representar superfícies com diversas características topológicas (BERTOLINO et al., 2021).

3 REVISÃO DE LITERATURA

Os temas de ajuste de curvas e superfície paramétricas recebem ampla atenção na literatura científica. Existe uma diversidade de trabalhos que tratam deste assunto com diferentes objetivos, propostas e áreas de aplicação (POTTMANN et al., 2005; JAVIDRAD, 2012; RODRÍGUEZ, 2017). A possibilidade de usar múltiplas curvas combinadas, uma ou múltiplas superfícies paramétricas, diversos tipos de modelos paramétricos com características e propriedades distintas, além das diversas variáveis que podem ser otimizadas, abrem um leque com diversas possibilidades de atacar o problema.

Neste capítulo é apresentada uma revisão de literatura sobre algoritmos de ajuste de curvas e superfícies paramétricas. Foram exploradas principalmente as abordagens e algoritmos encontrados na literatura científica com proposta de otimização e viés de suavização, de acordo com a classificação de Berger et al. (2017). Também foi realizada uma busca em várias fontes a fim de identificar ferramentas e bibliotecas que envolvem operações de ajuste de superfície, sendo que as ferramentas possuem uma interface com o usuário para realizar este tipo de operação e as bibliotecas fornecem operações atreladas ao ajuste para uso na criação de software.

3.1 ALGORITMOS DE AJUSTE DE CURVAS E SUPERFÍCIES

Os métodos desenvolvidos para ajuste de curvas ou superfícies paramétricas geralmente estão associados a um processo de minimização de erro ou energia. Estas medidas são empregadas para guiar o processo de ajuste e definição das variáveis envolvidas no problema e geralmente estão atreladas a aspectos geométricos. Existe uma grande diversidade de propostas envolvendo ajuste de curvas e superfícies na literatura científica, sendo que muitas delas são apropriadas para alguns tipos específicos de problemas e seus domínios (AFSHAR; SORYANI; RAHMANI, 2011). Elas variam de propostas específicas para curvas, superfícies ou ambos; variam na escolha de modelo paramétrico utilizado, nos parâmetros ajustados, na distribuição ou característica das amostras da nuvem de pontos e na entrada de dados requerida ao usuário, para citar alguns aspectos. Além destes fatores, um divisor marcante entre as técnicas encontradas é o uso ou a omissão do processo de parametrização das amostras da nuvem.

Apesar das curvas paramétricas e de seu ajuste sobre uma nuvem de pontos ser semelhante em diversos aspectos às superfícies paramétricas e seu ajuste, existem considerações adicionais e obstáculos próprios vinculados a este segundo caso, especialmente com relação à junção entre superfícies. Existem também os métodos de *Surface Skinning* para o ajuste de superfície, os quais ajustam e combinam múltiplas curvas paramétricas a fim de gerar uma superfície resultante. Este tipo de método é frequentemente utilizado com nuvens de pontos semidispersos, nas quais as amostras estão distribuídas em seções transversais (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012). Além desta classe de métodos, as ideias e métodos utilizados no ajuste de curvas podem ser utilizados ou adaptados para o ajuste de superfícies. Isto reforça a importância de revisar também a literatura de ajuste de curvas paramétricas.

A seguir serão mencionados diversos algoritmos determinísticos e estocásticos (não determinísticos). Como o objetivo deste trabalho não é se aprofundar em detalhes sobre o funcionamento destes métodos, indica-se os livros de Nocedal e Wright (2006), e Kochenderfer e Wheeler (2019) como referências sobre os métodos de otimização determinísticos. Já para os métodos de otimização estocásticos, indica-se os livros de Gonzalez (2007) e Luke (2013).

Dentre os algoritmos estocásticos, as inspirações da área de inteligência computacional são amplamente exploradas na literatura, destacando-se o uso de meta-heurísticas. Estas são utilizadas como ferramentas genéricas para problemas de otimização, especialmente otimização não-linear. Possuem característica de maior exploração do espaço de soluções, apresentando boas chances de convergir em soluções ótimas globais, mas que geralmente exigem alto poder computacional, o que na prática torna tais algoritmos mais demorados do que os métodos determinísticos (GÁLVEZ; IGLESIAS, 2013; DUNG; TJAHIJOWIDODO, 2017; ĆURKOVIĆ; ĆURKOVIĆ; VUČINA, 2018).

3.1.1 Ajuste de Curvas com Parametrização

Uma ampla variedade de métodos para o ajuste de curvas paramétricas é encontrada na literatura. No caso das curvas, as principais variáveis de otimização são: o vetor da posição dos pontos de controle (\mathbf{P}), o vetor de valores paramétricos associados a cada ponto da nuvem (\mathbf{U}), o vetor de nós (\mathbf{T}), no caso dos modelos de curvas que utilizam este conceito, o vetor de valores de peso associados a cada ponto de controle (\mathbf{W}), no caso de curvas racionais, o grau da curva (d) e a quantidade de pontos de controle utilizada (ρ). Para otimizar algumas ou todas estas variáveis, tanto métodos determinísticos como métodos não determinísticos são empregados.

No caso das abordagens determinísticas, métodos de otimização não lineares e iterativos – newtonianos e quase-newtonianos – fazem parte do repertório encontrado. Para otimizar os vetores \mathbf{P} e \mathbf{U} em ajuste com B-Spline cúbicas, por exemplo, Speer, Kuppe e Hoschek (1998) utilizaram o método iterativo de Gauss-Newton, requerendo o cálculo da matriz jacobiana e sua inversa a cada iteração, o que pode se tornar uma tarefa custosa. De forma semelhante, Zheng et al. (2012) utilizaram uma variante do método de otimização iterativo BFGS (Broyden-Fletcher-Goldfarb-Shanno), denominada como L-BFGS (*Limited memory BFGS*) para otimizar os vetores \mathbf{P} e \mathbf{U} . O método L-BFGS, poupa a computação e armazenamento da aproximação para a matriz hessiana e sua inversa, reduzindo significativamente o tempo de cada iteração. A otimização por L-BFGS ocorre após a computação preliminar dos parâmetros de localização com base em uma curva inicial definida pelo próprio usuário.

O método de Gauss-Newton, combinado com a metodologia de região de confiança, foi utilizado por Carlson e Gulliksson (2008) para otimizar os vetores \mathbf{P} , \mathbf{U} , \mathbf{T} e \mathbf{W} em curvas NURBS cúbicas. De modo similar, Laurent-Gengoux e Mekhilef (1993) utilizaram o algoritmo de gradiente conjugado de Polak-Ribière para otimizar os mesmos vetores e também utilizando curvas NURBS cúbicas. Neste último, a função de custo utiliza uma ponderação entre o erro quadrático do ajuste e a soma das distâncias quadráticas entre pontos de controle adjacentes.

Nos casos de abordagens não determinísticas, as meta-heurísticas vêm sendo exploradas em problemas de ajuste de curvas. O algoritmo genético, por exemplo, foi utilizado por Sarfraz, Hussain e Irshad (2013) em conjunto com funções racionais quadráticas. Neste método, o contorno de uma imagem é segmentado, detecta-se então os pontos de quina e uma parametrização é calculada para os pontos de borda. Os parâmetros de forma são então otimizados pelo algoritmo genético e os pontos dos segmentos que apresentam uma medida de erro superior a um limiar preestabelecido são utilizados para dividir os segmentos. Os novos segmentos são então ajustados por novas funções racionais quadráticas. Irshad et al. (2016) utilizaram a mesma abordagem, mas empregando funções racionais cúbicas no ajuste das curvas.

De modo semelhante, Pandunata e Shamsuddin (2010), e Hasegawa, Rosso e Tsuzuki (2013) usaram o algoritmo de evolução diferencial para o ajuste de uma única curva de Bézier otimizando o vetor \mathbf{P} , empregando a parametrização de comprimento de corda em ambos os casos. A diferença na proposta de Hasegawa, Rosso e Tsuzuki (2013) está na paralelização do procedimento, aplicando uma estratégia de subdivisão em grupos de populações com o uso de uma etapa adicional de migração de indivíduos (representação de soluções) entre os grupos. Com a mesma finalidade, a meta-heurística de recozimento simulado (também descrita como têmpera simulada) foi utilizada por Ueda et al. (2020), com versões mono-objetivo e biobjetivos, otimizando o vetor \mathbf{P} e os pontos de junção no ajuste de múltiplas curvas de Bézier. As funções objetivo utilizadas neste trabalho levam em consideração o comprimento da curva e o erro quadrático aproximado entre as amostras da nuvem de pontos e as amostras da discretização do modelo de curva resultante. A versão mono-objetiva pondera entre estes dois critérios, ao passo que a versão biobjetiva utiliza ambos, isoladamente.

Em outra abordagem, Gálvez e Iglesias (2013) empregaram a meta-heurística do algoritmo da colônia de vaga-lume a fim de otimizar tanto o vetor \mathbf{P} quanto o vetor \mathbf{T} , ajustando uma curva B-Spline quadrática ou cúbica sobre a nuvem de pontos. Na proposta, os valores iniciais do vetor de nós são definidos aleatoriamente, então são distribuídos uniformemente e ordenados de forma crescente. Gálvez e Iglesias (2013) testaram quatro funções objetivo de minimização de modo isolado, para comparações, utilizando como critério básico o erro quadrático entre as amostras e a curva gerada. Com a mesma proposta, Gálvez e Iglesias (2016) empregaram a meta-heurística de otimização por enxame de partículas comparando os resultados da utilização de parametrização uniforme, comprimento de corda e centrípeta. Neste caso, porém, o vetor \mathbf{T} é otimizado pelos mecanismos da meta-heurística e o vetor \mathbf{P} é determinado através do método de mínimos quadrados com a resolução de sistemas lineares.

Também com proposta de otimização dupla, Hasegawa, Tormena e Parpinelli (2014) utilizaram curvas de Bézier e combinaram o algoritmo evolutivo biobjetivo com a resolução de sistemas lineares, onde o primeiro é utilizado para otimizar o vetor \mathbf{U} e o segundo para determinar os valores de \mathbf{P} . A função objetivo utilizada neste caso é a minimização de erro e do comprimento de curva. Durante o procedimento são armazenadas as melhores soluções encontradas (os pontos de pareto) considerando os dois aspectos de otimização e uma resposta é

selecionada através de um algoritmo decisor que pondera entre os dois objetivos.

Sarfraz et al. (2006) utilizaram o mesmo princípio de otimização dupla mas, neste caso, com o uso de múltiplos segmentos de curva, delimitados por amostras da nuvem de pontos detectadas como pontos significativos. Foram escolhidas curvas NURBS (com $d = \rho - 1$), as quais têm seu vetor \mathbf{P} determinado pelo método de mínimos quadrados e o vetor \mathbf{W} otimizado pela meta-heurística de evolução simulada. Foi utilizada uma parametrização centrípeta e cada segmento de curva possui $\rho = d + 1$. Como função de erro, é utilizada uma relação entre o erro quadrático e o comprimento do vetor de nós de cada segmento de curva.

Ebrahimi e Loghmani (2017) utilizaram o método *diagonal approximation* BFGS – uma variante do método de otimização iterativo BFGS (Broyden-Fletcher-Goldfarb-Shanno) – com o objetivo de ajustar uma curva B-Spline cúbica sobre uma nuvem de pontos com otimização da quantidade de pontos de controle ρ e dos vetores \mathbf{P} e \mathbf{U} . O algoritmo *diagonal approximation* BFGS faz uso da diagonal da aproximação da matriz hessiana, ao invés de computar e usar a matriz completa. A convergência pode ser mais lenta, mas o algoritmo não exige o cálculo de sistemas lineares a cada iteração e pode ser implementado com menos operações e alocações de memória. Para seus experimentos, Ebrahimi e Loghmani (2017) utilizaram um número inicialmente pequeno de pontos de controle, um vetor de nós não-periódico e fixo e uma parametrização inicial uniforme. A computação de \mathbf{P} é realizada pelo método de mínimos quadrados com base na parametrização inicial e, na sequência, é aplicado o algoritmo *diagonal* BFGS. Caso necessário, novos pontos de controle são adicionados, individualmente, na região de maior erro, reiniciando o algoritmo.

Com os mesmos princípios básicos, Ebrahimi e Loghmani (2018) utilizaram o método *Scaled BFGS* (SBFGS). Neste caso porém, uma técnica própria foi utilizada para determinar a quantidade e posição inicial dos pontos de controle, baseada na definição inicial de um parâmetro de comprimento. Este parâmetro é utilizado para computar distâncias associadas à curvatura, sendo que a quantidade e posição dos pontos de controle são extraídas pelos picos de máximo.

Outra abordagem que mescla a inserção de pontos de controle com posterior otimização dos pontos de controle é apresentada por Dung e Tjahjowidodo (2017). Com a parametrização e o nível de ruído da nuvem de pontos estimados, a etapa de inserção de nós tem como objetivo gerar um conjunto preliminar de nós, sendo iniciada pela aplicação de uma variante paralela do método da bissecção para subdividir os pontos da nuvem e associar cada grupo a um segmento de curva B-Spline cúbica. Esta etapa possui três partes: subdivisão – ocorre sempre que um segmento não atende ao critério de erro estabelecido e está atrelado a um número mínimo de amostras; junção – segmentos vizinhos são verificados a fim de encontrar e eliminar nós redundantes; e deslocamento – segmentos maiores são expandidos em direção a segmentos menores. Na sequência é realizada uma etapa de otimização desses nós em relação a sua posição e multiplicidade, utilizando o método de Gauss-Newton e considerando simultaneamente o erro quadrático e o ângulo de dobra da curva de ajuste. Por fim, a posição dos pontos de controle é computada através do método de mínimos quadrados.

Outro método que aplica a estratégia de inserção de pontos de controle é apresentado por Aguilar et al. (2018), o qual utiliza um processo de reajuste de \mathbf{P} , com característica de ser localmente iterativo. A proposta funciona escolhendo inicialmente um conjunto de pontos de controle preliminares, a partir da própria nuvem de pontos e com base em critérios de curvatura. Estes critérios também são utilizados, na sequência, para restringir o espaço de varredura de adição de novos pontos de controle, os quais são inseridos nos segmentos de maior diferença de curvatura, escolhendo a posição ótima local pela minimização da função de erro quadrático. Assim, a inserção de pontos de controle é regida pela minimização de uma função objetivo de erro local, de tal forma que não se faz necessário reavaliar toda a curva a cada nova inserção. O método proposto emprega ainda um esquema de reajuste na vizinhança de pontos de controle recém incluídos, reaplicando o passo de otimização, com o objetivo de evitar a formação de aglomerados e micro-oscilações locais na curva. As adições de novos pontos de controle são realizadas até que um limiar de curvatura definido seja satisfeito em todos os segmentos. Quando esta condição é alcançada, \mathbf{P} é calculado através do método de mínimos quadrados.

A quantidade de pontos de controle também já foi utilizada como parâmetro de otimização em uma abordagem híbrida por Costa, Montemurro e Pailhès (2018). No método proposto, são otimizados os parâmetros de forma d , ρ , \mathbf{P} , \mathbf{T} e \mathbf{W} de um único segmento de curva NURBS, com parametrização de tamanho de corda. Para realizar tal tipo de operação com dimensionalidade modificável, utilizou-se a variante do algoritmo genético denominada como BIANCA, através da qual são otimizados os valores de d , ρ e \mathbf{T} . Na sequência, com uma solução sub-ótima encontrada, utiliza-se o algoritmo determinístico baseado em gradientes para aperfeiçoar a solução otimizando \mathbf{T} e \mathbf{W} . Em ambas as etapas, \mathbf{P} é calculado pelo método dos mínimos quadrados. Apesar do método proposto otimizar diversos parâmetros, exige-se a definição inicial de limiares máximos e mínimos para restringir o espaço de busca do algoritmo genético.

3.1.2 Ajuste de Superfícies com Parametrização

No ajuste de superfícies paramétricas sobre nuvens de pontos, as variáveis de otimização continuam sendo análogas às utilizadas no ajuste de curvas: a matriz de posições dos pontos de controle (\mathbf{P}); o vetor de pares de valores paramétricos associados a cada ponto da nuvem (\mathbf{U}); os vetores de nós ($\mathbf{T} = \{\mathbf{T}_u, \mathbf{T}_v\}$), no caso dos modelos de superfícies que utilizam este conceito; a matriz de valores de peso associados a cada ponto de controle (\mathbf{W}), no caso de superfícies racionais; os graus da superfície ($d = \{d_u \text{ e } d_v\}$); e a quantidade de pontos de controle (ρ) utilizada por linha (ρ_u) e coluna (ρ_v).

3.1.2.1 Métodos Determinísticos

Assim como realizado no ajuste de curvas, Laurent-Gengoux e Mekhilef (1993), e Carlson e Gulliksson (2008) realizam também a otimização das variáveis \mathbf{P} , \mathbf{U} , \mathbf{T} e \mathbf{W} empregando superfícies NURBS bicúbicas. O método de Gauss-Newton combinado com a metodologia

de Região de Confiança foi utilizado por Carlson e Gulliksson (2008). Já Laurent-Gengoux e Mekhilef (1993) utilizaram o algoritmo de gradiente conjugado de Polak-Ribi  re, empregando uma func  o de custo que pondera entre o erro quadr  tico do ajuste e a soma das distâncias quadr  ticas entre pontos de controle adjacentes.

Lai, Hu e Pottmann (2006) apresentam um m  todo para o ajuste de superf  cies B-Splines bic  ubicas, mantendo um vetor de n  os fixo e ajustando os pontos de controle de modo a privilegiar – com maior quantidade de pontos de controle – as regi  es de maior aspecto de curvatura. Baseado no argumento de que os aspectos geom  tricos s  o caracterizados pela varia  o das normais da superf  cie, Lai, Hu e Pottmann (2006) realizam a parametriza  o atrav  s de um mapeamento da nuvem de pontos em um hiperplano \mathbb{R}^6 , com base na posic  o e na normal de cada ponto, e convertem posteriormente para \mathbb{R}^2 . Esta parametriza  o ´  stica sob a perspectiva de mapeamento com preservac  o de ´ rea, utilizando os componentes de curvatura principal e o suporte de uma func  o gaussiana. Tamb  m s  o utilizadas no mapeamento vari  veis definidas pelo usu  rio, com o intuito de regular a magnitude e escala do aspecto de sensibilidade utilizado na detec  o e manipula  o dos aspectos geom  tricos. J  o ajuste da superf  cie, a qual tem o n  mero de pontos de controle definido pelo usu  rio, ´  stica realizado pelo m  todo m  nimos quadrados atrav  s de resolu  o de sistemas lineares, seguido por um p  s-processamento com uma t  cnica iterativa SDM, a fim de melhorar o resultado obtido.

Outra abordagem com objetivo de utilizar um maior espa  o de par  metros para regi  es de interesse, concentrando mais pontos de controle sobre estas, ´  stica apresentada por Kragi  , Ćurkovi   e Vu  cina (2018), atrav  s de uma reparametriza  o baseada em campos escalares arbitr  rios. Nesta proposta, a estimativa inicial da superf  cie ´  stica realizada pela t  cnica de mapas harm  nicos (REMACLE et al., 2010) para a parametriza  o e pelo m  todo de m  nimos quadrados para o ajuste da superf  cie. Ap  s gerar a estimativa inicial, um processo de reparametriza  o iterativa ´  stica aplicado com uso de uma t  cnica h  brida composta por: mapas harm  nicos e *elastic springs* (GREINER; HORMANN, 1997). Este processo ´  stica conduzido pela pondera  o entre os gradientes das regi  es e os ditos campos de relaxamento – campos escalares quaisquer no dom  nio de par  metros – com objetivo de aumentar a quantidade de pontos de controle nas regi  es de maior erro. Superf  cies B-Splines foram utilizadas nos experimentos e a determina  o de diversos par  metros de execu  o foram realizadas manualmente.

Ainda com uma estrat  gia similar de prover um aprimoramento em regi  es de curvatura, Ćurkovi  , Ćurkovi   e Vu  cina (2018) apresentam um m  todo de ajuste de superf  cie utilizando NURBS, com uma parametriza  o organizada em forma de matriz, e com a proposta de modificar a matriz de parametriza  o, aumentando a densidade de elementos em regi  es com mudan  as significativas de curvatura. Para isto, utiliza-se uma parametriza  o baseada na t  cnica de proje  o harm  nica seguida por um seccionamento da proje  o. O ajuste inicial da superf  cie ´  stica realizado com o m  todo de Levenberg-Marquardt, otimizando \mathbf{P} , \mathbf{U} e \mathbf{W} . A otimiza  o de \mathbf{U} , no entanto, possui a caracter  stica de trabalhar apenas sobre dois vetores, um na dire  o do par  metro u e outro na dire  o do par  metro v , mantendo os valores param  tricos em uma

estrutura matricial alinhada. Com uma aproximação inicial da superfície computada, é realizada uma redistribuição na matriz de parametrização com base em uma função que relaciona as coordenadas curvilíneas e os autovalores das amostras da nuvem. Na sequência, um procedimento de suavização baseado em uma convolução com distribuição gaussiana é aplicado, a fim de obter uma redistribuição suave de pontos e aumentar a sua participação nas áreas de curvatura. Os testes realizados com superfícies majoritariamente suaves e detalhes agudos, mostraram que o método efetivamente amplia a quantidade de pontos de controle sobre as regiões de variação geométrica, mas possui limitações para representar geometrias complexas, as quais exigem uma abordagem com múltiplos retalhos.

Zhang et al. (2016) apresentam outra abordagem com foco em priorizar regiões de maior curvatura em um ajuste de superfícies B-Spline bicúbicas, com foco na quantidade e na disposição dos elementos de T . O algoritmo recebe uma nuvem de pontos parametrizada e executa recursivamente um procedimento de refinamento de nós enquanto um limiar de erro predefinido não é atingido. A cada iteração, uma quantidade fixa de elementos é adicionada individualmente em cada vetor de nós, baseada nas medidas de erro local, dividindo o subdomínio (intervalo entre nós) de maior erro de ajuste. Na sequência, as posições dos nós são definidas através de uma função objetivo baseada nas medidas de curvaturas principais e nos subdomínios criados através do posicionamento dos nós. A função objetivo é minimizada através do método da descida coordenada (ou descida das coordenadas) e o vetor de pontos de controle é otimizado através de mínimos quadrados.

Também buscando privilegiar áreas com maior curvatura, Wang e Zheng (2013) apresentam um método de ajuste de superfície T-Spline bicúbicas sobre malhas triangulares com otimização adaptativa da quantidade de pontos de controle. O método proposto realiza uma parametrização inicial por coordenadas de valor médio e utiliza como função objetivo a minimização do erro quadrático entre os vértices da malha triangular até a superfície gerada, com o acréscimo de um termo de suavização. O processo principal iterativo é constituído por três etapas: reparametrização, readaptação da estrutura de nós e ajuste por mínimos quadrados. Na primeira etapa é utilizada a reparametrização fiel (*faithful reparameterization*), que impede a intersecção entre as arestas de conectividade da malha triangular. Na segunda etapa, a estrutura de nós é ajustada acrescentando novos nós nas regiões de características que são detectadas pela curvatura, a qual é estimada pela variação da normal na vizinhança dos vértices. Após o ajuste por mínimos quadrados em cada iteração, um limiar de erro tolerável é utilizado como critério de interrupção, o qual é ponderado pela curvatura estimada de cada vizinhança e pela área da malha triangular que forma a nuvem de pontos.

De modo semelhante, Feng e Taguchi (2017) apresentam o fastTfit, um método de ajuste de superfície T-Spline bicúbicas sobre uma nuvem de pontos, mas neste caso com foco na velocidade do ajuste. Para tanto, o método emprega uma abordagem de ajuste baseada em dividir-conectar-ajustar, na qual a nuvem de pontos é parametrizada uma única vez pelo método uniforme ou por um método baseado em análise de componentes principais (*Principal*

Component Analysis – PCA) e é iterativamente dividida e ajustada – pelo método de mínimos quadrados – com superfícies Bézier ou B-Spline. Este processo é repetido até que um limiar de erro seja alcançado para cada retalho criado, o que gera uma estimativa inicial para a topologia de nós de uma T-Spline. Aspectos de continuidade entre os retalhos podem ser estipulados em C^0 e C^1 , sendo considerados na inferência da malha de nós da T-Spline. Os pontos de controle são então globalmente otimizados por fatoração QR (também chamada de decomposição QR, remete a decomposição de uma matriz A em um produto $A = QR$, sendo Q uma matriz ortogonal e R uma matriz triangular superior), ou por um método de gradientes conjugados.

Shang et al. (2018) também apresentam uma abordagem de ajuste de T-Splines bicúbicas mas, neste caso, voltada para a reconstrução de objetos fechados, relacionados à área médica, e que podem ser representados como uma malha tubular. O método recebe como entrada um conjunto de imagens de tomografia computadorizada (CT), as quais são segmentadas, transformadas em uma malha triangular e por fim parametrizadas por softwares terceiros. Para a parametrização, dois pólos devem ser escolhidos manualmente ou automaticamente, através dos quais é traçado um caminho que delimita as bordas do plano de parametrização. A otimização de \mathbf{P} ocorre por mínimos quadrados, sendo utilizado um fator de suavização – que é calculado com base numa distribuição proporcional ao tamanho da matriz de nós da T-Spline – e um vetor de fatores locais de ponderação da suavização. O método proposto é constituído por duas etapas principais. A primeira é a etapa de refinamento, na qual a superfície T-Spline é iterativamente ajustada e tem sua estrutura de nós subdividida enquanto um valor de erro tolerável não é alcançado para todos os subdomínios (intervalos entre nós) da superfície. A segunda é a etapa de suavização, na qual os fatores locais de ponderação da suavização são incrementados iterativamente até que todos os subdomínios fiquem acima do limiar de erro tolerável e, na sequência, os fatores são decrementados iterativamente até que todos os subdomínios fiquem com um erro ligeiramente abaixo do limiar tolerável definido.

Em uma abordagem distinta, Kineri et al. (2012) apresentam um método de aproximação de superfícies abertas com B-Splines bicúbicas, baseado em um algoritmo de aproximação geométrico iterativo. Inicialmente são criadas quatro curvas que aproximam os pontos das bordas do objeto representado pela nuvem de pontos. Com base nestas curvas, uma superfície base é gerada por interpolação bilinear dos pontos de controle, sendo tomada como uma primeira aproximação do resultado. Na sequência uma parametrização inicial é feita projetando as amostras da nuvem de pontos sobre a superfície B-Spline de base. Assim, a função objetivo é formulada como a minimização do erro quadrático com o acréscimo de um termo de suavização ponderado. A superfície gerada é iterativamente adaptada movimentando os pontos de controle na direção dos vetores de erro locais resultantes, sendo computada, na sequência, uma reparametrização com base nas menores distâncias entre as amostras da nuvem de pontos e a superfície. Este processo é realizado iterativamente enquanto o erro médio e o erro máximo sejam superiores aos limiares previamente estabelecidos. Se o erro não se tornar baixo o suficiente para alguma região, uma nova linha e coluna de pontos de controle é adicionada. O método proposto emprega também um

mecanismo para preservar simetria baseada na definição de um plano de corte.

Considerando a característica peculiar das nuvens de pontos semidispersos, alguns trabalhos exploram a abordagem de reconstrução baseada em múltiplas curvas. Shamsuddin, Ahmed e Samian (2006) apresentam uma proposta de ajuste de superfícies por meio de *surface skinning* com NURBS bicúbicas. O método propõem a utilização da curva com mais pontos de controle como referência, calculando uma média sobre o seu vetor de nós – através de um algoritmo de parametrização proposto – e removendo os pontos de controle enquanto um limiar de erro não seja excedido. Na sequência, as demais seções de curvas recebem um refinamento de nós de modo a compatibilizar a estrutura de nós entre elas.

O processo de compatibilização entre vetores de nós é o principal motivo do aumento da quantidade de pontos de controle necessárias nas representações por *surface skinning*. Em uma tentativa de contornar este problema, Li et al. (2015) apresentam um método de ajuste por *surface skinning*, com foco em objetos com característica de forma tubular, também empregando curvas NURBS bicúbicas mas, gerando ao final, uma superfície T-Spline bicônica. O uso de T-Splines evita o problema de incompatibilidade de nós entre as seções transversais. O método proposto consiste em cinco etapas: primeiro ocorre a inicialização com o cálculo dos vetores de nós para cada seção transversal; na sequência ocorre uma etapa de pré-processamento, verificando a necessidade de refinamento nas seções transversais, de modo a garantir uma conectividade mínima entre seções consecutivas; na terceira etapa ocorrem inserções locais de seções transversais intermediárias; na quarta etapa ocorre a interpolação das seções transversais, gerando uma superfície; finalmente, na última etapa, o utilizador pode realizar modificações manuais sobre as seções. Através desta estratégia é possível realizar modificações locais sem a necessidade de realizar consecutivas otimizações globais da superfície, o que poderia ser computacionalmente custoso. Nesta proposta não fica claro se a otimização de \mathbf{P} é realizada por mínimos quadrados ou outro método.

Também considerando nuvens de pontos semidispersos, Mao et al. (2018) propõem um método de ajuste de superfície NURBS bicúbicas que explora as características deste tipo de nuvem de pontos com uma abordagem semelhante ao *surface skinning*. O método proposto é composto por três passos principais: um ajuste de curva NURBS para cada linha de amostras da nuvem de pontos; um procedimento de reamostragem da nuvem de pontos baseada nas curvas ajustadas e considerando aspectos de curvatura, produzindo a mesma quantidade de amostras para cada linha; e um procedimento de ajuste de superfície NURBS sobre os pontos reamostrados. Os ajustes são feitos por meio de mínimos quadrados, otimizando \mathbf{P} , e a parametrização inicial de comprimento de corda é iterativamente refinada através de um algoritmo de uma projeção iterativa baseado no método de Newton, otimizando \mathbf{U} . O vetor de nós é calculado com base em \mathbf{U} , de modo a prover um sistema de equações lineares bem condicionado. A interrupção do processo iterativo de ajuste e reparametrização é condicionada a um limiar de erro predefinido.

3.1.2.2 Métodos Estocásticos

As meta-heurísticas também são amplamente utilizadas no ajuste de superfícies, mas costumam ser acompanhadas da utilização de algoritmos determinísticos, especialmente o método de mínimos quadrados para o cálculo das posições dos pontos de controle (\mathbf{P}). Gálvez, Iglesias e Puig-Pey (2012), por exemplo, utilizam o algoritmo genético juntamente do método de mínimos quadrados a fim de otimizar os vetores \mathbf{P} , \mathbf{U} e \mathbf{T} em superfícies B-Splines (grau escolhido manualmente). O algoritmo genético é utilizado em dois passos do método: na otimização da parametrização – a qual é simplificada em estrutura matricial e é iniciada pelos métodos padrões (comprimento de corda e centrípeta) ou de forma aleatória – e na otimização do vetor de nós, o qual é inicializado aleatoriamente. Com a parametrização e vetor de nós definidos, os pontos de controle tem suas posições calculadas por mínimos quadrados através de SVD ou decomposição LU. As três etapas de otimização são repetidas até que uma quantidade de iterações seja atingida, ou o erro não seja melhorado, ou ainda caso um limiar de erro seja satisfeito.

Barhak e Fischer (2001) apresentam outro método híbrido, que mistura algoritmos estocástico e determinístico, no ajuste de superfícies B-Splines. O método é baseado em uma etapa de reparametrização composta por duas fases: uma parametrização inicial 2D e uma parametrização adaptativa 3D. Inicialmente ocorre a projeção dos pontos de amostragem em um plano e é realizada a construção de uma matriz de parametrização 2D através de: equações diferenciais parciais (técnica determinística) e rede neural de mapas auto-organizáveis (técnica estocástica). Os dados amostrais são parametrizados sobre a matriz construída, gerando uma estimativa inicial para os pontos de controle. Na sequência, o método utiliza passos de melhoramento adaptativo, que são repetidos enquanto o erro de aproximação da superfície for maior do que uma tolerância aceitável pré-definida. Este melhoramento adaptativo ocorre pela projeção da nuvem de pontos sobre a superfície gerada, seguida pelo cálculo da nova parametrização e da correção no ajuste da superfície B-Spline. Foram testadas três abordagens, uma não iterativa (mínimos quadrados) e outras duas baseadas na aproximação de malha de controle através de melhoramentos iterativos nas posições dos pontos de controle. Neste último caso, foram utilizados o algoritmo determinístico de gradientes descendentes e o algoritmo de correção aleatória do erro de superfícies (*Random Surface Error Correction Algorithm*), o qual é estocástico e se assemelha ao algoritmo de rede neural utilizado na parametrização.

Iglesias, Gálvez e Loucera (2015) também apresentam um método híbrido no ajuste de superfícies Bézier. O método proposto utiliza a meta-heurística de reconhecimento simulado para otimizar \mathbf{U} e mínimos quadrados para otimizar \mathbf{P} . Para encontrar uma quantidade adequada de pontos de controle, o método é executado com diversas combinações de valores para a quantidade de linhas e colunas da matriz de controle da superfície utilizando-se da métrica de erro BIC para ponderar entre qualidade e simplicidade do ajuste. De modo similar, Iglesias et al. (2018) utilizam outra meta-heurística bioinspirada: o algoritmo de busca cuco para ajustar uma superfície Bézier sobre uma nuvem de pontos. Esta meta-heurística é utilizada na otimização de \mathbf{U} , o qual é

iniciado com valores aleatórios, juntamente de mínimos quadrados que é utilizado para otimizar \mathbf{P} . Utiliza-se como função objetivo o erro quadrático e emprega-se a técnica de *Lévy flights* para promover maior diversidade entre as soluções encontradas durante o procedimento, evitando estagnação prematura. Com estratégia similar, Iglesias, Gálvez e Collantes (2018) adotam a meta-heurística do algoritmo do morcego para ajustar superfícies de Bézier racionais sobre nuvens de pontos. Neste método proposto, tanto \mathbf{U} como \mathbf{W} são otimizados individualmente através da meta-heurística, ao passo que \mathbf{P} é otimizado por mínimos quadrados.

Rodríguez (2017) apresenta um método de ajuste de superfícies NURBS utilizando o algoritmo genético com o objetivo de interpolar uma nuvem de pontos com perspectiva de vista superior, obtida por um sistema de visão. Com esta configuração, a proposta apresenta dois aspectos incomuns: a quantidade de pontos de controle utilizada é igual à quantidade de pontos da nuvem, e a otimização da posição dos pontos de controle ocorre apenas sob uma dimensão – a profundidade. O método inicia com a geração de uma superfície NURBS preliminar, biquadrática e com nós não periódicos, que é computada com base nos pontos da nuvem. Também são calculados limiares máximos e mínimos que são definidos para cada ponto na dimensão de otimização, o que limita consideravelmente o espaço de busca da meta-heurística. Após isto, as posições dos pontos de controle e seus pesos associados são otimizados como um termo unificado pelo algoritmo genético. Como função objetivo para o algoritmo genético é utilizada a minimização do erro quadrático médio entre a superfície gerada e a nuvem de pontos, e como condição de término é definido um número máximo de iterações e um limiar de erro aceitável.

Os trabalhos de ajuste de superfície mencionados até então não realizam otimização da quantidade de pontos de controle (ρ), esperando-se que este valor seja fornecido como entrada ao algoritmo. Considerando a importância deste valor no balanceamento entre fidelidade e simplicidade do modelo resultante, Yoshimoto, Moriyama e Harada (1999) propuseram o ajuste de uma superfície B-Spline bicúbica com a utilização de um algoritmo genético para otimizar ρ juntamente do método de mínimos quadrados para otimizar \mathbf{P} . Nesta proposta, o intervalo do vetor de nós é subdividido em L intervalos que são codificados pelo algoritmo genético de forma binária (utilizar ou não um nó em um dado intervalo) utilizando a métrica AIC. Esta estratégia limita a disposição dos nós, caso o valor de L seja pequeno, em contrapartida, se o valor utilizado for grande, será necessário maior esforço computacional.

Em outra abordagem utilizando múltiplas superfícies, Bertolino et al. (2021) utilizam o algoritmo genético ERASMUS (*EvolutionaRy Algorithm for optimiSation of ModUlar Systems*), o algoritmo de conjuntos ativos e mínimos quadrados para otimizar todos os parâmetros de forma de NURBS. O ERASMUS permite a otimização estocástica com alteração na dimensionalidade do problema (i.e. alterações no valor de ρ). A dinâmica do algoritmo ocorre em três etapas. Na primeira é realizada a parametrização da nuvem de pontos utilizando-se do método de preservação de forma (FLOATER, 1997). Na segunda, o ERASMUS é utilizado na busca por uma solução otimizando apenas ρ , d e \mathbf{T} . A melhor solução encontrada é então utilizada como entrada para a terceira etapa do método, na qual se utiliza o algoritmo de conjuntos ativos (determinístico)

para otimizar primeiramente \mathbf{T} e depois \mathbf{W} . Nas duas últimas etapas, \mathbf{P} é otimizado através de mínimos quadrados. Como função objetivo, utiliza-se o erro quadrático acrescido de um termo de suavização. Apesar de otimizar todas as variáveis do problema, limites inferior e superior devem ser definidos para algumas variáveis, como d e ρ .

Além de permitir o ajuste de superfícies de gênero zero, o método de Bertolino et al. (2021) também prevê uma dinâmica para a reconstrução de superfícies complexas com gênero diferente de zero. Para tanto, o usuário deve realizar uma segmentação manual dos retalhos de superfícies, determinando os pontos de borda de cada retalho e sua conectividade com outros retalhos. Cada retalho é, então, otimizado individualmente mas, acrescenta-se uma estratégia de mestre-escravo para a realização das junções entre os retalhos resultantes com continuidade C^0 e C^1 . Ambas as continuidades são impostas mantendo intacto o segmento selecionado como mestre, e adaptando os pontos de controle dos segmentos vizinhos (escravos). Para viabilizar a junção, a meta-heurística mantém a mesma quantidade de pontos de controle entre os retalhos.

3.1.3 Ajuste de Curvas e Superfícies sem Parametrização

Além das abordagens de ajuste baseadas em algum tipo de parametrização, existem métodos que empregam novas maneiras de associar a nuvem de pontos com a curva ou superfície gerada, evitando a etapa de parametrização. Apesar da remoção de uma etapa não trivial, sem a associação paramétrica entre os pontos da nuvem e os pontos do modelo gerado, deve-se buscar por modos alternativos de se calcular o erro de ajuste, sendo que estes podem ser mais custosos do que a própria parametrização. Usualmente, a fórmula de erro dada pela Equação 25 é adaptada de modo a utilizar a distância entre os pontos mais próximos da nuvem e do modelo ajustado.

Trazendo a ideia de modelos de contorno ativos (*active contour models*) e conjuntos de níveis (*level set*) das áreas de Visão Computacional e Processamento de Imagens, Pottmann, Leopoldseder e Hofer (2002), Pottmann et al. (2005) e posteriormente Wang, Pottmann e Liu (2006) apresentam um método que emprega estas técnicas no problema de ajuste de curvas e superfícies. Nesta proposta, uma curva ou superfície cúbica ativa é inicialmente definida suficientemente próxima da nuvem de pontos e adaptada iterativamente com base em um método de minimização de erro quadrático. A cada iteração do algoritmo são computados e aplicados deslocamentos relativos aos pontos de controle, de modo que o modelo ativo converja em direção à nuvem de pontos. Estes deslocamentos são guiados pelo novo termo de erro de ajuste proposto, o chamado erro das distâncias quadráticas (SDM), o qual é definido por uma aproximação quadrática, baseada na curvatura, das distâncias quadradas entre a nuvem e o modelo.

Diferente dos outros termos de erro que mensuram as distâncias entre pontos, ou pontos e vetor tangente, o uso deste termo de erro proposto por Pottmann, Leopoldseder e Hofer (2002) busca relacionar os pontos de controle e seus deslocamentos de ajuste a um termo de suavização ponderado e às aproximações locais quadráticas (fórmula de Taylor) da função de distância quadrada entre os pontos da nuvem e o modelo de contorno ativo. Este termo de erro, aplicado como função objetivo, é quadrático em relação aos pontos de controle e pode ser computado

como um sistema de equações não lineares através de métodos *quasi-Newton*. Wang, Pottmann e Liu (2006) mostram que, no contexto de ajuste de curva como um problema não linear de mínimos quadrados, a abordagem por SDM é um método de resolução *quasi-Newton*, no qual se emprega uma aproximação baseada em curvatura para a matriz Hessiana.

O método de ajuste de curvas ou superfícies baseado em SDM costuma apresentar rápida convergência e potencial para ser utilizado – com pequenas adaptações – em outras aplicações, como registro entre nuvens de pontos, por exemplo (POTTMANN et al., 2005). Entretanto, assim como diversos outros métodos, a aplicação deste método não permite mudanças topológicas no modelo ativo durante sua execução. Pode ser necessária também a utilização de uma função de penalidade para evitar o encolhimento total da superfície ativa. Além disso, os resultados deste método podem ser altamente sensíveis à definição inicial do modelo ativo, sendo necessário alojar uma quantidade suficiente de pontos de controle próximas a cada região da nuvem de pontos, uma vez que o método os adapta de forma local. Essa sensibilidade ao modelo ativo inicial se aplica tanto no sentido de acurácia, onde o modelo ativo pode não convergir integralmente sobre a nuvem de pontos alvo, quanto no sentido de compressão, onde pontos de controle em demasia não são removidos ou realocados em outras regiões distantes.

Considerando os eventuais problemas advindos de uma inicialização deficiente do modelo ativo no método baseado em SDM, Yang, Wang e Sun (2004) sugerem uma extensão que faz uso de uma técnica para ajuste automático e iterativo de pontos de controle em curvas B-Spline cúbicas. A extensão projetada realiza uma combinação do ajuste de um modelo ativo sobre uma nuvem de pontos, usando SDM, com uma posterior avaliação de cada segmento do modelo ativo, e eventual inclusão ou remoção – nesta ordem – de novos pontos de controle. Este processo se repete iterativamente até que seja atingido um limiar de erro aceitável.

A estratégia de inserção de novos pontos de controle aplicada por Yang, Wang e Sun (2004), tem como objetivo a redução de erro local de cada segmento da B-Spline ajustada, e ocorrem sempre que um segmento da curva não atinge o limiar de erro aceitável. Uma inserção de ponto de controle sempre ocorre sobre um segmento que apresenta o maior erro, subdividindo este em duas partes. Na sequência, as posições dos pontos de controle que constituem os segmentos envolvidos são ajustadas com base em uma otimização de uma função objetivo baseada em SDM, através de um método *quasi-Newton*. Já a estratégia de remoção de pontos de controle, tem como objetivo a compressão da representação da curva ajustada, e é realizada através da busca e eliminação de nós que são localmente redundantes. A busca por estes nós acontece por testes de potencial de redundância, e a efetiva eliminação ocorre de forma individual, através da redistribuição dos nós remanescentes de modo a minimizar o aumento de erro e não ultrapassar um limiar de erro aceitável pré-estabelecido.

No método apresentado por Yang, Wang e Sun (2004), também existe uma forte dependência entre o resultado e o modelo ativo inicial. Neste sentido, é apresentada também uma forma de inicialização de modelo ativo, sendo utilizada uma abordagem com parametrização neste, e somente neste, processo. Além disso, é proposta a pré-computação de campos de distâncias para

o modelo ou nuvem de pontos alvo com o objetivo de minimizar o tempo gasto na computação recorrente da associação entre os pontos do modelo ativo e os pontos do modelo alvo. Por fim, apesar das estratégias de ajuste de pontos de controle, os processos de inserção e remoção de nós unitários podem ser significativamente custosos caso precisem ser executados em demasia. De modo semelhante, a computação da associação entre os pontos do modelo ativo e os pontos do modelo alvo pode ser custosa para modelos que exigem o uso de uma malha refinada.

Com uma estratégia diferente, Afshar, Soryani e Rahmani (2011) apresentam uma abordagem de otimização para o problema de ajuste de curvas sem parametrização, utilizando múltiplas curvas de Bézier, as quais são otimizadas por uma meta-heurística de algoritmo genético coevolutivo. Trata-se de um método composto por duas etapas principais: clusterização dos dados e posterior otimização da posição dos pontos de controle através da meta-heurística. A clusterização é utilizada para distinguir os componentes conexos e agrupar os seus pontos por proximidade. Em seguida, um segmento de curva de Bézier cúbico é ajustado sobre cada agrupamento identificado, sendo que os extremos dos segmentos são compartilhados entre os agrupamentos considerados vizinhos. A otimização é então realizada através da meta-heurística, utilizando a posição dos pontos de controle como variável de otimização, considerando o compartilhamento dos pontos de controle dos extremos e ainda garantindo a condição de continuidade C^1 . Como função objetivo para a meta-heurística, utiliza-se a minimização de erro (não quadrático) entre os pontos da nuvem e os pontos das curvas. A proposta de Afshar, Soryani e Rahmani (2011) também apresenta a característica de escolha inicial automatizada da quantidade de pontos de controle a ser utilizada, que é fixada. Isso ocorre através do algoritmo de agrupamento, o qual determina a quantidade de grupos e, por consequência, a quantidade de segmentos de curvas de Bézier cúbicas utilizadas.

Outra proposta com estratégia similar é apresentada por Javidrad (2012), a qual utiliza uma variação da meta-heurística de reconhecimento simulado para ajustar a posição dos pontos de controle e minimizar o erro de ajuste de uma curva B-Spline cúbica. Esta variação da meta-heurística aplica uma adaptação no processo de resfriamento do algoritmo, no qual a temperatura é alterada de modo dependente à evolução na função objetivo. Esta última é composta por um termo de minimização das distâncias entre os pontos da nuvem e a curva, e por outro termo referente à suavidade da curvatura, os quais são multiplicados por um ponderador individual. A quantidade inicial de pontos de controle é ajustada pelo usuário, assim como um limiar de movimento máximo por iteração, que é utilizado com o intuito de prevenir modificações demasiadamente bruscas de uma iteração para outra. O método proposto consegue lidar com formas complexas fechadas ou abertas, mas tem problemas com curvas onde linhas se interceptam.

3.2 FERRAMENTAS E BIBLIOTECAS PARA AJUSTE DE SUPERFÍCIE

Além da busca por técnicas e abordagens de ajuste e reconstrução de superfícies listadas na literatura científica, também foram consultadas ferramentas (software pronto para uso) e

bibliotecas (fornecem funcionalidades para simplificar a criação de software) voltadas para este propósito e suas características. Além de resultados referentes ao objetivo da pesquisa, houve um número significativo de referências a publicações científicas e relatórios técnicos relacionados à temática, além de software que processam nuvens de pontos, mas não realizam a reconstrução de malha propriamente dita (e.g. Cloud Compare, 3DVEM, SURE e Phi.3D). Referências a tais ferramentas podem ter sido apontadas devido à ambiguidade em determinados termos que possuem significados sensíveis ao contexto de nuvens de pontos e reconstrução. Por exemplo, o termo reconstrução pode significar a geração de malha para um modelo 3D ou simplesmente a geração de uma nuvem de pontos densa, e o termo triangulação pode ser entendido como a geração de malha triangular para um modelo 3D ou como um processo de identificação da posição relativa entre câmeras (perspectiva em uma foto) e pontos identificados para a nuvem.

Apesar de muitas das páginas web encontradas não fornecerem uma listagem exaustiva sobre os recursos ou informações técnicas das ferramentas, procurou-se coletar informações básicas sobre os recursos, o funcionamento e a acessibilidade do software, identificando se este é livre, modificável, multiplataforma, se gera modelos com superfícies suaves e permite computação através da unidade de processamento gráfico (GPU). Além disso, como foi identificado que muitas ferramentas encontradas também suportam operações de geração de nuvem de pontos por fotogrametria, esta característica também foi ressaltada. Uma síntese das peculiaridades de cada ferramenta e biblioteca é apresentada no Apêndice A. Para sumarizar os pontos chaves das ferramentas encontradas, apresenta-se a Tabela 1, excluindo a ferramenta *Imag'Ing* devido à falta de informações fornecidas na sua página web, não sendo possível concluir uma posição em relação à maioria das características analisadas.

Com relação aos resultados, pode-se observar que a maioria das ferramentas encontradas não são livres e não podem ser modificadas ou adaptadas por seus usuários, seja através de alterações no próprio código fonte ou através de scripts que são interpretados. Dentre as ferramentas pagas, apenas duas delas – *3D Reshaper* e *Halcon* – possuem a característica de serem alteráveis pelos usuários, e outras duas possuem planos especiais que permitem o seu uso de maneira gratuita – *3DF Zephyr* com uma licença educacional que permite o uso completo para fins não comerciais e a *GOM Inspect* com uma versão gratuita que pode ser utilizada mas possui restrições de funcionalidades em relação à versão paga. Além destes casos especiais, a maioria das ferramentas pagas disponibiliza uma versão de teste limitada.

A característica de portabilidade das ferramentas também está elencada na Tabela 1, na qual pode-se perceber que a maioria delas não é multiplataforma, estando disponível essencialmente para o sistema operacional Windows. As demais ferramentas podem ser utilizadas pelo menos nos sistemas operacionais Windows e Linux, com exceção dos software *Context Capture* e *Halcon*, os quais não se fazem claros sobre esta informação.

Apesar de todas as ferramentas apresentadas serem capazes de realizar operações de reconstrução de superfícies, somente parte delas permite o uso de superfícies suaves. As demais ou utilizam apenas malha triangular ou não especificam claramente esta informação. Em alguns

Tabela 1 – Tabela comparativa entre características e recursos de softwares de reconstrução.

Software	S.L.	C.M.	M.P.	S.P.	GPU	Fot.
3D Reshaper	□	■	□	■	-	□
3DF Zephyr	▲	□	□	-	■	■
Agisoft PhotoScan	□	□	■	□	■	■
Alice Vision Meshroon	■	■	■	-	▲	■
AutoDesk ReCap	□	□	□	-	□	■
CGAL	■	■	■	□	□	□
Context Capture	□	□	-	□	-	■
Geomagic Wrap	□	□	□	■	-	□
GOM Inspect	▲	□	□	-	-	□
Halcon	□	■	-	-	■	■
JRC 3D Reconstructor	□	□	□	-	-	■
LibIGL	■	■	■	□	□	□
Mesh Lab	■	■	■	□	-	■
Mic Mac	■	■	■	-	-	■
Open3D	■	■	■	□	□	□
PCL	■	■	■	■	▲	□
Pix4D	□	□	□	-	■	■
Point Fuse	□	□	□	-	-	□
PointCab	□	□	□	□	□	■
Polyworks Modeler	□	□	□	■	▲	□
Revworks	□	□	□	■	-	□
Trimble Real Works	□	□	□	-	-	■
VR Mesh Reverse	□	□	□	■	-	□

Siglas: S.L. = Software Livre / C.M. = Código Modificável / M.P. = Multiplataforma / S.P. = Superfícies Paramétricas / GPU = Realiza computação com GPU / Fot. = Fotogrametria.

Legenda: ■ = atende à característica / ▲ = atende parcialmente à característica / □ = não atende à característica / - = não foi possível determinar se atende ou não à característica.

