

4

Marching Cubes 33: Qualidade da Malha Resultante

O algoritmo Marching Cubes é considerado simples, robusto e com baixo custo computacional, características que contribuíram para torná-lo popular entre os algoritmos de extração de isosuperfícies. Porém no que se refere a qualidade da triangulação da malha resultante, não raramente observamos um grande número de triângulos finos (triângulos com ângulos pequenos) e até mesmo degenerados (triângulos com área zero).

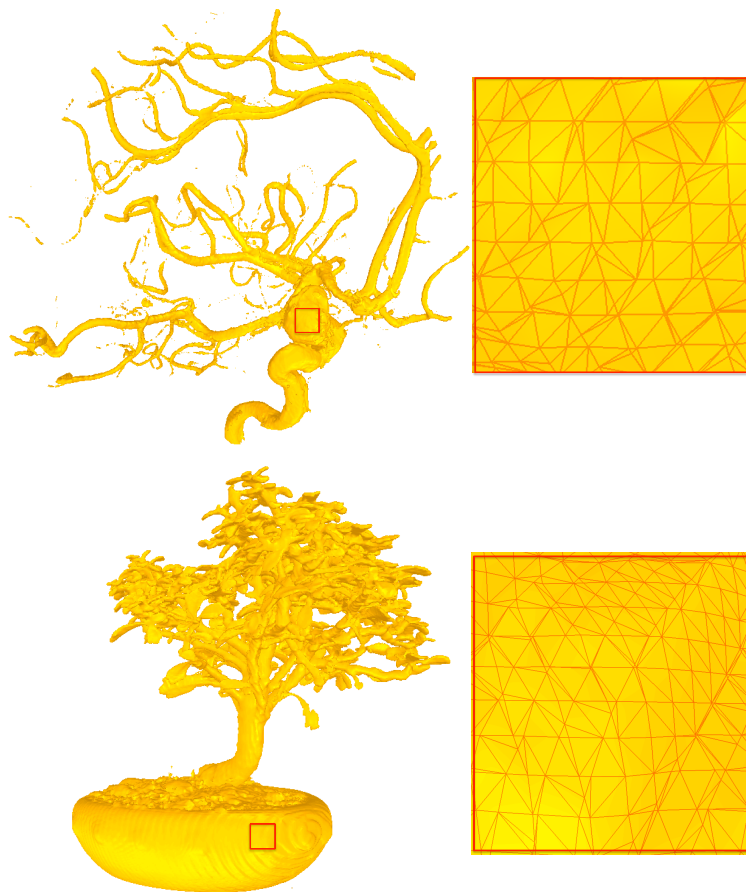


Figura 4.1: Exemplos de malhas geradas pelo C-MC33, extraída do dado Aneurisma, para o isovalor 100 e do dado Bonsai para o isovalor 39. Podemos observar no retângulo em vermelho à direita de cada malha, onde destacamos parte da malha extraída, a baixa qualidade da malha gerada pelo Marching Cubes.

Na Figura (4.1) apresentamos exemplos de malhas extraídas pelo C-MC33 e em ambos os casos. Apesar da maioria dos triângulos possuir boa qualidade, observamos um grande número de triângulos finos.

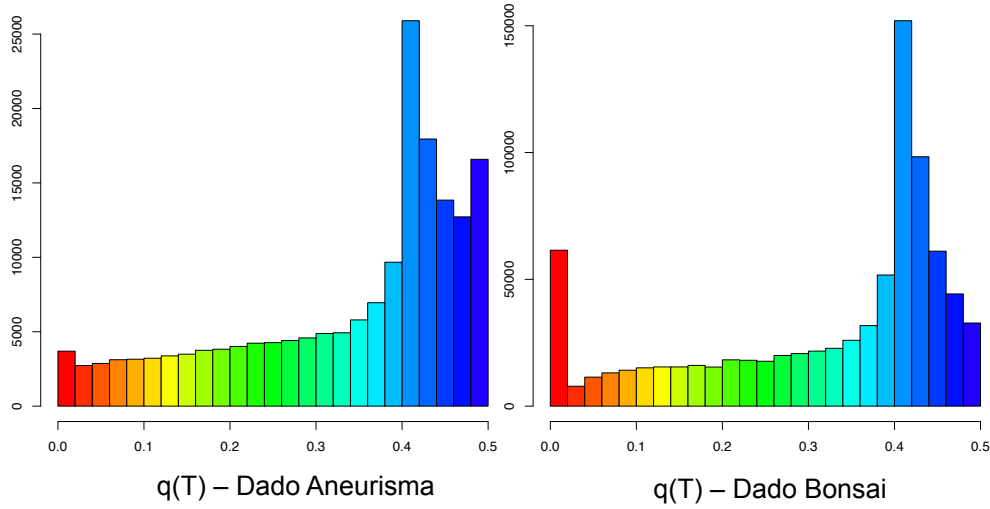


Figura 4.2: Estatísticas de qualidade das malhas apresentadas na Figura 4.1. Na malha extraída do dado Aneurisma, de um total de 173.940 triângulos gerados, 23.450 (13.48%) possuem razão radii inferior a 0.15. Na malha extraída do dado Bonsai, de um total de 822.663 triângulos gerados, 88.053 (10.7%) possuem razão radii inferior a 0.15.

Existem na literatura várias formas de medir a qualidade de um triângulo. Neste trabalho optamos por usar como medida de qualidade a razão radii, que é a razão entre os raios dos círculos inscrito e circunscrito do triângulo (para mais detalhes sobre a razão radii e outras formas de medir a qualidade de um triângulo veja (22)). Seja T um triângulo e q a medida de qualidade expressa pela razão radii, temos $q(T) \in [0, 0.5]$: a todo triângulo degenerado será associado ao valor 0, e triângulos com ângulos muito grandes ou muito pequenos também serão penalizados, sendo associados a valores muito próximos de zero, e todo triângulo equilátero será associado ao valor 0.5. Como podemos observar nos histogramas apresentados na Figura (4.2), em ambas as malhas ilustradas na Figura (4.1), o número de triângulos com qualidade inferior a 0.15, por exemplo, é superior a 10% do número de triângulos da malha.

Uma estratégia comumente usada para resolver esse problema é o pós-processamento da malha gerada pelo Marching Cubes (24, 1). No entanto, a maioria dessas técnicas se preocupam apenas com a qualidade da triangulação final, sem compromisso com a preservação das características geométricas e, em alguns casos, topológicas da isosuperfície de interesse. Por outro lado, como

descrito no Capítulo 2, as técnicas que buscam melhorar a qualidade da malha através das modificações feitas diretamente no algoritmo do Marching Cubes (8, 7) vêm se mostrando eficazes em melhorar a qualidade da malha gerada ao mesmo tempo que preserva características da isosuperfície desejada. Nessa direção, Ramam e Wenger (23) propõem um método que resolve o problema da baixa qualidade da malha gerada pelo Marching Cubes através de modificações em sua tabela de triangulação.

