

5 Resultados

5.1 Extração de silhueta

Zakaria e Seidel (15) propuseram o método *normal threshold* para nuvem de pontos, que define um ponto P da silhueta como um ponto cujo produto interno de sua normal com o vetor de visão do observador está suficientemente próximo de zero (Figura 5.1). Porém, a silhueta assim determinada pode apresentar falhas em algumas regiões e grande acúmulo de pontos em outras, como destacado na Figura 5.2.

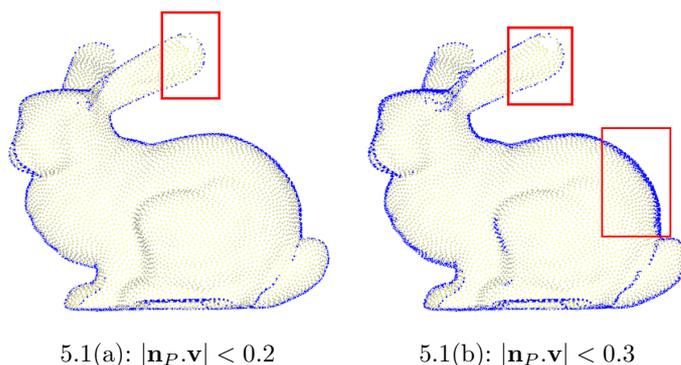


Figura 5.1: Notamos que, ao aumentar a tolerância para o produto interno, as falhas diminuem, entretanto aumentam os acúmulos de pontos em algumas regiões.

Além disso, esta silhueta é discreta, diferente da silhueta determinada pelo método que utilizamos que é uma curva (Figura 5.3). A Figura 5.4 mostra de outro ponto de vista os pontos que, pelo método *normal threshold*, pertencem às silhuetas da Figura 5.1. As imagens foram obtidas fixando a posição do observador e rotacionando o objeto.

Nosso objetivo central foi determinar a silhueta de uma superfície da qual conhecemos uma amostra de pontos, munidos de suas normais. Para tanto, reconstruímos localmente a superfície, utilizando dois métodos distintos, descritos nos Capítulos 3 e 4. Nas próximas seções, mostraremos mais exemplos de cada um desses procedimentos, gerados a partir de arquivos do tipo ply (8).

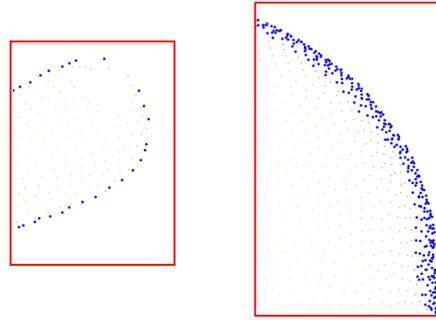


Figura 5.2: Imagem aproximada das partes destacadas na Figura 5.1(b). A silhueta apresenta falhas (esquerda) e acumulo de pontos (direita).

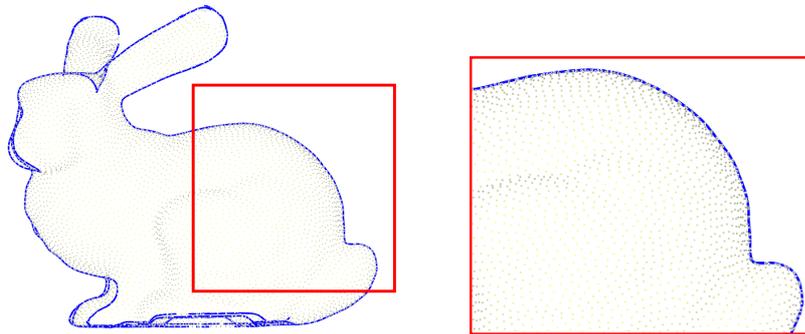
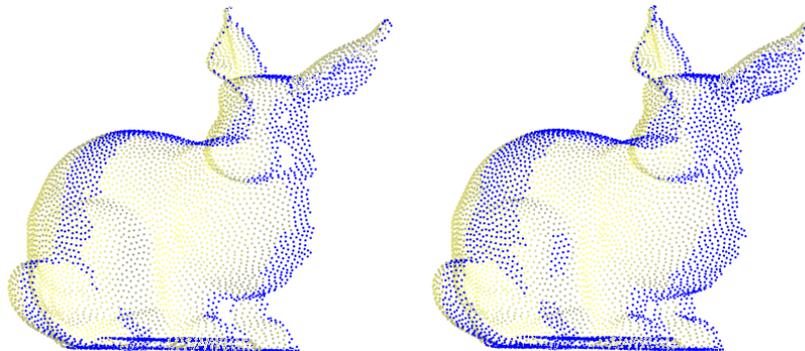


Figura 5.3: Silhueta gerada a partir de uma nuvem de pontos de 10020 vértices utilizando o método proposto no Capítulo 4, com $k = 15$, $\delta = 0.5$ e $D = 0.2$.



5.4(a): $|\mathbf{n}_P \cdot \mathbf{v}| < 0.2$

5.4(b): $|\mathbf{n}_P \cdot \mathbf{v}| < 0.3$

Figura 5.4: Foram plotados apenas pontos tais que $\mathbf{n}_P \cdot \mathbf{v} < 0$. Os pontos em azul são pontos da silhueta, pela definição de Zakaria e Seidel.

Os exemplos foram rodados numa máquina Mac OS X, processador 2.66 GHz Intel Core 2 Duo, com memória 4 GB 800 MHz DDR2 SDRAM.

