

RENATO VOGELAAR

RECONSTRUÇÃO 3D

Texto apresentado à Escola Politécnica da Universidade de São Paulo como requisito para a conclusão do curso de graduação em Engenharia, junto ao Departamento de Engenharia Mecatrônica (PMR).

São Paulo
2012

RENATO VOGELAAR

RECONSTRUÇÃO 3D

Texto apresentado à Escola Politécnica da Universidade de São Paulo como requisito para a conclusão do curso de graduação em Engenharia, junto ao Departamento de Engenharia Mecatrônica (PMR).

Área de Concentração:
Engenharia Mecatronica

Orientador:
Prof Dr Marcos de Sales Guerra Tsuzuki

São Paulo
2012

FICHA CATALOGRÁFICA

Vogelaar, Renato

Reconstrução 3D/ Vogelaar. São Paulo, 2012.

85 p.

Monografia (Graduação em Engenharia) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica (PMR).

1. Visão Computacional #1. 2. ICP #2. 3. Nuvem de Pontos #3.
I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica (PMR). II. t.

AGRADECIMENTOS

Agradeço Prof. Dr. Marcos de Sales Guerra Tsuzuki pela orientação durante a realização deste trabalho e Rogério Takimoto por sua ajuda na pesquisa e teste do protótipo.

Gostaria também de agradecer minha família pelo apoio e recursos que permitiram a realização deste projeto, Flavia Takahi pelo apoio e ajuda nos testes e Zeca pela companhia nas noites em claro.

RESUMO

O processo de reconstrução 3D de um objeto pode ser resumido em três partes principais: Obtenção da nuvem de pontos, alinhamento da nuvem de pontos, reconstrução da malha a partir da combinação de diversas malhas. Na primeira parte, a nuvem de pontos é obtida pelo Kinect, que gera um mapa de profundidade pelo seu sensor infravermelho. O alinhamento de nuvens de pontos utiliza o algoritmo ICP (Interactive Closest Point). Nele o método dos mínimos quadrados é aplicado para que a soma do quadrado das distâncias entre os pontos de duas nuvens seja a menor possível. Após o alinhamento das várias nuvens de pontos, para cada nuvem de pontos uma malha é reconstruída utilizando pontos adjacentes. A técnica de zippering é utilizada para juntar as malhas parcialmente sobrepostas sem aumentar o número de vértices ou triângulos da malha final. Esta técnica consiste em recortar as malhas e costurar pontos não sobrepostos na malha final. Após a malha final ter sido criada, seus vértices são movidos na direção de sua normal a fim de se posicionar na interseção de seu vetor normal com todas as malhas parciais, criando assim uma malha cuja superfície é uma média de todas as malhas parciais.

ABSTRACT

The process of 3D reconstruction of an object can be summarized into three main parts: point cloud acquisition, alignment of different point clouds and mesh reconstruction from the combination of several meshes. In the first part, the point cloud is obtained using the Kinect, which generates a depth map from its infrared sensor. The alignment of point clouds uses the ICP algorithm (Interactive Closest Point). The least squares method is applied so that the sum of the squared distances between two point clouds is the lowest possible.

After alignment of multiple point clouds, each point cloud is reconstructed using its adjacent points. The zipping technique is used to attach the partially overlapping meshes without increasing the number of triangles or vertices of the final mesh. non-overlapping part of each mesh are cutted and sewed together in the final mesh. After the final mesh has been created, its vertices are moved along its normal to the average position of all intersection of its normal vector with all partial meshes, creating a mesh whose surface is an average of all partial meshes.

SUMÁRIO

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

Lista de Símbolos

1	Introdução	15
1.1	Objetivos	17
2	Revisão da literatura	19
2.1	Pipeline	19
2.2	Obtenção de Mapas de Profundidade	20
2.2.1	Sensores de Contato	20
2.2.2	Sensores de Tempo de Voo (Time-of-Flight)	21
2.2.3	Sensores de Mudança de Fase	21
2.2.4	Triangulação	21
2.2.5	Sensor Fotométrico	21
2.2.6	Silhueta	22
2.2.7	Estereoscópica	23
2.2.8	Sensores Baseados em Luz Estruturada	23

2.3	Registro das Nuvens de Pontos	26
2.3.1	Signed Distance Fields	26
2.3.2	Interactive Closest Point	27
2.3.3	Point to Plane	30
2.4	Refino e Compensação de Erros	33
2.4.1	Metodo de Turk e Levoy	33
2.4.2	Distâncias com Sinal	33
2.5	Reconstrução da Malha	33
2.5.1	Zippering	34
2.5.2	Marching Cubes	35
2.6	Pós-Processamento	36
3	Metodologia	38
3.1	Pipeline escolhido	38
4	Sensor para Aquisição	39
4.1	Especificações do Kinect	39
4.2	Redução do Ruído do Mapa de Profundidade	42
4.3	Smoothing	43
5	Registro da Nuvem de Pontos	45
5.1	Métodos para Aumentar a Robustez do ICP	46
5.1.1	Método Estatístico para Eliminação de Outliers	47

5.1.2	Movimento de Corpo Rígido para Eliminação de Outliers	48
5.1.3	Truncamento	49
5.1.4	Aceleração da Convergência por Extrapolação	51
5.1.5	Utilização dos Dados de Cores (Color ICP)	53
6	Reconstrução de malha	56
6.1	Calculo do Isoplano	56
7	Pós Processamento e Formato Exportado	58
8	Resultados	59
9	Captura da Nuvem de Pontos Utilizando o Kinect	60
9.1	Discretização da Profundidade	60
9.2	Densidade de Pontos	61
9.3	Erros	62
9.4	Filtragem	64
9.5	Erros de Cores	65
10	Etapa de Registro Utilizando ICP e Variantes	68
10.1	Tempo de Iteração	68
10.2	Problemas de Convergência	71
10.3	Color ICP	72
10.4	Threshold de Cor	73
10.5	Métodos Estatístico e Trimmed	74

11 Reconstrução da Malha Utilizando Marching Cubes e Distância com Sinal	76
11.1 Criação da Iso-Superfície	76
11.2 Adaptative Marching Cubes	77
12 Conclusão	80
Referências	82

LISTA DE ILUSTRAÇÕES

1	Exemplo de um mapa de profundidade, os pixels mais claros representam uma distância maior da câmera ^[1]	16
2	À esquerda: diversas nuvens de pontos alinhadas. À direita: diversas malhas alinhadas ^[2]	16
3	Princípio de triangulação ^[3]	22
4	Princípio da fotometria ^[4]	22
5	Princípio do scanner por silhueta ^[5] . a) Imagem b) subtração do fundo c) silhueta d) estrudamento e) modelo.	23
6	Reconstrução a partir de um padrão de faixas de cores ^[6]	24
7	Codificação M-array baseada em três símbolos proposta por Griffin ^[7] , onde o ponto é decodificado utilizando sua vizinhança ^[8]	25
8	Distância do ponto ao plano entre duas superfícies ^[9]	31
9	representação da distância $d_i(v)$ ^[10]	34
10	Isosuperfície resultante de duas malhas ^[10]	34
11	Método de zipping para juntar duas malhas ^[11]	35
12	Analogia 2D da técnica de marching cubes ^[12]	36
13	Os 15 tipos de cubos utilizados na técnica marching cubes ^[12]	36
14	Sensor Kinect ^[13]	40
15	Imagem infravermelha da projeção. Pode-se ver as 9 repetições do padrão	40

16	Raios infravermelhos com diferentes orientações e formatos em função da distância ^[14]	41
17	Astigmatismo da lente e distâncias focais diferentes ^[14]	41
18	Filtro mediano ^[15]	42
19	Comparação de diferentes métodos de registro ^[16]	45
20	Duas nuvens de pontos (uma delas em mostrada em uma linha contínua por propositos de diferenciacao) convergem para um minimo local ^[17]	46
21	Problemas de nuvens com ruído e nuvens parcialmente sobrepostas no ICP ^[18]	47
22	Esquerda Acima: Vista de cima de duas nuvens de pontos, uma tracejada e outra contínua (pontos ligados para fins de diferenciação) Direita acima: Vista lateral da mesma imagem Esquerda abaixo: Pareamento dos pontos mais próximos, vemos claramente diversos outliers Direita abaixo: Após a remoção de pontos pelo método estatístico ^[17]	49
23	Comparação entre a composição de duas malhas que se sobrepõe parcialmente, a direita com TrICP e a esquerda com ICP com pareamento de todos os pontos ^[19]	50
24	Tempo de convergência para diferentes métodos de eliminação de outliers ^[18]	51
25	Uma direção constante nos permite a extrapolação do movimento e aceleração do método ICP ^[20]	53
26	Uma direção constante Diferença entre o ICP e color ICP ^[21]	54

27	Com a transformação inversa, acha-se a coordenada i e j no mapa de profundidade, subtraindo-se a distância do mapa de profundidade com a distância do voxel obtemos a distância com sinal d representado pelo segmento vermelho.	57
28	Codificação da distância em bits utilizada pelo Kinect.	61
29	Observa-se que quanto maior a distância mais evidente fica a discretização da distância.	62
30	Erros na medição. Acima: malha de uma obtida a 0.8m do sensor. Abaixo: malha da mesma parede obtida a 3.9m do sensor.	63
31	Erros na medição da nuvem de pontos vista de frente e cima. Os quadrados vermelhos mostram pontos com alta imprecisão da nuvem de pontos em relação ao modelo real.	64
32	Problemas na medição de objetos especulares ^[22]	65
33	a) nuvem de pontos colorida a 0.9 metros do sensor b) Malha gerada c) malha com filtro mediano de 5x5 pixels d) malha com filtro mediano de 5x5 pixels e duas passagens com filtro binomial de 5x5 pixels	66
34	a) nuvem de pontos colorida a 2.5m do sensor b) Malha gerada c) malha com filtro mediano de 5x5 pixels d) malha com filtro mediano de 5x5 pixels e duas passagens com filtro binomial de 5x5 pixels	67
35	À esquerda tem-se uma nuvem de pontos.À direita tem-se a combinação de 86 nuvens amostradas com 5000 pontos cada. .	67
36	Resultado do ICP para duas nuvens de pontos, à esquerda malhas não alinhadas e à direita malhas após 50 iterações.	68

37	a) Nuvem de pontos com 87.000 pontos, b) nuvem de pontos com 74.000 pontos, c) combinação de 86 nuvens de pontos, cada nuvem amostrada em 5000 pontos. d) Detalhes mostrando a convergência precisa das 86 nuvens.	69
38	O tempo de procura do ponto mais próximo é o principal fator no tempo total da iteração	70
39	Nota-se a diminuição do tempo de procura quando as malhas ficam mais próximas a cada iteração.	70
40	Erro do registro usando ICP para diversas amostragens.	71
41	Erro do registro causado por um grande plano	71
42	Testes mostrando convergências erradas causada por grandes planos	73
43	Convergência do ICP clássico e o Color ICP	74
44	Falha completa do Color ICP, Vemos as malhas antes(esquerda acima) e depois (direira acima) da iteração. A imagem abaixo mostra a captura da imagem a cores correspondente a uma das malhas.	75
45	Comparação entre o método ICP clássico (esquerda) e o ICP com Threshold de cor (direita). A Ilustração abaixo mostra as tentativas de um ponto achar o par mais próximo mas destros limites de diferença de cor impostas	75
46	cálculo da iso-superfície por distância com sinal	77
47	Reconstrução de malha de uma aquisição e 86 aquisições	78
48	Reconstrução de um teclado com uma aquisição e 43 aquisições	79

49	Reconstrucao de uma cena.	79
----	-----------------------------------	----

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

USP Universidade de São Paulo

ToF Time-of-Flight

PCA Principal component analysis

ICP Interactive Closest Point

SVD Singular Value Decomposition

LISTA DE SÍMBOLOS

$\Delta(h)$ Assinatura diádica

1 INTRODUÇÃO

A reconstrução 3D de geometrias é uma área de interesse em visão computacional^[23]. Suas aplicações variam de mapeamentos de interiores para arquitetura, localização e mapeamento em robótica, reconstrução de objetos arqueológicos para museus virtuais^[3], escaneamento aéreo de terreno^[24], realidade aumentada, jogos e filmes^[21].

As primeiras reconstruções 3D eram limitadas, mas avanços na capacidade computacional e poder de processamento permitiram que a reconstrução densa de pontos^[23] na ordem de milhões de pontos fosse possível. Apesar de diferentes métodos, um fluxo comum para aquisição de dados e geração de um modelo 3D é quase sempre seguida^[3].

Os hardwares utilizados para a aquisição dos dados estão cada vez mais baratos, e novas técnicas exploram o processamento computacional para eliminar erros, deixando a reconstrução 3D acessível a um público cada vez maior. Na maioria dos casos a saída de um sensor de profundidade é um mapa de profundidade, que é uma imagem em escalas de cinza ou coloridas onde o valor de cada pixel representa a distância da superfície da cena ao sensor como mostra a Figura 1.

Sabendo a característica do sensor, pode-se relacionar a posição de cada pixel e sua profundidade com um ponto no espaço real, gerando-se uma nuvem de pontos. Um mapa de profundidade pode ser triangulado para gerar

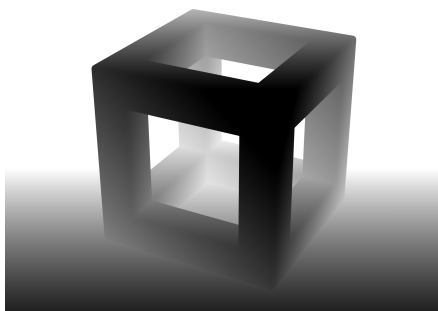


Figura 1: Exemplo de um mapa de profundidade, os pixels mais claros representam uma distância maior da câmera^[1].

uma malha que cobre parcialmente o objeto a ser reconstruído pela conexão dos pixels adjacentes. Sensores de profundidade que não geram um mapa de profundidade criam uma nuvem de pontos não organizada, e métodos para a triangulação são necessários.

Para uma reconstrução completa, diversas nuvens de pontos devem ser obtidas de diferentes pontos de vista para que partes do objeto obstruídas sejam obtidas e diminuir erros e ruídos causados por diversas características do sensor. Estas nuvens devem ser alinhadas, em um processo chamado registro como mostra a Figura 2, técnicas como o ICP são usadas para este fim.

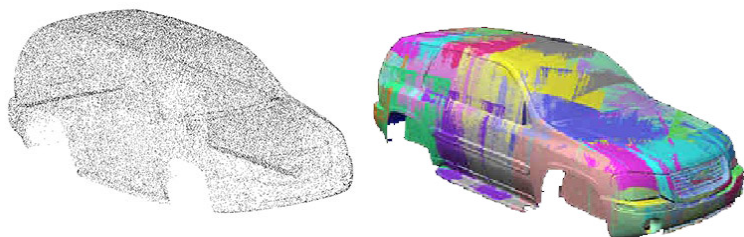


Figura 2: À esquerda: diversas nuvens de pontos alinhadas. À direita: diversas malhas alinhadas^[2].

Uma malha única é gerada a partir destas nuvens e depois otimizada para um uso específico.

1.1 Objetivos

O Objetivo principal do projeto é desenvolver uma ferramenta de reconstrução 3D com cores que seja simples, de baixo custo e acessível. Para isso diversos sensores, técnicas de registro e construção de malha foram pesquisados, comparados e um pipeline desenvolvido.

A reconstrução 3D esta voltada para a precisão, tentando aproximar ao Maximo do objeto reconstruído. O tamanho mínimo do objeto deve ser por volta de 1 metro, a fim de poder obter uma nuvem de pontos com pontos suficientes para reconstruir seus detalhes.

Reconstrução de cenas também será desenvolvida, com a limitação de serem ambientes internos, pois a luz solar gera interferência nos sensores lasers e infravermelhos criando problemas fora do escopo do projeto.

O projeto também ira dar ênfase a sensores baratos, mas sem diminuir a precisão proposta, para um acesso a mais pessoas e uma facilidade maior de se reproduzir os resultados.

A cor também é um ponto importante. Apesar de o projeto não propor a texturização da malha final, valores de cor serão atribuídas aos vértices, com isso criando-se uma reconstrução colorida, com cada superfície do triangulo sendo colorida pela interpolação de cores de seus vértices.

Esta monografia esta estruturada de seguinte maneira, no capítulo 2 é apresentado o pipeline comum e as diversas implementações da literatura para os diferentes módulos do pipeline. No capítulo 3 é apresentado a metodologia utilizada com as implementações selecionadas para cada modulo do pipeline. No capítulo 4 são apresentados os resultados preliminares aonde foram identificado diversos problemas. No capítulo 5 são apresentadas as so-

luções para os problemas encontrados. No capítulo 6 são apresentados os resultados finais e no capítulo 7 a conclusão.

2 REVISÃO DA LITERATURA

Inicialmente o fluxo comum utilizada na literatura é apresentada. Em seguida varias propostas para cada modulo são apresentadas.

2.1 Pipeline

A reconstrução 3D de um objeto normalmente segue os seguintes passos^[3]:

1. Obtenção de mapas de profundidade: a obtenção dos mapas de profundidade pode ser feita por diversos tipos de técnicas e sensores.
2. Registro das nuvens de pontos: nesta etapa mapas de profundidades de diferentes pontos de vista são registrados no espaço real e alinhados.
3. Refino e compensação de erros: após o alinhamento, erros devidos à imprecisão no registro, ruído no mapa de profundidade ou pequenos erros de calibração do sensor devem ser compensados.
4. Integração das nuvens em uma malha única: uma única malha deve ser gerada a partir da nuvem de pontos trianguladas ou não.
5. Pós-processamento e parametrização: após a geração da malha, operações são feitas para possibilitar a utilização da malha do objeto reconstruído.