Como descrito no Capítulo 2, o Marching Cubes divide os vértices do gride em duas classes: os vértices com valor escalar MAIOR ou IGUAL ao isovalor escolhido são etiquetados com o símbolo “+” e os vértices com valor escalar MENOR que o isovalor escolhido são etiquetados com o símbolo “-”. Dessa forma, vértices com valor escalar igual ao isovalor desejado são interpretados como vértices positivos. Observe que essa classificação é uma das origem dos triângulos degenerados gerados pelo Marching Cubes. Considere por exemplo, um cubo no qual um dos seus vértices possui valor 1 e todos os outros vértices valor 0. Para um isovalor igual a 1, o cubo terá um vértice positivo e sete vértices negativos. A triangulação para essa configuração é dada pelo caso 1 da tabela, composto por um único triângulo (veja Figura 2.1). No entanto, ao gerar o triângulo, seus três vértices serão colocados sobre o vértice positivo, criando assim um triângulo degenerado.

Para eliminar esse problema, Ramam e Wenger (23) propõem uma nova forma de classificação para os vértices do gride, etiquetando-os com os símbolos “+”, “-” ou “=”, quando possuem, respectivamente, valor escalar, maior, menor ou igual ao isovalor escolhido. E estendem a tabela proposta por Claudio Montani et al. (18), de forma a permitir que os vértices do gride façam parte da tabela de triangulação, o que resulta na eliminação dos triângulos degenerados. Adicionalmente, propõem uma adaptação no valor do campo escalar nos vértices do gride próximos a pontos de interseção da isosuperfície, eliminando assim os triângulos de baixa qualidade (descreveremos o algoritmo proposto na Seção 4.1).

No entanto, a tabela proposta por Claudio Montani et al. (18), usada como base no trabalho de Ramam and Wenger (23), não garante preservar a topologia do interpolante trilinear. Como consequência, a malha resultante do algoritmo proposto, apesar de possuir uma melhor qualidade da triangulação, pode apresentar incoerências topológicas.

Buscando unir no Marching Cubes 33 a coerência topológica (apresentada no capítulo anterior) com uma melhor qualidade da malha resultante, nesse capítulo apresentamos um método que, baseado no trabalho de Ramam e

Wenger, estende a tabela de triangulação do MC33, permitindo que os vértices do gride façam parte da triangulação, resultando em uma melhor qualidade na triangulação da malha gerada pelo C-MC33.

4.1

Algoritmo Proposto por Ramam e Wenger

O algoritmo de extração de isosuperfícies usado por Ramam e Wenger, a menos da tabela de triangulação proposta, não difere do algoritmo Marching Cubes proposto por Lorensen e Cline (16). A contribuição do trabalho está na criação de uma nova tabela de triangulação, onde os vértices do gride são usados na triangulação (o que elimina a possibilidade de geração de triângulos degenerados, apresentada no Marching Cubes original), e na etapa adicional, onde propõem a modificação dos valores escalares de vértices do gride próximos a pontos de interseção, que como veremos na Seção 4.1.2, elimina os triângulos de baixa qualidade.

4.1.1

Eliminando os Triângulos Degenerados: Construção da Tabela Estendida

A construção da tabela estendida é baseada nos trabalhos de Bhaniramka et al. (2, 3), e Lachaud e Montanvert (13), métodos que, dado um campo escalar amostrado sobre os vértices de um poliedro, usam o fecho convexo de um conjunto de pontos sobre esse poliedro para gerar automaticamente um triangulação em seu interior.

Para construir a tabela estendida, Ramam e Wenger (23) propõem, primeiramente, um algoritmo que, dada uma configuração de sinais sobre os vértices do cubo, retorna a triangulação correspondente à isosuperfície no interior do cubo. Em seguida, geram a tabela de triangulação aplicando esse método para todas as configurações de sinais possíveis sobre um cubo unitário.

Na forma de classificação proposta, para cada vértice do cubo existem três possibilidades (“+”, “-” ou “=”), logo, considerando os oito vértice do cubo, existem $3^8 = 6561$ configurações possíveis. A tabela estendida consiste em uma tabela de 6561 linhas, cada linha com uma triangulação correspondente a uma configuração de sinais sobre os vértices do cubo. Observe que aqui não são utilizadas as simetrias do cubo para reduzir o número de linhas da tabela.

Construção da Triangulação no Interior do Cubo

Dado um cubo C , com seus vértices etiquetados com “+”, “-” ou “=” (de acordo com o valor do campo escalar sobre eles), o algoritmo para a construção

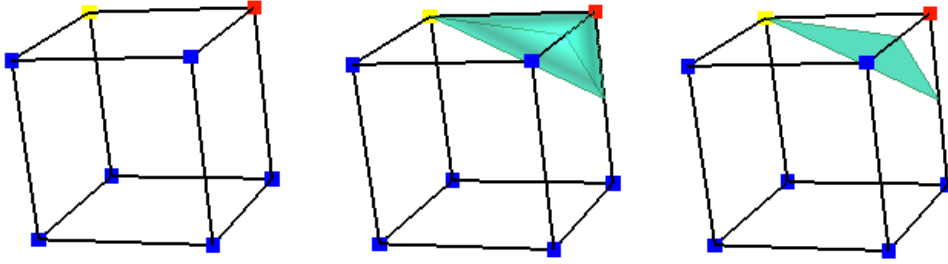


Figura 4.3: Construção da triangulação no interior do cubo proposta por Ramam e Wenger. Dada uma configuração de sinais sobre os vértices do cubo (à esquerda). É construído e triangulado o fecho convexo do conjunto de pontos formado pelos pontos com etiquetas ‘+’ (em vermelho) ou ‘=’ (em amarelo), e o pontos médios das arestas intersectadas (ao centro). Em seguida são removidas da triangulação os triângulos coplanares às faces do cubo. Os triângulos restantes representam a isosuperfície no interior do cubo (à direita).

da triangulação no interior do cubo, proposto por Ramam e Wenger (23), como ilustrado na Figura (4.3), é constituído pelas etapas a seguir:

- Dado W_C , o conjunto formado pelos pontos médios das arestas intersectadas e os vértices etiquetados com ‘+’ ou ‘=’, em C . É Construído o fecho convexo $conv(W_C)$ de W_C .
- Se $conv(W_C)$ é tridimensional, sua fronteira $\partial conv(W_C)$ é triangulada.
- Em seguida, são removidas da triangulação todas as faces coplanares às faces do cubo C . Os triângulos restantes formam a isosuperfície no interior do cubo correspondente à configuração de sinais dada.

4.1.2

Eliminando os Triângulos com Baixa Qualidade

Além dos triângulos degenerados, outro problema na malha gerada pelo Marching Cubes é a presença de triângulos com baixa qualidade..

No Marching Cubes, a qualidade de um triângulo é determinada pela forma que a isosuperfície intersecta o cubo (8, 7). Em geral, os triângulos “ruins” são caracterizados por possuírem dois de seus vértices próximos de um vértice do gride, enquanto o terceiro está afastado desse vértice do gride. Com base nessa observação, Ramam e Wenger (23) propõem uma modificação nos valores do campo escalar sobre o gride. Os vértices do gride próximos a pontos de interseção passam ter valor zero e etiquetados com “=”, eliminando assim os triângulos finos.