5.1.1

Silhueta via reconstrução local de malha poligonal

Na etapa de pré-processamento, realizamos todos os procedimentos que independem da posição do observador. É nesta etapa que é feita a triangulação de Delaunay na vizinhança de cada ponto, o passo mais demorado deste método. Uma vez que definimos uma estrela para cada ponto da nuvem, há a necessidade de verificar se cada aresta deve de fato ser processada para a busca de pontos sobre a silhueta, como foi comentado na seção 3.2, teste que também pode ser feito nesta primeira etapa.

Como a determinação dos pontos sobre a curva silhueta depende do vetor observador, os únicos cálculos que devem ser processados durante a execução deste método são o produto interno da normal de cada vértice com o vetor observador e a busca de zeros deste produto nas arestas da triangulação. Isto é, em uma segunda etapa, estamos determinando a silhueta sobre uma malha poligonal, que é um processo simples (seção 2.1) e pode ser feito em tempo real.

Em superfícies que apresentam regiões de alta curvatura foi encontrada uma dificuldade na reconstrução, inerente ao método utilizado para tal. A Figura 5.5 mostra a silhueta da nuvem de pontos sobre a superfície do cavalo, sem usar o filtro do cone considerado na seção 3.1. Notamos que em locais de alta curvatura, como é o caso das patas do cavalo, algumas estrelas reconstruídas não correspondem à superfície original, possuindo triângulos no interior do objeto (Figura 5.6(a)). Assim, a curva silhueta, que é determinada processando estes triângulos, se apresenta com arestas indesejadas (Figura 5.6(b)).

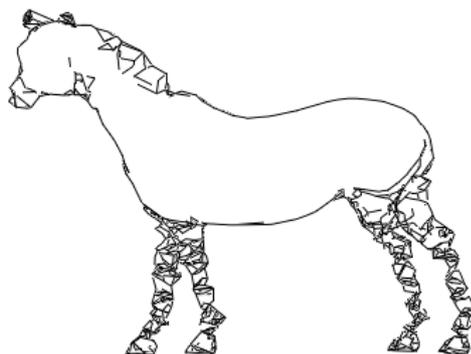
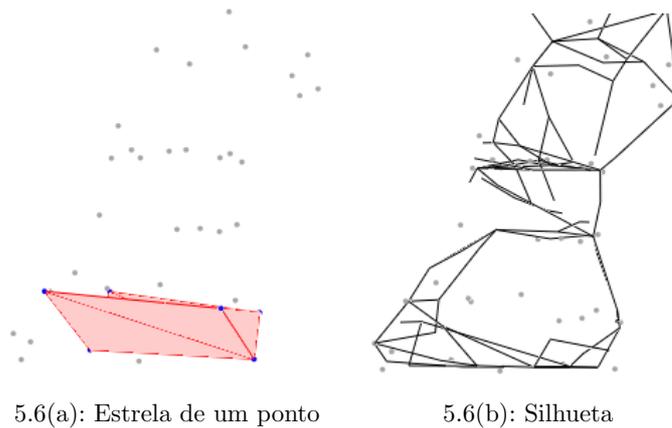


Figura 5.5: Silhueta de uma nuvem sobre a superfície do cavalo, com 927 vértices, sem utilizar o filtro e considerando uma vizinhança inicial de 16 pontos ($k = 16$).

Uma primeira estratégia foi reduzir o número de vizinhos considerados para a reconstrução. O problema da reconstrução de estrelas foi amenizado,

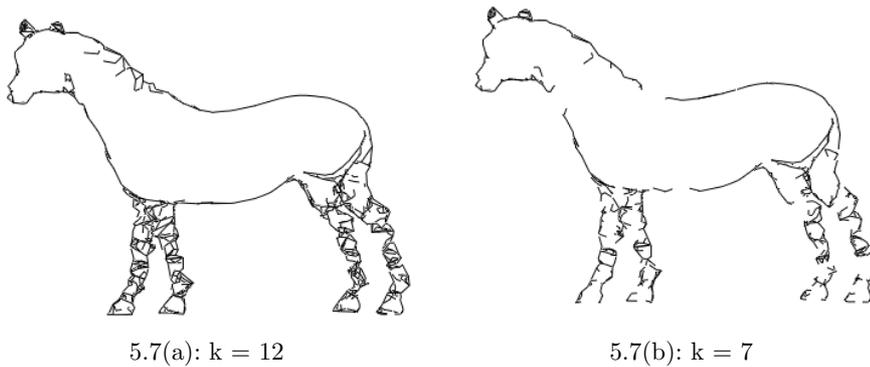


5.6(a): Estrela de um ponto

5.6(b): Silhueta

Figura 5.6: Sem o filtro, um exemplo de estrela reconstruída com $k = 16$ que não respeita a superfície do objeto (Figura 5.6(a)). A Figura 5.6(b) mostra mais de perto a consequência disto na silhueta da pata do cavalo.

como mostram as Figuras 5.7(a) e 5.7(b), com $k = 12$ e $k = 7$ respectivamente. Porém ocorrem falhas na silhueta de outras regiões.



5.7(a): $k = 12$

5.7(b): $k = 7$

Figura 5.7: Reduzindo o número de pontos vizinhos considerados na reconstrução, reduzimos o problema na pata, mas geramos falhas em outras regiões.

Logo, notamos que de fato o filtro proposto na seção 3.1 é importante para obtermos uma melhor reconstrução da malha. Com este filtro, para reconstruir a estrela de cada ponto P da nuvem, consideramos apenas os pontos Q tais que o ângulo θ entre a aresta \overline{PQ} e a normal n_P seja próximo de 90° . Diremos que $|\cos\theta| < \delta$ e mostraremos os resultados para diversos valores do parâmetro δ .

A Figura 5.8 mostra o resultado ao aplicarmos o filtro com $\delta = 0.7$ para diferentes valores de k .