Fonte: Elaborado pelo autor (2022).

casos, tal especificação foi encontrada apenas em fóruns oficiais, não sendo identificada na *homepage* do software. Além disso, as informações apresentadas nos conteúdos oficiais das ferramentas não expõem qual a técnica ou algoritmo utilizado para realizar a reconstrução, com exceção das ferramentas de código aberto, nas quais é possível analisar o código utilizado.

Também foram analisadas as ferramentas que permitem a aceleração de determinados procedimentos com o uso de placas gráficas (GPU), visto que muitas operações no processo de reconstrução podem exigir poder computacional significativo. Neste sentido, pode-se perceber que novamente muitas ferramentas não informam este recurso – ou pelo menos não o destacam – em suas *homepages* ou em seus fóruns oficiais. Dentre as seis ferramentas que suportam este recurso, duas delas o atendem parcialmente pois, exigem o uso de placas gráficas da

desenvolvedora NVIDIA¹, i.e. o recurso só pode ser utilizado com hardware específico.

Por fim, outro recurso analisado é a capacidade das ferramentas em trabalhar com fotogrametria. Apesar de não ser a única forma de se obter nuvens de pontos, este recurso é enaltecido pelas ferramentas que apresentam suporte a esta funcionalidade. Neste quesito, pode-se perceber que, em sua maioria, são as ferramentas pagas que possuem este recurso, ao passo que as bibliotecas não o incluem. Vale ressaltar ainda que alguns dos software encontrados que possuem o recurso de fotogrametria parecem ser especificamente voltados para imagens aéreas obtidas com o uso de drones, por exemplo.

Dentre as ferramentas e bibliotecas encontradas, três foram utilizadas no decorrer desta pesquisa para auxiliar no suporte a algumas operações. As bibliotecas CGAL e PCL foram empregadas para desenvolver uma ferramenta que realiza os principais tipos de parametrização da literatura. Já a ferramenta Mesh Lab foi utilizada para realizar a visualização dos resultados finais e intermediários do procedimento.

3.3 DISCUSSÃO DO CAPÍTULO

Foi apresentada neste capítulo uma revisão da literatura referente ao ajuste de curvas e superfícies paramétricas sobre nuvem de pontos, bem como um levantamento de bibliotecas e ferramentas que podem ser utilizadas para este objetivo. Com o intuito de sintetizar os trabalhos revisados, apresenta-se a Tabela 2 e a Tabela 3. São indicados nestas tabelas os seguintes dados: os autores; o modelo paramétrico utilizado na otimização com seu respectivo grau, identificando quando são empregadas múltiplas curvas ou superfícies, e as abordagens por *surface skinning*; o método de otimização utilizado, identificando se é uma abordagem determinística, estocástica ou uma mistura de ambas, e especificando os principais algoritmos utilizados; as variáveis otimizadas pelos métodos, de acordo com os símbolos utilizados neste trabalho; a especificação se os métodos utilizam ou não parametrização; as métricas de erro utilizadas para avaliar os resultados obtidos; e a origem das nuvens de pontos utilizadas nos experimentos.

Como todos os métodos que aparecem na síntese de superfícies utilizam parametrização, esta coluna foi omitida na Tabela 3. Quanto a origem da nuvem de pontos, elas foram classificadas em: próprias (quando não há menção de que a nuvem foi adquirida de algum repositório e a nuvem não foi utilizada em outros trabalhos revisados), *chain code*, equações (quando a nuvem de pontos foi gerada através da discretização de pontos com base em equações) e terceiras (quando a nuvem de pontos foi adquirida em algum repositório público, ou quando a nuvem de pontos utilizada foi encontrada em outros trabalhos revisados). A classificação do método proposto neste trabalho foi adicionada ao final da Tabela 3. O uso do termo mínimos quadrados na coluna do método de otimização, refere-se à utilização de algum método determinístico de resolução de sistemas lineares – como SVD ou decomposição LU – empregado na solução de uma formulação de mínimos quadrados, conforme apresentado na Equação 28.

¹ NVIDIA - Empresa desenvolvedora de placas gráficas. *Homepage*: <<https://www.nvidia.com>>

Tabela 2 – Síntese de trabalhos envolvendo ajuste de curvas paramétricas

(continua)

Pesquisa	Curva	Método de Otimização	Variáveis Otimizadas	Usa param.	Métricas de Erro	Origem da Nuvem
Afshar, Soryani e Rahmani (2011)	multi Bézier (d=3)	estocástico: algoritmo genético coevolutivo	\mathbf{P}	Não	ISE	próprias e <i>chain code</i>
Aguilar et al. (2018)	multi B-Splines (d=3)	determinístico: inserções baseadas em curvatura e mínimos quadrados	ρ e \mathbf{P}	Sim	RMSE	próprias
Costa, Montemurro e Pailhès (2018)	NURBS	híbrido: algoritmo genético BIANCA, método baseado em gradientes e mínimos quadrados	$d, \rho, \mathbf{P}, \mathbf{T}$ e \mathbf{W}	Sim	própria	equações e terceiras
Dung e Tjahjowidodo (2017)	multi B-Spline (d=3)	determinístico: método da bisseção, método de Gauss-Newton e mínimos quadrados	ρ, \mathbf{P} e \mathbf{T}	Sim	MSE e ME	equações e terceiras
Ebrahimi e Loghmani (2017)	B-Spline (d=3)	determinístico: <i>Diagonal Approximation</i> BFGS e mínimos quadrados	ρ, \mathbf{P} e \mathbf{U}	Sim	RMSE	próprias, equações e terceiras
Ebrahimi e Loghmani (2018)	B-Spline (d=3)	determinístico: <i>Scalled BFGS</i> e mínimos quadrados	ρ, \mathbf{P} e \mathbf{U}	Sim	RMSE	Equação e terceiras
Gálvez e Iglesias (2013)	B-Spline (d=2 e d=3)	estocástico: algoritmo do vaga-lume	\mathbf{P} e \mathbf{T}	Sim	RMSE, AIC e BIC	própria
Gálvez e Iglesias (2016)	B-Spline (d=3)	híbrido: algoritmo de enxame de partículas e mínimos quadrados	\mathbf{P} e \mathbf{T}	Sim	AIC e própria	próprias
Hasegawa, Rosso e Tsuzuki (2013)	Bézier	estocástico: evolução diferencial	\mathbf{P}	Sim	ISE	própria
Hasegawa, Tormena e Parpinelli (2014)	Bézier (d= ρ -1)	híbrido: algoritmo evolucionário biobjetivo e mínimos quadrados	\mathbf{P} e \mathbf{U}	Sim	ISE	equações

Fonte: Elaborado pelo autor (2022).

Tabela 2 – Síntese de trabalhos envolvendo ajuste de curvas paramétricas

(continuação)

Pesquisa	Curva	Método de Otimização	Variáveis Otimizadas	Usa param.	Métricas de Erro	Origem da Nuvem
Irshad et al. (2016)	funções racionais cúbicas	estocástico: algoritmo genético	P	Sim	próprias	próprias
Javidrad (2012)	B-Spline ($d=3$) e <i>Surface Skinning</i>	estocástico: recozimento simulado	P	Não	própria	próprias e equação
Pandunata e Shamsuddin (2010)	Bézier ($d=\rho-1$)	estocástico: evolução diferencial	P	Sim	própria	própria
Pottmann, Leopoldseder e Hofer (2002)	B-Spline ($d=3$)	determinístico: otimização quasi-newton	P	Não	-	próprias
Sarfraz et al. (2006)	NURBS ($d=\rho-1$)	híbrido: evolução simulada mínimos quadrados	P e W	Sim	ISE	terceiras
Sarfraz, Hussain e Irshad (2013)	funções racionais quadráticas	estocástico: algoritmo genético	P	Sim	-	terceiras
Speer, Kuppe e Hoschek (1998)	B-Spline ($d=3$)	determinístico: método de Gauss-Newton	P e U	Sim	ME	terceiras
Ueda et al. (2020)	multi Bézier ($d=3$)	estocástico: recozimento simulado mono-objetivo e biobjetivo	P	Sim	própria	próprias
Wang, Pottmann e Liu (2006)	multi B-Splines ($d=3$)	determinístico: otimização quasi-newton	P	Não	RMSE e ME	próprias
Yang, Wang e Sun (2004)	multi B-Splines ($d=3$)	determinístico: otimização quasi-newton	P	Não	MSE e ME	próprias
Zheng et al. (2012)	B-Spline	determinístico: L-BFGS	P e U	Sim	RMSE	terceiras

Fonte: Elaborado pelo autor (2022).

Tabela 3 – Síntese de trabalhos envolvendo ajuste de superfícies paramétricas

(continua)

Pesquisa	Superfície	Método de Otimização	Variáveis Otimizadas	Métricas de Erro	Origem da Nuvem
Barhak e Fischer (2001)	B-Spline (d=3)	híbrido: (equações diferenciais parciais / rede neural de mapas auto-organizáveis) + (mínimos quadrados / gradiente descendente / algoritmo de correção aleatória do erro de superfícies)	\mathbf{P} e \mathbf{U}	ME e mapa de calor	próprias e terceiras
Bertolino et al. (2021)	multi NURBS	híbrido: Algoritmo Genético ERASMUS e mínimos quadrados	$d, \rho, \mathbf{P}, \mathbf{U}, \mathbf{T}$ e \mathbf{W}	própria	terceiras
Carlson e Gulliksson (2008)	NURBS (d=3)	determinístico: Gauss Newton com Região de Confiança	$\mathbf{P}, \mathbf{U}, \mathbf{T}$ e \mathbf{W}	visual	própria
Ćurković, Ćurković e Vučina (2018)	NURBS	determinístico: projeção harmônica, Levenberg-Marquardt, convolução com distribuição Gaussiana	\mathbf{P}, \mathbf{U} e \mathbf{W}	mapa de calor	próprias ou terceiras
Feng e Taguchi (2017)	T-Spline (d=3)	determinístico: fatoração QR ou gradientes conjugados	$\rho, \mathbf{P}, \mathbf{U}$ e \mathbf{T}	RMSE e mapa de calor	próprias e terceiras
Gálvez, Iglesias e Puig-Pey (2012)	B-Spline (d=manual)	híbrido: Algoritmo Genético e mínimos quadrados	\mathbf{P}, \mathbf{U} e \mathbf{T}	MSE e ME	equações e própria
Laurent-Gengoux e Mekhlifi (1993)	NURBS (d=3)	determinístico: gradiente conjugado de Polak-Ribière	$\mathbf{P}, \mathbf{U}, \mathbf{T}$ e \mathbf{W}	ME	próprias ou terceiras
Iglesias, Gálvez e Loucera (2015)	Bézier	híbrido: Recozimento Simulado e mínimos quadrados	\mathbf{P} e \mathbf{U}	MSE e BIC	próprias ou terceiras
Iglesias et al. (2018)	Bézier	híbrido: algoritmo de Busca Cuco e mínimos quadrados	\mathbf{P} e \mathbf{U}	ISE e RMSE	próprias ou terceiras
Iglesias, Gálvez e Collantes (2018)	Bézier racional	híbrido: algoritmo do morcego e mínimos quadrados	\mathbf{P}, \mathbf{U} e \mathbf{W}	ISE e RMSE	terceiras

Fonte: Elaborado pelo autor (2022).

Tabela 3 – Síntese de trabalhos envolvendo ajuste de superfícies paramétricas

(continuação)

Pesquisa	Superfície	Método de Otimização	Variáveis Otimizadas	Métricas de Erro	Origem da Nuvem
Kineri et al. (2012)	B-Spline (d=3)	determinístico: aproximação geométrica iterativa	ρ , \mathbf{P} e \mathbf{U}	MSE, ME e mapa de calor	terceiras
Kragić, Ćurković e Vučina (2018)	B-Spline	determinístico: mínimos quadrados, mapas harmônicos e <i>elastic springs</i>	\mathbf{P} e \mathbf{U}	MSE, própria e mapa de calor	próprias ou terceiras
Lai, Hu e Pottmann (2006)	B-Spline (d=3)	determinístico: mínimos quadrados e técnica baseada em SDM	\mathbf{P}	mapa de calor	terceiras
Li et al. (2015)	<i>Surface Skinning:</i> NURBS e T-Spline (d=3)	determinístico: inserção e adaptação da malha de nós	ρ , \mathbf{P} e \mathbf{T}	visual	próprias ou terceiras
Mao et al. (2018)	NURBS (d=3)	determinístico: método de Newton, mínimos quadrados e algoritmo de reamostragem baseado em curvatura	\mathbf{P} e \mathbf{U}	MSE e ME	equações e própria
Rodríguez (2017)	NURBS	estocástico: algoritmo genético	\mathbf{P} e \mathbf{W}	RMSE	próprias
Shamsuddin, Ahmed e Samian (2006)	<i>Surface Skinning:</i> NURBS (d=3)	determinístico: inserção e remoção de nós	ρ , \mathbf{P}	visual	própria
Shang et al. (2018)	T-Spline (d=3)	determinístico: mínimos quadrados com fatores locais de ponderação de suavidade	ρ , \mathbf{P} e \mathbf{T}	RMSE e ME	terceiras
Wang e Zheng (2013)	T-Spline (d=3)	determinístico: mínimos quadrados, reparametrização fiel e readequação da estrutura de nós	ρ , \mathbf{P} , \mathbf{U} e \mathbf{T}	MSE e ME	terceiras
Yoshimoto, Moriyama e Harada (1999)	B-Spline (d=3)	híbrido: Algoritmo Genético e mínimos quadrados	ρ e \mathbf{P}	AIC	equação

Fonte: Elaborado pelo autor (2022).

Tabela 3 – Síntese de trabalhos envolvendo ajuste de superfícies paramétricas

(continuação)

Pesquisa	Superfície	Método de Otimização	Variáveis Otimizadas	Métricas de Erro	Origem da Nuvem
Zhang et al. (2016)	B-Spline (d=3)	determinístico: mínimos quadrados e método da descida coordenada	ρ , \mathbf{P} e \mathbf{T}	RMSE, ME e mapas de calor	terceiras
método proposto	multi B-Splines <i>Gregory patches</i> (d=3)	determinístico: divisão e conquista com mínimos quadrados	ρ , \mathbf{P} e \mathbf{U}	RMSE, ME, AIC e BIC	equações e terceiras

Fonte: Elaborado pelo autor (2022).

Todos os métodos de ajuste encontrados utilizam uma estratégia iterativa, através da qual o método realiza um procedimento de otimização repetidamente até que um ou múltiplos critérios de parada sejam alcançados. Os três principais critérios de parada utilizados nestes métodos são um número máximo de repetições, o uso de um limiar de erro de ajuste e o não melhoramento do resultado após uma sequência de iterações. Além disso, é comum que diversos atributos de curva e superfície ou do próprio método de otimização sejam fornecidos como entrada e estes valores costumam ser ajustados individualmente para cada caso de teste ou experimento. No caso das meta-heurísticas inspiradas na natureza, a escolha dos parâmetros influencia significativamente o desempenho do método (IGLESIAS et al., 2018). Em alguns casos, mesmo os atributos otimizados requerem uma definição de um valor inicial, sendo que este valor pode influenciar significativamente na qualidade do resultado obtido ou no tempo computacional necessário.

Considerando que a variável ρ desempenha um papel fundamental no balanceamento entre qualidade e simplicidade nos resultados obtidos, alguns métodos agregam estratégias de alteração dinâmica desta variável. A alteração dinâmica da quantidade de pontos de controle pode ser realizada através de inserção de novos pontos de controle (EBRAHIMI; LOGHMANI, 2018; DUNG; TJAHIJOWIDODO, 2017; PARK; LEE, 2007) ou através de remoção de pontos de controle (YOSHIMOTO; HARADA; YOSHIMOTO, 2003; GÁLVEZ; IGLESIAS, 2013; GÁLVEZ; IGLESIAS, 2016). Como a inserção ou remoção de um único ponto de controle por iteração é um procedimento custoso e que pode se tornar o gargalo no tempo de computação, é preferível adicionar ou remover múltiplos pontos de controle quando possível.

Em vários métodos propostos na literatura, percebe-se que ρ precisa ser determinado com antecedência, como parâmetro de entrada. Apesar de complexa, uma maneira de não delegar a escolha da quantidade de pontos de controle ao utilizador é definir alguma forma automática de estimá-la, mesmo que esta não seja feita de maneira ótima. Nesta estratégia, geralmente se utiliza uma abordagem de inserção ou remoção de pontos de controle durante a execução do algoritmo, atribuindo algum limiar de erro máximo ou compressão mínima aceitável, sendo o primeiro o mais comum. A escolha automática de ρ pode ser realizada com base na curvatura local estimada (AGUILAR et al., 2018), através de agrupamentos e particionamento das amostras (AFSHAR; SORYANI; RAHMANI, 2011), ou com alteração dinâmica e adaptativa (EBRAHIMI; LOGHMANI, 2017), por exemplo. Desta forma, o algoritmo executa iterativamente, melhorando o resultado no sentido de sua acurácia ou de sua compressão, sendo viável o ajuste de um balanceamento desejado entre estes dois objetivos. Além disso, pode-se notar que alguns métodos empregam uma parametrização fixa, mas técnicas de correção de parametrização já vem sendo adotadas com maior frequência. O uso destas técnicas permite a obtenção de melhores métricas de erro, no entanto, sua aplicação repetitiva em métodos iterativos pode também aumentar significativamente o tempo de execução do ajuste (KINERI et al., 2012).

Através da revisão dos métodos, pode-se observar que \mathbf{P} e \mathbf{U} são as principais variáveis otimizadas, ao passo que \mathbf{T} e \mathbf{W} são otimizadas com menos frequência, especialmente no caso

de ajuste de curvas. Durante a pesquisa foi possível detectar também que, em alguns casos, apesar dos métodos alegarem o uso de curvas ou superfícies racionais, as variáveis \mathbf{W} eram utilizadas com um valor fixo, sem otimização, o que descaracteriza e torna desnecessário os cálculos específicos do modelo racional. Apesar das T-Splines serem o modelo mais genérico para representação, o modelo que se destaca por ser mais utilizado nos trabalhos revisados é a B-Spline. Ademais, apesar da maioria dos trabalhos que optimiza ρ optar por um ajuste com refinamento incremental de uma única curva ou superfície, já existem abordagens que utilizam múltiplos segmentos de curva ou retalhos de superfície no ajuste. Este tipo de abordagem vem sendo empregado nos trabalhos mais recentes, especialmente no caso do ajuste de curvas, onde a junção entre os segmentos é mais simples de ser realizada. No caso das superfícies, no qual existe o problema de compatibilidade de torção entre a junções de três ou mais retalhos de superfície, acredita-se que a utilização de B-Splines com os princípios de retalhos de Gregory podem contribuir para a união destes retalhos com continuidade paramétrica em todas as suas bordas e quinas, criando superfícies contínuas.

Outra análise pode ser feita em relação as métricas de erro utilizadas. Apesar das métricas AIC e BIC ponderarem entre o erro de ajuste e a quantidade de variáveis empregadas na solução, elas foram menos utilizadas nos trabalhos do que as métricas tradicionais de erro (ISE, MSE, RMSE e ME). Inclusive, nos diversos métodos encontrados que utilizam um limiar de erro tolerável, este valor está associado com a métrica de RMSE e ME na maioria dos casos. Além do uso de métricas de erro, diversos trabalhos encontrados avaliam os resultados obtidos em superfícies ajustadas através de mapas de calor, diferenciando por cores as regiões da superfície que foram melhor ou pior ajustadas.

Outro aspecto que merece destaque é o fato de que grande parte dos trabalhos encontrados utilizam nuvens de pontos próprias ou não identificam sua fonte, dificultando ou impossibilitando a reproduzibilidade dos resultados. Além disso, mesmo entre os trabalhos que empregam nuvens de pontos sintéticas ou oriundas de repositórios abertos, existem aqueles que aplicam algum efeito de ruído randômico sobre a nuvem de pontos. Se por um lado, ao fazerem isso simulam dados reais, por outro, inviabilizam a reproduzibilidade ou a comparação, uma vez que em nenhum caso a nuvem gerada é disponibilizada. Outra dificuldade de comparação é a constatação da ausência de informações quantitativas de erro e de tempo de execução dos métodos apresentados e testados, conforme mencionam Feng e Taguchi (2017).

4 MÉTODO PROPOSTO

As técnicas de divisão e conquista resolvem diversos problemas de forma eficiente e são utilizadas para diversas finalidades (CORMEN et al., 2009). Esta estratégia pode reduzir a complexidade de tempo da resolução de problemas e possui a característica de que uma significativa parcela das suas etapas de execução podem ser paralelizadas, com poucos procedimentos seriais. Apesar do seu potencial, esta estratégia raramente é utilizada no ajuste de curvas e superfícies, com base nas pesquisas revisadas. Isto se deve, possivelmente, à difícil tarefa de combinar diversos retalhos de superfície com continuidade geométrica ou paramétrica, evitando o problema de compatibilidade de torção, e simplificar sua representação.

Algoritmos de divisão e conquista possuem pelo menos três etapas bem definidas: a divisão ou decomposição recursiva do problema em subproblemas menores de mesma natureza; a conquista ou resolução destes subproblemas e; a combinação das respostas gerando uma solução apropriada. Estes métodos, no entanto, não são aplicáveis a todo tipo de problema, pois a condição de particionar um problema em subproblemas não se aplica para toda e qualquer situação. Além disso, quando a condição de monotonicidade – isto é, a relação crescente ou decrescente entre as variáveis de otimização e a função objetivo – não se mantém durante toda a otimização, não existe garantia da obtenção de uma solução ótima para o problema. Outro possível obstáculo é que as operações de divisão, conquista e combinação podem ser mais custosas do que solucionar o problema original de forma direta.

O problema de ajuste de curvas e superfícies se adapta conceitualmente à proposta de uma estratégia de divisão e conquista. Por exemplo, uma nuvem de pontos que amostra um objeto pode ser iterativamente dividida em subconjuntos, os quais são ajustados individualmente e por fim combinados em uma única solução. Neste trabalho, optou-se por combinar a estratégia de divisão e conquista com a utilização de um limiar de erro (ϵ_{tol}) que é utilizado como referência e critério de escolha no processo de divisão iterativa do problema, contribuindo também na escolha automatizada da quantidade de pontos de controle (ρ).

Com a busca por um método eficiente, a otimização de variáveis não lineares em relação ao erro de ajuste foi evitada, uma vez que a resolução de sistemas de equações não lineares pode consumir um tempo significativo, especialmente ao utilizar nuvens de pontos com uma grande quantidade de pontos de controle, e não encontra necessariamente uma solução ótima. De fato, até mesmo a otimização por resolução de sistemas lineares gerado pelo método de mínimos quadrados pode requerer um tempo de computação considerável no caso de ajuste de nuvens de pontos com muitas amostras e especialmente com muitos pontos de controle. Isto porque os métodos de resolução de sistemas lineares possuem complexidade de tempo $\mathcal{O}(n\rho^2)$, onde n representa a quantidade de amostras da nuvem e ρ representa a quantidade de pontos de controle (GOLUB; LOAN, 1996; EBRAHIMI; LOGHMANI, 2018). Tendo isto em vista, o uso de uma estratégia de divisão e conquista busca realizar uma troca entre uma quantidade maior de operações pela redução do valor de ambas as variáveis envolvidas; em outras palavras, são

realizadas mais otimizações de menor custo.

4.1 MÉTODO DE DIVISÃO E CONQUISTA NO AJUSTE DE MÚLTIPLAS CURVAS

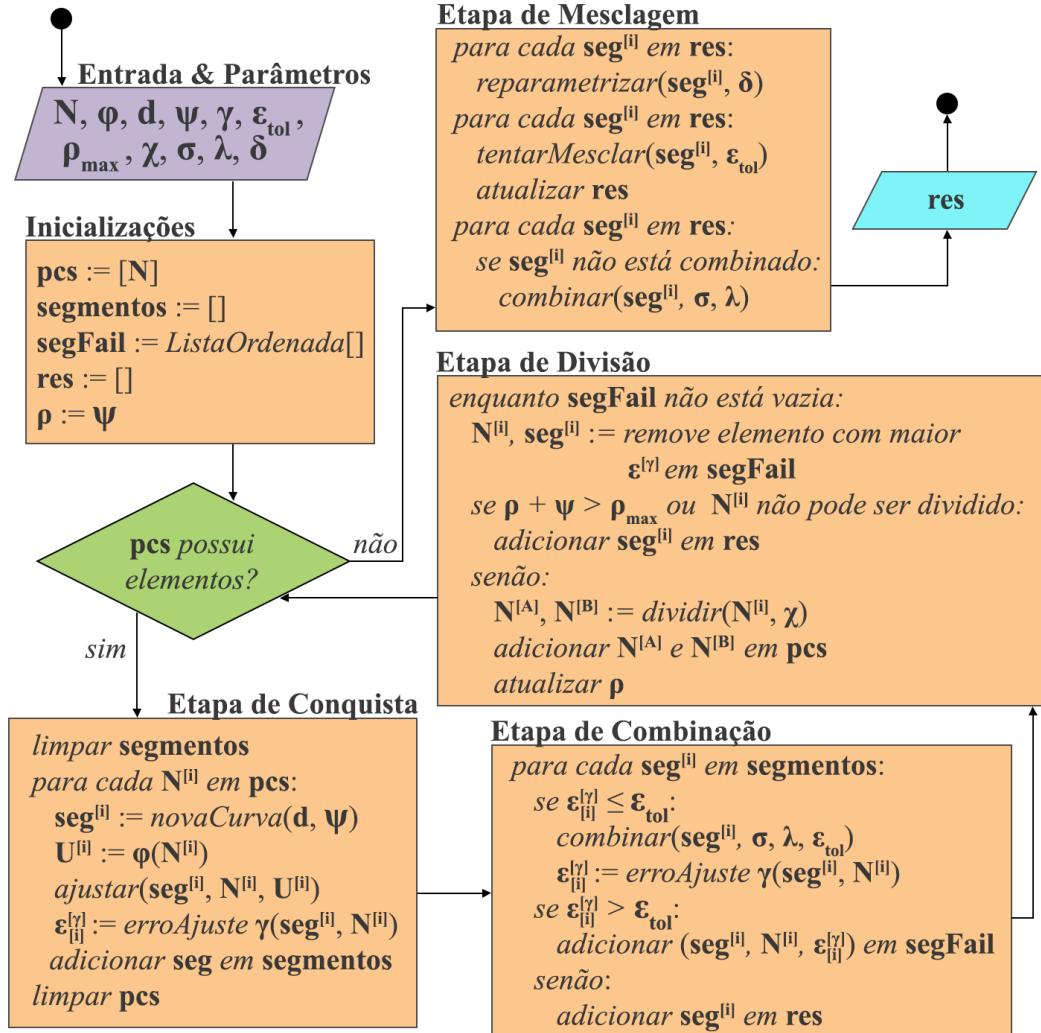
Considerando que o problema de ajuste de curvas é mais simples do que o ajuste de superfícies, optou-se por verificar inicialmente a eficácia e eficiência de um método de ajuste de múltiplas curvas paramétricas utilizando uma estratégia de divisão e conquista. Assim, o método de ajuste de múltiplas curvas foi um precursor ao ajuste de superfícies e a proposta apresentada a seguir foi publicada por Buchinger e Rosso (2022). Não é o foco deste trabalho se aprofundar no problema de ajuste de curvas, mas diversas ideias, ponderações e mecanismos empregados nesta proposta foram reaproveitados ou adaptados para o método de ajuste de superfícies proposto.

O método de ajuste de uma nuvem de pontos N por múltiplas curvas paramétricas B-Spline é composto por três etapas principais que são executadas recursivamente: conquista, combinação e divisão; além de uma etapa final de mesclagem. A Figura 13 ilustra um fluxograma com pseudocódigo de alto nível do método proposto para o ajuste de múltiplas curvas e seu funcionamento é descrito a seguir. Note que o método requer como entrada apenas uma nuvem de pontos organizada, sendo que os demais parâmetros possuem valores padrões preestabelecidos e testados para os principais cenários de ajuste. Os parâmetros utilizados pelo método serão descritos a seguir e seus valores padrões, assim como um intervalo de valores sugeridos, são apresentados na Tabela 4. Apesar de terem valores sugeridos, os parâmetros podem ser ajustados manualmente, na busca pelo aprimoramento dos resultados obtidos (*parameter tuning*).

Na etapa de conquista, todos os subconjuntos da nuvem de pontos (pcs) – inicialmente um – são (re)parametrizados com um método ϕ e uma nova curva paramétrica de grau d é ajustada para cada um deles através do método de mínimos quadrados, otimizando os pontos de controle P por sistemas lineares. O ajuste é realizado com uma quantidade fixa de ψ pontos de controle e, para cada nova curva ajustada, calcula-se o erro de ajuste $\epsilon_{tol}^{[\gamma]}$ baseado em uma métrica γ . O cálculo de erro, no entanto, utiliza um mecanismo de otimização que o interrompe caso o valor parcial calculado já ultrapasse o limiar ϵ_{tol} pré-estabelecido. Isso é realizado pois esta operação pode ser relativamente custosa e, caso o erro de ajuste seja maior do que o erro tolerável, não há razão pela qual calcular o seu valor exato. Por padrão, $\phi = \text{comprimento de corda}$; $\psi = \max(2d, \lceil \sqrt[3]{|N|} \rceil)$; e $\gamma = \text{ME}$.

Na etapa de combinação, todos os novos segmentos de curva que foram ajustados com um erro de ajuste menor do que um limiar pré-estabelecido $\epsilon_{tol}^{[\gamma]}$ são combinados com uma continuidade adaptativa. Dois segmentos de curva vizinhos são ajustados através da resolução de outro sistema linear, no qual σ pontos de controle mais próximos da ponta de junção de ambos os segmentos envolvidos são otimizados utilizando as restrições para obtenção de uma continuidade C^0 e C^1 . Mais detalhes de como esta solução é calculada são apresentados no trabalho de Buchinger e Rosso (2022), explicitando as matrizes e vetores que são utilizados no cálculo de um sistema linear produzido com base no método de mínimos quadrados. A solução

Figura 13 – Fluxograma de alto nível do método proposto para o ajuste de múltiplas curvas



Fonte: Adaptado de Buchinger e Rosso (2022)

com continuidade C^0 é escolhida somente caso o erro de ajuste com C^1 seja λ vezes maior do que o erro de ajuste com C^0 .

Por padrão, $\epsilon_{tol}^{[\gamma]} = \text{MMD}/2$ – onde MMD representa a Média das Menores Distâncias entre cada ponto da nuvem e seus vizinhos – quando a nuvem não apresenta ruído ($\epsilon_{tol}^{[\gamma]} = 3\text{MMD}/4$ para nuvens de pontos produzidas por *chain code*, $\epsilon_{tol}^{[\gamma]} = 4\text{MMD}/3$ no caso de nuvens com pouco ruído e $\epsilon_{tol}^{[\gamma]} = 3\text{MMD}$ no caso de nuvens com ruído significativo); $\sigma = \psi/2$; $\lambda = 4$ para formas suaves e $\lambda = 2$ nos demais casos. Na sequência, todos os segmentos de curva que não obtiveram um ajuste com erro menor do que o $\epsilon_{tol}^{[\gamma]}$ são colocados em uma lista de candidatos a divisão (*segFail*), ao passo que os segmentos de curva que obtiveram ajuste menor do que $\epsilon_{tol}^{[\gamma]}$ são adicionados em uma lista de segmentos de curva candidatos a solução (*res*).

Na etapa de divisão, cada segmento da lista de candidatos a divisão e seu respectivo subconjunto da nuvem de pontos são analisados. Inicia-se pelos segmentos de curva com maior erro de ajuste, verificando se existe um ponto válido para a divisão e garantindo um número

Tabela 4 – Lista de parâmetros para o método de ajuste por múltiplas curvas

Parâmetro	Intervalo Sugerido	Valor Padrão
ϕ	$\left\{ \begin{array}{l} \text{uniforme, centrípeto,} \\ \text{comprimento de corda} \end{array} \right\}$	comprimento de corda
d	$d < \psi$	Bézier: $\psi - 1$ / B-Splines: 3
ψ	$d < \psi \leq \left\lfloor \sqrt{ N } \right\rfloor$	$\psi = \max \left(2d, \left\lfloor \sqrt[3]{ N } \right\rfloor \right)$
γ	{RMSE, ME}	ME
$\varepsilon_{tol}^{[\gamma]}$	$\varepsilon_{tol}^{[\gamma]} \in \mathbb{R}$	$\begin{cases} 3 \cdot \text{MMD}/4 & \text{Chain Code} \\ \text{MMD}/2 & \text{Sem ruído} \\ 4 \cdot \text{MMD}/3 & \text{Pouco ruído} \\ 3 \cdot \text{MMD} & \text{Muito ruído} \end{cases}$
ρ_{max}	$\rho_{max} \geq \psi$	∞
χ	{BS, EB, GC, SC}	GC
σ	$3 \leq \sigma \leq \psi/2$	$\psi/2$
λ	$\{(C^0, -), (C^1, \mathbb{R}^+)\}$	$\begin{cases} (C^1, 4) & \text{Forma suave} \\ (C^1, 2) & \text{Forma não suave} \end{cases}$
δ	$1 < \delta \leq 100$	10

MMD = Média das Menores Distâncias entre cada amostra da nuvem e seus vizinhos

BS = Biseção / EB = Balanceamento de Erro / GC = Maior Curvatura /
SC = Menor Curvatura

Fonte: Adaptado de Buchinger e Rosso (2022)

mínimo de amostras por ajuste. Também é verificado se a quantidade de pontos de controle necessária ($\rho + \psi$) – considerando todos os segmentos – não extrapola um limite máximo de pontos de controle ρ_{max} . Se o subconjunto da nuvem $N^{[i]}$ não pode ser dividido, então ele é adicionado a lista de segmentos de curva candidatos a solução (res). Caso contrário, o ponto de divisão do subconjunto da nuvem de pontos é selecionado por um método de particionamento χ , o qual pode ser baseado em bisseção, balanceamento de erro, ponto de menor curvatura e ponto de maior curvatura. Buchinger e Rosso (2022) apresentam mais detalhes sobre o funcionamento dos métodos de partição utilizados. Por padrão, $\rho_{max} = \infty$ e χ = ponto de maior curvatura. Note que o valor padrão de ρ_{max} não limita a quantidade máxima de pontos de controle, sendo um mecanismo idealizado para situações específicas, nas quais se deseja obter um resultado que não ultrapasse uma quantidade de pontos de controle preestabelecida, garantindo um nível de simplicidade do modelo resultante.

As três etapas recursivas são executadas até que nenhum subconjunto da nuvem de pontos seja dividido. Neste caso, uma última etapa de mesclagem é realizada. Inicialmente um procedimento de reparametrização é realizado, no qual todos os segmentos de curva são discretizados com $\delta \cdot \rho$ amostras, as quais são igualmente espaçadas no domínio paramétrico – onde δ é um

fator pré-definido, com valor padrão $\delta = 10$. A nova parametrização é então escolhida com base nas menores distâncias encontradas entre as amostras da nuvem de pontos e os pontos discretos dos segmentos de curva. Na sequência, com o intuito de reduzir a quantidade de pontos de controle utilizada, aplica-se um procedimento de mesclagem. Neste, cada segmento de curva na lista de candidatos a solução (*res*) é analisado juntamente com um de seus vizinhos e tenta-se realizar uma fusão entre os dois segmentos. A nova curva gerada pela fusão é ajustada sobre a união destes dois subconjuntos da nuvem de pontos utilizando $\psi_{[c]} = [0,75(\psi_{[a]} + \psi_{[b]})]$ pontos de controle, onde $\psi_{[a]}$ e $\psi_{[b]}$ representam as quantidades de pontos de controle dos segmentos de curva vizinhos, antes da união. Note que esta expressão não garante uma minimização dos pontos de controle, mas se mostrou efetiva nos diversos experimentos realizados. Por fim, cada segmento de curva que não se encontra unido com seus vizinhos é combinado compulsoriamente pelo procedimento de combinação para gerar uma única curva final.

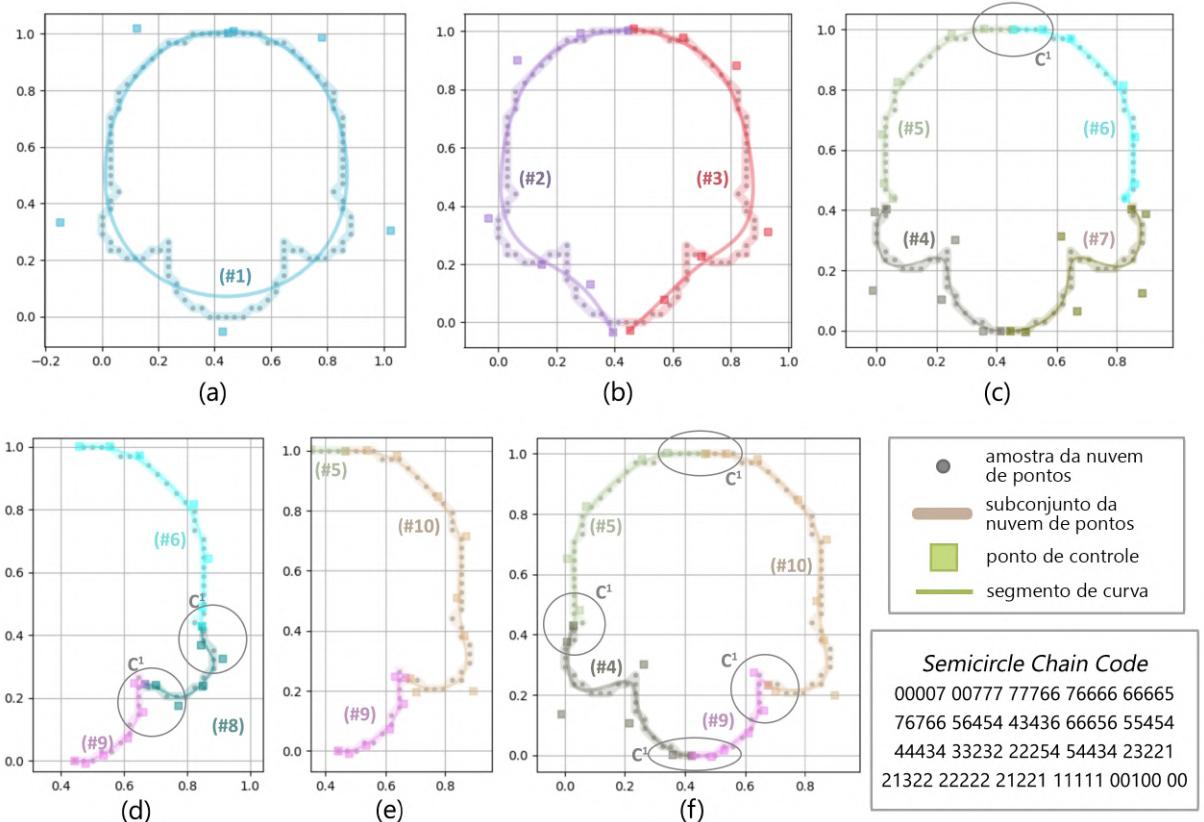
4.1.1 Exemplo do funcionamento do método

A seguir é apresentado um exemplo que ilustra o funcionamento do método proposto para o ajuste de múltiplas curvas, utilizando o *benchmark* do *semicircle* (TEH; CHIN, 1989). Foram utilizados os seguintes valores de entrada para melhor demonstrar alguns detalhes de funcionamento do método: N =(nuvem de pontos *semicircle* normalizada), $d = 3$, $\psi = 7$, ϕ =(comprimento de corda), γ =ME, $\epsilon_{tol}^{[ME]} = 0,03$, $\rho_{max} = 35$, χ =(bissecção), $\sigma = 3$, $\lambda = (C^1, 2)$ e $\delta = 10$. Inicialmente o método cria e realiza o ajuste de um único segmento de curva $c_{\#1}$ com grau 3 (d) e 7 (ψ) pontos de controle sobre a nuvem de pontos N parametrizada com o método de comprimento de corda (ϕ). O ajuste otimiza apenas os pontos de controle do segmento de curva $c_{\#1}$ e o resultado é apresentado na Figura 14(a). Como o erro de ajuste obtido é maior do que 0,03 ($\epsilon_{tol}^{[ME]}$) e $\rho < \rho_{max}$, a nuvem de pontos é dividida em dois subconjuntos através do método da bissecção.

Na segunda iteração pelo laço de repetição principal do método, os dois novos subconjuntos são parametrizados com ϕ e ajustados isoladamente por dois segmentos de curva $c_{\#2}$ e $c_{\#3}$, os quais possuem ambos 7 (ψ) pontos de controle e grau 3 (d). O resultado deste ajuste é apresentado na Figura 14(b). Como a métrica ME de ambos os ajustes é superior a 0,03 ($\epsilon_{tol}^{[ME]}$), a etapa de combinação não realiza alteração alguma nas curvas e, como nenhum critério de impedimento de divisão é válido, tanto o subconjunto da nuvem de pontos de $c_{\#2}$ quanto de $c_{\#3}$ são subdivididos pelo método da bissecção (χ).

Os quatro novos subconjuntos da nuvem de pontos são parametrizados e ajustados independentemente por quatro novos segmentos de curva: $c_{\#4}$, $c_{\#5}$, $c_{\#6}$ e $c_{\#7}$. Como resultado, apenas os segmentos $c_{\#5}$ e $c_{\#6}$ obtiveram um erro de ajuste inferior a 0,03 ($\epsilon_{tol}^{[ME]}$). Assim, estes dois segmentos de curvas foram combinados primeiramente com continuidade C^0 e depois C^1 . Como o erro de ajuste dessa combinação com C^1 não foi λ vezes maior o que a combinação com C^0 – i.e. $\epsilon_{C^1}^{[ME]} \leq \lambda \cdot \epsilon_{C^0}^{[ME]}$ – a solução com continuidade C^1 foi escolhida e o resultado geral é apresentado na Figura 14(c).

Figura 14 – Exemplo de funcionamento do método proposto para o ajuste de múltiplas curvas sobre o *benchmark semicircle*



Fonte: Adaptado de Buchinger e Rosso (2022)

Na sequência, como tanto $c_{\#4}$ e $c_{\#7}$ precisam reduzir seu erro de ajuste, apenas um deles pode ser subdividido, uma vez que $\rho = 24$, e duas novas subdivisões fariam ρ extrapolar ρ_{max} . Deste modo, como $c_{\#7}$ possui o maior erro de ajuste, o seu subconjunto da nuvem de pontos é escolhido para ser subdividido. Os novos subconjuntos gerados são então parametrizados e ajustados por dois novos segmentos de curva: $c_{\#8}$ e $c_{\#9}$. Como o erro de ajuste foi inferior a $\epsilon_{tol}^{[ME]}$, a combinação entre $c_{\#8}$ e $c_{\#6}$ foi testada, assim como a combinação entre $c_{\#8}$ e $c_{\#9}$. Em ambos os casos uma combinação com continuidade C^1 foi selecionada e o resultado parcial é apresentado na Figura 14(d).

Por fim, como nenhum segmento de curva pode ser refinado, o laço de repetição principal termina e a etapa de mesclagem é executada, computando primeiramente uma reparametrização para cada ajuste individual. Esta operação não altera a geometria do ajuste, mas refina a aproximação do erro de ajuste. Após esta operação, cada segmento de curva é testado com uma operação de mesclagem com seus vizinhos diretos. Inicialmente $c_{\#5}$ é testado com seu vizinho $c_{\#4}$ e então $c_{\#6}$, mas o erro de ajuste após estas operações de mesclagem é maior do que $\epsilon_{tol}^{[ME]}$, então a mesclagem é descartada. Na sequência, $c_{\#6}$ é testado e mesclado com sucesso com seu vizinho direto $c_{\#8}$, gerando um novo segmento de curva $c_{\#10}$ que é composto por 10 pontos de controle ($[0, 75(\psi_{[#6]} + \psi_{[#8]})]$) e que foi combinado com seus vizinhos $c_{\#5}$ e $c_{\#9}$ com

continuidade C^1 . O resultado desta operação de mesclagem é apresentado na Figura 14(e).

Os segmentos de curva $c_{\#4}$, $c_{\#9}$ e $c_{\#10}$ foram testados para a operação de mesclagem com seus vizinhos, mas todas falharam por não obter um erro de ajuste inferior ao limiar imposto e foram descartados. Após todas as verificações por mesclagem, os segmentos de curva $c_{\#4}$ e $c_{\#5}$, e também os segmentos de curva $c_{\#4}$ e $c_{\#10}$ foram combinados com continuidade C^1 , finalizando o ajuste. O resultado final é apresentado na Figura 14(f).

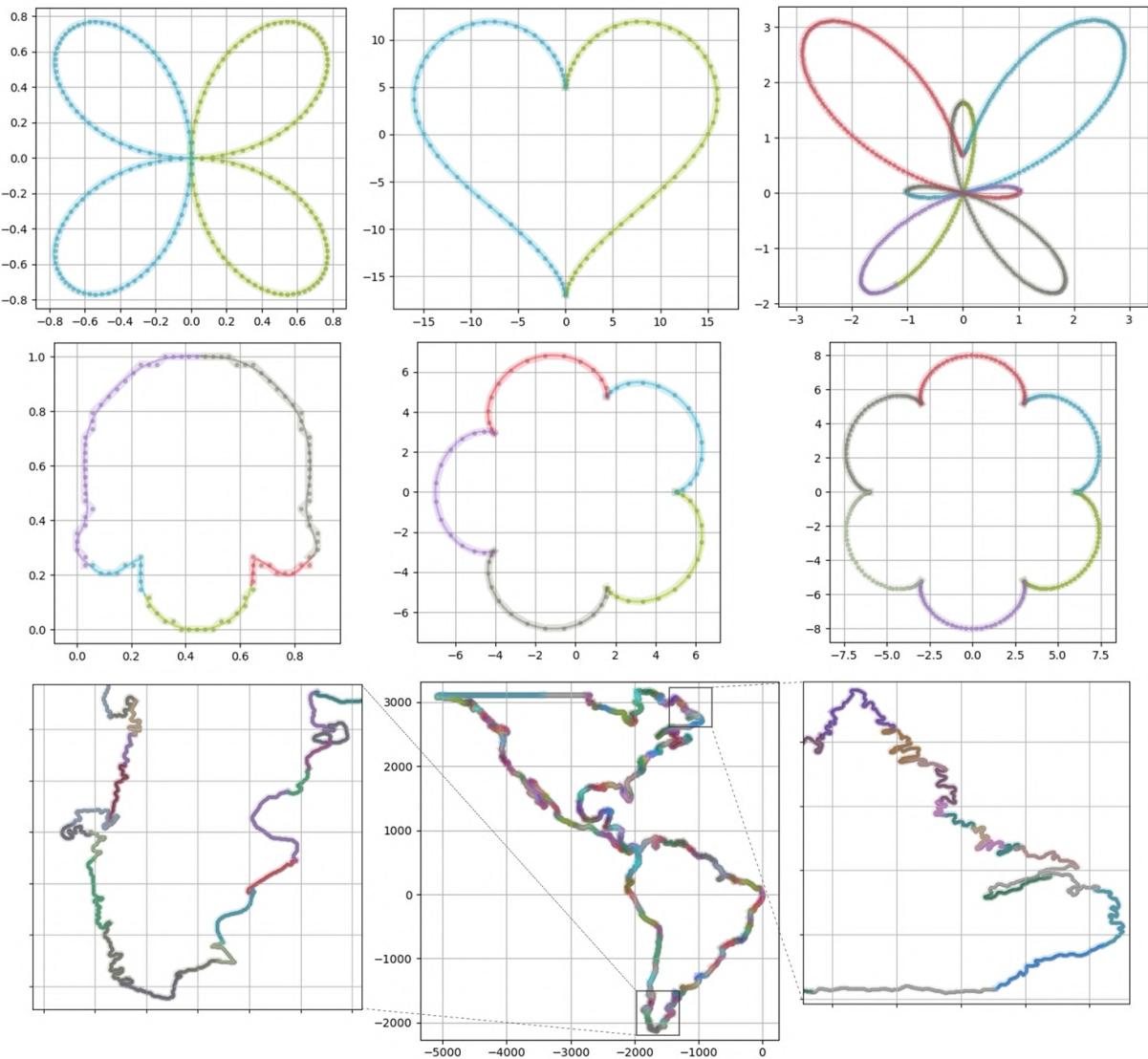
4.1.2 Experimentos e Considerações sobre Eficiência

Em relação ao tempo de execução, as operações mais custosas do método proposto são os múltiplos ajustes de curvas necessários e, especialmente, a operação de mesclagem das curvas. O método de ajuste de múltiplas curvas foi testado com diversos *benchmarks* clássicos de curvas definidas por equações e por *chain-code* (TEH; CHIN, 1989), obtendo bons resultados quantitativos e qualitativos, utilizando os parâmetros indicados por padrão (BUCHINGER; ROSSO, 2022). Em especial, os quatro métodos de partição foram testados e comparados e o valor de $\delta = 100$ foi ajustado para os casos de comparação de resultado da métrica de erro, de modo a permitir uma estimativa mais apurada do erro de ajuste real. O método de particionamento da nuvem por maior curvatura resultou em melhores aproximações para aspectos de continuidade em alguns *benchmarks*, mas não ficou evidente que seus resultados são, em média, melhores do que os demais métodos testados.

De modo geral, os resultados obtidos pelo método de ajuste por múltiplas curvas foram promissores, com a obtenção de resultados tão bons quanto e até mesmo melhores do que os identificados na literatura (BUCHINGER; ROSSO, 2022). A Figura 15 apresenta alguns destes resultados obtidos com a utilização do método proposto, na qual cada traçado de uma cor representa um segmento de curva distinto. Os *benchmarks* apresentados, da esquerda para a direita e de cima para baixo, são os seguintes: *quadrifolium*, *heart shape*, *butterfly curve*, *semicircle*, duas variações do *epicycloid*, e uma imagem digital representando o contorno de grande parte do continente americano, detalhando duas áreas específicas.

Note que ao utilizar a mesma quantidade de pontos de controle com o método tradicional, o qual realiza o ajuste de uma única curva B-Spline (monocurva), e utilizando a mesma parametrização (comprimento de corda), este se torna mais eficaz, pois não há necessidade em alocar pontos de controle específicos para viabilizar junções. Na prática, no entanto, esta quantidade de pontos de controle não é conhecida, ao passo que o método proposto calcula de forma automatizada este valor. Além disso, ao testar o crescimento do tempo de execução da abordagem proposta (multicurva) e do método tradicional (monocurva) no experimento do contorno do continente americano, uma nuvem de pontos composta por 34.159 amostras, o ajuste de múltiplas curvas B-Splines se mostrou assintoticamente mais eficiente do que o método tradicional de ajuste por uma única curva B-Spline. Este resultado deixou em evidência o comportamento de crescimento quadrático do ajuste tradicional e o comportamento logarítmico do ajuste de múltiplas curvas.

