

**UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
NÚCLEO AVANÇADO DE NOVA CRUZ
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JURASILDO OLIVEIRA REINALDO

**RECONSTRUÇÃO DE SUPERFÍCIES EM TEMPO REAL USANDO VISÃO
ESTÉREO**

**NOVA CRUZ - RN
OUTUBRO/2012**

JURASILDO OLIVEIRA REINALDO

**RECONSTRUÇÃO DE SUPERFÍCIES EM TEMPO REAL USANDO VISÃO
ESTÉREO**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação.

ORIENTADOR: Prof. Dr. Anderson Abner de Santana Souza

CO-ORIENTADOR: Profa. Ms. Rosiery da Silva Maia

NOVA CRUZ - RN

OUTUBRO/2012

JURASILDO OLIVEIRA REINALDO

**RECONSTRUÇÃO DE SUPERFÍCIES EM TEMPO REAL USANDO VISÃO
ESTÉREO**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação.

Aprovado em: **15/10/2012**

Conceito Final: **10.0**

Banca Examinadora

Profa. Dra. Adriana Takahashi (**UERN**)

Prof. Dr. Anderson Abner de Santana Souza (**UERN**)

Profa. Ms. Rosiery da Silva Maia (**UERN**)

Dedico esse trabalho a todos aqueles que fizeram parte direta ou indiretamente de minha vida durante esses quatro anos, em especial a minha família.

AGRADECIMENTOS

Durante esses quatro anos muitas coisas aconteceram: coisas boas e coisas ruins, o que importa é o que levamos de conhecimento disso tudo. Muitas amizades foram feitas e novos costumes assimilados, tudo isso buscando novos horizontes.

Agradeço a Dona Maria e seu filho César, que me abriram as portas de sua casa, para receber um estranho, que tinha apenas como referência a indicação de sua professora Tânia (*ensino médio*). Assim como outras tantas pessoas que conheci na cidade de Nova Cruz, amigos e vizinhos que passaram a fazer parte de minha vida e de tantos outros estudantes que ali chegaram.

Agradeço aos amigos Weskilley, Sebastião e Antônio Neto da cidade de Riacho da Cruz, onde passei aproximadamente seis meses de convivência e considero uma escola de vida, principalmente pela força que deram em optar pela faculdade e deixar o emprego da prefeitura.

Agradeço aos amigos da cidade de Francisco Dantas, que me acompanham desde a época do ginásio.

Agradeço a paciência e a atenção dada pelo meu orientador Anderson Abner em todo o decorrer da confecção desta monografia.

Agradeço a minha família que me ajudou no que foi preciso, mesmo dispondo de poucos recursos financeiros sempre estava ali para o que precisava.

Agradeço a Deus, pois foi ele quem me guiou até essas pessoas que hoje fazem parte de minha vida e sem ele esse trabalho não estaria sendo concluído.

Obrigado a todos! Obrigado meu Deus!

Com a visão, o infinito nos é dado de uma só vez; a riqueza é sua descrição.

(Caleb Gattegno)

RESUMO

Diante de muitas pesquisas realizadas, a visão biológica tornou-se um referencial para o surgimento do campo da visão computacional. Com o estudo das propriedades da visão estéreo, descobriu-se a possibilidade de se obter a profundidade dos objetos, simulando a visão biológica através de duas câmeras. De posse desses valores de profundidade torna-se possível utilizá-los nas mais variadas aplicações: na robótica, para saber a distância dos objetos e evitar colisões; na realidade aumentada, para conseguir interatividade e especialmente na reconstrução de superfícies. Sabendo da aplicabilidade da reconstrução 3D, resolveu-se desenvolver uma ferramenta de reconstrução de superfícies, que utilizando uma webcam 3D, fosse possível extrair informações tridimensionais do ambiente e projetar essas coordenadas num espaço virtual 3D, através de uma nuvem de pontos. Todo o processamento de captura e reconstrução das informações é feito em tempo real e paralelamente visualizado em um ambiente da biblioteca PCL (*Point Cloud Library*). Para alcançar os resultados desejados foi necessário passar por várias etapas conceituais desde a calibração de câmera até a criação de um mapa de disparidades que serão expostas durante a descrição deste trabalho. Dessa forma, preocupou-se em fazer uma aplicação totalmente com software livre com equipamentos de baixo custo, que possibilite ser reutilizada para projetos de baixo orçamento.

PALAVRAS-CHAVES: Visão Estéreo, Reconstrução de Superfícies, Nuvem de Pontos.

ABSTRACT

Faced with many researches, the biological vision became a reference to the emergence of the field of computer vision. Studying the properties of stereo vision, it has been found the possibility to obtain the depth of objects, simulating the biological eyes through two cameras. With these values of depth, it is possible to use them in different applications: in robotics, to find the distance of objects and obstacle avoidance; in augmented reality to achieve interactivity, especially in the reconstruction of surfaces used in many areas of science. Knowing the field of applicability of 3D reconstruction, this work proposes the development of a computational tool for tridimensional surfaces reconstruction using a 3D webcam, through points cloud. With this tool, it was possible to extract three-dimensional information of the environment and redesign these coordinate data in a 3D virtual space. The capture and reconstruction of information is done in real time and simultaneously displayed on a viewer of the library PCL (*Point Cloud Library*). To achieve the desired results it was necessary to pass through several steps ranging from conceptual calibration camera to disparity map estimation, which will be described in this work. This tool was implemented using free software with low-cost equipment, which enables reuse for inexpensive projects.

KEYWORDS: Stereo Vision, Surface Reconstruction, Points Cloud.

LISTA DE ABREVIATURAS E SIGLAS

2D: Duas dimensões

3D: Três dimensões

API: Application Programming Interface

BSD: Termo de licença que designa um software de domínio público

CCD: Charge Coupled Device

cm: Centímetros

CMOS: Complementary Meta-Oxide Semiconductor

DSI: Disparity Space Image

ELAS: Efficient LArger-scale Stereo

ms: Milissegundos

NURBS: Non-Uniform Rational B-Splines

OpenCV: Open Source Computer Vision

PCL: Point Cloud Library

SAD: Sum of Absolute Differences

SIFT: Scale Invariant Feature Transform

SURF: Speeded Up Robust Features

V4L2: Documentação para drivers de vídeo Linux

LISTA DE QUADROS

Quadro 1: Parâmetros intrínsecos e extrínsecos da webcam Minoru.	40
Quadro 2: Equipamentos utilizados para desenvolvimento da aplicação.	43
Quadro 3: Dados comparativos da distância real do objeto em relação à distância calculada pela câmera.	46
Quadro 4: Relação do tempo de processamento da aplicação.	50

LISTA DE FIGURAS

Figura 1: Representação da Distância Focal.	18
Figura 2: Fundamentos da Perspectiva.....	19
Figura 3: Representação da visão binocular.....	20
Figura 4: Modelo de representação da geometria estéreo num sistema de câmeras.....	22
Figura 5: Representação da Geometria Epipolar.....	23
Figura 6: Projeção de câmera em relação a um ponto no mundo.	25
Figura 7: Gabarito de calibração. (A) Intersecção das linhas verticais e horizontais indicando os vértices dos quadrados, (B) Altura e largura dos quadrados.....	27
Figura 8: Método de calibração baseado em objetos de referência 3D.....	27
Figura 9: Método de calibração de Zhang.....	28
Figura 10: Retificação com base na geometria epipolar.	30
Figura 11: Diferença entre antes e depois de aplicar a retificação num par de imagens.	31
Figura 12: Representação de um DSI. Sendo (A) Os planos do DSI, (B) Imagens estéreo esquerda e direita, (C e D) Representam o corte no DSI na posição $y=151$, onde a linha vermelha indica o caminho de minimização de energia (regiões mais escuras).	32
Figura 13: Janelas de Pixels. (A) Janela esquerda extática, (B) Janela direita móvel.	33
Figura 14: Calculo da disparidade relacionando a região dos pontos oclusos.	35
Figura 15: Mapa de disparidades. (A) Par de imagens estéreo, (B) Imagem em tons de cinza representando o mapa de disparidades obtido com o par imagens acima.	36
Figura 16: Método Multiview-Stereo.....	38
Figura 17: Fluxograma dos estágios necessários para se realizar a reconstrução de uma superfície.	39
Figura 18: Webcam 3D Minoru.	42
Figura 19: Diferentes pontos de vista de uma reconstrução.	47
Figura 20: Reconstrução da superfície de uma caixa. (A) Imagem esquerda e direita retificada, (B) Vista com a mesma pose das imagens capturadas, (C) Vista superior, (D) Vista da lateral direita da reconstrução e (E) Vista da lateral esquerda.....	48
Figura 21: Reconstrução realizada com distância em torno de 40 centímetros da câmera. (A) Imagem superior dos objetos, (B) Vista superior da reconstrução, (C) Vista da lateral esquerda da reconstrução, (D) Vista superior rotacionada e (E) Vista superior.....	48

Figura 22: Reconstrução realizada com distância aproximada de 80 centímetros. (A) Vista com a mesma pose das imagens capturadas. (B) Vista frontal, (C) Vista por trás da reconstrução, (D) Vista da lateral esquerda, (E) Vista da lateral direita e (F) Vista inferior relacionando a parte das pernas da cadeira.....49

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Objetivo Geral.....	15
1.2	Objetivos Específicos	16
1.3	Justificativa	16
1.4	Estrutura do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Captura de Imagens por Câmera.....	18
2.2	Visão Monocular.....	19
2.3	Visão Estéreo	19
2.4	Geometria Estéreo.....	21
2.5	Geometria Epipolar.....	23
2.6	Calibração de Câmeras	24
2.6.1	Parâmetros intrínsecos.....	24
2.6.2	Parâmetros Extrínsecos	24
2.6.3	Técnicas de calibração	26
2.6.3.1	Calibração baseada em objetos 3D ou método de Tsai.	27
2.6.3.2	Calibração baseada em imagens 2D ou método de Zhang	27
2.6.3.3	Auto-calibração	28
2.7	Retificação de Imagens	29
2.8	Correspondência entre Pontos	31
2.8.1	Algoritmos Globais	32
2.8.2	Algoritmos Locais	33
2.9	Mapa de Disparidade	34
2.10	Reconstrução 3D a partir de Visão Computacional	36
2.10.1	Shape From Focus	37

2.10.2	Shape From Shading.....	37
2.10.3	Structure From Motion	37
2.10.4	Shape From Silhouette	37
2.10.5	Structure From Stereo.....	38
3	DESENVOLVIMENTO E IMPLEMENTAÇÃO	39
3.1	Metodologia Adotada no Desenvolvimento da Aplicação	39
3.2	Escopo da Aplicação.....	42
3.2.1	Biblioteca OpenCV (<i>Open Source Computer Vision</i>).....	43
3.2.2	Biblioteca PCL (<i>Point Cloud Library</i>).....	44
3.2.3	Software v4l2stereo.....	44
3.2.4	Classe Stereo	45
4	EXPERIMENTOS E ANÁLISES	46
4.1	Tempo de Processamento	46
4.1.1	Mapa de disparidades	46
4.1.2	Reconstrução da superfície.....	46
5	CONCLUSÕES E TRABALHOS FUTUROS.....	51
	REFERÊNCIAS.....	52

1 INTRODUÇÃO

No nosso dia-a-dia recebemos muitas informações relacionadas a cores, formas e distância dos objetos presente no ambiente a nossa volta. À medida que vamos crescendo e aprofundando nossas experiências, nosso cérebro vai cruzando essas informações e possibilitando a tomada de decisões com base naquilo que vemos e sentimos (BRADSKI & KAEHLER, 2008).

A visão computacional é uma área que tem como objetivo transformar dados provenientes de imagens ou vídeos em alguma decisão ou em uma nova representação (BRADSKI & KAEHLER, 2008). Para o computador uma imagem representa somente um conjunto de dados relacionados à cor, brilho, intensidade, contraste entre outras informações. A capacidade de tomar uma decisão ou gerar uma nova representação com base nesses valores é resultado dos algoritmos de visão computacional desenvolvidos para manipular e processar dados provenientes de imagens. Por exemplo, uma câmera de segurança conectada a um computador fica responsável por controlar a entrada de pessoas autorizadas a uma sala através do reconhecimento facial. Nesse caso a decisão tomada pelo algoritmo de visão computacional seria a de abrir a porta somente para as pessoas autorizadas e os procedimentos feitos em cima das imagens adquiridas para se extrair as informações seriam as novas representações.