Mesmo com o filtro utilizado (descrito na seção 3.1), em alguns casos, como no cavalo, não foram encontrados parâmetros que fossem suficientes para evitar estes problemas e ainda garantir uma boa reconstrução das outras regiões da nuvem, apresentando falhas na curva silhueta.

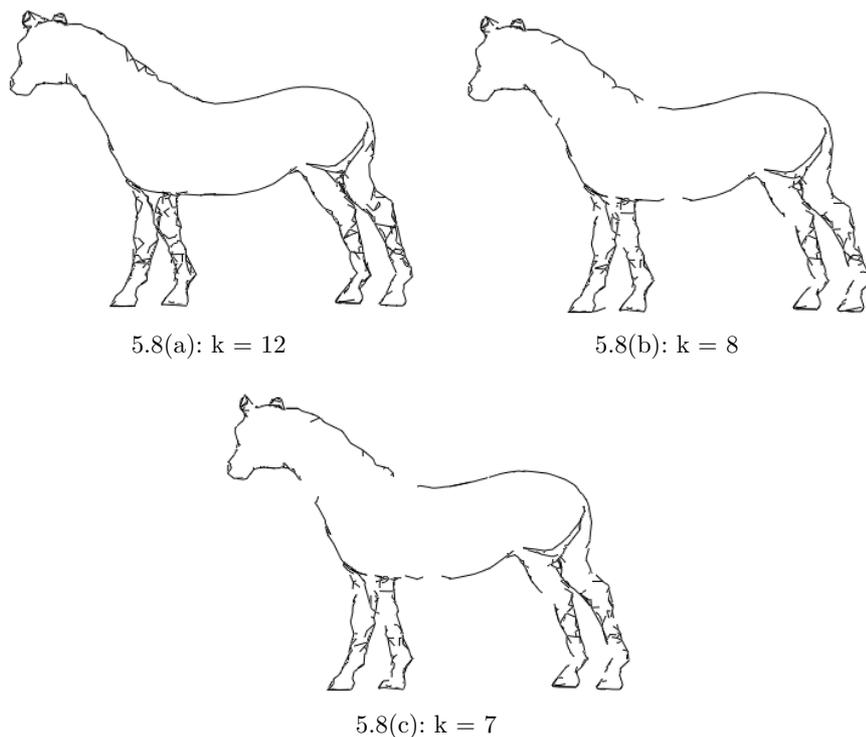


Figura 5.8: Com um mesmo valor de $\delta = 0.7$, reduzindo o valor de k , há a redução do problema nas patas, porém a silhueta apresenta falhas em outras regiões.

Observamos também que o sucesso deste método de reconstrução requer uma nuvem de pontos com uma boa amostragem. Em exemplos com poucos pontos, encontramos problemas na reconstrução da estrela, pois dificulta a determinação dos vizinhos, podendo acontecer de pontos que não são de fato vizinhos serem conectados. Porém, em nuvens com amostragem bem distribuída, este problema é bastante reduzido, como mostra a Figura 5.9. Neste exemplo, encontramos pequenos problemas na reconstrução apenas entre os dedos, o que pode ser visto com mais detalhes na Figura 5.10. Na seção 5.1.2, mostramos que, com outro método de reconstrução é possível utilizar um passo adaptativo para corrigir este tipo de problema.

Outro exemplo de uma nuvem bem amostrada pode ser visto na Figura 5.11, que mostra que um bom parâmetro pode ser escolhido para o filtro, gerando uma boa reconstrução das estrelas e reduzindo os problemas nas orelhas do coelho.

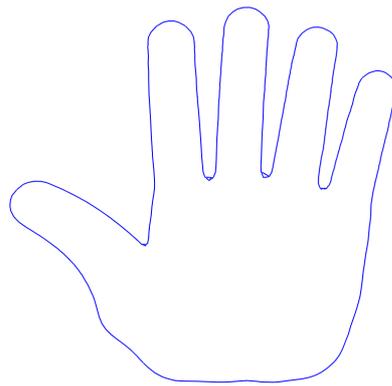


Figura 5.9: Em nuvens com melhor distribuição dos pontos, a reconstrução é bem sucedida. Esta nuvem possui 6191 vértices. Para a silhueta, usamos $k = 8$ e não foi utilizado o filtro do cone.

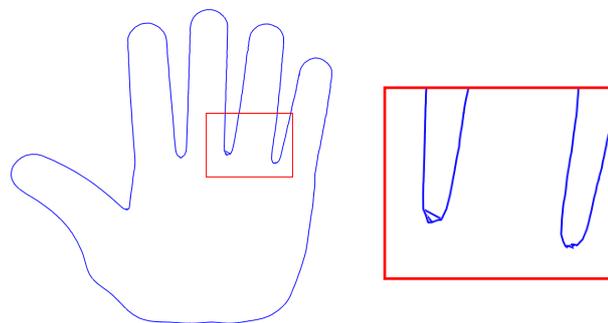
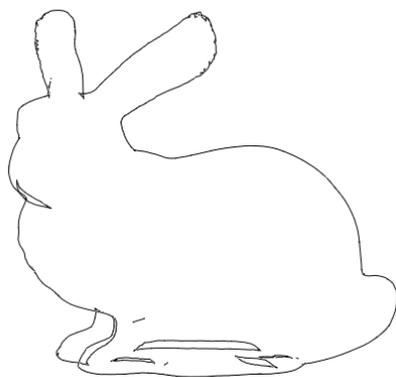
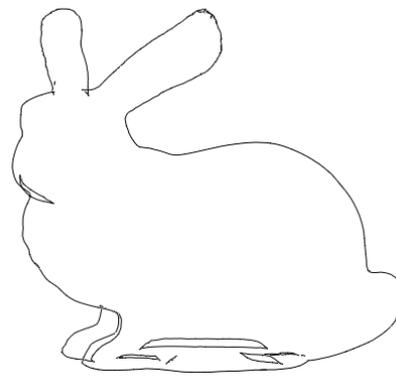


Figura 5.10: Silhueta, com $k = 15$ e $\delta = 0.5$. Mesmo com o uso do filtro, pequenos erros são encontrados na reconstrução das estrelas dos pontos que estão entre os dedos.