Figura 15 – Resultados obtidos com o ajuste de múltiplas curvas

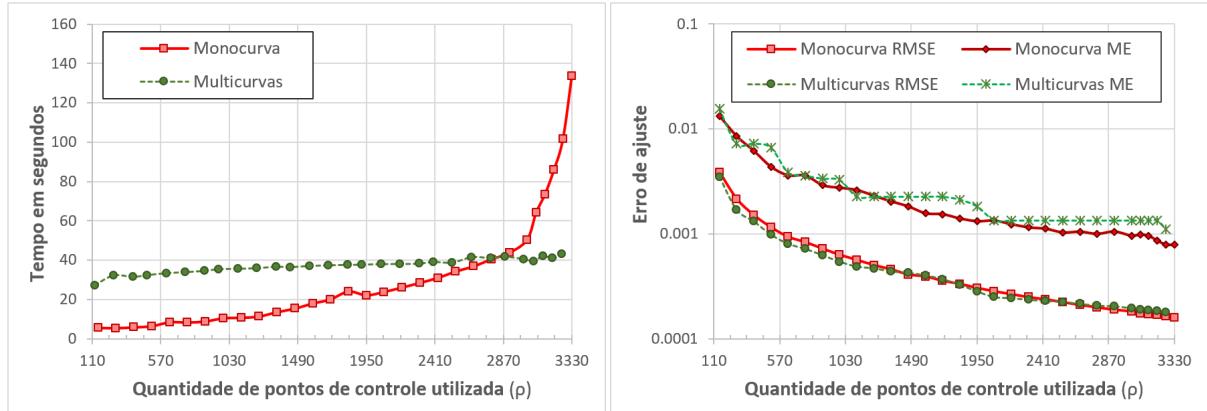


Fonte: Adaptado de Buchinger e Rosso (2022)

A Figura 16 apresenta a comparação de tempo e erro de ajuste para o método tradicional com uma única curva (monocurva) e o método de ajuste de múltiplas curvas desenvolvido (multicurvas), com base no experimento do contorno do continente americano. O eixo horizontal se refere a quantidade de pontos de controle (ρ) empregada ao final do ajuste, que no caso do método tradicional é escolhida manualmente e no caso do método desenvolvido, se refere ao valor resultante do ajuste, após todas as operações de divisão, conquista e mesclagem. Este experimento foi realizado em um computador com processador AMD FX(tm)-8350 de oito núcleos (4.00GHz) e 16GB de memória RAM, sendo que ambos os métodos de ajuste foram implementados na linguagem de programação Python¹.

¹ Python v3.8.3 (<<https://www.python.org/>>)

Figura 16 – Comparação de tempo e erro de ajuste entre o método tradicional de ajuste de curvas e o método proposto para o ajuste de múltiplas curvas para a nuvem de pontos do contorno do continente americano



Fonte: Elaborado pelo autor (2022).

4.2 MÉTODO DE DIVISÃO E CONQUISTA NO AJUSTE POR MÚLTIPLAS SUPERFÍCIES

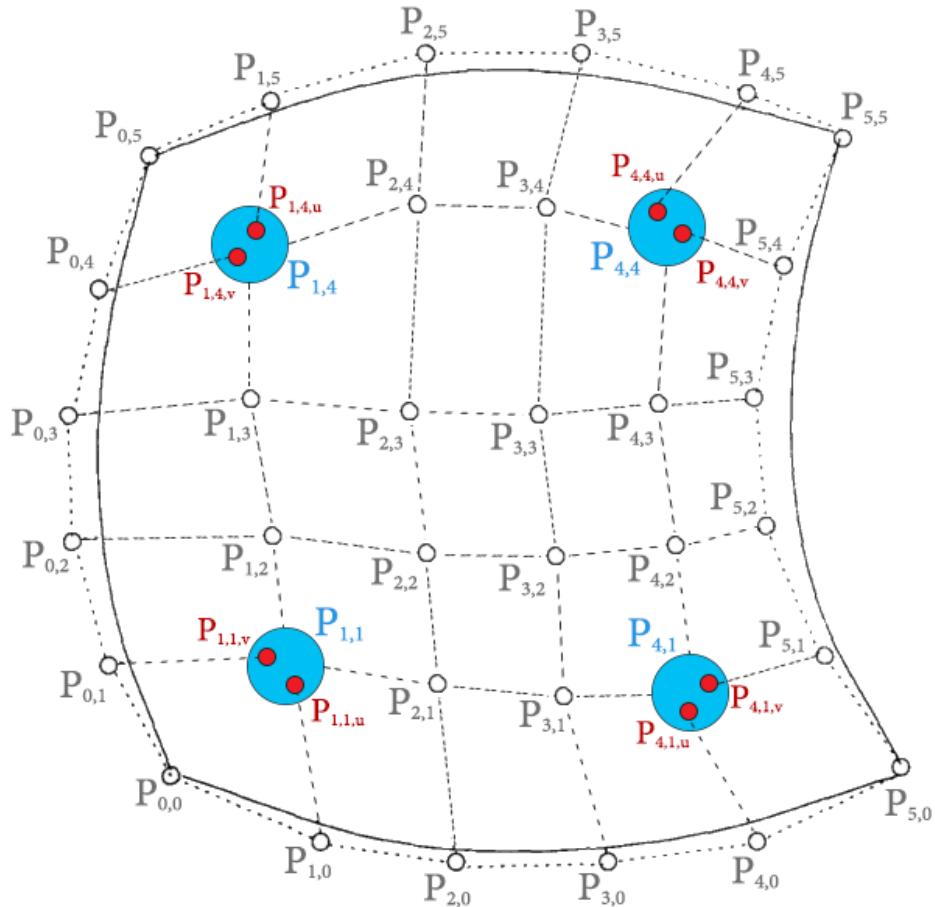
Com a verificação de que a estratégia de divisão e conquista é eficaz e eficiente na resolução do problema de ajuste de curvas por múltiplos segmentos de curvas, os mesmos princípios e procedimentos foram utilizados e adaptados para uso no problema de ajuste de superfícies. Diversas especificidades deste segundo problema tiveram de ser consideradas e exigiram adequações significativas no fluxo do método proposto. A visão geral do método e o detalhamento de cada uma das etapas principais serão apresentados nesta seção. Além disso, para assegurar um grau de continuidade entre os eventuais retalhos de superfície vizinhos, evitando o problema da incompatibilidade de torção, foram empregadas superfícies B-Spline com princípios de retalhos de Gregory. Deste modo, foi elaborado e implementado um algoritmo para ajuste de superfícies B-Spline com princípios de retalhos de Gregory com uma estratégia de divisão e conquista, e seleção automática da quantidade de pontos de controle – *a Divide-and-Conquer algorithm for surface fitting with multi-Bspline-gregory-patches with Adaptive control points selection: DCBA*.

4.2.1 Superfícies B-Spline com Princípios de Retalhos de Gregory

Não foram encontrados na literatura pesquisas que vinculassem as superfícies de B-Spline com os princípios de retalhos de Gregory, apenas trabalhos que utilizam superfícies de Bézier com estes princípios. Dentre as superfícies utilizadas, foram identificados superfícies bicúbicas, com matriz de controle 4x4 (HETTINGA; KOSINKA, 2008), utilizadas para garantir continuidade C^1 nas bordas; e superfícies biquânticas, com matriz de controle 6x6, utilizadas para garantir continuidade C^2 . Estes modelos, no entanto, não possibilitam a utilização de superfícies bicúbicas com matrizes de controle de tamanhos diferentes, devido a relação direta entre o grau da superfície e sua quantidade de pontos de controle.

O modelo de superfície B-Spline com princípios de retalhos de Gregory funciona com os mesmos princípios básicos aplicados sobre as superfícies de Bézier, conforme apresentado na Subseção 2.3.1. Considere uma superfície B-Spline bicúbica com matriz de controle de dimensão $m \times n$. Cada ponto de controle na diagonal das quinas, $P_D = \{P_{1,1}, P_{1,n-2}, P_{m-2,1}, P_{m-2,n-2}\}$, é representado por dois pontos de controle, um relacionado ao parâmetro u e outro relacionado ao parâmetro v , os quais são ponderados por uma função de mistura racional. A Figura 17 ilustra a estrutura deste modelo de superfície destacando os elementos de controle mencionados, na qual os pontos de controle direcionais, vinculados a um parâmetro específico, são representados pelos círculos vermelhos, e sua combinação pelas funções de mistura racional geram o ponto de controle resultante denotado pelo círculo azul. Os demais pontos de controle apresentam seu comportamento e estrutura padrão de uma superfície B-Spline.

Figura 17 – Exemplo de um retalho quadrilátero de Gregory como uma superfície B-Spline bicúbica



Fonte: Elaborado pelo autor (2022).

As funções de mistura racionais, que ponderam os pontos de controle direcionais, são

análogas às apresentadas na Equação 23:

$$\begin{aligned} P_{1,1} &= \frac{v P_{1,1,u} + u P_{1,1,v}}{u+v} & P_{4,1} &= \frac{(1-v) P_{4,1,u} + u P_{4,1,v}}{u+(1-v)} \\ P_{1,4} &= \frac{v P_{1,4,u} + (1-u) P_{1,4,v}}{(1-u)+v} & P_{4,4} &= \frac{(1-v) P_{4,4,u} + (1-u) P_{4,4,v}}{(1-u)+(1-v)} \end{aligned} \quad (35)$$

Como uma implicação deste modelo, a superfície B-Spline com princípios de retalhos de Gregory pode ser entendida como uma superfície na qual os pontos de controle P_D possuem uma posição possivelmente diferente para cada par de valor paramétrico utilizado. Pode-se perceber que assim como ocorre nas superfícies B-Spline regulares, alguns pontos de controle podem não influenciar no cálculo das funções de mistura da superfície para determinados valores paramétricos. Deste modo, os pontos de controle direcionais também mantêm a característica de influência local. Por exemplo, considerando a Figura 17, os pontos de controle $P_{1,1,u}$ e $P_{1,1,v}$ (e por consequência $P_{1,1}$ que é constituído por eles) não possuem influência sobre os pontos da superfície próximos a quina da outra extremidade.

Através das características deste modelo de superfície proposto, pode-se evitar o problema da incompatibilidade de torção resolvendo cada condição de igualdade isoladamente. Por exemplo, o ponto de controle $P_{1,1,u}$ da Figura 17 pode ser ajustado de tal forma a garantir as condições de C^1 com uma superfície que se liga na borda inferior, ao passo que o ponto de controle $P_{1,1,v}$ pode ser ajustado garantindo as condições de C^1 em relação a uma superfície que se conecta na borda esquerda da superfície ilustrada. Através do efeito da função de mistura racional que pondera entre estes dois pontos de controle direcionais, ao avaliar valores paramétricos próximos à borda inferior, o ponto $P_{1,1,u}$ terá maior influência sobre $P_{1,1}$, tornando-o compatível com a condição de continuidade desta borda. Já ao avaliar os valores paramétricos próximos à borda esquerda, o ponto $P_{1,1,v}$ terá maior influência sobre $P_{1,1}$, tornando-o compatível com a condição de continuidade desta borda.

4.2.2 Definição do Problema

Considere o problema de ajustar um conjunto conectado de r retalhos de superfícies paramétricas retangulares B-Spline, $\mathcal{S} = \{s_{[1]}, \dots, s_{[r]}\}$, sobre uma nuvem de pontos \mathbf{N} , composta por n amostras, onde cada superfície $s_{[i]}$ está relacionada a um subconjunto $N^{[i]}$ da nuvem de pontos com uma parametrização $\tau^{[i]} = \{(u_1^{[i]}, v_1^{[i]}), \dots, (u_{|N^{[i]}|}^{[i]}, v_{|N^{[i]}|}^{[i]})\}$. Dado um limiar de erro tolerável $\epsilon_{tol}^{[\gamma]}$, referente a uma métrica de erro γ – onde $\gamma \in \{\text{RMSE}, \text{ME}\}$ – o objetivo é encontrar um ajuste de superfície, usando a menor quantidade possível de pontos de controle (ρ), de tal forma que o erro de ajuste $\epsilon^{[\gamma]}$ da métrica considerada seja menor do que o limiar $\epsilon_{tol}^{[\gamma]}$. Ao utilizar $\gamma = \text{RMSE}$, o objetivo é encontrar uma solução onde:

$$\sqrt{\frac{1}{n} \sum_{i=1}^r \sum_{j=1}^{|N^{[i]}|} \|N_j^{[i]} - s_{[i]}(u_j^{[i]}, v_j^{[i]})\|^2} \leq \epsilon_{tol}^{[\text{RMSE}]} \quad (36)$$

De forma similar, ao utilizar $\gamma = \text{ME}$, o objetivo é encontrar uma solução onde:

$$\max_{\substack{i \in [1, r] \\ j \in [1, |N^{[i]}|]}} \|N_j^{[i]} - s_{[i]}(u_j^{[i]}, v_j^{[i]})\| \leq \epsilon_{tol}^{\text{[ME]}} \quad (37)$$

Adicionalmente, pode-se estabelecer uma restrição para a quantidade máxima de pontos de controle (ρ_{max}) a ser utilizada no ajuste. Esta restrição pode ser vista como um mecanismo de compressão que garante um certo nível de simplicidade no modelo resultante. No entanto, ao utilizá-lo, pode-se impedir a obtenção de uma solução que atenda ao objetivo de minimização de erro proposto. Considerando que a notação $|P^{[i]}|$ representa a quantidade de pontos de controle da superfície $s_{[i]}$ e r representa o número de retalhos, a restrição pode ser escrita como:

$$\sum_{i=1}^r |P^{[i]}| \leq \rho_{max} \quad (38)$$

4.2.3 Visão Geral do Método

O método DCBA proposto para o ajuste de múltiplas superfícies recebe como entrada dois valores: uma nuvem de pontos N representando um objeto com topologia homeomórfica a um disco – ou que possua uma borda de corte que possa ser utilizada para planificar o objeto e obter tal topologia – e sua parametrização τ . No caso de nuvens organizadas, a parametrização pode ser calculada de forma eficiente através do método de comprimento de corda, por exemplo. Já no caso de nuvens semi-organizadas ou não organizadas, os métodos de parametrização podem exigir um tempo de execução significativo, dependendo da quantidade de amostras da nuvem de pontos. Tendo isto em vista, considerou-se adequado realizar um único procedimento de parametrização, diferentemente do que foi feito no caso do ajuste de múltiplas curvas, uma vez que sua execução repetitiva poderia comprometer a eficiência do método DCBA. Ademais, ao fazer isso, permite-se a desvinculação entre o procedimento de parametrização e o ajuste de superfície propriamente dito, e viabiliza a utilização dos métodos que se julgam mais apropriados para a realização desta operação.

Com os dados de entrada recebidos, o método DCBA calcula os parâmetros padrões a serem utilizados na sua execução ou permite que estes sejam adaptados manualmente. Se por um lado a dispensa da definição de parâmetros simplifica o uso do método, por outro, a possibilidade de adaptar manualmente parâmetros de execução pode viabilizar o refinamento de resultados. Os parâmetros ajustáveis do método proposto e seus respectivos símbolos são: o grau $d = (d_u; d_v)$ e a quantidade de pontos de controle inicial ψ^2 de cada retalho de superfície; um valor opcional que estabelece uma quantidade máxima de amostras ζ para a realização de ajuste de superfícies; a métrica de erro γ e um valor de tolerância relativo a esta métrica $\epsilon_{tol}^{[\gamma]}$; uma quantidade máxima de pontos de controle ρ_{max} ; um método χ de particionamento da nuvem de pontos em dois subconjuntos; um multiplicador de amostragem δ empregado na discretização dos retalhos de superfície; e um fator de redução de pontos de controle ω . A Tabela 5 lista estes parâmetros apresentando o intervalo de valores sugeridos e uma lista com os valores padrões.

Tabela 5 – Lista de parâmetros para o método de ajuste por múltiplas superfícies

Parâmetro	Intervalo Sugerido	Valor Padrão
$(d_u; d_v)$	$2 \leq d_u, d_v < \psi$	$(3; 3)$
ψ	$d < \psi \leq \left\lfloor \sqrt{ N } \right\rfloor$	$\psi = \max(7; \left\lceil \sqrt[4]{ N } \right\rceil)$
ζ	$\psi^3 \leq \zeta \leq \infty$	$\psi^{\frac{7}{2}}$
γ	{RMSE, ME}	ME
$\varepsilon_{tol}^{[\gamma]}$	$\varepsilon_{tol}^{[\gamma]} \in \mathbb{R}$	MMD*
ρ_{max}	$\rho_{max} \geq \psi^2$	∞
χ	{PBS, PMaC, PMeC}**	PBS
δ	$1 < \delta \leq 50$	10
ω	$0,1 \leq \omega \leq 0,5$	0,1

* MMD = Média das Menores Distâncias entre cada amostra da nuvem e seus vizinhos

** PBS = Partição por Bisseção; PMaC = Partição na Maior Curvatura;
PMeC = Partição na Menor Curvatura

Fonte: Elaborado pelo autor (2022).

Na sequência, inicia-se um laço de repetição iterativo de conquista e divisão. No procedimento de conquista, uma nova superfície B-Spline de grau d e $\psi \cdot \psi$ pontos de controle é criada e ajustada para cada elemento de uma lista η de subconjunto da nuvem de pontos original N e suas parametrizações. O conjunto de pontos de controle $P^{[i]}$ de cada superfície paramétrica $s_{[i]}$ é otimizado através de mínimos quadrados e o erro de ajuste $\varepsilon_{[i]}^{[\gamma]}$ é calculado para cada uma destas superfícies com base na métrica γ . Todas as superfícies $s_{[i]}$ são candidatas a superfícies resultantes, mas aquelas que não obtiverem um erro de ajuste tolerável são analisadas para possível particionamento. Assim, uma tupla $(N^{[i]}; \varepsilon_{[i]}^{[\gamma]})$ é adicionada em uma lista de candidatos a particionamento (μ), ordenada pelos valores do erro de ajuste, para toda superfície $s_{[i]}$ na qual $\varepsilon_{[i]}^{[\gamma]} > \varepsilon_{tol}^{[\gamma]}$. Para finalizar o procedimento, todos os elementos da lista η são removidos.

O procedimento de divisão recebe a lista μ , com os subconjuntos de N que ainda não foram ajustados abaixo do erro tolerável, e inicia iterando sobre os elementos desta lista, escolhendo sempre a tupla $(N^{[i]}; \varepsilon_{[i]}^{[\gamma]})$ com o maior erro de ajuste calculado. O objetivo é encontrar uma única linha de divisão no domínio paramétrico, a qual pode ser posicionada na vertical ou na horizontal, e que irá cortar o subconjunto $N^{[i]}$ da nuvem de pontos e também outros possíveis subconjuntos interceptados. Primeiramente, para escolher a orientação de corte, realiza-se uma varredura de todos os subconjuntos da nuvem que estão na mesma linha vertical ou horizontal do subconjunto escolhido $N^{[i]}$ no domínio paramétrico. Nesta varredura, somam-se todos os erros de ajuste dos subconjuntos da nuvem de pontos que são interceptados pela linha considerando a métrica γ . Escolhe-se então a orientação que somar o maior valor de erro.

Com a escolha da orientação realizada, um método de particionamento χ escolhe uma posição no espaço paramétrico para a linha de corte. Tanto o método χ como a escolha da posição serão discutidos adiante neste capítulo. Caso uma linha de corte não possa ser definida por alguma restrição, o particionamento é descartado e o próximo elemento de maior erro na lista de candidatos a particionamento μ passa a ser analisado. Caso contrário, os subconjuntos da nuvem de pontos tocados por esta linha são substituídos pelos novos subconjuntos gerados, os quais têm sua parametrização ajustada e são adicionados à lista η . A etapa iterativa de conquista e divisão se repete enquanto a lista η não estiver vazia.

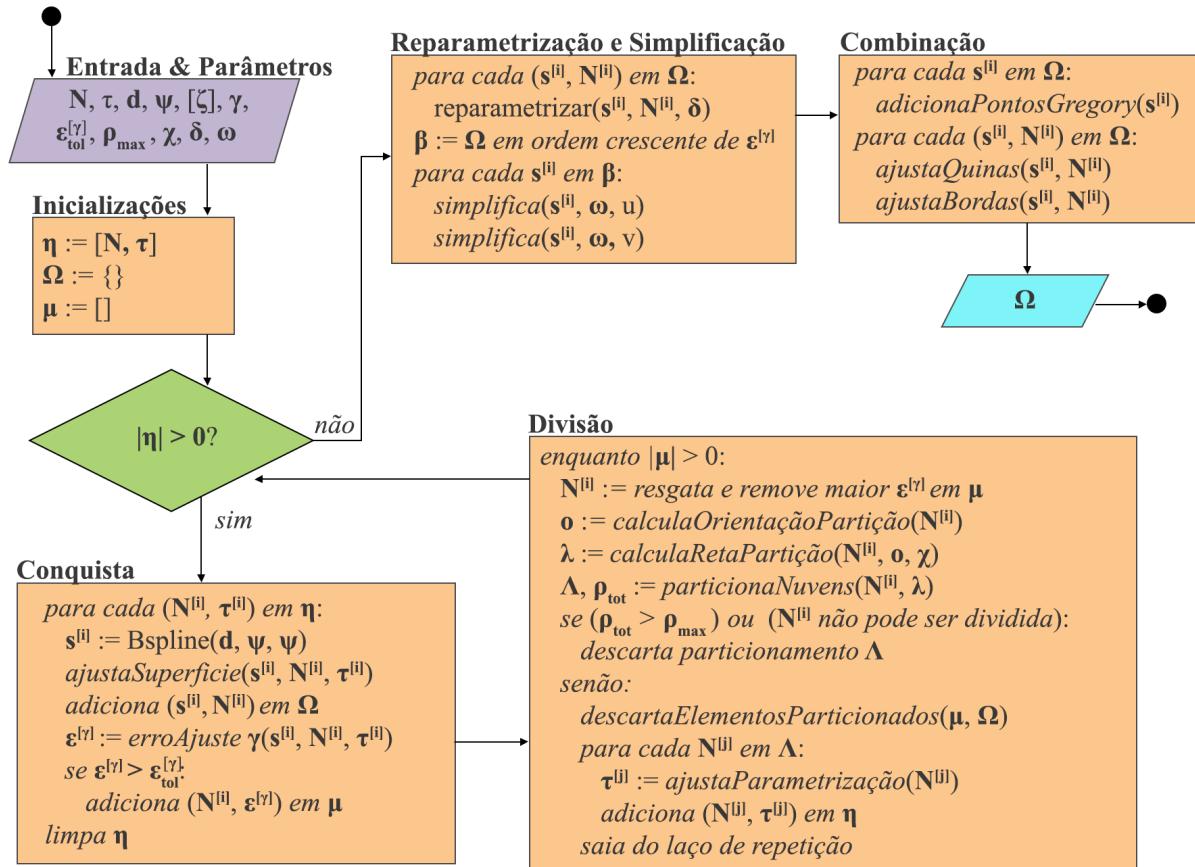
Note que diferentemente do método de ajuste de múltiplas curvas, o método proposto não realiza um procedimento de combinação entre as superfícies ajustadas dentro do laço de repetição principal. Esta decisão foi tomada com base no custo temporal de execução deste procedimento, que é significativamente mais elevado do que no caso das curvas. Isto ocorre porque, ao invés do ajuste de apenas duas pontas, como necessário no caso das curvas, neste problema é necessário ajustar todas as quatro quinas e as quatro bordas de cada superfície. Além de uma quantidade maior de ajustes necessários, o peso de cada um destes ajustes também é superior, uma vez que a quantidade de amostras das nuvens de pontos e a quantidade de pontos de controle costumam ser maiores. No entanto, esta decisão traz também um impacto negativo na tentativa de alcançar o objetivo de otimização do problema. Ao deixar a combinação fora do laço de repetição principal, não é possível garantir que o erro total de ajuste, após as combinações entre as superfícies, continue abaixo do limiar de erro estabelecido. Apesar disto, percebeu-se experimentalmente que a impossibilidade de divisão da nuvem de pontos costuma ser o principal motivo para a não obtenção de um erro de ajuste menor do que o limiar estabelecido, e o aumento no erro de ajuste devido a combinação tem menor impacto neste problema.

Com o término do laço de repetição iterativo do algoritmo, tem-se como resultado parcial um conjunto de superfícies desconexas umas das outras, todas com $\psi \cdot \psi$ pontos de controle. O próximo passo poderia ser simplesmente a combinação destas superfícies, mas desta forma a quantidade de pontos de controle não estaria sendo efetivamente otimizada. Assim, antes da realização da operação final de combinação entre as superfícies, realiza-se primeiramente um procedimento de reparametrização das superfícies, com base em um fator de discretização δ , para refinar os valores numéricos do erro de ajuste. Note que esta reparametrização não altera a geometria das superfícies, mas busca aprimorar a relação entre as amostras e seu valor paramétrico associado. Na sequência, um procedimento de simplificação dos retalhos de superfície é realizado através da tentativa de diminuição de pontos de controle utilizados. Esta redução é realizada selecionando as superfícies de menor erro de ajuste, tentando reduzir iterativamente a quantidade de pontos de controle baseado em um fator de redução ω . Isto é feito para cada orientação paramétrica possível, de modo individual, afetando todas as superfícies que estão na mesma linha horizontal ou vertical, possibilitando assim a obtenção de uma matriz de controle geral na qual o número de linhas e colunas não seja necessariamente igual.

Com a finalização do procedimento de simplificação das superfícies, inicia-se então

o procedimento final de combinação. Neste procedimento são realizados dois tipos distintos de otimização por mínimos quadrados: um para o ajuste individual de cada ponto de quina e os pontos de controle adjacentes a eles em ambas as orientações; outro para o ajuste de cada borda individual. Antes da realização destes ajustes, no entanto, todas as superfícies recebem os ditos pontos de Gregory, os quais viabilizam a combinação de superfícies em ambas as orientações paramétricas com uma continuidade C^1 , evitando o problema da compatibilidade de torção. Após executar o procedimento de combinação para cada retalho de superfície, o método proposto retorna um conjunto de retalhos de superfície que representam a nuvem de pontos original, com nível de continuidade C^1 para cada junção de superfícies e possivelmente – mas não necessariamente – com um erro de ajuste menor que $\epsilon_{tol}^{[\gamma]}$, ou um valor próximo a este. Para complementar a descrição da visão geral do método, apresenta-se a Figura 18, com as principais etapas em forma de fluxograma com pseudocódigo de alto nível.

Figura 18 – Visão geral do método de ajuste por múltiplas superfícies – macro procedimentos



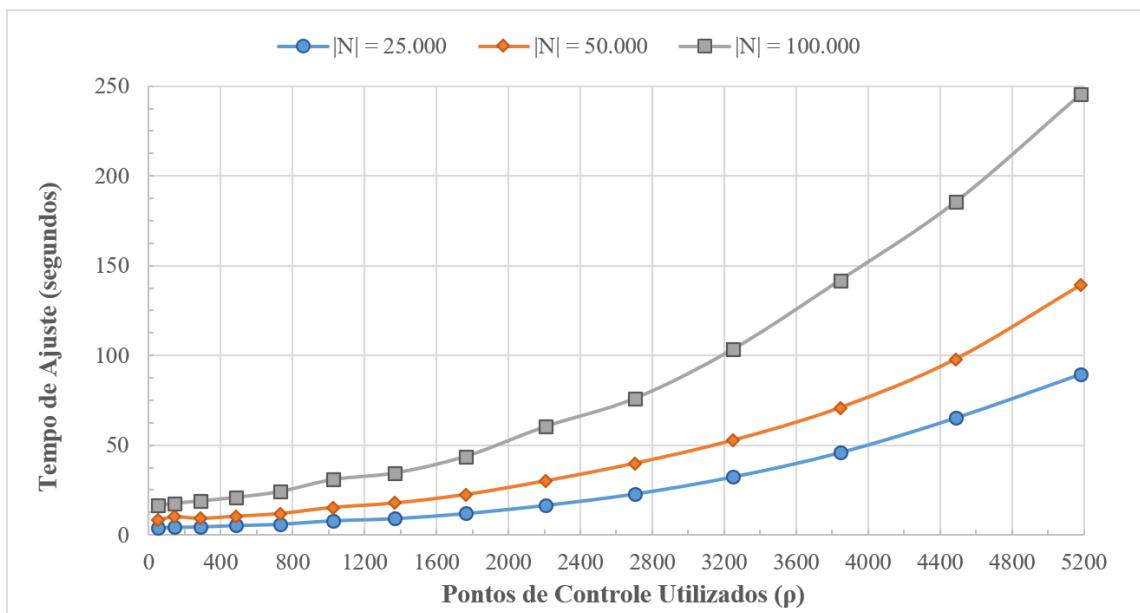
Fonte: Elaborado pelo autor (2022).

Na sequência deste capítulo, cada procedimento do algoritmo será melhor detalhado. Seraõ abordadas as dificuldades enfrentadas em determinadas etapas, as motivações que resultaram em adaptações no método proposto e demais observações relevantes sobre os procedimentos. Também serão apresentados alguns mecanismos opcionais de aceleração que foram elaborados e testados, com o intuito de melhorar a eficiência do método.

4.2.4 Procedimento de Conquista

O procedimento de conquista foi uma das etapas que necessitou a menor quantidade de alterações em relação ao que foi realizado no ajuste de múltiplas curvas. O princípio fundamental utilizado nesta etapa está relacionado com uma observação referente a uma análise de complexidade. Considere a complexidade de tempo referente a resolução de um sistema linear para efetuar o ajuste de uma superfície otimizando os seus pontos de controle. Os principais métodos da literatura podem realizar tal otimização com uma complexidade de $\mathcal{O}(n\rho^2)$, onde n representa a quantidade de amostras da nuvem ($n = |N|$) e ρ representa a quantidade de pontos de controle utilizados (GOLUB; LOAN, 1996; EBRAHIMI; LOGHMANI, 2018). O problema em ajustes tradicionais de nuvens de pontos com uma grande quantidade de amostras é que, além do maior valor de n contribuir linearmente no tempo de computação necessário, as nuvens com mais amostras costumam requerer uma maior quantidade de pontos de controle para obter um resultado com um menor erro de ajuste. Essa maior quantidade de pontos de controle, por sua vez, contribui de forma quadrática no tempo de computação. Para ilustrar este crescimento, a Figura 19 apresenta este aumento quadrático em três casos de exemplo: uma nuvem de pontos com 25 mil amostras, outra com 50 mil amostras e uma terceira com 100 mil amostras. Em todos os casos os ajustes de superfícies foram realizados de forma tradicional, com uma superfície B-spline bicúbica de mesmas dimensões de linhas e colunas no poliedro de controle, totalizando ρ pontos de controle.

Figura 19 – Análise do tempo de ajuste de superfície utilizando o método tradicional



Fonte: Elaborado pelo autor (2022).

Considerando os princípios da estratégia de divisão e conquista, a divisão do problema original em subproblemas menores, de mesma natureza, permite a redução de ambas as variáveis envolvidas no custo da operação, especialmente a quantidade de pontos de controle. Por

outro lado, quanto menor a quantidade de pontos de controle utilizada para cada ajuste de um subproblema, maiores as chances do ajuste realizado não alcançar o erro de ajuste tolerável e, por consequência, mais operações de divisão e ajuste serão realizadas. Com base nestes aspectos, sugere-se a utilização de um total de \sqrt{n} pontos de controle para cada superfície, onde n é a quantidade de amostras da nuvem de pontos original. Em virtude da dificuldade em escolher a melhor distribuição de linhas e colunas destes poliedros de controle e, considerando também a necessidade de equivalência destes valores para as superfícies que pertencem a mesma linha ou coluna, optou-se por utilizar inicialmente apenas poliedros de controle quadrados. Isto implica em uma matriz de controle com dimensões $\psi \cdot \psi$, onde $\psi = \lceil \sqrt[4]{n} \rceil$. Assim, a complexidade de tempo de execução de cada ajuste necessário será reduzida para $\mathcal{O}(n_{[i]}\sqrt{n})$, onde $n_{[i]}$ é o tamanho do i -ésimo subconjunto da nuvem de pontos. Além disso, a limitação estrutural da matriz de controle neste procedimento não se torna um empecilho para a simplicidade do resultado, uma vez que este problema será tratado adiante, na etapa de simplificação das superfícies.

Diferentemente do que é feito no método de ajuste de múltiplas curvas, optou-se por não calcular uma nova parametrização sempre que um subconjunto da nuvem de pontos é dividido. Considerou-se inadequado fazer isto, visto que no caso de nuvens de pontos semi-estruturadas ou não organizadas, os métodos da literatura consumiriam um tempo significativo neste procedimento. Todavia, uma função de ajuste de parametrização por requadramento é utilizada logo após o particionamento dos subconjuntos da nuvem de pontos. Esta função de requadramento aloca as quatro amostras mais próximas das novas quinas do espaço paramétrico nestas posições e ajusta os demais valores paramétricos das amostras por interpolação da sua posição atual em relação ao deslocamento das quatro amostras de quina.

Com relação aos parâmetros das superfícies de ajuste, foram utilizadas somente superfícies bi-cúbicas, i.e. $d = (3; 3)$, assim como na maioria dos métodos da literatura que fixam os graus da superfície. Por fim, vale destacar ainda que, além do custo do ajuste das superfícies propriamente dito – que apresenta o maior impacto sobre o tempo do procedimento de conquista – o custo do cálculo do erro de ajuste é o segundo fator que mais contribui no aumento do tempo de execução. No caso de ajuste de curvas, foi implementado um mecanismo de interrupção seletiva do erro de ajuste, que interrompe o cálculo assim que o erro parcial extrapola o limiar de erro tolerável estabelecido. No caso das superfícies, no entanto, este mecanismo foi abolido ao perceber que ele limitava a qualidade de escolha no procedimento de divisão. Isto ocorre porque no caso das curvas, a divisão de um segmento não influencia diretamente na eventual divisão de outro segmento. Já no caso das superfícies, como apenas um único retalho é selecionado como ponto central para o particionamento de possivelmente vários subconjuntos da nuvem de pontos, esta escolha deve ser ter base em algum critério de seleção, pois possui um impacto global no particionamento.

4.2.5 Procedimento de Divisão

O procedimento de divisão necessitou diversas considerações e adaptações em relação ao mesmo procedimento no ajuste de múltiplas curvas. Um tempo significativo foi despendido nesta etapa do trabalho, buscando encontrar uma solução eficiente para este subproblema e suas implicações no procedimento de combinação. A seguir é apresentado um breve resumo da trajetória de pesquisa nesta operação, visto que as dificuldades, considerações e ideias que emergiram durante o processo podem ser úteis de alguma forma para trabalhos futuros. A argumentação apresentada tem uma estreita relação com o procedimento de combinação, uma vez que é através da forma com que a divisão é realizada que se manifestam características e subproblemas nesta etapa subsequente.

Antes de apresentar o resumo de pesquisa sob este procedimento, deve-se enfatizar que, no ajuste utilizando superfícies paramétricas com uma estrutura de nós fixa, em forma de matriz (como no caso de superfícies de Bézier, B-Splines e NURBS), a divisão não ocorre com base em um ponto de referência sobre a nuvem de pontos, mas sim uma reta ou curva de corte no espaço paramétrico, a qual subdivide a nuvem de pontos em duas partes. Por simplificação e, considerando a estrutura retangular do espaço paramétrico das superfícies utilizadas, optou-se pela utilização de retas de corte somente nas orientações de base do espaço paramétrico – os valores paramétricos u e v . Em uma analogia, são permitidos apenas cortes verticais e horizontais no espaço paramétrico.

4.2.5.1 Investida com Particionamento Local e Junções T

Ao realizar uma divisão de apenas um subconjunto da nuvem de pontos que possui relação de vizinhança com outros subconjuntos, cria-se a necessidade de junções em T entre as superfícies, pois as bordas não são totalmente compartilhadas. Prosseguir com esta situação gera um complicador no procedimento de combinação, pois uma borda completa de uma superfície deve ser compartilhada e combinada – com um grau de continuidade – com apenas um intervalo de borda de outra superfície vizinha. Buscou-se por algum método na literatura que atingisse o objetivo de ajustar uma curva completa sobre um subintervalo de outra curva, com a mesma quantidade de pontos de controle, mas nenhum método foi encontrado.

Considerando que a borda completa de uma das superfícies tem mais variáveis de otimização para se ajustar fielmente a um intervalo da outra borda, tentou-se obter continuidade C^0 através de mínimos quadrados, utilizando um vetor de nós uniforme aberto. Esta solução se mostrou válida, com um erro residual menor do que dez casas decimais para vários casos de teste, mas em algumas situações, principalmente quando há muitas inflexões no início e no fim do intervalo da curva de borda, a solução não atinge o objetivo desejado. Identificou-se então, a possibilidade de tornar as influências dos vetores de nós – sobre as funções de mistura de ambas as curvas – o mais próximo possível, através de uma alocação adaptativa destes vetores.

Utilizando-se de um vetor de nós não uniforme, resolve-se um problema, mas outro é

criado: nos casos de junção T em ambas as bordas de uma mesma orientação paramétrica, o vetor de nós precisaria ter um conjunto de valores em uma borda e outro conjunto de valores na outra borda. A alternativa cogitada para resolver este problema foi sugerir um novo tipo de superfície na qual existem quatro vetores de nós, um para cada borda da superfície, sendo que o valor real do vetor de nós empregado na função paramétrica da superfície seria obtido com uma interpolação linear entre os vetores de nós em uma mesma orientação. Com esta nova superfície sugerida, o problema de ajuste com duplas junções em T foi sanado e conseguiu-se obter uma aparente continuidade C^0 para cada borda com erro desprezível. Todavia, quando os vetores de nós das duas bordas de uma orientação são muito diferentes, os resultados não se mostraram visualmente satisfatórios. Desta forma, optou-se por seguir uma estratégia ligeiramente diferente.

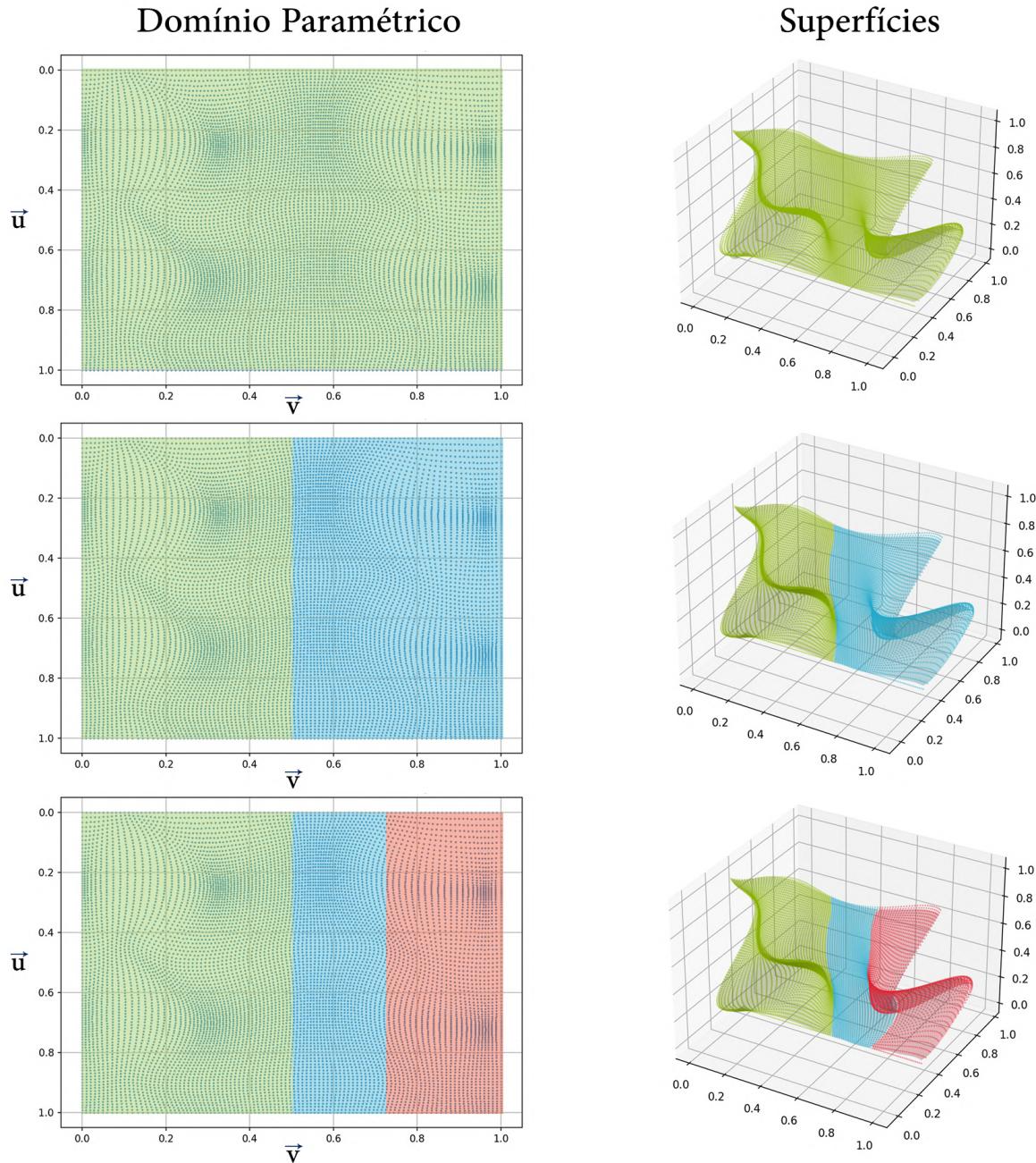
4.2.5.2 *Métodos de Particionamento sem Junções T*

A abordagem escolhida para contornar o problema da geração e posterior combinação entre superfícies com junções em T foi evitar a necessidade de tal tipo de junção. Para tanto, decidiu-se que ao invés de realizar um corte de divisão horizontal ou vertical para uma única superfície e seu respectivo subconjunto da nuvem de pontos, o corte de divisão seria estendido para todo o intervalo paramétrico. Em outras palavras, em um corte vertical, todos os subconjuntos da nuvem de pontos da coluna selecionada são particionados; o mesmo é feito, analogamente, para o caso de um corte horizontal. Assim, garante-se que todas as bordas sejam compartilhadas integralmente entre superfícies adjacentes. Para simplificar o procedimento de divisão, ao considerar que múltiplas nuvens de pontos podem ser particionadas por um único corte de divisão, decidiu-se limitar a um único corte por iteração, já que múltiplos cortes simultâneos poderiam particionar um mesmo subconjunto diversas vezes. A Figura 20 demonstra alguns passos do processo iterativo de conquista e divisão. As imagens da esquerda representam o domínio paramétrico, ao passo que as imagens da direita representam o espaço \mathbb{R}^3 , com as nuvens de pontos e as superfícies ajustadas a cada iteração.

Como mencionado na visão geral do método proposto, a lista μ armazena os subconjuntos com um ajuste de erro maior do que $\epsilon_{tol}^{[\gamma]}$, classificados em ordem decrescente pelo valor de erro de ajuste na métrica γ . Esta estratégia foi elaborada de modo a privilegiar a redução do maior erro de ajuste na iteração corrente e considerando também o caso em que um valor de ρ_{max} é definido, o que pode restringir o refinamento das superfícies. Se um retalho de superfície aleatório fosse escolhido, a superfície de maior erro local de ajuste poderia continuar sem ser subdividida, resultando na conservação do pior erro de ajuste.

Após a escolha do retalho de superfície candidato a divisão, calcula-se primeiramente a orientação de corte que será utilizada. Esta análise é feita no domínio paramétrico considerando ambas as orientações (orientação relacionada ao parâmetros u e v). É escolhida aquela na qual for encontrada a maior média da métrica de erro γ , considerando a superfície analisada e as demais superfícies que se encontram na mesma linha ou coluna, dependendo da orientação. Note que, ao utilizar a média, não se privilegia nem se desfavorece uma orientação que possui maior

Figura 20 – Demonstração de uma sequência iterativa de procedimentos de conquista e divisão



Fonte: Elaborado pelo autor (2022).

quantidade de retalhos de superfície.

Com uma orientação válida e de maior média de erro para a reta divisória, o próximo passo é determinar um valor paramétrico α para esta reta. Este valor é escolhido de modo que cada subconjunto da nuvem de pontos $N^{[i]}$, cortados por esta linha no domínio paramétrico, seja subdividido em dois novos subconjuntos $N^{[j]}$ e $N^{[k]}$. No caso de uma reta divisória paralela à direção do valor paramétrico v , $N^{[j]}$ é composto por todos os pontos em $N^{[i]}$ que apresentam valor paramétrico na direção do valor paramétrico u menor ou igual a α e $N^{[k]}$ é composto pelos demais pontos de $N^{[i]}$. Os pontos de $N^{[j]}$ e $N^{[k]}$ são escolhidos de forma análoga no caso de uma reta

divisora paralela à direção do valor paramétrico u . Como é inviável otimizar o valor de α , a fim de obter a divisão de subconjuntos da nuvem que gere o menor erro de ajuste, foram elaborados três métodos de particionamento similares aos utilizados no caso de ajuste de múltiplas curvas, os quais utilizam certos critérios para esta seleção. Estes métodos seguem princípios encontrados na literatura e seus detalhes de funcionamento serão apresentados a seguir.

Antes de um particionamento ser efetuado, o procedimento verifica se todos os novos subconjuntos envolvidos possuem uma quantidade mínima de amostras (n_{min}). Essa quantidade foi fixada como: $n_{min} = 10\psi^2$ e é justificada como um mecanismo que tenta evitar a ocorrência de *overfitting* e de um sistema linear mal condicionado, os quais têm maiores probabilidades de ocorrer quando há poucas amostras e estas estão mal espalhadas pelo domínio paramétrico. Assim, se alguma nuvem de pontos não puder ser cortada por causa de alguma restrição, ou se a quantidade total de pontos de controle necessária para ajustar os novos subconjuntos (gerados pela divisão) ultrapassar o limite ρ_{max} estabelecido, o particionamento é descartado.

Por fim, caso o particionamento seja válido, realiza-se um ajuste de parametrização por requadramento nos novos subconjuntos gerados. Neste processo, os valores paramétricos são normalizados dentro do quadrado unitário paramétrico (quadramento), e os pontos mais próximos de cada quina do quadrado unitário são movidos para estas respectivas posições, garantindo uma amostra em cada quina. Através deste procedimento, evita-se a ocorrência de transposição entre vizinhos, o que altera a correspondência de vizinhança e pode gerar artefatos estranhos no ajuste posterior. Em contrapartida, diferentemente do que é realizado por Mao et al. (2018), o simples quadramento utilizado não assegura que ao menos um parâmetro seja alocado para cada intervalo de nós, o que garantiria que a matriz gerada para o sistema linear seja bem condicionada. Após o ajuste de parametrização, os novos subconjuntos ($N^{[j]}$ e $N^{[k]}$) da nuvem de pontos e suas respectivas parametrizações ($\tau^{[j]}$ e $\tau^{[k]}$) são adicionados à lista η , os quais serão submetidos ao procedimento de conquista.

4.2.5.3 *Métodos de Particionamento*

Foram implementados e testados três métodos de particionamento dos subconjuntos da nuvem de pontos: o método de partição por bisseção (PBS), o método de partição pela maior curvatura (PMaC) e o método de partição pela menor curvatura (PMeC); cada um com suas próprias motivações. O primeiro deles, o PBS, calcula a linha de partição (em uma dada orientação paramétrica) de modo a partitionar o subconjunto de amostras pela metade, tentando manter a menor diferença possível de elementos nos novos subconjuntos resultantes. Este método de partição foi motivado pela busca por um bom balanceamento na quantidade de amostras do retalho de superfície utilizado como referência. A divisão pela metade também é guiada por motivos de eficiência: os algoritmos de divisão e conquista para ordenação conseguem obter o melhor desempenho quando as divisões são realizadas desta forma.

Os outros dois métodos de particionamento, como os próprios nomes já sugerem, são calculados com base em um dado de curvatura. Para computar a curvatura, utilizou-se o método de

aproximação da curvatura média por amostra da nuvem de pontos, baseada no algoritmo utilizado por Miyazaki et al. (2021) e apresentado na subseção 2.1.3. Como este valor é computado para cada amostra da nuvem de pontos, pode-se utilizar este algoritmo uma única vez, armazenando e reutilizando os valores de curvatura. Com os valores de curvatura computados, o método PMaC utiliza o valor paramétrico da amostra (do subconjunto da nuvem de pontos) que possui o maior valor de curvatura, ao passo que o método PMeC utiliza analogamente o menor valor de curvatura. Para evitar a seleção de maior ou menor curvatura próximas à borda dos retalhos de superfície – o que aumenta as chances de uma divisão inválida, uma vez que existe maior probabilidade de poucas amostras serem alocadas para um subconjunto e a maioria das amostras serem alocada para outro subconjunto – estes métodos de particionamento utilizam uma margem de valores não selecionáveis, limitando o espaço de valores paramétricos que podem ser escolhidos. Assim, neste trabalho, considerando o espaço paramétrico normalizado (entre zero e um), os métodos de particionamento baseados em curvatura consideram apenas o intervalo: $0,3 \leq u \leq 0,7$ ou $0,3 \leq v \leq 0,7$, dependendo da orientação do corte.

