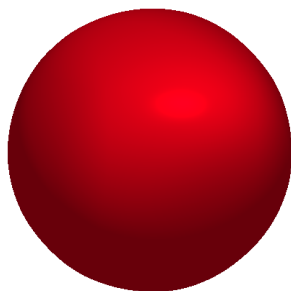


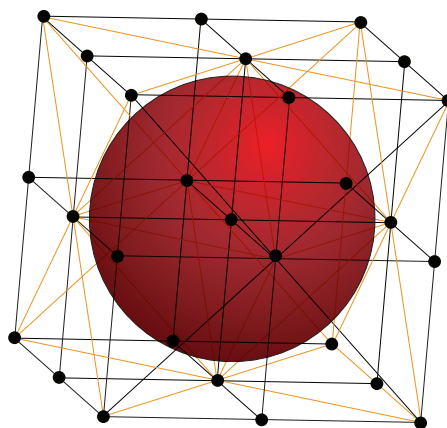
1

Introdução

Uma superfície é definida implicitamente quando é descrita como um conjunto de pontos com uma propriedade em comum. A formulação mais utilizada é, dada uma função $F : \mathbb{R}^3 \rightarrow \mathbb{R}$, descrevê-la como o conjunto $F^{-1}(c)$ onde c é um valor no contradomínio de F (figura 1.1(a))(1). Em computação gráfica, para representarmos superfícies desse tipo uma abordagem comum é criar uma partição do espaço (2) (figura 1.1(b)) e avaliar em cada parte um pedaço da superfície, interpolando F por funções cujas superfícies implícitas são fáceis de aproximar (figura 1.2(a)). Para este trabalho, utilizamos a decomposição em tetraedros.



1.1(a): $S : x^2 + y^2 + z^2 = c$.

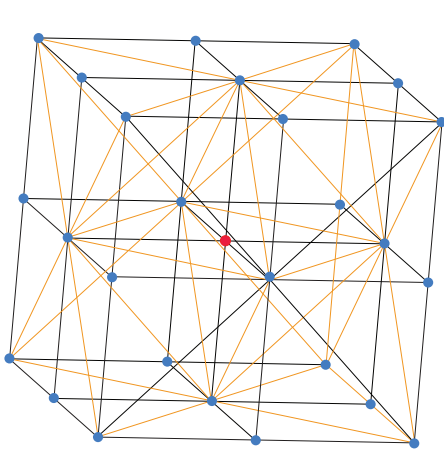


1.1(b): Partição do espaço.

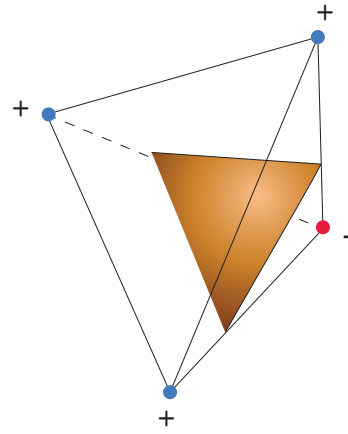
Figura 1.1: Partição do espaço contendo a esfera S .

Por exemplo, podemos criar uma aproximação poligonal fácil de transpor para o computador (figura 1.2(c)) que é essencialmente uma parametrização da superfície descrita na figura 1.2(d) como um conjunto de triângulos com vértices definidos por interpolação linear sobre as arestas em que o valor da função muda de sinal (figura 1.2(b))(3).

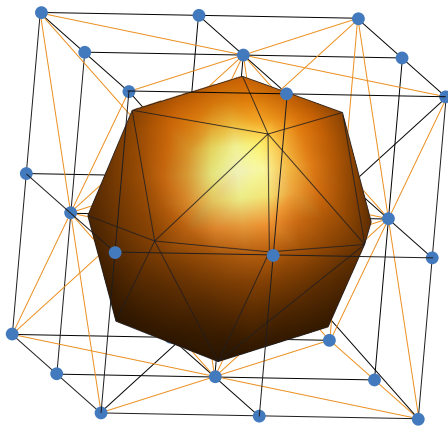
Essas duas representações fundamentais, implícita e paramétrica, têm vantagens complementares. Neste trabalho estamos interessados em unir o



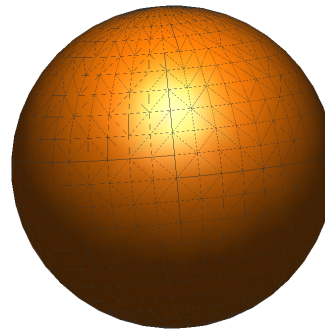
1.2(a): Partição do espaço: pontos vermelhos estão dentro da superfície e pontos azuis estão fora dela.