O algoritmo proposto, nomeado SnapMC, tem como entrada um parâmetro $\lambda \in [0, 0.5]$, e consiste em um pre-processamento do gride, antes que o Marching Cubes com a tabela estendida seja aplicado. No SnapMC, dado um isovalor α e o parâmetro λ , antes que a isosuperfície seja extraída, são detectadas todas as arestas intersectadas pela isosuperfície e, para cada aresta intersectada, se o ponto de interseção dista menos que λL (sendo L o comprimento da aresta intersectada) de um vértice do gride, esse ponto de interseção é colapsado para o vértice do gride. Na Figura (4.4) apresentamos uma comparação dos resultados obtidos pelo SnapMC e o C-MC33, onde a malha extraída pelo C-MC33 é comparada com a malha extraída pelo SnapMC com $\lambda = 0.1$ e $\lambda = 0.2$. No Histograma (4.5), podemos comprovar a melhor qualidade dos triângulos nas malhas geradas pelo SnapMC.

4.1.3

Incoerência Topológica

Apesar da eficiência comprovada em seus resultados, como apresentado na Figura (4.6), nossos experimentos mostraram que incoerências topológicas estão presentes em algumas malhas resultantes do SnapMC. Como mencionado acima, a tabela estendida proposta no SnapMC é baseada na tabela de triangulação proposta por Claudio Montani et al. (18) que, com apenas 23 casos, não é capaz de representar corretamente o comportamento do interpolante trilinear no interior do cubo.

Os erros causados pelo uso de uma tabela que não garante a topologia correta podem ser observados na Figura (4.6), onde apresentamos as malhas extraídas pelo SnapMC de dados randomicamente gerados. É importante ressaltar que os erros topológicos observados na figura não foram causados ou agravados pelo pré-processamento do gride. As malhas apresentadas na Figura (4.6) foram extraídas pelo SnapMC com parâmetro $\lambda = 0$, ou seja, sem que houvesse modificação no valor de campo escalar sobre o gride.

4.2

Marching Cubes 33: Melhorando a Qualidade da Malha

Apesar dos erros topológicos observados na Seção anterior, os resultados alcançados pelo SnapMC em melhorar a qualidade da malha gerada pelo Marching Cubes nos motivou a obter uma tabela de triangulação estendida a partir da tabela de triangulação proposta por Chernyaev que, como discutido no capítulo anterior, busca representar corretamente a topologia do interpolante trilinear. Com isso, buscamos adicionar ao C-MC33 uma melhor qualidade na

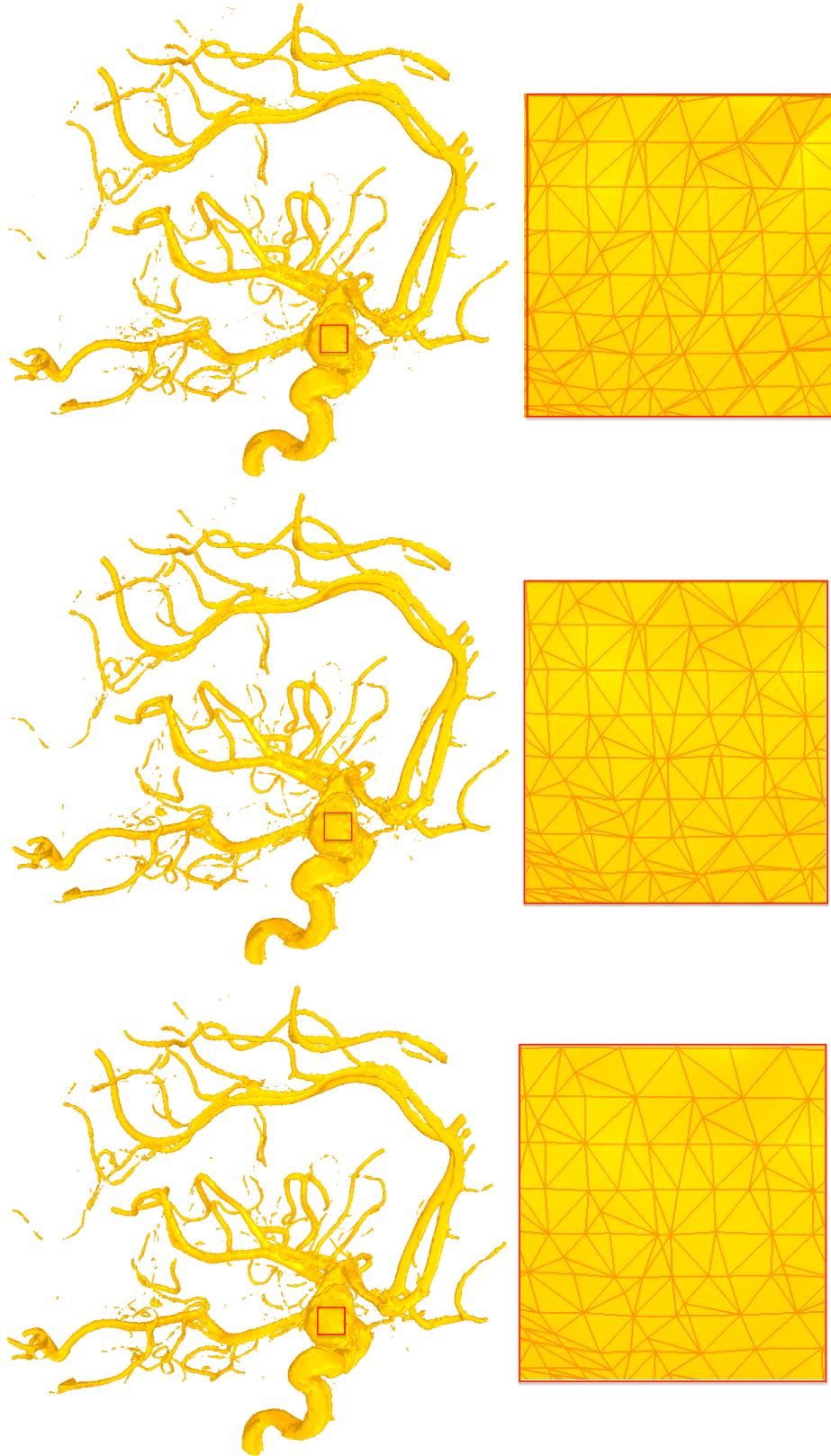


Figura 4.4: Malhas extraídas do dado aneurismo para o isovalor 100. (a) Malha extraída pelo C-MC33. (b) Malha extraída pelo SnapMC, com $\lambda = 0.1$. (c) Malha extraída pelo SnapMC, com $\lambda = 0.2$. Apresentamos as estatísticas da qualidade das malhas nos histogramas apresentados na Figura (4.5).