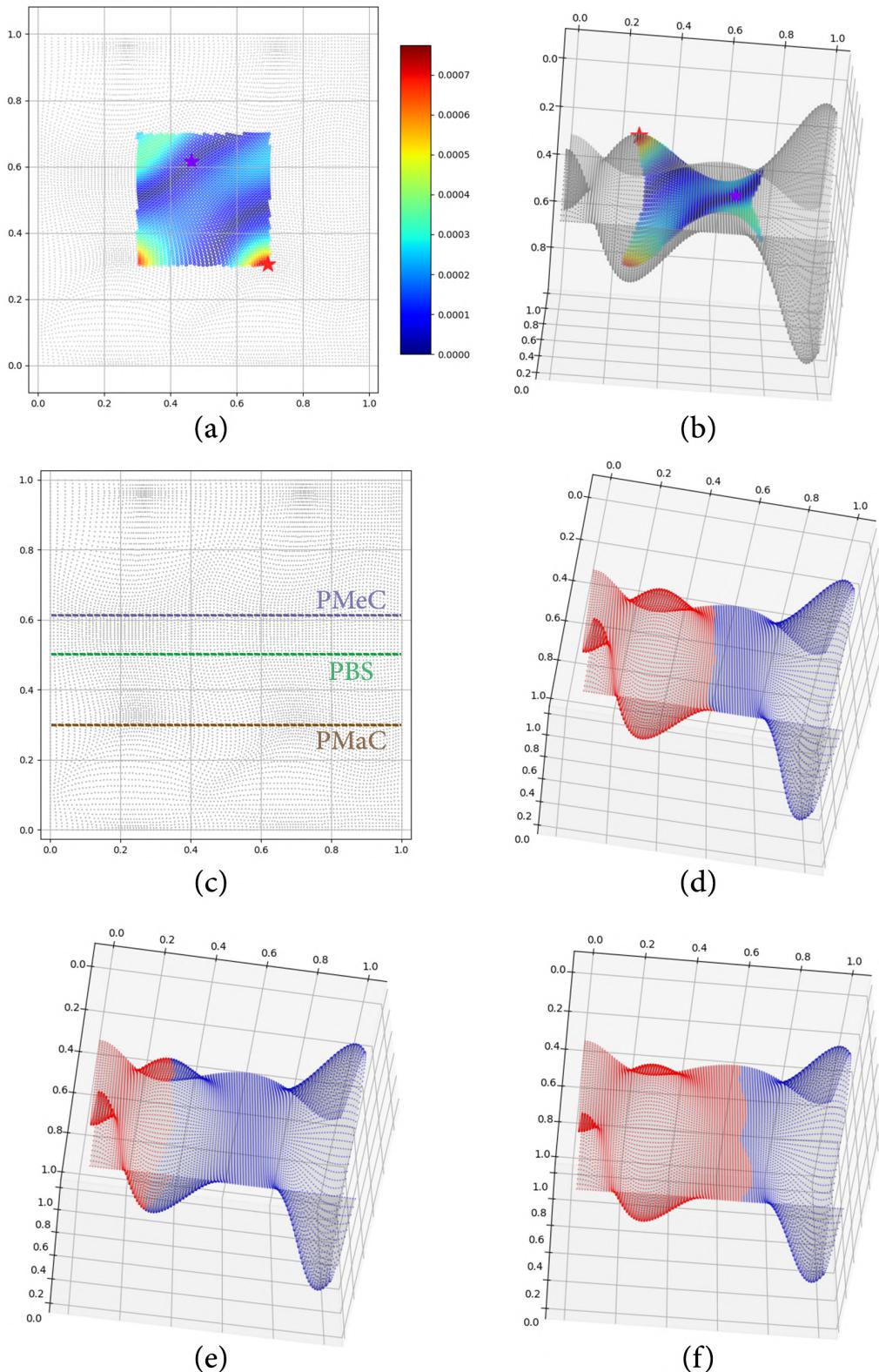
A Figura 21 ilustra um exemplo de aplicação dos métodos de particionamento implementados. A Figura 21(a) e Figura 21(b) ilustram uma nuvem de pontos, já parametrizada, no domínio paramétrico e na sua representação geométrica, respectivamente. Ambas as imagens destacam os valores de curvatura computados através do algoritmo de aproximação utilizado. Os valores de curvatura utilizam as cores destacadas na barra entremeio as duas imagens, sendo que os maiores valores de curvatura são representados em vermelho, ao passo que os menores valores de curvatura são representados em azul. O ponto de maior curvatura foi destacado por uma estrela vermelha e o ponto de menor curvatura foi destacado por uma estrela roxa. Note ainda que diversos pontos não foram coloridos, pois não estão dentro do intervalo considerado para o particionamento.

Ao utilizar os métodos de particionamento implementados e uma orientação de corte horizontal no domínio paramétrico, foram geradas três linhas de corte, uma para cada método, conforme ilustrado na Figura 21(c). O método de PBS resultou em uma linha sobre o valor $u = 0,5066$ (o parâmetro u é indicado na vertical e o parâmetro v é indicado na horizontal); a linha verde na Figura 21(c). O resultado da partição pelo método de PBS no domínio geométrico é apresentado na Figura 21(d). Já o método de PMaC resultou em uma linha sobre o valor $u = 0,3063$; a linha marrom na Figura 21(c), sendo que o resultado da partição por este método no domínio geométrico é apresentado na Figura 21(e). Por fim, o método de PMeC resultou em uma linha sobre o valor $u = 0,6155$; a linha roxa na Figura 21(c), sendo que o resultado da partição por este método no domínio geométrico é apresentado na Figura 21(f).

4.2.6 Procedimento de Reparametrização

A reparametrização é o primeiro procedimento da segunda etapa do método proposto, após o ciclo de operações de divisão e conquista. Esta etapa tem dois propósitos fundamentais: aproximar melhor o erro de ajuste e aprimorar a parametrização para futuros ajustes. A repara-

Figura 21 – Exemplo de divisão realizada pelos métodos de particionamento implementados



Fonte: Elaborado pelo autor (2022).

trização pode ser realizada de diversas formas e, para este trabalho, optou-se pela estratégia de discretização dos retalhos de superfície, de modo similar ao que foi feito por Ueda et al. (2020) e

ao que foi realizado no método para ajuste de múltiplas curvas.

Na reparametrização utilizada, o parâmetro δ é utilizado para determinar o detalhamento da discretização. Considere uma superfície paramétrica $s_{[A]}$ que ajusta um subconjunto $N^{[A]}$ da nuvem de pontos original. A superfície $s_{[A]}$, composta por $\psi \cdot \psi$ pontos de controle, é inicialmente discretizada por $\psi^2 \cdot \delta$ amostras com valores paramétricos uniformemente distribuídos. Na sequência, calcula-se a menor distância entre cada amostra de $N^{[A]}$ e os pontos discretos da superfície $s_{[A]}$, associando os valores paramétricos da menor distância encontrada. Ao final, obtém-se novos valores paramétricos que são comparados à parametrização original, sendo escolhida a alternativa de menor distância, i.e. menor erro de ajuste.

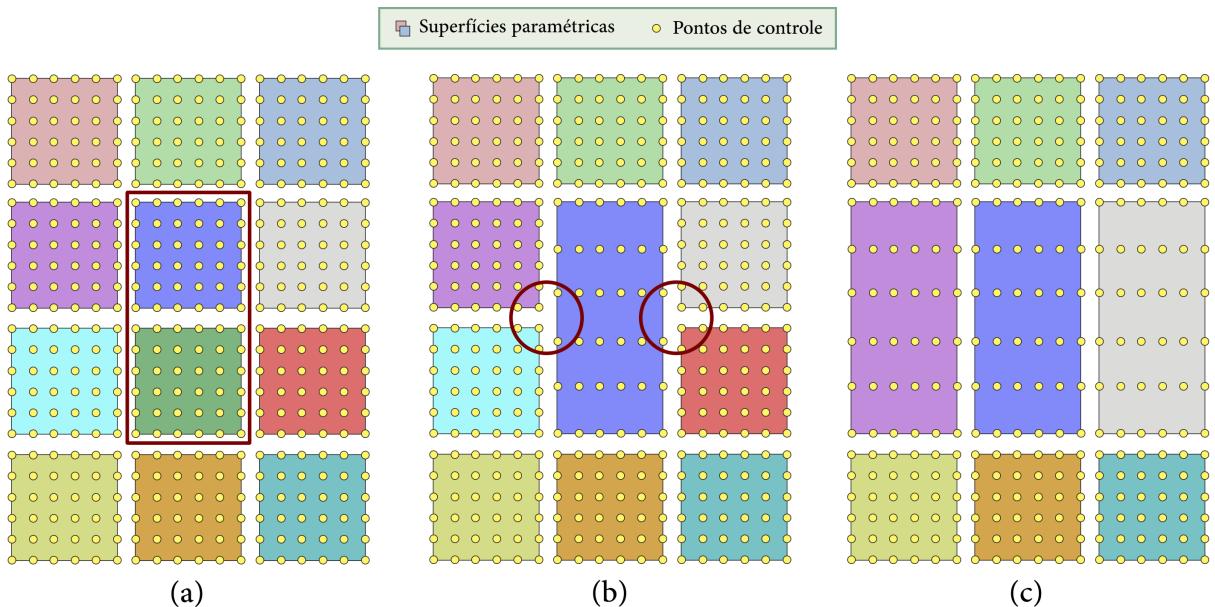
A escolha do valor δ impacta na qualidade e no tempo necessário para o cálculo da reparametrização. Ao utilizar valores pequenos para δ , garante-se um menor tempo de cálculo mas, em contrapartida, a discretização pode não ser suficiente para se encontrar melhores valores paramétricos. Por outro lado, ao utilizar maiores valores para δ , garante-se uma melhor discretização, aumentando as chances de identificar melhores associações de valores paramétricos, no entanto, demandam uma quantidade significativamente maior de tempo de cálculo. Valores próximos a 10 apresentaram um equilíbrio entre estes dois aspectos nos testes realizados.

4.2.7 Procedimento de Simplificação

A inclusão de um procedimento de redução ou simplificação de superfícies é outra alteração significativa entre o método de ajuste de múltiplas curvas e o método de ajuste por múltiplas superfícies. Este procedimento foi inserido como um substituto para as operações de mesclagem (*merge*), as quais apresentam um custo computacional considerável, mesmo no ajuste de múltiplas curvas. Um dos principais problemas na operação de mesclagem no ajuste de superfícies proposto é a impossibilidade de mesclar apenas dois retalhos de superfície vizinhos sem evitar junções T, quando há outras superfícies na orientação perpendicular à mesclagem. Neste caso, faz-se necessária a mesclagem de todas as superfícies que criariam uma junção T. Além de exigir a realização de mais operações, o que demanda mais tempo computacional, essa necessidade diminui a probabilidade de se obter mesclagens com um erro de ajuste inferior ou igual ao $\epsilon_{tol}^{[\gamma]}$. A Figura 22 ilustra o caso de uma operação de mesclagem entre dois retalhos de superfície. Na imagem da esquerda (a) são apresentadas as superfícies, destacando os retalhos que se deseja mesclar. Na imagem do meio (b) mostra-se a criação de junções T ao mesclar apenas os dois retalhos desejados. Por fim, a imagem da direita (c) apresenta a solução de mesclagem múltipla que evita a ocorrência de junções T.

O procedimento de simplificação consiste na tentativa de diminuição de pontos de controle dos retalhos de superfície. Neste caso, não há a exigência de que todas superfícies continuem com uma estrutura de pontos de controle compatível pois, o retalho de superfície com mais pontos de controle em uma determinada borda, poderia ser combinado com seu vizinho através de operações de inserção ou refinamento de nós. No entanto, como estas operações alteram o vetor de nós, pode ocorrer o mesmo caso já discutido na Subseção 4.2.5.1, que trata

Figura 22 – Demonstração de considerações sobre operações de mesclagem com superfícies



Fonte: Elaborado pelo autor (2022).

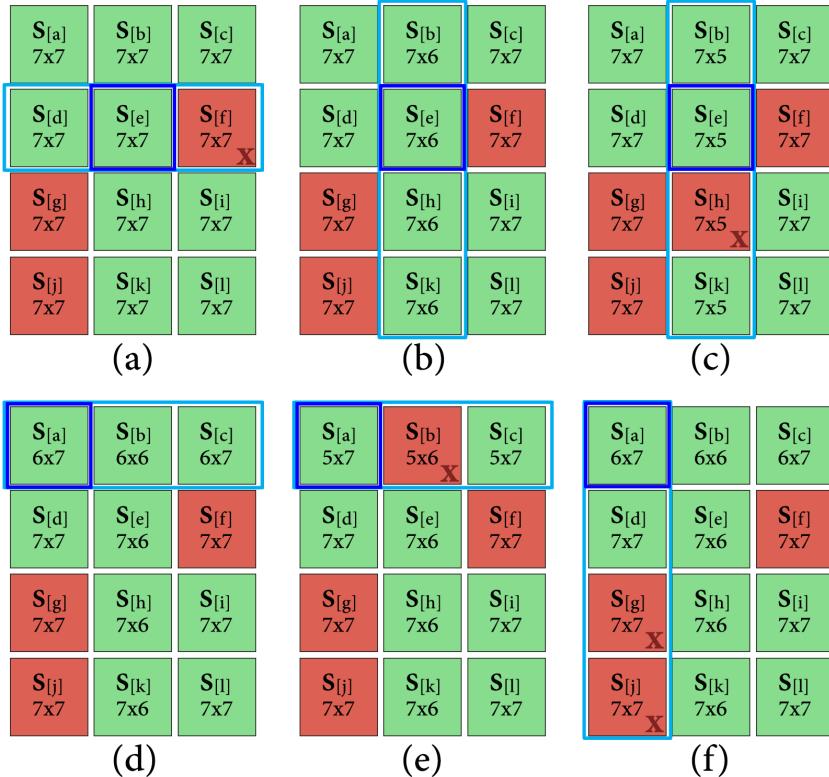
do procedimento de divisão: a necessidade de um vetor de nós diferente para cada borda das superfícies. Assim, com o intuito de simplificar conceitualmente este procedimento, optou-se por manter a estrutura de pontos de controle compatível. Isto implica na necessidade de reduzir a quantidade de pontos de controle igualmente para toda uma linha ou coluna de superfícies.

Com o objetivo de não se limitar apenas a superfícies com um poliedro de controle com mesmo número de linhas e colunas, adotou-se uma estratégia de tentar reduzir tais valores isoladamente. O algoritmo deste procedimento é parcialmente ilustrado na Figura 23 – na qual as cores de fundo indicam se o erro de ajuste é menor do que o limiar $\varepsilon_{tol}^{[\gamma]}$ (cor verde), ou maior do que ele (cor vermelha) – e funciona da seguinte forma. Escolhe-se iterativamente, até se esgotarem as opções, o retalho de superfície que possui o menor erro de ajuste, sendo este erro necessariamente menor do que $\varepsilon_{tol}^{[\gamma]}$ e que a superfície ainda não tenha sido analisada durante o procedimento. Note que a superfície com menor erro de ajuste é escolhida considerando que esta tem maior probabilidade de ser reduzida com sucesso.

Considere, por exemplo, o retalho $s_{[e]}$ – composto por uma matriz 7×7 de pontos de controle e identificado na Figura 23a – como o retalho escolhido para análise em uma dada iteração deste procedimento. O algoritmo tentará reduzir iterativamente a quantidade de linhas e depois colunas deste retalho e de sua vizinhança orientada removendo a seguinte quantidade de linhas ou colunas da matriz de controle, por iteração: $\max(1, \lceil \omega \cdot \psi \rceil)$. Note que o fator de redução ω evita a redução unitária de linhas e colunas no caso de superfícies com maior quantidade de pontos de controle, mesmo se utilizado com seu valor mínimo sugerido (0,1). A dita vizinhança orientada se refere a todos os retalhos que compartilham a mesma linha de retalhos ou coluna de retalhos, de acordo com o que se está reduzindo.

Por exemplo, ao reduzir a quantidade de colunas da matriz de controle, todas as superfícies

Figura 23 – Exemplo do processo de redução



Fonte: Elaborado pelo autor (2022).

que se encontram na mesma coluna de retalhos serão testadas para a redução – conforme ilustrado na Figura 23b. O teste de redução nada mais é do que uma nova operação de ajuste (conquista) com a quantidade reduzida de pontos de controle. Se o erro de ajuste do teste de redução de todas as superfícies ficar abaixo ou igual a $\varepsilon_{tol}^{[\gamma]}$, como ocorrido na Figura 23b, a redução é aceita e tenta-se aplicar uma nova redução. Por outro lado, caso ao menos um teste de redução resulte em um erro de ajuste superior a $\varepsilon_{tol}^{[\gamma]}$, a redução é descartada, conforme ilustrado na Figura 23c, caso em que as superfícies s_h resultou em erro superior a $\varepsilon_{tol}^{[\gamma]}$. Outra restrição imposta que interrompe a redução é a quantidade mínima de linhas e colunas para os retalhos de superfície que é determinado como 5 linhas e 5 colunas.

Os testes de redução ocorrem primeiramente em relação ao número de linhas e posteriormente ao número de colunas. Note que antes mesmo de tentar aplicar o teste de redução, caso alguma superfície da dita vizinha orientada já possua um erro de ajuste maior do que $\varepsilon_{tol}^{[\gamma]}$, o teste certamente fracassará e pode ser omitido – como ocorrido na Figura 23a e Figura 23f, por exemplo. Todos os retalhos de superfícies válidos são analisados pelo algoritmo que finaliza com um conjunto de retalhos possivelmente mais simples, isto é, com menos pontos de controle. No exemplo apresentado na Figura 23, as imagens da linha superior apresentam os testes de redução realizados para a superfície s_e e as imagens da linha inferior apresentam os testes de redução realizados para a superfície s_a . A menor redução obtida é ilustrada na Figura 23f.

4.2.8 Procedimento de Combinação

O procedimento de combinação adiciona um conceito chave para realizar a junção entre todas as bordas e quinas dos retalhos de superfície e obter um grau de continuidade geométrica: a utilização dos princípios das superfícies de Gregory, com emprego dos ditos pontos de Gregory. Isto significa que cada retalho de superfície B-Spline duplica seus pontos de controle diagonais às quinas – i.e. os pontos $P_D = \{P_{1,1}, P_{1,c-1}, P_{\ell-1,1} \text{ e } P_{\ell-1,c-1}\}$, em uma superfície com matriz de controle composta por $\ell + 1$ linhas e $c + 1$ colunas – sendo que um ponto fica atrelado à direção do valor paramétrico u e outro à direção do valor paramétrico v . Note que ao utilizar os mesmos valores geométricos dos pontos originais para as duplicatas, as superfícies continuam as mesmas.

Com todas as superfícies B-Spline com princípios de Gregory prontas, inicia-se um procedimento iterativo de combinação de bordas na qual cada retalho é processado por vez, sem uma ordem específica. Bertolino et al. (2021) utilizaram uma estratégia de mestre-escravo para determinar os pontos de controle – isto é, uma superfície é fixada e outra é ajustada de modo a garantir um grau de continuidade – mas, ao adotar tal estratégia, percebeu-se que dependendo da condição geométrica da superfície mestre, a superfície escrava pode resultar em junções com aspecto geométrico estranho, como torções consecutivas. No intuito de tentar evitar este tipo de problema, optou-se neste trabalho pela otimização conjunta dos pontos de controle de duas ou mais superfícies distintas, simultaneamente, através do método de mínimos quadrados. Nesta estratégia, os pontos de controle de quina e seus vizinhos adjacentes são inicialmente otimizados de modo a obter uma continuidade de quina C^1 nas orientações relacionadas a ambos os valores paramétricos u e v . Na sequência, estes pontos otimizados são fixados e todos os outros pontos de borda que impactam sobre a continuidade da junção são otimizados. Estas duas microetapas são detalhadas a seguir, mas antes disto, a fundamentação matemática e as notações utilizadas nas otimizações são explicadas.

Para apresentar a fundamentação matemática e os princípios básicos que serão utilizados a seguir, serão empregados alguns abusos de notação de modo a simplificar as equações matriciais. Considere uma superfície B-Spline $s_{[a]}$ e a seguinte equivalência entre a notação original do termo $B_{i,d_u}^{[a]}(u_k) \cdot B_{j,d_v}^{[a]}(v_k)$, que multiplica o ponto de controle $P_{i,j}^{[a]}$, e sua simplificação: $B_{i,j}^{[a]}(\tau_k^{[a]})$. Perceba a omissão dos graus d_u e d_v e a fusão das funções de mistura B e das variáveis paramétricas u_k e v_k . Considerando as definições apresentadas na Equação 27, a multiplicação da matriz $\mathbf{B}^{[a]}$ – composta por $\ell^{[a]} + 1$ linhas e $c^{[a]} + 1$ colunas – pelo vetor coluna $\mathbf{Q}^{[a]}$, pode ser simplificada como:

$$\mathbf{B}^{[a]} \cdot \mathbf{Q}^{[a]} = \begin{bmatrix} B_{0,0}^{[a]}(\tau_1^{[a]}) & B_{0,1}^{[a]}(\tau_1^{[a]}) & \dots & B_{\ell^{[a]}, c^{[a]}}^{[a]}(\tau_1^{[a]}) \\ B_{0,0}^{[a]}(\tau_2^{[a]}) & B_{0,1}^{[a]}(\tau_2^{[a]}) & \dots & B_{\ell^{[a]}, c^{[a]}}^{[a]}(\tau_2^{[a]}) \\ \vdots & \vdots & \vdots & \vdots \\ B_{0,0}^{[a]}(\tau_n^{[a]}) & B_{0,1}^{[a]}(\tau_n^{[a]}) & \dots & B_{\ell^{[a]}, c^{[a]}}^{[a]}(\tau_n^{[a]}) \end{bmatrix} \begin{bmatrix} P_{0,0}^{[a]} \\ P_{0,1}^{[a]} \\ \vdots \\ P_{\ell^{[a]}, c^{[a]}}^{[a]} \end{bmatrix} \quad (39)$$

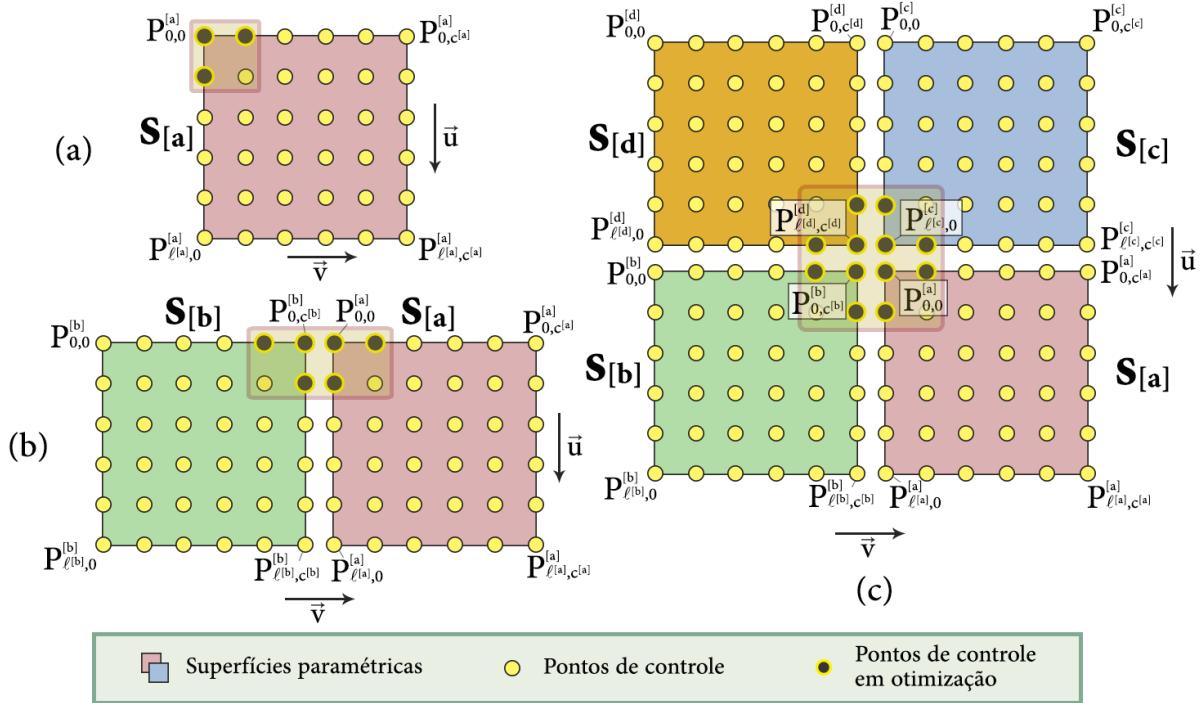
considerando que estes elementos estão atrelados à superfície $s_{[a]}$, a qual é ajustada sobre o

subconjunto $\mathbf{N}^{[a]} = [N_1^{[a]}, \dots, N_n^{[a]}]^\top$ da nuvem de pontos original. Por fim, considere também a notação para uma matriz nula com α linhas e β colunas, denotada por $\mathbf{Z}_{\alpha \cdot \beta}$.

4.2.8.1 Ajuste Múltiplo de Quinas

Na microetapa de ajuste múltiplo de quinas, cada quina é computada isoladamente. Como este ajuste considera as quinas adjacentes é necessário antecipar todas as situações possíveis. A Figura 24 ilustra os três possíveis casos de ajuste de quinas, identificando os pontos de controle otimizados e a relação de vizinhança entre as superfícies. No primeiro caso, o mais simples entre os três e representado na Figura 24a, a quina em questão não possui superfície vizinha, circunstância na qual os pontos de controle de quina podem ser mantidos como estão, já que já foram otimizados em relação ao subconjunto da nuvem de pontos ao qual estão relacionados.

Figura 24 – Possíveis casos de ajuste múltiplo de quinas



Fonte: Elaborado pelo autor (2022).

No segundo caso, a quina possui apenas uma relação de vizinhança com outra superfície, seja na orientação relacionada ao valor paramétrico u ou ao valor paramétrico v . Considerando uma superfície vizinha na orientação do valor paramétrico v , como ilustrado na Figura 24b, três pontos de controle de cada superfície (pontos destacados) são tomados como variáveis de otimização. Neste caso, empregando os abusos de notação mencionados anteriormente, pode-se otimizar o posicionamento destes pontos de controle – minimizando o erro quadrado de ajuste – através de mínimos quadrados, com o uso de uma vinculação de sistemas lineares e uso de

matrizes nulas:

$$\min_{Q^{[a]}, Q^{[b]}} \left(\begin{bmatrix} \mathbf{B}^{[a]} & \mathbf{Z}_{\ell^{[a]} \times c^{[b]}} \\ \mathbf{Z}_{\ell^{[b]} \times c^{[a]}} & \mathbf{B}^{[b]} \end{bmatrix} \begin{bmatrix} Q^{[a]} \\ Q^{[b]} \end{bmatrix} - \begin{bmatrix} N^{[a]} \\ N^{[b]} \end{bmatrix} \right) \quad (40)$$

Como uma continuidade C^1 é almejada, dois pontos de cada superfície são compartilhados e os outros dois pontos restantes estão atrelados pela relação de igualdade resultante da derivada que garante C^1 . Assim, restam apenas três pontos de controle como variáveis de otimização. Considerando o exemplo ilustrado na Figura 24b, e a Equação 22, têm-se as seguintes associatividades:

$$P_{0, c^{[b]}}^{[b]} = P_{0, 0}^{[a]} \quad P_{1, c^{[b]}}^{[b]} = P_{1, 0}^{[a]} \quad P_{0, c^{[b]}-1}^{[b]} = (1 + f_{[ab]}) \cdot P_{0, 0}^{[a]} - f_{[ab]} \cdot P_{0, 1}^{[a]}$$

sendo: $f_{[ab]} = \frac{d_v^{[a]}}{d_v^{[b]}} \cdot \frac{(1 - t_v^{[b]} (m_v - d_v - 2))}{t_v^{[a]} (d_v + 1)}$

As mesmas operações e igualdades podem ser aplicadas analogamente ao caso em que a combinação das superfícies ocorre na outra orientação paramétrica. Deve-se ressaltar ainda o caso especial da ocorrência de uma auto-relação, isto é, quando uma superfície possui relação de vizinhança consigo mesma. Por exemplo, quando a borda paramétrica da esquerda ($v = 0$) deve coincidir com a borda paramétrica da direita ($v = 1$). Neste caso, a Equação 41 pode ser simplificada, já que as funções de mistura e os pontos de controle se referem à mesma superfície.

Por fim, no terceiro caso, a quina possui relação de vizinhança com outras três superfícies: uma na orientação do valor paramétrico u , outra na orientação do valor paramétrico v e outra na diagonal paramétrica, conforme ilustrado na Figura 24c. Nesta circunstância, e considerando o exemplo ilustrado, três pontos de controle de cada superfície (pontos destacados) são tomados como variáveis de otimização novamente. Com os abusos de notação utilizados, pode-se otimizar o posicionamento destes pontos de controle, minimizando o erro quadrado de ajuste, com a mesma estratégia do caso anterior:

$$\min_{Q^{[a]}, Q^{[b]}, Q^{[c]}, Q^{[d]}} \left(\begin{bmatrix} \mathbf{B}^{[a]} & \mathbf{Z}_{\ell^{[a]} \times c^{[b]}} & \mathbf{Z}_{\ell^{[a]} \times c^{[c]}} & \mathbf{Z}_{\ell^{[a]} \times c^{[d]}} \\ \mathbf{Z}_{\ell^{[b]} \times c^{[a]}} & \mathbf{B}^{[b]} & \mathbf{Z}_{\ell^{[b]} \times c^{[c]}} & \mathbf{Z}_{\ell^{[b]} \times c^{[d]}} \\ \mathbf{Z}_{\ell^{[c]} \times c^{[a]}} & \mathbf{Z}_{\ell^{[c]} \times c^{[b]}} & \mathbf{B}^{[c]} & \mathbf{Z}_{\ell^{[c]} \times c^{[d]}} \\ \mathbf{Z}_{\ell^{[d]} \times c^{[a]}} & \mathbf{Z}_{\ell^{[d]} \times c^{[b]}} & \mathbf{Z}_{\ell^{[d]} \times c^{[c]}} & \mathbf{B}^{[d]} \end{bmatrix} \begin{bmatrix} Q^{[a]} \\ Q^{[b]} \\ Q^{[c]} \\ Q^{[d]} \end{bmatrix} - \begin{bmatrix} N^{[a]} \\ N^{[b]} \\ N^{[c]} \\ N^{[d]} \end{bmatrix} \right) \quad (41)$$

Mais uma vez, como uma continuidade C^1 é desejada, alguns pontos de cada superfície são compartilhados, ao passo que outros estão atrelados pela relação de igualdade resultante da derivada direcional, sob a orientação da combinação, que garante C^1 . Deste modo, apenas três pontos de controle são tomados como variáveis de otimização. Considerando o exemplo ilustrado

na Figura 24c, têm-se as seguintes relações:

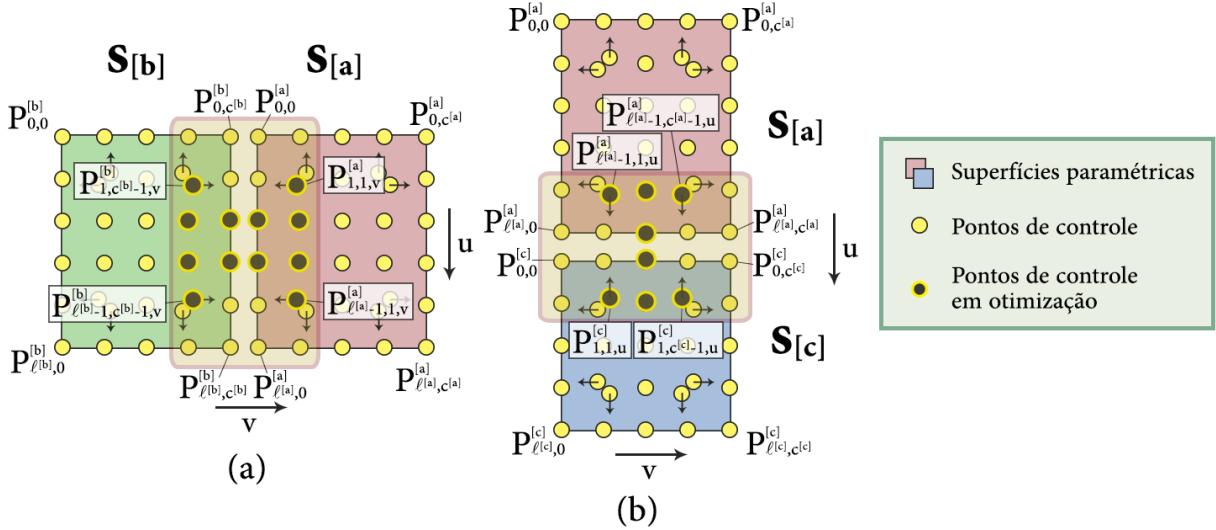
$$\begin{aligned}
 P_{\ell^{[d]}, c^{[d]}}^{[d]} &= P_{\ell^{[c]}, 0}^{[c]} = P_{0, c^{[b]}}^{[b]} = P_{0, 0}^{[a]} \\
 P_{1, c^{[b]}}^{[b]} &= P_{1, 0}^{[a]} \\
 P_{\ell^{[c]}, 1}^{[c]} &= P_{0, 1}^{[a]} \\
 P_{\ell^{[d]}, c^{[d]-1}}^{[d]} &= P_{0, c^{[b]-1}}^{[b]} = (1 + f_{[ab]}) \cdot P_{0, 0}^{[a]} - f_{[ab]} \cdot P_{0, 1}^{[a]} \\
 P_{\ell^{[d]-1}, c^{[d]}}^{[d]} &= P_{\ell^{[c]-1}, 0}^{[c]} = (1 + f_{[ac]}) \cdot P_{0, 0}^{[a]} - f_{[ac]} \cdot P_{1, 0}^{[a]} \\
 \text{sendo: } f_{[ab]} &= \frac{d_v^{[a]}}{d_v^{[b]}} \cdot \frac{(1 - t_v^{[b]}(m_v - d_v - 2))}{t_v^{[a]}(d_v + 1)} \quad \text{e} \quad f_{[ac]} = \frac{d_u^{[a]}}{d_u^{[c]}} \cdot \frac{(1 - t_u^{[c]}(m_u - d_u - 2))}{t_u^{[a]}(d_u + 1)}
 \end{aligned}$$

4.2.8.2 Ajuste Múltiplo de Bordas

A microetapa de ajuste múltiplo de bordas opera com um princípio similar à microetapa anterior. Cada borda do retalho de superfície sendo analisado é ajustada individualmente com a borda do retalho vizinho com o qual deverá ser combinada com continuidade. Para tanto, todas as variáveis que possuem relação com a junção destas duas superfícies com continuidade C^1 são utilizadas como variáveis de otimização, com exceção dos pontos de controle de quina e seus pontos de controle adjacentes, os quais já se encontram ajustados com continuidade. Os demais pontos são tomados como constantes para a otimização que será realizada. A Figura 25 ilustra dois casos de junção de borda entre superfícies, uma em cada orientação paramétrica, destacando os pontos de controle que serão utilizados como variáveis com cores diferentes para cada ajuste de borda. Note o destaque para os pontos de controle duais utilizados nas diagonais das quinas, em conformidade com os princípios de Gregory. É essa dualidade de pontos de controle que viabiliza a otimização individual de cada borda dos retalhos.

Neste caso, todas as situações de ajuste de bordas agregam apenas duas superfícies em um problema de mínimos quadrados com resolução de sistemas lineares. Considere a combinação das bordas adjacentes das superfícies $s_{[a]}$ e $s_{[b]}$, e das superfícies $s_{[a]}$ e $s_{[c]}$, ilustradas na Figura 25, assim como o mesmo abuso de notação já utilizado. A minimização empregada pode ser formalizada exatamente como na Equação 40, mas a expressão matricial terá de ser manipulada diferentemente para isolar as variáveis das constantes neste caso. Além disso, as igualdades entre os pontos de controle que são tomados como variáveis de otimização são diferentes. Com base na Equação 18, na Equação 19 e considerando que as superfícies possuem

Figura 25 – Ilustração de casos de ajustes múltiplos de bordas



Fonte: Elaborado pelo autor (2022).

a mesma quantidade de pontos de controle nas bordas, pode-se utilizar as seguintes igualdades:

$$P_{i, c^{[b]}}^{[b]} = P_{i, 0}^{[a]} \quad \text{para: } 2 \leq i < \ell^{[a]-1}$$

$$P_{\ell^{[c]}, j}^{[c]} = P_{0, j}^{[a]} \quad \text{para: } 2 \leq j < c^{[a]-1}$$

$$P_{i, c^{[b]}-1, v}^{[b]} = (1 + f_{[ab]}) \cdot P_{i, 0, v}^{[a]} - f_{[ab]} \cdot P_{i, 1, v}^{[a]}$$

$$P_{1, j, u}^{[c]} = (1 + f_{[ac]}) \cdot P_{\ell^{[a]}, j, u}^{[a]} - f_{[ac]} \cdot P_{\ell^{[a]}-1, j, u}^{[a]}$$

$$\text{onde: } \begin{cases} 1 \leq i < \ell^{[a]} & 1 \leq j < c^{[a]} \\ f_{[ab]} = \frac{d_v^{[a]}}{d_v^{[b]}} \cdot \frac{(1 - t_v^{[b]}(m_v - d_v - 2))}{t_v^{[a]}(d_v + 1)} & f_{[ac]} = \frac{d_u^{[a]}}{d_u^{[c]}} \cdot \frac{t_u^{[c]}(d_u + 1)}{(1 - t_u^{[a]}(m_u - d_u - 2))} \end{cases}$$

Perceba que foi utilizada a notação de pontos de controle orientados nas equações apresentadas.

Esta notação foi utilizada porque alguns dos pontos de controle envolvidos são pontos de Gregory:

$P_{1, 1, v}^{[a]}, P_{\ell^{[a]}-1, 1, v}^{[a]}, P_{\ell^{[a]}-1, 1, u}^{[a]}, P_{\ell^{[a]}-1, c^{[a]}-1, u}^{[a]}, P_{1, c^{[b]}-1, v}^{[b]}, P_{\ell^{[b]}-1, c^{[b]}-1, v}^{[b]}, P_{1, 1, u}^{[c]} \text{ e } P_{1, c^{[c]}-1, u}^{[c]}$. Além disso, note que as igualdades apresentadas incluem pontos de controle que são utilizados como constante na otimização. No caso ilustrado, omitindo os símbolos de orientação que não possuem influência neste caso, são eles: $P_{1, 0}^{[a]}, P_{\ell^{[a]}-1, 0}^{[a]}, P_{1, c^{[b]}}^{[b]}, P_{\ell^{[b]}-1, c^{[b]}}^{[b]}, P_{0, 1}^{[c]} \text{ e } P_{0, c^{[c]}-1}^{[c]}$.

Assim como realizado na proposta de ajuste de múltiplas curvas, também se idealizou a utilização de um seletor λ que viabilizaria a seleção adaptativa de continuidade em cada borda de junção entre os retalhos de superfície. Todavia, tal mecanismo não foi utilizado para este trabalho pois, nos diversos experimentos realizados, todas as junções tiveram a continuidade C^1 escolhida, sem exceção. Mais do que isso, o tempo de cálculo para esta escolha adaptativa se mostrou significativo. Vale salientar que não se está afirmando que tal mecanismo não funcione, mas que não se encontrou um critério de seleção adequado para determinar quando uma combinação

com continuidade C^0 seria mais adequada do que uma combinação com C^1 . Ou ainda, pode-se cogitar que os casos de teste utilizados nos experimentos e as respectivas linhas de divisão não contribuíram para a validação adequada deste mecanismo.

4.2.9 Mecanismos de Aceleração

Assim como no trabalho de Kineri et al. (2012), mecanismos de aceleração foram propostos para o método desenvolvido. Ao todo, foram adicionados quatro mecanismos: explosão da nuvem de pontos, programação dinâmica no cálculo das funções de mistura, cálculo de erro de ajuste com limite de erro e redução dos sistemas lineares em junções de superfície. Estes mecanismos são opcionais, mas visam o melhoramento da eficiência do método proposto.

4.2.9.1 Explosão da Nuvem de Pontos

Este procedimento tem como intuito acelerar as primeiras etapas do ciclo iterativo de conquista e divisão. Para nuvens de pontos com muitas amostras, mesmo o ajuste por uma superfície com poucos pontos de controle pode demandar um tempo significativo. Além disso, é remota a chance de que uma nuvem de pontos com muitas amostras seja reconstruída dentre o erro de ajuste definido, uma vez que o valor de ψ sugerido é relativamente pequeno. Tendo isto em vista, a explosão da nuvem de pontos realiza iterativamente a divisão da nuvem de pontos em pedaços menores, sem realizar o ajuste de superfície. Assim, pode-se entender este procedimento como uma operação de divisão, sem o procedimento de conquista subsequente.

As explosões – ou divisões – são realizadas com base na quantidade de amostras $n^{[i]}$ do subconjunto da nuvem de pontos, sendo divididos apenas os subconjuntos nos quais a condição $n^{[i]} > \zeta$ é satisfeita, sendo ζ a quantidade máxima de amostras para a realização de uma operação de ajuste (ver Tabela 5). Além disso, são consideradas todas as demais restrições empregadas no procedimento de divisão apresentado na Subseção 4.2.5, com exceção daquelas envolvendo o erro de ajuste, uma vez que nenhum ajuste foi realizado ainda. As operações de divisão empregadas aqui também utilizam o mesmo algoritmo de particionamento χ , gerando uma reta de corte no espaço paramétrico, vertical ou horizontal, sendo realizado um único particionamento por iteração – o qual pode envolver um ou múltiplos subconjuntos da nuvem de pontos.

O valor de ζ pode ser adaptado de acordo com a situação de ajuste. Por exemplo, no caso do ajuste de uma nuvem de pontos que representa uma superfície plana, com uma grande quantidade de pontos, pode-se utilizar $\zeta = \infty$, fazendo com que a etapa de explosão da nuvem de pontos seja ignorada. Desde modo, pode-se possibilitar a tentativa de ajuste de uma superfície com poucos pontos de controle sobre uma nuvem de pontos geometricamente simples, mas que possui muitas amostras. Para os casos genéricos, sugere-se a utilização de $\zeta = \psi^{\frac{7}{2}}$, valor o qual se mostrou satisfatório nos testes empíricos realizados.

4.2.9.2 Programação Dinâmica

Considerando que as superfícies iniciais compartilham as mesmas características – como os graus da superfície, a quantidade de pontos de controle e o vetor de nós – e que o ciclo inicial de conquista e divisão requer o cálculo repetitivo das funções de base, implementou-se um mecanismo de programação dinâmica para este aspecto. Toda vez que uma função de base é calculada, seus parâmetros e resultado são armazenados em uma estrutura de dados de rápida consulta e inserção. Deste modo, sempre que a função base for usada com os mesmos parâmetros, o resultado pode ser obtido sem necessidade de recalculá-la. Através deste mecanismo, realiza-se uma troca entre o consumo adicional de memória por um melhoramento no tempo de execução.

4.2.9.3 Cálculo de Erro de Ajuste com Limite de Erro

O cálculo de erro de ajuste é um procedimento de custo temporal significativo. No entanto, em determinadas situações não é necessário descobrir o erro exato de ajuste, mas apenas se este valor excede um limite pré-estabelecido $\epsilon_{tol}^{[\gamma]}$. Nestes casos, pode-se reduzir o tempo de cálculo interrompendo o procedimento no momento em que é detectado um erro parcial (cumulativo ou não) maior do que o limite imposto. No caso de $\gamma = \text{ME}$, basta comparar o valor individual ϵ_i de cada erro de ajuste e suspender o procedimento caso $\epsilon_i > \epsilon_{tol}^{\text{ME}}$. Já para o caso de $\gamma = \text{RMSE}$, deve-se comparar o valor de erro acumulativo ϵ_{acu} com um limiar pré-processado $\epsilon_{pp} = (\epsilon_{tol}^{\text{RMSE}})^2 \cdot n$, onde n representa o tamanho da nuvem de pontos, e interromper o procedimento caso $\epsilon_{acu} > \epsilon_{pp}$.

No caso do método de ajuste de múltiplas curvas, este mecanismo foi implantado em quase todas os processos de cálculo de erro de ajuste, com exceção da reparametrização e o erro final de ajuste. Para o método de ajuste por múltiplas superfícies, no entanto, se faz necessário conhecer o erro de ajuste exato em alguns procedimentos. Por exemplo, na etapa iterativa de conquista e divisão, é necessário saber o valor exato do erro de ajuste a fim de determinar o ponto e a orientação da reta de partição. O uso deste mecanismo de aceleração foi testado nesta etapa iterativa e resultou em ajustes piores devido à más escolhas do particionamento automático. Já para outras etapas, como no procedimento de redução, não há necessidade de conhecer o erro exato. Sendo assim, neste caso, este mecanismo de aceleração foi utilizado.

4.2.9.4 Redução de Sistemas Lineares em Junções

O último mecanismo de aceleração está atrelado ao procedimento de combinação das superfícies e à utilização de sistemas lineares associados. Ao utilizar a abordagem de mínimos quadrados para o ajuste das quinas e das bordas, com apenas alguns pontos de controle como variáveis de otimização no problema, pode-se limitar os sistemas lineares. Essa limitação pode levar em conta as regiões de influência desses pontos de controle, removendo do sistema linear todos os termos que estão envolvidos com valores paramétricos que não são influenciados pelos pontos de controle sendo otimizados. Por exemplo, ao otimizar o ponto de controle $P_{1,0}$ em

uma das bordas ($v = 0$), nenhuma influência deste ponto será exercida sobre a borda oposta no domínio paramétrico ($v = 1$).

Estabeleceu-se, por simplificação, o uso de apenas um quarto dos valores paramétricos no caso das otimizações de quina das superfícies. Este intervalo de valores paramétricos a serem utilizados se refere à área de maior influência dos pontos de controle da quina sob otimização. Para a otimização das bordas das superfícies, estabeleceu-se o uso de no mínimo metade dos valores paramétricos de maior influência, ou o uso de todos os valores paramétricos influenciados pelos pontos de controle em otimização.

4.3 CONSIDERAÇÕES

Neste capítulo foi apresentado o novo método de ajuste por múltiplas superfícies: DCBA. Iniciou-se apresentando a justificativa para a elaboração deste método, um resumo do método precursor de ajuste de múltiplas curvas e finalmente a visão geral e o detalhamento das operações que constituem o método proposto. Os procedimentos e mecanismos utilizados incluem: um ciclo iterativo de conquista (ajuste) e divisão de nuvens de pontos, um procedimento de reparametrização e simplificação das superfícies, um procedimento de combinação e mecanismos de aceleração. O objetivo de realizar ajustes sem a necessidade de determinar uma quantidade de pontos de controle de forma manual foi alcançada. Mais do que isso, buscou-se fornecer uma interface simplificada de uso, não exigindo a definição de parâmetros de funcionamento do método, mas viabilizando o ajuste dos valores padrões utilizados para fins de refinamento.

Com a opção da utilização de superfícies paramétricas B-Splines, algumas restrições de estrutura das matrizes de controle foram respeitadas a fim de evitar junções de tipo T. A utilização de T-Splines poderia viabilizar a combinação de superfícies com este tipo de junção, no entanto, uma estrutura irregular de nós não permitiria um ajuste trivial com múltiplas superfícies que evite o problema de compatibilidade de torção. Ao menos não foi encontrado algum trabalho na literatura que mencionasse tal tipo de solução, tão pouco a utilização de princípios de Gregory em superfícies T-Splines.

A separação de funções e operações é um princípio que deve ser ressaltado na proposta deste método. Esta separação permite que operações específicas sejam aprimoradas ou testadas sem a necessidade de alterar todo o funcionamento do método, desde que utilizem os mesmos princípios fundamentais utilizados como a utilização de superfícies paramétricas retangulares e não existência de junções do tipo T.

O tempo de execução dos procedimentos varia significativamente dependendo do estado intermediário das divisões e ajustes realizados. Por exemplo, o procedimento de simplificação pode demandar um tempo desprezível quando os ajustes já possuem um erro maior ou muito próximo ao limiar de tolerância definido. Por outro lado, quando os erros de ajuste estão significativamente menores do que o limiar de tolerância, uma maior quantidade de operações de redução é realizada, implicando em um maior tempo de execução. De modo geral, entre os

procedimentos, as etapas iterativas de conquista e divisão e o procedimento de combinação são os mais custosos. Todavia, o mecanismo de aceleração opcional de explosão da nuvem de pontos reduz significativamente a quantidade de ajustes realizada pelo processo iterativo de conquista e divisão, em especial as primeiras iterações que realizam o ajuste de uma maior quantidade de amostras e são mais custosos. Além disso, a divisão compulsória das nuvens de pontos com grande quantidade de amostras, contribuiu em diversos casos para um melhor ajuste final.

Com base na utilização de superfícies B-Spline bicúbicas, as quais já fornecem continuidade C^2 , o processo de combinação poderia utilizar as mesmas técnicas empregadas para o ajuste com C^2 entre os retalhos vizinhos. A decisão por se limitar a C^1 foi baseada na revisão de literatura, sendo usual a utilização de junções com apenas C^1 – tanto em curvas quanto em superfícies – e também pela maior simplicidade das equações envolvidas. Considerou-se que, neste primeiro momento, a validação das combinações por C^1 seriam motivo para justificar a tentativa posterior da utilização de continuidade C^2 entre as junções de superfícies.

Note que o método proposto é capaz de ajustar superfícies tubulares, i.e. fechadas em relação à um dos valores paramétricos. O método pode receber tal tipo de indicação como parâmetro de entrada adicional e produzir uma autorelação de vizinhança entre as bordas na orientação relacionada ao valor paramétrico u (borda superior e inferior coincidem) ou na orientação relacionada ao valor paramétrico v (borda esquerda e direita coincidem) para a superfície inicial. De modo semelhante, uma superfície totalmente fechada também pode ser ajustada indicando autorelação em ambas as orientações.

Deve-se enfatizar que, apesar do método proposto utilizar a estratégia de divisão e conquista, e fazer uso de um requadramento no espaço paramétrico para cada subconjunto da nuvem de pontos, ele ainda exige uma parametrização global inicial. Em outras palavras, a nuvem de pontos inicial – completa – precisa ser parametrizada e entregue como entrada ao algoritmo. Todavia, o procedimento de parametrizar uma nuvem de pontos com uma grande quantidade de amostras pode requerer um tempo significativo de cálculo e a qualidade do resultado desta operação possui forte influência sobre o resultado obtido no ajuste de superfícies. Além disso, a exigência de uma parametrização global acaba impedindo que o método explore seu potencial de realizar o ajuste de superfície com topologias não triviais, que não podem ser planificadas em um círculo ou quadrado.

5 EXPERIMENTOS E RESULTADOS

O método proposto de ajuste por múltiplas superfícies foi implementado na linguagem de programação Python¹, utilizando as bibliotecas: numpy², scipy³, matplotlib⁴ e KdQuery⁵. Como não foram encontradas bibliotecas ou similares que viabilizassem a realização de parametrização de malhas triangulares com os principais métodos da literatura em Python, implementou-se também um módulo de parametrização na linguagem de programação C++ com auxílio da biblioteca CGAL⁶. Todos os experimentos foram executados em um computador com um processador AMD FX(tm)-8350 de oito núcleos (4.00GHz) e 16GB de memória RAM.

Em termos de base do funcionamento do método proposto, existe uma função principal que é utilizada nas operações. Trata-se da função `lstsq`, do módulo `linalg`, da biblioteca Numpy: `np.linalg.lstsq`. Esta função é uma rotina de álgebra linear que calcula o vetor P que resolve aproximadamente ou exatamente a equação $A \times P = B$; em outras palavras, calcula a solução de mínimos quadrados para uma equação matricial linear. Esta função é empregada em todos os ajuste de superfícies no qual o vetor de pontos de controle é otimizado. Outros dois métodos que foram utilizados da biblioteca CGAL e merecem destaque são o CGAL::Polygon_mesh_processing::longest_border e o SMP::parameterize que foram utilizados, junto com outros métodos da mesma biblioteca, para calcular uma parametrização em nuvens não organizadas. O primeiro método calcula a maior borda em uma malha triangular, ao passo que o segundo realiza efetivamente a operação de parametrização dados diversos parâmetros, incluindo a maior borda – deste aspecto surge a restrição de testes imposta neste trabalho para superfícies homeomórficas a um disco ou que podem ser transformadas nesta topologia através do corte da superfície com uso de costuras (*Seams*). Essas costuras, ou *Seams*, demarcam arestas compartilhadas por duas faces que podem ser utilizadas como bordas para a parametrização.

Como o objetivo deste trabalho não é explorar amplamente a possibilidade de refinamento de resultados com o ajuste de parâmetros, foram utilizados os parâmetros padrões definidos na Tabela 5. Como exceção, apenas o parâmetro χ foi testado amplamente com os seus três possíveis valores sugeridos a fim de identificar as características dos métodos de partição idealizados. Além da análise do aspecto visual, os resultados foram analisados em relação ao tempo de computação, identificando o custo de processamento nas diferentes etapas do método, o erro de ajuste obtido em relação às métricas RMSE e ME, juntamente de um comparativo com as métricas AIC e BIC, e uma análise da variação na quantidade de pontos de controle otimizadas durante o processo.