A seguir, diversas propostas existentes na literatura para implementar cada um dos passos são apresentadas

2.2 Obtenção de Mapas de Profundidade

A nuvem de pontos pode ser obtida por diversas técnicas que podem ser classificadas em dois grandes grupos: contato e não-contato^[3]. Os sensores de não-contato podem ser subdivididos em sensores ativos e passivos.

Os sensores de contato tocam no objeto e armazenam as coordenadas do ponto de contato, como a máquina de medição de coordenadas^[25], os sensores ativos emitem radiação ou um pulso de energia para conseguir obter a nuvem de pontos, como sensores ToF (time-of-flight), de mudança de fase (phase shift), de triangulação e de luz estruturada. Os sensores passivos se utilizam da luz ambiente, como a técnica estereoscópica, de silhueta e fotométrica.

2.2.1 Sensores de Contato

Os sensores de contato utilizam um braço com um sensor na ponta, quando este toca o objeto, a posição do ponto de contato é registrada. A aquisição de pontos pode ser demorada na ordem de 10^3Hz ^[25], pois a máquina se move e registra cada ponto. Esta lentidão é compensada pela precisão da medida e pela falta de ruído na nuvem final.

Um problema desta máquina é não conseguir registrar a geometria de objetos relativamente macios ou objetos frágeis, como artefatos arqueológicos ou um rosto humano. A máquina de medição de coordenadas é um exemplo de sensor de contato.

2.2.2 Sensores de Tempo de Voo (Time-of-Flight)

Esses tipos de sensores utilizam um pulso de luz e contabilizam o tempo de seu retorno. Sabendo a velocidade na qual este pulso se propaga, sabemos a distância que ele percorreu. Apenas um ponto pode ser registrado por vez, com auxílio de espelhos e prismas rotativos a frequência de registro de pontos é da ordem de 10^5Hz ^[26].

2.2.3 Sensores de Mudança de Fase

Outro método de medição é por modulação de sinal^[27]. Ao invés de um pulso, uma onda contínua é emitida com sinal modulado, normalmente modulação de amplitude^[24], essa técnica deixa a aquisição de dados rápida, a ordem de 10^6Hz . Sua desvantagem é o alcance^[26] curto.

2.2.4 Triangulação

O sensor de triangulação também utiliza laser, mas diferente dos sensores time off flight, ele é utilizado para triangular o ponto iluminado com auxílio de uma câmera como mostra a Figura 3. Sabendo as características e posição da câmera que captura este ponto laser, podemos triangular a posição de cada ponto do objeto que o laser reflete.

2.2.5 Sensor Fotométrico

Este sensor utiliza diversas imagens de um objeto, cada uma com uma iluminação diferente, assumido uma refletância difusa ou lambertiana, pode-se utilizar-se da formula $I_D = L.N.I_L$ aonde I_D é a intensidade da luz refletida, N é a normal da superfície e I_L é a intensidade da luz emitida para descobrir a normal da superfície refletida como mostra a Figura 4.

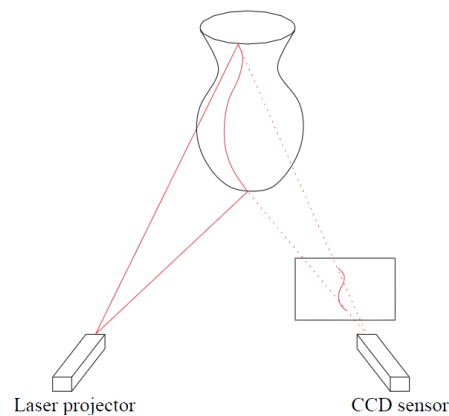


Figura 3: Princípio de triangulação^[3].

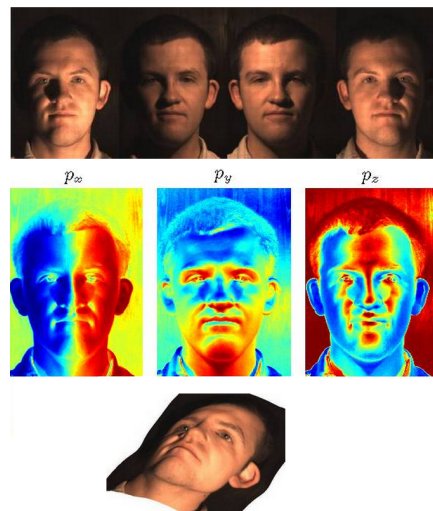


Figura 4: Princípio da fotometria^[4].

2.2.6 Silhueta

Esta técnica consiste em analisar diversas imagens de um objeto sobre uma mesa giratória em um fundo contrastante, obter a silhueta subtraindo-se o fundo e estrudando-as. Com diversos pontos de vista o objeto pode ser reconstruído como mostra a figura 5. A vantagem é a simplicidade do método e do hardware, apenas uma câmera e um computador e uma mesa giratória. Uma das desvantagens desta técnica é a perda de informações de objetos côncavos^[5].

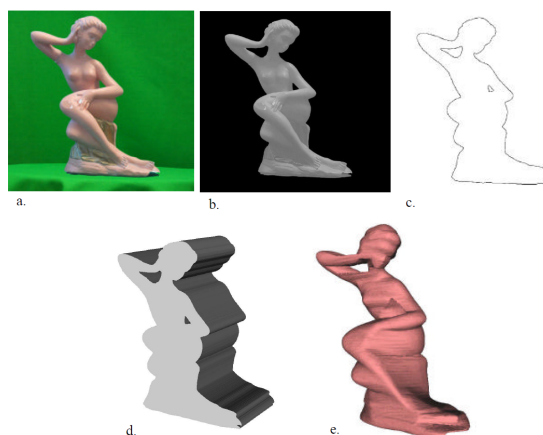


Figura 5: Princípio do scanner por silhueta^[5]. a) Imagem b) subtração do fundo c) silhueta d) estrudamento e) modelo.

2.2.7 Estereoscópica

Esta técnica utiliza várias câmeras ou diferentes imagens de uma câmera para reconstruir o objeto. Seu funcionamento é comparado ao de um olho humano. Pontos característicos são determinados em cada imagem e estes correlacionados com diferentes métodos para determinar a geometria epipolar (REFERENCIA Tsuzuki).

Outro método é o PTAM (Parallel Tracking and Mapping) que usa o movimento exclusivamente paralelo da câmera para calcular a paralaxe e profundidade das imagens captadas.

2.2.8 Sensores Baseados em Luz Estruturada

Estes tipos de sensores utilizam um padrão de luz projetada em uma cena para obter uma nuvem de pontos ou superfície de um objeto. Como o padrão de luz é conhecido, podemos relacionar cada ponto da imagem capturada pela câmera do sensor ao padrão projetado. Estes pontos podem ser triangulados e a informação 3D da cena é obtida^[8]. Na Figura 6 tem-se um exemplo de um dos tipos de luz estruturada^[17].

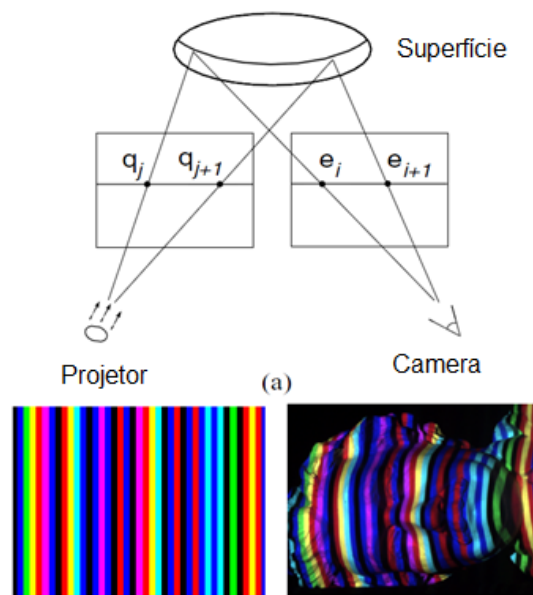


Figura 6: Reconstrução a partir de um padrão de faixas de cores ^[6].

O padrão utiliza a sequência de Bruijn para gerar uma sequência de faixas coloridas onde cada três cores consecutivas são únicas, assim podendo relacionar cada ponto da imagem projetada a uma faixa de pixels na imagem capturada à esquerda. A triangulação é mostrada na Figura 6 (A) e com isto consegue-se uma reconstrução da cena em 3D.

Diversos padrões de codificação foram criados. A codificação direta, ou em cores ou em escalas de cinza, para cada pixel tem-se um correspondente único em RGB ou cinza. A codificação é simples, mas está sujeita a ruídos já que para um grande número de pontos a diferença de cores é quase nenhuma^[8] e diversas imagens devem ser obtidas, pois a cor intrínseca de cada objeto na cena afeta as cores do padrão.

Os padrões podem mudar em relação ao tempo, assim obtendo uma série de imagens codificadas confiáveis e precisas. Um exemplo deste método é o código binário, na qual a projeção é codificada aos poucos com sucessivas projeções^[8].

Apesar de apresentar bons resultados, estes métodos não podem ser uti-

lizados em tempo real, pois precisam de diversas aquisições para poder codificar o padrão e para triangular os pontos.

No método da vizinhança (Spatial Neighborhood), a codificação do pixel a um certo ponto do padrão é obtido pela vizinhança de pontos em volta dele. Um exemplo é a sequencia de Bruijn, onde a vizinhança é gerada pseudo randomicamente, demonstrada na Figura 6, na qual a codificação de uma faixa é obtida pelas faixas adjacentes, já que não existe duas sequencias de três cores iguais no padrão.

Outro método é uma extensão da vizinhança pseudo randômica para o caso 2D, chamada de M-Array^[8] mostrado na Figura 7. Este método é utilizado pelo Kinect para a reconstrução 3D da cena capturada.

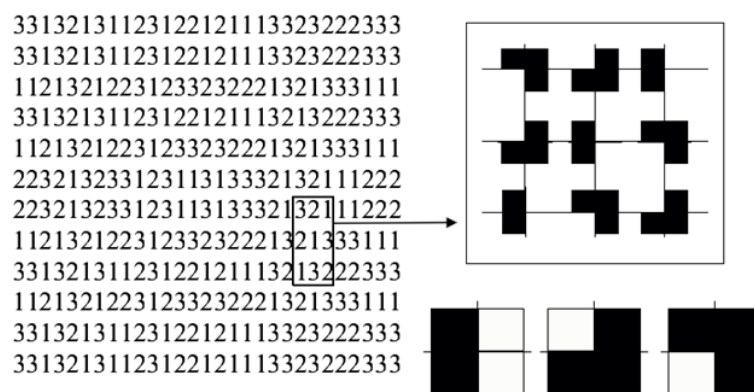


Figura 7: Codificação M-array baseada em três símbolos proposta por Griffin^[7], onde o ponto é decodificado utilizando sua vizinhança^[8]

2.3 Registro das Nuvens de Pontos

Na maior parte das técnicas de reconstrução com múltiplos registros não se tem o conhecimento da posição e rotação relativa da câmera em cada registro, portanto diversas técnicas de alinhamento foram criadas nestes últimos 20 anos.

Estas técnicas podem ser divididas em duas categorias: técnicas de regis-

tro grosseiro (coarse registration) e fino^[16](fine registration). O registro grosseiro é utilizado para uma estimativa inicial da diferença da posição e rotação entre duas nuvens de pontos. Entre as técnicas de registro grosseiro temos a Point Signature, Spin Image, e outras. As técnicas para registro fino incluem o ICP(Interactive Closest Point), Point-To-Plane, Signed Distance Fields e outras.

Serão apresentados apenas o registro fino.

2.3.1 Signed Distance Fields

Esta técnica utiliza a função de distância com sinal para criar um método robusto de registro. Diferente das demais técnicas, todas as nuvens de pontos são processadas ao mesmo tempo, eliminando a propagação de erros^[16].

Primeiramente alguma técnica de registro grosseira é utilizada para aproximar um primeiro alinhamento. Depois são gerados uma série de pontos-chave p numa grade tridimensional com tamanho δ e procura-se o ponto mais próximo de cada ponto-chave com a distância máxima permitida de $\sqrt{3}\delta$. Diferente do método dos mínimos quadrados, o calculo da distância do ponto mais próximo p_i^α é dada pela equação abaixo.

$$p_i^\alpha = (T_i^\alpha)^{-1}p - (n_p^{\alpha t}(T_i^\alpha)^{-1}p - k_p^\alpha)n_p^\alpha \quad (2.1)$$

n_p^α é a normal de p e k_p^α representa a distância com sinal dada pela equação abaixo.

$$k_p^\alpha = n_p^{\alpha t}c_p^\alpha - d_p \quad (2.2)$$

c_p^α representa o ponto mais próximo de cada ponto-chave para todas as

nuvens de pontos e d_p é a distância média entre o ponto chave p e cada c_p^α .

$$d_p = \frac{1}{\sum_{\alpha=1}^{N_R} w_p^\alpha} \sum_{\alpha=1}^{N_R} w_p^\alpha \quad (2.3)$$

d_p é a distância entre p e o ponto-chave mais próximo e w_p^α é o peso do ponto p .

Com os pontos mais próximos calculados, a transformação T^α é calculada.

Apesar de eliminar os erros de propagação comparada as duas outras técnicas mostradas a seguir, ela não pode ser utilizada em tempo real como em sistemas de localização e mapeamento, pois todas as nuvens de pontos tem de estar disponíveis para começar o algoritmo^[16].

2.3.2 Interactive Closest Point

O algoritmo ICP consiste em diminuir a diferença entre o quadrado da distância entre duas nuvens de pontos. Para duas nuvens de pontos representados por C_1 e C_2 , para cada ponto y da nuvem C_2 acha-se o ponto x mais próximo da nuvem C_1 . Assim se tem dois conjuntos de pontos no qual y_i é o ponto mais próximo a x_i .

Como para cada ponto y_i verifica-se todos os pontos da nuvem C_1 , este passo tem o custo de $O(NM)$ sendo N e M o numero de pontos em C_1 e C_2 , respectivamente.

Este passo é o mais custoso do processo, mas pode ser otimizado com algoritmos de procura como K-D tree^[17], onde o processo tem o custo de criação de $O(N \log_2 N)$ e, para árvores de 3 dimensões (3-D tree) tem-se o custo de procura de $O(N^{\frac{2}{3}})$. Outros métodos de procura mais eficientes existem como o direct Access method, apesar deste consumir muito mais memória^[17].

Para achar o movimento ótimo a fim de minimizar a distância entre os dois conjuntos de pontos utiliza-se o método dos mínimos quadrados 2.3.4

$$f(R, t) = \frac{1}{N} \sum_{i=1}^N \|Rx_i + t - y_i\|^2 \quad (2.3.4)$$

R e t são a matriz de rotação e translação, respectivamente.

Diversos métodos podem ser utilizados para achar a rotação R e a translação t que gera o menor erro, como os algoritmos baseados em dual quaternions^[28], SVD (Singular Value Decomposition)^[29], em matrizes ortogonais^[30] e em quaternions^[31]. Um estudo feito por EGGERT^[32] mostra que nenhum algoritmo se mostra superior comparado aos outros. Tendo este resultado, o algoritmo baseado em quaternion sera por sua simplicidade^[20].

Seja os conjuntos de pontos X e Y, cada ponto y_i é o ponto mais próximo a x_i . O centro de massa μ_x e μ_y é dado por:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.3.5)$$

$$\mu_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.3.6)$$

A matriz covariante Σ_{YX} é representada por:

$$\Sigma_{YX} = \sum_{i=1}^N [(y_i - \mu_y)(x_i - \mu_x)^t] = \sum_{i=1}^N [t_i x_i^t] - \mu_y \mu_x^t \quad (2.3.7)$$

As componentes da matriz $A_{ij} = (\Sigma_{YX} - \Sigma_{YX}^t)_{ij}$ são usadas para construir o vetor Δ , formado pelas componentes $[A_{23}, A_{23}, A_{23}]^t$. Com isto pode-se cons-

truir a matriz 4x4 $Q(\Sigma_{YX})$:

$$Q\left(\sum_{YX}\right) = \begin{bmatrix} tr(\Sigma_{YX}) & \Delta^t \\ \Delta & \Sigma_{YX} + \Sigma_{YX}^t - tr(\Sigma_{YX})I_3 \end{bmatrix} \quad (2.3.8)$$

I_3 é uma matriz identidade 3X3 e $tr(\Sigma_{YX})$ é o traço da matriz covariante Σ_{YX} . O auto-vetor $q_r = [q_0, q_1, q_2, q_3]^T$ corresponde ao máximo auto-valor de $Q(\Sigma_{YX})$ afim de se obter o quaternion para a rotação ótima.