A visão computacional está a cada dia ganhando mais espaço em áreas como robótica, medicina, automação e realidade aumentada. No entanto, desenvolver aplicações ligadas à visão computacional não é uma tarefa fácil, algo que é possível fazer facilmente como, identificar um objeto no meio de vários outros ou estimar a distância dos mesmos, para um computador são necessários vários procedimentos para se conseguir o mesmo objetivo.

A visão computacional veio para simular e ampliar nossa capacidade visual, contornando as limitações imposta pela natureza. Isso pode ser compreendido ao se observar uma folha de papel branca: aparentemente a folha é totalmente homogênea, mas na verdade existem vários tons de branco que os olhos não conseguem enxergar. Esses diferentes níveis de branco podem ser vistos tirando uma imagem por uma câmera e verificando os valores dos pixels. Esses valores são resultantes de fatores externos causados principalmente pela iluminação que atingi de forma não linear a superfície da folha.

Como ciência dedicada à extração e interpretação das informações presentes em imagens, a visão computacional vem se utilizando das propriedades da visão estéreo (por

exemplo, a visão humana) para criar sistemas com duas ou mais câmeras capazes de obter informações tridimensionais do ambiente onde estão situadas.

A capacidade de obter a profundidade com base num par de imagens estéreo teve sua inspiração vinda da natureza, a partir do estudo dos sistemas visuais dos seres vivos. Com características similares aos olhos humanos, um par de câmeras pode ser configurado de tal forma que torne possível obter dados sobre a profundidade dos objetos presentes num ambiente por meio de cálculos da geometria do sistema estéreo. Informações de profundidade são essenciais para a maioria das aplicações de visão computacional que necessitam ter a noção de distância e localização num ambiente. Uma delas é a reconstrução de superfícies, que se utiliza da visão computacional como meio de captura dos dados tridimensionais para modelar geometricamente superfícies de objetos em um ambiente.

Algumas aplicações realizam a reconstrução de superfícies trabalhando sobre um conjunto de imagens estéreo tiradas previamente, o que lhes permite trabalhar com imagens com muito mais informação sem se preocupar com o tempo; outras aplicações recebem como entrada um par de imagens estéreo e reconstrói a cena em tempo real. No entanto, existem muitos trabalhos como, (MARQUES; CABRAL; PEIXOTO, 2012), (OLIVEIRA, 2007), (NEVES & TSUZUKI, 2012), (CAMARGO, 2008) e (AZEVEDO, 2009), que não precisam que as imagens sejam providas por um sistema estéreo para se realizar a reconstrução, necessitando-se apenas de duas ou mais imagens captadas em diferentes ângulos; é o caso dos sistemas com uma única câmera, também chamados de sistemas monoculares. O problema dessa técnica é que não é possível estabelecer uma medida real de distância sem antes estabelecer um fator de inicialização métrico. Não é possível conhecer a distância em metros apenas com duas imagens tomadas a partir de pontos de vistas diferentes se não houver o conhecimento métrico sobre a posição dos pontos de vistas. A principal vantagem dos sistemas estéreos sobre os monoculares é a possibilidade de estimar a profundidade de forma direta, sem a necessidade de uma inicialização métrica prévia.

1.1 Objetivo Geral

Diante do contexto apresentado, o objetivo desse trabalho é desenvolver uma ferramenta de reconstrução 3D de superfícies em tempo real por meio de nuvens de pontos, utilizando como sensor de aquisição de dados uma webcam estéreo que irá enviar as imagens capturadas para serem processadas por um computador portátil. Essa ferramenta poderá ser utilizada em aplicações da área de robótica como, por exemplo, navegação com desvio de

obstáculos e reconhecimentos de objetos; poderá ser também aplicada em realidade virtual em modelagem de ambientes 3D entre outras.

1.2 Objetivos Específicos

- Inicialmente, investigar toda a teoria envolvida na visão estéreo, analisando desde a captura de imagens a partir de câmeras estéreo aos cálculos geométricos necessários para obter coordenadas 3D dos pontos detectados.
- Estimar os parâmetros internos e externos da câmera através da calibração estéreo.
- Aplicar o processo de retificação nas imagens capturadas, para retirar efeitos de distorções.
- Realizar a correspondência de pontos no par de imagens retificadas.
- Gerar um mapa de disparidade contendo as informações tridimensionais¹ dos pontos homólogos² retirados das imagens.
- Reconstruir a superfície dos objetos através da projeção das coordenadas dos pontos contidos no mapa de disparidades, usando nuvens de pontos.
- Testar a qualidade da reconstrução com base em comparações entre a profundidade calculada e a profundidade real.

1.3 Justificativa

Diante dos inúmeros trabalhos realizados sobre o tema reconstrução de superfícies, percebeu-se que grande parte deles utiliza sensores caros como o *lazer* e o Kinect (XBOX, 2012) da Microsoft, utilizados para extração de dados tridimensionais. Resolve-se nesse trabalho utilizar uma webcam como sensor de captura dos dados de profundidade, por ser um equipamento de baixo custo se comparado aos dispositivos citados anteriormente e

¹ Referente aos valores das três dimensões: x, y e z.

² Pontos da imagem que possuem características semelhantes entre si.

pela grande quantidade de informações que as câmeras podem fornecer (cor, textura, movimento, entre outras) em comparação a sensores como sonar que apenas oferece informações sobre o tempo de ida e volta das ondas sonoras emitidas. Outro fator que contribuiu para a utilização das câmeras nesse trabalho foi o de acreditar na popularidade das câmeras para o crescimento das aplicações de visão computacional nos diversos setores da sociedade.

1.4 Estrutura do Trabalho

Esse trabalho está dividido em cinco capítulos: O primeiro capítulo faz uma rápida análise do problema a ser resolvido, mostrando os objetivos e justificativas do trabalho proposto. O segundo capítulo contém toda a teoria necessária para desenvolvimento da ferramenta: captura das imagens, retificação, correspondência de pontos, geração do mapa de disparidades e reconstrução. São todas partes interligadas que necessitam seguir a sequência estabelecida para um completo entendimento. O terceiro capítulo é formado pelo escopo da ferramenta, explicando os equipamentos e métodos utilizados para cada etapa do desenvolvimento do sistema. O quarto capítulo está relacionado à parte de testes e análises dos resultados, verificando e comparando os dados coletados com os dados reais. O quinto capítulo contém as considerações finais sobre a aplicação e as propostas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Captura de Imagens por Câmera

A maioria das pessoas que possuem câmeras fotográficas digitais, nem param para pensar como funciona o processo de captura das imagens dentro delas. A formação das imagens é produzida pela incidência dos raios luminosos sobre um sensor que fica por traz da lente, responsável por armazenar os valores correspondentes à magnitude dos raios luminosos e também pela qualidade (resolução) das imagens tiradas. Estão entre os mais utilizados atualmente os sensores CCD (*Charge Coupled Device*) e CMOS (*Complementary Meta-Oxide Semiconductor*) (STEMMER, 2005).

A distância que existe entre o centro da lente e o sensor é conhecida como distância focal e varia de câmera para câmera. Essa distância limita o tamanho do campo de visão que é projetado no sensor ou no plano de imagem equivalente. (BRADSKI & KAEHLER, 2008) comentam a distância focal utilizando como base um dos modelos mais simples de câmera chamado *pinhole*. A Figura 1 mostra a relação de quanto maior a distância focal, menor o campo de visão.

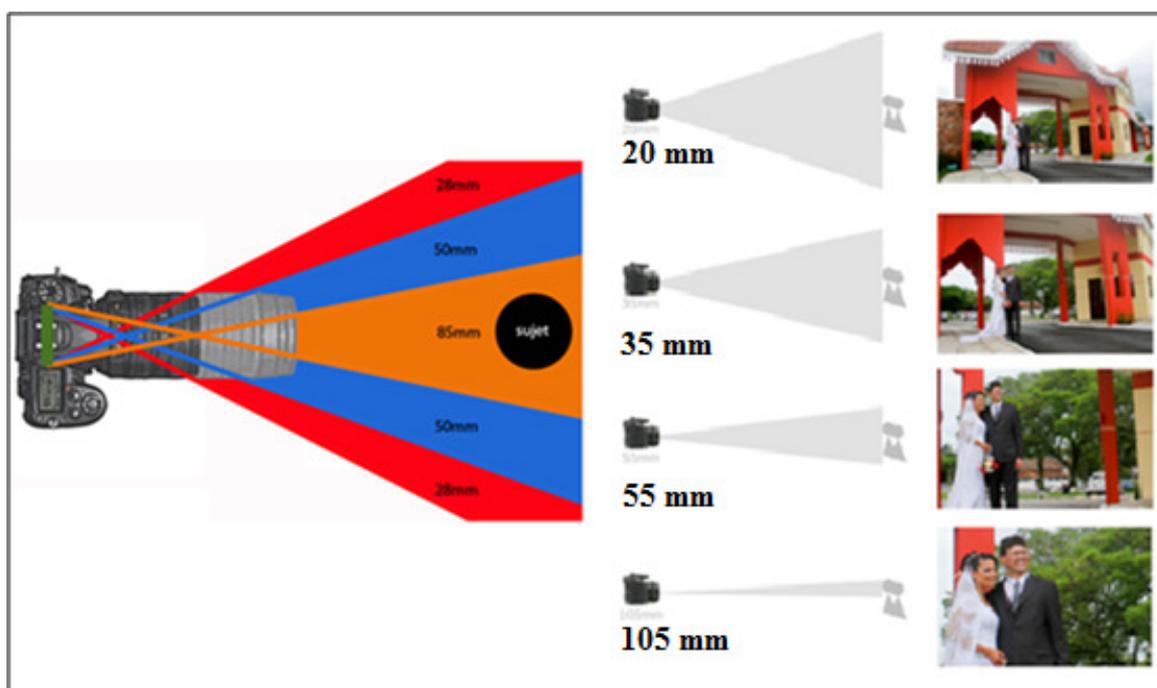


Figura 1: Representação da Distância Focal.

Fonte: (ACHARD, 2012).

Existem duas formas de se ver o ambiente utilizando o sentido visual biológico: a visão monocular e estéreo.

2.2 Visão Monocular

A visão monocular é uma forma primitiva e rudimentar de se ver o mundo, tendo como fundamento as leis da perspectiva (KIRNER & TORI, 2004). Através dessas leis é possível ter a idéia da distância dos objetos com base no seu tamanho. Quanto mais próximo um objeto está do observador, maior será seu tamanho em relação aos demais objetos da cena. Isso é conseguido através do posicionamento de um ponto de fuga no horizonte, como pode ser visto na Figura 2, onde o ponto de fuga está representado por **PF** e a linha horizontal por **LH**. As linhas pontilhadas representam as linhas de fuga que surgem do centro do ponto **PF** em direção ao infinito.

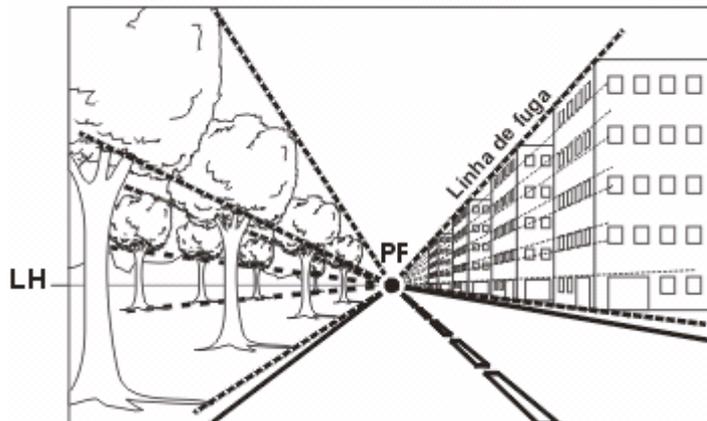


Figura 2: Fundamentos da Perspectiva.

Fonte: (SOBRE ARTE, 2012).

2.3 Visão Estéreo

O ser humano, assim como a maioria dos animais, possui a chamada visão binocular ou visão estéreo. A Visão estéreo é caracterizada pela obtenção das características tridimensionais dos objetos presente no ambiente, através da captação das imagens por cada olho projetadas sobre as duas retinas. Como os olhos possuem certa distância um do outro, no caso do ser humano aproximadamente 65 milímetros (KIRNER & TORI, 2004), possibilita obter duas imagens de um mesmo ponto no espaço de perspectivas diferentes.