A maior dificuldade encontrada na comparação de métodos de ajuste de superfície em relação a outros métodos é justamente a inviabilidade de realizá-las com o devido rigor científico.

¹ Python v3.8.3 (<<https://www.python.org/>>)

² NumPy v1.22.3 (<<https://numpy.org/>>)

³ SciPy v1.8.0 (<<https://scipy.org/>>)

⁴ MatPlotLib v3.5.1 (<<https://matplotlib.org/>>)

⁵ KdQuery 0.2.2 (<<https://pypi.org/project/KdQuery/>>)

⁶ The Computational Geometry Algorithms Library – CGAL (<<https://www.cgal.org/>>)

Por um lado, a maioria dos trabalhos analisados durante a revisão de literatura, ou utilizam nuvens de pontos e malhas triangulares próprias, sem identificar ou disponibilizar uma fonte para obtê-las, ou realizam algum pré-processamento nos pontos, como a simulação de ruído ou escolha randômica de parte da nuvem de pontos. Isto inviabiliza tanto a reproduzibilidade destes métodos como a possibilidade de comparar resultados.

Os métodos identificados na literatura não foram encontrados em bibliotecas ou repositórios públicos, e sua reimplementação não é simples, pois nem sempre a apresentação dos métodos é clara em relação a todos os detalhes de funcionamento. Em face deste problema, e de modo similar ao que foi realizado por (IGLESIAS et al., 2018), realizou-se uma comparação com o método tradicional de ajuste, isto é, o ajuste de uma única superfície B-Spline bicúbica. No entanto, esta comparação não foi realizada para todos os experimentos, pois o método tradicional exige a definição da quantidade de pontos de controle ao passo que o método proposto calcula adaptativamente este valor. Assim, esta comparação foi realizada apenas no final deste capítulo, ao avaliar a eficiência do método proposto com nuvens de grande quantidade de amostras que necessitam uma grande quantidade de pontos de controle para o ajuste.

Para a realização dos testes foram utilizadas nuvens de pontos organizadas e suas respectivas malhas triangulares, as quais foram calculadas com base em equações explícitas e paramétricas encontradas na literatura e outras fontes. Foram utilizadas também nuvens de pontos e malhas triangulares provenientes de repositórios externos com o intuito de testar determinadas características. Para os *benchmarks* de nuvens de pontos organizadas se utilizou o método de parametrização comprimento de corda implementado em Python, ao passo que para os *benchmarks* de nuvens de pontos não organizadas, utilizou-se o método de parametrização equiareal (*authalic*) implementado em C++. Como a nuvem de pontos e a parametrização são requisitadas como entrada – assim como feito por Zhang et al. (2016) – o tempo de computação da parametrização inicial é desconsiderado no tempo de execução do método proposto.

5.1 EXPERIMENTOS COM NUVENS ORGANIZADAS

Nesta seção são apresentados os experimentos realizados com *benchmarks* por equações paramétricas e explícitas, as quais podem ser utilizadas para produzir nuvens de pontos organizadas e uma malha triangular que define a relação de vizinhança e organização entre os pontos. Diversos trabalhos utilizam nuvens de pontos próprias ou de fontes não divulgadas, o que dificulta a realização de experimentos semelhantes. Os trabalhos de Lee (2002), Gálvez, Iglesias e Puig-Pey (2012) e Mao et al. (2018) foram os únicos encontrados na literatura os quais utilizaram *benchmarks* identificados e definidos na apresentação dos experimentos. Assim, buscou-se produzir nuvens de pontos semelhantes as que foram reportadas nestes trabalhos. Vale ressaltar que as métricas de erro obtidas não foram comparadas com estes trabalhos uma vez que não foi identificado claramente se as nuvens de pontos foram normalizadas, o que acaba alterando as medidas de distância e por consequência as métricas de erro.

Para avaliar os resultados são apresentadas imagens das superfícies geradas para uma validação qualitativa, possibilitando a visualização da aproximação geométrica destas e da distribuição dos retalhos de superfícies, os quais foram renderizados com cores distintas para melhor distingui-los. Para cada experimento realizado é apresentada uma figura padronizada, a qual é composta por quatro imagens: (a) a nuvem de pontos, (b) o resultado obtido com $\chi=\text{PBS}$, (c) o resultado obtido com $\chi=\text{PMaC}$ e, (d) o resultado obtido com $\chi=\text{PMeC}$. De modo semelhante, também são apresentadas figuras com uma imagem de mapa de calor – representando o erro de ajuste – para cada resultado obtido pelas métodos de particionamento testados. Ressalta-se antecipadamente a qualidade das junções com continuidade C^1 obtidas com o método proposto, de forma totalmente automatizada.

São apresentados também dados que permitem uma análise quantitativa dos resultados, juntamente com uma discussão sobre os valores obtidos. Destaca-se em um primeiro conjunto o tempo de execução utilizado para cada etapa do método proposto, com as três abordagens de particionamento. Foram realizados dois experimentos para cada tipo de particionamento: um teste sem a utilização dos mecanismos de aceleração por explosão da nuvem de pontos e por redução de sistemas lineares em junções; e outro teste com todos os mecanismos de aceleração apresentados. Para estes dados, foram elencados os tempos necessários para a realização da explosão da nuvem de pontos, do procedimento iterativo de conquista e divisão, a reparametrização, a simplificação e a combinação. Além destes, foi incluído também o tempo utilizado para outras operações que são necessárias para o funcionamento do método, como inicialização de variáveis e cálculo dos parâmetros padrões. Ressalta-se que não estão incluídas nestes valores as operações de parametrização, cálculo da MMD (Média das Menores Distâncias entre cada amostra da nuvem e seus vizinhos) e cálculo dos índices de curvatura para cada amostra da nuvem de pontos.

Em um segundo grupo de dados é destacada a variação dos erros de ajuste (RMSE e ME) durante quatro etapas do algoritmo: após o término do procedimento iterativo de conquista e divisão; após a operação de reparametrização; após a simplificação das superfícies; e o resultado final obtido. Essas métricas foram separadas em dois gráficos para cada experimento, visto que na maioria dos casos a sua ordem de grandeza é diferente e prejudica a visualização conjunta dos dados. Por fim, um terceiro grupo de dados também é apresentado, indicando o grau de redução das variáveis de representação dos retalhos de superfície através do procedimento de simplificação, evidenciando a quantidade de pontos de controle antes e após este procedimento. Vale enfatizar que os dados destes dois últimos grupos foram obtidos com a utilização dos mecanismos de aceleração.

5.1.1 Benchmark Shell

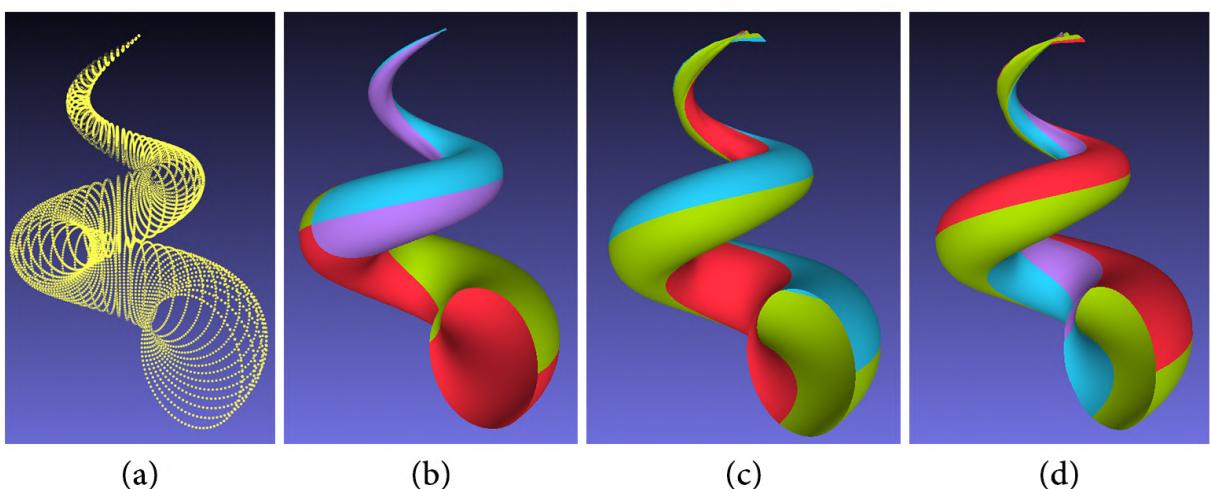
O *benchmark* de superfície *Shell* é definido pela seguinte equação paramétrica (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012; MAO et al., 2018):

$$\begin{aligned}x(u, v) &= \frac{1}{5} \left(1 - \left(\frac{v}{2\pi}\right)\right) \cdot \cos(2v) \cdot (1 + \cos(u)) + \frac{1}{10} \cos(2v) \\y(u, v) &= \frac{1}{5} \left(1 - \left(\frac{v}{2\pi}\right)\right) \sin(2v) \cdot (1 + \cos(u)) + \frac{1}{10} \sin(2v) \\z(u, v) &= \left(\frac{v}{2\pi}\right) + \frac{1}{5} \cdot \left(1 - \frac{v}{2\pi}\right) \cdot \sin(u) \\u &\in [0; 2\pi] \quad \text{e} \quad v \in [0; 2\pi]\end{aligned}$$

Tentando se aproximar das 7.500 amostras utilizadas por Gálvez, Iglesias e Puig-Pey (2012), foram geradas 87 amostras uniformemente distribuídas para u e v , totalizando 7.569 pontos (87^2). A Figura 26a ilustra a nuvem de pontos gerada. Mao et al. (2018) também utilizaram este *benchmark* em seus experimentos, gerando 2.475 pontos uniformemente distribuídos e selecionando aleatoriamente 2.102 amostras.

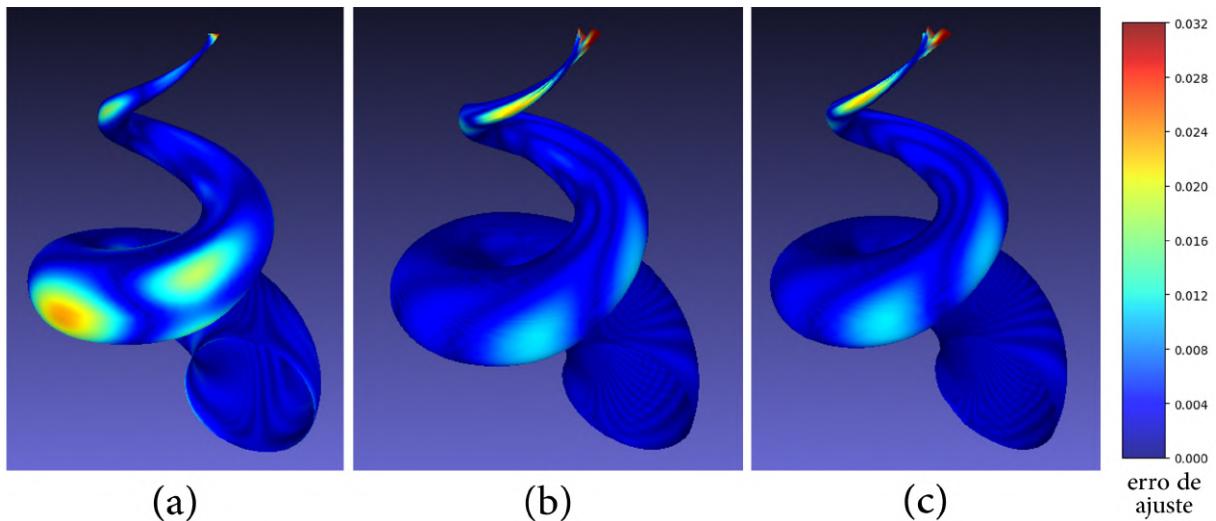
Como a superfície representada por esta equação é fechada (tubular) em relação aos valores paramétricos de v – como uma superfície de revolução – foi adicionado este indicador ao método de ajuste, para que as bordas relacionadas a esta orientação paramétricas sejam conectadas. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros (ver Tabela 5), foram utilizados os seguintes valores: $\varepsilon_{tol}^{[\gamma]} = 0,032$, $\psi = 10$ e $\zeta = 3.162$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 26, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 27 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC.

Figura 26 – Nuvem de pontos e resultados obtidos para o ajuste sobre o *benchmark Shell*



Fonte: Elaborado pelo autor (2022).

Figura 27 – Mapas de calor dos resultados obtidos para o ajuste sobre o *benchmark Shell*



Fonte: Elaborado pelo autor (2022).

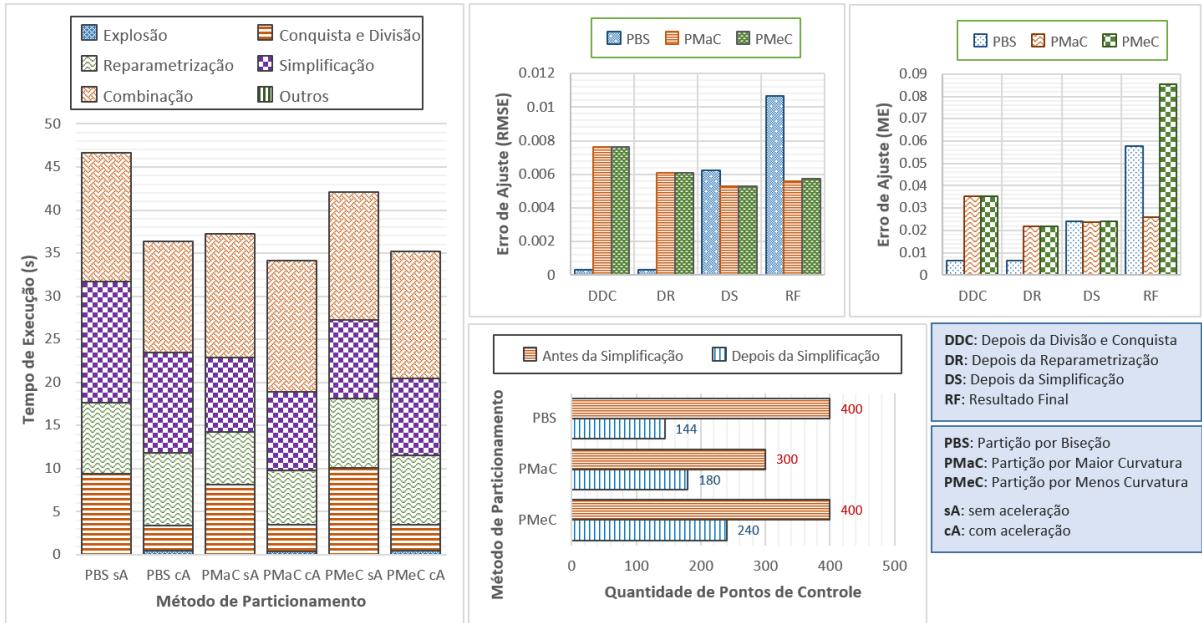
Pode-se perceber visualmente que os resultados obtidos pelos diferentes métodos de particionamento são similares, mas apresentam diferenças nas linhas de cortes utilizadas. O método PBS realizou cortes em ambas as orientações, ao passo que os outros dois métodos baseados em curvatura realizaram cortes apenas em uma orientação paramétrica. Esta diferença de particionamento fez com que estes dois últimos métodos não conseguissem representar de forma satisfatória a ponta superior aguda do *benchmark*, pois as poucas amostras desta região foram divididas entre os quatro retalhos e sua influência se tornou menos significativa no método de mínimos quadrados.

Para avaliar os dados quantitativos obtidos através da execução do método DCBA, apresenta-se a Figura 28. Através do gráfico de tempo de execução, pode-se perceber que o método de partição PBS foi o mais lento e o método PMaC foi o mais rápido dentre os três. No entanto, essa diferença foi pouco significativa quando os mecanismos de aceleração foram utilizados. Como comparativo, o DCBA não exigiu mais do que 50 segundos em qualquer experimento para realizar o ajuste dos 7.569 pontos, ao passo que o método de Mao et al. (2018) necessitou de pouco mais de 2 minutos para realizar o ajuste sobre 2.102 pontos utilizando uma matriz de controle com 11×14 elementos. Para o método de Gálvez, Iglesias e Puig-Pey (2012), foi necessário cerca de 11 a 19 minutos para obter um resultado utilizando uma matriz de controle com 20×20 elementos. Apesar do ambiente computacional de execução não ser o mesmo, a diferença no tempo é significativa.

Note que o tempo necessário para a realização do procedimento de explosão é desprezível em relação ao tempo total, e nos três casos em que tal mecanismo foi utilizado, o tempo de conquista e divisão caiu mais do que a metade. Em todos os casos, o tempo necessário para realizar a simplificação e a combinação, foram um dos principais fatores influenciadores do tempo de execução: pelo menos metade do tempo de computação foi utilizado nestas etapas.

O tempo considerável de simplificação é justificável nestes casos ao tomar o comparativo

Figura 28 – Resultados quantitativos do ajuste do *benchmark Shell*



Fonte: Elaborado pelo autor (2022).

na redução de pontos de controle, especialmente no caso do método de PBS, no qual houve uma redução de 64% destas variáveis – a matriz de controle com forma inicial 20×20 foi reduzida para 12×12 . Nos casos dos métodos de PMaC e PMeC esta redução foi de 40%, sendo que para o PMaC, a matriz de controle com forma inicial 30×10 foi reduzida para 18×10 , e no caso da PMeC, a matriz de controle com forma inicial 40×10 foi reduzida para 24×10 . Para comparação, Mao et al. (2018) geraram um resultado com uma matriz de controle composta por 154 pontos (11×14) e Gálvez, Iglesias e Puig-Pey (2012) geraram um resultado com uma matriz de controle composta por 400 pontos (20×20), ambos utilizando mais pontos de controle do que o melhor resultado obtido pelo método de ajuste DCBA.

Se por um lado a simplificação permitiu esta redução na quantidade de variáveis de representação, por outro lado, naturalmente ela fez crescer o erro de ajuste próximo do limiar de erro aceitável. Existe uma probabilidade do erro de ajuste diminuir mesmo na operação de simplificação, devido à reparametrização prévia, mas a redução dos pontos de controle costuma ter maior impacto (negativo) sobre as métricas de erro.

Analizando os dados de erro de ajuste, percebe-se que o resultado referente ao particionamento pela PBS foi o que mais cresceu com os procedimentos de simplificação e de combinação, sendo que a métrica ME final para este e para o caso de PMeC, ficou acima do limiar de erro estabelecido. Ao utilizar as métricas AIC e BIC, os resultados do método de PMaC obtiveram os melhores valores, para ambas as métricas de erro. Os valores numéricos das métricas AIC e BIC não são apresentados neste e nos demais experimentos, visto que em todos os casos testados eles acabam replicando os dados das demais métricas de erro, apenas com uma faixa de valores diferente (valores negativos e de maior grandeza). Ao final do capítulo é apresentada

uma consideração em relação a este aspecto.

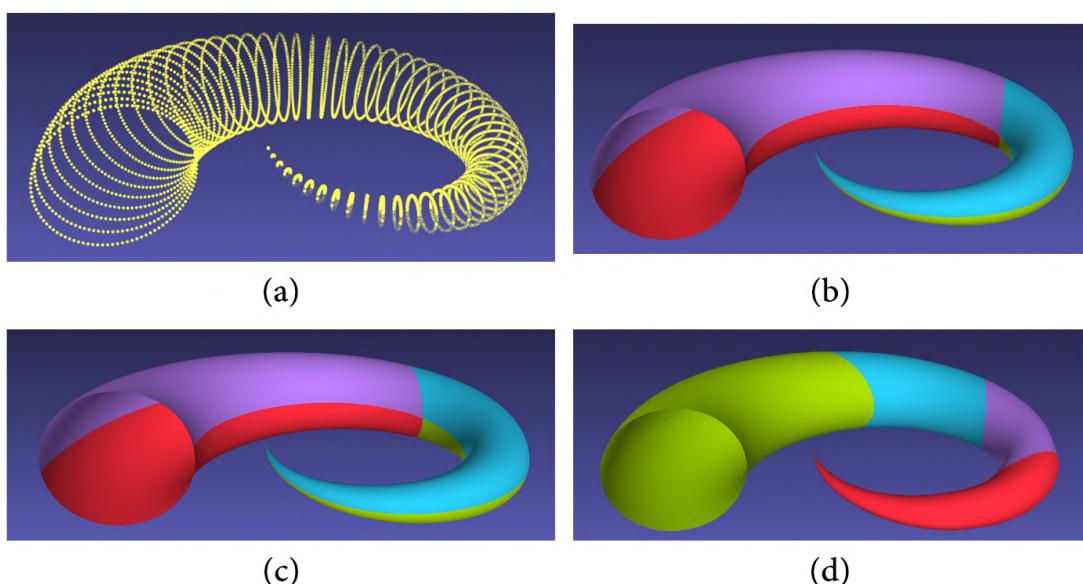
5.1.2 Benchmark Horn

O *benchmark* de superfície *Horn* é definido pela seguinte equação paramétrica (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012):

$$\begin{aligned}x(u, v) &= (2 + u \cdot \cos(v)) \cdot \sin(2\pi \cdot u) \\y(u, v) &= (2 + u \cdot \cos(v)) \cdot \cos(2\pi \cdot u) + 2u \\z(u, v) &= u \cdot \sin(v) \\u &\in [0; 1] \quad \text{e} \quad v \in [0; 2\pi]\end{aligned}$$

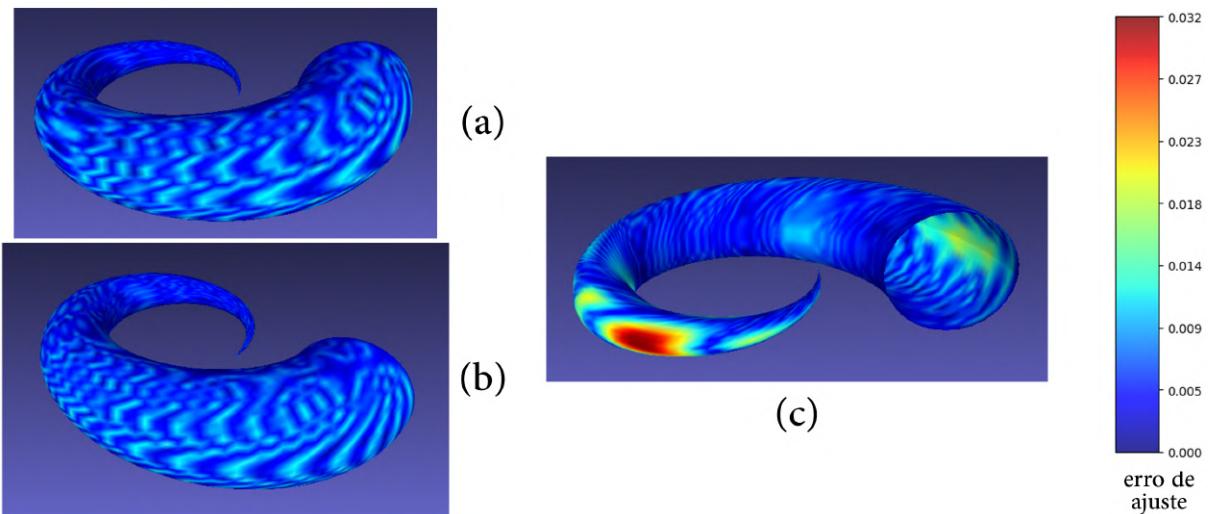
Tentando utilizar uma nuvem de pontos semelhante à utilizada por Gálvez, Iglesias e Puig-Pey (2012), com 6.000 pontos, foram utilizadas 60 amostras uniformemente distribuídas para u e 100 amostras uniformemente distribuídas para v , totalizando 6.000 pontos ($60 \cdot 100$). Neste caso, a superfície representada é fechada (tubular) em relação aos valores paramétricos de u . A Figura 29a ilustra a nuvem de pontos gerada. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,032$, $\psi = 9$ e $\zeta = 2.187$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 29, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 30 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\epsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

Figura 29 – Nuvem de pontos e resultados obtidos para o ajuste sobre o *benchmark Horn*



Fonte: Elaborado pelo autor (2022).

Figura 30 – Mapas de calor dos resultados obtidos para o ajuste sobre o *benchmark* Horn



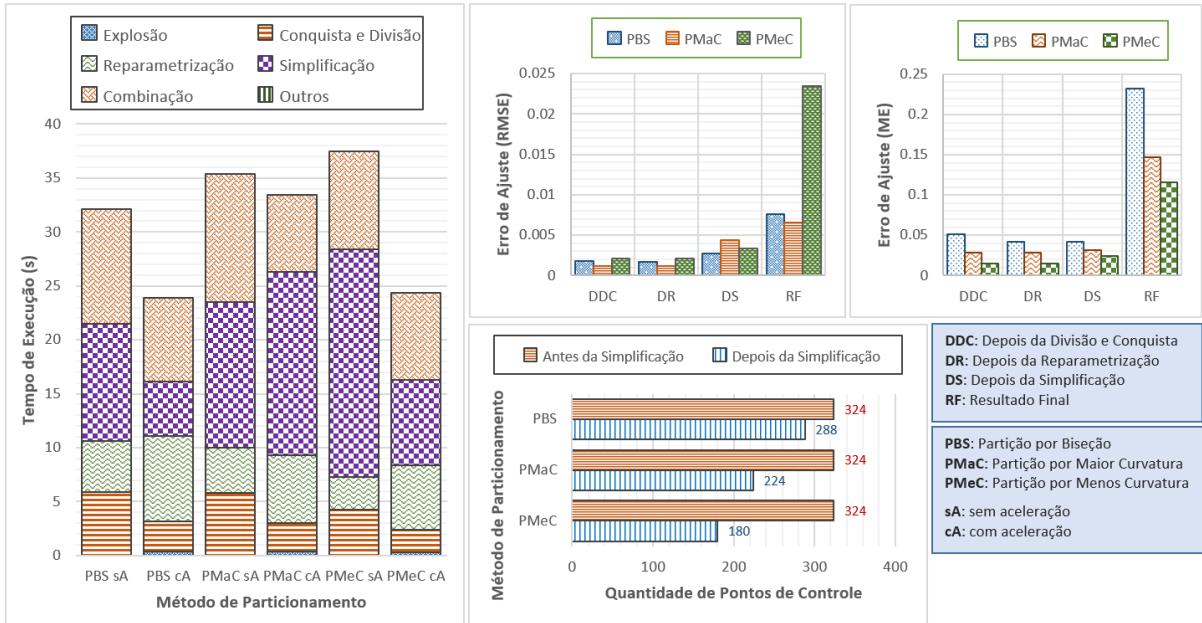
Fonte: Elaborado pelo autor (2022).

Novamente pode-se perceber visualmente que os resultados obtidos são similares, apresentando diferenças nas linhas de cortes utilizadas. Destaque para o resultado de PMeC que gerou três linhas de corte na mesma orientação paramétrica, se diferenciando dos outros dois métodos. Note ainda que para este *benchmark*, as pontas agudas do formato geométrico foram devidamente ajustadas pelos três métodos, sem a geração de artefatos estranhos.

A Figura 31 apresenta os dados quantitativos obtidos na execução do método DCBA com os diferentes métodos de particionamento. Através do gráfico de tempo de execução, pode-se perceber mais uma vez a influência positiva dos mecanismos de aceleração sobre o tempo de computação, especialmente para os métodos de PBS e PMeC. Nestes casos, os mecanismos de aceleração possibilitaram uma redução significativa no tempo de execução da operação de simplificação. Alias, é justamente a operação de simplificação que se mostra como uma das mais custosas nos casos em que os mecanismos de aceleração não foram utilizados.

Ainda sobre o aspecto de simplificação, apesar do método de PMeC ter requerido menos tempo para realizar esta operação do que o método de PMaC, sua efetividade na redução do número de pontos de controle foi maior, reduzindo uma matriz de controle inicial de 36×9 para 20×9 (quatro retalhos alinhados com forma 9×9 reduzidos para a forma 5×9). A justificativa para este caso é que o método de PMaC testou mais simplificações que não foram aceitas por extrapolar o limiar de erro tolerável, demandando mais tempo de execução. Em ambos os casos, o procedimento de simplificação evidencia sua importância no processo, reduzindo cerca de 44% dos pontos de controle utilizados, no caso de PMeC, e cerca de 31%, no caso de PMaC, com um impacto irrisório sobre as métricas de erro de ajuste, como pode ser observado nos gráficos de erro. Já no caso de PBS essa redução foi menor, de apenas 11%. Comparando os resultados obtidos, o método de Gálvez, Iglesias e Puig-Pey (2012) resultou em um ajuste com uma matriz de controle de 12×12 elementos, sendo que este resultado demorou em média entre um a dois minutos para ser calculado.

Figura 31 – Resultados quantitativos do ajuste do *benchmark Horn*



Fonte: Elaborado pelo autor (2022).

Ao analisar as variações das métricas de erro durante as etapas do processo, pode-se perceber pouca variação nas três primeiras etapas, mas um crescimento abrupto após a etapa de combinação. Com isso, os resultados obtidos ficaram todos com ME acima do limiar estabelecido. Em contrapartida, a métrica RMSE ficou abaixo do limiar para todos os casos, com destaque para o método de PMeC que obteve um valor para esta métrica significativamente maior do que os demais. Esta diferença, no entanto, não apresentou características visuais significativas e não foi identificado um motivo para esta diferença. Ao utilizar as métricas AIC e BIC, o método de PMaC obteve os melhores valores para o RMSE e o método de PMeC obteve os melhores valores para o ME. Por fim, note que apesar de utilizar a maior quantidade de pontos de controle entre os três métodos de particionamento, o método de PBS apresentou o pior índice de ME e foi apenas o segundo melhor no caso da métrica RMSE. Isso reforça a importância da qualidade dos cortes, sendo importante alocar uma maior quantidade de pontos de controle nos retalhos de superfície das regiões de maior complexidade de representação.

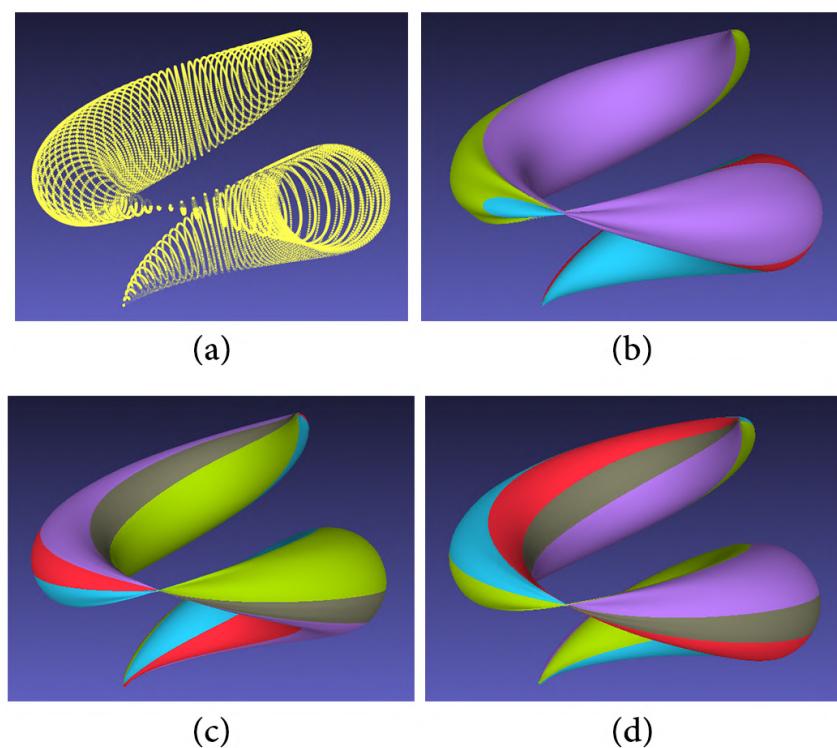
5.1.3 Benchmark Crescent

O *benchmark* de superfície *Crescent* é definido pela seguinte equação paramétrica (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012):

$$\begin{aligned}
 x(u, v) &= (2 + \sin(2\pi \cdot u) \cdot \sin(2\pi \cdot v)) \cdot \sin(3\pi \cdot v) \\
 y(u, v) &= (2 + \sin(2\pi \cdot u) \cdot \sin(2\pi \cdot v)) \cdot \cos(3\pi \cdot v) \\
 z(u, v) &= \cos(2\pi \cdot u) \cdot \sin(2\pi \cdot v) + 4v - 2 \\
 u &\in [0; 1] \quad \text{e} \quad v \in [0; 1]
 \end{aligned}$$

Tentando se aproximar da amostragem de 14.200 pontos utilizada por Gálvez, Iglesias e Puig-Pey (2012), foram utilizadas 120 amostras uniformemente distribuídas para u e v , totalizando 14.400 pontos (120^2). A superfície representada por essa equação paramétrica é fechada em relação aos valores paramétricos de v . A Figura 32a ilustra a nuvem de pontos gerada. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,166$, $\psi = 11$ e $\zeta = 4.414$. Os resultados obtidos utilizando os métodos de particionamento χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 32, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 33 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\epsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

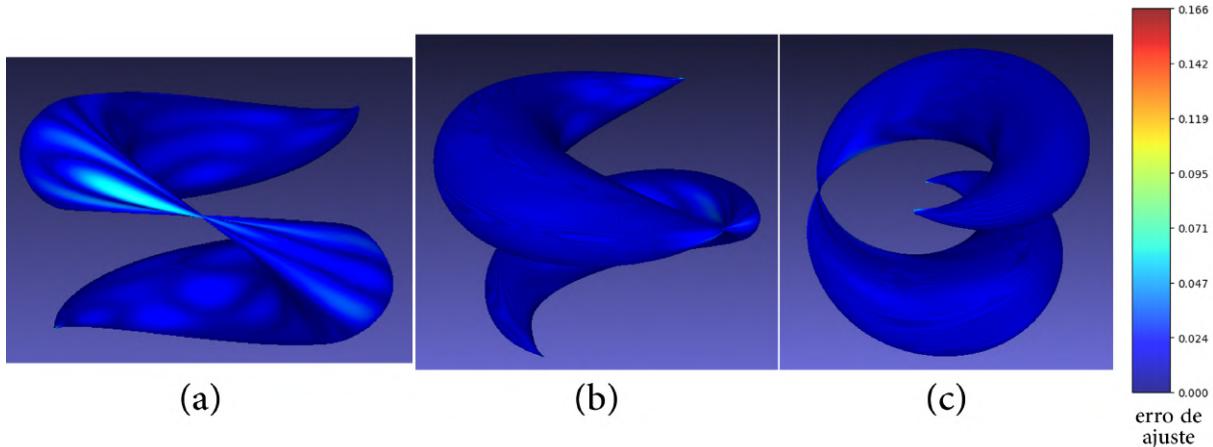
Figura 32 – Nuvem de pontos e resultados obtidos para o ajuste sobre o *benchmark* *Crescent*



Fonte: Elaborado pelo autor (2022).

A superfície deste *benchmark* possui a característica marcante de possuir duas pontas agudas, além de um ponto de singularidade na qual as bordas se interceptam, conectando as duas partes simétricas do modelo. Todos os métodos de particionamento obtiveram resultados similares em relação à qualidade visual, sendo que as pontas agudas foram bem representadas assim como o aspecto geral de curvatura do modelo. Todavia, o método de PBS obteve um resultado visivelmente pior em relação aos outros dois devido a representação do ponto de singularidade central do modelo e da região próxima a ele. Isto se deve as restrições de continuidade C^1 aplicadas sobre as bordas dos retalhos naquela região, sendo que apenas para o caso da utilização do método de PBS foi realizado um particionamento próximo ao ponto de singularidade. Perceba

Figura 33 – Mapas de calor dos resultados obtidos para o ajuste sobre o *benchmark Crescent*



Fonte: Elaborado pelo autor (2022).

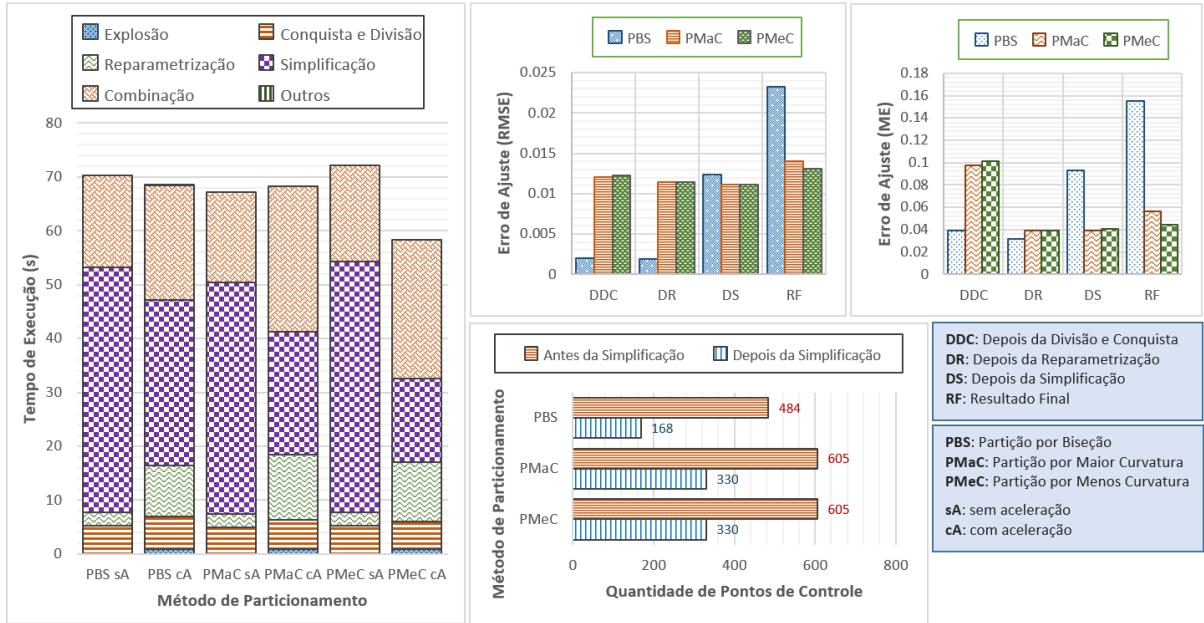
ainda que os métodos de particionamento baseados em curvatura realizaram cortes apenas em uma orientação paramétrica, ao passo que o método PBS realizou um corte em cada orientação paramétrica.

A pior representação visual do método de PBS é explicada pela menor quantidade de retalhos de superfície e quantidade de pontos de controle utilizadas na sua representação. A Figura 34 mostra alguns dados dos resultados obtidos e através desta pode-se perceber que tanto antes como depois do procedimento de simplificação, a quantidade de pontos de controle é menor no caso do uso do método de PBS. Enquanto que ambos os métodos de particionamento baseados em curvatura reduziram uma matriz de controle de 55×11 para 30×11 (cinco retalhos alinhados com forma inicial 11×11 reduzidos para a forma 6×11), o PBS reduziu a matriz de controle de 22×22 para 12×14 (quatro retalhos com forma inicial 11×11 foram reduzidos para: dois retalhos com forma 6×8 e dois retalhos com forma 6×6).

Ainda em relação ao procedimento de simplificação, perceba o impacto desta operação sobre todos os casos de teste realizados neste experimento. A redução de pontos de controle variou entre 45% e 65%. Como consequência, no entanto, o tempo de execução de ajuste de superfícies foi dominado pelo tempo necessário para o procedimento de simplificação nos casos em que os mecanismos de aceleração não foram utilizados. Já nos casos em que os mecanismos de aceleração foram utilizados, o tempo de simplificação também foi significativo, mas ficou equilibrado com o tempo utilizado para o procedimento de combinação. Realizando uma comparação, os resultados obtidos por Gálvez, Iglesias e Puig-Pey (2012) foram gerados com uma B-Spline de matriz de controle composta por 30×30 pontos de controle, necessitando um tempo de cálculo entre 14 a 20 minutos. Novamente o resultado obtido pelo DCBA foi significativamente mais rápido e escolheu de modo adaptativo uma menor quantidade de pontos de controle, gerando um resultado visualmente semelhante.

Perceba que apesar da redução no tempo de simplificação, neste experimento o impacto dos mecanismos de aceleração foi menor, com exceção do método de PMeC. Se por um lado o

Figura 34 – Resultados quantitativos do ajuste do *benchmark Crescent*



Fonte: Elaborado pelo autor (2022).

procedimento de simplificação exigiu menor tempo com o uso dos mecanismos de aceleração, por outro lado, devido as condições de distribuição dos retalhos de superfície, as etapas de reparametrização e combinação tiveram um aumento no tempo de execução necessário.

Analizando os erros de ajuste elencados na Figura 34, pode-se verificar que o método de PBS iniciou com os melhores índices de erro de ajuste. Estes valores, no entanto, tiveram um crescimento significativo nas etapas de simplificação e combinação, mas se mantiveram abaixo do limiar de erro de ajuste estabelecido. Este crescimento ocorreu porque como o erro estava significativamente abaixo do limiar estipulado, foi possível simplificar mais os retalhos resultantes, aumentando o erro. Além disso, com um número menor de pontos de controle, as restrições da combinação favoreceram o aumento do erro de ajuste. Já no caso dos métodos de particionamento baseados em curvatura, o comportamento da variação de erro foi equivalente, com poucas alterações nos valores no decorrer das etapas do método de ajuste. A única exceção foi a redução significativa de ME ocorrida após o procedimento de reparametrização. Ao utilizar as métricas AIC e BIC, o método de PMeC obteve os melhores valores para RMSE e ME.

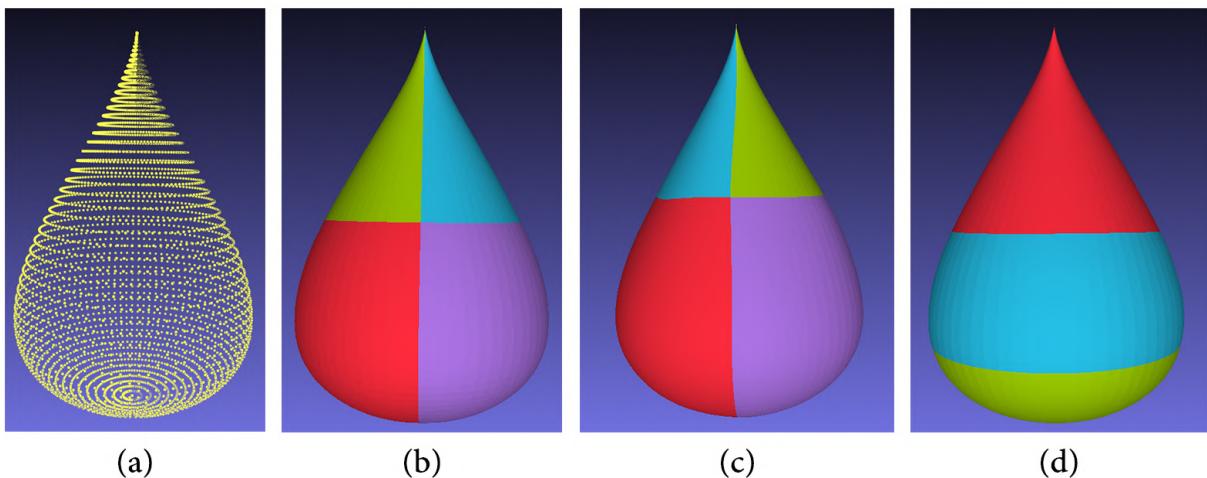
5.1.4 Benchmark TearDrop

O *benchmark* de superfície *TearDrop* é definido pela seguinte equação paramétrica (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012):

$$\begin{aligned}x(u, v) &= \frac{1}{2}(1 - \cos(u)) \cdot \sin(u) \cdot \cos(v) \\y(u, v) &= \frac{1}{2}(1 - \cos(u)) \cdot \sin(u) \cdot \sin(v) \\z(u, v) &= \cos(u) \\u &\in [0; \pi] \quad \text{e} \quad v \in [0; 2\pi]\end{aligned}$$

Tentando utilizar uma nuvem de pontos semelhante a utilizada por Gálvez, Iglesias e Puig-Pey (2012), com 6.000 pontos, foram utilizadas 60 amostras uniformemente distribuídas para u e 100 amostras uniformemente distribuídas para v , totalizando 6.000 pontos ($60 \cdot 100$). Neste caso, a superfície representada é fechada em relação aos valores paramétricos de u . A Figura 35a ilustra a nuvem de pontos gerada. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,020$, $\psi = 9$ e $\zeta = 2.187$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 35, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 36 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\epsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

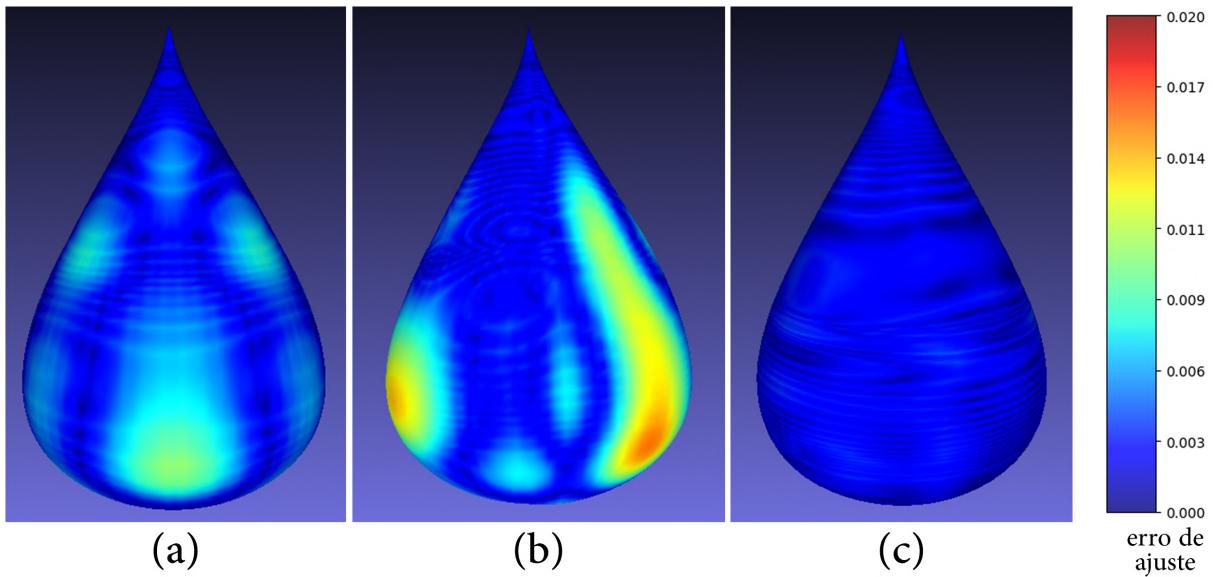
Figura 35 – Nuvem de pontos e resultados obtidos para o ajuste sobre o *benchmark* *TearDrop*



Fonte: Elaborado pelo autor (2022).

Todos os resultados obtidos conseguiram representar os aspectos geométricos da nuvem de pontos gerada, sendo que os resultados obtidos pelos métodos PBS e PMaC são similares em relação aos cortes realizados. Neste experimento, o método de PMeC se destacou por realizar apenas cortes em uma mesma orientação paramétrica, gerando um retalho de superfície a menos

Figura 36 – Mapas de calor dos resultados obtidos para o ajuste sobre o *benchmark* *TearDrop*

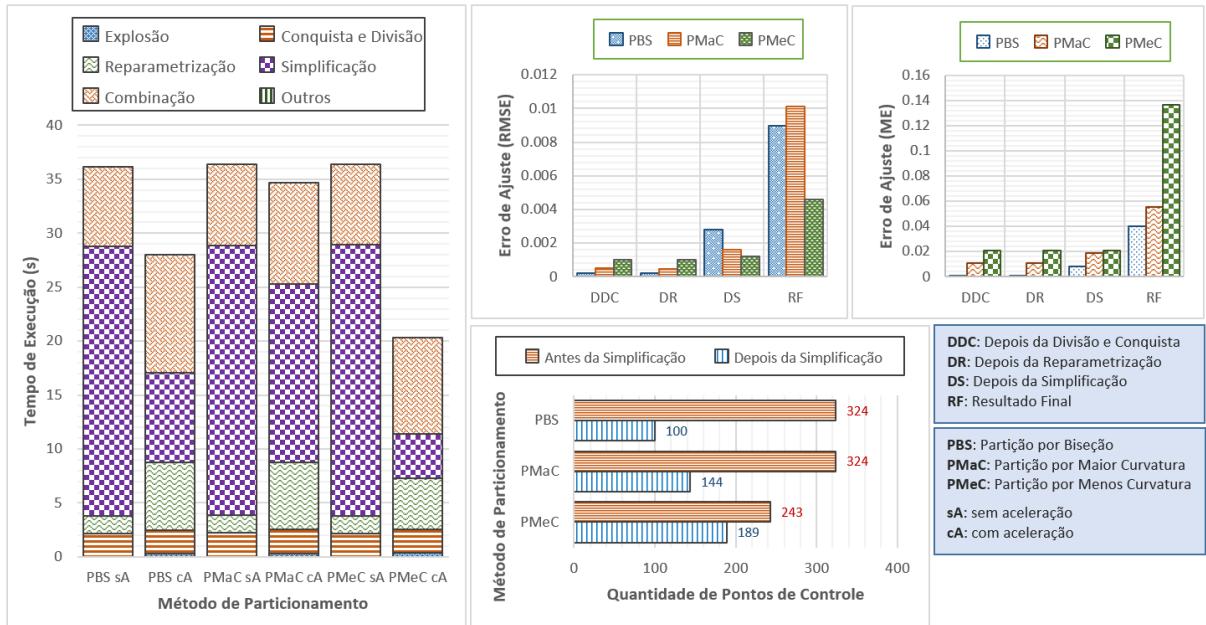


Fonte: Elaborado pelo autor (2022).

do que os demais. No entanto, vale destacar que apesar de utilizar um retalho a menos, a quantidade de pontos de controle total utilizada por este método é maior do que foi empregado pelos dois outros métodos de partição. A Figura 37 apresenta esta diferença, evidenciando que apesar de inicialmente a quantidade de pontos de controle ser menor para o método de PMeC, após o procedimento de simplificação os outros dois métodos de particionamento conseguiram diminuir significativamente este valor. No caso do método de PBS, a redução foi de aproximadamente 69% – reduzindo uma matriz de controle de 18×18 para 10×10 (quatro retalhos de superfícies de forma inicial 9×9 reduzidos para a forma 5×5) – e no caso de PMaC, a redução foi de cerca de 55% – reduzindo uma matriz de controle de 18×18 para 12×12 (quatro retalhos de superfícies de forma inicial 9×9 reduzidos para: retalhos de forma 5×5 , 5×7 , 7×5 e 7×7).