5.11(a): $\delta = 0.7$



5.11(b): $\delta = 0.5$

Figura 5.11: Silhueta da nuvem do coelho, com 10020 vértices.

5.1.2

Extração de silhueta via superfícies implícitas

Na etapa de pré-processamento, determinamos para cada ponto P da nuvem a sua *bounding box*. Para tanto, foram utilizados os k pontos mais

próximos de P que passam no teste do filtro do cone considerado na seção 3.1. Também nesta etapa é possível fazer a reconstrução local da superfície para cada *bounding box*. Entretanto, optamos por fazer a reconstrução local durante o percorrimento da curva.

Conforme comentamos na seção 4.2, para perseguir a curva silhueta na nuvem de pontos, inicialmente buscamos uma caixa que contenha algum segmento de silhueta. Em seguida, vamos pesquisar dentre as *bounding boxes* dos pontos desta caixa, quais ainda não foram visitadas e têm silhueta. Assim, precisamos reconstruir apenas a superfície nestas caixas. Uma vez que esta curva foi inteiramente percorrida, repetimos este procedimento um número definido de vezes, de forma a encontrar outras curvas silhuetas. Nos exemplos aqui apresentados, este número foi uma porcentagem B da quantidade de pontos da nuvem. As silhuetas das Figura 1.1, 5.3 e 5.12 foram determinada por meio desta primeira estratégia.

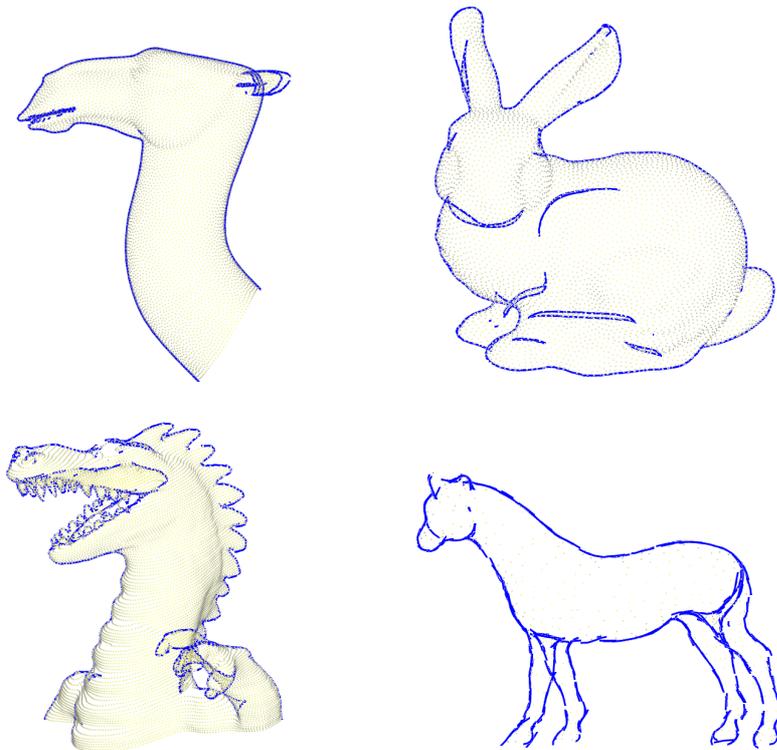


Figura 5.12: Exemplos de silhueta via reconstrução local de superfícies implícitas.

Entretanto, devido à auto-interseção de *bounding boxes* de pontos muito próximos, notamos que, procedendo dessa maneira, redesenhamos partes da curva silhueta diversas vezes. Por esse motivo, buscamos outras estratégias visando evitar processamento desnecessário da curva em regiões já visitadas. Uma vez estando em uma caixa bb_P com silhueta e sabendo que os k pontos que

geraram esta caixa estão já ordenados de acordo com suas distâncias ao ponto P , uma segunda estratégia foi buscar, dentre estes pontos, aquele que está mais distante de P e cuja *bounding box* apresente silhueta. Os demais pontos da caixa foram marcados como visitados, o que impossibilita que sejam escolhidos para uma nova busca de silhueta. Um exemplo pode ser visto na Figura 5.13, onde D representa a fração da diagonal da caixa que foi considerada como raio da bola (Figura 4.3).

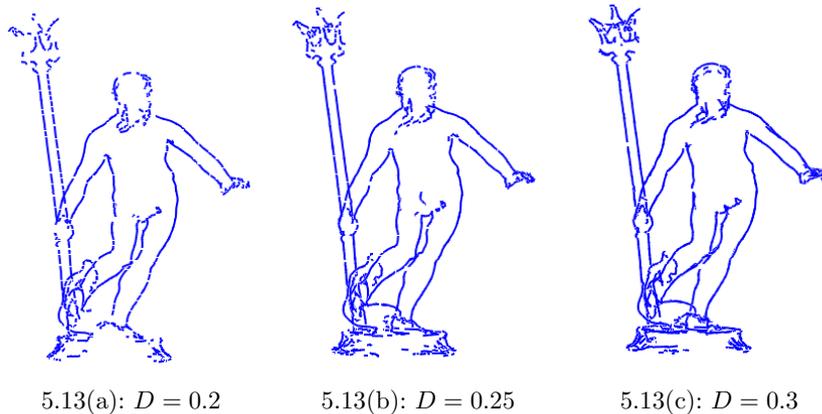


Figura 5.13: Estratégia 2. A busca por uma primeira caixa com silhueta foi feita segundo uma porcentagem $B = 20\%$ do número de vértices.