1.2(b): Gerando os triângulos.



1.2(c): Aproximação grosseira da esfera.



1.2(d): Refinando a malha.

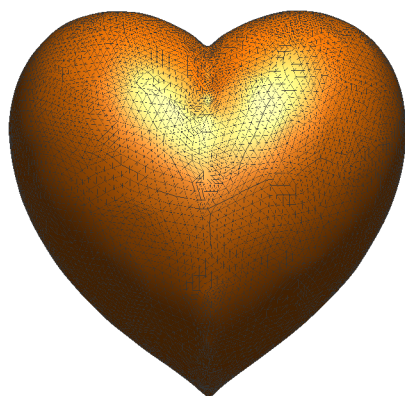
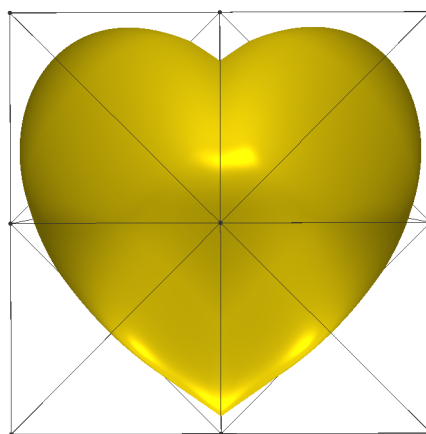
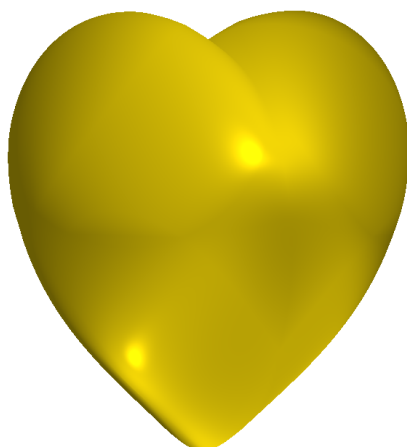
Figura 1.2: Criando uma aproximação poligonal da esfera.

mundo implícito e o paramétrico ao apresentarmos uma nova estrutura para representação de superfícies que nomearemos de *TetraQuad*. O *TetraQuad* é uma cúbica, superfície implícita de grau 3, definida pela interpolação de quádricas em um tetraedro. Utilizamos esses objetos para aproximar superfícies como a esfera da figura 1.1(a), criada com apenas um tetraedro de maneira mais eficiente que os triângulos da figura 1.2(d).

O objeto resultante mantém as vantagens da definição implícita. A conversão em imagens é natural e ainda podemos extrair facilmente informações como normais e curvaturas, bem como fazer alterações na topologia ou consultar se um ponto pertence ou não a superfície através do valor da função. Além disso, mantemos a estrutura paramétrica que é de fácil assimilação pelo

computador com a vantagem da redução do número de dados armazenados já que para cada *TetraQuad* deixamos de processar inúmeros triângulos. Um trabalho pioneiro foi proposto por Sederberg (4) para modelagem de superfícies por *A-patches* (superfícies algébricas) definidas em uma malha de controle.

Como aplicação, modelamos um sistema para reconstruir superfícies a partir de uma nuvem de pontos utilizando *TetraQuads*, inspirado em abordagens bem conhecidas (5, 6) junto com um algoritmo para visualização utilizando *ray casting* em GPU¹. Na figura 1.3 temos o modelo conhecido como *Cœur* visualizado com 8974 triângulos e a sua reconstrução com 36 *TetraQuads*.

1.3(a): Malha de triângulos do *Cœur*.1.3(b): Reconstrução por *TetraQuads*.

1.3(c): Resultado final

Figura 1.3: Resultado obtido com *TetraQuads*.

¹A GPU é um processador construído para processamento gráfico, mas a sua arquitetura constituída de centenas de unidades de processamento estimulam o seu uso em algoritmos paralelizáveis (7).

1.1 Trabalhos Relacionados

A inspiração para este trabalho veio da tese de Toledo (8), mais especificamente os capítulos referentes à reconstrução de superfícies implícitas e a visualização de modelos constituídos de formas simples como esferas, cones, cilindros e superfícies toroidais. A figura 1.4 ilustra partes de uma plataforma de petróleo reconstruída utilizando a técnica proposta por Toledo. O *pipeline* de parâmetros segue o mesmo modelo de outros trabalhos, são eles: estimativa, segmentação, classificação e reconstrução. As duas etapas principais são as de segmentação e classificação e para a execução delas foram utilizadas duas abordagens, uma geométrica e outra topológica.