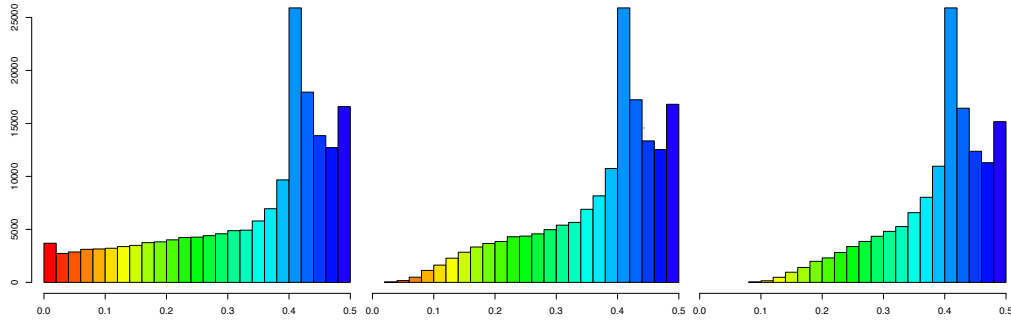


Figura 4.5: Estatísticas da qualidade das malhas apresentadas na Figura (4.4). À esquerda, a malha extraída pelo C-MC33. Ao centro, a malha extraída pelo SnapMC, com $\lambda = 0.1$. À direita, a malha extraída pelo SnapMC, com $\lambda = 0.2$.

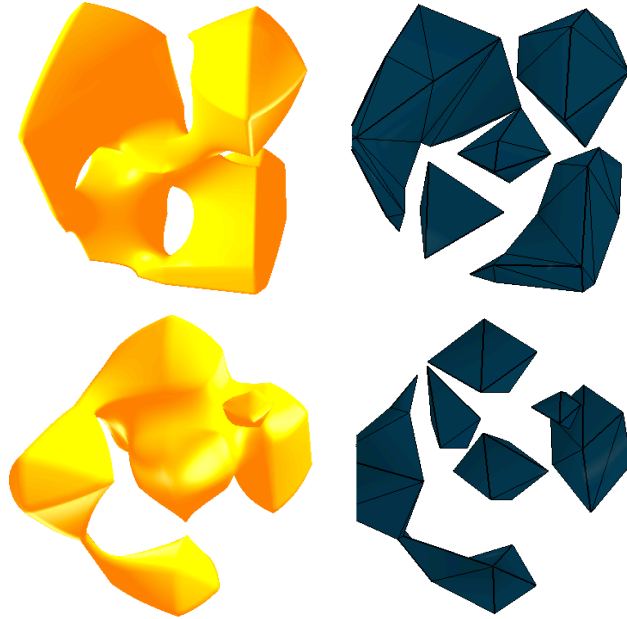


Figura 4.6: Erros topológicos apresentados nas malhas extraídas pelo SnapMC. Na primeira coluna, em amarelo, malhas extraídas pelo C-MC33 de dois campos escalares randomicamente gerado, amostrados em grides $64 \times 64 \times 64$, formado apenas por casos simples da tabela do Marching Cubes. Na segunda coluna, em azul, malhas extraídas pelo SnapMC dos mesmos campos escalares apresentados na primeira coluna, agora amostrados em grides $5 \times 5 \times 5$, onde casos ambíguos passam a fazer parte do grille. Podemos observar que o SnapMC falha em representar corretamente a topologia do interpolante trilinear.

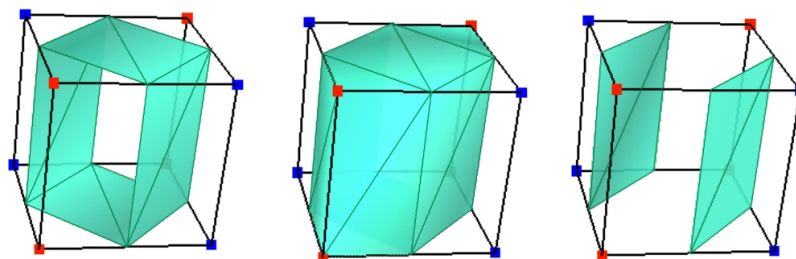


Figura 4.7: Exemplo de incoerência topológica no SnapMC. À esquerda, o caso 10.1.2 na tabela do MC33. Ao centro, a primeira etapa do algoritmo proposto por Ramam e Wenger. Triangulação do fecho convexo dos vértices positivos e pontos médios das arestas intersectadas. Ao removermos os triângulos coplanares às faces do cubo (à direita), temos a triangulação obtida pelo algoritmo proposto. Observe que o método falha em construir corretamente o túnel no interior do cubo.

malha gerada.

Um dos casos onde o algoritmo o SnapMC falha em obter a triangulação correta é quando o interpolante descreve um túnel no interior do cubo. Ilustramos esse problema na figura (4.7). Considere, por exemplo o caso 10.1.2 da tabela proposta por Chernyaev, onde o interpolante trilinear representa um túnel no interior do cubo. Para essa mesma configuração de sinais sobre os vértices do cubo, se aplicarmos o SnapMC, a triangulação obtida será correspondente a duas folhas em seu interior. O mesmo problema ocorre na construção da triangulação de qualquer caso da tabela proposta por Chernyaev onde o interpolante descreve um túnel no interior do cubo.

Para solucionar este problema propomos um algoritmo para construir uma tabela estendida de triangulação que engloba os 33 casos da tabela proposta por Chernyaev. Assim como o SnapMC, dada uma configuração de sinais sobre seus vértices, usaremos o fecho convexo de um dado conjunto de pontos sobre cubo para construir a triangulação que represente a isosuperfície em seu interior.

4.2.1

Construção da Triangulação no Interior do Cubo

Nessa seção, propomos um algoritmo para construção da triangulação que tem como base a análise da conectividade dos vértices do cubo. Apresentamos a seguir algumas definições de conceitos fundamentais para a construção de nosso algoritmo. Dado um cubo e uma configuração de sinais sobre seus vértices:

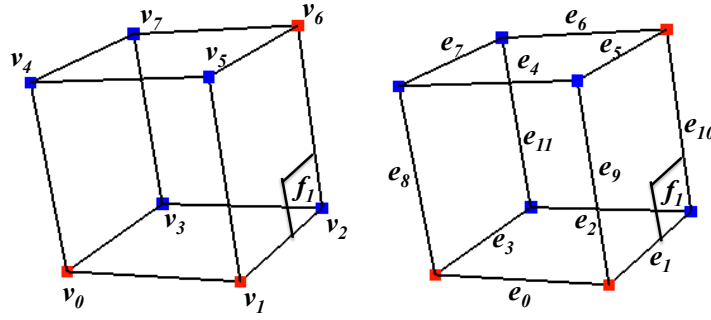


Figura 4.8: Configuração de sinais do caso 6 e índice numérico dos vértices e arestas do cubo.

Definição 1 Dizemos que um cubo é positivo se seu número de vértices etiquetados com “+” ou “=” for menor ou igual a 4. Consequentemente, um cubo será dito negativo se seu número de vértices etiquetados com “+” ou “=” for maior que 4.