Uma terceira estratégia que implementamos foi buscar a silhueta na *bounding box* do ponto M cuja distância ao ponto P é a mediana das distâncias de P aos outros pontos da caixa. Caso seja encontrada silhueta, marcamos como visitadas as caixas dos pontos Q de bb_P que estão mais próximos de P do que M . Se não encontramos silhueta, buscamos nas caixas dos pontos que estão mais próximos de P , ou seja, se bb_P é determinada pelos pontos $P_1, P_2, \dots, P_m = M, \dots, P_k$, e bb_M não tem silhueta, buscamos em $bb_{P_{m-1}}$ e marcamos as caixas de P_1, P_2, \dots, P_{m-2} como visitadas. A Figura 5.14 mostra exemplos do uso desta estratégia.

No método proposto não priorizamos o tempo e como resultado observamos que nos exemplos das Figuras 5.13 e 5.14 o tempo de processamento foi em média 35 segundos. Para os exemplos da Figura 5.12 os tempos variam de 1 a 10 minutos. Esses tempos podem ser melhorados investindo em melhorias nas estratégias 2 e 3.

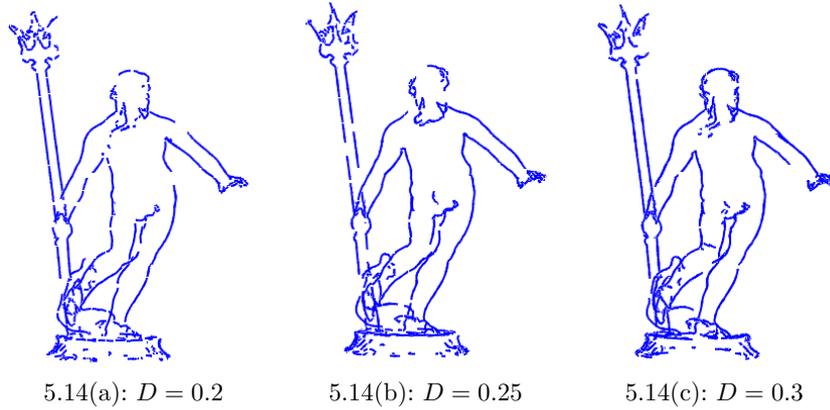


Figura 5.14: Estratégia 3. A busca por uma primeira caixa com silhueta foi feita segundo uma porcentagem $B = 20\%$ do número de vértices.

5.2 Silhueta de nuvens com ruídos

Os resultados que obtivemos indicam que, em geral, a reconstrução via superfícies implícitas (seção 4.1) gera melhores resultados de silhuetas de nuvens de pontos com ruídos, em comparação com o uso da reconstrução local de malha.

Para simular tais ruídos, partimos de um arquivo ply no qual estão disponíveis as coordenadas dos vértices, além de suas normais e a forma como estão conectados, e perturbamos os pontos.

Numa nuvem aparentemente sem ruídos (exemplo de silhueta na Figura 5.15), aplicamos uma perturbação nas coordenadas de todos os pontos: a cada coordenada x_i somamos $x_r \cdot p$, com $i = 1, 2, 3$, onde x_r é um número randômico entre -1 e 1 e p é um fator determinado pelo usuário.

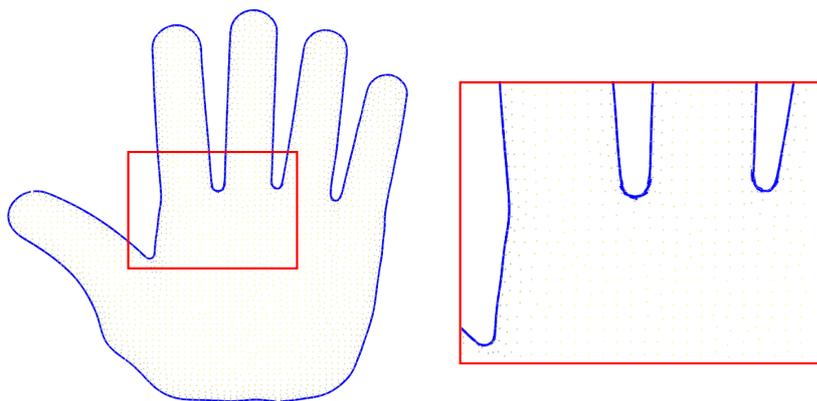


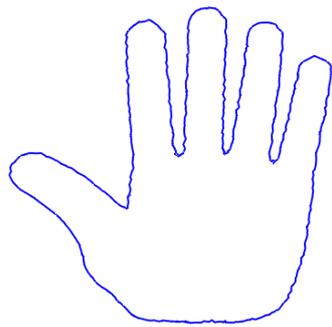
Figura 5.15: Silhueta da nuvem de pontos sem ruídos, com $k = 15$ e $D = 0.24$

Após a perturbação da nuvem, é necessário recalculer as normais, pois estas são dados de entrada para o método de reconstrução local da superfície

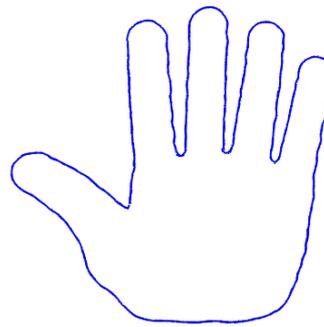
implícita. Para tanto, utilizamos a malha do modelo de onde foi extraída a nuvem de pontos.