Apesar da redução significativa na quantidade de pontos de controle, o tempo gasto na operação de simplificação do método de PBS, com a utilização de mecanismos de aceleração, não foi a operação mais custosa no processo de ajuste. Apenas no caso de PMaC o tempo gasto na operação de simplificação teve o maior impacto sobre o tempo total necessário para o ajuste. Em contrapartida, note que para os três métodos de particionamento, quando os mecanismos de aceleração não foram empregados, o tempo gasto em simplificação foi predominante. Vale ainda observar que as operações de combinação apresentaram um impacto semelhante e significativo no tempo de execução total, e que o método de PMeC com mecanismos de aceleração foi a abordagem que obteve o menor tempo de execução. Comparando os resultados obtidos, Gálvez, Iglesias e Puig-Pey (2012) utilizaram este *benchmark* obtendo um ajuste com uma superfície com uma matriz de controle composta por 16×16 elementos, um valor maior do que todos os resultados obtidos pelo método proposto. Neste caso, no entanto, o tempo de cálculo necessário para o método de Gálvez, Iglesias e Puig-Pey (2012) foi semelhante ao do método proposto,

Figura 37 – Resultados quantitativos do ajuste do *benchmark TearDrop*



Fonte: Elaborado pelo autor (2022).

exigindo entre 20 a 65 segundos para calcular as respostas.

Analizando o impacto das etapas do método sobre o erro de ajuste, pode-se perceber que apesar da significativa diminuição na quantidade de pontos de controle, ocorrida no procedimento de simplificação, o erro de ajuste não sofreu um aumento equivalente. Isso reforça novamente a importância da etapa de simplificação no método proposto. Já no caso da operação de combinação, o impacto sobre o erro de ajuste foi significativamente maior, conforme ilustrado na Figura 37. Note que este impacto da operação de combinação fez com que todos os valores de ME crescessem acima do limiar de erro estabelecido. Além disso, a diferença no erro de ajuste final, fez o método de PMeC apresentar os melhores valores para as métricas AIC e BIC referentes à RMSE, e o método de PBS apresentar os melhores valores para estas mesmas métricas referentes à ME.

5.1.5 Benchmark Pisot Triaxial

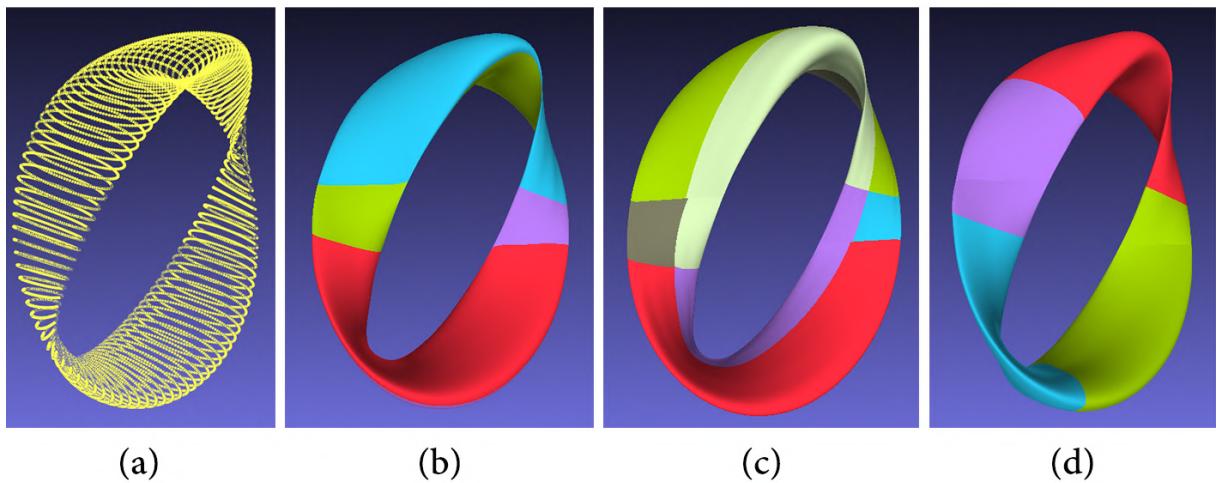
O *benchmark* de superfície *Pisot Triaxial* é definido pela seguinte equação paramétrica (GÁLVEZ; IGLESIAS; PUIG-PEY, 2012; MAO et al., 2018):

$$\begin{aligned}
 x(u, v) &= A \cdot \cos(B + u) \cdot (2 + \cos(v)) \\
 y(u, v) &= C \cdot \cos(D - u) \cdot (2 + E \cdot \cos(F + v)) \\
 z(u, v) &= E \cdot \cos(F + u) \cdot (2 + G \cdot \cos(H - v)) \\
 u &\in [0; 2\pi] \quad \text{e} \quad v \in [0; 2\pi]
 \end{aligned}$$

Tentando se aproximar da amostragem de 10.500 pontos utilizados por Gálvez, Iglesias e Puig-Pey (2012) e Mao et al. (2018), foram utilizadas 100 amostras uniformemente distribuídas

para u e 105 amostras uniformemente distribuídas para v , totalizando 10.500 pontos ($100 \cdot 105$). Foram utilizados também os mesmos valores empregados nestes trabalhos para as constantes: $A = 0,655866$, $B = 1,03002$, $C = 0,754878$, $D = 1,40772$, $E = 0,868837$, $F = 2,43773$, $G = 0,495098$ e $H = 0,377696$. Neste *benchmark*, a superfície representada é fechada tanto em relação aos valores paramétricos de u quanto v e, além disso, a superfície não apresenta uma ponta aguda, como nos demais casos, mas apresenta características desafiadoras que se assemelham a uma fita de Möbius (HENLE, 1994). Note que diante à organização da nuvem de pontos, existem duas regiões de torção nas quais os retalhos de superfície do ajuste se cortam. A Figura 38a ilustra a nuvem de pontos gerada e as regiões de corte entre retalhos mencionada fica mais evidente na Figura 38b, onde pode-se perceber que o retalho de superfície verde inicia em um lado da superfície, transpassa o retalho azul e continua no outro lado da superfície.

Figura 38 – Nuvem de pontos e resultados obtidos para o ajuste sobre o *benchmark Pisot Triaxial*

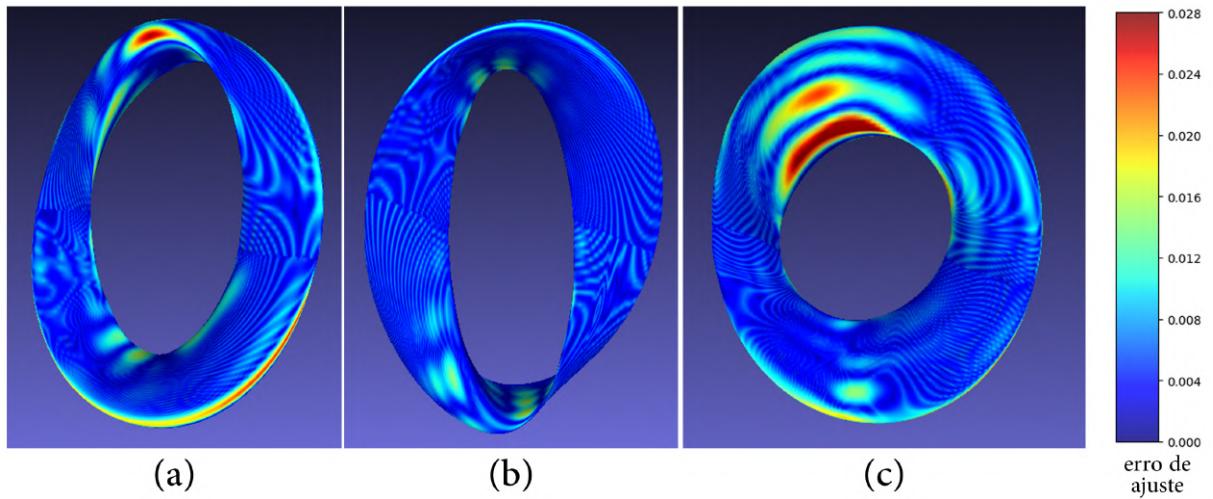


Fonte: Elaborado pelo autor (2022).

Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,028$, $\psi = 11$ e $\zeta = 4,414$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 38, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 39 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\epsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

Percebe-se que apesar das características geométricas mais desafiadoras deste modelo, todos os métodos de partição utilizados conseguiram capturar a geometria do objeto. Neste caso o método de PMaC se destaca por ser o único método a ter utilizado linhas de corte em ambas as orientações paramétricas, o que resultou em uma maior quantidade de retalhos de superfícies e, neste caso, uma quantidade significativamente maior de pontos de controle. Apesar desta maior quantidade de pontos de controle, o resultado com este método de partição obteve o melhor

Figura 39 – Mapas de calor dos resultados obtidos para o ajuste sobre o *benchmark Pisot Triaxial*



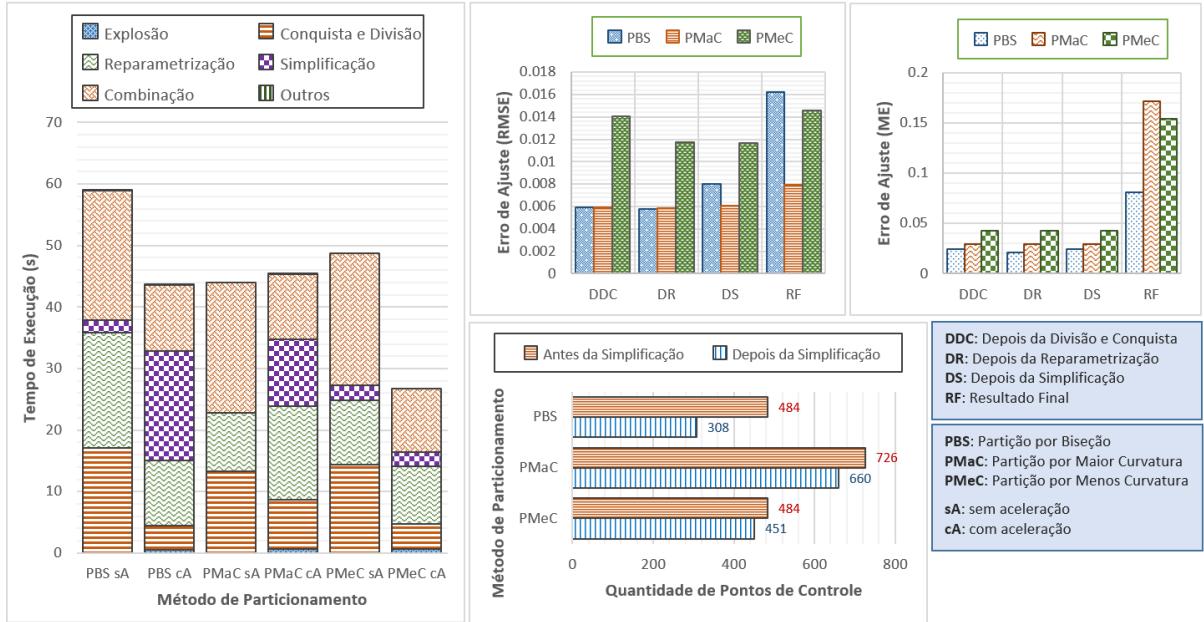
Fonte: Elaborado pelo autor (2022).

resultado apenas em relação à métrica RMSE.

A Figura 40 apresenta os dados quantitativos obtidos na execução do método com os diferentes modos de particionamento. Analisando ainda as quantidades de pontos de controle utilizadas nos ajustes, pode-se perceber que para este caso, a redução na quantidade destas variáveis foi mais limitada do que nos demais casos, variando entre 36% no caso de PBS – no qual a matriz de controle inicial de 22×22 foi reduzida para 22×14 (quatro retalhos de superfície com forma inicial 11×11 reduzidos para: dois retalhos de forma 11×6 e dois retalhos de forma 11×8) – 7% no caso de PMaC – no qual a matriz de controle inicial de 22×33 foi reduzida para 22×30 (seis retalhos de superfície com forma inicial 11×11 , sendo que apenas dois retalhos foram reduzidos para uma forma 11×8) – e 7% no caso de PMeC – no qual a matriz de controle inicial de 44×11 foi reduzida para 41×11 (quatro retalhos de superfície com forma inicial 11×11 , sendo que três deles mantiveram a mesma estrutura e um retalho foi reduzido para 8×11). Comparando com os resultados da literatura, Gálvez, Iglesias e Puig-Pey (2012) obtiveram uma superfície de matriz de controle composta por 20×20 elementos, a qual exigiu cerca de 2 a 3 horas para ser ajustada. Já o resultado de Mao et al. (2018) utilizou uma superfície composta por uma matriz de controle com 21×21 elementos, demandando pouco menos de 4 minutos. Assim, o resultado obtido pelo método proposto se mostra atrativo, principalmente em relação ao tempo de cálculo necessário.

Ainda com relação ao processo de simplificação, pode-se perceber uma alteração significativa no tempo necessário para esta operação ao comparar os métodos de PBS e PMaC sem mecanismos de aceleração em relação aos mesmos métodos de partição com mecanismos de aceleração. Note que ao não utilizar a aceleração inicial por explosão da nuvem de pontos, os métodos de particionamento empregam um tempo substancialmente maior nas operações de divisão e conquista, mas posteriormente, gastam pouco tempo na simplificação, por não

Figura 40 – Resultados quantitativos do ajuste do *benchmark Pisot Triaxial*



Fonte: Elaborado pelo autor (2022).

haver simplificação possível a ser realizada. Através da Figura 40 é possível verificar a redução considerável no tempo de execução ao utilizar os mecanismos de aceleração, quando os métodos de PBS e PMeC são aplicados. O mesmo não ocorreu para o método de PMaC porque o tempo extra de reparametrização e de simplificação compensaram o menor tempo nas operações de conquista e divisão, e de combinação.

Analizando o impacto das etapas do método sobre o erro de ajuste, pode-se perceber novamente que as métricas de RMSE e ME sofreram pouca variação entre as etapas de conquista e divisão, reparametrização e simplificação, apesar de que neste caso a simplificação não foi tão expressiva como em outros experimentos. Já no caso da operação de combinação, o impacto sobre a métrica ME foi significativamente maior. Este impacto da operação de combinação fez com que todos os valores de ME crescessem acima do limiar de erro estabelecido. Note que no caso específico do método de PMeC, tal métrica de erro já estava acima deste limiar, mesmo depois da reparametrização. Com base nos resultados, o método de PMaC apresentou os melhores valores para as métricas AIC e BIC referentes à RMSE, e o método de PBS apresentou os melhores valores para estas mesmas métricas referentes à ME.

5.1.6 Benchmark Elliptic Torus

O *benchmark* de superfície *Elliptic Torus* é definido pela seguinte equação paramétrica, a qual foi encontrada em *Wolfram MathWorld*⁷:

$$x(u, v) = (C + A \cdot \cos(v)) \cdot \cos(u)$$

$$y(u, v) = (C + A \cdot \cos(v)) \cdot \sin(u)$$

$$z(u, v) = B \cdot \sin(v)$$

$$u \in [0; 2\pi] \quad \text{e} \quad v \in [0; 2\pi]$$

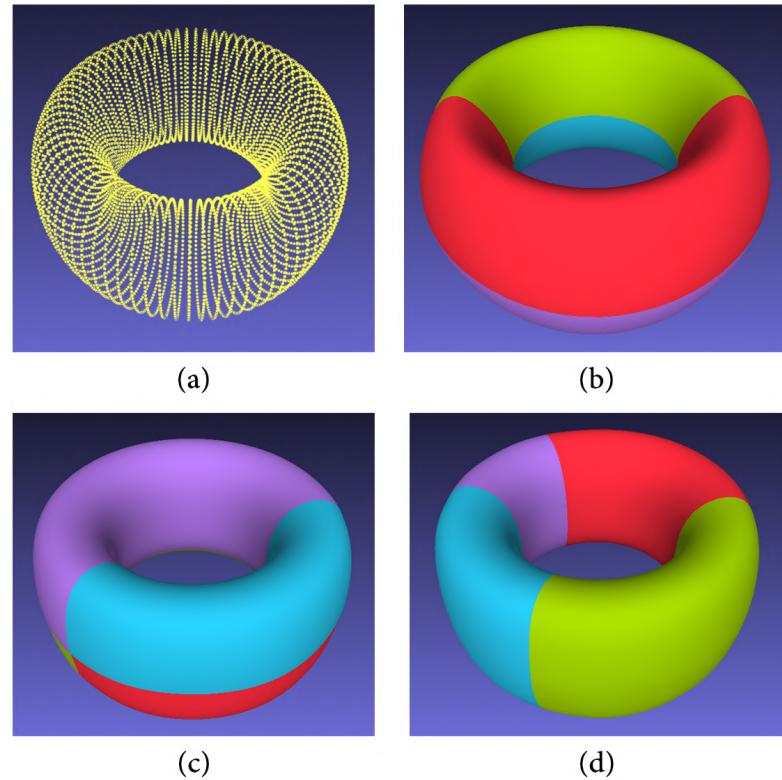
Foram utilizadas 80 amostras uniformemente distribuídas para u e v , totalizando 6.400 pontos (80^2), empregando os seguintes valores para as constantes da equação: $A = 1$, $B = 2$, $C = 3$, os quais foram definidos conforme demonstração na página de consulta desta equação paramétrica. Este *benchmark* é um objeto clássico de modelagem geométrica e sua superfície também é fechada em relação aos valores paramétricos de u e v . Apesar de ser amplamente conhecido, não foi encontrado outro trabalho na literatura que tenha usado este objeto em seus experimentos. A Figura 41a ilustra a nuvem de pontos gerada. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\varepsilon_{tol}^{[\gamma]} = 0,121$, $\psi = 9$ e $\zeta = 2.187$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 41, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 42 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\varepsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

Os resultados visuais das geometrias obtidas evidenciam a qualidade nos ajustes obtidos, incluindo a continuidade obtida entre as superfícies. Note que os métodos de PBS e PMaC realizaram cortes em ambas as orientações paramétricas, ao passo que o método de PMeC realizou cortes em apenas uma direção paramétrica, gerando superfícies com relação de autovizinhança em uma orientação paramétrica (superfícies tubulares). Em ambos os casos foi gerado um total de quatro retalhos de superfícies.

Alguns resultados numéricos obtidos são apresentados na Figura 43, dentre os quais são elencados os valores da quantidade de pontos de controle utilizada pelo ajuste de cada método de partição. Conforme mencionado, em todos os casos foram geradas quatro superfícies paramétricas, totalizando inicialmente 324 pontos de controle. Apesar desta isonomia inicial, o resultado após o procedimento de simplificação foi consideravelmente diferente. No caso do método de PBS foi possível reduzir em 69% a quantidade de pontos de controle, reduzindo uma matriz de controle de 18×18 para 10×10 ; no caso do método de PMaC foi possível reduzir em 40% a quantidade de pontos de controle, reduzindo a matriz de controle de 18×18 para 16×12 ;

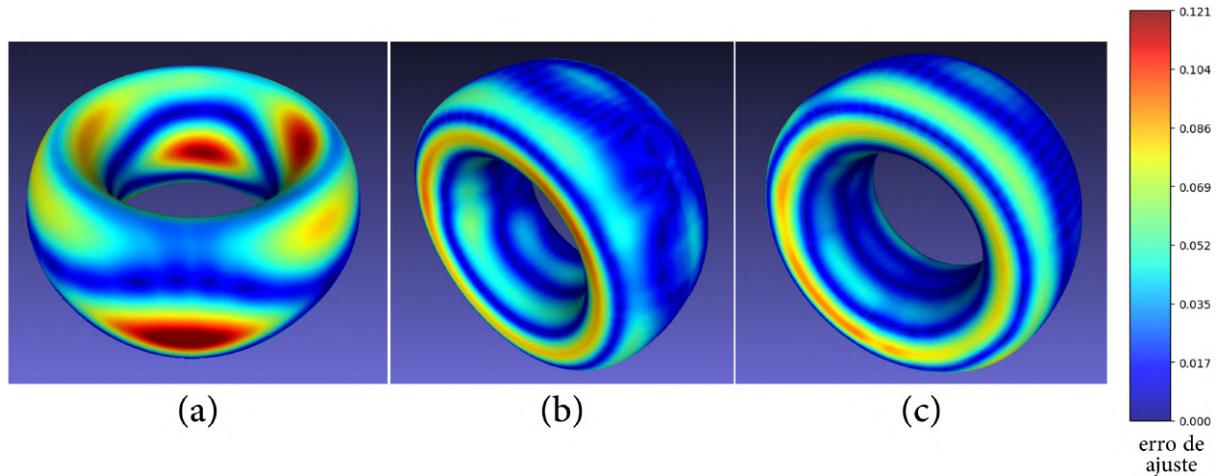
⁷ Wolfram MathWorld – Elliptic Torus (<<https://mathworld.wolfram.com/EllipticTorus.html>>)

Figura 41 – Nuvem de pontos e resultados obtidos para o ajuste sobre o *benchmark Elliptic Torus*



Fonte: Elaborado pelo autor (2022).

Figura 42 – Mapas de calor dos resultados obtidos para o ajuste sobre o *benchmark Elliptic Torus*

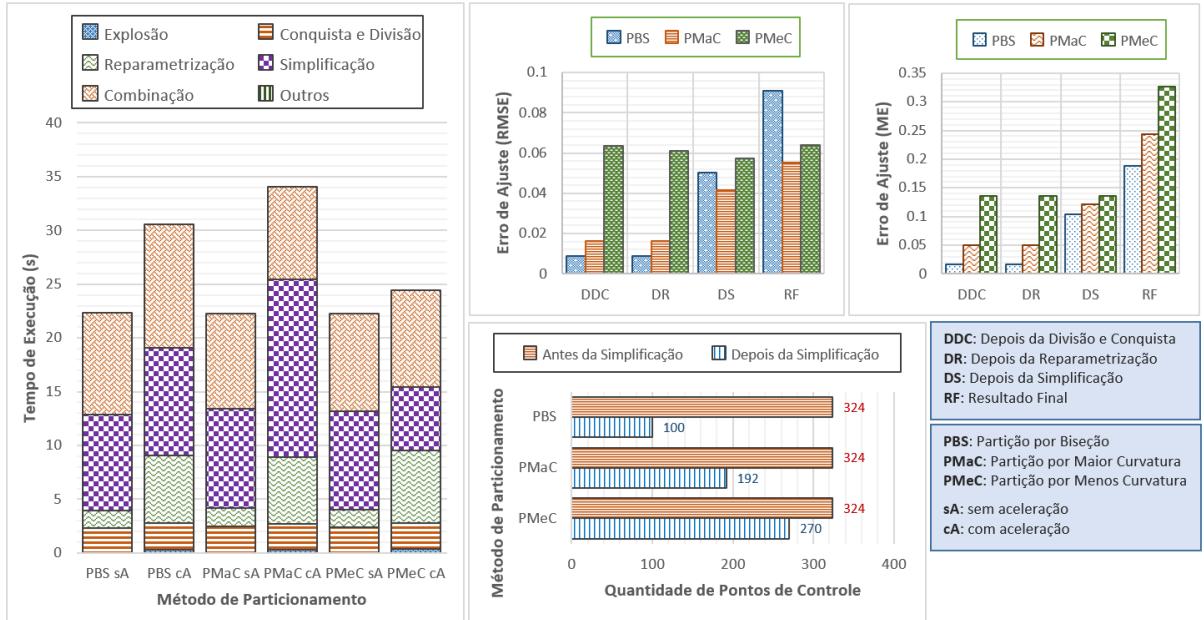


Fonte: Elaborado pelo autor (2022).

e no caso do método de PMeC foi possível reduzir apenas em 16% a quantidade de pontos de controle, reduzindo a matriz de controle de 36×9 para 30×9 .

No tocante ao tempo de execução, pode-se observar uma inversão de resultados, na qual a utilização dos mecanismos de aceleração resultaram em um ajuste com maior tempo de execução. Isso ocorreu porque, para este experimento, ao não utilizar o mecanismo de explosão

Figura 43 – Resultados quantitativos do ajuste do *benchmark Elliptic Torus*



Fonte: Elaborado pelo autor (2022).

da nuvem de pontos, o ajuste inicial com apenas uma partição já foi suficiente para obter uma métrica de erro menor do que o limiar estipulado. Como consequência, uma menor quantidade de superfícies precisaram ser reparametrizadas, reduzindo o tempo total necessário. Isto indica que, para este caso, o parâmetro ζ poderia ter sido utilizado com um valor maior, evitando uma partição obrigatória que não seria necessária para obter o limiar de erro de ajuste.

Em relação à evolução do erro de ajuste obtido nas diferentes etapas, percebe-se que a reparametrização teve pouco efeito sobre o erro de ajuste. Por outro lado, no procedimento de simplificação, os métodos que realizaram maior redução na quantidade de pontos de controle apresentaram um aumento nas métricas de erro. Perceba que apesar deste aumento e, considerando que os ajustes realizados com o método de PBS e PMaC utilizam significativamente menos pontos de controle, as métricas de erro dos três casos são similares após a etapa de simplificação. Por fim, a etapa de combinação fez crescer mais uma vez o erro de ajuste em todos os casos, fazendo com que o erro de ajuste final ficasse acima do limiar estabelecido. Ao utilizar as métricas AIC e BIC, o método de PMaC obtém os melhores resultados em relação à RMSE e o método de PBS obtém os melhores resultados em relação à ME.

5.1.7 Superfícies Explícitas

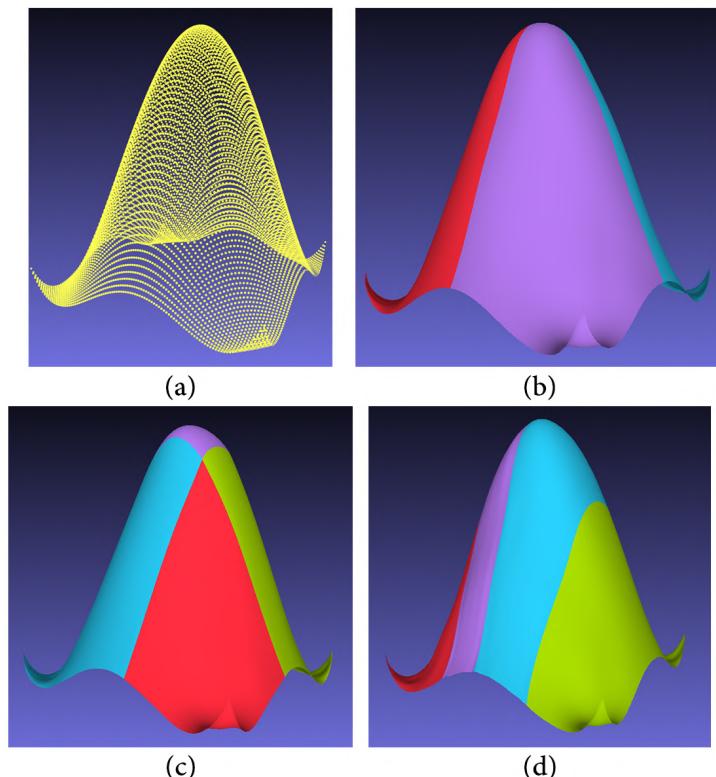
Explorando *benchmarks* de nuvens de pontos definidas por equações explícitas, foram testadas duas superfícies abertas utilizadas nos experimentos de Lee (2002). A primeira equação

escolhida, a qual será denominada por Equação A, é definida por:

$$\begin{aligned}x &\in [0; 1] \quad / \quad y \in [0; 1] \\z &= f(x, y) = 24,234 \cdot (r^2 \cdot (0,75 - r^2)) \\ \text{onde: } r^2 &= (x - 0,5)^2 + (y - 0,5)^2\end{aligned}$$

Foram utilizadas 100 amostras uniformemente distribuídas para x e y , totalizando 10.000 pontos (100^2) ao todo. Como a nuvem de pontos resultante se mostrou desproporcional no eixo z , multiplicou-se a função $f(x, y)$ por um fator $\frac{1}{3}$, resultando em uma escala mais semelhante entre os eixos e mais próxima ao que foi apresentado no trabalho de Lee (2002). A nuvem de pontos resultantes desta amostragem utilizada é apresentada na Figura 44a. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,018$, $\psi = 10$ e $\zeta = 3.162$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 44, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 45 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\epsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

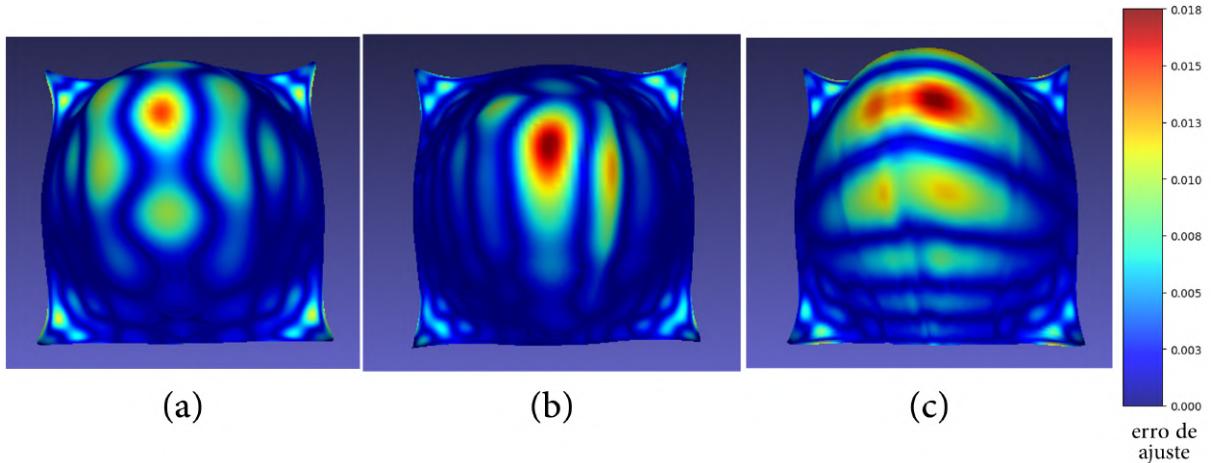
Figura 44 – Nuvem de pontos e resultados obtidos para o ajuste sobre Equação A



Fonte: Elaborado pelo autor (2022).

Os ajustes realizados para cada método de particionamento puderam capturar bem a forma geral da superfície, replicando seus detalhes de curvatura. Além disso, mais uma vez

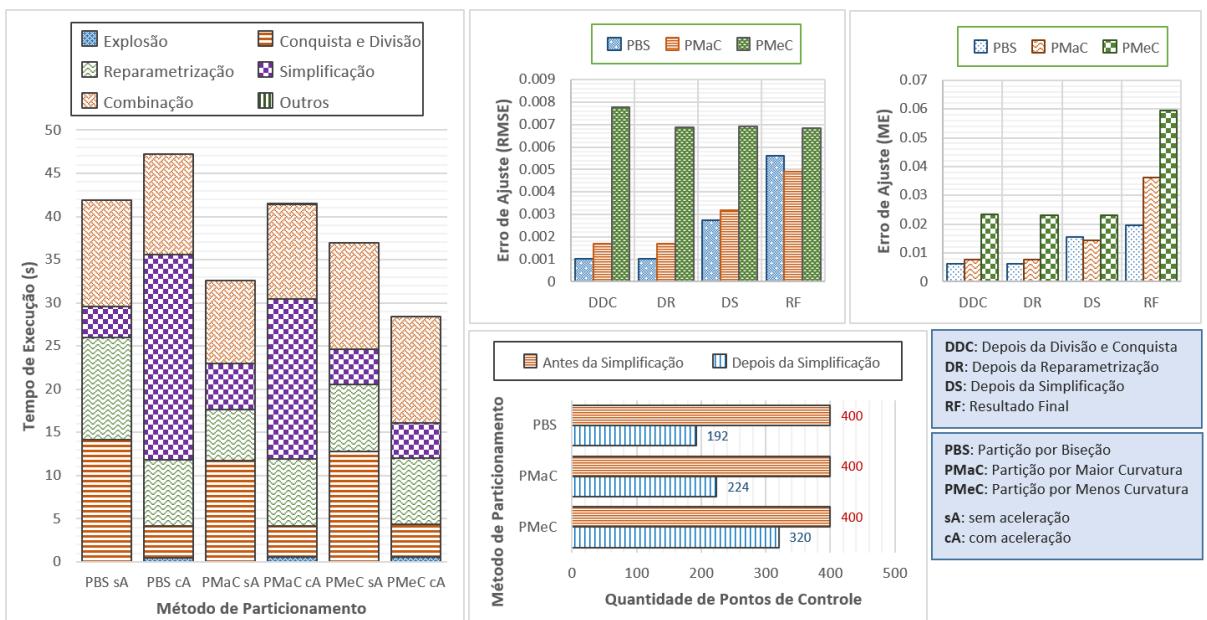
Figura 45 – Mapas de calor dos resultados obtidos para o ajuste sobre Equação A



Fonte: Elaborado pelo autor (2022).

ocorreu a criação da mesma quantidade de retalhos de superfície com uma disparidade entre as orientações de corte, conforme evidenciado na Figura 44. Os resultados quantitativos são apresentados na Figura 46 e ajudam a entender melhor as características destes ajustes realizados. Pode-se perceber que os 400 pontos de controle iniciais foram reduzidos diferentemente para cada método de particionamento. A redução variou entre 52% no caso do método de PBS – reduzindo a matriz de controle de 20×20 para 12×16 – 44% no caso do método de PMaC – reduzindo a matriz de controle de 20×20 para 14×16 – e 20% no caso do método de PMeC – reduzindo a matriz de controle de 40×10 para 32×10 .

Figura 46 – Resultados quantitativos do ajuste sobre a nuvem de pontos gerada pela Equação A



Fonte: Elaborado pelo autor (2022).

Neste experimento, o tempo de simplificação foi proporcional à redução na quantidade de

pontos de controle, sendo que para o caso do uso dos métodos de PBS e PMaC, com mecanismos de aceleração, o tempo desta etapa dominou o tempo geral de execução. Isto fez com que a obtenção do resultado por estes métodos de particionamento fosse mais demorada com a utilização de mecanismos de aceleração do que sem eles. Todavia, deve-se ressaltar que o resultado obtido sem os mecanismos de aceleração não são necessariamente os mesmos daqueles obtidos com a utilização destes mecanismos. Isto é, apesar do maior tempo de execução, não se pode afirmar que o uso dos mecanismos de aceleração piorou o resultado. Note ainda que o gráfico de tempo de execução da Figura 46 mostra a maior quantidade de tempo utilizado na etapa de conquista e divisão, quando o procedimento de aceleração por explosão da nuvem de pontos não foi empregado.

Analizando os erros de ajuste obtidos, percebe-se um crescimento linear nas etapas de simplificação e de combinação das superfícies, tanto para a métrica RMSE quanto para a métrica ME. A exceção ocorre para o caso do método de PMeC, no qual o RMSE é consideravelmente maior nas primeiras etapas do ajuste, se comparado aos resultados dos demais métodos de particionamento. Já no caso do erro máximo, a diferença não é tão substancial, mas após o procedimento de combinação, o erro aumenta expressivamente. Ao utilizar os índices AIC e BIC, o método de PMaC obtém os melhores resultados em relação à RMSE, e o método de PBS obtém os melhores resultados em relação à ME.

A segunda equação explícita escolhida dentre os experimentos de Lee (2002), a qual também representa uma superfície aberta e será denominada por Equação B, é definida como:

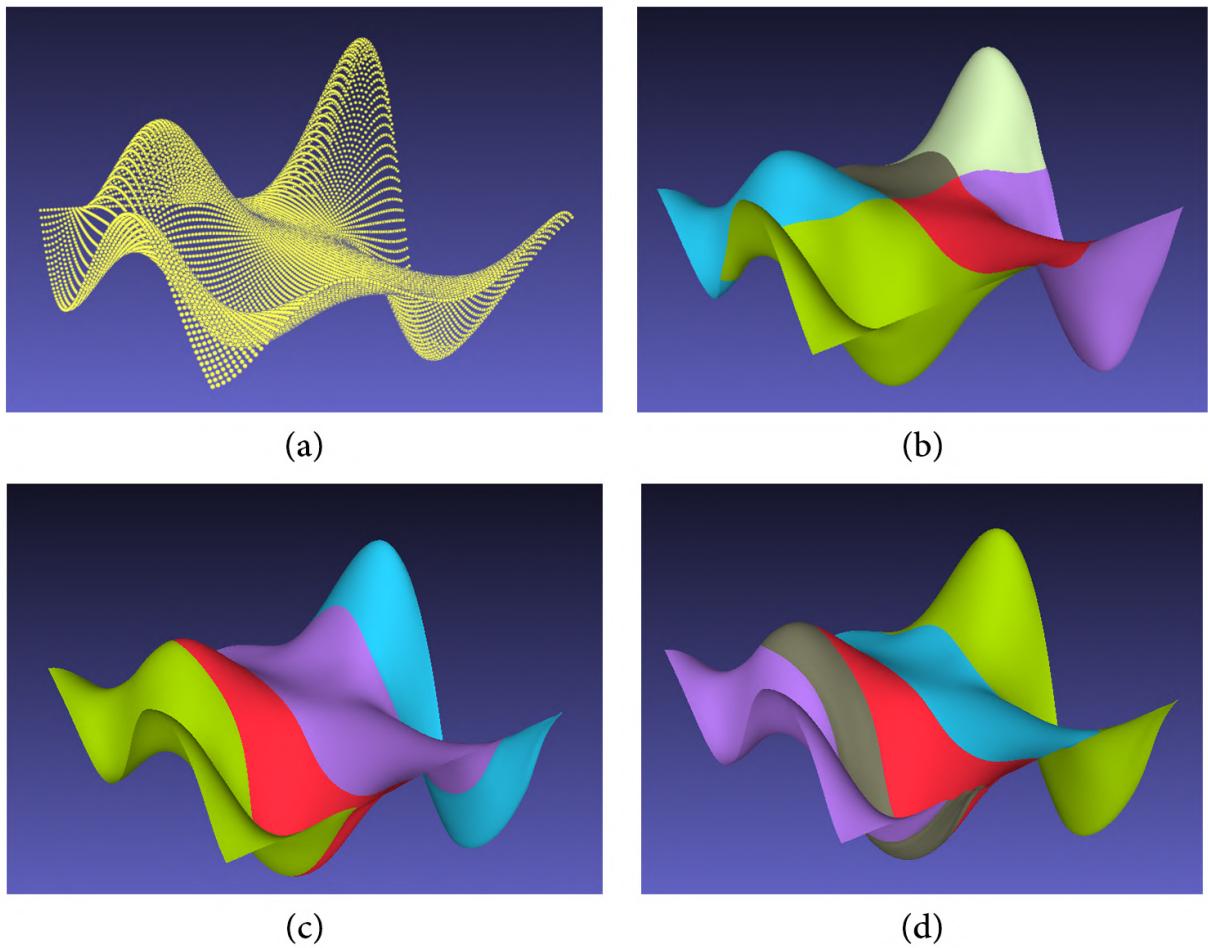
$$x \in [0; 1] \quad / \quad y \in [0; 1]$$

$$f(x, y) = 1,9 * (1,35 + e^x \cdot \sin(13 \cdot (x - 0,6)^2) \cdot e^{-y} \cdot \sin(7y))$$

Novamente foram utilizadas 100 amostras uniformemente distribuídas para x e y , totalizando 10.000 pontos (100^2) ao todo. A nuvem de pontos resultante também se mostrou desproporcional no eixo z , e um multiplicador $\frac{1}{7}$ foi utilizado sobre a função $f(x, y)$ de modo a gerar um resultado com aspecto uniforme entre os eixos e mais semelhante ao que foi apresentado por Lee (2002). A nuvem de pontos resultantes desta amostragem utilizada é apresentada na Figura 47a. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,013$, $\psi = 10$ e $\zeta = 3.162$. Os resultados obtidos utilizando os métodos propostos para χ (PBS, PMaC e PMeC) são apresentados nas imagens (b), (c) e (d) da Figura 47, respectivamente. Para auxiliar na identificação das medidas de erro de ajuste é apresentada também a Figura 48 que apresenta os mapas de calor dos resultados obtidos com a utilização dos métodos propostos para χ : (a) PBS, (b) PMaC e (c) PMeC. O valor de $\epsilon_{tol}^{[\gamma]}$ foi utilizado como maior valor da escala de erro para os mapas apresentados.

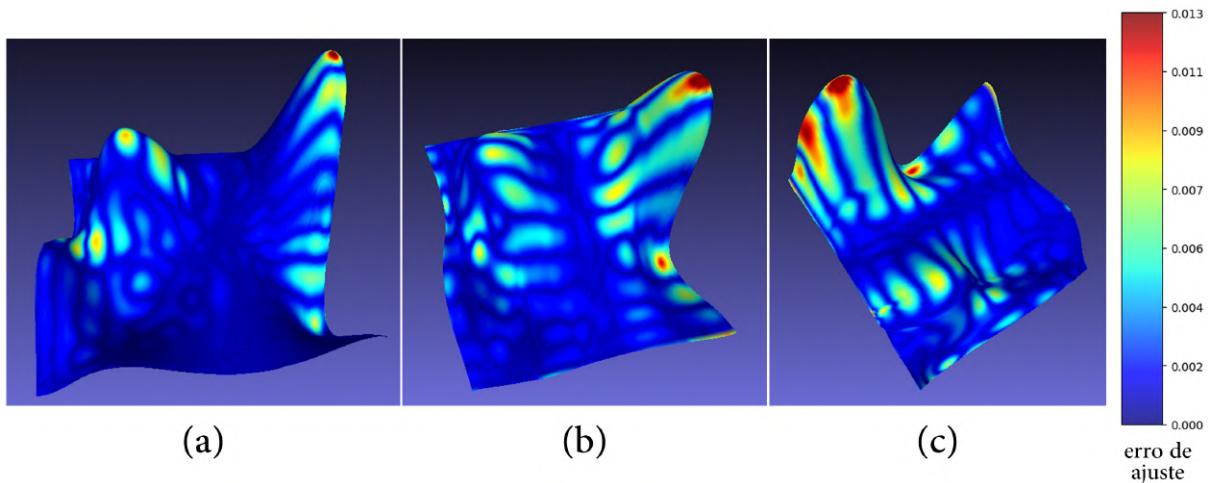
Os ajustes realizados conseguiram representar bem as alterações de relevo deste *benchmark*. Neste caso, houve uma diferença na quantidade de retalhos de superfícies para cada método de particionamento. O método de PBS gerou seis retalhos, formando uma matriz de controle inicial 30×20 ; o método de PMaC gerou quatro retalhos, formando uma matriz de

Figura 47 – Nuvem de pontos e resultados obtidos para o ajuste sobre Equação B



Fonte: Elaborado pelo autor (2022).

Figura 48 – Mapas de calor dos resultados obtidos para o ajuste sobre Equação B

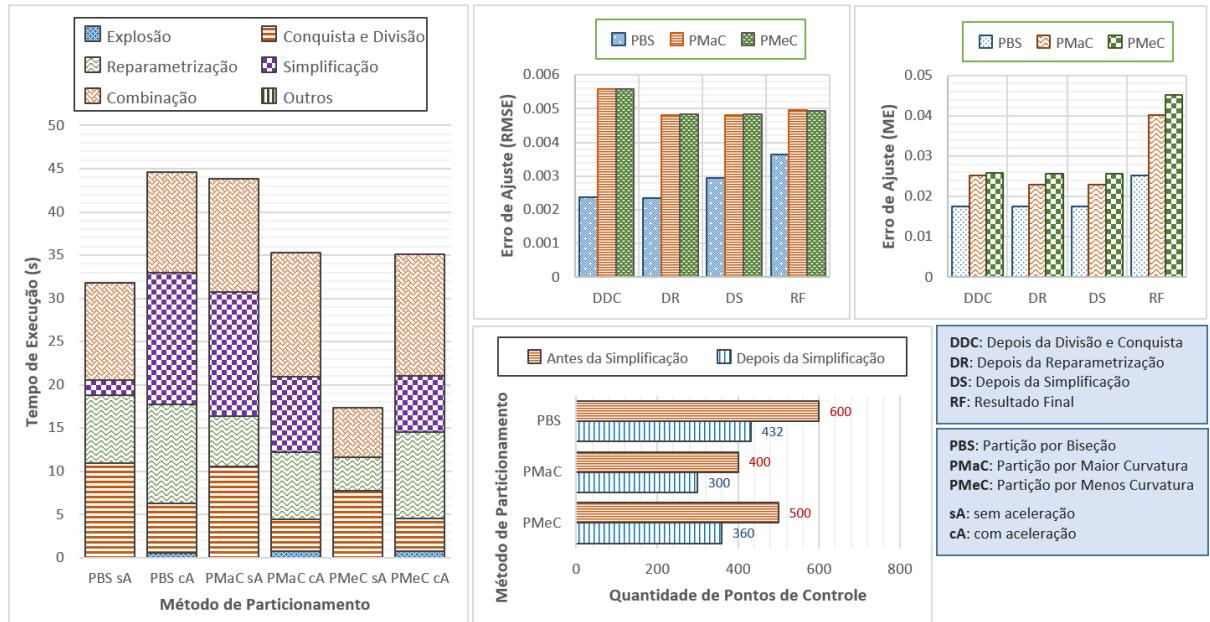


Fonte: Elaborado pelo autor (2022).

controle inicial de 40×10 ; e o método PMeC gerou cinco retalhos, formando uma matriz de controle inicial de 50×10 . Para contribuir no entendimento destes processos de ajustes é apre-

sentada a Figura 49, que exibe alguns dados quantitativos referentes aos ajustes. Continuando pela análise da matriz de controle, pode-se perceber que todas as matrizes de controle iniciais foram reduzidas proporcionalmente para cada método de particionamento. Esta redução variou de 28%, no caso do método de PBS e PMeC – reduzindo as matrizes de controle para 24×18 e 36×10 , respectivamente – a 25%, no caso do método de PMaC – reduzindo a matriz de controle para 30×10 .

Figura 49 – Resultados quantitativos do ajuste sobre a nuvem de pontos gerada pela Equação B



Fonte: Elaborado pelo autor (2022).

Em relação ao tempo de execução, as operações de conquista e divisão e combinação dominaram o tempo de execução quando os mecanismos de aceleração não foram utilizados, com exceção do caso em que o método de PMaC foi utilizado, no qual a etapa de simplificação também exigiu um fração considerável do tempo total. Já nos casos de ajuste utilizando os mecanismos de aceleração, três etapas demandaram a maior quantidade de tempo: reparametrização, simplificação e combinação. Além disso, novamente foram registrados dois casos em que o tempo total de execução foi maior ao utilizar os mecanismos de aceleração.

Por fim, com relação às métricas de erro de ajuste obtidas, pode-se observar um comportamento de poucas alterações nestas medidas durante as etapas do método. Isso pode ser explicado por uma reparametrização que obteve pouco impacto sobre o melhoramento das métricas de erro e por um processo de simplificação que conseguiu remover pontos de controle que contribuíam pouco para a solução. Já a alteração na métrica de ME após a etapa de combinação foi mais significativa. Com base nestes dados, as métricas de AIC e BIC indicaram os resultados do método de PBS como os melhores para este experimento e parâmetros utilizados.

5.2 EXPERIMENTOS COM NUVENS NÃO ORGANIZADAS

Nuvens de pontos não organizadas também foram utilizadas para verificar a efetividade do método proposto para reconstruir superfícies. A principal diferença prática entre estas duas modalidades de nuvem para o método proposto é que a parametrização de uma nuvem de pontos não organizada dificilmente será tão boa quanto a parametrização de uma nuvem organizada. Esta é uma característica relevante, visto que para muitos métodos de ajuste baseados em parametrização, incluindo o ajuste tradicional de superfícies, uma boa parametrização é essencial para produzir bons resultados.

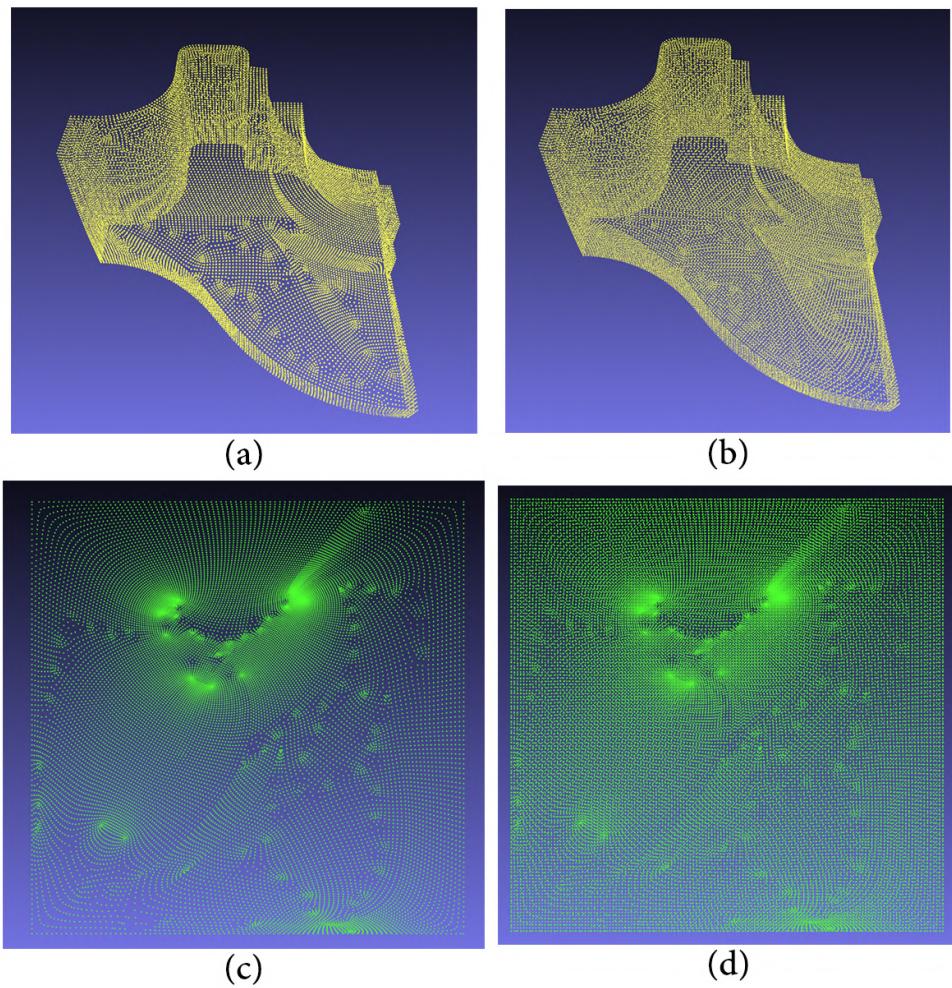
Foram testados três modelos de malha triangular encontrados em repositórios públicos e que foram mencionados em alguns trabalhos da literatura. Para obter uma parametrização de entrada para estes modelos, aplicou-se o método equiareal (*authalic*) sobre as malhas triangulares obtidas. Esta escolha teve como base testes realizados com o mapeamento conforme, equiareal e *Mean Value Coordinates* (FLOATER, 2003; FLOATER; KÓS; REIMERS, 2005), utilizando a biblioteca CGAL já mencionada no início deste capítulo. Como nenhum método pareceu fornecer resultados significativamente melhores do que os outros, optou-se pelo método equiareal pelo seu objetivo de tentar reduzir distorções nas áreas, visto que alguns problemas neste sentido foram detectados nas parametrizações realizadas.