Quando o cérebro recebe as duas imagens, ele verifica informações sobre o grau de convergência dos eixos visuais, deslocados pelos músculos dos globos oculares e o valor da disparidade retinal causada pela diferença de projeção de um mesmo ponto em regiões diferentes da retina. Na Figura 3 tem-se a representação da visão estéreo, sendo a Figura 3(A) a ilustração das imagens capturadas por cada olho com perspectivas diferentes e a Figura 3(B) a representação geométrica da projeção dos pontos O , A , B no mundo. Sendo F_E , T_E e N_E as respectivas projeções de O , A e B no plano da imagem esquerda e F_D , N_D e T_D as projeções dos mesmos pontos no plano da imagem direita. As variáveis d e e mostram o grau de convergência dos eixos visuais direito e esquerdo respectivamente, dados pelos ângulos formados pelos segmentos $\overline{N_E B}$, $\overline{T_D B}$, $\overline{T_E A}$ e $\overline{N_D A}$ em relação ao eixos ópticos $\overline{F_E O}$ e $\overline{F_D O}$.

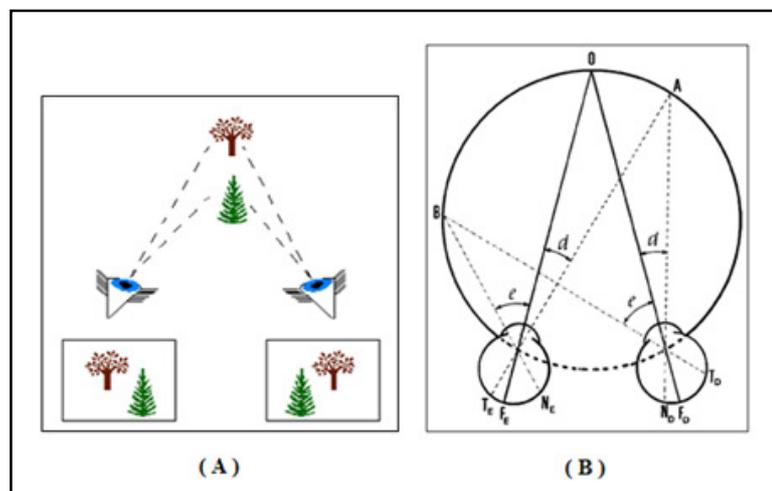


Figura 3: Representação da visão binocular.

Fonte: (KIRNER & TORI, 2004).

Com o estudo da visão estéreo e os avanços da tecnologia, conseguiu-se recriar matematicamente as características da visão biológica. Com isso, tornou-se possível gravar e ver imagens em dispositivos eletrônicos simulando a mesma sensação de profundidade obtida nas imagens vistas no dia-a-dia.

O funcionamento da visão estéreo em câmeras é baseado no funcionamento do sentido visual biológico. O que muda da visão estéreo para o sistema de câmeras é quem cria a estereoscopia, pois, nesse caso, o responsável que vai pegar as duas imagens capturadas e junta-las de forma que seja possível ter a noção de profundidade, é o computador. Com os sistemas computacionais usando algoritmos e fórmulas matemáticas sofisticadas, é possível conseguir informações muito precisas, com erros em torno de milímetros em sistemas de visão bem calibrados e usando equipamentos de boa qualidade (HOLANDA et al, 2010).

2.4 Geometria Estéreo

O cálculo da geometria estereo é compreendido de forma mais fácil com auxílio da Figura 4, partindo-se da idéia que as câmeras esquerda e direita estejam perfeitamente alinhadas e que possuam a mesma distância focal. Sendo essas propriedades satisfeitas, a profundidade por um par de imagens estereo é conseguida, através da semelhança de triângulos ou triangulação, demonstrado por (BRADSKI & KAEHLER, 2008).

Para se aplicar a triangulação, necessita-se dos valores da distancia focal f e da distância T (distância entre os centros de projeção O' e O , ou simplesmente, distancia entre as câmeras). Também necessita-se das coordenadas do centro do plano de imagem da câmera esquerda (x_0', y_0') , onde passa o eixo óptico, além dos valores das coordenadas de projeção (x', y') da imagem esquerda e (x, y) da imagem direita.

Quando o ponto P é projetado nos planos da imagem esquerda e da imagem direita simultaneamente, é possível saber as coordenadas nos planos de imagem em que o ponto foi projetado. Isso porque essas coordenadas correspondem diretamente aos pixels do sensor da câmera, que recebe o feixe de luz refletido na superfície do objeto, equivalente ao ponto P . Como as câmeras geralmente estão em posições diferentes, as coordenadas do ponto P também serão diferentes de um plano para o outro. As coordenadas (x', y') da imagem esquerda são diferentes das coordenadas (x, y) da imagem direita. Essa diferença é chamada de disparidade, o mesmo fenômeno que ocorre no nosso olho, com a diferença que nos olhos ocorre a disparidade retinal (KIRNER & TORI, 2004).

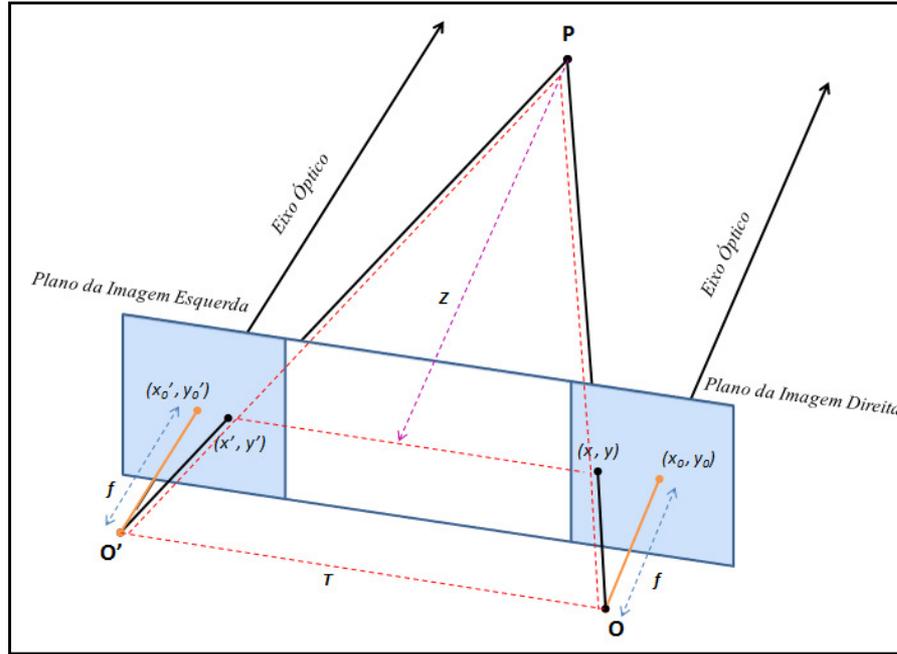


Figura 4: Modelo de representação da geometria estereóscópica em um sistema de câmeras.

A profundidade Z em relação ao ponto P , até os planos da imagem, é conseguido aplicando-se a Equação 1 (BRADSKI & KAEHLER, 2008), (SOUZA, 2011), (NOGUEIRA & TOZZI, 1999). Após encontrar o valor de Z , pode-se achar os demais valores das coordenadas X e Y do sistema estereóscópico através das Equações 2 e 3, encontradas em (SOUZA, 2011).

Nessas equações foram utilizadas as coordenadas do centro do eixo óptico da imagem esquerda tomada como padrão. No entanto, isso não impossibilita que se usem as coordenadas da imagem direita, tomando o cuidado de fazer antes o cálculo da disparidade de forma inversa: ($d = x - x'$).

$$Z = \frac{Tf}{x' - x} \quad (1)$$

$$X = Z * \frac{(x' - x_0')}{f} \quad (2)$$

$$Y = Z * \frac{(y' - y_0')}{f} \quad (3)$$

2.5 Geometria Epipolar

A Geometria Epipolar é a relação geométrica da intersecção entre os planos das imagens com o conjunto dos planos epipolares, tendo a reta da linha de base como intersecção entre eles (AIRES, 2010). Com base na Figura 5, sendo o ponto P um ponto qualquer no espaço, e os centros ópticos definidos como O e O' . Toma-se como plano epipolar, a intersecção dos pontos OP e PO' .

Quando se tem duas câmeras situadas a uma determinada distância e ambas detectam um mesmo ponto P no espaço, surge o problema da correspondência entre esses pontos. Pois, para gerar a estereoscopia necessita-se que o mesmo ponto seja encontrado em ambas as imagens. A Geometria Epipolar veio para facilitar a busca desses pontos.

Sendo O o eixo óptico da câmera esquerda, o e chamado de epípolo, correspondente a projeção do eixo O' sobre o plano da imagem esquerda. De forma inversa, o epípolo e' é a projeção do eixo O sobre o plano da imagem direita. Sabendo que e e e' são os pontos em que a reta, chamada de linha de base, intercepta os planos quando liga os dois eixos O e O' , consegue-se gerar uma reta l e outra l' denominadas retas epipolares, que surgem a partir da intersecção do plano epipolar com os planos das imagens (KARLSTROEM, 2007).

Para cada ponto projetado em um plano de imagem, tem-se uma reta epipolar correspondente no plano inverso (AIRES, 2010). Isso facilita a busca por pontos correspondentes, pois, sabendo-se as coordenadas de um ponto no plano de projeção, busca-se somente pela linha horizontal do plano oposto o valor correspondente ao ponto homólogo sobre sua linha epipolar. Essa propriedade é conhecida como restrição epipolar e pode ser conseguida simplesmente com os parâmetros intrínsecos e extrínsecos da câmera (OLIVEIRA, 2006). Nos trabalhos de (ARAÚJO, 2010) e (AIRES, 2010) encontram-se mais detalhes sobre Geometria Epipolar.

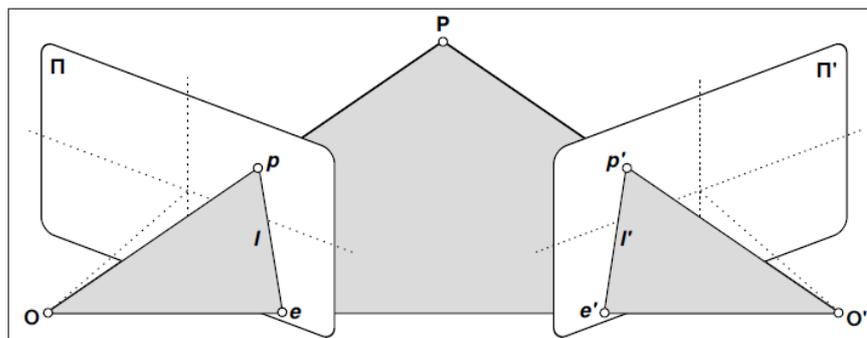


Figura 5: Representação da Geometria Epipolar.

Fonte: (ARAÚJO, 2010).

2.6 Calibração de Câmeras

A calibração de câmeras pode ser definida como o processo pelo qual, busca-se encontrar os parâmetros intrínsecos e extrínsecos de configuração de cada câmera.

2.6.1 Parâmetros intrínsecos

São valores próprios de fabricação de cada câmera, basicamente se resumem a distância focal, a dimensões dos pixels, a distorção das lentes e a distância de base. Alguns desses parâmetros podem ser organizados na matriz A de parâmetros intrínsecos ou matriz de calibração abaixo (OPENCV, 2012a):

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & -f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Sendo $f_x = Kf$ e $-f_y = Kf$, onde K corresponde ao fator de conversão de unidade métrica para pixel (MEDEIROS, 2008) e f o valor da distância focal. Esse fator K é utilizado em situações em que a largura do pixel se difere do valor de altura, necessitando dessa constante. O c_x e c_y são as coordenadas do centro da imagem, ou seja, as coordenadas do plano da imagem onde o eixo óptico o intercepta.

Por padrão, uma vez descoberto esses valores, eles não se alteram. No entanto, caso seja dado zoom ou modificado alguma característica física da câmera, é necessário novamente atualizar todos esses valores.

2.6.2 Parâmetros Extrínsecos

São valores que indicam a localização da câmera no mundo, levando-se em conta movimentos de rotação e translação. São compostos por três matrizes de rotação e um vetor de translação.

$$\text{Matriz de rotação do eixo } x: \quad \mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

$$\text{Matriz de rotação do eixo } y: \quad \mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}$$

Matriz de rotação do eixo z :
$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

O produto dessas matrizes forma uma matriz única de rotação $\mathbf{R}_{\alpha\beta\theta}$, levando-se em conta que as rotações são realizadas em sequências acumulativas, ou seja, quando se necessita fazer uma rotação simultânea nos eixos x e y , aplica-se primeiramente uma rotação a x e depois aproveitando os valores resultantes da rotação de x , aplica-se outra rotação sobre y . Isso também vale com relação ao eixo z e é definida pela rotação dos ângulos de Euler (GALO & TOZZI, 2001).