Definição 2 Dado um cubo positivo, um grupo de vértices, que denotaremos como G_v , é formado pelo conjunto de vértices positivos conectados pelas arestas ou pelas faces do cubo.

Definição 3 Um grupo de arestas, que denotaremos como G_a é formado pelo conjunto de arestas intersectadas pela isosuperfície e incidentes aos vértices que constituem um grupo de vértices.

O caso 6 da tabela do MC33 é um exemplo que ilustra o conceito de grupos de vértices e grupos de arestas. Vamos assumir um cubo positivo com os vértices v_0 , v_1 e v_6 positivos e o restante dos vértices negativos, como na Figura (4.8).

- No subcaso 6.1.1 (à esquerda da Figura 4.9), os vértices v_0 e v_1 estão conectados pela aresta e_0 e estão separados do vértice v_6 pelas faces e pelo interior do cubo. Para essa configuração, existem no cubo dois grupos de vértices, $G_{v_0} = \{v_0, v_1\}$ e $G_{v_1} = \{v_6\}$. E associados a eles, os grupos de arestas $G_{a_0} = \{e_1, e_3, e_8, e_9\}$ e $G_{a_1} = \{e_5, e_6, e_{10}\}$.
- No subcaso 6.1.2 (no centro da Figura 4.9), os vértices v_0 e v_1 estão conectados pela aresta e_0 e estão conectados ao vértice v_6 pelo interior do cubo. No entanto, como essa conexão não é feita por uma das faces, temos no caso 6.1.2 os mesmos grupos de vértices e arestas do caso 6.1.1.

- No subcaso 6.2 (à direita da Figura 4.9), os vértices v_0 e v_1 estão conectados pela aresta e_0 e estão conectados do vértice v_6 pela face f_1 . Para essa configuração existe um único grupo de vértices $G_{v0} = \{v_0, v_1, v_6\}$ e associado a ele, o grupo de arestas $G_{a0} = \{e_1, e_3, e_5, e_6, e_8, e_9, e_{10}\}$.

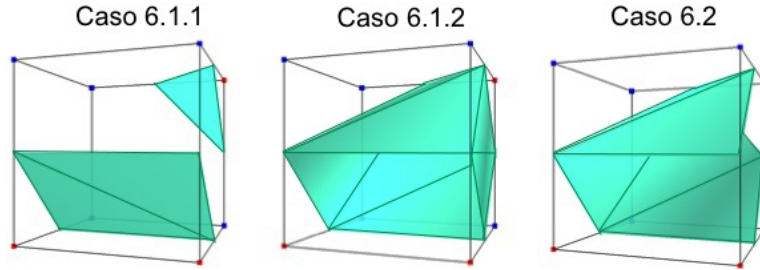


Figura 4.9: Configuração de sinais do caso 6 e índice numérico dos vértices a arestas do cubo.

A primeira etapa em nosso algoritmo consiste na análise da conectividade dos vértices do cubo e na construção dos grupos de vértices e grupos de arestas. Em seguida, para os casos ambíguos, a topologia descrita pelo interpolante trilinear no interior do cubo é determinada através dos testes descritos no capítulo anterior. Uma vez determinada a topologia no interior do cubo, seguimos para a terceira etapa, onde é construída a triangulação no interior do cubo.

Nessa terceira etapa, o processo de triangulação é dividido em três métodos, de acordo com a topologia descrita no interior do cubo.

Triangulação de Folhas

Para construir a triangulação de folhas no interior do cubo aplicamos, para cada grupo de vértices e seus respectivos grupos de arestas, o algoritmo de construção da triangulação proposto no SnapMC. Observe que o conceito de grupos de vértices e arestas e a aplicação do algoritmo em cada grupo, separadamente, é fundamental para que os casos ambíguos sejam triangulados corretamente. Tomando novamente o caso 6 como exemplo, tanto no subcaso 6.1.1, quanto no subcaso 2.1, o interpolante trilinear descreve folhas no interior do cubo, no entanto os vértices positivos (considerando um cubo positivo) se conectam de formas diferentes em cada subcaso. No subcaso 6.1.1, como descrito acima, devem ser considerados dois grupos de vértices e aplicar separadamente o algoritmo proposto para cada um deles (veja Figura (4.10)). Já no subcaso 6.2, onde existe apenas um grupo de vértices e arestas, a

construção da triangulação é feita exatamente como proposta no SnapMC (veja Figura 4.10).

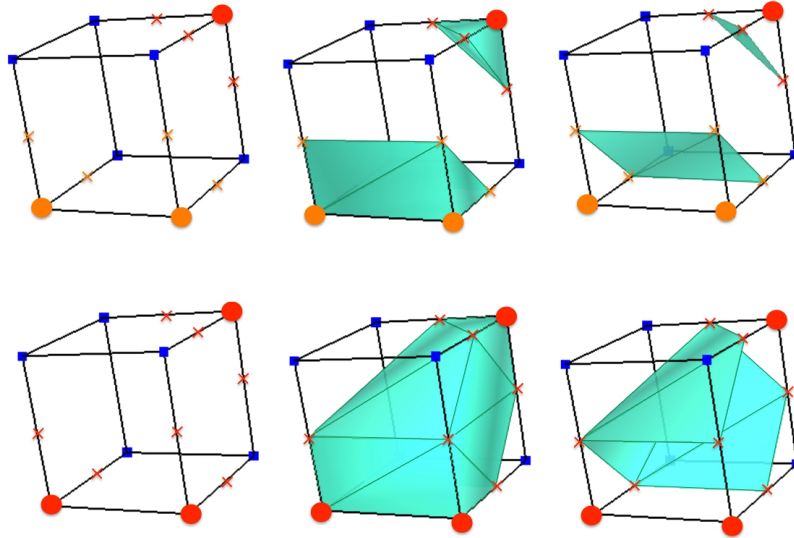


Figura 4.10: Etapas no processo de triangulação de folhas no interior do cubo para os subcasos do caso 6. Na primeira linha apresentamos as etapas da triangulação do subcaso 6.1.1, onde existem dois grupos de vértices e arestas, e o método de triangulação deve ser aplicado separadamente em cada um deles. Na segunda linha apresentamos as etapas da triangulação do subcaso 6.2, no qual existe apenas um grupo de vértices e arestas e o processo de triangulação é feito exatamente como proposto por Ramam e Wenger.

Observe que no método original, independente de como os vértices se conectem, a triangulação gerada será a do subcaso 6.2, não considerando a existência do caso 6.1.1 e do caso 6.1.2, no qual interpolante trilinear descreve um túnel no interior do cubo.

Nas seções 4.2.1 e 4.2.1 propomos métodos para construir a triangulação de casos não considerados no SnapMC, os túneis, já mencionados anteriormente, e folhas que necessitam que um ponto seja adicionado no interior do cubo na construção de sua triangulação.

Triangulação de Túneis

Em nosso algoritmo, a construção da triangulação de túneis no interior do cubo é constituída pelas seguintes etapas:

- Dado um cubo e o conjunto W formado pelos pontos médios das arestas intersectadas, construímos o fecho convexo $conv(W)$ de W .