Obtido o quaternion ótimo q_r , a matriz de rotação é gerada:

$$Q\left(\sum_{YX}\right) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.3.9)$$

Por fim a translação ótima é dada por:

$$t = \mu_x - R(q_r)\mu_y \quad (2.3.10)$$

Como existem outliers no encontro do ponto mais próximo x_i , a transformação não é a desejada, mas com um numero suficiente de pontos, após diversas interações, ela convergirá para um mínimo local^[17]. Diversos critérios de parada podem ser aplicados, como a diferença entre duas translações e rotações consecutivas serem menores que 1%^[17] ou a diferença do quadrado do erro entre duas iterações $d_k - d_{k+1}$ ser menor que um valor especificado pelo usuário^[20].

$$t = d_k = \frac{1}{N} \sum_{i=1}^N \|x_{ik} - y_{ik}\|^2 \quad (2.3.11)$$

O algoritmo se resume a

1. Achar para cada ponto y_i o ponto mais próximo a x_i .
2. criar a matriz Σ_{YX}

3. gerar a matriz $Q(\Sigma_{YX})$ a partir de Σ_{YX} .
4. Achar o maior auto-valor da matriz $Q(\Sigma_{YX})$. O auto-vetor correspondente é o quaternion q_r que criará a rotação ótima deste conjunto de soluções.
5. Achar a matriz de rotação $R(q_r)$ e a translação t
6. Caso o critério de parada não for atingido, fazer mais uma iteração.

2.3.3 Point to Plane

Uma modificação do ICP foi proposta por Yang Chen e Gerard Medioni^[33] onde, diferente do ICP, utiliza-se a normal de cada ponto em uma nuvem de pontos para achar seu par mais próximo. Para isso tenta-se minimizar a soma do quadrado da distância de cada ponto y_i da nuvem C_1 com o plano tangente do ponto mais próximo x_i da nuvem C_2 , como mostra a Figura 8. Esta forma não existe uma solução em forma fechada, mas pode ser resolvida por mínimos quadrados não linear, como o método de Levenberg-marquardt^[9]. Para incrementos de rotação e translação muito pequenos pode-se linearizar o problema^[9].

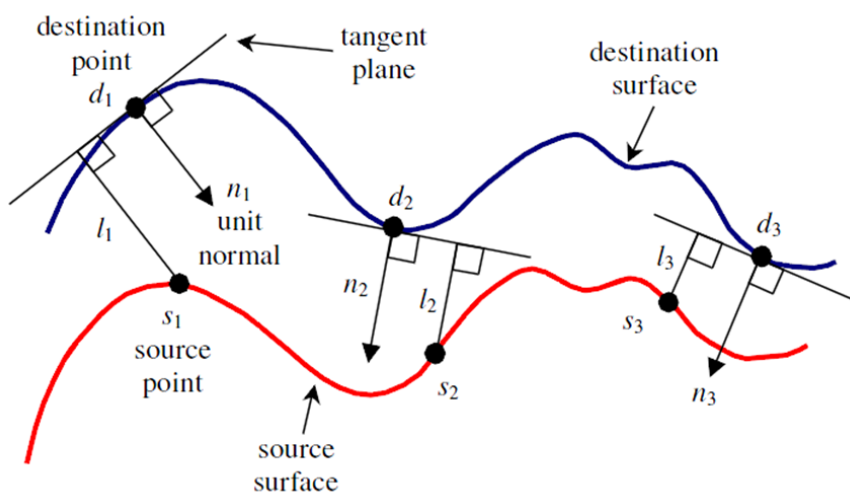


Figura 8: Distância do ponto ao plano entre duas superfícies^[9]

O objetivo de cada interação desta técnica é achar a matriz homogênea M_{opt} que reduz ao máximo o lado direito da seguinte equação:

$$M_{opt} = \sum_i [(Ms_i - d_i) * n_i]^2 \quad (2.3.12)$$

M é uma matriz de transformação homogênea, s_i é um ponto na nuvem C_1 , d_i o ponto mais próximo a s_i e n_i é um vetor unitário representando a normal da superfície no ponto d_i .

A matriz de transformação 4x4 representa a translação t_x, t_y, t_z , e a rotação α, β e γ nos eixos x, y e z respectivamente.

A matriz $M(t_x, t_y, t_z, \alpha, \beta, \gamma)$ então tem a forma:

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3.13)$$

Onde:

$$r_{11} = \cos \gamma \cos \beta$$

$$r_{12} = -\sin \gamma \cos \alpha + \cos \gamma \sin \beta \sin \alpha$$

$$r_{13} = \sin \gamma \sin \alpha + \cos \gamma \sin \beta \sin \alpha$$

$$r_{21} = \sin \gamma \cos \beta$$

$$r_{22} = \cos \gamma \cos \alpha + \sin \gamma \sin \beta \sin \alpha$$

$$r_{23} = -\cos \gamma \sin \alpha + \sin \gamma \sin \beta \cos \alpha$$

$$r_{31} = -\sin \beta$$

$$r_{32} = \cos \beta \sin \alpha$$

$$r_{33} = \cos \beta \cos \alpha$$

Com isto tem-se um sistema não-linear. Para pequenos deslocamentos podemos linearizar a matriz M afim de linearizar o sistema^[9]. Linearizando tem-se

$$\bar{M} = \begin{bmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3.14)$$

Agora pode-se escrever $(\bar{M}s_i - d_i) * n_i$ para todos os N pares na forma $Ax - B$ aonde

$$x = (\alpha, \beta, \gamma, t_x, t_y, t_z)^t \quad (2.3.15)$$

O sistema agora linearizado pode ser resolvido pelo SVD^[29].

O método point-to-plane (linearizado ou não) é mais dispendioso do que o ICP comum, mas foi observada maior frequência de convergências corretas^[18].

2.4 Refino e Compensação de Erros

Erros residuais de imprecisões da etapa do registro, ruídos do sensor e erros de calibração podem ser eliminados por um segundo alinhamento. Diversas técnicas foram desenvolvidas, a seguir são apresentadas algumas delas.

2.4.1 Método de Turk e Levoy

Esta técnica desenvolvida por Turk e Levoy^[11] triangula cada nuvem de ponto separadamente. Cada vértice da malha final é movido na direção de

sua normal para uma posição calculada pela média de todas as superfícies das malhas parciais intersectadas pela linha que passa por ele com direção paralela a sua normal.

2.4.2 Distâncias com Sinal

Esta técnica utiliza a função de distâncias com sinal entre a interceptação de uma linha que vai da posição do sensor até um voxel com a superfície triangulada, representada por $d_i(v)$, associada com a malha m_i em função do voxel v . A função $D(v) = \sum_{i=1}^N d_i(v)$ representa a média das funções de distância de todas as malhas para o voxel v . Cria-se uma isosuperfície com os voxels onde $D(v) = 0$ ^[10]. Isto gera uma isosuperfície com a média de todas as malhas como mostra a figura 10.

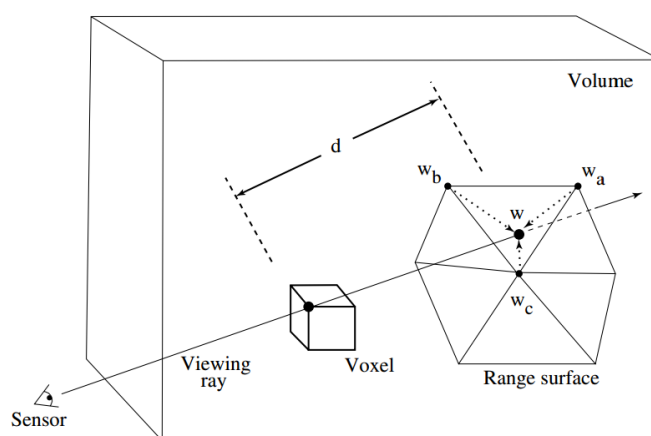


Figura 9: representação da distância $d_i(v)$ ^[10].

2.5 Reconstrução da Malha

As nuvens de pontos ou malhas de cada mapa de profundidade são processados para gerar uma malha única. Os métodos podem ser baseados em superfície, volume, superfícies deformáveis entre outros. Alguns destes métodos pressupõem conectividade dos pontos, outros pressupõem uma nuvem de

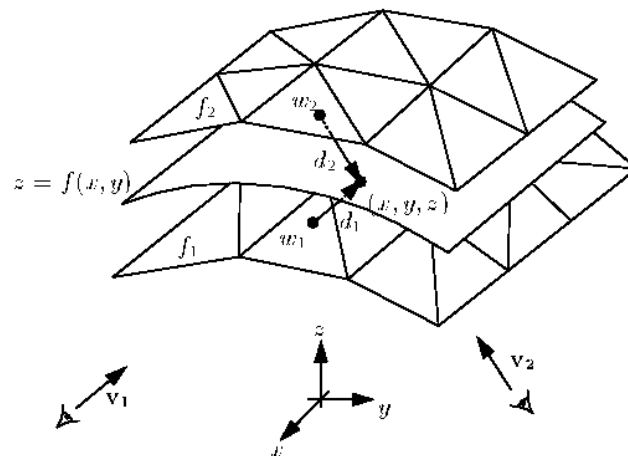


Figura 10: Isosuperfície resultante de duas malhas^[10]

pontos desestruturada.

2.5.1 Zippering

Este método criado por Turk^[11] pressupõe que cada nuvem de pontos pode ser triangulada em uma malha.

O método se resume a remover parcelas sobrepostas de duas malhas, juntar uma malha a outra e remover os triângulos pequenos produzidos na junção. Para eliminar os triângulos sobrepostos verifica-se se o ponto em uma malha está a uma distância maior do que uma distância pré-determinada. Os tres passos seguintes são demonstrados na Figura11.

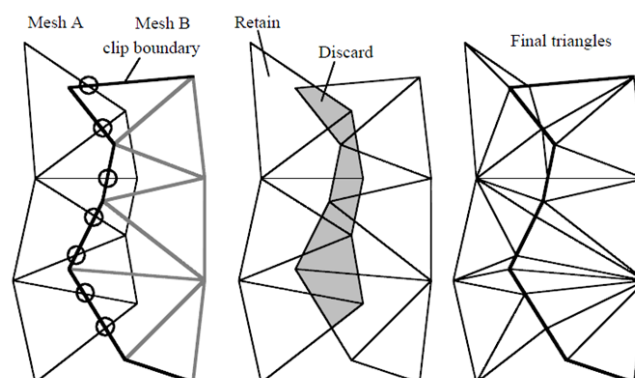


Figura 11: Metodo de zippering para juntar duas malhas^[11].

Os triângulos menores são eliminados de uma forma simples. Quando um triângulo menor que um tamanho pré-determinado é detectado, um dos vértices é removido junto com os triângulos a qual ele pertencia^[34]. Depois a técnica de compensação de erros de Turk e Levoy^[11] é aplicada.

2.5.2 Marching Cubes

O volume onde o objeto reconstruído se situa é subdividido em voxels, quando um destes voxels cruza com uma isosuperfície, sabemos que a superfície do objeto intersecta este cubo. Dependendo de quantos vértices deste cubo estão dentro ou fora da isosuperfície ($D(v) > 0$ e $D(v) < 0$ respectivamente) pode-se determinar qual dos tipos de cubos usa-se para melhor corresponder a superfície. Como o exemplo 2D mostrado na Figura 12, existem 16 tipos de quadrados $(2^4)^{[12]}$.

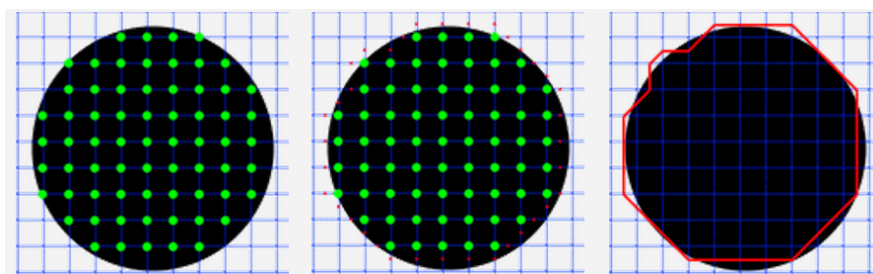


Figura 12: Analogia 2D da técnica de marching cubes^[12].

Já em um exemplo 3D, a combinação de 8 quinas gera 256 quadrados diferentes (2^8) . Essa quantidade pode ser simplificada com os seguintes critérios:

1. Rotação em qualquer um dos eixos principais
2. Espelhamento em qualquer um dos eixos principais
3. Invertendo o estado de todos as quinas e invertendo suas normais

Com isso as 256 possibilidades de cubos se reduzem a 15, mostrada na Figura 13. Para aumentar a precisão podemos discretizar o volume em mais cubos. Como visto na Figura 12, algumas vezes o vértice dos triângulos de um cubo acabam longe da superfície, gerando imprecisões. Um método para melhorar os resultados, chamado Adaptive Marching Cubes^[35] verifica se um vértice está fora da superfície, e move-o ao longo da aresta do cubo, adaptando-o a curvatura da superfície do objeto.

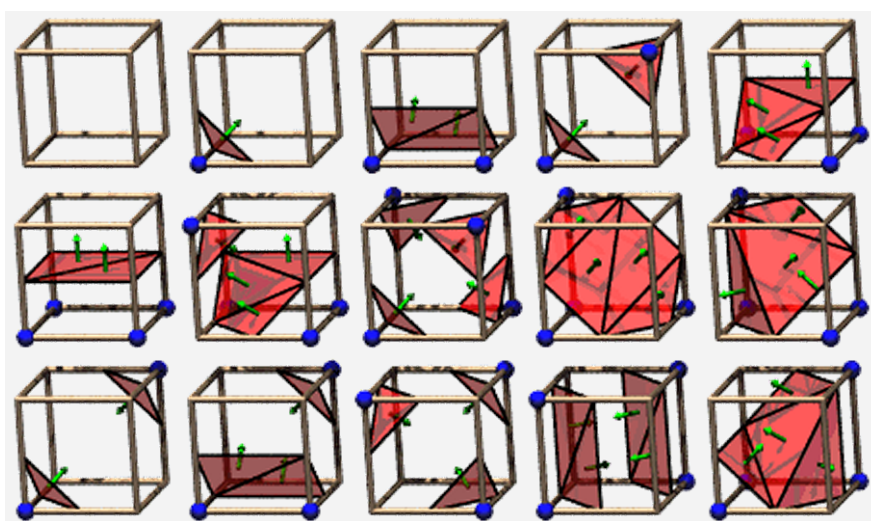


Figura 13: Os 15 tipos de cubos utilizados na técnica marching cubes^[12].

2.6 Pós-Processamento

Nesta etapa a malha final é modificada para poder ser usada em aplicações práticas, como simplificação de malha, geração de um mapa de texturas e sua correlação dos vértices, geração de NURBS (Non Uniform Rational Basis Spline) a partir da malha, exportação para determinado formato de arquivo como FBX, OBJ, PLY, entre outros.

3 METODOLOGIA

Nos próximos capítulos são apresentados o pipeline adotado e as propostas selecionadas para implementar cada método.

3.1 Pipeline escolhido

Os hardwares, métodos e técnicas escolhidos para o pipeline de reconstrução 3D foram:

1. Obtenção de mapas de profundidade: o hardware escolhido foi o Kinect, baseado em luz estruturada infravermelha.
2. Registro das nuvens de pontos: técnica ICP com kD-tree^[17] e sub-sampling.
3. Refino e compensação de erros: método da distância com sinal^[10].
4. Integração das nuvens em uma malha única: método de Marching Cubes^[12].
5. Pós-processamento e parametrização: exportação em formato .PLY (Polygon File Format) com informações da posição e cor de cada vértice.

A explicação das escolhas de cada módulo são tratadas a seguir, junto com suas implementações e variantes.

4 SENSOR PARA AQUISIÇÃO

O sensor escolhido para a aquisição dos dados de distância foi o Kinect. Sua principal vantagem do Kinect é o baixo custo, e sua popularidade: 18 milhões de cópias vendidas até janeiro de 2012^[36]. Em fevereiro de 2012, a Microsoft lançou o SDK (Software Development Kit) para o Kinect, possibilitando a utilização desta ferramenta para os mais diversos fins. Seu baixo custo, a qualidade do sensor de profundidade e a possibilidade de aquisição em tempo real, permitiram que o Kinect se popularizar entre pesquisadores e entusiastas^[23].

4.1 Especificações do Kinect

O Kinect foi lançado em 2010 como um controlador para o console Xbox com o propósito de rastrear corpos a fim do usuário interagir com o console sem a utilização de controles convencionais.

Entre os diversos componentes do Kinect, os três principais componentes são: uma câmera CMOS BGR com foco automático para captura de cor, um laser infravermelho e uma câmera CMOS para a captura da projeção infravermelha gerada pelo laser.

A obtenção da nuvem de pontos é realizada pelo sensor infravermelho do Kinect, onde um padrão pseudo-aleatório (M-array) é projetado em infraver-



Figura 14: Sensor Kinect^[13].

melho e sua reflexão é capturada por uma câmera CMOS. Sendo a projeção do padrão em infravermelho, a captura da textura em luz visível pode ser lida simultaneamente. Este padrão é composto de 3×3 repetições de 211×165 pontos de luz, totalizando em 633×495 pontos, muito parecido com a resolução VGA de 640×480 para a localização destes padrões. Um ponto com luminosidade mais intensa está localizado no centro de cada repetição como referência como mostra a Figura 15.