Produto das matrizes de rotação:
$$\mathbf{R}_{\alpha\beta\theta} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

O vetor de translação contém as coordenadas de deslocamento x , y e z da câmera no mundo.

Vetor de translação:
$$\mathbf{T} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

De posse das matrizes de rotação e do vetor de translação, consegue-se mapear as coordenadas de um ponto no mundo, para as coordenadas de câmera, a partir de sua projeção no plano da imagem. A Figura 6 mostra o mapeamento das coordenadas do ponto \mathbf{M} , para as coordenadas de projeção \mathbf{m} .

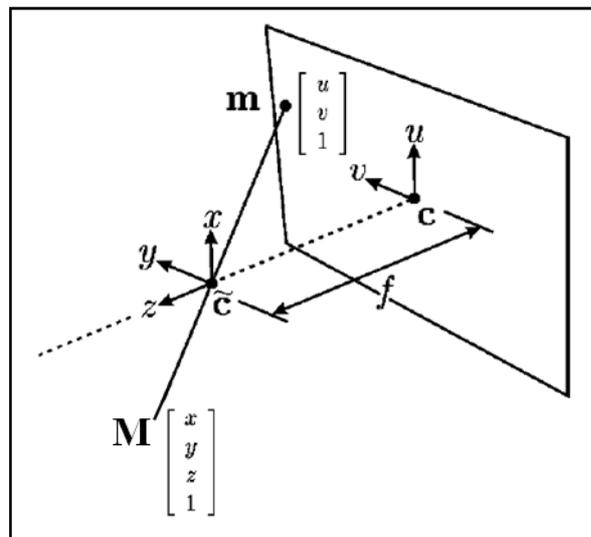


Figura 6: Projeção de câmera em relação a um ponto no mundo.

Fonte: (GUERRA, 2004).

A projeção de um ponto qualquer no mundo para coordenadas de um plano de imagem pode ser expressa pela Equação 5 (OPENCV, 2012a):

$$sm' = A[R | T]M' \quad (5)$$

- Onde A é a matriz de calibração
- R é a matriz de rotação
- T é o vetor de translação das coordenadas do ponto M
- M' representa as coordenadas do ponto M
- $[R | T]$ é a matriz conjunta de rotação-translação

Para um maior aprofundamento, nos seguintes trabalhos (AIRES, 2010), (OPENCV, 2012a), (GALO & TOZZI, 2001) e (MEDEIROS, 2008), encontram-se informações mais detalhadas sobre cada uma das matrizes que compõe a Equação 5.

2.6.3 Técnicas de calibração

Para obter todos os parâmetros intrínsecos e extrínsecos, deve-se usar alguma técnica de calibração: a calibração baseada em objetos de referência ou a auto-calibração.

A calibração baseada em objetos de referência possui duas metodologias de abordagem mostradas nos seguintes trabalhos (AIRES, 2010) e (GUERRA, 2004), sendo que ambas utilizam um gabarito como referência. Esse gabarito é formado por um conjunto de linhas verticais e horizontais perpendiculares, em forma de malha, como demonstrado na Figura 7. Suas dimensões são previamente conhecidas servindo de entrada para o algoritmo de calibração, permitindo assim, estabelecer um referencial de câmera em relação aos outros objetos do mundo. Entre esses valores estão o número de quadrados que possuem o gabarito e a altura e largura de cada quadrado.

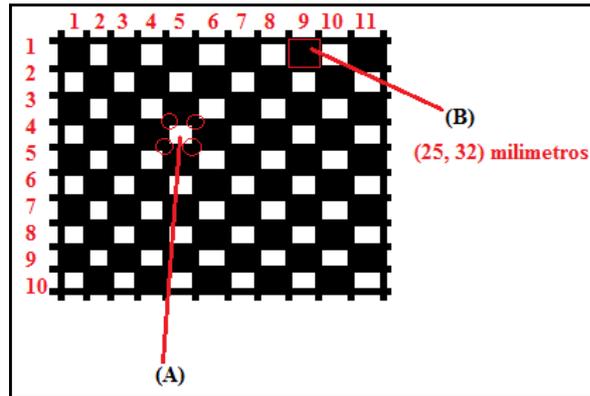


Figura 7: Gabarito de calibração. (A) Intersecção das linhas verticais e horizontais indicando os vértices dos quadrados, (B) Altura e largura dos quadrados.

2.6.3.1 Calibração baseada em objetos 3D ou método de Tsai.

O método de Tsai permite descobrir todos os parâmetros intrínsecos e extrínsecos de cada câmera, apenas utilizando como referência uma foto do objeto 3D que fica situado na frente do gabarito representado pela Figura 8 (AIRES, 2010).

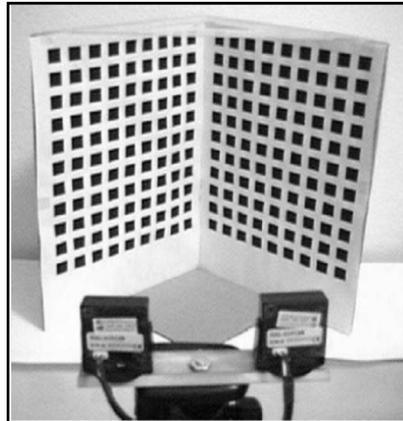


Figura 8: Método de calibração baseado em objetos de referência 3D.

Fonte: (FRANÇA, 2003).

2.6.3.2 Calibração baseada em imagens 2D ou método de Zhang

O método de Zhang demonstrado na Figura 9, funciona de forma semelhante ao método anterior. No entanto, enquanto a abordagem anterior baseava-se em objetos 3D, o método de Zhang, detalhado no trabalho de (AZEVEDO; TAVARES; VAZ, 2008) e utilizado na etapa de calibração do presente trabalho, utiliza-se de uma imagem em forma de malha (por padrão um tabuleiro de xadrez), com suas dimensões previamente conhecidas.

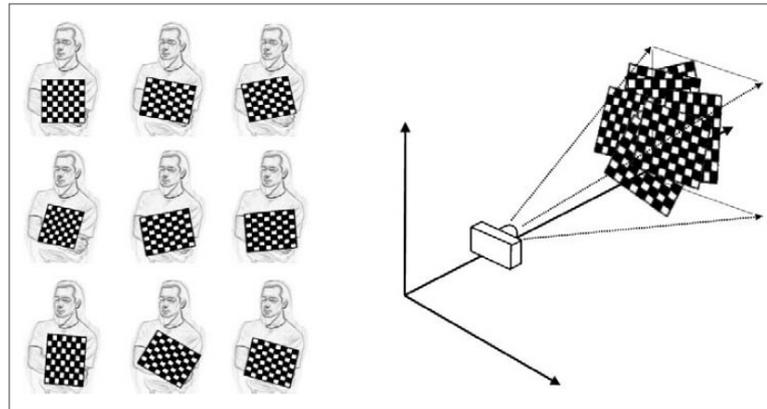


Figura 9: Método de calibração de Zhang.

Fonte: (BRADSKI & KAEHLER, 2008).

O contraste entre os vértices de cada quadrado permite que eles sejam encontrados facilmente e assim estabelecer seu referencial em relação aos demais pontos do tabuleiro (AZEVEDO; TAVARES; VAZ, 2008). Ao movimentar o tabuleiro em frente à câmera ou de forma inversa; movimentar a câmera em frente ao tabuleiro consegue-se estimar a matriz de rotação e o vetor de translação, assim como os parâmetros intrínsecos da câmera. Quanto mais imagens do tabuleiro forem tiradas de diferentes pontos de vista, mais precisa será a calibração final.

O problema desse método é ser menos eficaz que o método de Tsai quando avaliado a quantidade e a qualidade dos parâmetros de câmera obtidos. No entanto, sua implementação é bem mais fácil se comparado ao método baseado em objetos 3D.

2.6.3.3 Auto-calibração

A auto-calibração diferentemente dos métodos de Tsai e Zhang não utiliza nenhum marcador como referência, apenas com os movimentos e operação normal da câmera consegue-se extrair uma grande quantidade de dados correspondente aos parâmetros intrínsecos e extrínsecos (FRANÇA, 2003). Essa técnica veio para resolver os problemas em atividades que necessitam constantemente de mudanças em seus parâmetros de câmera para se adaptar às variações das condições do ambiente, como, por exemplo, a robótica.

2.7 Retificação de Imagens

O processo de retificação de imagens é uma forma que busca minimizar os problemas enfrentados pela correspondência de pontos homólogos em pares de imagens estéreo, utilizando as propriedades da geometria epipolar (OLIVEIRA, 2006).

Embora a definição de correspondência entre pontos seja explicada na próxima seção, basta entender que a correspondência busca encontrar as coordenadas da projeção de um mesmo ponto nos planos das imagens esquerda e direita. Dependendo do posicionamento das câmeras, as coordenadas desse ponto podem ter valores extremamente diferentes, tanto no eixo x como no eixo y . Isso torna a possibilidade de se encontrar o ponto correspondente na outra imagem equivalente a $n_x \times n_y$, sendo n_x o número de pixels na horizontal e n_y da vertical. Ou seja, para cada ponto encontrado numa imagem é necessário fazer uma busca bidimensional na outra imagem para se achar o seu correspondente homólogo.

Sabendo dessa dificuldade em varrer todas as linhas e colunas de uma imagem em busca de se encontrar seu ponto homólogo, resolveu-se utilizar a propriedade da restrição epipolar como princípio da retificação.

Aplicando a restrição epipolar, passou-se a fazer a busca pelos pontos correspondentes apenas na horizontal dos planos de imagem (NOGUEIRA & TOZZI, 1999), (OPENCV, 2012b). Isso porque, segundo as propriedades da geometria epipolar (AIRES, 2010): para cada ponto projetado num plano x , existe uma reta epipolar correspondente na outra imagem, na qual está seu ponto homólogo. Sabendo disso, conclui-se que achando a coordenada de um ponto numa imagem, basta procurar na linha horizontal, paralelamente a linha de base o ponto correspondente situado na sua reta epipolar, localizado na imagem do plano oposto. Na Figura 10 têm-se duas imagens retificadas, onde a projeção p_l do ponto P está situada na reta epipolar da projeção p_r e a projeção p_r se encontra na reta epipolar de p_l . Percebe-se que as retas epipolares estão colineares³ e que os epípolos tendem ao infinito. Isso acontece porque a reta base entre os eixos ópticos O_l e O_r tornou-se paralela aos planos das imagens.

³ Situados em um mesmo plano, ou seja, pertencentes a uma linha reta em comum.

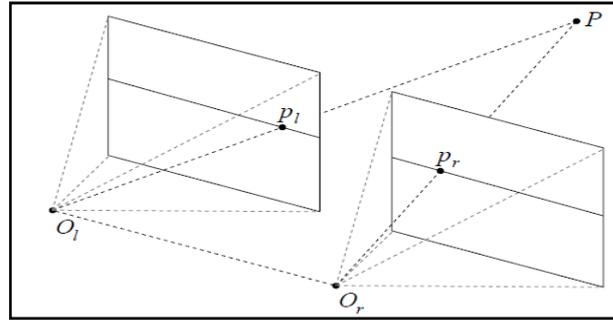


Figura 10: Retificação com base na geometria epipolar.

Fonte: (OLIVEIRA, 2006)

A retificação veio para reduzir a busca bidimensional do processo de correspondência para uma busca unidimensional através do alinhamento horizontal das duas imagens, de tal forma que um ponto x encontrado na imagem esquerda, situado na linha x_l possa ser encontrado na mesma linha x_l da imagem direita.

Geralmente as posições das câmeras nunca estão perfeitamente alinhadas, por isso quem fica responsável por fazer esse alinhamento e aplicar as propriedades da geometria epipolar são duas matrizes previamente calculadas com base nos valores obtidos na calibração. Essas matrizes são chamadas de matriz essencial e matriz fundamental (AIRES, 2010), (FRANÇA, 2003), (ALVES, 2007).

Como pode ser visto na Figura 11 a retificação permite alinhar as duas imagens de tal forma que as linhas horizontais passam exatamente no mesmo ponto em ambas as imagens. Enquanto que no mesmo par de imagens sem retificação, as linhas horizontais da imagem esquerda não coincidem com as linhas da imagem direita.