A primeira utiliza mínimos quadrados para fazer o *fitting* (ajuste) dos pontos de entrada após o processo de segmentação, a partir daí, para cada porção de superfície é associada uma quádrlica para representá-la constituindo a etapa de classificação. A segunda abordagem utiliza as próprias características dos modelos, por exemplo, o posicionamento de arestas e vértices. A malha é percorrida em busca de padrões (anéis e tubos) para definir quais objetos dentre, cilindros, partes de cones e toros, representam cada porção da superfície sendo reconstruída.

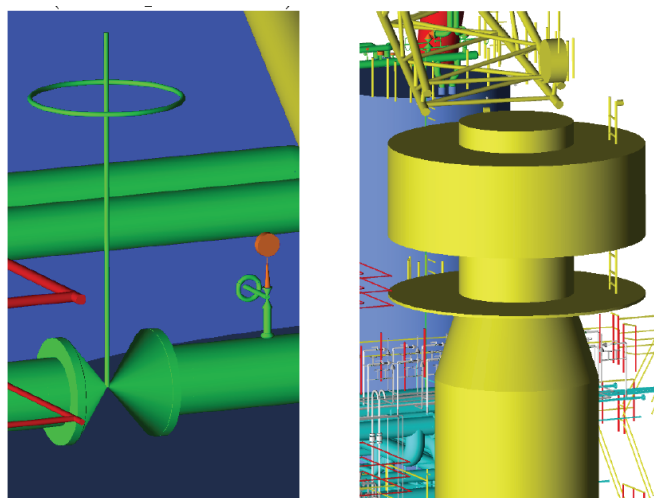


Figura 1.4: Reconstruções obtidas a partir da técnica proposta por Toledo, onde são empregadas superfícies simples como esferas, cones, cilindros e toros.

O trabalho corrente expande as possibilidades da tese de Toledo. Nosso intuito é reconstruir superfícies com objetos representados por cúbicas que apresentam uma maior variedade de formas e podem representar modelos mais complexos. Também excluimos a etapa de classificação já que o número de objetos gerados a partir das cúbicas inviabilizaria esse processo. Outra

diferença encontra-se no emprego de malhas de tetraedros para reconstruir os modelos, o que inclui uma etapa de decomposição em triângulos com o uso do *geometry shader*.

Um trabalho semelhante, combinando as abordagens paramétrica e implícita, foi desenvolvido por Mello (9) para superfícies em dimensão arbitrária (hipersuperfícies) obtidas através de difeomorfismos que permitem, por exemplo, definir um campo de normais contínuo (figura 1.5). A generalidade dessa construção de difeomorfismos simpliciais permite modelar objetos mais complexos que os usados nessa tese, com um custo adicional para renderizar o resultado. Em comparação, o *TetraQuad* apresentado aqui foi desenvolvido a partir dos recursos da placa gráfica.

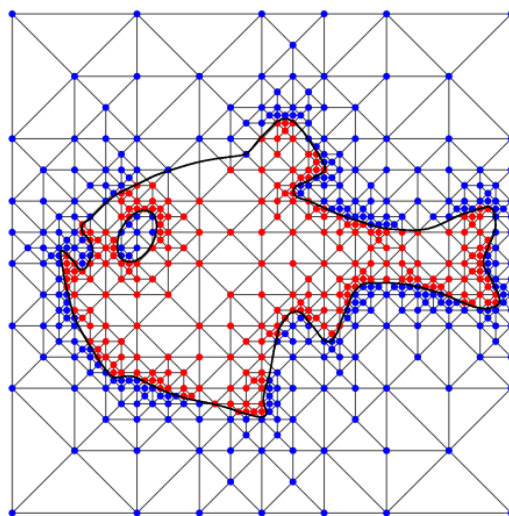


Figura 1.5: Estimação de topologia apresentada no trabalho de Mello.

Uma técnica muito próxima é também descrita no artigo de Loop e Blinn (10) utilizando uma estrutura baseada em tetraedros de Bézier (11) para visualização de superfícies de no máximo grau 4 (figura 1.6). Os modelos são reconstruídos por pedaços de superfícies de Bézier que são definidos implicitamente por pesos posicionados sobre nós dos tetraedros. Uma implementação em *shader* é utilizada para processar os coeficientes e encontrar raízes que determinem pontos visíveis na superfície e suas normais para o processo de renderização.