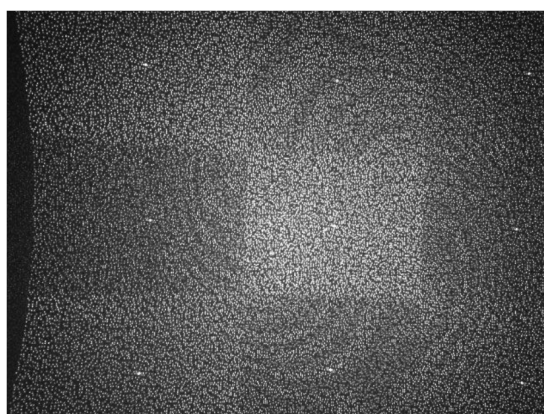


Figura 15: Imagem infravermelha da projeção. Pode-se ver as 9 repetições do padrão

Além do padrão, o projetor utiliza uma lente astigmática com diferentes distâncias focais nos eixos X e Y, perpendiculares ao feixe infravermelho, transformando os pontos do padrão em elipses, onde sua orientação e formato

depende da profundidade como mostrado na Figura 16.

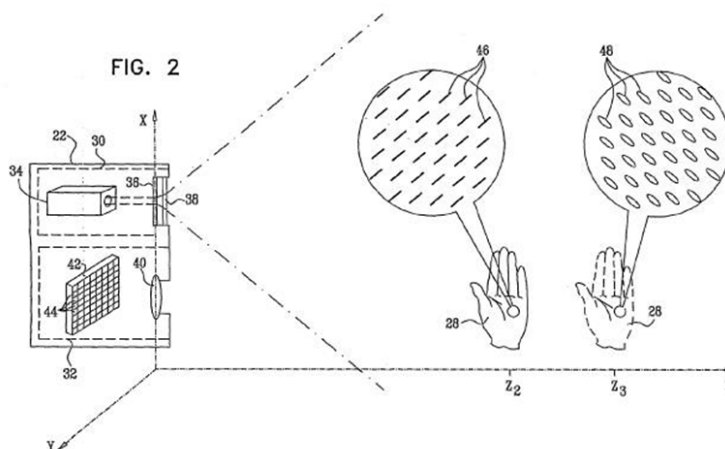


Figura 16: Raios infravermelhos com diferentes orientações e formatos em função da distância^[14].

Quando este padrão é projetado em uma cena, aliados com o astigmatismo da lente, ele se deforma e esta deformação é processada pelo Kinect que retorna um mapa de profundidade com uma resolução VGA (640x480) com valores de 11 bits. O sensor é capaz de obter imagens a 30 quadros por segundo, gerando assim nuvens de pontos com poucas mudanças entre duas nuvens consecutivas para velocidades pequenas, aumentando a chance de uma reconstrução correta utilizando o algoritmo ICP.

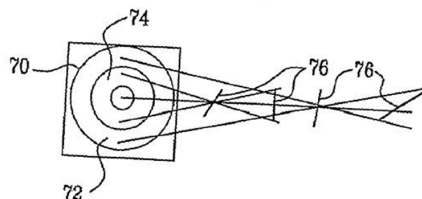


Figura 17: Astigmatismo da lente e distâncias focais diferentes^[14].

O Kinect tem 3 áreas com precisões diferentes: a primeira, indo de 0.8m até 1.2m, a segunda de 1.2m até 2m e a terceira indo de 2m até 3.5m com alta, média e baixa precisão, respectivamente.

4.2 Redução do Ruído do Mapa de Profundidade

O mapa de profundidade do Kinect é preciso mas existe ruídos do tipo salpicado, no qual poucos pontos da nuvem saltam de uma profundidade discretizada para a outra.

Para o ICP estes ruídos não são relevantes, mas quando é gerada a malha, triângulos destoantes são gerados, dificultando a geração da malha final e dificultando sua visualização.

Um método para a eliminação deste tipo de ruído é o filtro mediano, onde um kernel $n \times n$ com o pixel p ao centro com os valores representados pela profundidade de cada pixel como mostra a Figura 18.

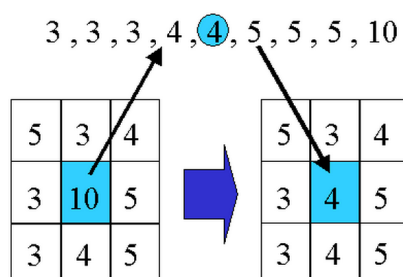


Figura 18: Filtro mediano^[15].

Este método pode ser utilizado para preencher buracos na nuvem de pontos, aonde o sensor não conseguiu captar a profundidade do pixel. Isso é feito quando o pixel no ponto central do kernel tem valor nulo mas um número mínimo de valores da matriz é não-nulo, utilizando das medidas dos valores não-nulos podemos estimar o valor do pixel com uma boa precisão para kernels pequenos. Este método, diferente do filtro Gaussiano preserva as bordas dos objetos.

4.3 Smoothing

O Kinect gera um mapa de profundidade discretizado em 2048 profundidades diferentes, com precisão decaindo de forma logarítmica. Isso gera curvas de nível e a normal dos triângulos na borda de cada curva de nível são quase perpendiculares aos triângulos adjacentes ficando muito aparentes quando shaders e iluminação são aplicados. Um filtro binomial foi aplicado para diminuir este problema.

Este filtro é um filtro Gaussiano com valores discretizados, e o kernel é gerado pela multiplicação de dois vetores com valores de uma respectiva linha do triângulo de Pascal. Um exemplo para um kernel 5×5 está abaixo, utilizando a linha 4 do triângulo de Pascal tem-se:

$$p = [1, 4, 6, 4, 1]^t$$

$$p * p^t = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

A soma dos valores da matriz é $(2^n)^2 = 256$ onde n é o número da linha do triângulo de Pascal. O valor do pixel central é a média de todos os valores multiplicados pelos respectivos pesos do kernel, assim quando mais distante é o pixel do ponto central, menos influência ele tem no valor computado do ponto central. Depois divide-se o valor computado pela soma dos pesos da matriz.

Apesar deste filtro arredondar as bordas dos objetos, para objetos próximos no qual a densidade de pontos é alta e a discretização da profundidade

é de apenas alguns milímetros, algumas passagens com um kernel pequeno não afetam as características principais do modelo reconstruído.

5 REGISTRO DA NUVEM DE PONTOS

O ICP foi escolhido para o registro das nuvens de pontos por sua simplicidade e velocidade. Como mostra a Figura 19, onde o ICP de Besl^[20], Zinsser^[37] e Jost^[38] são muito mais rápidos do que o point to plane de Chen^[33] e algoritmo genético de Chow^[39].

Points	Method	Translation error (mm)	$\alpha-\beta$ (rad)	RMS error (mm)	Time (s)
500	Besl	3.34	-0.09	4.18	1.41
	Zinsser	0.68	-0.06	2.35	1.95
	Jost	2.01	-0.07	4.54	0.36
	Trucco	2.87	-0.05	3.93	19.33
	Chen	1.3373	-0.008	2.1608	18.391
	Chow	0.27	0.0	4.60	154.05
1000	Besl	3.47	-0.09	3.78	5.22
	Zinsser	0.47	-0.02	1.42	7.97
	Jost	2.38	-0.08	3.70	1.38
	Trucco	2.71	-0.05	5.13	33.95
	Chen	0.29957	0.003	1.7305	67.954
	Chow	0.12	-0.01	4.51	281.61
5000	Besl	3.12	-0.08	3.20	57.36
	Zinsser	0.26	-0.02	0.85	281.58
	Jost	2.24	-0.09	2.83	9.14
	Trucco	2.57	-0.05	1.77	149.59
	Chen	1.2535	-0.015	1.2543	993.09
	Chow	0.06	0.0	3.57	1776.00

Figura 19: Comparação de diferentes métodos de registro^[16].

A desvantagem do ICP é a imprecisão maior comparada a outras técnicas, mas erros pequenos não são críticos para um sensor com imprecisões como o Kinect. Pequenos erros no alinhamento podem ser reparados na etapa da reconstrução da malha.

O algoritmo de Jost^[40] usa uma nuvem amostrada em suas etapas iniciais, aumentando o número de pontos ao longo das iterações, pois as primeiras aproximações do ICP são grosseiras e não necessitam de todos os pontos.

5.1 Métodos para Aumentar a Robustez do ICP

Um problema do ICP e Point-to-Plane é que eles convergem para um mínimo local, provado por Besl^[20]. Portanto, não se tem garantia que as duas nuvens irão convergir corretamente como mostra a Figura 20. Com a frequência de aquisição de imagens do sensor de profundidade rápida, a diferença de posição e rotação de uma nuvem de pontos para a nuvem seguinte é pequena, diminuindo quase por completo o problema de mínimos locais.

Nos casos em que os movimentos são rápidos e o problema de mínimos locais aparecer, uma possibilidade é iniciar iterações com perturbações iniciais em uma nuvem e selecionando o melhor resultado^[41]. Ainda não há garantia de que o a iteração convergirá para o mínimo global, mas tem a possibilidade de evitar alguns mínimos locais, melhorando a robustez do algoritmo.

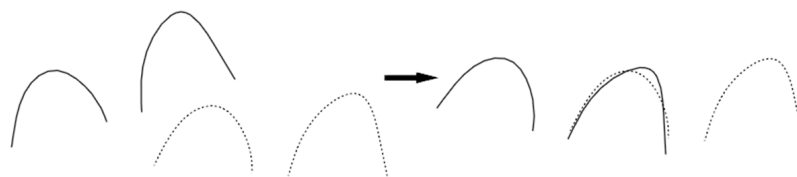


Figura 20: Duas nuvens de pontos (uma delas em mostrada em uma linha contínua por propositos de diferenciacao) convergem para um mínimo local^[17]

O ICP apresenta outros problemas, como nuvens de pontos com ruído ou não alinhadas^[18], aonde muitos outliers são gerados comprometendo o alinhamento das nuvens de pontos e gerando propagação de erro em múltiplas nuvens de pontos sequenciais, como mostra a figura 21.

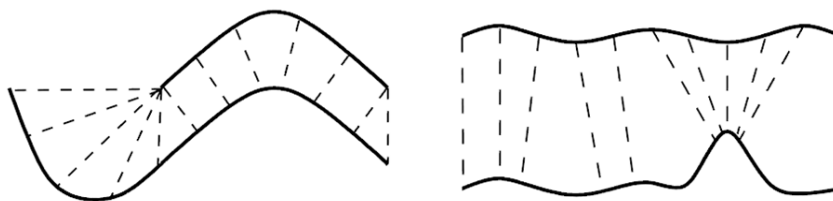


Figura 21: Problemas de nuvens com ruído e nuvens parcialmente sobrepostas no ICP^[18]

A fim de deixar o ICP mais robusto, diversos métodos foram criados. A seguir serão explicados alguns. Um dos problemas do ICP é que ele assume que todos os pontos podem ser pareados. Na realidade isso não pode ser suposto, pois interferência no sensor, oclusão, sobreposição parcial, entre outros fatores fazem que as duas nuvens de pontos tenham números diferentes de pontos ou pontos que não tem correspondência.

Diversos autores utilizaram diferentes critérios para a remoção de outliers como descartar pareamentos feitos entre pontos na borda de cada nuvem de pontos, descartar os n% piores pares em relação a alguma métrica (distância entre os pontos do par ou cor^[42] ou obter o histograma dos pontos e eliminar os pontos com base no desvio padrão^[43].

5.1.1 Método Estatístico para Eliminação de Outliers

Um método proposto por Zhengyou (1992) é de eliminar outliers por um método estatístico dinâmico, calculado a cada iteração. Um fator ζ é setado pelo usuário, onde este valor é a distância que ele considera que o registro das duas nuvens de pontos é satisfatório.

Seja D_{max}^i a distância máxima que um par de pontos pode ter antes de ser descartado na interação i e d_i , a distância entre x_i e y_i . Na interação i , consideramos todos os pontos com distância menor do que $D_{max}^{(i-1)}$ e a

media μ e o desvio padrão σ é obtido:

$$\mu = \frac{1}{n} \sum_{i=1}^n d_i \quad (5.1.1)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \mu)^2} \quad (5.1.2)$$

n é o numero de pares não descartados. Dependendo de μ a distância D_{max}^i é modificada:

$$\mu < \zeta \rightarrow D_{max}^i = \mu + 2\sigma$$

$$\zeta < \mu < 3\zeta \rightarrow D_{max}^i = \mu + 2\sigma$$

$$\zeta < \mu < 6\zeta \rightarrow D_{max}^i = \mu + \sigma$$

$$\mu > 6\zeta \rightarrow D_{max}^i = \xi$$

ξ é a mediana de todas as distâncias, assim se o registro das duas nuvens de pontos for ruim, apenas metade dos pontos serão considerados. Na Figura 22 um exemplo de remoção de outliers utilizando este método.

5.1.2 Movimento de Corpo Rígido para Eliminação de Outliers

Outro método para eliminação de outliers é proposta por^[44] onde, assumindo que a cena é estática, as duas nuvens de pontos se movem rigidamente e verificando a consistência de um par com n pares vizinhos pode-se verificar se este par está se movendo como um corpo rígido em relação á nuvem de pontos. Se há inconsistências, o par é descartado.

Sendo p_i e q_i pontos na nuvem de pontos P e Q na interação i tem-se a

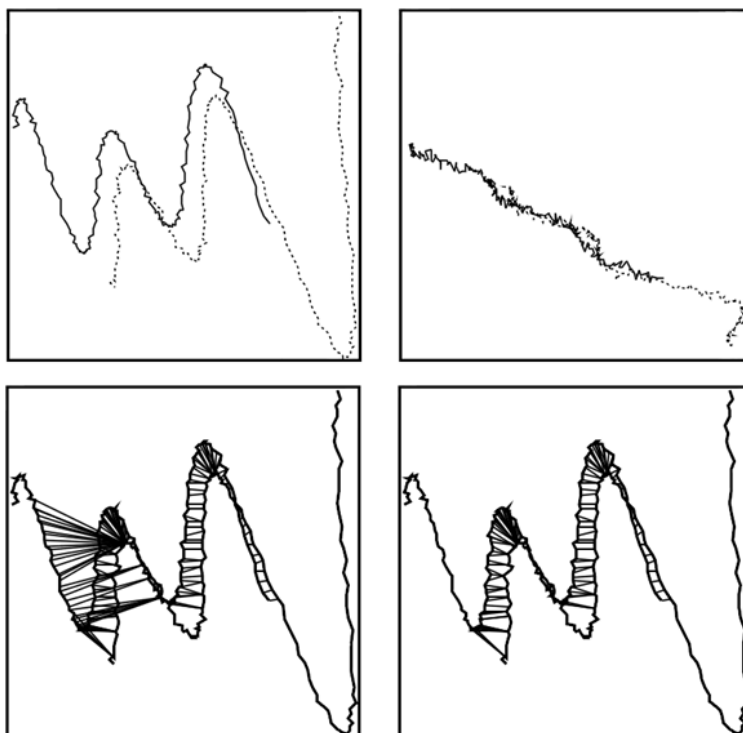


Figura 22: Esquerda Acima: Vista de cima de duas nuvens de pontos, uma tracejada e outra continua (pontos ligados para fins de diferenciação) Direita acima: Vista lateral da mesma imagem Esquerda abaixo: Pareamento dos pontos mais próximos, vemos claramente diversos outliers Direita abaixo: Após a remoção de pontos pelo método estatístico^[17].

restrição de corpo rígido

$$\|p_i - p_{i-1}\| = \|q_i - q_{i-1}\| \quad (5.1.3)$$

É utilizado um limite máximo t para que este par de pontos seja válido:

$$-t \leq \frac{\|p_i - p_{i-1}\| - \|q_i - q_{i-1}\|}{\max(\|p_i - p_{i-1}\|, \|q_i - q_{i-1}\|)} \leq t \quad (5.1.4)$$

Se o valor absoluto for maior que o limite t , o ponto é então descartado.

5.1.3 Truncamento

O Trimmed ICP^[19] propõe um método para se alinhar 2 nuvens de pontos onde há pouca sobreposição, tentando resolver o problema de malhas parcialmente sobrepostas. Sendo ξ um parâmetro conhecido que mostra o mínimo de sobreposição que pode ser garantida, assim o número de pontos que pode ser pareado é $N_{po} = \xi N_p$.

O procedimento é o mesmo, mas ao invés de se utilizar todos os pontos pareados Y e X , os pares são organizados em ordem crescente de acordo com a distância e seleciona-se as primeiras N_{po} correspondências e utilizando apenas estas para o cálculo dos mínimos quadrados. Na Figura 23 os resultados de Chetverikov^[19] são apresentados:

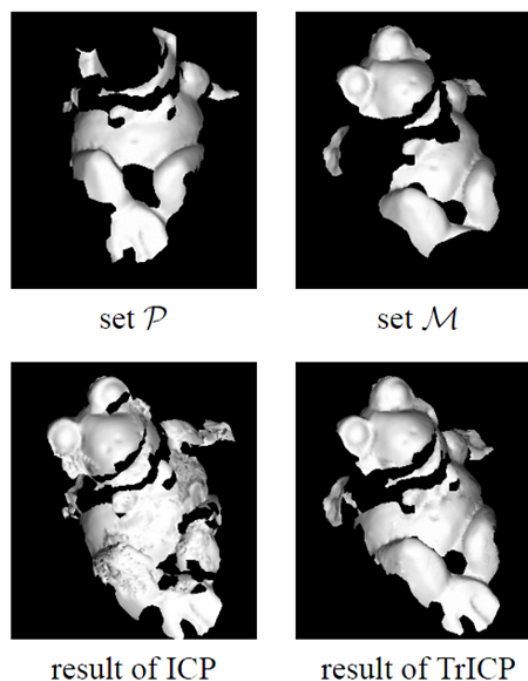


Figura 23: Comparação entre a composição de duas malhas que se sobrepõe parcialmente, a direita com TrICP e a esquerda com ICP com pareamento de todos os pontos^[19].