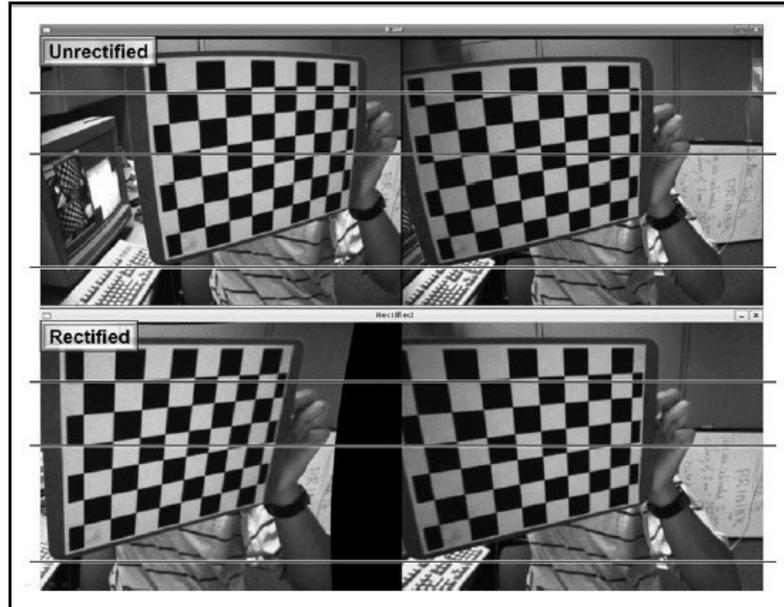


Figura 11: Diferença entre antes e depois de aplicar a retificação num par de imagens.

Fonte: (BRADSKI & KAEHLER, 2008).

2.8 Correspondência entre Pontos

A correspondência entre pontos homólogos em diferentes imagens é uma das técnicas mais pesquisada na área de visão computacional, principalmente voltada para aplicações de detecção de objetos, reconstrução 3D, mapeamento robótico e outras áreas que necessitam correlacionar pontos com características únicas entre duas ou mais imagens (SASDELLI, 2010).

Para que a correspondência entre pontos comece a ser feita, é necessário fazer a detecção de um conjunto de pontos com características, de preferência, únicas nas imagens, encontradas principalmente em cantos, retas e outras áreas com diferenças bruscas de luminosidade e textura. Essa atividade é feita pelos detectores de pontos ou detectores de características, que normalmente vêm como rotinas dos algoritmos de correspondência. Como características desejadas, os pontos encontrados devem ser invariantes a certas mudanças ocorridas em relação à imagem, como por exemplo: escala rotação, translação e mudanças de iluminação (SASDELLI, 2010).

Vale ressaltar que os detectores de características não fazem correspondência. Eles fornecem informações sobre os pontos selecionados para se realizar a correspondência. O processo de relacionar um ponto com seu correspondente é outra etapa que pode ser feita por um algoritmo de correspondência separadamente.

Compete aos algoritmos de correspondência encontrar para cada ponto numa imagem o seu correspondente nas outras imagens. Os algoritmos de correspondência que trabalham com imagens estéreo são basicamente divididos em dois grupos: globais e locais.

2.8.1 Algoritmos Globais

Os algoritmos globais utilizam o conceito de minimização de energia para extrair os valores de disparidade para cada pixel relacionando as duas imagens estéreo (JOSÉ, 2008). Sua abordagem começa analisando a estrutura de um *Disparity Space Image (DSI)* demonstrado na Figura 12(A).

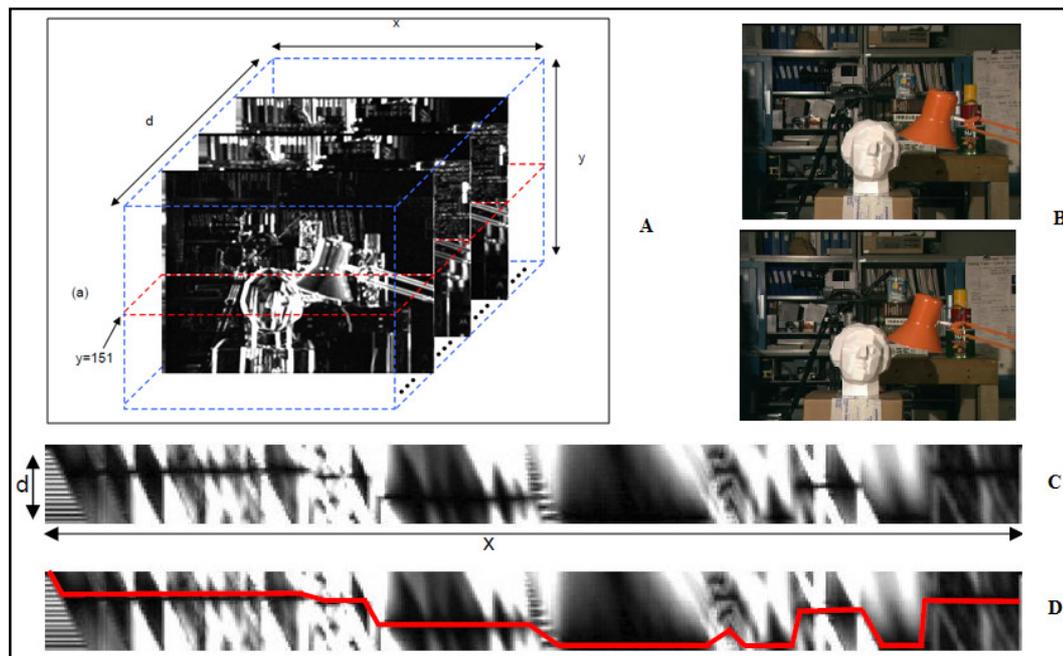


Figura 12: Representação de um DSI. Sendo (A) Os planos do DSI, (B) Imagens estéreo esquerda e direita, (C e D) Representam o corte no DSI na posição $y=151$, onde a linha vermelha indica o caminho de minimização de energia (regiões mais escuras).

Fonte: (JOSÉ, 2008).

Os planos do DSI são formados pela diferença quadrática pixel a pixel da imagem esquerda em relação à imagem direita. Sendo que o deslocamento da imagem direita em relação à esquerda começa de 0 até o valor máximo de disparidade estabelecido, representado na Figura 12(C) por d . Um dos algoritmos globais mais populares que utilizam a minimização de energia é o *Graph-Cuts* (SOUZA, 2011).

2.8.2 Algoritmos Locais

Grande parte dos algoritmos locais utiliza o conceito da soma das diferenças absolutas (SAD – *Sunof Absolute Differences*). Nessa técnica a disparidade é calculada subtraindo os valores das cores dos pixels da imagem esquerda em relação aos valores da imagem direita. No entanto, o cálculo da correspondência de apenas um pixel torna-se impreciso em termos de ambiguidades. Para isso utilizam-se janelas que limitam a área de busca da correspondência. Tomando-se como base uma janela fixa na imagem esquerda representada pela Figura 13(A) e outra móvel na imagem direita representada pela Figura 13(B), somam-se os valores da diferença em módulo de todos os pixels da janela esquerda em relação à janela direita. Em seguida move-se a janela da imagem direita até atingir o valor máximo de disparidade estipulado.

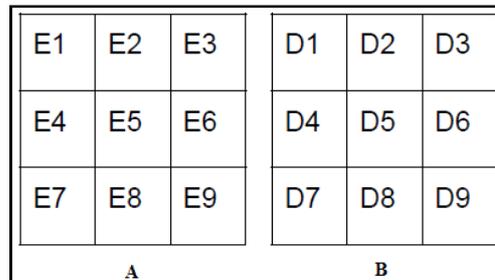


Figura 13: Janelas de Pixels. (A) Janela esquerda estática, (B) Janela direita móvel.

Fonte: (JOSÉ, 2008).

Para cada vez que a janela da direita se mover cria-se um valor de disparidade para ela. O menor valor encontrado entre as diferentes posições da janela direita é considerado a melhor disparidade em relação à janela esquerda (JOSÉ, 2008). Abaixo segue o cálculo para se achar a disparidade para uma janela, levando em consideração que os valores dos pixels da Figura 13 estejam em tons de cinza.

$$Disp1 = (E1 - D1)$$

$$Disp2 = (E2 - D2)$$

... ..

$$Disp9 = (E9 - D9)$$

$$Disp.JanelaPosição1 = Disp1 + Disp2 + (...) + Disp9$$

O valor “*Disp.JanelaPosição1*” representa a disparidade total da janela na primeira posição. Caso a janela direita se desloque 7 pixels para a direita, os valores de disparidades para cada deslocamento da janela formarão uma fila na qual, será escolhido o menor valor como disparidade da janela esquerda.

$$Disp.Posição1 = 10$$

$$Disp.Posição2 = 8$$

... ..

$$Disp.Posição7 = 15$$

$$Disp.Escolhida = Disp.Posição2 = 8$$

Além dos algoritmos locais como o Block-Matching (SOUZA, 2011), o conceito da soma das diferenças absolutas também está sendo utilizado em alguns algoritmos globais.

2.9 Mapa de Disparidade

Como o próprio nome diz, disparidade está relacionada à diferença, e no contexto da geometria estéreo representa a distância entre um mesmo ponto projetado na imagem esquerda e na imagem direita. Essa distância é medida em pixels e seu valor é inversamente proporcional à profundidade relativa ao ponto no mundo (OLIVEIRA, 2006). Essa afirmação pode ser compreendida com base na Figura 14.

Sabendo que o ponto ***P1*** possui as coordenadas de projeção nas imagens esquerda e direita respectivamente a ***P1_{xl}*** e ***P1_{xr}***; e o ponto ***P2*** as coordenadas ***P2_{xl}*** e ***P2_{xr}***, então a diferença da coordenada de projeção do ponto ***P1*** na imagem esquerda em relação à imagem direita é de 135 pixels e a do ponto ***P2*** é 279. Essas são as medidas de disparidade. Esses valores obtidos provam que à medida que um ponto ***P*** no espaço se distancia em relação ao plano da câmera, sua disparidade diminui, chegando ao ponto de ser próximo de zero ou tendendo ao infinito. Com os valores de disparidade tão próximos de zero, a distância entre os eixos ópticos torna-se tão mínima que se perde todas as propriedades da visão estéreo relativa a esse ponto.

Outra característica importante sobre o cálculo da disparidade é que ele é feito somente com os pares de pontos idênticos, ou seja, somente os pontos que são visíveis por ambas as câmeras. A Figura 14 mostra a representação de um triângulo amarelo que limita a região máxima onde os pontos que estão no plano da imagem esquerda, também estão no

plano da imagem direita. Os pontos que ficam fora dessa região são chamados de pontos oclusos, pois são projetados apenas em um dos planos de imagem.

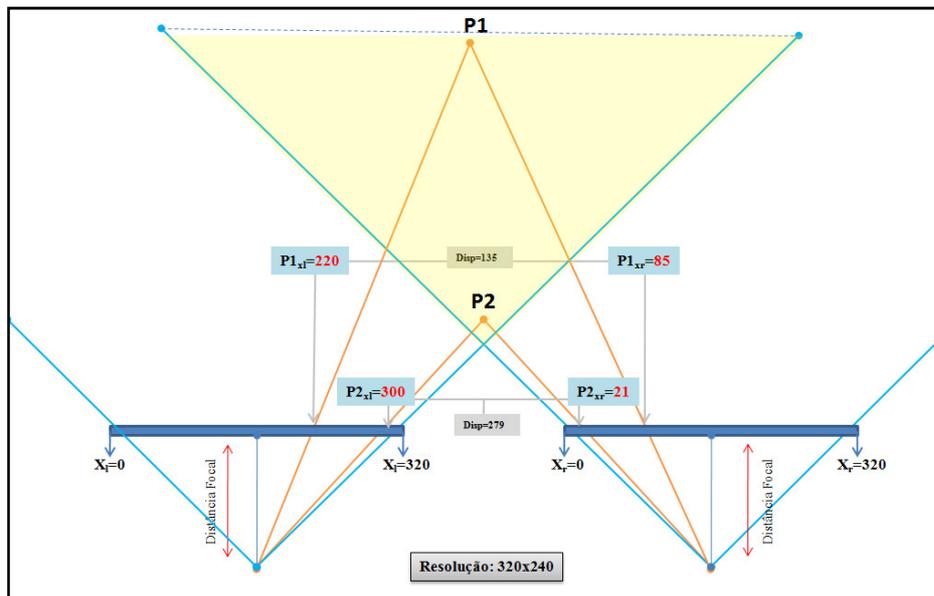


Figura 14: Cálculo da disparidade relacionando a região dos pontos oclusos.