Ao utilizar o método DCBA com os mesmos parâmetros e esquemas empregados no ajuste de nuvens de pontos organizadas, obteve-se um resultado de má qualidade. Apesar de diversos retalhos de superfície serem ajustados com precisão, alguns retalhos sofreram com problemas de falta de amostras no domínio paramétrico, conduzindo a sistemas lineares mal condicionados. A seguir, nas próximas subseções serão apresentadas figuras das parametrizações obtidas, através das quais pode-se perceber que é comum a ocorrência de uma maior concentração de amostras no centro do espaço paramétrico, ao passo que as periferias costumam apresentar uma baixa amostragem. É exatamente esta característica da parametrização que produz o problema identificado, visto que após alguns particionamentos, as regiões periféricas ficam subamostradas.

Em face deste problema de distribuição paramétrica, adicionou-se para os casos apresentados a seguir, um mecanismo de superamostragem que ajuda a evitar ou minimizar este problema. Este mecanismo foi criado considerando também a hipótese de que o método proposto admite operar sobre uma maior quantidade de amostras na nuvem de pontos, sem aumentar seu tempo de execução da mesma forma que o método tradicional. Ele é utilizado uma única vez no início do processo, adicionando uma amostragem extra sobre os pontos da nuvem originais. Esta amostragem adicional é computada distribuindo uniformemente novos pontos sobre o espaço paramétrico, calculando suas respectivas posições geométricas com base em uma interpolação linear geométrica dos elementos da nuvem. A única restrição desta interpolação é que os novos pontos paramétricos devem estar dentro da mesma caixa delimitadora dos pontos já existentes (i.e. $0 \leq u, v \leq 1$). Como a nuvem de pontos já é parametrizada com um algoritmo que gera um mapeamento em um quadrado unitário paramétrico, no qual é garantida a existência dos quatro

pontos de quina ($(u = 0, v = 0)$, $(u = 0, v = 1)$, $(u = 1, v = 0)$ e $(u = 1, v = 1)$), esta restrição não é um problema para a utilização deste mecanismo. Optou-se por adicionar um total de α amostras por linhas e colunas, totalizando α^2 novas amostras, sendo que $\alpha = \lfloor(\sqrt{n})\rfloor$ e n representa o tamanho da nuvem de pontos original. Com a utilização deste valor, a nuvem de pontos dobra sua quantidade de amostras. A Figura 50 apresenta um exemplo desta amostragem adicional sendo aplicada sobre um dos modelos testados, o *FanDisk*⁸, tanto no espaço geométrico – imagens (a) e (b) – quanto no paramétrico – imagens (c) e (d).

Figura 50 – Exemplo de reamostragem inicial da nuvem de pontos



Fonte: Elaborado pelo autor (2022).

Além do mecanismo de reamostragem inicial, também foi percebido que era comum a ocorrência da repetição contínua da mesma orientação de corte para a partição, o que favorece a criação de subconjunto da nuvem de pontos com poucas amostras. A fim de evitar este tipo de situação, optou-se por simplificar o método de escolha da orientação de particionamento da nuvem por uma escolha intercalada entre cortes paralelos à orientação dos parâmetros u e v . Isto é, se um subconjunto da nuvem de pontos foi gerado a partir de um corte paramétrico na orientação

⁸ Mesh-Paramterization: ./examples/gim/fandisk.obj – <<https://github.com/Sanaxen/Mesh-Paramterization>>

associada ao valor paramétrico u , o seu particionamento será realizado necessariamente na orientação paramétrica associada ao valor paramétrico v e vice-versa.

Por fim, diferentemente do que foi apresentado na seção anterior, optou-se por verificar a efetividade do método no ajuste de nuvem de pontos não organizadas apenas com o particionamento por bissecção (PBS). Esta opção foi realizada considerando que para os métodos baseados em curvatura, seria necessário gerar uma nova triangulação para calcular os índices de curvatura, ou então aproximá-los através de interpolação.

5.2.1 Superfície Max Planck

O primeiro modelo testado de nuvem de pontos não organizada foi a superfície que retrata a cabeça de Max Planck⁹. Este mesmo modelo foi utilizado por Wang e Zheng (2013). Por se tratar de uma superfície fechada, realizou-se um corte na região do pescoço para transformar a superfície equivalente a um disco, gerando uma região de borda que é utilizada pelos algoritmos de parametrização usuais. Com este corte, a nuvem de pontos ficou constituída por 41.117 amostras. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[ME]} = 0,839$, $\psi = 15$ e $\zeta = 13.071$. A Figura 51 ilustra a nuvem de pontos na imagem (a) e a parametrização obtida pelo método equiareal na imagem (b). O resultado obtido através do ajuste pelo método proposto, utilizando PBS, é apresentado em ângulos diferentes nas imagens (c) e (d) desta mesma figura.

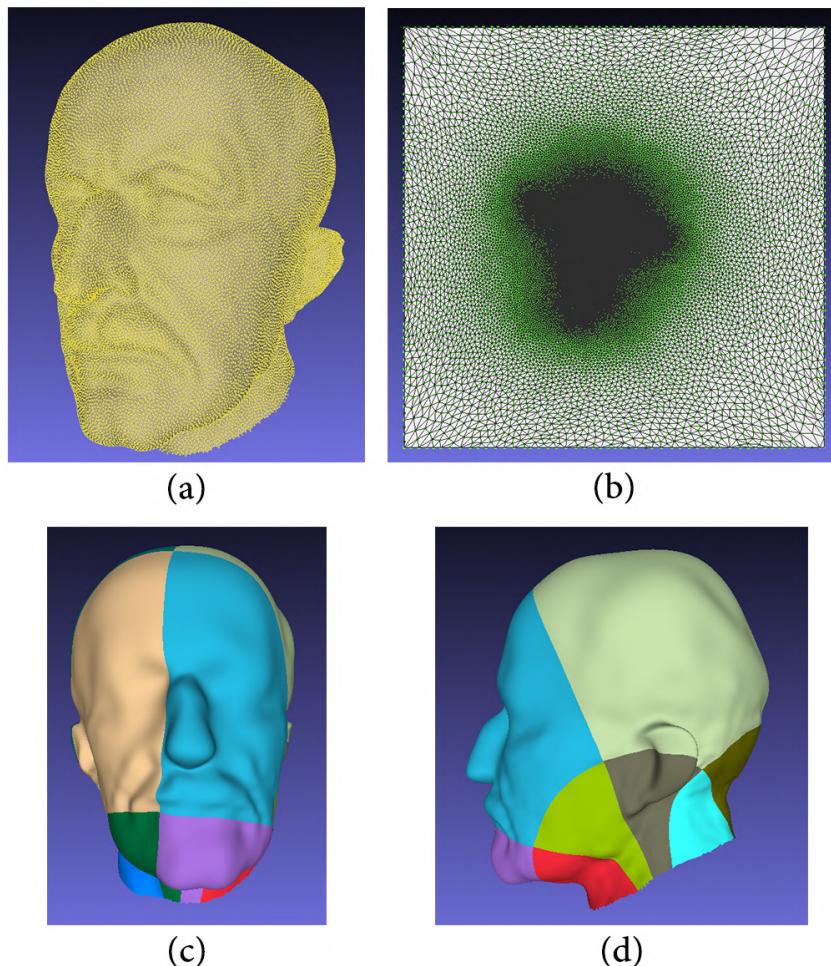
Foram utilizados 16 retalhos de superfície para realizar o ajuste, totalizando 3.600 pontos de controle empregados em uma matriz de controle total de dimensão 60×60 . Apesar do corte utilizado na região do pescoço que simplifica a superfície ajustada, o resultado obtido com os parâmetros padrões foi similar ao resultado de 3.510 pontos de controle obtido por Wang e Zheng (2013). Este resultado é relevante já que este segundo trabalho utiliza T-Splines, as quais apresentam vantagens neste aspecto em virtude da possibilidade do uso de junções do tipo T.

Destacando alguns dados quantitativos deste ajuste realizado, as métricas de erro obtidas, sem a realização de uma normalização das posições geométricas das amostras, foram: $\epsilon^{[RMSE]} = 0,788$, $\epsilon^{[ME]} = 11,798$. O valor de ME pode chamar atenção neste resultado devido a sua maior grandeza. Todavia, deve-se observar que este valor está dentro dos padrões dos demais resultados obtidos nos experimentos apresentados na seção 5.1, sendo que o RMSE ficou abaixo do $\epsilon_{tol}^{[ME]}$ e que o valor de ME é cerca de quinze vezes maior do que o RMSE, assim como ocorrido em outros resultados. Assim como realizado na seção anterior, apresenta-se a Figura 52 para auxiliar na identificação das medidas de erro de ajuste locais através do mapa de calor obtido do resultado com a utilização do método de particionamento de PBS. Neste caso, o maior valor das menores distâncias entre duas amostras da nuvem de pontos original (3,784) foi utilizado como maior valor da escala de erro para o mapa apresentado.

Através dos resultados numéricos destacados, pode-se notar que, ao utilizar os valores

⁹ Mesh-Paramterization: ./examples/gim/maxplanck.obj – <<https://github.com/Sanaxen/Mesh-Paramterization>>

Figura 51 – Nuvem de pontos, parametrização e resultado obtido para o ajuste sobre a superfície que retrata a cabeça de Max Planck

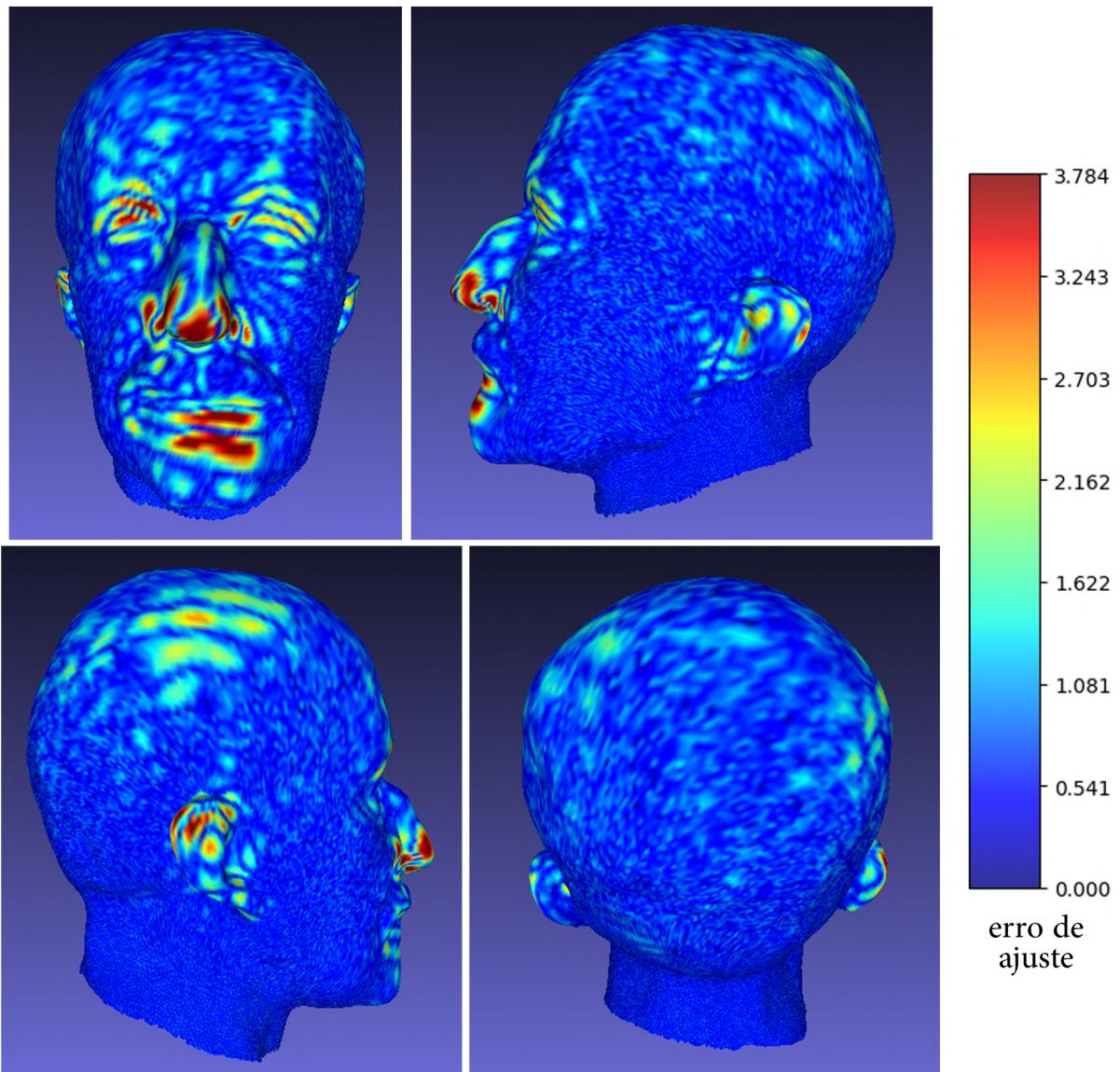


Fonte: Elaborado pelo autor (2022).

padrões para os parâmetros, mesmo particionando a nuvem de pontos até alcançar as menores quantidades de amostras permitidas, não foi suficiente para obter o limiar de erro aceitável, indicando que, ou a quantidade mínima de amostras para viabilizar um particionamento está muito elevada, ou que este limiar de erro aceitável está muito restritivo. Como consequência natural disto, o procedimento de simplificação não teve possibilidade de reduzir a quantidade de pontos de controle.

Com relação ao tempo de execução, foram necessários 220,4 segundos para finalizar o ajuste desta superfície, dos quais 2,9% foram gastos na explosão da nuvem de pontos, 33,4% foram gastos na etapa iterativa de conquista e divisão, 32,1% na reparametrização, 0,0% na simplificação (etapa foi ignorada devido ao erro de ajuste parcial ser maior do que o limiar de erro tolerável utilizado) e 31,6% na combinação dos retalhos de superfície. Dentre as três operações mais custosas deste processo, pode-se perceber que a reparametrização é a única não essencial para a obtenção de uma resposta válida. Além disso, pode-se mensurar uma queda de 12% na métrica de RMSE e de apenas 2,5% na métrica de ME. Isto é, para este caso, a reparametrização

Figura 52 – Mapa de calor do resultado obtido para o ajuste sobre a superfície que retrata a cabeça de Max Planck



Fonte: Elaborado pelo autor (2022).

utilizou uma parcela significativa do tempo total de execução, mas sua contribuição na redução de erro foi pouco relevante.

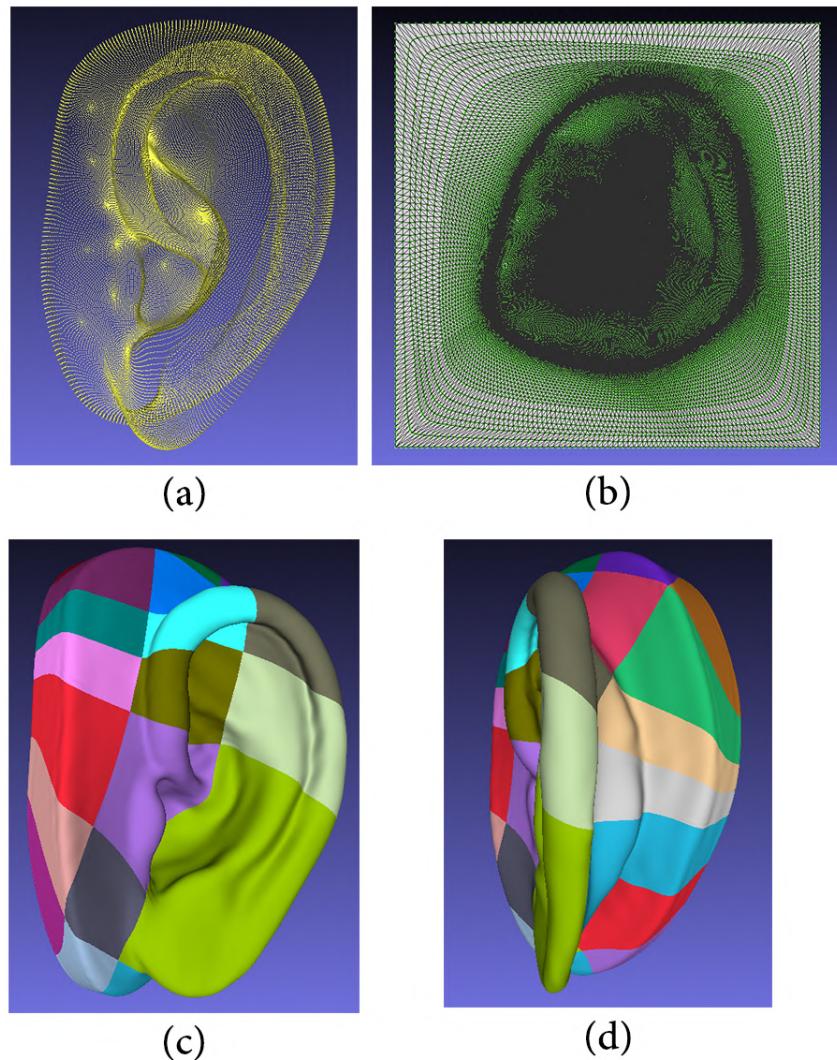
5.2.2 Superfície Modelo de Orelha

O segundo modelo testado de nuvem de pontos não organizada foi a superfície que retrata um modelo de orelha, constituído por 73.185 amostras¹⁰. Trata-se de uma superfície aberta (na parte interna da orelha), sendo que este mesmo modelo foi utilizado também por Bertolino et al. (2021). Todavia, os resultados não podem ser comparados, visto que o modelo utilizado por estes outros autores é composto por apenas 5.630 amostras. Com base nos dados da nuvem de

¹⁰ Free 3D: Ear V1 Modelo 3D – <<https://free3d.com/3d-model/ear-v1--113169.html>>

pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\varepsilon_{tol}^{[ME]} = 0,011$, $\psi = 17$ e $\zeta = 20.256$. A Figura 53 ilustra a nuvem de pontos na imagem (a) e a parametrização obtida pelo método equiareal na imagem (b), enquanto que o resultado obtido através do ajuste pelo método proposto, utilizando PBS, é apresentado em ângulos diferentes nas imagens (c) e (d).

Figura 53 – Nuvem de pontos, parametrização e resultado obtido para o ajuste sobre a superfície Modelo de Orelha

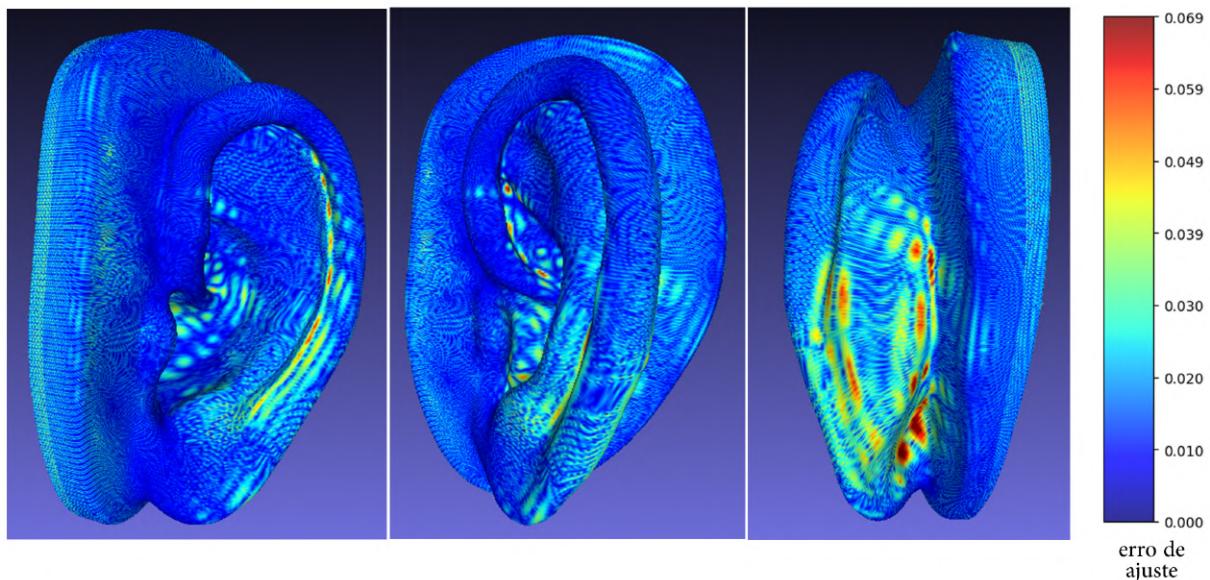


Fonte: Elaborado pelo autor (2022).

Foram utilizados 28 retalhos de superfície para realizar o ajuste, totalizando 8.092 pontos de controle empregados em uma matriz de controle total de dimensão 68×119 . Através do ajuste pelo método proposto, sem normalizar as posições geométricas dos pontos da nuvem, obteve-se as seguintes métricas de erro: $\varepsilon^{[RMSE]} = 0,012$, $\varepsilon^{[ME]} = 0,164$. Novamente, ao utilizar os valores padrões para os parâmetros, mesmo particionando a nuvem até alcançar as menores quantidades de amostras permitidas, não foi suficiente para obter o limiar de erro aceitável, fortalecendo o padrão que ocorreu no experimento apresentado na subseção 5.2.1.

Para auxiliar na análise dos erros de ajuste locais obtidos, apresenta-se a Figura 54 que mostra o mapa de calor obtido do resultado com a utilização do método DCBA com uso de particionamento de PBS sobre a nuvem de pontos do modelo de orelha. Novamente foi utilizado o maior valor das menores distâncias entre duas amostras da nuvem de pontos original (0,069) como o maior valor da escala de erro para o mapa apresentado.

Figura 54 – Mapa de calor do resultado obtido para o ajuste sobre a superfície Modelo de Orelha



Fonte: Elaborado pelo autor (2022).

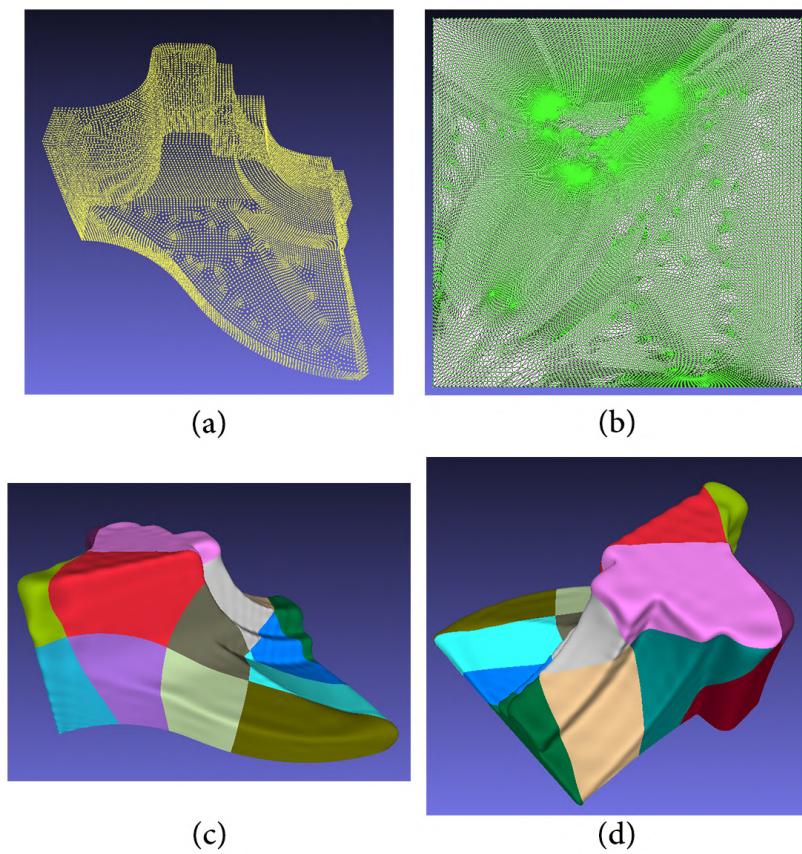
Realizando uma análise do tempo de execução, foram necessários 492,3 segundos para completar o ajuste desta superfície. Deste montante, 2,1% foi utilizado na operação de explosão da nuvem de pontos, 37,1% foi utilizado nas operações de conquista e divisão, 32,3% na reparametrização, 0,0% na simplificação (a qual foi ignorada devido ao erro de ajuste parcial ser maior do que o limiar de erro tolerável utilizado) e 28,5% nas operações de combinação das superfícies. Em um comparativo direto deste ajuste com o ajuste da superfície da cabeça de Max Planck – desprezando as diferenças nas características entre as nuvens – tem-se um aumento de aproximadamente 78% das amostras contra um aumento de 123% no tempo de computação total. Já no comparativo da eficiência da reparametrização para este caso, pode-se observar uma contribuição ainda menos significativa do que no experimento anterior. Apesar de ser a segunda operação mais custosa do método, a redução na métrica de erro RMSE foi de apenas 4,8% e de 0% no caso do ME, ao utilizar o procedimento de reparametrização.

5.2.3 Superfície *FanDisk*

O terceiro modelo testado de nuvem de pontos não organizada foi a superfície denominada na literatura como *FanDisk*, o qual já foi mencionado no inicio desta seção. Este modelo apresenta características de curvaturas suaves e regiões de quinas, sendo utilizado também nos experimentos de Wang e Zheng (2013). Novamente, por se tratar de uma superfície fechada, realizou-se um

corte na região plana inferior da superfície, transformando-a em uma superfície equivalente a um disco e gerando assim uma região de borda. Com este corte, a nuvem de pontos é constituída por 20.027 amostras. Com base nos dados da nuvem de pontos, e os valores padrões definidos para os parâmetros, foram utilizados os seguintes valores: $\epsilon_{tol}^{[\gamma]} = 0,002$, $\psi = 12$ e $\zeta = 5.985$. A Figura 55 ilustra a nuvem de pontos na imagem (a) e a parametrização obtida pelo método equiareal na imagem (b), ao passo que o resultado obtido através do ajuste pelo método proposto, utilizando PBS, é apresentado em ângulos diferentes nas imagens (c) e (d).

Figura 55 – Nuvem de pontos, parametrização e resultado obtido para o ajuste sobre a superfície *FanDisk*



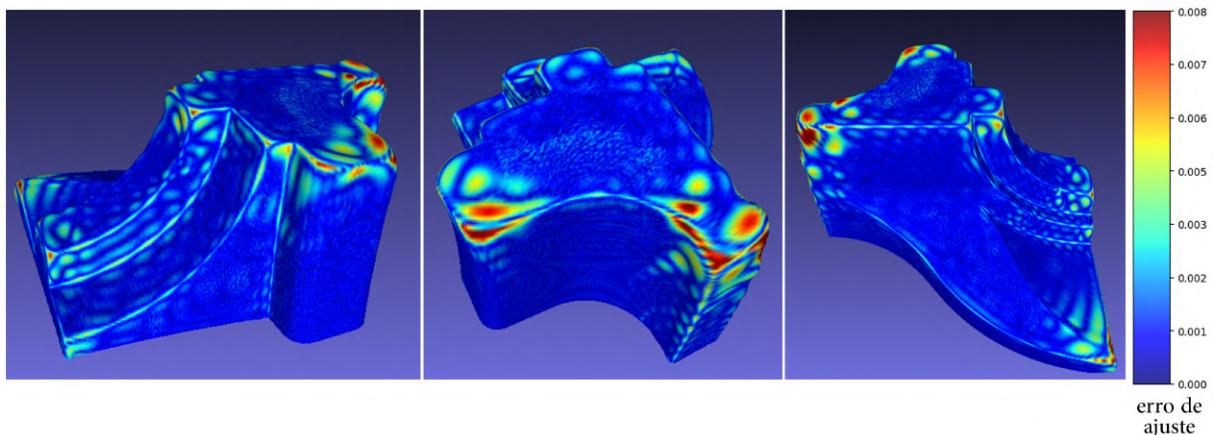
Fonte: Elaborado pelo autor (2022).

Foram utilizados 16 retalhos de superfície para realizar o ajuste, totalizando 2.304 pontos de controle empregados em uma matriz de controle total de dimensão 48×48 . Novamente a quantidade de pontos de controle utilizada foi semelhante ao resultado obtido por Wang e Zheng (2013), no qual foram empregados 2.093 pontos de controle, principalmente ao se comparar a vantagem das T-Splines neste aspecto da representação de superfície. Apesar disto, os resultados obtidos pelo método proposto mostram sua deficiência no ajuste sobre as regiões de quina da superfície, sendo que estes aspectos geométricos foram todos representados por regiões com curvatura acentuada. Para este tipo de cenário, a opção pela escolha manual das bordas dos retalhos de superfície, ou a utilização de um método de particionamento mais robusto, combinada

com uma escolha adaptativa de continuidade de bordas, viabilizaria o ajuste das regiões de quina com continuidade C^0 , ideal para este caso.

Para auxiliar na análise dos erros de ajuste locais obtidos para este experimento, apresenta-se a Figura 56 que mostra o mapa de calor obtido do resultado com a utilização do método DCBA com uso de particionamento de PBS sobre a nuvem de pontos do modelo *FanDisk*. Novamente foi utilizado o maior valor das menores distâncias entre duas amostras da nuvem de pontos original (0,008) como o maior valor da escala de erro para o mapa apresentado.

Figura 56 – Mapa de calor do resultado obtido para o ajuste sobre a superfície *FanDisk*



Fonte: Elaborado pelo autor (2022).

Apesar do aspecto de continuidade mencionado, o resultado obtido conseguiu capturar a essência do objeto. Obteve-se as seguintes métricas de erro: $\epsilon^{[RMSE]} = 0,002$, $\epsilon^{[ME]} = 0,027$. Com relação ao tempo de execução, foram necessários 119 segundos para realizar o ajuste desta superfície, sendo que: 2,4% foi utilizado na operação de explosão da nuvem de pontos, 28,6% foi utilizado nas operações de conquista e divisão, 36,2% na reparametrização, 0,0% na simplificação (ignorada devido ao erro de ajuste parcial ser maior do que o limiar de erro tolerável utilizado) e 32,8% nas operações de combinação. Mais uma vez um quantidade significativa de tempo foi utilizada na reparametrização, mas a redução de erro de ajuste obtida foi pequena, de apenas 2,8% para RMSE e de menos de 0,1% para ME.

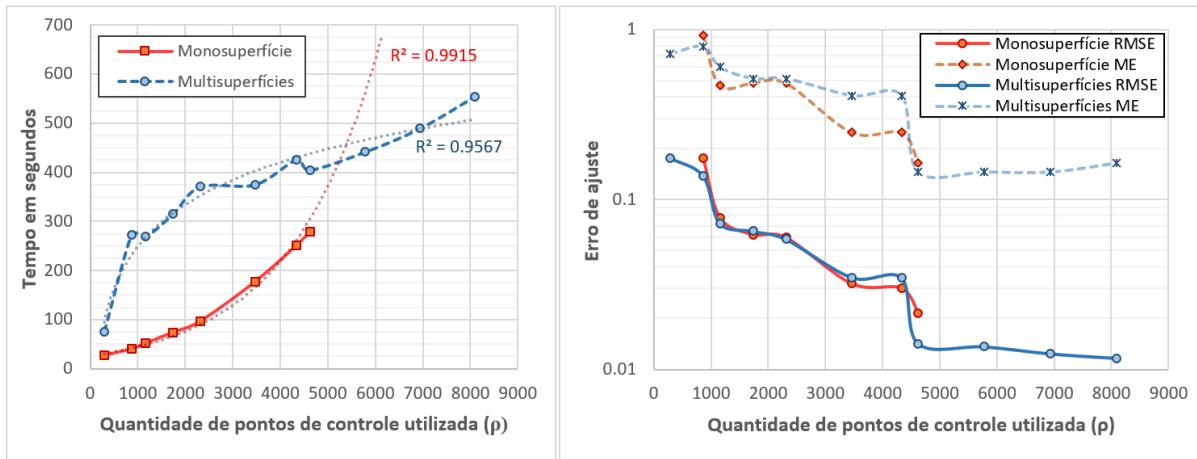
5.3 COMPARAÇÕES COM O MÉTODO DE AJUSTE PROPOSTO

Com o intuito de comparar a eficiência e a eficácia do método proposto para o ajuste por múltiplas superfícies, foi realizado um comparativo de tempo e erro de ajuste entre o método tradicional de ajuste por uma única superfície (monosuperfície) e o método DCBA (multisuperfície). Uma nuvem de pontos com uma grande quantidade de amostras foi buscada para verificar se o método proposto é mais rápido do que o método tradicional, quando a quantidade de amostras e a quantidade de pontos de controle é maior. Inicialmente foi encontrada

e experimentada a nuvem de pontos que retrata um trecho do lago Maggiore¹¹, com exatas 810.000 amostras, distribuídas de forma organizada. Esta nuvem de pontos seria ideal para realizar a comparação idealizada, no entanto, mesmo com uma pequena quantidade de pontos de controle, o método tradicional exige uma quantidade de memória principal maior do que o suportável pelo ambiente computacional utilizado. Por outro lado, o método proposto conseguiu realizar o ajuste desta superfície sem apresentar este problema.

Como alternativa, a superfície do modelo de orelha, utilizada na subseção 5.2.2, foi escolhida para esta comparação, considerando a sua maior quantidade de amostras dentre os experimentos realizados. Para fazer uma comparação justa entre os métodos, tanto em relação ao erro como tempo de ajuste, a nuvem de pontos foi reamostrada antes de ser fornecida como entrada ao método tradicional, assim como é realizado no método DCBA. Com a reamostragem, a quantidade de amostras da nuvem de pontos é de $n = 146.085$ sendo utilizada a mesma parametrização para ambos os casos. A Figura 57 apresenta dois gráficos relacionando o tempo de ajuste – no gráfico da esquerda – e as métricas RMSE e ME – no gráfico da direita – para o método tradicional (monosuperfície) e para o método DCBA (multisuperfícies).

Figura 57 – Comparação de tempo e erro de ajuste entre a abordagem tradicional e a abordagem de ajuste por múltiplas superfícies



Fonte: Elaborado pelo autor (2022).

Como o método DCBA não prevê (e nem faz sentido) a delimitação de uma quantidade fixa de pontos de controle a ser utilizada, atribuiu-se ao parâmetro ρ_{max} diversos valores para limitar a quantidade de pontos de controle máxima a ser utilizada. Com isso, diferentes resultados e configurações gerais da matriz de controle foram obtidas. Estes mesmos valores foram utilizados para o método tradicional, buscando obter a maior isonomia de parâmetros possível. As configurações do tamanho da matriz de controle utilizadas foram: (17×17) , (17×51) , (34×34) , (34×51) , (34×68) , (51×68) , (51×85) , (68×68) , (68×85) , (68×102) , (68×119) . Destas

¹¹ Visionair: A World-class Infrastructure for Advanced 3D visualization-based Reserach – <<http://visionair.ge.imati.cnr.it/ontologies/shapes/viewmodels.jsp>>

configurações, as três últimas também não puderam ser executadas com o método tradicional por problemas de capacidade de memória principal.

Primeiramente em uma análise do tempo de execução, pode-se entender melhor o comportamento dos dois métodos analisados em relação ao aumento da quantidade de pontos de controle. O comportamento do método tradicional se mostra previsível, ao passo que o crescimento temporal do método DCBA apresenta maiores variações. Estas variações são causadas pela diferença de tempo utilizado nos diversos procedimento do método, sendo que alguns deles são influenciados pela distribuição dos particionamentos realizados. Por exemplo, em um resultado com 16 retalhos de superfície, podem ocorrer a criação de uma conectividade de retalhos 4×4 ou 2×8 , por exemplo. O primeiro caso terá maiores chances de exigir um maior tempo de cálculo, visto que existem mais bordas compartilhadas do que no segundo caso.

Para poder visualizar melhor a diferença na tendência de crescimento, além dos dados obtidos no experimento, foram traçadas duas curvas de tendência (curvas tracejadas) geradas a partir de uma análise de regressão linear. A curva de tendência relacionada ao ajuste com o método tradicional teve a melhor correspondência com um comportamento quadrático ($R^2=99,15\%$), o que já era esperado dada a sua complexidade de tempo. Já no caso do método DCBA, a melhor curva de tendência encontrada é logarítmica ($R^2=95,67\%$). Através destas curvas de tendência por regressão linear, é possível perceber que a partir de um determinado ponto (neste caso, entre 5.000 e 6.000 pontos de controle) o método DCBA se torna mais rápido para fazer o ajuste – determinando uma quantidade de pontos de controle julgada como adequada – do que um único ajuste pelo método tradicional com a mesma quantidade de pontos de controle escolhida. Além disso, ao analisar o erro de ajuste obtido com estas diferentes formas de ajuste, percebe-se que as métricas obtidas foram semelhantes na maioria dos casos.

5.4 CONSIDERAÇÕES

O método DCBA obteve bons resultados visuais e numéricos para os *benchmarks* e superfícies testadas utilizando apenas os parâmetros padrões definidos. Dentre os procedimentos e etapas que compõem o método, a etapa de reparametrização apresentou resultados positivos – contribuindo para uma redução significativa da métrica de erro e viabilizando posterior simplificação das superfícies – nos experimentos envolvendo nuvens de pontos organizadas, mas teve uma contribuição menos expressiva nos experimentos realizados com nuvens de pontos não organizadas. A utilização dos mecanismos de aceleração propostos também apresentou um retorno variante. Eles contribuíram para uma redução significativa do tempo de execução, no entanto, em alguns casos influenciaram na necessidade de operações mais custosas, gerando um resultado alternativo com melhores métricas de erro, mas de maior custo computacional.

A partir dos resultados obtidos pode-se supor que os valores padrões utilizados para o limiar de erro $\varepsilon_{tol}^{[\gamma]}$ foram adequados para os experimentos envolvendo as nuvens de pontos organizadas, visto que permitiram que operações de redução fossem realizadas e ainda um resultado

geométrico satisfatório pôde ser obtido. Já no caso das nuvens de pontos não organizadas, este valor não pareceu ser adequado, inviabilizando qualquer operação de redução, visto que todos os ajustes parciais, sem combinação, já eram superiores a métrica imposta. Mais testes precisam ser realizados para verificar se estes resultados constituem um padrão, tornando necessária a revisão deste valor padrão para este tipo de nuvem de pontos.

O método DCBA se mostrou capaz de realizar o particionamento das nuvens de pontos de forma automatizada, obtendo resultados com superfícies suaves e com continuidade C^1 nas junções dos retalhos. Em casos específicos, no entanto, esta escolha automatizada pode dificultar ou tornar mais custoso o ajuste de determinados aspectos geométricos como quinas e pontas agudas, dependendo de como o particionamento é realizado. Idealmente, a representação de quinas seria melhor realizada com a junção de dois retalhos de superfície distintos com continuidade C^0 , sendo que a mesma representação com uma única superfície paramétrica exige a alteração na multiplicidade dos nós, ou a sobreposição de pontos de controle. Para estes casos, o método de particionamento poderia ser adaptado e melhorado para permitir este tipo de escolha.

Dentre os três métodos de partição apresentados e testados, nenhum deles se mostrou substancialmente melhor do que os demais. Assim como no caso das curvas, possivelmente o método de PMaC seria o mais indicado para a escolha adaptativa do nível de continuidade de junção, mas como nos casos testados apenas a continuidade C^1 foi escolhida, este aspecto não teve influência. No geral, as melhores métricas de erro foram obtidas com a utilização de PBS e de PMaC, com diferença pouco significativa em relação ao método de PMeC.

Com relação ao uso das métricas de erro AIC e BIC, percebeu-se que suas medidas são pouco influenciadas pela quantidade de pontos de controle quando comparada com a influência da métrica de erro para os casos testados. Apenas para ilustrar esta observação, considere os ajustes realizados para o *benchmark Pisot Triaxial*, no qual o método de PBS obteve $\epsilon^{[RMSE]} = 1.618 \cdot 10^{-2}$ com 308 pontos de controle e o método de PMaC obteve $\epsilon^{[RMSE]} = 7.9 \cdot 10^{-3}$ com 660 pontos de controle. Apesar do erro do ajuste pelo método de PBS ser pouco mais do que o dobro do ajuste pelo método de PMaC, mesmo se o ajuste pelo método PBS fosse realizado com apenas um ponto de controle, ambas as métricas AIC e BIC ainda seriam melhores para o ajuste utilizando PMaC, o que parece desprezar parcialmente a dificuldade em realizar ajustes com poucas variáveis de otimização.

Vale ainda observar que a utilização da variante de superfícies B-Spline bicúbicas, com princípios das superfícies de Gregory, foi capaz de promover uma suavidade adequada para as junções de superfície, resolvendo o problema da compatibilidade de torção. Entre os métodos encontrados na literatura científica, o método DCBA é o único que utiliza tais princípios para realizar este tipo de ajuste, combinado ainda com superfícies de diferentes dimensões. Conforme destacado durante o trabalho, uma continuidade C^2 seria mais adequada para conectar as superfícies bicúbicas empregadas, mas para fins de teste e validação da proposta, as conexões com continuidade C^1 já se mostraram suficientes para produzir soluções atrativas.

Por fim, assim como no caso do ajuste por múltiplas curvas, o método proposto para o

ajuste de múltiplas superfície passa a ser mais rápido do que um único ajuste tradicional a partir de uma determinada quantidade de variáveis. A diferença do crescimento assintótico do custo temporal entre o método tradicional e o método de ajuste por múltiplas superfícies pode ser observado na Figura 57. Além do método DCBA ser mais rápido a partir de uma determinada quantidade de pontos de controle utilizada no ajuste, ele também requer uma menor quantidade de memória para realizar o cálculo das soluções, quando comparado ao método tradicional. Isto pôde ser observado em um experimento utilizando uma nuvem de pontos composta por exatas 810 mil amostras, a qual não pôde ser ajustada pelo método tradicional por falta de memória, ao passo que o método DCBA foi capaz de realizar o ajuste.

6 CONCLUSÃO

O ajuste ou reconstrução de superfícies sob uma nuvem de pontos é um problema amplamente explorado na literatura científica, com diversas aplicações possíveis. No entanto, a complexidade desta tarefa torna difícil a obtenção de uma solução ótima, sendo comum a exploração por métodos que encontrem soluções subótimas, as quais representem adequadamente os objetos de interesse. Buscando por formas otimizadas de se representar tais superfícies, diversos métodos fazem uso das superfícies paramétricas, as quais são definidas por pontos de controle, funções de mistura e possivelmente outros elementos adicionais. Dentre estas superfícies, as B-Splines se destacam na literatura científica sendo uma das mais utilizadas.

O ajuste de superfícies paramétricas pode ser realizado de diversas formas, sendo comum a utilização de sistemas lineares e não lineares para determinar as variáveis envolvidas, ou soluções que utilizam meta-heurísticas. Habitualmente, a quantidade de pontos de controle é testada por tentativa e erro até se encontrar um ajuste com aspecto geométrico ou métrica de erro adequada. Mais recentemente, uma maior quantidade de trabalhos vem introduzindo abordagens de inserção e remoção de pontos de controle de maneira iterativa. Além disso, trabalhos ainda mais recentes já começam a explorar a utilização de múltiplos retalhos de superfície combinados.

Seguindo a tendência de ajuste de objetos baseado em uma solução que escolhe adaptativamente a quantidade de pontos de controle e ainda utilizando múltiplos retalhos de superfície, foi apresentado neste trabalho um novo método de ajuste de superfícies baseado em uma estratégia de divisão e conquista. O objetivo principal desta proposta foi validar a viabilidade, a qualidade e a eficiência de ajustes com múltiplas superfícies utilizando a estratégia de divisão e conquista. Primeiramente foi verificado e confirmado que o mesmo objetivo poderia ser alcançado através de uma proposta similar para o ajuste baseado em múltiplas curvas. Na sequência, os mesmos princípios e mecanismos foram estendidos e adaptados para o caso do ajuste de superfícies, aumentando um grau de liberdade (uma dimensão) e a complexidade da modelagem e do processamento dos dados. Para viabilizar a criação e combinação de superfícies em todas as bordas, resolvendo o problema da compatibilidade de torção, foram empregadas B-Splines bicúbicas com adição dos princípios das superfícies de Gregory, criando pontos de controle adicionais e orientados. O método proposto foi denominado como DCBA – *a Divide-and-Conquer algorithm for surface fitting with multi-Bspline-gregory-patches with Adaptive control points selection*.

O método DCBA é constituído por quatro etapas principais: procedimento iterativo de divisão e conquista, reparametrização, simplificação e combinação, além mecanismos de aceleração: explosão da nuvem de pontos, programação dinâmica, cálculo de erro de ajuste com utilização de limiar de erro e redução de sistemas lineares em junções. Todas as operações de ajuste são realizadas através do método de mínimos quadrados com a resolução de sistemas lineares, fixando os parâmetros de localização e do vetor de nós. Como não existe um algoritmo que determina uma posição ótima para particionar uma nuvem de pontos em dois subconjuntos considerando seu posterior ajuste, foram elaborados e testados três métodos de divisão: o

Particionamento por Bisseção (PBS), o Particionamento pela Maior Curvatura (PMaC) e o Particionamento pela Menor Curvatura (PMeC), todos baseados na repartição por reta horizontal ou vertical no espaço paramétrico – i.e. paralelas as direções associadas aos valores paramétricos u e v .

O método DCBA recebe como entrada apenas a nuvem de pontos e sua parametrização, mas permite o ajuste de diversos parâmetros de funcionamento, os quais viabilizam o refinamento dos resultados. Foram definidos valores padrões para estes parâmetros – alguns fixos e outros baseados em dados da própria nuvem de pontos – e foram indicados intervalos de valores possíveis e sugeridos para a adaptação dos parâmetros. Em especial, destaca-se a utilização de um limiar de erro tolerável $\epsilon_{tol}^{[\gamma]}$ baseado em uma métrica de erro γ , que é utilizado como principal critério para a otimização da quantidade de pontos de controle do ajuste. Todavia, o método não assegura que o resultado obtido terá uma métrica de erro $\epsilon^{[\gamma]}$ menor ou igual ao limiar estabelecido – i.e. não se garante $\epsilon^{[\gamma]} \leq \epsilon_{tol}^{[\gamma]}$.

Para testar o método proposto foram realizados oito experimentos com nuvens de pontos organizadas e três experimentos com nuvens de pontos não organizadas. Todas as superfícies geradas capturaram os aspectos geométricos dos objetos amostrados pelas nuvens de pontos, com exceção da representação de arestas de quinas, as quais foram suavizadas (pela natureza do algoritmo). Além da qualidade de ajustes obtidas, o método também se mostrou eficiente, principalmente ao se considerar que a quantidade de pontos de controle não é fornecida como entrada, mas ajustada durante o processo. Não foi verificada uma diferença significativa entre os resultados gerados pelos métodos de partição apresentados. Já em relação aos resultados encontrados na revisão de literatura, o método apresenta um dos melhores tempos de execução considerando os tamanhos das entradas utilizadas. O método, inclusive, pode ser mais rápido do que um único ajuste tradicional utilizando uma única superfície, quando o tamanho da nuvem é consideravelmente grande e é necessária uma grande quantidade de pontos de controle.

Baseado no estudo realizado, o método de ajuste proposto apresenta a versatilidade de escolher a quantidade de pontos de controle de forma adaptativa, com base em um limiar de erro aceitável; realiza segmentação automática da nuvem de pontos, sem necessidade da definição manual de bordas de limite; e é capaz de realizar o ajuste de superfície mesmo para nuvens com uma quantidade consideravelmente alta de amostras (pelo menos 810 mil pontos, conforme testado), devido ao seu mecanismo de divisão antes de ajustar.

Como resultados desta pesquisa pode-se destacar a elaboração de uma ferramenta de código aberto, disponibilizada e mantida em um repositório¹, que realiza os ajustes de curvas e superfícies apresentados, além de diversas operações que são utilizadas neste processo ou que estão de alguma forma vinculadas a ele. A ferramenta desenvolvida foi escrita em linguagem de programação Python, com utilização de diversas bibliotecas. Além disso, a pesquisa sobre o método de ajuste de múltiplas curvas foi publicado na revista *Computer-Aided Design* (BUCHINGER; ROSSO, 2022), na qual maiores detalhes podem ser obtidos sobre este método e

¹ Repositório do projeto *Reconstructor*: <<https://github.com/buchinger/Reconstructor>>

sobre os experimentos realizados.

Apesar dos resultados obtidos e das características notáveis do método DCBA, melhorias e estudos futuros podem ser realizados com base nesta pesquisa. A seguir é apresentada uma lista de possíveis trabalhos futuros que estendem ou adaptam funcionalidades do método proposto nesta pesquisa:

- Realizar a implementação de ajustes de combinação entre retalhos de superfície com continuidade C^2 . Isto pode ser feito utilizando os mesmos princípios já aplicados para a obtenção de uma continuidade C^1 , os quais foram apresentados nas Subseções 4.2.8.1 e 4.2.8.2. Deste modo, garante-se o mesmo nível de continuidade interna que os retalhos de superfície B-Splines bicúbicos fornecem através de sua estrutura;
- Buscar por alguma forma de permitir a geração de junções do tipo T, viabilizando a partição de um único subconjunto da nuvem de pontos por vez, sem a necessidade de particionar todo um grupo de subconjuntos, apenas para manter a correspondência estrutural de nós. Neste sentido, as T-Splines poderiam ser utilizadas, seja como uma única superfície global ou verificando a viabilidade e a eficiência na junção entre retalhos de superfícies T-Splines. Outra ramificação é a continuação dos estudos apresentados na Subseção 4.2.5.1, buscando por adaptações nas B-Splines que permitam a junção parcial com continuidade entre os retalhos de superfícies;
- Investigar com maior detalhamento o mecanismo de escolha adaptativa de continuidade. Para este trabalho, o mecanismo foi desconsiderado por ter escolhido apenas ajustes com continuidade C^1 , mas possivelmente esta escolha tenha sido afetada pelas linhas de partição escolhidas automaticamente ou pela característica das nuvens de pontos utilizadas. Para uma melhor escolha nesta adaptatividade automática do nível de continuidade entre retalhos, pode ser necessária também a elaboração de métodos de particionamento mais robustos, que consigam identificar regiões de quinas e arestas, realizando o particionamento nestas regiões. Isto é importante porque determinadas características geométrica, como regiões de quina e arestas, usualmente são representadas de forma mais adequada por junções com continuidade C^0 , não sendo adequado suavizá-las. Esta alteração, no entanto, demandaria uma análise do impacto sobre a parametrização utilizada;
- Investigar por soluções alternativas que evitem a necessidade de uma parametrização global inicial homeomórficas a um disco. Esta parametrização que é demandada como entrada para o algoritmo impossibilita o ajuste de superfícies com topologias não triviais, que não podem ser planificadas em forma de quadrado ou círculo. Além disso, o cálculo da parametrização para nuvens de pontos com grande quantidade de amostras pode exigir um tempo significativo. Soluções neste sentido podem ampliar a gama de aplicações do método proposto sem a necessidade de segmentação manual antes da execução do método;

- Analisar com maior profundidade os valores dos parâmetros utilizados e seu impacto sobre os resultados obtidos nos ajustes. Pode-se perceber através dos experimentos realizados que os parâmetros escolhidos provavelmente não correspondem a boas escolhas, especialmente para o caso de nuvens não organizadas, nas quais o erro aceitável não pode ser atingido com as restrições impostas sobre a divisão e conquista. Nesta análise, diversos *benchmarks* e valores de parâmetros podem ser utilizados e comparados para verificar se existem valores padrões que podem ser utilizados para gerar melhores resultados de um modo geral;
- Utilizar superfícies NURBS ou outras superfícies paramétricas, testando quanto os resultados podem ser melhorados com a otimização de outros parâmetros da superfície, como os pesos e os vetores de nós. Essa otimização pode ser feita apenas no final do método, como uma última etapa, ou pode ser incluída em uma ou mais etapas durante os procedimentos que constituem o método;
- Implementar as principais operações do método com paralelismo. Considerando que diversas operações utilizadas são altamente paralelizáveis – por exemplo, os procedimentos de conquista, reparametrização e simplificação podem ser executados em paralelo; os procedimentos de divisão e combinação também podem ser executados em paralelo, mas requerem maiores cuidados com a divisão das operações e seu sincronismo. Esta modificação pode reduzir substancialmente o tempo de execução total do método;
- Realizar uma análise de possíveis melhorias para as operações específicas do método, em especial, aquelas que demandam mais tempo de execução, como as operações de combinação, e em alguns casos as operações de simplificação. Deste modo, pode-se obter uma redução significativa no tempo de execução necessário para os ajustes.