Um dos principais problemas é que o parâmetro ξ tem de ser conhecido, que pode ser impossível de ser obtido quando está se registrando nuvens de

pontos.

Estudos mostrados por Rusinkeewics^[18] mostram que para nuvens totalmente sobrepostas e sem ruído a escolha do método de remoção de outliers não é relevante para o tempo de convergência, assim qualquer uma delas pode ser escolhida com base puramente em estabilidade e robustez.

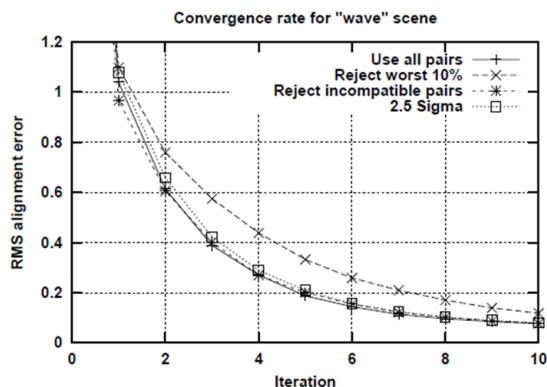


Figura 24: Tempo de convergência para diferentes métodos de eliminação de outliers^[18].

5.1.4 Aceleração da Convergência por Extrapolação

Esta técnica consiste em extrapolar o movimento da iteração do ICP, afim de deixa-lo mais robusto e acelerar sua convergência. Este método é explicado a seguir^[20].

A cada iteração do ICP, uma sequência de vetores é registrada, mostrando o caminho da nuvem de pontos até o seu mínimo local com a nuvem de referência. A diferença de vetores é definida por:

$$\Delta \vec{q}_k = \vec{q}_k - \vec{q}_{k-1} \quad (5.1.5)$$

O vetor $\Delta \vec{q}_k$ define a direção em que a nuvem se moveu entre a iteração $k - 1$ e k . Seja o ângulo θ_k o ângulo entre as duas últimas direções da nuvem

de pontos definido por:

$$\theta_k = \cos^{-1}\left(\frac{\Delta\vec{q}_k^t - \Delta\vec{q}_{k-1}}{\|\Delta\vec{q}_k^t\| \|\Delta\vec{q}_{k-1}\|}\right) \quad (5.1.6)$$

Sendo $\delta\theta$ a tolerância angular máxima, tem-se $\theta_k < \delta\theta$ e $\theta_{k-1} < \delta\theta$, o alinhamento do movimento das últimas 3 interações é boa. Seja v_k , v_{k-1} e v_{k-2} o arco aproximado

$$v_k = 0$$

$$v_{k-1} = -\|\Delta\vec{q}_k^t\|$$

$$v_{k-2} = -\|\Delta\vec{q}_{k-1}\| + v_{k-1}$$

A seguir computa-se os parâmetros de uma interpolação linear e parabólica para os três últimos movimentos

$$d_1(v) = c_1v + b_1$$

$$d_2(v) = a_2v^2 + b_2v + c_2$$

Que permite uma interpolação linear, e uma parabólica:

$$v_1 = \frac{-b_1}{a_1}$$

$$v_2 = -\frac{-b_2}{2a_2}$$

Para evitar extrapolações extremas, adota-se um valor máximo v_{max} . Besl^[20] adota valor de $v_{max} = 25\|\Delta\vec{q}_k^t\|$ em seus estudos.

Se $0 < v_2 < v_1 < v_{max}$ ou $0 < v_2 < v_{max} < v_1$ o valor v_2 é utilizado. Caso $0 < v_1 < v_2 < v_{max}$ ou $0 < v_1 < v_{max} < v_2$, v_1 é utilizado e caso v_2 e v_1 forem maior

que v_{max} , v_{max} é utilizado.

Com isto o deslocamento da nuvem de pontos fica:

$$\vec{q}'_k = \vec{q}_k + \frac{v_{escolhido} \Delta \vec{q}_k}{\|\Delta \vec{q}_k\|} \quad (5.1.7)$$

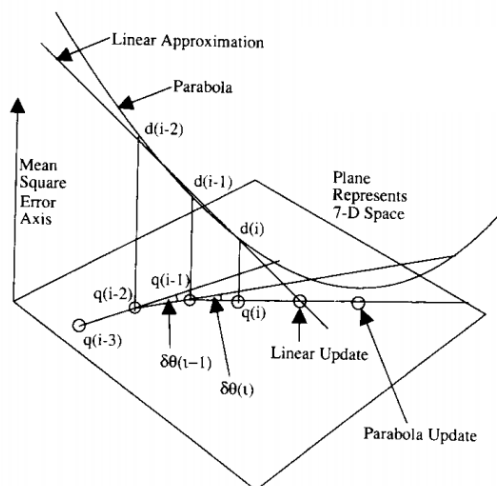


Figura 25: Uma direção constante nos permite a extrapolação do movimento e aceleração do método ICP^[20].

5.1.5 Utilização dos Dados de Cores (Color ICP)

Outra opção para aumentar a robustez e eliminar possíveis outliers é utilizar a informação de cor de cada ponto da nuvem de pontos na equação dos mínimos quadrados^[21]. Esta aplicação é especialmente interessante para aplicações com o Kinect, pois temos esta informação de cor para a nuvem de pontos.

A equação da distância entre dois pontos $p_1 = (x_{11}, x_{12}, x_{13})$ e $p_2 = (x_{21}, x_{21}, x_{21})$ com as cores $c_1 = (c_{11}, c_{12}, c_{13})$ e $c_2 = (c_{21}, c_{21}, c_{21})$ respectivamente

é descrita abaixo

$$d^2(p_1, p_2) = [(x_{11}-x_{21})^2 + (x_{12}-x_{22})^2 + (x_{13}-x_{23})^2 + \alpha_1(c_{11}-c_{21})^2 + \alpha_2(c_{12}-c_{22})^2 + \alpha_3(c_{13}-c_{23})^2] \quad (5.1.8)$$

Os multiplicadores α_1 , α_2 e α_3 são fatores que pesam a importância de cada componente de cor. O ICP é então iterado normalmente.

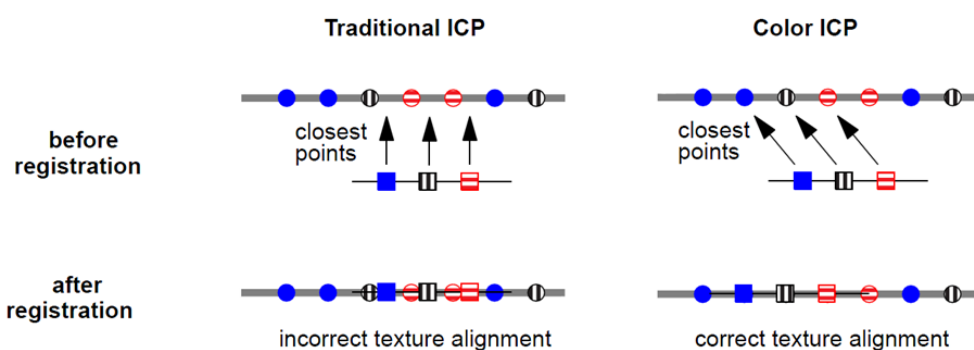


Figura 26: Uma direção constante Diferença entre o ICP e color ICP^[21]

Jhonson^[21] mostra que utilizando o espaço de cor YIQ e utilizando o peso de 0.1 para o canal Y, obteve excelentes resultados de convergência.

Como a câmera do Kinect utiliza o espaço de cor BGR, utiliza-se uma matriz de transformação para adquirir os pesos α_B , α_G e α_R . A matriz de transformação é descrita abaixo^[45]:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.9563 & 0.6210 \\ 1 & -0.2721 & -0.6474 \\ 1 & -1.1070 & 1.7046 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (5.1.9)$$

6 RECONSTRUÇÃO DE MALHA

O processo de Adaptive Marching Cubes foi escolhido, por sua natureza volumétrica de construção de malhas, gerando modelos com poucos buracos. Este método é aplicado facilmente quando se tem a superfície a ser reconstruída. Ela é obtida pelo isoplano calculado na etapa de refino e compensação de erros pela soma da função de distância.

6.1 Cálculo do Isoplano

Além da velocidade, vantagem deste processo é que a isosuperfície é calculada e sua resolução é uniforme, a mesma do tamanho do voxel escolhido resultando em uma malha com resolução uniforme. A escolha do tamanho do voxel é crucial, escolhendo um tamanho grande, detalhes do objeto se perdem, escolhendo um tamanho muito menor do que a resolução dos pontos gera uma malha pesada e com subdivisões de triângulos desnecessárias.

Um dos problemas deste método é que o processo com força bruta é dispendioso, pois para cada voxel próximo a superfície temos que fazer um ray-cast na malha. O processo pode ser muito dispendioso para milhões de voxels e malhas com muitos triângulos, mas uma solução criada foi utilizar o mapa de profundidade.

Primeiramente os voxels próximos às nuvens de pontos têm de ser mar-

cado afim de não utilizar uma procura em todo o volume, o que seria ineficiente. Para isso um cubo com volume de k^3 voxels é posto em cada vértice, e os voxels dentro deste cubo são marcados para a computação da função de distância com sinais.

Depois transformação global da nuvem de pontos a ser processada em relação a primeira nuvem tem de ser invertida a fim de se posicionar o sensor de volta ao zero global. Aplica-se a transformação inversa a posição do voxel, assim possibilitando a relação do voxel v com o pixel do mapa de profundidade.

Como a saída do sensor é um mapa de profundidade ela é transformada em uma nuvem de pontos, pode-se utilizar a posição do voxel para fazer a transformação inversa e achar a posição i e j no mapa de profundidade como mostra a Figura 27.

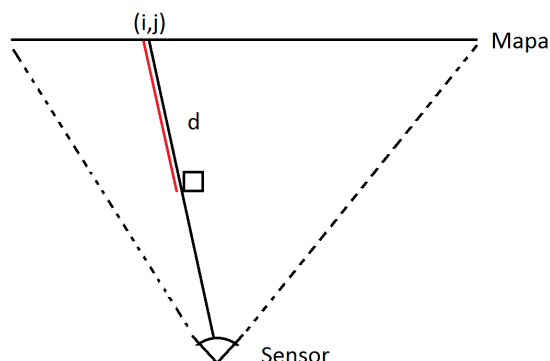


Figura 27: Com a transformação inversa, acha-se a coordenada i e j no mapa de profundidade, subtraindo-se a distância do mapa de profundidade com a distância do voxel obtemos a distância com sinal d representado pelo segmento vermelho.

Gerada a isosuperfície utiliza-se a técnica de Marching Cubes para gerar a malha final.

7 PÓS PROCESSAMENTO E FORMATO EXPORTADO

O formato PLY(Poligon File Format) foi criado principalmente para armazenar dados de scanner 3D por Greg Turkos anos 90. Este formato permite descrever propriedades dos pontos além de sua posição como cor, transparência, normal, coordenadas de textura e coeficientes de confiança das medições. Por estes motivos este formato foi escolhido para o armazenamento dos dados da malha final.

8 RESULTADOS

Nos próximos capítulos são apresentados os resultados do processo de reconstrução 3D. A apresentação divide-se em três partes, sensor, processo de registro e reconstrução da malha, com um capítulo dedicado a cada parte.

9 CAPTURA DA NUVEM DE PONTOS UTILIZANDO O KINECT

A seguir serão apresentados resultados de testes do sensor do Kinect com a finalidade de entender suas limitações. Duas características importantes serão estudadas, precisão do mapa de profundidade e densidade de pontos. Também foram estudados erros e suas causas.

9.1 Discretização da Profundidade

Apesar do mapa de profundidade do Kinect retornar uma imagem na resolução VGA com 11 bits de informação por pixel, ou seja, discretizando em 2048 níveis diferentes de profundidade, seu valor para a distância máxima não ultrapassa 1020 para 4 metros, ou seja, utiliza-se apenas 10 bits. Sua precisão se apresentou não linear, onde quanto mais distante a superfície está do sensor, maior a distância entre dois níveis de profundidade. Para a distância mínima de 0.8 metros o valor retornado é de 650, ou seja, entre 0.8 metros e 4 metros tem-se uma discretização de apenas 370 níveis, muito menor do que os 2048 considerados previamente.

Pelo gradiente da curva do gráfico 28, a precisão de profundidade a 1 metro é de aproximadamente 2.9mm por unidade enquanto a 3.5 metros é de 45 mm por unidade como pode-se ver na Figura 29. A 5m sua precisão decai para 70mm^[22].

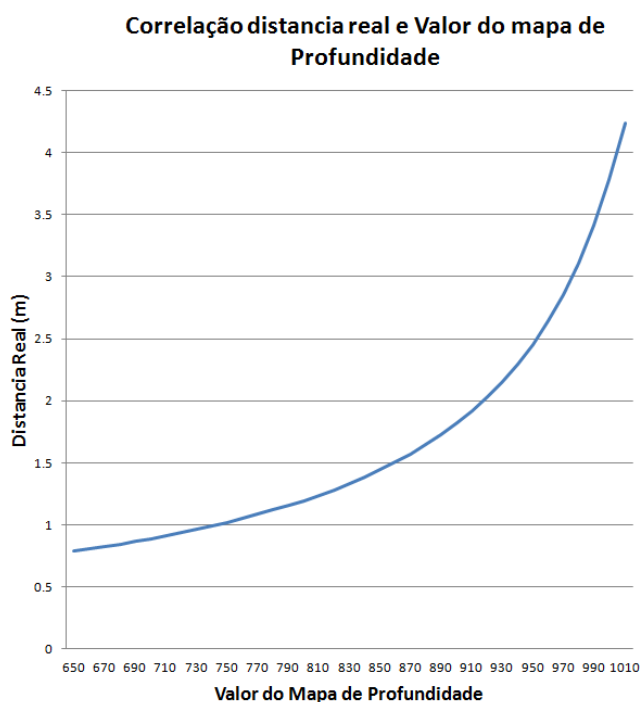


Figura 28: Codificação da distância em bits utilizada pelo Kinect.

A alta discretização da profundidade em grandes distâncias geram erros de que podem aumentar o erro quadrático do ICP. O Kinect for Windows, diferente do Kinect para X-Box possui um segundo modo mais preciso, onde seu alcance mínimo e Máximo é de 0,4m a 3 metros, respectivamente, aumentando em muita sua precisão.

9.2 Densidade de Pontos

A densidade de pontos decai quadraticamente com a distância. Verificou-se que a densidade de pontos é de aproximadamente de 0,4 pontos por mm^2 a 1 metro e decai para aproximadamente 0,025 pontos por mm^2 a 4 metros.

A diminuição da densidade de pontos também afeta o ICP, pois pontos próximos entre duas nuvens não são necessariamente os mesmos pontos no objeto gerando um aumento do erro. O método point-to-plane contorna este erro já que a distância é em relação a normal do plano, portanto nuvens com

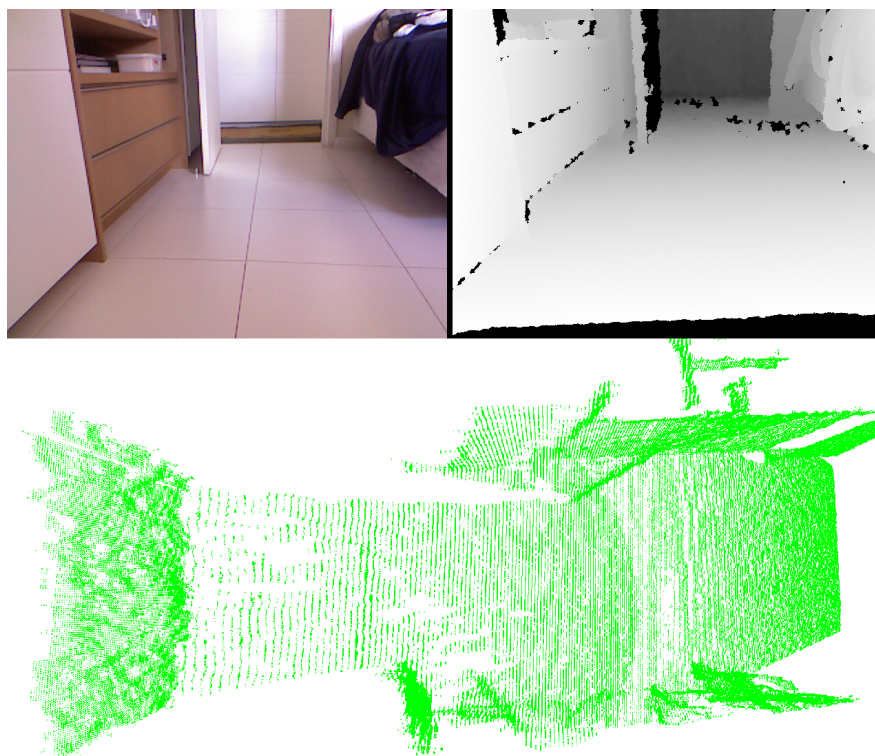


Figura 29: Observa-se que quanto maior a distância mais evidente fica a discretização da distância.

pontos esparsos não são afetadas, apesar deste método gerar outros erros como deslizamento de planos.