O cálculo da disparidade se resume à equação 6, onde d equivale ao valor da disparidade; x_l a coordenada do pixel no plano da imagem esquerda e x_r a coordenada no plano da imagem direita (NOGUEIRA & TOZZI, 1999).

$$d = x_l - x_r \quad (6)$$

Tendo o valor de disparidade em mãos é só aplicá-lo na Equação 1 vista nos conceitos de geometria estéreo, para achar a profundidade Z da câmera em relação a um ponto no espaço (OPENCV, 2012a), (SOUZA, 2011), (NOGUEIRA & TOZZI, 1999). As demais coordenadas: X e Y são obtidas após descoberta a profundidade correspondente ao eixo Z , sendo representadas pelas equações 2 e 3 (SOUZA, 2011).

Um mapa de disparidades é formado por todos os valores de disparidade encontrados pela diferença de coordenadas dos pontos obtidos pelo processo de correspondência. Fazendo uma distribuição percentual dos valores de disparidade sobre o intervalo 0 a 255, consegue-se gerar uma imagem em tons de cinza que representa os níveis de profundidade da cena. Quanto mais próximo do ponto a câmera estiver, mais branco será o pixel e mais preto quando a câmera estiver mais distante. A Figura 15 mostra o mapa de disparidade gerado por um par de imagens estéreo.

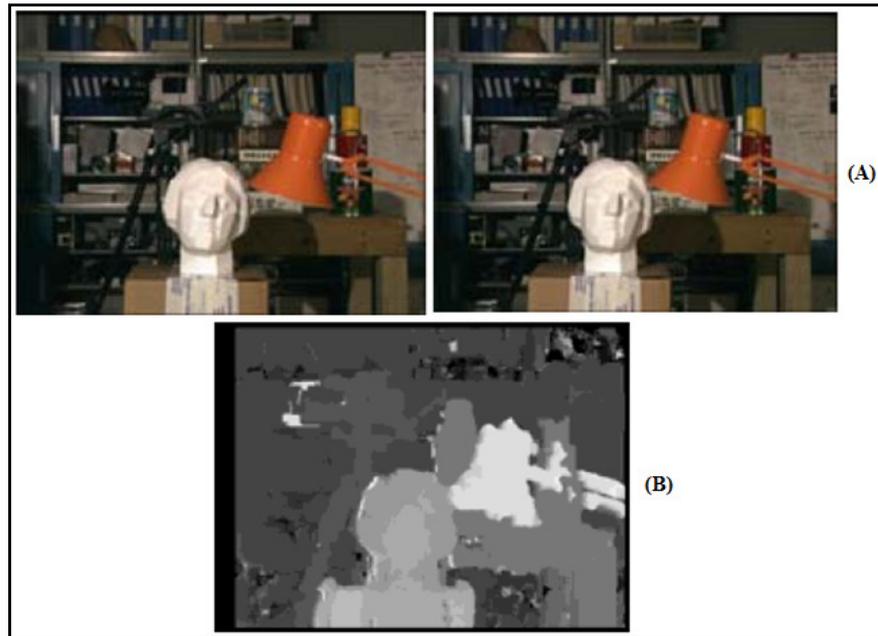


Figura 15: Mapa de disparidades. (A) Par de imagens estéreo, (B) Imagem em tons de cinza representando o mapa de disparidades obtido com o par imagens acima.

Fonte: (JOSÉ, 2008).

2.10 Reconstrução 3D a partir de Visão Computacional

A aquisição de informações tridimensionais de objetos reais está a cada dia ganhando mais espaço nas mais diversas aplicações. Setores como paleontologia utilizam a reconstrução 3D para arquivamento da estrutura óssea dos fósseis, assim como a medicina utiliza-a na criação de próteses e exames de ressonância magnética. Para cada tipo de aplicação pode-se aplicar uma técnica diferente de reconstrução.

Entre as técnicas de aquisição de informações tridimensionais predominantes destacam-se o grupo das técnicas de aquisição passivas sem contato. Essas técnicas se caracterizam por utilizarem a luz ambiente refletida nos objetos para conseguir extrair informações tridimensionais. As câmeras de vídeo são os principais tipos de sensores responsáveis pela captura da reflexão da luz provida pelos objetos. Muitos métodos de reconstrução foram criados para modelagem 3D de superfícies utilizando apenas uma ou mais câmeras de vídeo. Entre os principais estão: *shape from focus*, *shape from shading*, *structure from motion*, *shape from silhouette* e *structure from stereo* (ROCHA & GONÇALVES, 2012), (ALBURNIO, 2012).

2.10.1 Shape From Focus

Esse método consegue extrair informações tridimensionais da geometria dos objetos utilizando a seleção dos pontos mais bem focados, encontrados através de uma sequência de imagens tiradas de um mesmo ponto de vista, mudando apenas os valores de foco (AYDIN & AKGUL, 2008) da câmera. Assim, a profundidade é calculada para cada pixel, levando em consideração a distância focal da lente e a profundidade da câmera em relação ao objeto. Entre os vários trabalhos que utilizam essa técnica estão (AYDIN & AKGUL, 2008) e (NIEDERÖST; NIEDERÖST; SKUCKA, 2003).

2.10.2 Shape From Shading

O *Shape From Shading* é um método de reconstrução tridimensional que utiliza os valores de variância na iluminação presente numa imagem para extrair dados relativos à superfície dos objetos. Sabendo previamente os dados correspondentes a localização da fonte de luz e aplicando os conceitos da geometria de projeção e iluminação, torna-se possível descobrir a trajetória de cada raio e a região da superfície que foi atingida (VOGEL et al, 2009).

2.10.3 Structure From Motion

Esse método consiste em encontrar a estrutura tridimensional dos objetos utilizando como base os seus movimentos em relação à câmera ou os movimentos da câmera em relação à cena (ROCHA & GONÇALVES, 2012). Através de uma sequência de imagens obtidas por uma ou mais câmeras, tiradas seguindo uma ordem temporal, realiza-se uma busca a partir da primeira imagem por pontos com características de preferência únicas que possibilite serem encontrados nas demais imagens.

2.10.4 Shape From Silhouette

É um método de reconstrução baseado nas silhuetas presentes nos objetos captadas por diferentes pontos de vista. Com a intersecção dessas silhuetas consegue-se reconstruir a estrutura do objeto. Geralmente esse método é utilizado em conjunto com outras técnicas constituindo os chamados métodos híbridos (ALBURNIO, 2012). Isso acontece porque, para

se captar as silhuetas, utilizam-se uma plataforma giratória ou necessita-se controlar diretamente o ângulo de rotação da câmera para cada imagem obtida, tornando o processo dispendioso e cansativo.

2.10.5 Structure From Stereo

O *Structure From Stereo* é um dos métodos mais populares e eficientes de reconstrução 3D. Utilizado como método de reconstrução para o sistema desenvolvido nesse trabalho, consiste em simular o sistema visual humano, através de um sistema, composto por duas câmeras posicionadas de forma colinear. De forma, que possibilite mapear as informações de profundidade presente no ambiente.

Uma derivação desse método que está sendo muito utilizada em trabalhos de vídeo 3D, realidade aumentada e reconstrução 3D se chama *Multiview-Stereo* (ROCHA & GONÇALVES, 2012). O *Multiview-Stereo*, consiste em tirar várias imagens simultâneas de diferentes pontos de vista através de várias câmeras posicionadas estrategicamente, para depois através de algum algoritmo especial relacionar os pontos encontrados em cada imagem com a orientação de todas as câmeras.

A Figura 16 mostra uma configuração de câmeras sendo utilizada pelo método *Multiview-Stereo*, juntamente com os diferentes pontos de vista de cada câmera.

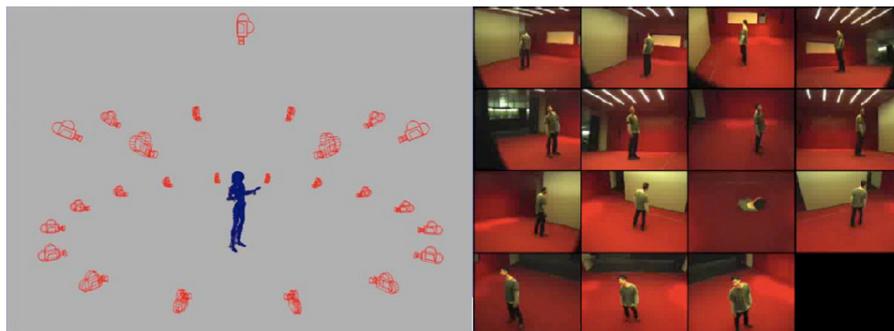


Figura 16: Método Multiview-Stereo.

Fonte: (ROCHA & GONÇALVES, 2012).

3 DESENVOLVIMENTO E IMPLEMENTAÇÃO

3.1 Metodologia Adotada no Desenvolvimento da Aplicação

O funcionamento do sistema desenvolvido segue o fluxograma da Figura 17.

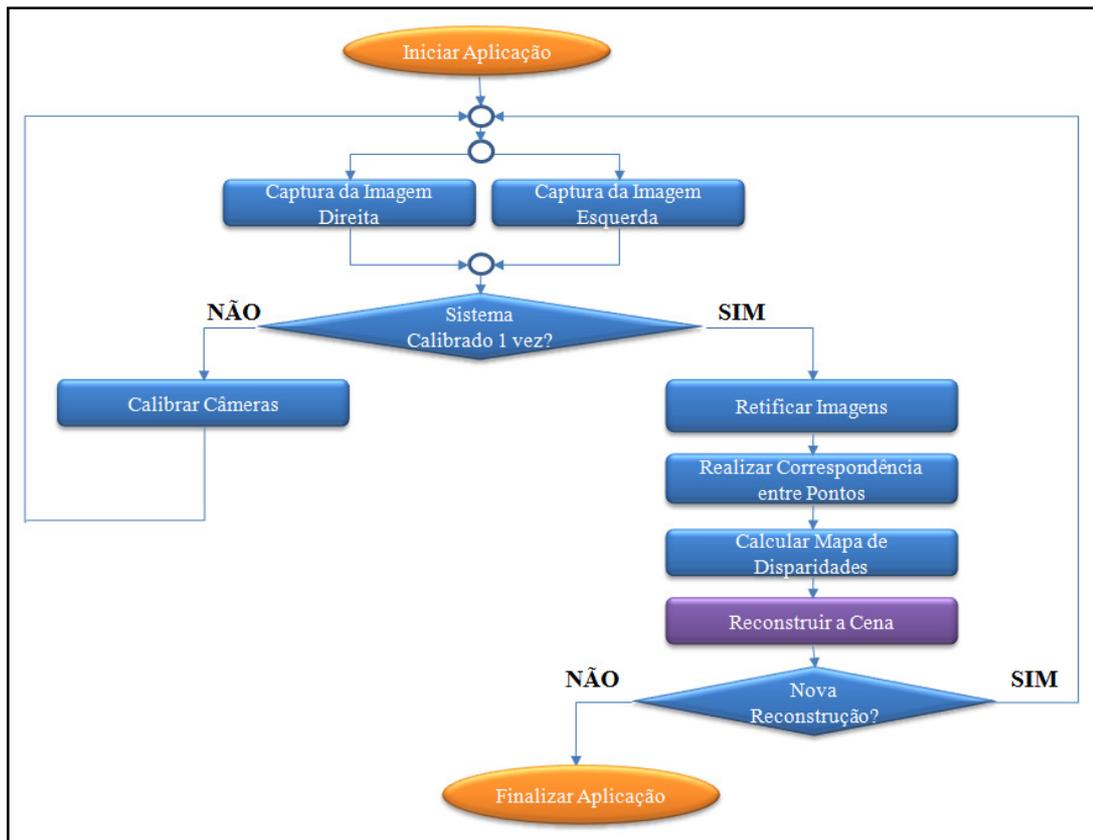


Figura 17: Fluxograma dos estágios necessários para se realizar a reconstrução de uma superfície.

Inicialmente o sistema começa capturando duas imagens simultaneamente, sendo uma para a câmera esquerda e outra para a câmera direita. No entanto, se os parâmetros intrínsecos e extrínsecos das câmeras ainda não forem conhecidos é preciso realizar a etapa de calibração, representada na Figura 17 pelo bloco “Calibrar Câmeras”.

Para realizar a calibração não foi necessário ter a preocupação com relação ao tempo, pois, como o método de calibração adotado foi o de Zhang (AZEVEDO; TAVARES; VAZ, 2008) baseado em imagens 2D e os parâmetros intrínsecos e extrínsecos da câmera não iriam se alterar, a calibração foi feita previamente, apenas uma vez.

No quadro abaixo estão alguns parâmetros intrínsecos e extrínsecos da câmera Minoru obtidos com o método de Zhang.