- Triangulamos $\partial conv(W)$, a fronteira do fecho convexo de W .
- Removemos da triangulação de $\partial conv(W)$ todo triângulo cujo os três vértices estão sobre as arestas de um mesmo grupo de arestas.

Ilustramos na Figura (4.11) as etapas descritas acima, para a construção da triangulação do caso 6.1.2.

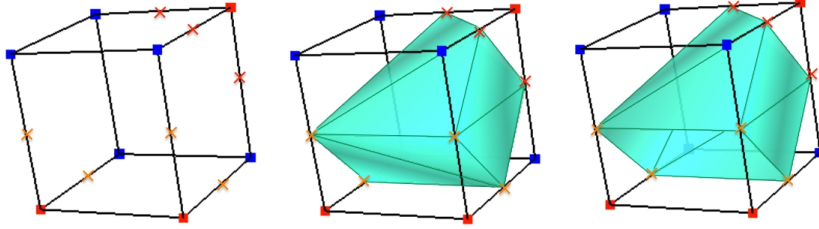


Figura 4.11: Método para construção da triangulação de túneis. Construção da triangulação para o subcaso 6.1.2. À esquerda, o conjunto W , formado pelos pontos médio das arestas intersectadas, pontos destacados com cruzes sobre as arestas. Ao centro, a triangulação do fecho convexo de W . E à direita, são removidos da triangulação de W os triângulos cujos os três vértices estão sobre um mesmo grupo de arestas.

Triangulação Com Pontos Adicionados no Centro do Cubo

A triangulação de folhas e túneis descritas acima cobrem 27 dos 32 casos da tabela do MC33, restando os casos 7.3, 10.2, 12.2, 13.3, 13.4, para os quais a triangulação exige que um ponto seja adicionado no centro do cubo. Uma vez que um desses casos é detectado, construímos a triangulação no interior do cubo gerando para cada face com arestas intersectadas, triângulos com um par de vértices que estão sobre as arestas intersectadas dessa face (respeitando a conectividade dos vértices sobre a face) e o terceiro vértice no centro do cubo.

4.3

Resultados

Nas Figuras (4.12) e (4.14) utilizamos grides randomicamente gerados para comparar os resultados obtidos pelo SnapMC e pelo nosso método. Em amarelo apresentamos a malha extraída pelo C-MC33 de um campo escalar amostrado em um gride $64 \times 64 \times 64$, formado apenas por casos simples da tabela do Marching Cubes. Em verde apresentamos a malha extraída, pelo C-MC33, do mesmo campo escalar, agora amostrado em um gride $6 \times 6 \times 6$,

onde casos ambíguos passam a fazer parte do gride. Em azul malhas extraídas pelo SnapMC e em vermelho as malhas extraídas pelo C-MC33 utilizando a tabela estendida proposta, do campo escalar amostrado no gride $6 \times 6 \times 6$.

Podemos observar na Figura (4.12) que o nosso algoritmo se mostrou eficaz em melhorar a qualidade da malha resultante e manter as características topológicas da isosuperfície induzida pelo interpolante trilinear. A redução do número de triângulos de baixa qualidade pode ser observada nos histogramas apresentados na Figura (4.13). Observamos que as malhas extraídas pelo SnapMC apresentam uma maior redução no número de triângulos de baixa qualidade, no entanto, como podemos observar na Tabela (4.1), tal melhoria não se reflete na preservação das características topológicas do isosuperfície. De fato, como mencionado anteriormente, o SnapMC pode falhar em representar corretamente a isosuperfície mesmo para $\lambda = 0$ (quando não houve modificação nos valores do campo escalar sobre o gride).

Observamos que, assim como no método original, em nosso algoritmo a modificação do valor escalar sobre os vértices do gride pode resultar em incoerências topológicas. Observe na Figura 4.15 que o nosso método, apesar de representar corretamente o isosuperfície quando $\lambda = 0$, deixa de detectar uma pequena componente conexa e gera um vértice não variedade quando extraímos a malha para $\lambda = 1$. Mas, como podemos observar na Tabela (4.1), ainda assim o nosso método apresenta melhores resultados que o SnapMC, as malhas extraídas pelo nosso método melhor aproximam a isosuperfície de interesse.

Como trabalho futuro, a nossa próxima etapa será, dado um valor de λ , detectar as alterações topológicas decorrentes dessa modificação do campo escalar, e permitir que o usuário decida até que ponto ele pode abrir mão da coerência topológica em prol de uma melhor qualidade da malha.

	C-MC33	SnapMC			C-MC33 + Tab. Estendida		
		$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$
Comp. Conexa	1	4	7	5	1	1	1
Genus	6	6	indefinido	indefinido	6	6	6
Vértices não variedade	0	0	4	5	0	0	0
Arestas não variedade	0	0	2	4	0	0	0

Tabela 4.1: Estatísticas topológicas das malhas apresentadas na Figura 4.12.