9.3 Erros

Ruídos causados pelo Kinect são originados principalmente de três fontes^[22]:

1. Sensor.
2. Etapa de medição.
3. Propriedades da superfície do objeto.

Erros do sensor se devem a calibração inadequada e medição imprecisa de disparidades. Pela Figura 30 verifica-se um aumento no ruído de medição.

Segundo Khoshelham^[22] este ruído randômico aumenta quadraticamente com a distância.

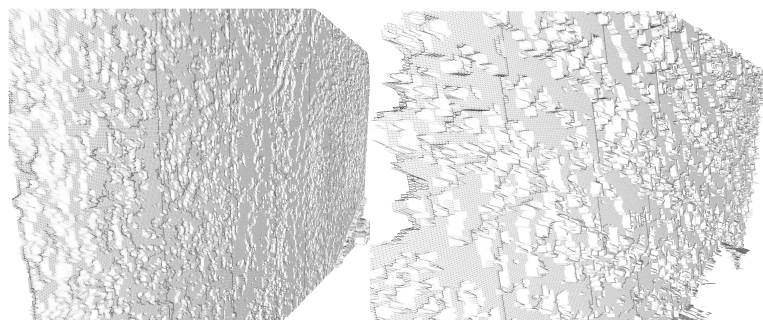


Figura 30: Erros na medição. Acima: malha de uma obtida a 0.8m do sensor. Abaixo: malha da mesma parede obtida a 3.9m do sensor.

Durante a aquisição dos resultados foi encontrado um erro onde os pontos perto das bordas têm forte distorção como mostra a figura 31. Esta distorção é causada pela distorção da lente, onde a formula para se obter o ponto 3D a partir do mapa de profundidade não se aplica. Este erro se classifica como de erro de calibração, e seu erro é sistemático. Mesmo utilizando a função de transformação do SDK esta distorção dos pontos se mantém. Estes pontos foram desconsiderados na etapa de registro afim de não gerar uma propagação de erros.

Erros causados na etapa de medição são relacionados a condição de luz e da geometria do objeto. A luz solar pode interferir ou até impedir a medição infravermelha. Os pontos de luz infravermelha do laser ficam pouco contrastantes, gerando ruídos ou falha completa na medição, criando buracos na nuvem de pontos^[22].

Propriedades da superfície do objeto causam um impacto na medição dos pontos. Superfícies reflexivas aparecem sobre-expostas na câmera infravermelha que impedem a medição e gerando buracos na nuvem de pontos (vide Figura 32).



Figura 31: Erros na medição da nuvem de pontos vista de frente e cima. Os quadrados vermelhos mostram pontos com alta imprecisão da nuvem de pontos em relação ao modelo real.

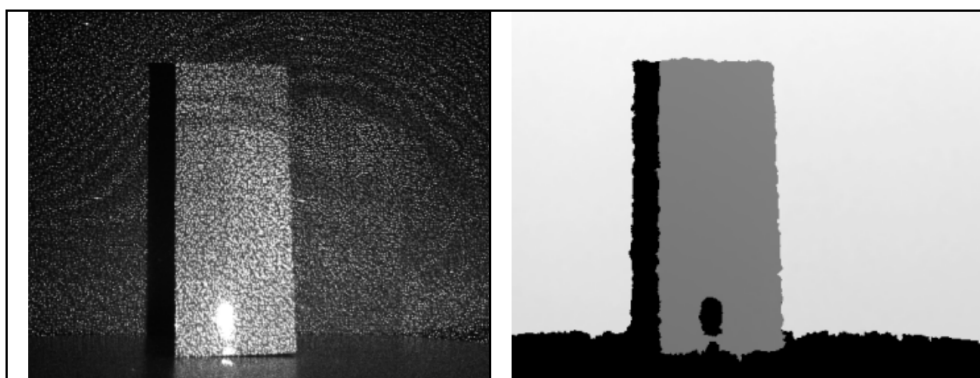


Figura 32: Problemas na medição de objetos especulares^[22].

9.4 Filtragem

O filtro mediano se apresentou eficiente no preenchimento de buracos e filtragem de ruídos a Figura 33 (c) mostra o preenchimento de diversos buracos na malha e para pontos longes do sensor o ruído é drasticamente diminuído como mostra a Figura 34 (C). Nela podemos ver uma clareza muito maior nas curvas de nível.

Testando a precisão do ICP para malhas com filtro mediano e binomial apresentaram melhoras, mas não significativas. As normais da superfície

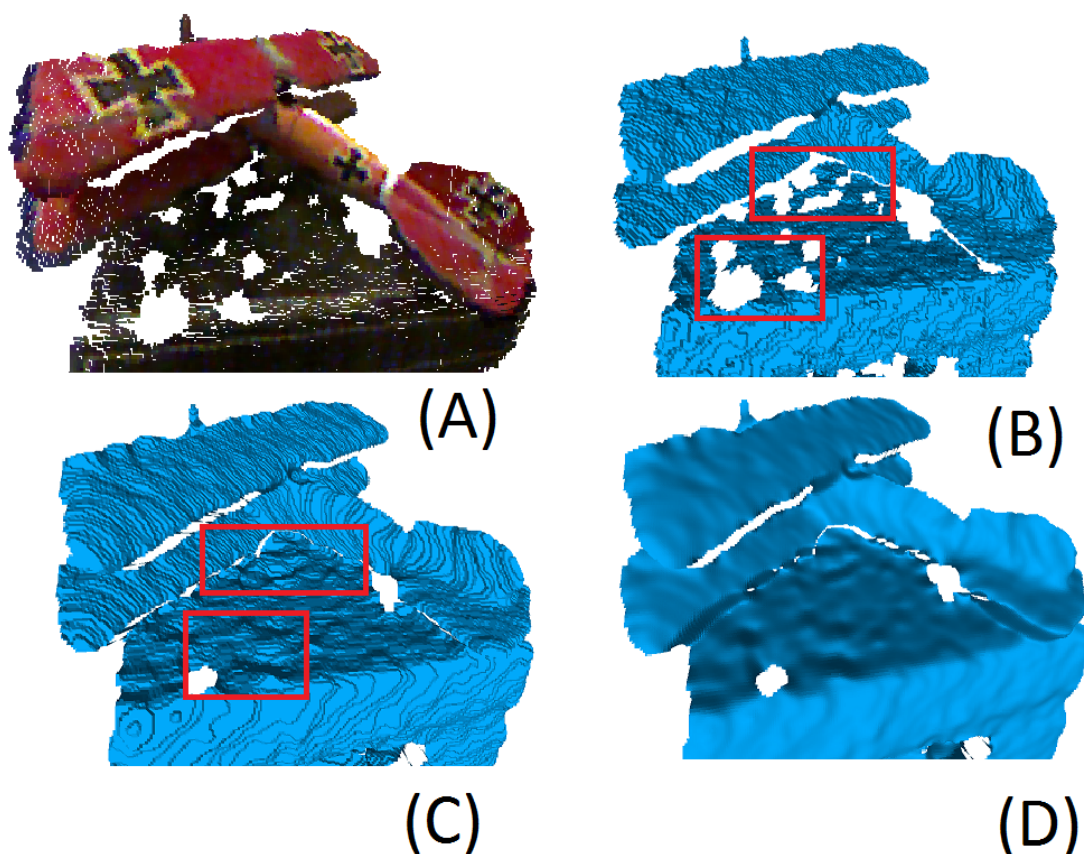


Figura 33: a) nuvem de pontos colorida a 0.9 metros do sensor b) Malha gerada c) malha com filtro mediano de 5x5 pixels d) malha com filtro mediano de 5x5 pixels e duas passagens com filtro binomial de 5x5 pixels

após a aplicação do filtro binomial ficaram muito mais próximas a do objeto real do que sem filtragem, principalmente para objetos próximos como mostra a Figura 33. Objetos longe do sensor, onde a curva de nível causada pela discretização é maior, apesar de amenizadas, apresenta ainda curvas de nível após a aplicação do filtro binomial.

9.5 Erros de Cores

Foi verificado imprecisões na correspondência entre a câmera RGB e o mapa de profundidade disponível no SDK. Isto gera problemas quando se une as nuvens de pontos, pois erros nas cores dos vértices são gerados. Foi veri-

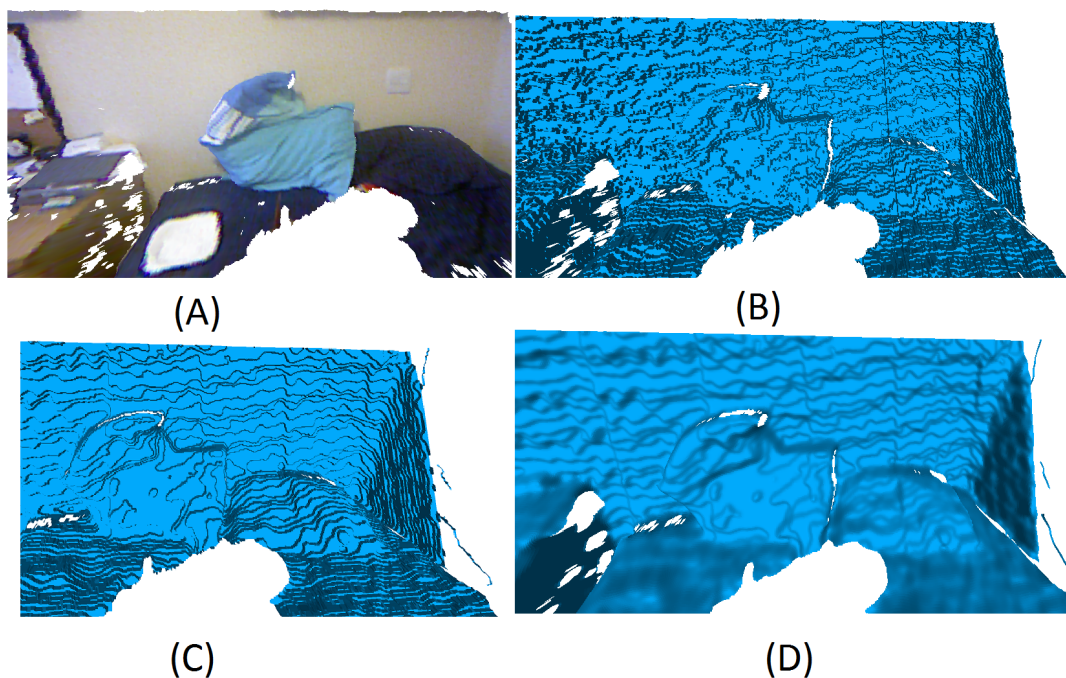


Figura 34: a) nuvem de pontos colorida a 2.5m do sensor b) Malha gerada c) malha com filtro mediano de 5x5 pixels d) malha com filtro mediano de 5x5 pixels e duas passagens com filtro binomial de 5x5 pixels

ficado que no limite do objeto de acordo com cada campo de visão do sensor temos o erro da correspondência alguns pixels, a figura 35 mostra que Alguns pixels do cabelo foram correspondidos com cores da parede ao fundo. À direita verifica-se que após juntar 86 nuvens de pontos com amostragem de 5000 pontos cada uma, obtendo-se erros nas cores de diversos pontos do modelo, mais aparentes no cabelo e no rosto.



Figura 35: À esquerda tem-se uma nuvem de pontos. À direita tem-se a combinação de 86 nuvens amostradas com 5000 pontos cada.

10 ETAPA DE REGISTRO UTILIZANDO ICP E VARIANTES

A seguir são apresentados diversos experimentos com o ICP, como tempo de iteração, convergência e erros. A Figura 36 mostra um teste com 50 iterações de duas nuvens de 307.200 pontos. As nuvens foram amostradas com 40.000 pontos para o cálculo do ICP e um limite de 50mm foi imposto após 40 iterações para o descarte de outliers. A Figura 37 mostra o resultado da nuvem resultante em 86 mapas de profundidade.

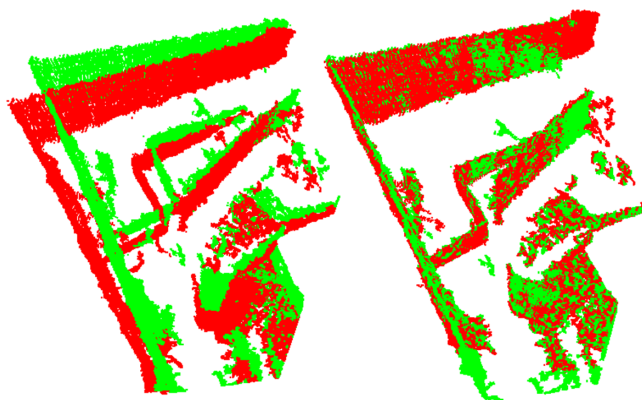


Figura 36: Resultado do ICP para duas nuvens de pontos, à esquerda malhas não alinhadas e à direita malhas após 50 iterações.

10.1 Tempo de Iteração

A interação do ICP pode ser dividida em três partes: pareamento dos pontos mais próximos, criação da matriz Q e computação da matriz homogênea. De acordo com os testes representados pela Figura 38 a parte mais dispen-

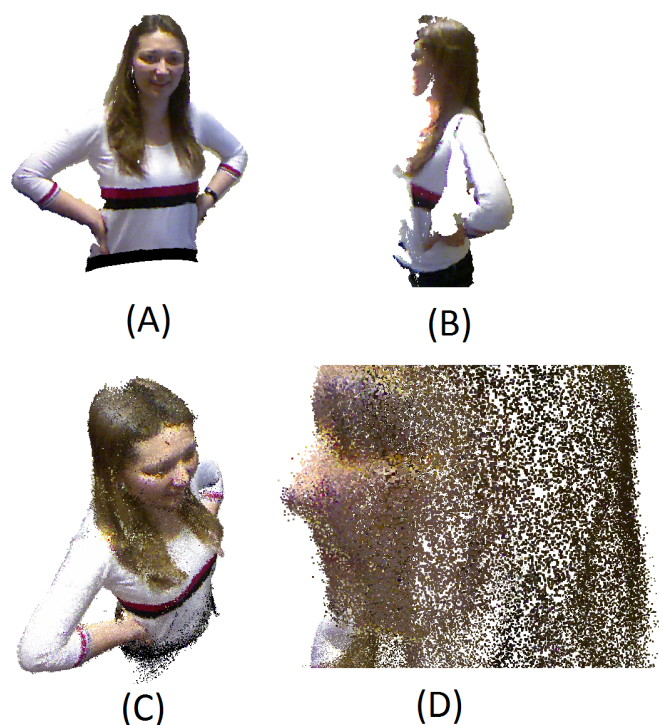


Figura 37: a) Nuvem de pontos com 87.000 pontos, b) nuvem de pontos com 74.000 pontos, c) combinação de 86 nuvens de pontos, cada nuvem amostrada em 5000 pontos. d) Detalhes mostrando a convergência precisa das 86 nuvens.

diosa deste processo é o pareamento dos pontos mais próximos, com uma ordem $O(n^2)$ para força bruta e $O(\log N)$ para a K-Dtree. Resultados da Figura 39 mostram que o processo de procura por KD-tree é muito mais dispendioso nas primeiras iterações na qual as duas superfícies estão separadas (pior caso $O(n^{\frac{2}{3}})$) e mais galhos da árvore binária devem ser acessados para a verificação do ponto mais próximo. Todos os testes sintéticos foram realizados por duas malhas completamente sobrepostas e com 100% de pareamento dos pontos mais próximos. O processador utilizado foi um Intel Core2 Quad 6600.

Pode-se concluir destes resultados que o tempo de iteração aumenta drasticamente com o tamanho da nuvem de pontos, por isso precisa-se verificar a relação do tamanho de uma nuvem e o erro quadrático resultante do processo ICP. Se utilizar uma nuvem com muitos pontos, o calculo do ICP se tornará preciso, mas lento, e se escolher uma nuvem menos densa, o processo se torna

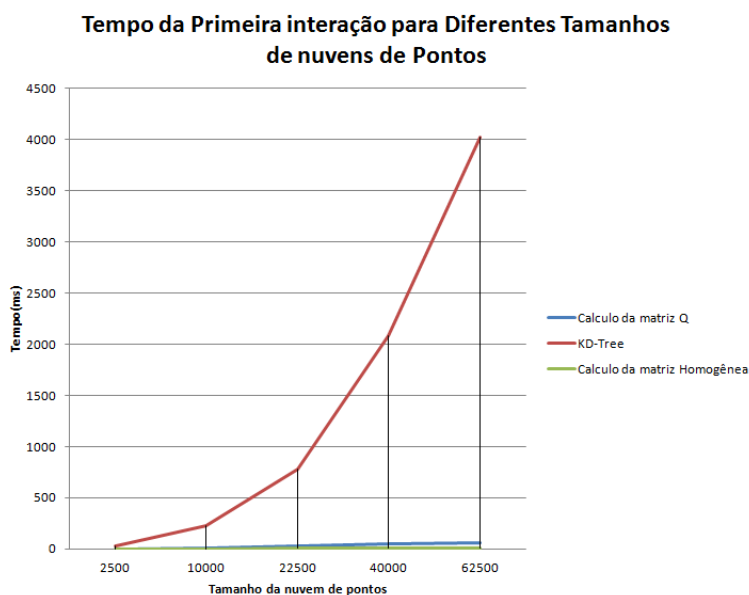


Figura 38: O tempo de procura do ponto mais próximo é o principal fator no tempo total da interação

rápido, mas impreciso.