Coeficientes de distorção da lente esquerda				
-0.153223	0.801177	0	0	-2.82163
Coeficientes de distorção da lente direita				
-0.176281	0.7265	0	0	-3.35778
Matriz da Câmera Esquerda				
439.581	0	166.805		
0	439.581	106.484		
0	0	1		
Matriz da Câmera Direita				
439.581	0	159.044		
0	439.581	113.12		
0	0	1		
Matriz de Rotação				
0.999951	-0.000391714	-0.00987879		
0.000390736	1	-0.000100871		
0.00987882	9.70059e-05	0.999951		
Vetor de Translação				
-6.23354	0.176608	-0.227403		

Quadro 1: Parâmetros intrínsecos e extrínsecos da webcam Minoru.

Quando a calibração termina, utilizam-se os dados obtidos como entrada para a etapa de retificação, representada pelo bloco “Retificar Imagens” no fluxograma para poder aplicar as propriedades da geometria estéreo e conseqüentemente alinhar as duas imagens. A retificação é feita sempre que acontece a captura do par de imagens. As imagens retificadas são utilizadas na etapa de correspondência representada na Figura 17 pelo bloco “Realizar Correspondência entre Pontos”.

Para desenvolver essa aplicação, inicialmente foi necessário conhecer diversos algoritmos de estimação de correspondência em visão estéreo e compará-los quanto ao custo computacional. Isso porque, como a aplicação tem por objetivo reconstruir superfícies em tempo real, o tempo de processamento foi considerado essencial para a escolha dos algoritmos a serem utilizados.

Para realizar a correspondência entre pontos, foram testados os algoritmos: *Block-Matching* (SOUZA, 2011) e *Graph-Cuts* (SOUZA, 2011). Ambos encontrados nas rotinas da biblioteca OpenCV (OPENCV, 2012b).

Esses dois algoritmos englobam as duas etapas de “Realizar Correspondência entre Pontos” e “Calcular Mapa de Disparidades” presente no fluxograma da Figura 17. Como a disparidade é calculada fazendo a diferença entre as coordenadas de um mesmo ponto na imagem esquerda e direita, esses algoritmos fazem a correspondência e disponibilizam o mapa de disparidades resultante.

O *Graph-Cuts* (GC) consegue gerar mapas de disparidades mais ricos em detalhes, com base numa boa correspondência de pontos. No entanto, o preço desse mapa é um alto custo computacional, em torno de 40 a 60 segundos para cada par de imagens com resolução

de 320 por 240 pixels. Com esse tempo gasto somente para realizar a correspondência e calcular o mapa de disparidades, torna o *Graph-Cuts* (sem nenhum tipo de otimização) uma alternativa inviável para se trabalhar com aplicações em tempo real que utilize um computador de baixa capacidade de processamento.

O algoritmo *Block-Matching (BM)* é bem rápido, para imagens com resoluções de 320 por 240 (tamanho adotado para essa aplicação), obtendo um tempo entre 140 a 280 milissegundos. Um tempo razoável para algumas aplicações de tempo real. O problema fica com a baixa qualidade de sua técnica de correspondência de pontos quando se trabalha com câmeras de baixo custo e propriedades intrínsecas e extrínsecas desfavoráveis para se aplicar a geometria estéreo.

Com o dilema de ter bons algoritmos de correspondência que geram mapas de qualidade, mas inviáveis para serem utilizados em tempo real e algoritmos rápidos, mas com mapas de disparidades insatisfatórios, optou-se em utilizar a técnica de correspondência estéreo *ELAS – Efficient Larger-scale Stereo*, desenvolvida por (GEIGER; ROSER; URTASUN, 2010) e implementada por Bob Motram para obtenção do mapa de disparidades. Mesmo a técnica sendo desenvolvida para imagens em alta resolução, os resultados são bem satisfatórios, principalmente pelo tempo necessário para se calcular o mapa de disparidades. Para imagens com resolução de 320 por 240 a implementação de Bob Motram da técnica desenvolvida por (GEIGER; ROSER; URTASUN, 2010), conseguiu gerar mapas de disparidades no intervalo de 254 a 333 milissegundos. Essa capacidade de calcular um mapa de disparidades denso de forma muito rápida foi um dos fatores que contribuíram para adoção dessa técnica nesse trabalho. No entanto, foram os valores de disparidade que confirmaram a qualidade da técnica utilizada.

Para se fazer tanta correspondência de pontos correta e em tão pouco tempo, o *ELAS* realiza uma correspondência prévia para detectar pontos com mais características distinguíveis. Em seguida aplica-se a esses pontos o método de triangulação 2D de Delaunay, permitindo limitar a área de busca para se realizar a correspondência e conseqüentemente reduzir as ambigüidades em relação aos pontos restantes que ficam fora do critério da triangulação de Delaunay (GEIGER; ROSER; URTASUN, 2010).

A última etapa do fluxograma da Figura 17 definida por “Reconstruir a Cena” consiste em pegar o valor de profundidade de cada ponto obtido pelo cálculo da geometria estéreo e juntá-los para formar uma nuvem de pontos.

Uma nuvem de pontos consiste matematicamente de uma estrutura de dados homogênea composta por vários pontos tridimensionais, aos quais são atribuídos os valores

das cores dos pixels da imagem esquerda. Visualmente, a nuvem de pontos representa a projeção dos pontos do mundo detectados por ambas as câmeras em um plano 3D, permitindo auferi-los diferentes valores de profundidade.

As reconstruções baseadas em nuvens de pontos permitem configurar a qualidade da reconstrução de acordo com a quantidade de pontos que é utilizada para representar uma superfície 3D. Para se realizar a reconstrução nesse trabalho optou-se por simplesmente projetar os valores de profundidade de cada pixel do mapa de disparidade em um espaço virtual formando uma nuvem de pontos.

3.2 Escopo da Aplicação

Para desenvolver uma aplicação de reconstrução de estruturas 3D, que usa câmeras estéreo como sensor de captura das informações tridimensionais, é preciso compreender os pontos abordados nos capítulos anteriores. Isso porque o processo de reconstrução depende de todas as outras fases abordadas anteriormente.

Nesse trabalho optou-se por utilizar como sensor tridimensional, uma câmera estéreo chamada Minoru (MINORU, 2012) representada na Figura 18. Essa decisão foi tomada pelo princípio de se desenvolver uma aplicação de reconstrução 3D de código aberto e com poucos recursos financeiros, que possibilitasse sua utilização por usuários domésticos que trabalham com modelagem e para fins acadêmicos onde se dispõe de poucos recursos para compra de sensores caros, como o *laser*.



Figura 18: Webcam 3D Minoru.

Fonte: (MINORU, 2012).

Seque abaixo os equipamentos de hardware e software utilizados para desenvolvimento da aplicação e suas especificações:

Hardware		Software	
<i>WebCam 3D Minoru</i>	<i>30 frames por segundo</i> Resolução máxima: <i>640 x 480</i> Censor: <i>CMOS</i>	Sistema Operacional: <i>Linux</i>	Distribuição: <i>Ubuntu 11.04</i>
Computador: <i>HP Pavilion DM 1 Notebook PC</i>	Processador: <i>Dual Core AMD E-350 ~1.6GHz</i> Memória RAM: <i>2 Giga bytes</i> Placa de Video: <i>AMD Radion HD 6310 Graphics</i>	Biblioteca: OpenCV (<i>Open Source Computer Vision</i>)	<i>Versão 2.3.1</i>
		v4l2stereo	<i>Versão 1.1</i>
		Biblioteca PCL (<i>Point Cloud Library</i>)	<i>Versão 1.5.1</i>

Quadro 2: Equipamentos utilizados para desenvolvimento da aplicação.

3.2.1 Biblioteca OpenCV (*Open Source Computer Vision*)

A biblioteca OpenCV (BRADSKI & KAEHLER, 2008) foi desenvolvida inicialmente pela equipe da Intel e que posteriormente veio a liberá-la sobre a licença *BSD*. Com a liberação, muitos grupos de desenvolvedores vêm contribuindo para a melhoria da biblioteca. Hoje a OpenCV conta com mais de 2500 algoritmos otimizados e uma vasta quantidade de fóruns e desenvolvedores associados por todo o mundo. Todos os passos necessários para sua instalação estão no site (OPENCV, 2012b).

Escrita em C e otimizada em C++, a OpenCV é composta por diversos módulos funcionais de manipulação de imagens, sendo reconhecida como uma das melhores bibliotecas para se trabalhar com visão computacional e processamento de imagens.

A OpenCV foi utilizada de forma predominante no desenvolvimento desse trabalho, principalmente nas etapas de calibração e retificação do par de imagens estéreo. Além disso, aproveitando sua grande quantidade de exemplos de algoritmos disponíveis, foram avaliados

os métodos de correspondência *Block-Matching* (SOUZA, 2011) e *Graph-Cuts* (SOUZA, 2011) com relação ao custo computacional e a qualidade da correspondência.

3.2.2 Biblioteca PCL (*Point Cloud Library*)

Escrita em C++ e sobre a licença *BSD*, a *PCL* (PCL, 2012) é uma biblioteca voltada especialmente para se trabalhar com nuvens de pontos, ou seja, pontos no espaço *2D* ou *3D*. Composta por vários módulos: reconstrução de superfícies, segmentação, filtros, detectores de características e visualização, a *PCL* também possibilita a entrada de dados por dispositivos externos como o *PrimerSensor*⁴ e o *Kinect* (bastante utilizado nas aplicações atuais).

Desenvolvida e mantida por um grande número de engenheiros espalhados por várias partes do mundo, a *PCL* está se tornando uma das melhores e mais completa biblioteca para reconstrução de superfícies. Conta com o apoio financeiro de inúmeras empresas do mercado, entre elas a Google e a NVIDIA. O passo a passo de sua instalação está disponível no site da PCL (PCL, 2012).

Nesse trabalho a PCL foi de extrema importância para se implementar a etapa de visualização da nuvem de pontos. Oferecendo um ambiente de visualização com suporte a interação em tempo real, essa biblioteca permite atualizar a nuvem de pontos à medida que eles são disponibilizados com suas respectivas coordenadas tridimensionais geradas pelo mapa de disparidades.

3.2.3 Software v4l2stereo

Pertencente ao projeto *libv4l2cam* (LIBV4L2CAM, 2012), o *v4l2stereo* é uma aplicação desenvolvida por Bob Mottram especialmente para a câmera estéreo *Minoru 3D* (MINORU, 2012) e que utiliza a biblioteca *LIBELAS* (LIBELAS, 2012).

O *libv4l2cam* é um projeto dedicado à implementação de uma interface para ambiente Linux, responsável por abstrair toda a complexidade por trás da API *V4L2* (V4L2spec, 2012), em aplicações na qual os desenvolvedores precisam acessar parâmetros de baixo nível das câmeras sem necessidade de ter que aprender a API *V4L2*.

A aplicação *v4l2stereo* foi utilizada nesse trabalho por implementar a técnica desenvolvida por (GEIGER; ROSER; URTASUN, 2010) de correspondência densa de pixels

⁴Sensores externos, que podem ser acoplados a robôs e utilizados em automação residencial.

em imagens estéreo de alta resolução o *ELAS – Efficient LArger-scale Stereo*. Essa técnica utiliza a triangulação de Delaunay2D para relacionar pontos de características altamente marcantes, chamados de pontos de apoio, a fim de obter as disparidades por interpolação de uma função linear. Segundo (GEIGER; ROSER; URTASUN, 2010) as imagens em alta resolução podem gerar muita ambiguidade, mas em contra partida, oferecem a possibilidade de fácil detecção de pontos com características altamente distinguíveis. Para garantir a consistência das correspondências, os pontos são relacionados tanto da esquerda para a direita como da direita para a esquerda. Como resultado, essa técnica consegue gerar os mapas de disparidades tanto da imagem esquerda como da imagem direita, tornando possível gerar mapas muito bons e com pouco tempo de processamento.

3.2.4 Classe Stereo

Para auxiliar na rapidez do desenvolvimento da aplicação foi utilizado a classe de dados *Stereo* desenvolvida por (SOUZA, 2011) para trabalhar com visão estéreo.

Utilizada como espinha dorsal desse sistema, essa classe oferece métodos para calibração de câmeras, retificação de imagens, captura e visualização das imagens em tempo real, além de oferecer toda a parte de alocação de memória para as variáveis que irão armazenar os valores processados na etapa de calibração, retificação e visualização das imagens.

4 EXPERIMENTOS E ANÁLISES

Para verificar se o software atendeu aos requisitos de tempo real e o processo de reconstrução atingiu o resultado esperado, foram realizados alguns testes para verificar separadamente e em conjunto, o tempo de processamento de todas as partes do sistema, assim como a veracidade dos valores de disparidade.