De posse das normais, reconstruímos a superfície e determinamos a silhueta (Figura 5.16(b)).

Em nossos resultados observamos que o método de reconstrução via triangulação de Delaunay descrito na seção 3.1 é mais sensível a perturbações. Nas Figuras 5.16, 5.17, 5.18, 5.19, 5.20 e 5.21 estão exemplos nos quais visualmente notamos que o uso da reconstrução local de superfícies implícitas gera resultados com menor sensibilidade a ruídos.

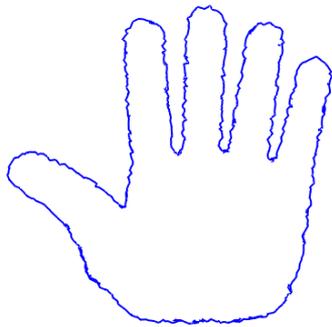


5.16(a): Reconstrução via triangulação de Delaunay

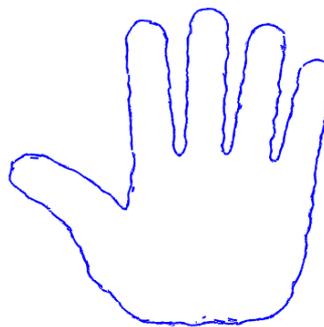


5.16(b): Reconstrução via superfícies implícitas

Figura 5.16: Comparação da silhueta para nuvem com ruídos, com $p = 0.007$.



5.17(a): Reconstrução via triangulação de Delaunay



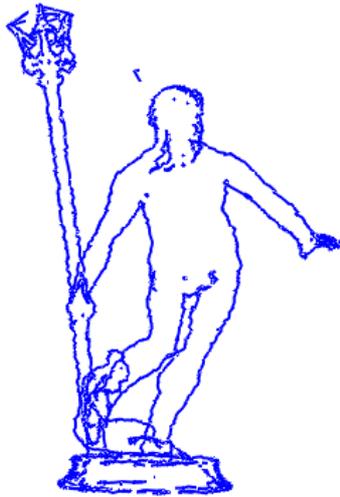
5.17(b): Reconstrução via superfícies implícitas

Figura 5.17: Comparação da silhueta para nuvem com ruídos, com $p = 0.013$.

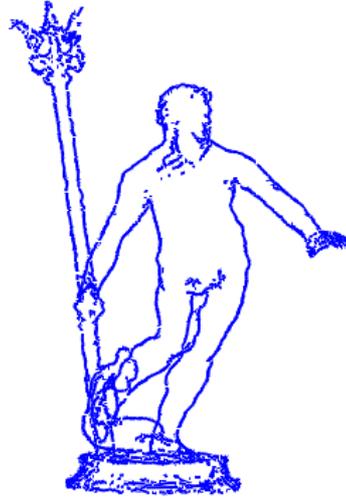
5.3

Extração de Linhas Laplacianas

A determinação das Linhas Laplacianas foi feita em uma nuvem de pontos reconstruída localmente via triangulação de Delaunay, segundo descrevemos na seção 3.1. Tal método, para cada ponto P da nuvem N , determina a estrela

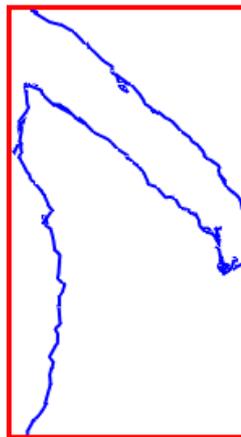


5.18(a): Reconstrução via triangulação de Delaunay



5.18(b): Reconstrução via superfícies implícitas

Figura 5.18: Comparação da silhueta gerada para nuvem com ruídos, com $p = 0.006$.



5.19(a): Reconstrução via triangulação de Delaunay



5.19(b): Reconstrução via superfícies implícitas

Figura 5.19: Detalhes da Figura 5.18.

$star(P)$ e é realizado em etapa de pré-processamento. Ainda nesta etapa, utilizamos estas estrelas para criar uma lista L de triângulos, como foi descrito na seção 3.3 e, assim, calculamos o Laplaciano Δn_P .

Durante o processamento, para cada triângulo da lista L , procedemos semelhante à determinação da silhueta: em cada vértice P_i do triângulo, calculamos o produto interno $\Delta n_{P_i} \cdot v = \Delta I(P_i)$, sendo v o vetor do observador, buscamos zeros do Laplaciano da iluminação nas arestas do triângulo e, assim, determinamos uma aresta da Linha Laplaciana. Por esse motivo, a extração

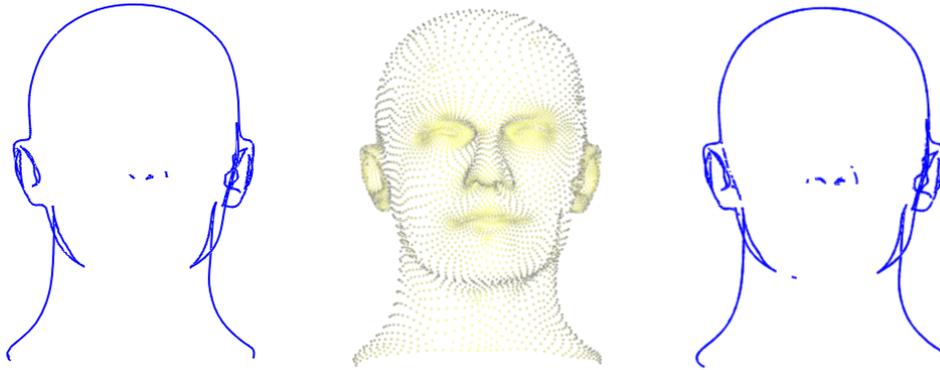


Figura 5.20: Esquerda: silhueta da reconstrução via triangulação de Delaunay; centro: nuvem de pontos sem ruído; direita: silhueta da reconstrução via superfícies implícitas.