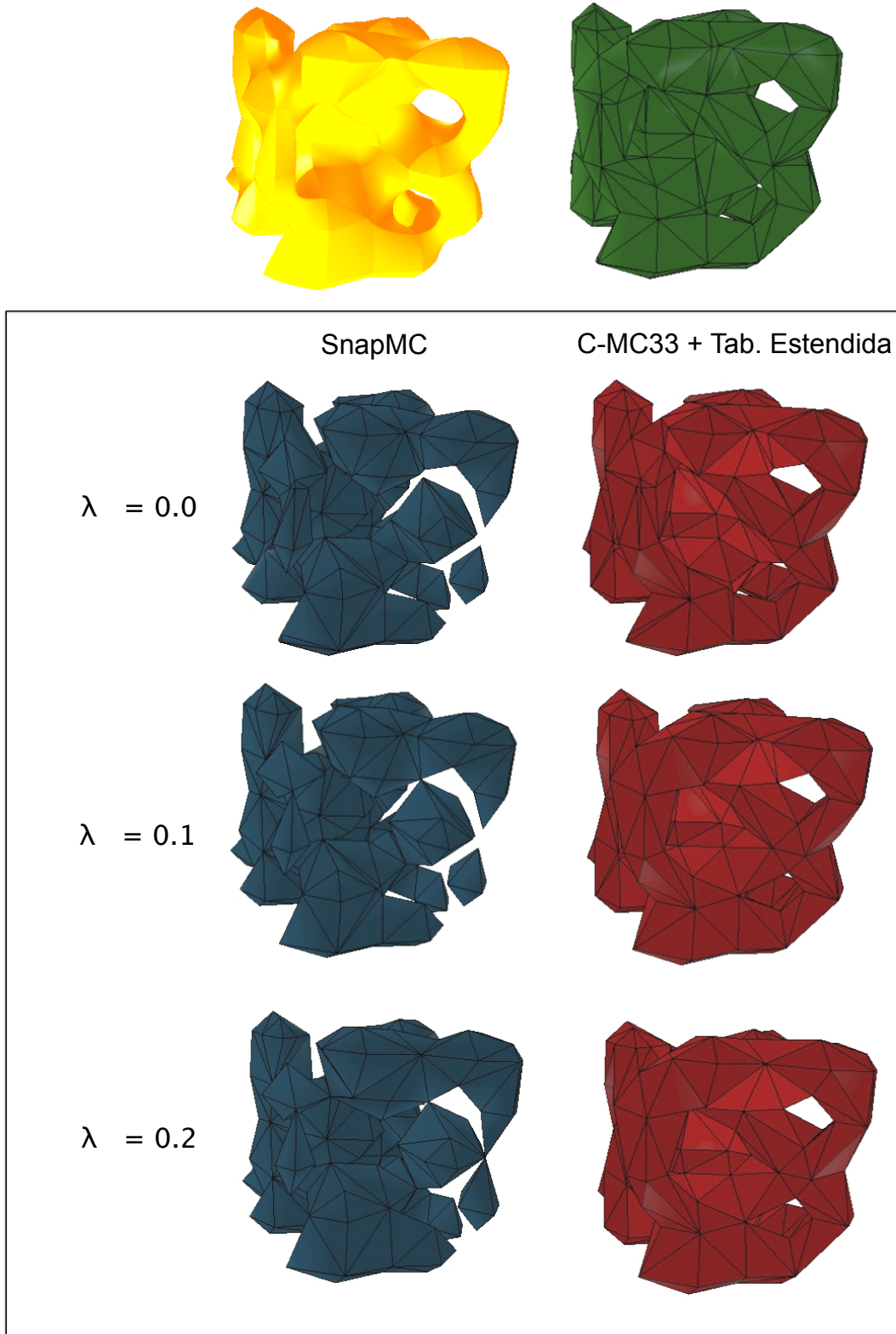


Figura 4.12: Comparação dos obtidos pelo SnapMC e pelo nosso método. Em amarelo apresentamos a malha extraída pelo C-MC33 de um campo escalar amostrado em um gride $64 \times 64 \times 64$ (formado apenas por casos simples da tabela do Marching Cubes). Em verde, azul e vermelho apresentamos, respectivamente, as malhas extraídas, pelo C-MC33, SnapMC e C-MC33 utilizando a tabela estendida proposta (para três diferentes valores de λ), do mesmo campo escalar, agora amostrado em um gride $6 \times 6 \times 6$, onde casos ambíguos passam a fazer parte do gride.

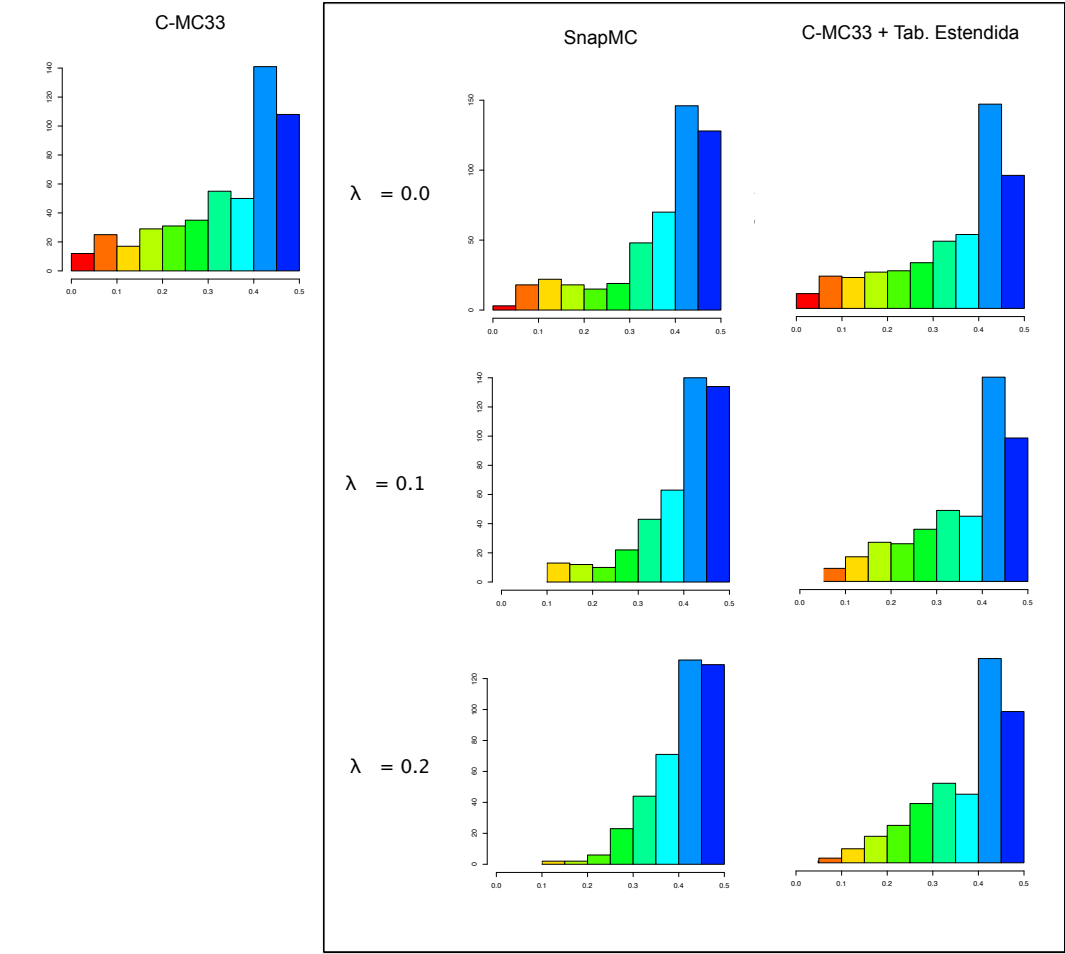


Figura 4.13: Estatísticas da qualidade dos triângulos nas malhas apresentadas na Figura 4.12.

	C-MC33	SnapMC			C-MC33 + Tab. Estendida		
		$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$
Comp. Conexa	2	6	6	6	2	1	1
Genus	4	0	indefinido	indefinido	4	indefinido	indefinido
Vértice não variedade	0	0	3	4	0	1	2
Aresta não variedade	0	0	4	12	0	0	0

Tabela 4.2: Estatísticas topológicas das malhas apresentadas na Figura 4.14.