Na Figura 40 o ICP foi computado com diferentes amostragens de uma nuvem de pontos e depois o erro foi calculado. A nuvem original possui 90 mil pontos e tem dimensões de 1m x 1m. O método de amostragem utilizada foi randômico e o erro é uma média de 10 registros de ICP. No teste verificou-se que o erro não diminui significativamente após uma amostragem de 5000 pontos ou uma densidade de 0.005 pontos por mm^2 .

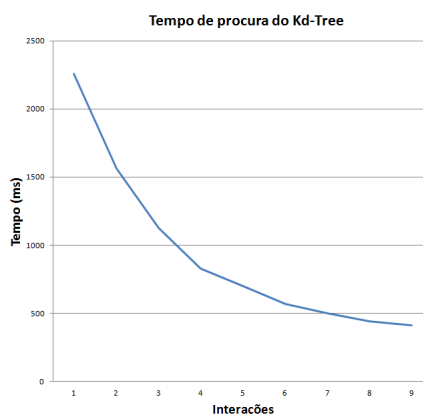


Figura 39: Nota-se a diminuição do tempo de procura quando as malhas ficam mais próximas a cada iteração.

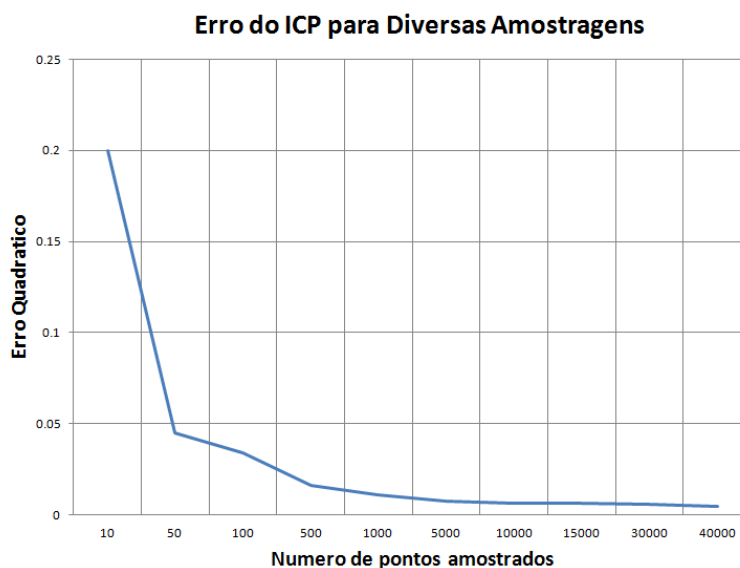


Figura 40: Erro do registro usando ICP para diversas amostragens.

10.2 Problemas de Convergência

Um problema encontrado foi o registro de nuvem de pontos com um grande plano, como uma parede, no qual o pareamento de pontos da parede inicial, é regido pela profundidade. Mesmo para pequenas rotações, o deslocamento em x e y pode ser relativamente grandes em relação á profundidade como mostra a figura 41 (A). A transformação da primeira iteração se resume principalmente em uma rotação no eixo vertical, resultando na Figura 41 (B).

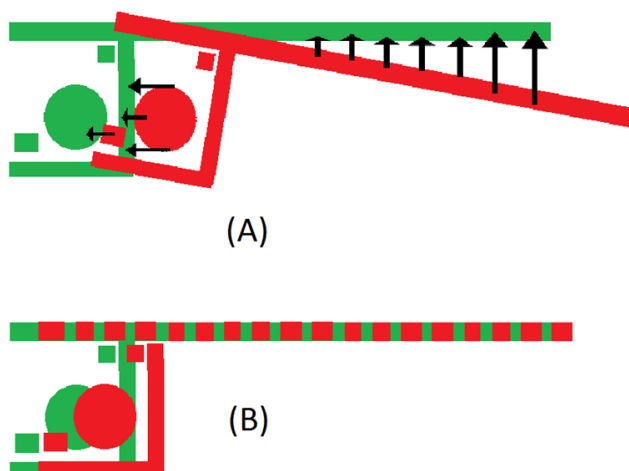


Figura 41: Erro do registro causado por um grande plano

Como os pontos da parede estão pareados e a distância dos pares é pequena, a convergência das outras formas como uma mesa ou um vaso se tornam extremamente lentas. Com isso, o algoritmo alcança o critério de parada e retorna uma nuvem de pontos deslizada em relação à primeira. Mesmo forçando mais de 100 iterações, a convergência se torna tão lenta que o truncamento dos números utilizados no cálculo causa uma transformação onde a rotação e a translação são zero.

Testes mostram que o ICP clássico funciona muito bem em nuvens de pontos com muitas formas diferentes, mas falha quando a maior parte da cena é um plano, como mostra a Figura 42. 20 nuvens de pontos foram iteradas pelo ICP, partindo da esquerda, onde se localiza a caixa de papelão, indo em direção aos computadores.

As primeiras 11 nuvens convergiram corretamente, criando uma reprodução da caixa de papelão nítida e correta, por causa das várias formas (quina da parede, caixa de papelão, etc...) registradas.

A partir da nuvem 11 começ-se a detectar uma grande quantidade de pontos na parede atrás dos computadores gerando o problema descrito acima. Pode-se ver claramente o deslizamento de uma nuvem em relação à outra.

10.3 Color ICP

O Color ICP apresentou uma convergência mais rápida e menores erros comparados ao ICP como mostra a Figura 43. Ela também é mais robusta para casos ideais. Mas depois de diversos testes notou-se uma instabilidade no algoritmo causada pela compensação de cor e exposição de luz da câmera CMOS. Imagens com diferenças de iluminação causavam o algoritmo falhar completamente, pois quando uma lâmpada ou fonte de luz forte se aproximava

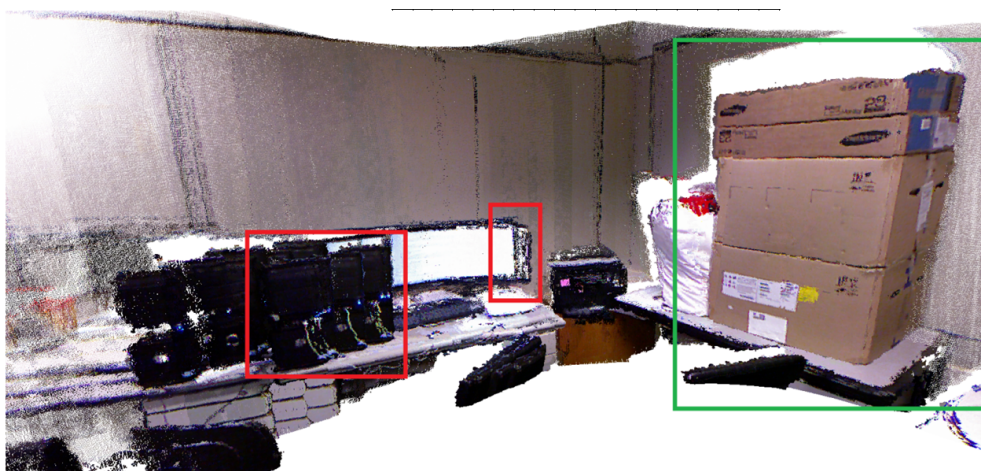


Figura 42: Testes mostrando convergências erradas causada por grandes planos

do centro da imagem, a câmera mudava a sua exposição causando uma diferença de intensidade e cores entre imagens capturadas, Como a distância da cor na computação do ICP é da ordem de um metro comparada a distância real o algoritmo falhava completamente.

Para casos ideais o algoritmo funciona satisfatoriamente, mas por sua instabilidade ele foi descartado.

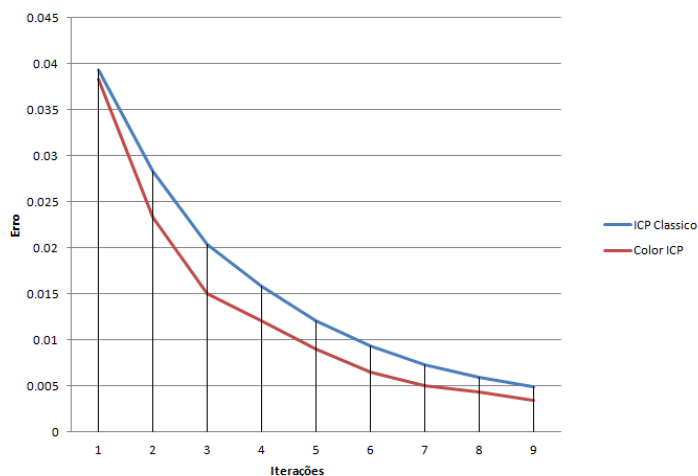


Figura 43: Convergência do ICP clássico e o Color ICP

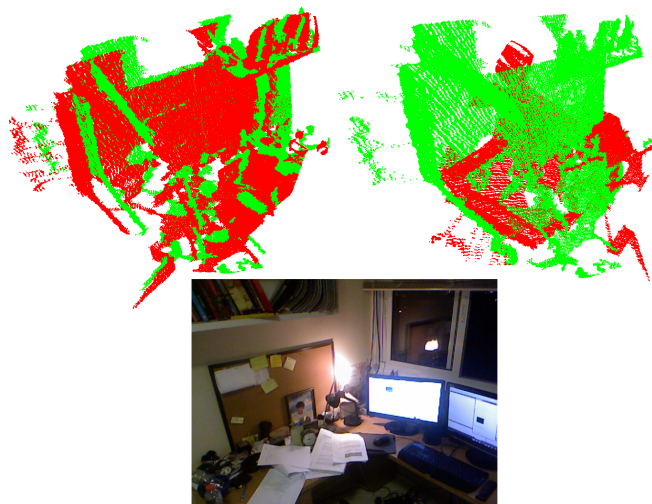


Figura 44: Falha completa do Color ICP, Vemos as malhas antes(esquerda acima) e depois (direita acima) da iteração. A imagem abaixo mostra a captura da imagem a cores correspondente a uma das malhas.

10.4 Threshold de Cor

Visto a falha do Color ICP para casos reais, um método verificação por cor foi criado, na qual a distância é calculada pelo método clássico, mas depois da verificação da distância, se a diferença de cores em um par de pontos estiver acima de um limite o ponto é descartado e procura-se o próximo ponto mais próximo. Este método mostrou-se robusto e seu erro quadrático final menor do que o ICP clássico. O problema de grandes planos pode ser resolvido caso o plano tenha múltiplas cores contrastantes como mostra a Figura 45.

O maior problema deste método é a verificação múltipla de pontos que deixa a iteração muito lenta, principalmente quando se utiliza KD-Tree com remoção de galhos. A procura por força bruta foi utilizada nos testes.

10.5 Métodos Estatístico e Trimmed

O método estatístico se mostrou mais robusto para a maioria dos casos, pois como ele se adapta de acordo com o histograma de distâncias, ele pode

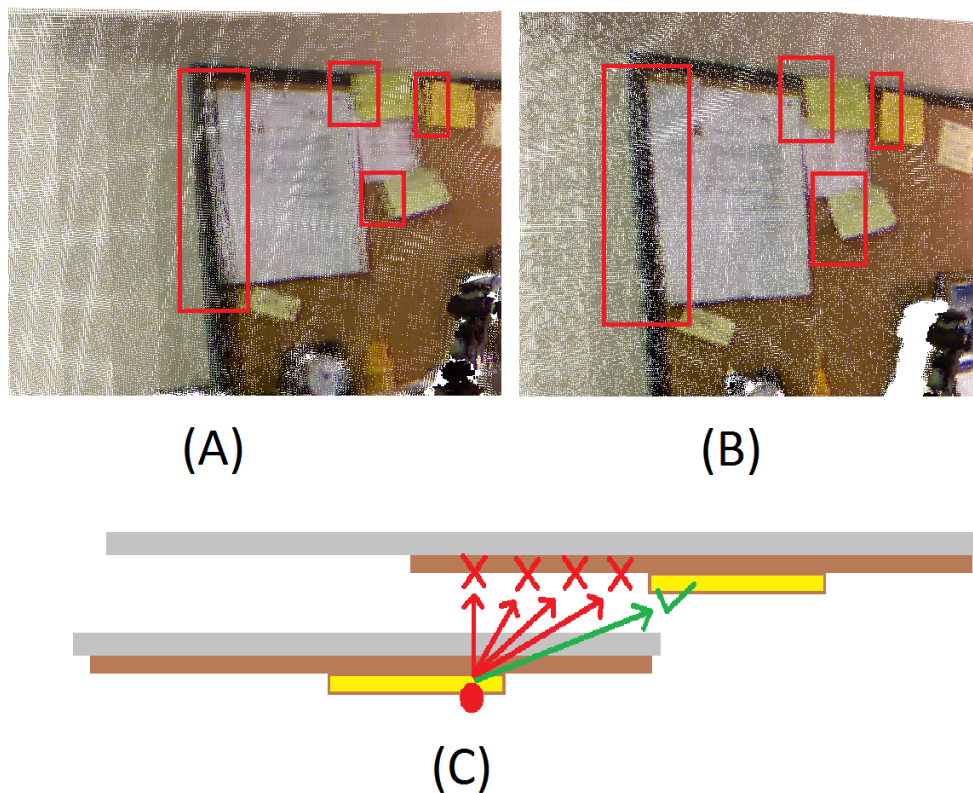


Figura 45: Comparação entre o método ICP clássico (esquerda) e o ICP com Threshold de cor (direita). A Ilustração abaixo mostra as tentativas de um ponto achar o par mais próximo mas dentro dos limites de diferença de cor impostas

convergir mais rápido e mais precisamente. Comparado ao método de descarte de pares acima de uma distância fixada após um número predeterminado de iterações usado nos resultados parciais, se apresentou muito mais adequado, principalmente para convergências lentas.

11 RECONTRUCAO DA MALHA UTILIZANDO MARCHING CUBES E DISTÂNCIA COM SINAL

Esta etapa se mostrou importante pela melhoria da precisão da malha final, como será mostrado a seguir.

11.1 Criação da Iso-Superfície

O cálculo da iso-superfície para cada aquisição do sensor pode ser resumida em 3 passos: Procura de voxels perto das nuvens de pontos, transformação da nuvem de pontos já alinhadas para coordenadas locais junto com os voxels selecionados, transformação da posição real dos voxels para a posição no mapa de profundidade e cálculo da distância.

A Figura 46 mostra uma cena com voxels perto das nuvens de pontos em destaque. Voxels vermelhos representam uma soma da função de distâncias negativa, os verdes representam uma soma positiva e os azuis representam voxels não acessados. Para cenas internas, voxels de 10 a 20mm de aresta se mostraram suficientes para capturar detalhes relativamente pequenos em relação da escala da cena como porta retratos e teclados de computador. Para objetos entre 1 e 2m, voxels de 2 a 4mm se mostraram adequados para a reprodução de detalhes.

Problemas foram encontrados quando a inclinação da superfície era

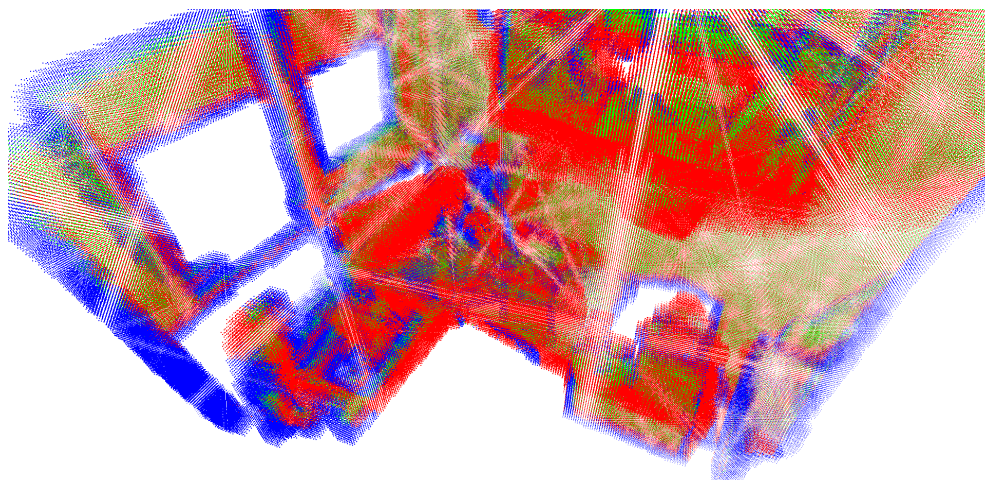


Figura 46: cálculo da iso-superfície por distância com sinal

grande em relação ao sensor, portanto foi implementado um threshold aonde a distancia é computada apenas se a superfície estiver com um angulo menor do que um ângulo Maximo escolhido pelo usuário, um angulo de aproximadamente 30 deg foi escolhido. Como a nuvem de pontos foi criada com discretização de distancias, os ângulos de um ponto a seu vizinho é sempre 0 deg ou aproximadamente 90 deg, por isso os ângulos foram calculados com vários vizinhos distantes, horizontal e verticalmente.

11.2 Adaptive Marching Cubes

O isoplano calculado na sessão anterior gerou uma malha com mais precisão do que o sensor do kinect. Quanto mais aquisições, mais precisa a malha final. É importante ressaltar que aquisições de um mesmo ponto de vista não aumentaram mais a precisão, pois é o movimento em conjunto com a soma das funções de distancia que eliminam as curvas de níveis e geram uma malha com precisão até duas vezes maior do que uma única aquisição do kinect. Pode-se ver este resultado na Figura 47, As curvas de nível desaparecem e detalhes menores são revelados.