Figura 5.21: Esquerda: silhueta da reconstrução via triangulação de Delaunay; centro: nuvem de pontos com ruído $p = 0.025$; direita: silhueta da reconstrução via superfícies implícitas.

das Linhas Laplacianas é feita em tempo real.

Estratégias para o cálculo do Laplaciano das normais

Para o cálculo de Δn_P , inicialmente adotamos uma primeira estratégia diferente do que mostramos na seção 3.3. Uma vez que o Laplaciano proposto em Zhang (16) prioriza os pontos que estão mais próximos de P e sabendo que reconstruímos apenas a estrela deste ponto, na primeira estratégia utilizamos para o cálculo do Laplaciano de n_P apenas pontos que estão na estrela de P .

Assim, para o cálculo de Δn_P , consideramos apenas os triângulos de $star(P)$:

$$L^h f(\mathbf{P}) = \frac{1}{4\pi h^2(\mathbf{P})} \sum_{\Delta_i \in star(P)} \frac{A(\Delta_i)}{3} \sum_{\mathbf{Q} \in \Delta_i} e^{-\frac{\|\mathbf{Q}-\mathbf{P}\|^2}{4h(\mathbf{P})}} (f(\mathbf{Q}) - f(\mathbf{P})).$$

Contudo, observamos um número excessivo de linhas nos resultados (Figura

5.22).

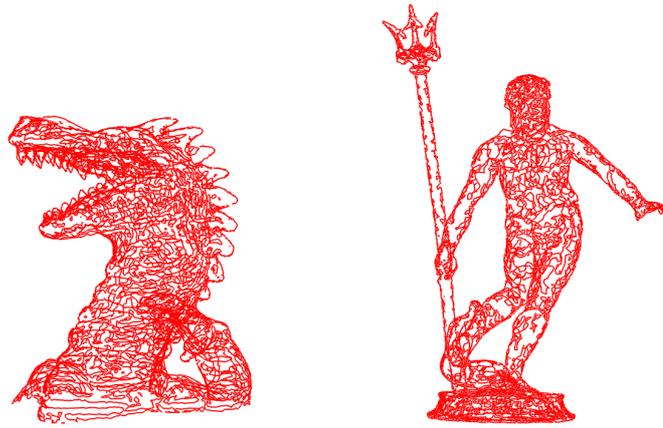


Figura 5.22: Linhas Laplacianas do dragão e do Netuno, considerando para o cálculo do Laplaciano apenas a estrela de cada ponto como vizinhança.

Por esse motivo, propomos uma segunda estratégia que foi criar a lista de triângulos, como descrito na seção 3.3, e utilizar essa lista para o cálculo do Laplaciano das normais. As Figuras 5.23 e 5.24 mostram exemplos obtidos utilizando esta estratégia. Na Figura 5.23, observamos que um menor valor de h fornece maior quantidade de linhas.

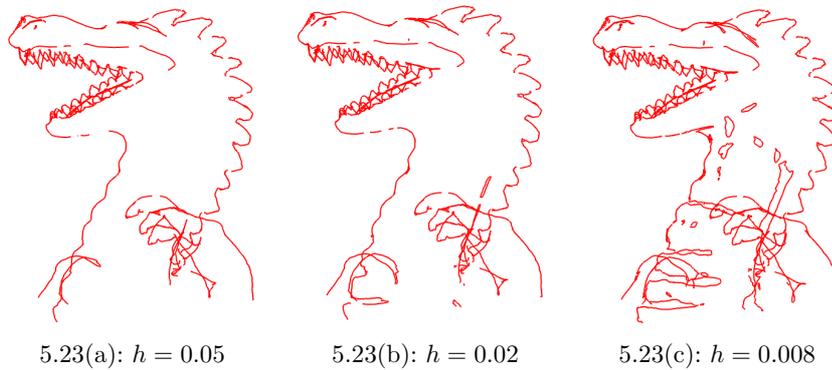
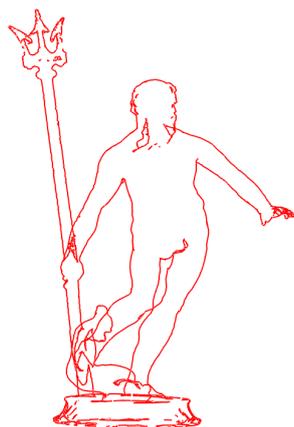


Figura 5.23: Linhas Laplacianas para diferentes valores de h .

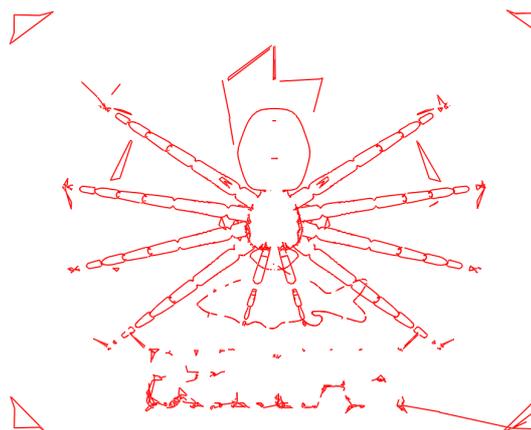
As Linhas Laplacianas exibem mais detalhes do que a silhueta. Em nossos experimentos obtivemos bons resultados na extração dessas linhas na nuvem de pontos.