REFERÊNCIAS

- AFSHAR, Nejat A.; SORYANI, Mohsen; RAHMANI, Adel T. Curve fitting using coevolutionary genetic algorithms. In: PANIGRAHI, B. K. et al. (Ed.). **Swarm, Evolutionary, and Memetic Computing (SEMCCO 2011). Lecture Notes in Computer Science, vol 7077.** [S.I.]: Springer-Verlag Berlin Heidelberg, 2011. p. 201–210. ISBN 978-3-642-27242-4.
- AGUILAR, Enrique et al. An adaptive curvature-guided approach for the knot-placement problem in fitted splines. **Journal of Computing and Information Science in Engineering**, v. 18, n. 4, Paper No: JCISE-15-1390, p. 9, 2018.
- ALEXA, Marc et al. Computing and rendering point set surfaces. **IEEE Transactions on Visualization and Computer Graphics**, v. 9, n. 1, p. 3–15, 2003.
- AUTODESK. **Continuity 1: G0, G1, G2, G3.** 2017. Disponível em: <<http://help.autodesk.com/view/ALIAS/2017/ENU/?guid=GUID-E1BDFBD0-33CC-44C4-866D-5F367105A050>>. Acesso em: 07 abr. 2022.
- AUTODESK. **From Solids to Organics – when and how to use Solids, Surfaces, and T-Splines.** 2020. Disponível em: <https://www.autodesk.com/products/fusion-360/blog/from-solids-to-organics-when-and-how-to-use-solids-surfaces-and-t-splines/?us_oa=dotcom-us&us_si=81c4e66f-bf3d-495e-a54d-ffa2f4e5b6c0&us_st=tsplines>. Acesso em: 14 abr. 2022.
- AVRON, Haim et al. ℓ_1 -sparse reconstruction of sharp point set surfaces. **ACM Transactions on Graphics (TOG)**, v. 29, n. 5, p. 12, oct. 2010.
- BARHAK, J.; FISCHER, A. Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques. v. 7, n. 1, p. 1–16, 2001.
- BÉNIÈRE, Roseline et al. A comprehensive process of reverse engineering from 3D meshes to cad models. **Computer-Aided Design**, v. 45, n. 11, p. 1382–1393, 2013.
- BERGER, Matthew et al. A benchmark for surface reconstruction. **ACM Transactions on Graphics (TOG)**, v. 32, n. 2, p. 17, 2013.
- BERGER, Matthew et al. State of the art in surface reconstruction from point clouds. In: LEFEBVRE, Sylvain; SPAGNUOLO, Michela (Ed.). **Eurographics 2014 - State of the Art Reports.** Strasbourg, France: The Eurographics Association, 2014. p. 25. ISSN 1017-4656.
- BERGER, Matthew et al. A survey of surface reconstruction from point clouds. **Computer Graphics Forum**, v. 36, n. 1, p. 301–329, 2017.
- BERNARDINI, F. et al. The ball-pivoting algorithm for surface reconstruction. **IEEE Transactions on Visualization and Computer Graphics**, v. 5, n. 4, p. 349–359, oct. 1999. ISSN 1077-2626.
- BERTOLINO, Giulia et al. An efficient hybrid optimization strategy for surface reconstruction. **Computer Graphics Forum**, v. 40, n. 6, p. 215–241, sep. 2021.
- BOISSONNAT, J.D.; CHAZAL, F.; YVINEC, M. **Geometric and Topological Inference.** [S.I.]: Cambridge University Press, 2017. 300 p.

- BOTSCH, Mario et al. **Polygon Mesh Processing**. 1st. ed. Massachusetts: AK Peters/CRC Press, 2010. 250 p. ISBN 978-1-56881-426-1.
- BRADLEY, Derek; NOWROUZEZAHRAI, Derek; BEARDSLEY, Paul. Image-based reconstruction and synthesis of dense foliage. **ACM Transactions on Graphics (TOG)**, v. 32, n. 4, p. 10, jul. 2013.
- BUCHINGER, D.; ROSSO, R.S.U. A divide-and-conquer algorithm for curve fitting. **Computer-Aided Design**, v. 151, n. Article No. 103362, p. 17, 2022. ISSN 0010-4485.
- CALLIERI, M. et al. VCLab's tools for 3D range data processing. In: **Proceedings of the 4th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST'03)**. Brighton, United Kingdom: [s.n.], 2003. p. 13–22. ISBN 3-905673-08-8.
- CARLSON, Nils; GULLIKSSON, Mårten. Surface fitting with NURBS: A gauss newton with trust region approach. In: **13th WSEAS International Conference on Applied Mathematics**. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2008. p. 169–174. ISBN 9789604740345.
- CATMULL, Edwin; ROM, Raphael. A class of local interpolating splines. **Computer Aided Geometric Design**, Elsevier Inc., p. 317–326, 1974.
- CHIYOKURA, Hiroaki; KIMURA, Fumihiko. Design of solids with free-form surfaces. **ACM SIGGRAPH Computer Graphics**, v. 17, p. 289–298, jul. 1983.
- CHOI, Sungjoon et al. A large dataset of object scans. **Computing Research Repository (CoRR)**, abs/1602.02481, 2016. Disponível em: <<http://arxiv.org/abs/1602.02481>>.
- CIGNONI, Paolo et al. Meshlab: an open-source mesh processing tool. In: **Sixth Eurographics Italian Chapter Conference**. Salerno, Italy: The Eurographics Association, 2008. p. 129 – 136. ISBN 978-3-905673-68-5.
- CIGNONI, Paolo; ROCCHINI, Claudio; SCOPIGNO, Roberto. Metro: Measuring error on simplified surfaces. **Computer Graphics Forum**, v. 17, n. 2, p. 167 – 174, nov. 1998.
- CORMEN, T.H. et al. **Introduction to Algorithms**. 3rd. ed. [S.l.]: MIT Press and McGraw-Hill, 2009. ISBN 0-262-03384-4.
- COSTA, Giulio; MONTEMURRO, Marco; PAILHÈS, Jérôme. A general hybrid optimization strategy for curve fitting in the non-uniform rational basis spline framework. **Journal of Optimization Theory and Applications**, v. 176, n. 1, p. 225–251, jan. 2018.
- ĆURKOVIĆ, Milan; ĆURKOVIĆ, Andrijana; VUČINA, Damir. Novel re-parameterization for shape optimization and comparison with knot-based gradient fitting method. **Computer Methods in Applied Mechanics and Engineering**, v. 336, p. 304–332, 2018.
- DESBRUN, Mathieu; MEYER, Mark; ALLIEZ, Pierre. Intrinsic parameterizations of surface meshes. In: **Proceedings of Eurographics 2002, Computer Graphics forum**. Saarbrücken, Germany: Wiley, 2003. p. 209–218.
- DUNG, Van Than; TJAHIJOWIDODO, Tegoeh. A direct method to solve optimal knots of B-spline curves: An application for non-uniform B-spline curves fitting. **PLoS ONE**, v. 12, n. 3, p. 24, e0173857, mar. 2017.

EBRAHIMI, Alireza; LOGHMANI, Ghasem. B-spline curve fitting by diagonal approximation BFGS methods. **Iranian Journal of Science and Technology, Transactions A: Science**, v. 43, n. 1, p. 12, dez. 2017.

EBRAHIMI, A.; LOGHMANI, G. B. Shape modeling based on specifying the initial B-Spline curve and scaled BFGS optimization method. **Multimedia Tools and Applications**, v. 77, n. 23, p. 30331–30351, dec. 2018.

EDELSBRUNNER, H.; KIRKPATRICK, D.; SEIDEL, R. On the shape of a set of points in the plane. **IEEE Transactions on Information Theory**, v. 29, n. 4, p. 551–559, jul. 1983.

EILERS, Paul H. C.; MARX, Brian D. Flexible smoothing with B-splines and penalties. **Statistical Science**, v. 11, n. 2, p. 89–121, 1996.

FENG, Chen; TAGUCHI, Yuichi. FasTFit: A fast T-spline fitting algorithm. **Computer-Aided Design**, v. 92, p. 11–21, Nov. 2017.

FLOATER, Michael S. Parametrization and smooth approximation of surface triangulations. **Computer Aided Geometric Design**, v. 14, n. 3, p. 231–250, apr. 1997.

FLOATER, Michael S. Mean value coordinates. **Computer Aided Geometric Design**, v. 20, n. 1, p. 19–27, mar. 2003.

FLOATER, Michael S.; HORMANN, Kai. Surface parameterization: a tutorial and survey. **Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization**, Springer, p. 157–186, 2005.

FLOATER, Michael S.; KÓS, Géza; REIMERS, Martin. Mean value coordinates in 3D. **Computer Aided Geometric Design**, v. 22, n. 7, p. 623–631, oct. 2005.

FORSEY, David R.; BARTELS, Richard H. Hierarchical B-Spline refinement. **ACM SIGGRAPH Computer Graphics Homepage**, v. 22, n. 4, p. 205–212, aug. 1988.

FRÉDÉRIC, Cazals; JOACHIM, Giesen. **Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms**. RR-5393, INRIA - Institut National De Recherche En Informatique Et En Automatique, 2004. 42 p.

GABA, Eric (Sting). **Minimal surface curvature planes**. 2006. Disponível em: <<https://commons.wikimedia.org/w/index.php?curid=1325452>>. Acesso em: 22 jul. 2022.

GÁLVEZ, Akemi; IGLESIAS, Andrés. Firefly algorithm for explicit B-Spline curve fitting to data points. In: **Mathematical Problems in Engineering**. [S.l.]: Hindawi Publishing Corporation, 2013. v. 2013, n. Article ID 528215, p. 12.

GÁLVEZ, Akemi; IGLESIAS, Andrés. Particle-based meta-model for continuous breakpoint optimization in smooth local-support curve fitting. **Applied Mathematics and Computation**, v. 275, n. 15, p. 195–212, feb. 2016.

GÁLVEZ, Akemi; IGLESIAS, Andrés; PUIG-PEY, Jaime. Iterative two-step genetic-algorithm-based method for efficient polynomial B-Spline surface reconstruction. **Information Sciences**, v. 182, n. 1, p. 56–76, jan. 2012.

- GIANNELLI, Carlotta; JÜTTLER, Bert; SPELEERS, Hendrik. THB-splines: The truncated basis for hierarchical splines. **Computer Aided Geometric Design**, v. 29, n. 7, p. 485–498, oct. 2012.
- GOLUB, G. Numerical methods for solving linear least squares problems. **Numerische Mathematik**, v. 7, n. 3, p. 206–216, jun. 1965.
- GOLUB, Gene H.; LOAN, Charles F. Van. **Matrix Computations**. 3rd. ed. London: The Johns Hopkins Press Ltd., 1996. 694 p. ISBN 0-8018-5413-8.
- GONZALEZ, Teofilo F. **Handbook of Approximation Algorithms and Metaheuristics**. first. [S.I.]: Chapman & Hall/CRC - Taylor & Francis Group, 2007. 1432 p.
- GRAY, Alfred; ABBENA, Elsa; SALAMON, Simon. **Modern Differential Geometry of Curves and Surfaces with Mathematica**. 3rd. ed. [S.I.]: Chapman and Hall/CRC, 2006. 1016 p. ISBN 978-1584884484.
- GREGORY, John A. Smooth interpolation without twist constraints. **Computer Aided Geometric Design**, p. 71–87, 1974.
- GREINER, Günther; HORMANN, Kai. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-Splines. In: **Surface Fitting and Multiresolution Methods**. [S.I.]: Vanderbilt University Press, 1997. p. 163–172.
- HAAR, Frank Ter et al. A comparison of systems and tools for 3D scanning. In: **3D Digital Imaging and Modeling: Applications of Heritage, Industry, Medicine and Land**. Workshop Italy-Canada: Session P.12: [s.n.], 2005. p. 8.
- HASEGAWA, Allan Yoshio; ROSSO, Roberto Silvio Ubertino; TSUZUKI, Marcos Sales Guerra. Bézier curve fitting with a parallel differential evolution algorithm. **IFAC Proceedings Volumes**, v. 46, n. 7, p. 233–238, may 2013.
- HASEGAWA, Allan Yoshio; TORMENA, Camila; PARPINELLI, Rafael Stubs. Bézier curve parametrization using a multiobjective evolutionary algorithm. **International Journal of Computer Science and Applications**, v. 11, n. 2, p. 18, 2014.
- HENLE, Michael. **A Combinatorial Introduction to Topology**. 3rd. ed. New York: Dover, 1994. 110 p. ISBN 0486679667.
- HERMANN, T. G^2 interpolation of free form curve networks by biquintic gregory patches. **Computer Aided Geometric Design**, v. 13, p. 873–893, dec. 1996.
- HETTINGA, Gerben J.; KOSINKA, Jirí. Multisided generalisations of gregory patches. **Computer Aided Geometric Design**, v. 62, p. 166–180, may 2008.
- HOPPE, Hugues et al. Surface reconstruction from unorganized points. **ACM SIGGRAPH Computer Graphics**, v. 26, n. 2, p. 71–78, jul. 1992.
- IGLESIAS, Andrés; GÁLVEZ, Akemi. Hybrid functional-neural approach for surface reconstruction. **Mathematical Problems in Engineering**, v. 2014, n. 1, p. 13, 2014.
- IGLESIAS, Andrés; GÁLVEZ, Akemi; COLLANTES, Marta. Iterative sequential bat algorithm for free-form rational bézier surface reconstruction. **International Journal of Bio-Inspired Computation**, v. 11, n. 1, p. 15, Jan. 2018.

- IGLESIAS, Andrés; GÁLVEZ, Akemi; LOUCERA, Carlos. A simulated annealing approach for data fitting with bézier surfaces. In: **Proceedings of the International Conference on Intelligent Information Processing (IPAC '15), Security and Advanced Communication.** New York, NY, USA: Association for Computing Machinery, 2015. p. 6. ISBN 9781450334587.
- IGLESIAS, Andrés et al. Cuckoo search algorithm with lévy flights for global-support parametric surface approximation in reverse engineering. **Symmetry**, v. 10, n. 3, p. 25, 2018.
- IRSHAD, Misbah et al. Outline capturing using rational functions with the help of genetic algorithm. **Applied Mathematics and Computation**, v. 274, n. 1, p. 661–678, feb. 2016.
- JANCOSEK, Michal; PAJDLA, Tomas. Multi-view reconstruction preserving weakly-supported surfaces. In: **IEEE Computer Vision and Pattern Recognition (CVPR) 2011.** Colorado Springs, USA: [s.n.], 2011. p. 3121–3128. Disponível em: <<https://doi.org/10.1109/cvpr.2011.5995693>>.
- JAVIDRAD, F. An accelerated simulated annealing method for B-Spline curve fitting to strip shaped scattered points. **International Journal of CAD/CAM (IJCC)**, v. 12, n. 1, p. 11, 2012.
- KAZHDAN, Michael. Reconstruction of solid models from oriented point sets. In: **Proceedings of the third Eurographics symposium on Geometry processing (SGP '05).** Vienna, Austria: [s.n.], 2005. p. 11. ISBN 3-905673-24-X. Artigo No. 73.
- KAZHDAN, Michael; BOLITHO, Matthew; HOPPE, Hugues. Poisson surface reconstruction. In: **Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP'06).** Sardinia, Italy: [s.n.], 2006. p. 61–70. ISBN 3-905673-36-3.
- KINERI, Yuri et al. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. **Computer-Aided Design**, v. 44, n. 7, p. 697–708, jul. 2012.
- KOCHENDERFER, Mykel J.; WHEELER, Tim A. **Algorithms for Optimization.** 1st. ed. [S.I.]: The MIT Press – Cambridge, Massachusetts, 2019. 520 p. ISBN 9780262039420.
- KRAGIĆ, Ivo Marinic; ĆURKOVIĆ, Milan; VUČINA, Damir. Adaptive re-parameterization based on arbitrary scalar fields for shape optimization and surface fitting. **Engineering Applications of Artificial Intelligence**, v. 67, p. 39–51, 2018.
- LAI, Kevin et al. A large-scale hierarchical multi-view RGB-D object dataset. In: **Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).** [S.I.: s.n.], 2011. p. 1817–1824. ISBN 978-1-61284-385-8.
- LAI, Yu-Kun; HU, Shi-Min; POTTMANN, Helmut. Surface fitting based on a feature sensitive parametrization. **Computer-Aided Design**, v. 38, n. 7, p. 800–807, jul. 2006.
- LANKVELD, Thijs van; KREVELD, Marc Van; VELTKAMP, Remco C. Watertight scenes from urban lidar and planar surfaces. **Computer Graphics Forum**, v. 32, n. 5, p. 217–228, 2013.
- LAURENT-GENGOUX, Pascal; MEKHILEF, Mounib. Optimization of a NURBS representation. **Computer-Aided Design**, v. 25, n. 11, p. 699–710, nov. 1993.
- LEE, E.T.Y. Choosing nodes in parametric curve interpolation. **Computer-Aided Design**, v. 21, n. 6, p. 363–370, jul./aug. 1989.

- LEE, Thomas. On algorithms for ordinary least squares regression spline fitting: A comparative study. **Journal of Statistical Computation and Simulation**, v. 72, n. 8, p. 647–663, oct. 2002.
- LEON, Steven J. **Algebra linear com aplicações**. 9th. ed. Rio de Janeiro: LTC, 2018. Disponível em: <[https://app.minhabiblioteca.com.br/reader/books/9788521635789/epubcfi/6/38\[%3Bvnd.vst.idref%3Dbm03\]!/](https://app.minhabiblioteca.com.br/reader/books/9788521635789/epubcfi/6/38[%3Bvnd.vst.idref%3Dbm03]!/)>. ISBN 9788521635789 (eletrônico).
- LEVOY, Marc et al. The digital michelangelo project: 3D scanning of large statues. In: **Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)**. Louisiana, USA: [s.n.], 2000. p. 131–144. ISBN 1-58113-208-5.
- LI, Yusha et al. Surface skinning using periodic T-spline in semi-NURBS form. **Journal of Computational and Applied Mathematics**, v. 273, p. 116–131, jan. 2015.
- LI, Yangyan et al. Globfit: consistently fitting primitives by discovering global relations. **ACM Transactions on Graphics (TOG)**, v. 30, n. 4, p. 12, jul. 2011.
- LOOP, Charles et al. Approximating subdivision surfaces with gregory patches for hardware tessellation. **ACM Transactions on Graphics**, v. 28, p. 1–9, dec. 2009.
- LUKE, Sean. **Essentials of Metaheuristics**. second. [S.I.]: Lulu, 2013. 261 p. Disponível em: <<http://cs.gmu.edu/~sean/book/metaheuristics/>>.
- MADSEN, Kaj; NIELSEN, Hans Bruun; TINGLEFF, Ole. **Methods for Non-Linear Least Squares Problems**. 2nd. ed. [S.I.]: Informatics and Mathematical Modelling (IMM), Technical University of Denmark, 2004.
- MANSON, J.; PETROVA, G.; SCHAEFER, S. Streaming surface reconstruction using wavelets. In: **Proceedings of the Symposium on Geometry Processing (SGP '08)**. Copenhagen, Denmark: [s.n.], 2008. v. 27, p. 1411–1420.
- MAO, Qing et al. Surface fitting for quasi scattered data from coordinate measuring systems. **Sensors**, v. 18, n. 1: 214, p. 26, jan. 2018.
- MIYAZAKI, Lucas K. et al. Teeth mesh segmentation through curvature analysis. In: CHENG, Liang-Yee (Ed.). **ICGG 2020 - Proceedings of the 19th International Conference on Geometry and Graphics**. Cham: Springer International Publishing, 2021. p. 235–243. ISBN 978-3-030-63403-2.
- MORTENSON, M.E. **Geometric Modeling**. 3rd. ed. New York: Industrial Press Inc., 2006. 505 p. ISBN 0-831-13298-1.
- MOULON, Pierre; MONASSE, Pascal; MARLET, Renaud. Adaptive structure from motion with a contrario model estimation. In: **Proceedings of the Asian Computer Vision Conference (ACCV 2012)**. [S.I.]: Springer Berlin Heidelberg, 2012. p. 257 – 270.
- NOCEDAL, Jorge; WRIGHT, Stephen J. **Numerical Optimization**. 2nd. ed. [S.I.]: Springer, 2006. ISBN 0-387-30303-0.
- OHTAKE, Yutaka et al. Multi-level partition of unity implicits. **ACM Transactions on Graphics (TOG)**, v. 22, n. 3, p. 463–470, jul. 2003.

- PANDUNATA, Priza; SHAMSUDDIN, Siti Mariyam Hj. Differential evolution optimization for bezier curve fitting. In: **Seventh International Conference on Computer Graphics, Imaging and Visualization**. Sydney, Australia: IEEE, 2010. p. 68–72. ISBN 978-1-4244-7840-8.
- PARK, Hyungjun; LEE, Joo-Haeng. B-spline curve fitting based on adaptive curve refinement using dominant points. **Computer-Aided Design**, v. 39, n. 6, p. 439–451, jun. 2007.
- PAULY, Mark et al. Example-based 3D scan completion. In: **Proceedings IEEE of the third Eurographics symposium on Geometry processing (SGP'05)**. Vienna, Austria: [s.n.], 2005. p. 10. ISBN 3-905673-24-X.
- PIEGL, Les; TILLER, Wayne. **The NURBS Book**. 2nd. ed. Berlin, Heidelberg: Springer-Verlag Berlin, Heidelberg, 1997. 646 p. ISBN 3-540-61545-8.
- POTTMANN, H.; LEOPOLDSEDER, S.; HOFER, M. Approximation with active B-Spline curves and surfaces. In: **Proceedings of the 10th pacific conference on computer graphics and applications**. Beijing, China: IEEE, 2002. p. 8–25. ISBN 0-7695-1784-6.
- POTTMANN, H. et al. Industrial geometry: recent advances and applications in CAD. **Computer-Aided Design**, v. 37, n. 7, p. 751–766, jun. 2005.
- REMACLE, J-F. et al. High-quality surface remeshing using harmonic maps. **International Journal for Numerical Methods in Engineering**, v. 83, n. 4, p. 403–425, jul. 2010.
- RODRÍGUEZ, J. Apolinar Muñoz. Efficient NURBS surface fitting via GA with SBX for free-form representation. **International Journal of Computer Integrated Manufacturing**, v. 30, n. 9, p. 981–994, 2017.
- RUSU, Radu Bogdan; COUSINS, Steve. 3D is here: Point cloud library (PCL). In: **IEEE International Conference on Robotics and Automation (ICRA)**. Shanghai, China: [s.n.], 2011. p. 4. ISBN 978-1-61284-385-8.
- SAMET, Hanan. The quadtree and related hierarchical data structures. **Computing Surveys**, v. 16, n. 2, p. 187 – 260, jun. 1984.
- SARFRAZ, Muhammad; HUSSAIN, Malik Zawwar; IRSHAD, Misbah. Reverse engineering of the digital curve outlines using genetic algorithm. **International Journal of Computers**, v. 7, p. 10, 2013.
- SARFRAZ, Muhammad et al. Computing optimized NURBS curves using simulated evolution on control parameters. In: TIWARI, A. et al. (Ed.). **Applications of Soft Computing, vol. 36. Advances in Intelligent and Soft Computing**. Heidelberg: Springer, Berlin, 2006. p. 35–44.
- SEDERBERG, Thomas W. **System and method for defining T-spline and T-NURCC surfaces using local refinements**. 2003. Procurador: Brigham Young University. US7274364B2. Depósito: 17 Mar. 2004. Concessão: 5 Sep. 2007. Disponível em: <<https://patents.google.com/patent/US7274364B2/en>>. Acesso em: 14 abr. 2022.
- SEDERBERG, Thomas W. et al. T-splines and T-NURCCs. **ACM Transactions on Graphics (TOG)**, v. 22, n. 3, p. 477–484, jul. 2003.
- SEDERBERG, Thomas W. et al. Non-uniform recursive subdivision surfaces. In: **Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98)**. Orlando, USA: ACM New York, 1998. p. 387–394. ISBN 0-89791-999-8.

SHAMSUDDIN, S. M.; AHMED, M. A.; SAMIAN, Y. NURBS skinning surface for ship hull design based on new parameterization method. **The International Journal of Advanced Manufacturing Technology**, v. 28, p. 936–941, Jul. 2006.

SHANG, Ce et al. Closed T-Spline surface reconstruction from medical image data. **International Journal of Precision Engineering and Manufacturing**, v. 19, n. 11, p. 1659–1671, Nov. 2018.

SHAO, Lejun; ZHOU, Hao. Curve fitting with b閦ier cubics. **Graphical Models and Image Processing**, v. 58, n. 3, p. 223–232, may 1996.

SINGH, Arjun et al. Bigbird: A large-scale 3D database of object instances. In: **Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2014. p. 509–516. ISBN 978-1-4799-3685-4.

SPEER, Thomas; KUPPE, Markus; HOSCHEK, Josef. Global reparametrization for curve approximation. **Computer Aided Geometric Design**, v. 15, n. 9, p. 869–877, oct. 1998.

STROUD, I. **Boundary Representation Modelling Techniques**. 1st. ed. London: Springer London, 2006. 498 p. ISBN 1-84628-312-4.

SÜBMUTH, J.; MEYER, Q.; GREINER, G. Surface reconstruction based on hierarchical floating radial basis functions. **Computer Graphics Forum**, v. 29, n. 6, p. 1854–1864, sep. 2010.

TAGLIASACCHI, Andrea; ZHANG, Hao; COHEN-OR, Daniel. Curve skeleton extraction from incomplete point cloud. **ACM Transactions on Graphics (TOG)**, v. 28, n. 3, p. 10, aug. 2009.

TEH, C.-H.; CHIN, R. T. On the detection of dominant points on digital curves. **IEEE Transactions on pattern analysis and machine intelligence**, v. 11, p. 859–872, aug. 1989.

UEDA, E.K et al. Curve approximation by adaptive neighborhood simulated annealing and piecewise b閦ier curves. **Soft Computing**, v. 24, p. 18821–18839, dec. 2020.

ÜNSALAN, Cem; ERÇİL, Aytül. Conversion between parametric and implicit forms for computer graphics and vision. In: **Proceedings of the International Symposium on Computer and Information Sciences (ISCIS'99)**. Kusadasi, Turkey: [s.n.], 1999. p. 916–923.

VÁRADY, Tamás; MARTIN, Ralph R.; COX, Jordan. Reverse engineering of geometric models - an introduction. **Computer Aided Design**, v. 29, n. 4, p. 255–268, 1997.

WANG, Wenping; POTTMANN, Helmut; LIU, Yang. Fitting B-Spline curves to point clouds by curvature-based squared distance minimization. **ACM Transactions on Graphics (TOG)**, v. 25, n. 2, p. 214–238, apr. 2006.

WANG, Yimin; ZHENG, Jianmin. Curvature-guided adaptive T-spline surface fitting. **Computer-Aided Design**, v. 45, n. 8-9, p. 1095–1107, ago. 2013.

XIE, Hui; QIN, Hong. Automatic knot determination of NURBS for interactive geometric design. In: **International Conference on Shape modeling and applications (SMI 2001)**. Genova, Italy: IEEE, 2001. p. 267–276. ISBN 0-7695-0853-7.

YAN, Feilong et al. Proactive 3D scanning of inaccessible parts. **ACM Transactions on Graphics (TOG)**, v. 33, n. 4, p. 8, jul. 2014.

- YANG, Huaiping; WANG, Wenping; SUN, Jiaguang. Control point adjustment for B-Spline curve approximation. **Computer-Aided Design**, v. 36, n. 7, p. 639–652, jun. 2004.
- YIN, Kangxue et al. Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control. **ACM Transactions on Graphics (TOG)**, v. 33, n. 6, p. 12, nov. 2014.
- YOSHIMOTO, Fujiichi; HARADA, Toshinobu; YOSHIMOTO, Yoshihide. Data fitting with a spline using a real-coded genetic algorithm. **Computer-Aided Design**, v. 35, n. 8, p. 751–760, jul. 2003.
- YOSHIMOTO, F.; MORIYAMA, M.; HARADA, T. Automatic knot placement by a genetic algorithm for data fitting with a spline. In: **International Conference on Shape Modeling and Applications**. Aizu-Wakamatsu, Japan: IEEE, 1999. p. 8.
- ZACH, Christopher; POCK, Thomas; BISCHOF, Horst. A globally optimal algorithm for robust TV-L¹ range image integration. In: **Proc. IEEE 11th International Conference on Computer Vision (ICCV'07)**. Rio de Janeiro, Brasil: [s.n.], 2007. p. 8.
- ZHANG, Hao; XU, Feng. Mixed fusion: Real-time reconstruction of an indoor scene with dynamic objects. **IEEE Transactions on Visualization and Computer Graphics (TVCG)**, v. 24, n. 12, p. 3137–3146, dec. 2018.
- ZHANG, Yuhua et al. B-spline surface fitting with knot position optimization. **Computers & Graphics**, v. 58, n. Special Issue on SMI 2016, p. 73–83, aug. 2016.
- ZHENG, Wenni et al. Fast B-Spline curve fitting by L-BFGS. **Computer Aided Geometric Design**, v. 29, n. 7, p. 448–462, oct. 2012.
- ZOLLHÖFER, Michael et al. State of the art on 3D reconstruction with RGB-D cameras. **EUROGRAPHICS 2018 - STAR - State of The Art Report**, v. 37, n. 2, p. 28, 2018.

APÊNDICE A – SÍNTESE DA REVISÃO DE FERRAMENTAS E BIBLIOTECAS PARA RECONSTRUÇÃO

A fim de consultar as principais ferramentas e bibliotecas vinculadas ao processo de ajuste e reconstrução, foi realizada uma busca em três dos principais mecanismos de buscas da internet – Bing, Google e Yahoo – sendo utilizada as seguintes palavras chaves (sem o uso de aspas): "Point cloud 3D reconstruction Software". Foram então analisados os cinquenta primeiros resultados retornados por cada mecanismo de busca através da leitura do resumo de conteúdo e de varreduras na página de destino. É apresentada a seguir uma listagem ordenada alfabeticamente das ferramentas encontradas, trazendo uma síntese das principais características e funcionalidades destacadas no material informativo que foi coletado sobre elas.

3DF Zephyr¹: um software de engenharia reversa desenvolvido pela 3DFlow que permite a obtenção de nuvens de pontos através de fotogrametria e técnicas de escaneamento com laser. Este software utiliza um algoritmo proprietário de geração de malha através de uma nuvem de pontos, denominado de 3DF Sasha, o qual não é detalhado pelos desenvolvedores na página web. Existem quatro versões distintas que são oferecidas aos usuários: *Free*, *Lite*, *Pro* e *Aerial*. A primeira é uma versão gratuita que possui as principais funcionalidades com algumas limitações (e.g. limite de imagens, menos funcionalidades de edição e exportação), ao passo que as demais ferramentas são contempladas de maneira incremental com mais funcionalidades e menos limitações, comercializadas através de licenças perpétuas. Por fim, o software permite a realização de processamento utilizando GPU e conta com uma versão educacional gratuita com todas as funcionalidades mas, que não pode ser utilizada para propósitos comerciais;

3DReshaper²: um software desenvolvido pela Technodigit (Hexagon), a qual dispõe de soluções para processamento de nuvens de pontos (e.g. captura, filtragem e segmentação), modelagem 3D, engenharia reversa, controle e inspeção 3D, entre outras funcionalidades. A ferramenta possui recursos e *plugins* voltados para diversos domínios de aplicação como arquitetura, engenharia civil e modelagem de terreno. Existem três versões oferecidas: *3DReshaper* (padrão), *3DReshaper Meteor* e *3DReshaper SDK*. A primeira versão disponibiliza as principais funcionalidades da ferramenta com o aspecto multidomínio. A segunda versão possui um enfoque em metrologia, complementado com ferramentas de inspeção 3D e CAD, permitindo a criação de malhas mais precisas, a importação de dados provindos de máquinas de tomografia computadorizada e uma interface que permite a transferência dos dados diretamente da ferramenta de digitalização para o programa. Ambas as versões permitem automatizar tarefas repetitivas através da execução de scripts (Javascript). Já na terceira versão, além de todos os instaladores do software, é fornecido

¹ 3DF Flow homepage: <<https://www.3dflow.net>>

² 3DReshaper homepage: <<https://www.3dreshaper.com/en/>>

também o código fonte da ferramenta (em C++) para que as funcionalidades possam ser adaptadas ou embutidas em outro software. Todas as versões do 3DReshaper são voltadas ao sistema operacional Windows e são pagas, mas existe uma versão de teste gratuita que pode ser utilizada por um mês com algumas limitações. As funcionalidades principais da ferramenta são comercializadas por uma licença básica que pode ser suplementada com extensões que são compradas individualmente. Dentre estas, destaca-se a extensão que permite a criação de BSplines e NURBS para os modelos 3D e a extensão que permite exportar os modelos no formato IGES e STEP;

Agisoft PhotoScan³: software de fotogrametria que, além de gerar malhas texturizadas para modelos 3D, também permite a edição e classificação de elementos das nuvens de pontos, a mensuração de distâncias, áreas e volumes, a criação de scripts (em Python) para a automatização de tarefas, a distribuição do processamento através de uma rede de computadores, e a capacidade de tratar cenas dinâmicas (4D). O PhotoScan possibilita aceleração no processamento utilizando GPU, mas não gera superfícies suaves, apenas malha triangular. É um software multiplataforma e pago – tanto a versão regular como a educacional, mas possui também uma versão de teste válida por 30 dias e uma versão de demonstração limitada sem validade;

Alice Vision⁴: um *framework* europeu desenvolvido como resultado de uma parceria entre academia e indústria, englobando vários grupos de pesquisa de universidades e empresas, com o intuito de prover software com algoritmos de visão computacional estado da arte. Trata-se de uma iniciativa de código aberto para desenvolvimento de software e bibliotecas para fotomodelagem, rastreamento de câmera e reconstrução 3D. Dentre os trabalhos publicados, existe o projeto *Meshroom* que é voltado especialmente para o problema de engenharia reversa, gerando nuvens de pontos através de fotogrametria e permitindo a reconstrução de malhas texturizadas a partir de tais nuvens, as quais podem ser exportadas para softwares de modelagem 3D (JANCOSEK; PAJDLA, 2011; MOULON; MONASSE; MARLET, 2012). Além de ser um projeto de código aberto, o grupo desenvolvedor do *framework* *Alice Vision* conta com uma abordagem de contribuição livre ao projeto utilizando um repositório Git. O projeto *Meshroom* oferece versões pré-compiladas para os sistemas operacionais Windows (64 bits) e Linux. O *framework*, entretanto, foi desenvolvido exclusivamente para hardware gráfico da NVIDIA, permitindo aceleração no processamento utilizando GPU, mas requer uma placa gráfica com tecnologia CUDA (com *compute capability* ≥ 2.0);

CGAL⁵: CGAL é um projeto de código aberto e multiplataforma que provê a implementação de diversos algoritmos e estruturas de dados geométricas escritas em forma de bibli-

³ Agisoft PhotoScan *homepage*: <<http://www.agisoft.com/>>

⁴ Alice Vision *homepage*: <<https://alicevision.github.io/>>

⁵ CGAL *homepage*: <<https://www.cgal.org>>

teca na linguagem de programação C++. A biblioteca é amplamente documentada e é modularizada em diversos pacotes com funções específicas, como funções relacionadas ao núcleo de geometria básica, busca e ordenação espacial, computação do envoltório convexo e da geração malha, por exemplo. O software não possui um módulo ou operações de fotogrametria, também não utiliza computação em GPU e também não possui um módulo oficial que opera com superfícies suaves;

ContextCapture⁶: produto desenvolvido e comercializado pela Bentley, realiza a reconstrução de objetos aceitando como entrada uma nuvem de pontos ou um conjunto de imagens (utiliza técnicas de fotogrametria). Possui cinco módulos de funcionalidades, destacando-se a possibilidade de utilização de um serviço de computação (processamento) em nuvem, uso integrado da ferramenta com um smartphone e um visualizador que permite realizar mensurações e análise da malha resultante. Com exceção do módulo visualizador, o *ContextCapture* é um software pago, mas que disponibiliza uma versão de teste sob demanda. Não foram encontradas informações sobre as arquiteturas alvo deste software;

Geomagic Wrap⁷: uma das diversas ferramentas desenvolvidas e comercializado pela 3D Systems para a área de manufatura. Também chamada antigamente de *Rapidform*, trata-se de uma ferramenta que auxilia no processo de escaneamento de objetos e oferece recursos para reconstrução de malha, sendo capaz de gerar superfícies suaves NURBS e modelos CAD paramétricos. Esta e demais ferramentas produzidas pela 3D Systems são pagas, com possibilidade de uso gratuito por 15 dias para teste, e são voltadas para o sistema operacional Windows;

GOM Inspect⁸: software com enfoque em inspeção 3D com análise de mensurações, mas que também realiza a reconstrução de malhas poligonais. Possui um viés industrial, realizando o processo de reconstrução baseado em nuvens de pontos e permitindo a manipulação e processamento posterior da malha gerada, além de diversas funcionalidades atreladas a simulação e análise de propriedades. A GOM, empresa desenvolvedora do *GOM Inspect*, fornece também outras soluções pagas nas áreas de análise e simulação voltadas para o sistema operacional Windows, sendo algumas delas disponibilizadas gratuitamente com restrições de uso;

Halcon⁹: um *framework* que permite o desenvolvimento rápido e mais simplificado de aplicações de visão computacional com viés industrial. Desenvolvido e comercializado pela MVTec Software, este *framework* contém uma biblioteca com diversas funcionalidades de visão computacional e aprendizagem de máquina que podem ser utilizadas para criar uma solução específica para um determinado problema através de seu ambiente integrado de

⁶ ContextCapture *homepage*: <<https://www.bentley.com/en/products/brands/contextcapture>>

⁷ Geomagic Wrap *homepage*: <<https://www.3dsystems.com/software/geomagic-wrap>>

⁸ GOM *homepage*: <<http://www.gom.com>>

⁹ Halcon *homepage*: <<https://www.mvtac.com/products/halcon>>

desenvolvimento. Com o uso destas funcionalidades é possível obter nuvens de pontos através de técnicas de fotogrametria, realizar registro das nuvens de pontos e reconstruir malhas triangulares, utilizando inclusive aceleração de processamento por GPU. Apesar de pago, o *framework Halcon* pode ser utilizado gratuitamente com uma licença de teste que pode ser obtida por um distribuidor local, e seus resultados podem ser distribuídos para diversos microprocessadores e sistemas operacionais. Por outro lado, não fica explicitamente identificado se o *framework* em si é multiplataforma;

Imag'Ing¹⁰: trata-se de uma empresa que atua na área de escaneamento e reconstrução de objetos, prestando serviços de processamento de nuvens de pontos e geração de malha através de computação (processamento) em nuvem. Pouca informação técnica é divulgada na página web da empresa ou em suas redes sociais;

JRC 3D Reconstructor¹¹: uma ferramenta da Gexcel (Geomatics & Excellence) que permite importar e processar nuvens de pontos geradas por LiDAR gerando malhas que podem ser exportadas para software de CAD e modelagem. A ferramenta é distribuída em seis versões distintas com especificidades para determinadas áreas de aplicação indicadas pelo seu nome: *construction, mining-tunnelling, heritage-architectural, forensic, photo e full*. Todas as versões são destinadas ao sistema operacional Windows e são pagas, mas, são oferecidas também versões de teste válidas por 30 dias com limitações de uso;

Lib IGL¹²: uma biblioteca de código aberto e multiplataforma voltada para o processamento geométrico. Escrita na linguagem de programação C++, utiliza a abordagem de estrutura baseada somente de arquivos de cabeçalhos, sendo necessário apenas incluir tais arquivos na solução do projeto desenvolvido. Apesar de possuir diversas implementações de funções geométricas, a biblioteca é limitada em relação a funções de reconstrução, disponibilizando apenas uma função de triangulação de Delaunay.

MeshLab¹³: projeto de código aberto voltado para o processamento de nuvens de pontos e malhas. Desenvolvido pelo grupo de pesquisa *Visual Computing Lab*, do Instituto di Scienze e Tecnologie dell'Informazione (ISTI-CNR), o software possui diversas funcionalidades, dentre as quais se destacam: registro de nuvens de pontos, coloração e texturização automática através de imagens, filtros de limpeza de *outliers*, segmentação de objetos, comparação entre modelos e reconstrução de objetos (CIGNONI et al., 2008). Com relação a geração de malha para os modelos 3D, o software oferece algoritmos baseados em pivoteamento de bolas (BERNARDINI et al., 1999) e baseados na definição de superfícies implícitas (KAZHDAN; BOLITHO; HOPPE, 2006; CALLIERI et al., 2003). Além do código fonte do software, também são disponibilizadas as versões mais recentes já compiladas para os

¹⁰ Imag'Ing homepage <<https://www.imag-ing.com/en/home.html>>

¹¹ JRC 3D Reconstructor homepage: <<https://gexcel.it/en/software/jrc-3d-reconstructor>>

¹² Lib IGL homepage: <<https://libigl.github.io>>

¹³ MeshLab homepage: <<http://www.meshlab.net/>>

sistemas operacionais Windows (64 bits), MacOS e Linux. Vídeos tutoriais sobre o seu uso também foram publicados e disponibilizados gratuitamente na internet;

MicMac¹⁴: um software de código aberto e multiplataforma para fotogrametria desenvolvido em uma parceria entre o IGN (*French National Geographic Institute*) e a ENSG (*French National School for Geographic Sciences*). Diversos módulos compõem o MicMac, incluindo o módulo *TiPunch* para geração e filtragem de malha através de nuvem de pontos. O projeto é mantido em um repositório Git e possui uma página Wiki com as principais informações sobre a ferramenta, além de um fórum;

Open3D¹⁵: uma biblioteca de código aberto e multiplataforma que ampara no desenvolvimento de software que trabalham com dados tridimensionais. Conta com algoritmos e estruturas de dados escritas nas linguagens C++ e Python, permitindo a paralelização de determinadas operações. Não possui funções relacionadas a fotogrametria, e não há menção de suporte a geração de superfícies suaves e computação utilizando GPU.

Pix4D Model¹⁶: a Pix4D comercializa um conjunto de software baseados em fotogrametria para mapeamento utilizando drones (imagens aéreas), com o intuito de gerar mapas e modelos tridimensionais precisos. Dentre os produtos vendidos está o *Pix4D Model*, o qual é destinado para a reconstrução de objetos, sendo possível modificar a nuvem de pontos gerada com ferramentas de edição, visualizar e exportar os resultados gerados e inclusive realizar determinadas operações com aceleração baseada em GPU. Todos os produtos da Pix4D são pagos por mensalidade ou anuidade (exceto as licenças educacionais que são perpétuas), mas existe a possibilidade de utilização do software de forma gratuita por um período de teste;

PointCab 3DPro¹⁷: um software para processamento de nuvem de pontos que realiza a operação de registro entre nuvens, a geração de malha triangular no processo de reconstrução de superfície, além da análise de malha. Desenvolvido pela PointCab como parte de um conjunto de ferramentas e módulos voltados ao processamento de nuvens de pontos com objetivos específicos, como por exemplo *Building Information Modeling* (BIM). Trata-se de uma ferramenta paga que pode ser testada gratuitamente por 30 dias, sendo específica para o sistema operacional Windows 64 bits e não utiliza mecanismos de aceleração por GPU no processamento;

PointFuse¹⁸: ferramenta destinada a conversão de nuvem de pontos em malha tridimensional de modelos, prometendo processamento rápido, flexível e geração de arquivos de pequeno

¹⁴ MicMac homepage: <<https://micmac.ensg.eu/index.php>>

¹⁵ Open3D homepage: <<http://www.open3d.org>>

¹⁶ Pix4D Model homepage: <<https://www.pix4d.com/product/pix4dmodel>>

¹⁷ PointCab homepage: <<https://www.pointcab-software.com/en/>>

¹⁸ PointFuse homepage: <<https://pointfuse.com/>>

tamanho. A ferramenta também possui recursos para seleção, separação, classificação e auto classificação de objetos, além de disponibilizar uma funcionalidade de computação (processamento) em nuvem (*PointFuse Connect/Bolt*). Destinado apenas ao sistema operacional Windows, a *PointFuse* é uma ferramenta paga por mensalidade ou anuidade, mas possui uma versão de testes com limitações que pode ser utilizada por 30 dias. Este ferramenta também pode ser utilizada como um plugin para o software Autodesk Revit;

Point Cloud Library - PCL¹⁹: uma biblioteca de código aberto e multiplataforma, livre para usos comerciais e de pesquisa, desenvolvida como um projeto aberto de larga escala voltada para o processamento de imagens e nuvens de pontos bidimensionais e tridimensionais (RUSU; COUSINS, 2011). Escrita em linguagem de programação C++, possui uma implementação modular de recursos para processamento de nuvens de pontos, filtragem, segmentação e reconstrução de superfícies, contendo diversos algoritmos estado da arte que podem ser compilados de acordo com a necessidade do projeto. Este *framework* não apresenta um módulo de fotogrametria, mas pode processar os dados gerados por certas cameras 3D, como Microsoft Kinect e Asus XTionPRO. PCL faz uso de bibliotecas de terceiros como Eigen, FLANN, OpenMP e TBB (Threading Building Blocks), sendo que estas duas últimas, por exemplo, possibilitam paralelização multicore na execução de determinadas operações. A biblioteca possui integração com placas gráficas da NVIDIA, permitindo que determinadas operações sejam computadas em GPU e também possui um módulo voltado à superfícies suaves NURBS;

Polyworks Modeler²⁰: software específico de engenharia reversa desenvolvido pela Innovmetric, que realiza a geração de malha por superfícies suaves (NURBS) sobre nuvens de pontos com controle de continuidade geométrica G^0 , G^1 e G^2 . O software também possui recursos para edição e otimização da malha gerada, inclusive operações booleanas e extrusão, por exemplo. O *Polyworks Modeler* é um software pago, comercializado em duas versões (*Standard* e *Premium*), mas pode ser testado através de um versão de demonstração. É destinado especificamente ao sistema operacional Windows e possui suporte a computação acelerada por GPU para placas de vídeo da NVIDIA;

ReCap²¹: a Autodesk, empresa proprietária de diversas ferramentas conhecidas na área de engenharia, comercializa também as ferramentas *ReCap* e *ReCap Pro*, as quais possuem funcionalidades específicas para o processamento de nuvens de pontos. A *ReCap Pro*, ferramenta mais completa entre as duas, possui funcionalidades para gerar nuvens de pontos com base em um grupo de fotos aéreas, realizar a operação de registro entre escaneamentos, mensurar, marcar, filtrar e editar nuvens de pontos, e também realizar a engenharia reversa, transformando uma nuvem de pontos em uma malha geométrica.

¹⁹ Point Cloud Library *homepage*: <<http://www.pointclouds.org>>

²⁰ Polyworks Modeler *homepage*: <<https://www.innovmetric.com/en/products/polyworks-modeler>>

²¹ ReCap *homepage*: <<https://www.autodesk.com/products/recap>>

O *ReCap* não utiliza mecanismos de aceleração no processamento utilizando GPU e é disponibilizado apenas para o sistema operacional Windows, sendo pago através de mensalidades ou planos anuais. Entretanto, estudantes e educadores podem utilizar a ferramenta de forma gratuita por 3 anos para fins não comerciais, e também existe uma versão de teste que pode ser utilizada gratuitamente por 30 dias;

Revworks²²: a empresa Revware trabalha com ferramentas que auxiliam no processo de engenharia reversa. Além das ferramentas físicas de captura e mensuração, também foi desenvolvida uma ferramenta que faz interface com estes e outros dispositivos de escaneamento, permitindo a coleta de dados com alta taxa de entrada, a reconstrução de objetos e a manipulação destes em um ambiente integrado com o software CAD SolidWorks, parceiro da empresa. Não há muitos detalhes sobre o procedimento de reconstrução mas, de acordo com especificações técnicas do produto, a ferramenta gera superfícies suaves do tipo NURBS. Atualmente a ferramenta *Revworks* é comercializada apenas para o sistema operacional Windows 64 bits;

Trimble Real Works²³: ferramenta de cunho geoespacial desenvolvida pela Trimble – empresa que também comercializa dispositivos de aquisição de nuvem de pontos – permite a análise e processamento de nuvens de pontos. Além das operações básicas de importação, visualização e edição de nuvens de pontos, o software permite também a criação de malha, cálculo de volume, segmentação, classificação automática de objetos, projeção multi-ortogonal, entre outras funcionalidades. Em especial, existe a descrição de uma função de comparação entre os dados construídos e a realidade, mas não fica claro se esta funcionalidade faz uso de alguma métrica para esta comparação, ou se é apenas um suporte visual ao usuário da ferramenta, carregando a nuvem de pontos e o modelo gerado lado a lado. Esta ferramenta é comercializada em quatro versões com diferente disponibilidade de recursos, são pagas e estão disponíveis apenas para o sistema operacional Windows;

VRMesh Reverse²⁴: software que faz parte da família de ferramentas desenvolvidas pela VR-Mesh destinadas ao processamento de nuvens de pontos e à engenharia reversa, permitindo a computação de malha triangular ou superfícies NURBS. Também possui funcionalidades para classificação de objetos em nuvens de pontos, extração de características, realização de registro, reparo e edição de malha (e.g. fechar furos na malha), e extração de medidas. As ferramentas são destinadas exclusivamente ao sistema operacional Windows e são comercializadas através de licenças anuais ou perpétuas, havendo a possibilidade de utilizar gratuitamente uma versão de teste com limitações por 30 dias.

²² Revworks homepage: <<https://revware.net/products/revworks>>

²³ Trimble Real Works homepage: <<https://geospatial.trimble.com/products-and-solutions/trimble-realworks>>

²⁴ VRMesh homepage: <<http://vrmesh.com/>>