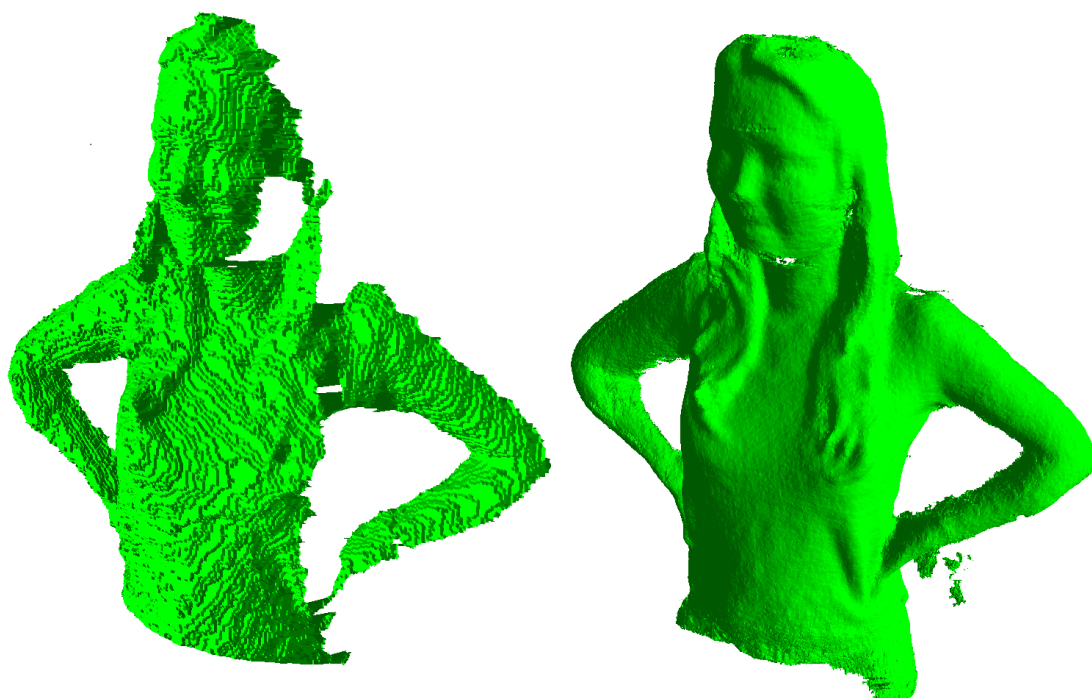


Figura 47: Reconstrução de malha de uma aquisição e 86 aquisições

Pode-se verificar também a diminuição do ruído randômico do sensor na Figura 48. Com o isoplano calculado com 43 aquisições o ruído é quase todo cancelado e com isso as teclas com aproximadamente 3mm de altura ficam aparentes.

Para reconstrução de cenas o aumento da precisão fica mais evidente, pois detalhes como um porta retrato de 20mm de espessura se torna aparente, assim como uma folha pregada em um mural, quando a precisão do kinect a esta distancia eh de 40mm, pode-se ver este resultado na Figura 49.

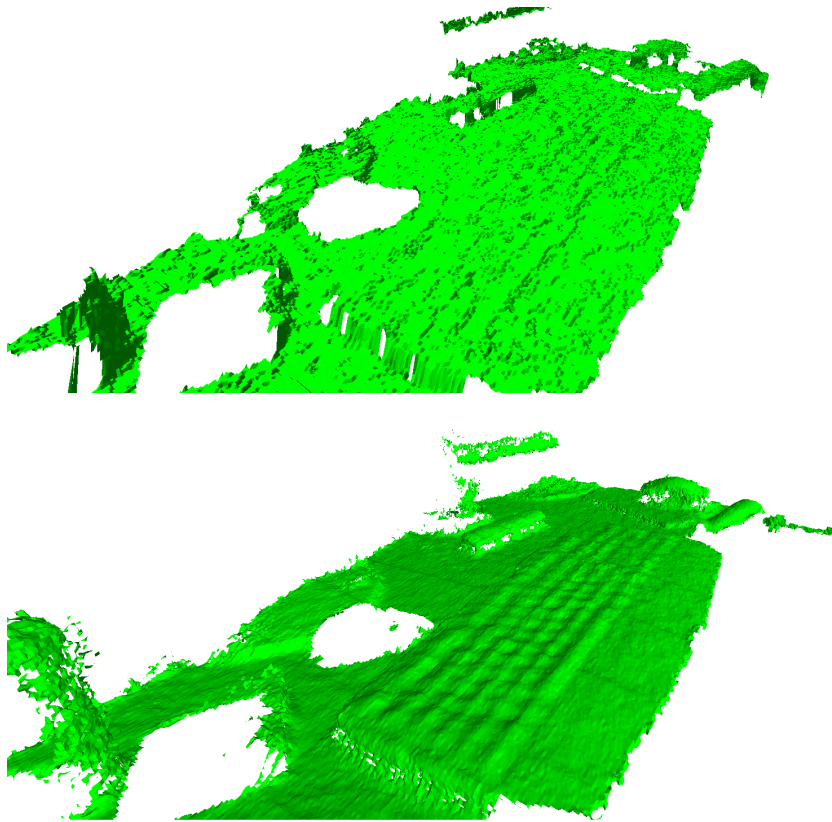


Figura 48: Reconstrução de um teclado com uma aquisição e 43 aquisições

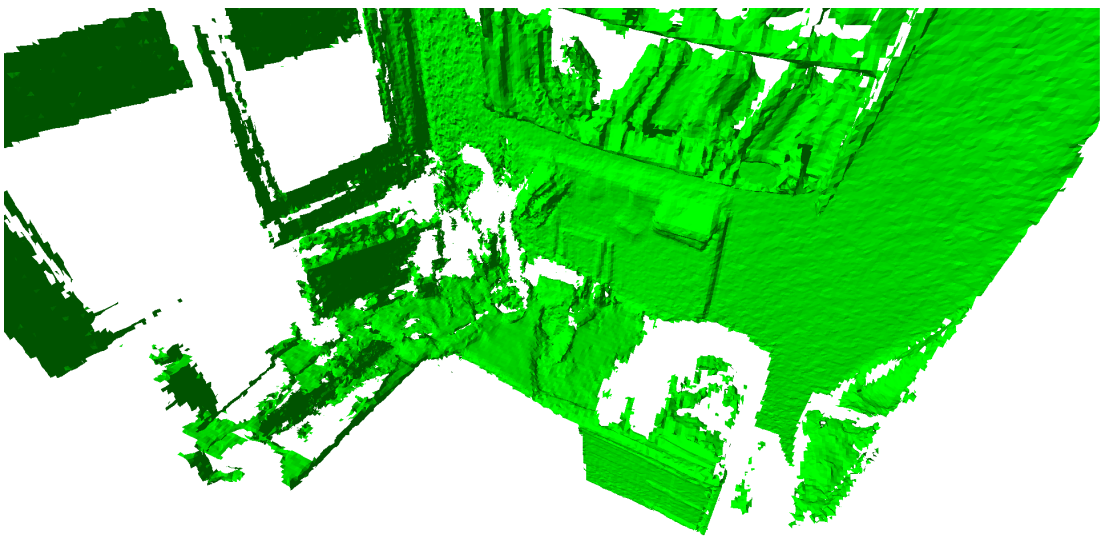


Figura 49: Reconstrucao de uma cena.

12 CONCLUSÃO

Os resultados mostram que é possível a reconstrução 3D utilizando o pipeline escolhido. O método de superfícies para a reconstrução da malha final foi descartada e o método volumétrico foi escolhido em seu lugar.

Os testes mostraram que o sensor é mais impreciso do que inicialmente suposto, mas sua precisão a curta distancia é o suficiente para detectar detalhes em torno de 3mm. Constatou o aumento do ruído em relação ao distanciamento da superfície ao sensor e a incapacidade do sensor de detectar objetos com especular idade alta devido a sua natureza ativa. As informações da câmera RGB integrada ao sensor infravermelho se mostraram úteis, conseguindo adicionar cor a nuvem de pontos apesar de erros em pixels nas bordas dos objetos e em superfícies muito inclinadas em relação ao sensor.

O processo de registro pelo ICP se mostrou eficiente e preciso, com o tempo de iteração para 5000 pontos de 50ms em media e 200ms para o pior caso, causado pelo distanciamento das nuvens nas primeiras iterações. A convergência para um mínimo local cria algumas limitações. A diferença entre dois mapas de profundidade consecutivos deve ser pequena para aumentar sua robustez e chance de convergir para o mínimo global e os cuidados devem ser tomados quando estiver registrando nuvens de pontos com pouca variação de geometria como paredes. A amostragem da nuvem de pontos para o calculo do ICP se mostrou útil para a diminuição do tempo de iteração

sem o aumento significativo do erro quadrático.

O refino de erros por distância com sinal se mostrou extremamente eficiente, não apenas por eliminar erros residuais do processo de registro, mas por filtrar ruídos das nuvens de pontos e aumentar a precisão final da malha, eliminando as discretizações de profundidade e mostrando detalhes dimensões até duas vezes menor que a precisão do sensor para determinada distância. Essa precisão aumenta com o aumento das aquisições das nuvens, com cada nuvem trazendo mais informação a malha final. Cada aquisição tem de estar em uma posição e rotação diferente das demais aquisições para esta melhora ocorrer. Caso o sensor se mantenha na mesma posição, apenas os ruídos randômicos serão filtrados, mas a discretização da profundidade e precisão se manterão os mesmos.

O processo de Adaptive Marching Cubes recriou o objeto com uma boa precisão sem ter de discretizar excessivamente o espaço. Pelo fato dos vértices poderem se mover ao longo das arestas, o isoplane pode ser recriado com muito mais fidelidade.

REFERÊNCIAS

- 1 ALVEZ, D. *CubicFrameStructureandFloorDepthMap.jpg*. 2011. Disponível em: http://en.wikipedia.org/wiki/File:Cubic_Frame_Structure_and_Floor_Depth_Map.jpg. Acesso em: 10 de dezembro de 2012.
- 2 POINT-CLOUD64.JPG. 2011. Disponível em: http://www.shapecapture.com/SC_poly.htm. Acesso em: 10 de dezembro de 2012.
- 3 BERNARDINI, H. E. R. F. The 3d model acquisition pipeline. *Comput. Graph Forum*, Blackwell, New York, USA, v. 21, n. 2, p. 149–172, 2002.
- 4 ATKINSON, G. *Face recognition using photometric stereo*. 2012. Disponível em: <http://www1.uwe.ac.uk/et/mvl/projects/facerecognition.aspx>. Acesso em: 10 de dezembro de 2012.
- 5 OLSSON, T. P. K. Shape from silhouette scanner. 1997.
- 6 ZHANG BRIAN CURLESS, S. S. L. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In: *In The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*. Padova, Italy: [s.n.], 2002. p. 24–36.
- 7 GRIFFIN LAKSHMI S. NARASIMHAN, S. R. Y. P. M. Generation of uniquely encoded light patterns for rangedataacquisition. *Pattern Recognition*, v. 25, n. 6, p. 609–616, 1992.
- 8 SALVIA SERGIO FERNANDEZA, T. P. J. A state of the art in structuredlightpatterns for surface profilometry. *Pattern Recognition*, v. 43, n. 8, p. 2666–2680, 2010.
- 9 KOK, L. L. Linear least-squares optimization for point-to-plane icp surface registration.
- 10 CURLESS, M. L. B. A volumetric method for building complex models from range images. *Computer Graphics*, v. 30, p. 303–312, 1996.
- 11 TURK, M. L. G. Zippered polygon meshes from range images. *Proceedings of SIGGRAPH 94*, p. 311–318, 1994.
- 12 SHARMAN, J. *mc2.gif*. Disponível em: <http://www.exaflop.org/docs/marchcubes/>. Acesso em: 10 de dezembro de 2012.
- 13 KINECT.JPEG. 2012. Disponível em: <http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx>. Acesso em: 10 de dezembro de 2012.

- 14 FREEDMAN, A. S. B. *Distance-Varying Illumination and Imaging Techniques for Depth Mapping*. 2010. Patente US 2010/0290698.
- 15 MEDIANALGORITHM.GIF. Disponível em: <http://www.scimedia.com/fis/support/download/bva/ver0703/RelNote-V0703.html>. Acesso em: 10 de dezembro de 2012.
- 16 SALVI CARLES MATABOSCH, D. F. J. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, Butterworth-Heinemann, Newton, MA, USA, v. 25, n. 5, p. 578–596, 2007.
- 17 ZHANG, Z. Iterative point matching for registration of freeform curves and surfaces. *International Journal of Computer Vision*, Academic Publishers, France, v. 13, n. 2, p. 119–152, 1994.
- 18 RUSINKIEWICZ, S. Efficient variants of the icp algorithm. *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference*, IEEE Computer Society, Washington, DC, USA, p. 145–152, 2001.
- 19 CHETVERIKOV, D. Fast neighborhood search in planar point sets. *Pattern Recognition Letters*, Elsevier Science, New York, NY, USA, v. 12, n. 7, p. 409–412, 1991.
- 20 BESL, N. D. M. P. J. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 14, n. 2, p. 239–256, 1992.
- 21 JOHNSON, S. B. K. A. E. Registration and integration of textured 3-d data. *NRC '97 Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, p. 234, 1997.
- 22 KHOSHELHAM, K. Accuracy analysis of kinect depth data. 1997.
- 23 IZADI DAVID KIM, O. H. S. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. 2010.
- 24 PFEIFER, C. B. N. *Three-Dimensional Mapping with Time-of-Flight Cameras*. 2007. Não publicado. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.130.1824>. Acesso em: 10 de dezembro de 2012.
- 25 SOBOTA MAROS ROVNAK, C. S. B. 3d scanner data processing. *Journal of Information, Control and Management Systems*, Blackwell, Slovak Republic, v. 7, n. 2, p. 175, 2009.
- 26 PHASE-BASED vs time-of-flight. 2012. Disponível em: <http://www.smartgeometrics.com/rentals/phase-based-vs-time-of-flight/>. Acesso em: 10 de dezembro de 2012.
- 27 MAY DAVID DROESCHEL, D. H. S. Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, Wiley Periodicals, v. 26, n. 11, p. 934–975, 2009.

- 28 WALKER LEJUN SHAO, R. A. V. M. W. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, Academic Press, Orlando, FL, USA, v. 54, n. 3, p. 358–367, 1991.
- 29 ARUN T. S. HUANG, S. D. B. K. S. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 9, n. 5, p. 698–700, 1987.
- 30 HORN H.M. HILDEN, S. N. B. K. P. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, Academic Press, Cambridge, Massachusetts, USA, v. 5, n. 7, p. 1127–1135, 1988.
- 31 HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, Academic Press, Cambridge, Massachusetts, USA, v. 4, n. 4, p. 629–642, 1987.
- 32 EGGERT A. LORUSSO, R. B. F. D. W. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Journal of the Optical Society of America*, Springer-Verlag, New York, USA, v. 9, n. 5-6, p. 272–290, 1997.
- 33 CHEN, G. M. Y. Object modelling by registration of multiple range images. *Image and Vision Computing - Special issue: range image understanding*, Butterworth-Heinemann, Newton, MA, USA, v. 10, n. 3, p. 145–155, 1992.
- 34 BERN, D. E. M. Mesh generation and optimal triangulation. *Technical Report P92-00047*, 1992.
- 35 SHU CHEN ZHOU, M. S. K. R. Adaptive marching cubes. *THE VISUAL COMPUTER*, v. 11, p. 202–217, 1995.
- 36 TAKAHASHI, D. *Xbox 360 surpasses 66M sold and Kinect passes 18M units*. 2012. Venturebeat. Disponível em: <http://venturebeat.com/2012/01/09/xbox-360-surpassed-66m-sold-and-kinect-has-sold-18m-units/>. Acesso em: 10 de dezembro de 2012.
- 37 ZINSSER H. SCHNIDT, J. N. T. A refined icp algorithm for robust 3-d correspondences estimation. *International Conference on Image Processing*, p. 695–698, 2003.
- 38 HUGLI, J. H. A multi-resolution scheme icp algorithm for fast shape registration. *First International Symposium on 3D Data Processing Visualization and Transmission*, 2002.
- 39 CHOW H. TSUI, T. L. C. Surface registration using a dynamic genetic algorithm. *Pattern Recognition*, v. 37, n. 1, 2004.
- 40 JOST, H. H. T. A multi-resolution scheme icp algorithm for fast shape registration. *First International Symposium on 3D Data Processing Visualization and Transmission*, p. 540–543, 2002.

- 41 SIMON, D. A. Fast and accurate shape-based registration. Pittsburgh, PA, USA, 1996.
- 42 PULLI, K. A. Surface reconstruction and display from range and color data. 1997.
- 43 MASUDA K. SAKAUE, N. Y. T. Registration and integration of multiple range images for 3-d model construction. *ICPR 96 Proceedings of the 1996 International Conference on Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, v. 1, p. 879, 1996.
- 44 DORAI GANG WANG, A. K. J. C. Registration and integration of multiple object views for 3d model construction. *EEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 1, p. 83–89, 1998.
- 45 FOLEY, J. D. Fundamentals of interactive computer graphics. Addison-Wesley Publishing Company, 2002.