5.4 Linhas Laplacianas de nuvens de pontos com ruídos

A princípio, como o Laplaciano possui efeito de suavização, esperávamos que as Linhas Laplacianas seriam visivelmente menos sensíveis a ruídos do que



5.24(a): $h = 0.008$

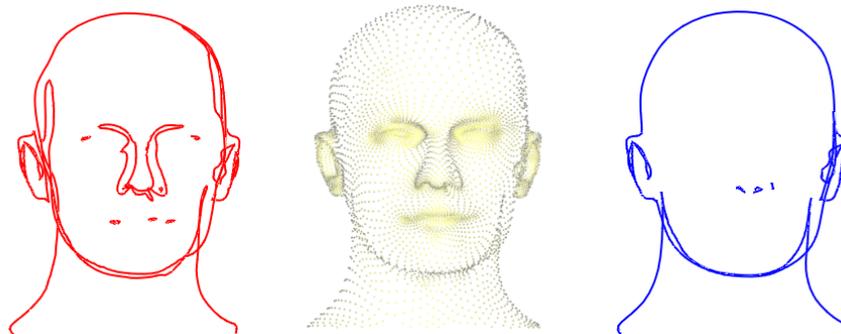


5.24(b): $h = 0.008$

Figura 5.24: Linhas Laplacianas, com $h = 0.05$.

a curva silhueta. Porém, experimentalmente verificamos que isto não procede no caso de reconstrução local de malha via triangulação de Delaunay. Nas Figuras 5.25 e 5.26 vemos exemplos de Linhas Laplacianas e silhuetas de uma mesma nuvem, antes e depois de uma perturbação que simula ruídos.

Observando resultados como estes, notamos que as Linhas Laplacianas se mostram tão sensíveis a ruídos quando a silhueta quando determinadas após uma reconstrução local de malha.

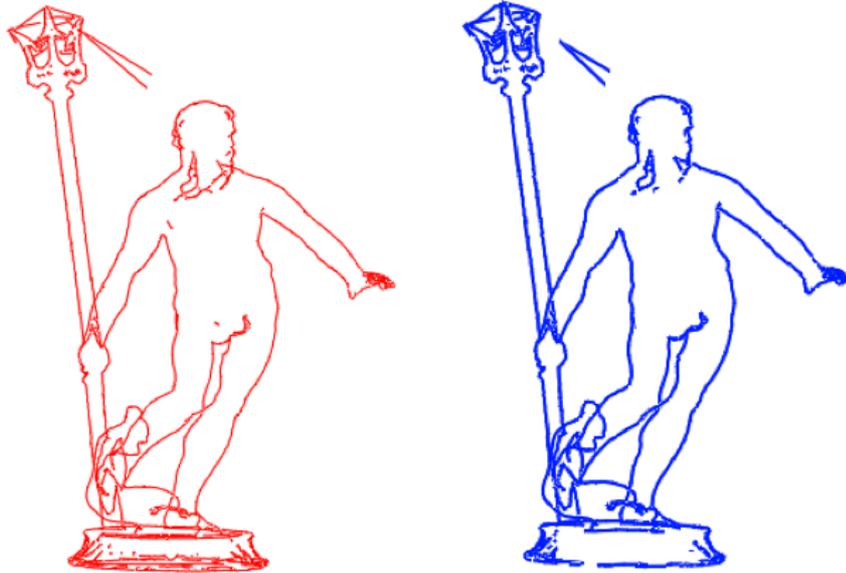


5.25(a): Linhas laplacianas (esquerda), nuvem (centro) e silhueta (direita) sem perturbação

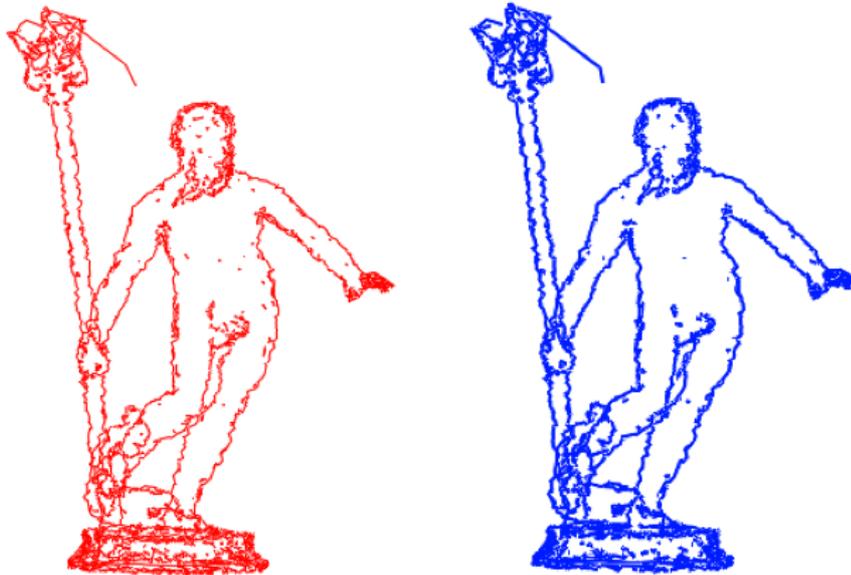


5.25(b): Linhas laplacianas (esquerda), nuvem (centro) e silhueta (direita) após perturbação, com $p = 0.01$

Figura 5.25: Efeitos da perturbação da nuvem nas Linhas Laplacianas, com $h = 0.07$, e silhueta.



5.26(a): Sem perturbação



5.26(b): Após perturbação, com $p = 0.01$

Figura 5.26: Em vermelho, as Linhas Laplacianas, com $h = 0.01$, e, em azul, a silhueta.