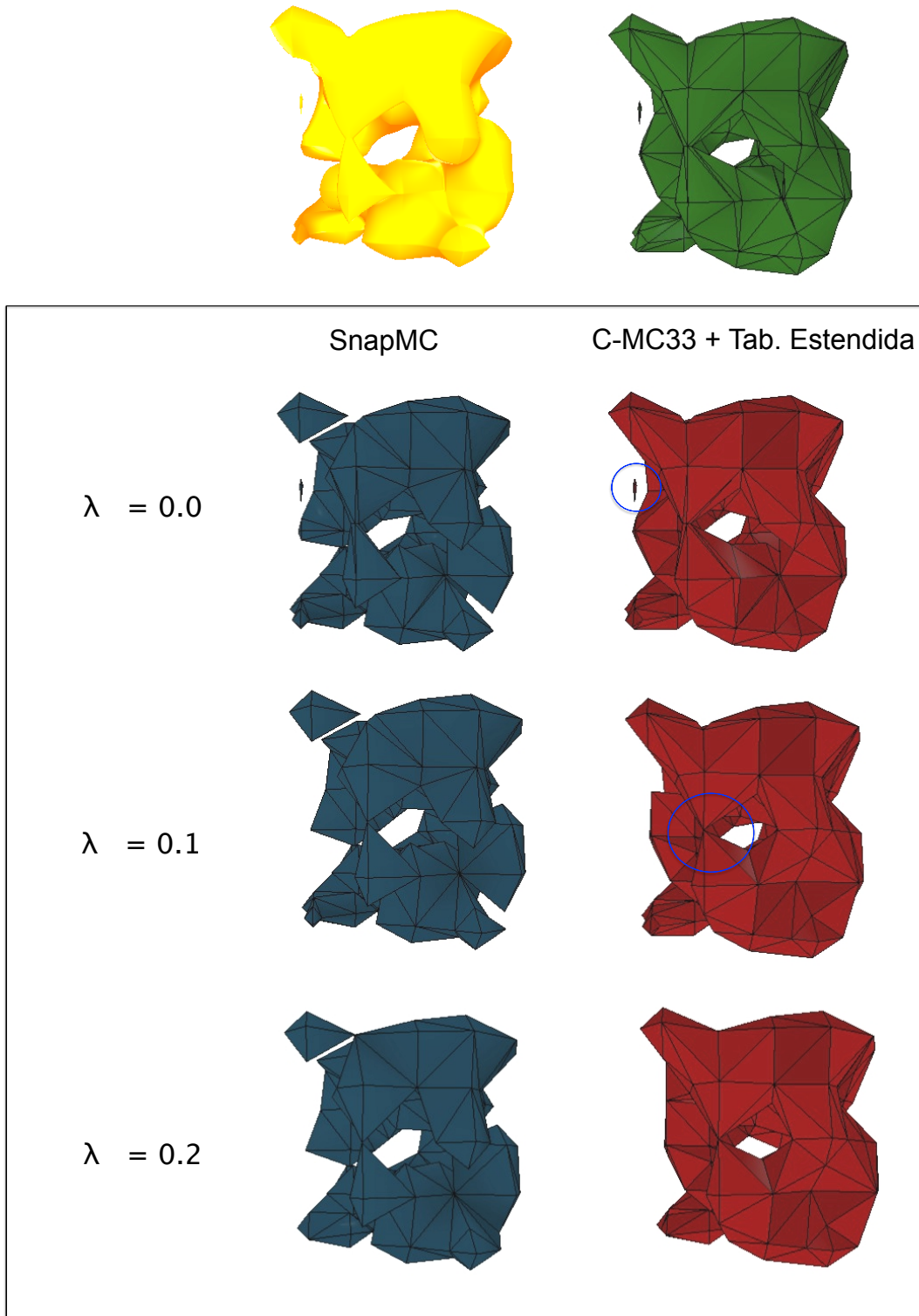


Figura 4.14: Comparação dos obtidos pelo SnapMC e pelo nosso método. Em amarelo apresentamos a malha extraída pelo C-MC33 de um campo escalar amostrado em um gride $64 \times 64 \times 64$ (formado apenas por casos simples da tabela do Marching Cubes). Em verde, azul e vermelho apresentamos, respectivamente, as malhas extraídas, pelo C-MC33, SnapMC e C-MC33 utilizando a tabela estendida proposta (para três diferentes valores de λ), do mesmo campo escalar, agora amostrado em um gride $6 \times 6 \times 6$, onde casos ambíguos passam a fazer parte do gride.

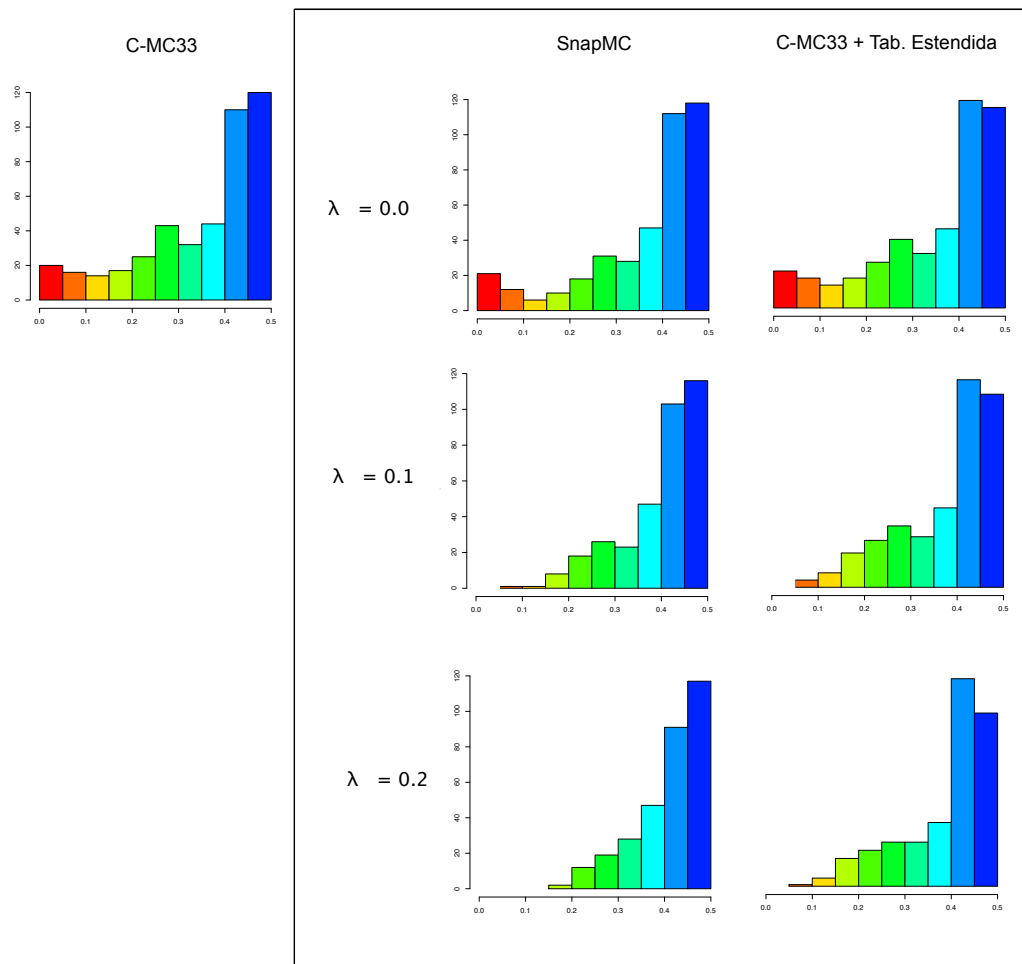


Figura 4.15: Estatísticas da qualidade dos triângulos nas malhas apresentadas na Figura 4.14.