Temos aqui algumas diferenças em relação ao trabalho de Loop e Blinn (10). Talvez a mais importante delas é que não consideramos a malha de tetraedros como intrínseca ao dado, permitindo uma melhor separação das descrições implícita e paramétrica. Além disso, permitimos um certo grau de extrapolação, ao reconstruirmos partes que faltam da superfície apenas



Figura 1.6: Animação de uma superfície de grau 4 utilizando a técnica de Loop e Blinn.

posicionado novos tetraedros na malha sem preocupação com os pesos da abordagem com Bézier.

Outra diferença é que o *TetraQuad* chega à etapa de renderização sem a necessidade de nenhuma transformação adicional enquanto as superfícies de Bézier necessitam passar por transformações realizadas ainda em CPU já que o número de instruções necessárias inviabiliza a execução na placa gráfica.

Uma formulação 2D do *TetraQuad* para triângulos, chamada de *TriQuad*, está descrita na dissertação de mestrado de Gomes (12). Em Gomes (12), podemos descobrir como construir curvas em uma malha de triângulos a partir da combinação baricêntrica de polinômios de grau 2 posicionados nos vértices. Os pedaços de curvas resultantes desse processo são também polinômios cúbicos definidos implicitamente. Essa foi uma ferramenta importante para interpretar o novo objeto que estamos propondo. Alguns dos problemas verificados para o caso 2D já haviam sido observados no *TetraQuad* e puderam ser melhor compreendidos por estarmos trabalhando com triângulos. Um exemplo de *TriQuad* pode ser observado na figura 1.7.

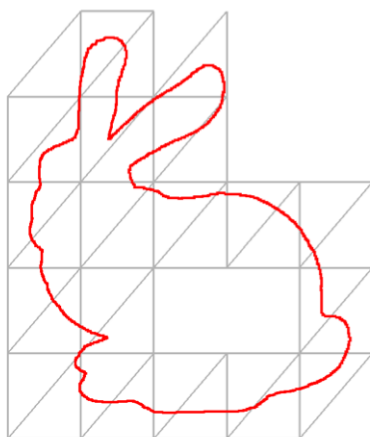


Figura 1.7: Exemplo de *TriQuad*.

No capítulo 4 apresentamos os passos para reconstrução de superfícies, no sentido de encontrar uma formulação matemática para descrevê-la, a partir

de um conjunto de pontos. Para isto desenvolvemos um sistema para *fitting* (ajuste) de pontos inspirado em ferramentas como MPU de Ohtake et al (6) com uma partição uniforme do espaço (em tetraedros) sem a parte adaptativa, mas levando em consideração os vértices da malha localmente no *fitting*, e nas RBF's de Carr et al (5) construindo uma função de base radial tendo como restrições eventuais outros níveis da função $F^{-1}(J)$.

1.2 Contribuições

O *TetraQuad* pela maneira como foi formulado, construído por interpolação baricêntrica nos tetraedros, facilita o seu uso em aplicações que utilizem múltiplas partições do domínio (9, 6). Isso acontece porque ao subdividir um tetraedro inserindo novos vértices, o valor da quádrlica nesse vértice é obtido facilmente através das suas coordenadas baricêntricas.

Além disso, o *TetraQuad* apresenta uma série de vantagens do ponto de vista da geometria e topologia do objeto resultante em relação à utilização de triângulos para aproximação. Por termos acesso a forma implícita desses objetos, é simples obter informações como normais e curvaturas, fazer alterações na topologia e consultar pontos pertencentes à superfície. Esses fatos abrem a possibilidade para a sua aplicação em métodos de deformação de superfícies, *morphing* de objetos, construção de modelos do tipo CSG² (13), dentre outros.

As vantagens que acabamos de descrever esbarram na dificuldade em modelarmos um sistema para reconstruir superfícies a partir de *TetraQuads*. Por esse motivo, uma outra contribuição deste trabalho é a descrição de passos para modelagem de um sistema de equações para aproximar superfícies a partir de uma nuvem de pontos e suas normais baseada em abordagens conhecidas na literatura (5, 6). Fazemos uma descrição das dificuldades encontradas durante o processo e dos problemas ainda por resolver.

Para visualização implementamos um algoritmo de *ray casting* em GPU utilizando a decomposição dos tetraedros em triângulos. Uma abordagem semelhante já havia sido utilizada por Loop e Blinn (10) e uma implementação eficiente em paralelo aplicada na visualização de volumes é descrita no trabalho de Maximo et al (14).

²A sigla significa geometria sólida construtiva (*Constructive Solid Geometry*) que consiste na técnica de descrever superfícies a partir da combinação lógica de objetos.