4.1 Tempo de Processamento

4.1.1 Mapa de disparidades

O mapa de disparidades foi calculado num tempo médio de 332.655ms, levando-se em conta que foram capturadas várias imagens em diferentes cenários, com iluminação ambiente. O quadro abaixo mostra os valores calculados da distância da câmera a um objeto de teste, assim como a proporção de erro, para diferentes distâncias.

Distância Real	Distância Calculada	Erro
70cm	45cm	25cm
47cm	36cm	11cm
34cm	29cm	5cm
28cm	25cm	3cm
23cm	21cm	2cm
19cm	17cm	2cm

Quadro 3: Dados comparativos da distância real do objeto em relação à distância calculada pela câmera.

Para uma maior precisão é aconselhável limitar a reconstrução somente a pontos com valores de Z menores que 35 cm. Outra opção seria modelar o comportamento desses valores de erro com algum método de computação numérica em uma função, que poderia ser agregada ao processo de estimação de distância.

4.1.2 Reconstrução da superfície

A etapa de reconstrução foi realizada em média a 77.39075ms. A Figura 19 mostra vários pontos de vista da reconstrução de uma imagem.

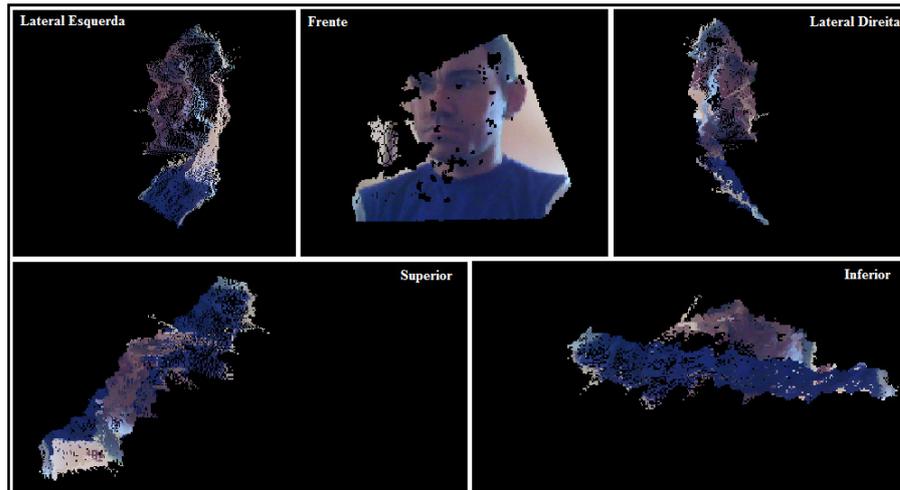


Figura 19: Diferentes pontos de vista de uma reconstrução.

Como pode ser observado na Figura 19 e nas figuras seguintes, quando se altera o ponto de vista da nuvem de pontos, ocasiona uma distorção na reconstrução. Isso acontece porque, como se trata de uma nuvem de pontos sem interligação, se torna possível ver pontos que estão em outros planos que não seria possível vê-los caso todos eles fossem unidos por algum método de triangulação. A distância real entre a câmera e a imagem da Figura 19 foi de aproximadamente 70 cm.

Na Figura 20 tem-se a reconstrução de uma caixa situada a aproximadamente 65 cm da câmera.

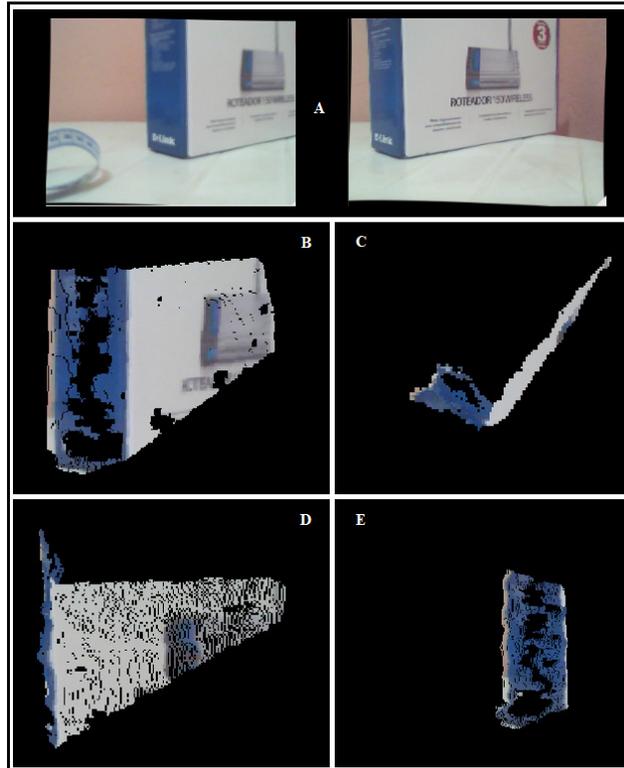


Figura 20: Reconstrução da superfície de uma caixa. (A) Imagem esquerda e direita retificada, (B) Vista com a mesma pose das imagens capturadas, (C) Vista superior, (D) Vista da lateral direita da reconstrução e (E) Vista da lateral esquerda.

A Figura 21 mostra, através de vários pontos de vista, a reconstrução de quatro objetos posicionados em diferentes distâncias.

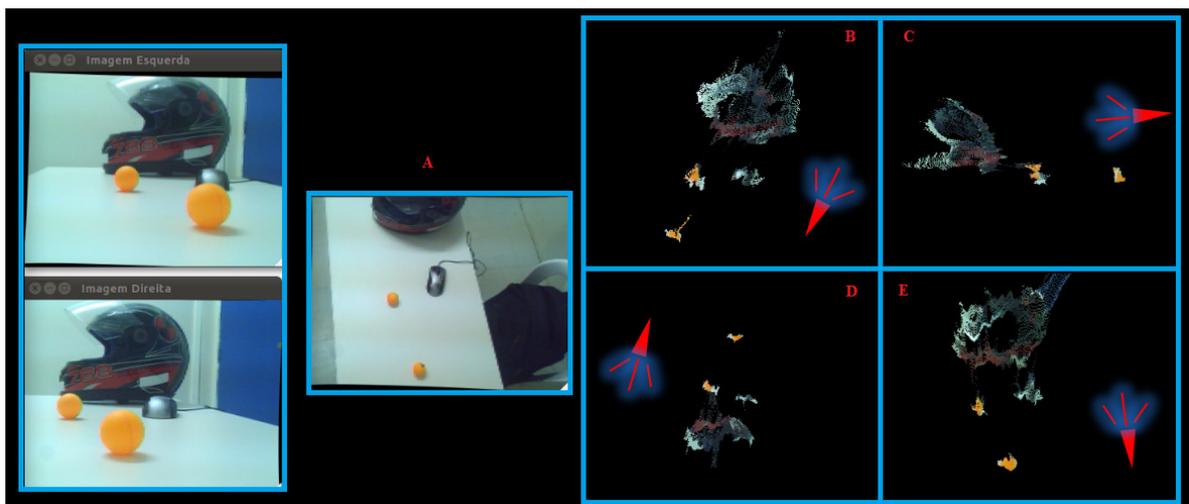


Figura 21: Reconstrução realizada com distância em torno de 40 centímetros da câmera. (A) Imagem superior dos objetos, (B) Vista superior da reconstrução, (C) Vista da lateral esquerda da reconstrução, (D) Vista superior rotacionada e (E) Vista superior.

Assim como na Figura 21, a Figura 22 mostra através de diferentes perspectivas a reconstrução dos diferentes níveis de profundidade equivalente à distância entre as partes de um objeto.

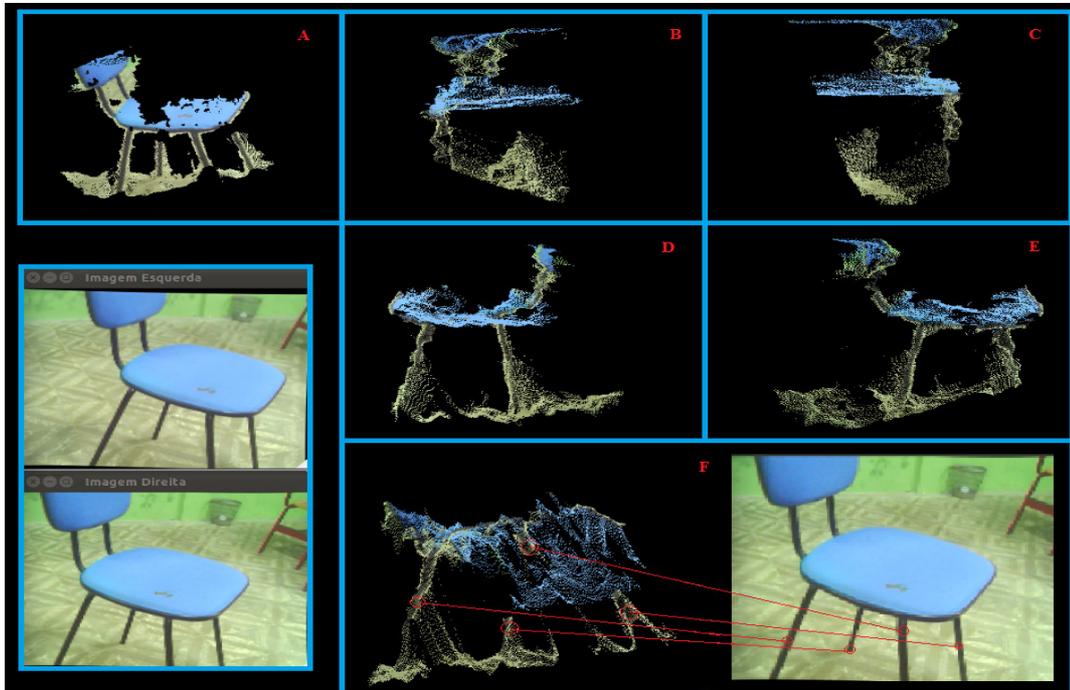


Figura 22: Reconstrução realizada com distância aproximada de 80 centímetros. (A) Vista com a mesma pose das imagens capturadas. (B) Vista frontal, (C) Vista por trás da reconstrução, (D) Vista da lateral esquerda, (E) Vista da lateral direita e (F) Vista inferior relacionando a parte das pernas da cadeira.

Como característica das reconstruções, observou-se que a disparidade dos pontos é calculada mesmo para objetos distantes. No entanto, à medida que a distância real entre a câmera e a superfície ultrapassa valores acima de 35 centímetros, os erros de profundidade vão aumentando proporcionalmente.

Quando o sistema está em funcionamento nota-se um pequeno atraso nas respostas quando se altera a posição da câmera. Uma das razões para que isso aconteça é devido a um pequeno atraso entre a captura das imagens pela câmera e seu processamento pelo computador (delay). Esse atraso também é ocasionado devido á utilização de uma área de armazenamento temporário (buffer) parecido com a fila de uma impressora, na qual ele armazenar uma sequência de imagens que vão sendo liberadas para visualização na ordem que entram no *buffer*. Isso causa alguns problemas, pois, no *buffer* fica constando imagens que já teriam sido modificadas no mundo, mas que ainda estariam esperando na fila para serem processadas.

Como padrão, o tamanho do *buffer* é de quatro imagens. No entanto, para diminuir o *delay*, diminui-se o tamanho do *buffer* para uma imagem, sendo o mínimo que o driver da câmera suporta.

No geral, o tempo de processamento de cada etapa da aplicação está relacionado no quadro abaixo:

Etapa	Tempo
Captura das imagens pela câmera	19.1874ms a 29.4994ms
Criação do mapa de disparidade	253.599ms a 332.655ms
Gerar reconstrução	40.0645ms a 114.717ms
Processamento total	312.8509ms a 476.8714ms

Quadro 4: Relação do tempo de processamento da aplicação.

5 CONCLUSÕES E TRABALHOS FUTUROS

A reconstrução de superfícies por visão estéreo está a cada dia ganhando mais espaço em diversas áreas de pesquisa. Entre os motivos está a relação do barateamento dos computadores aliado com os avanços da visão computacional e computação gráfica. Isso permitiu que diversos algoritmos de processamento de imagens fossem criados de acordo com o tipo de aplicação a que ele se destina.

Com a utilização de novos algoritmos e técnicas de programação a reconstrução em tempo real utilizando câmeras estéreo passou a deixar de ser impossível em computadores de pequeno porte, passando a ser predominante em aplicações de robótica e modelagem.

O sistema desenvolvido nesse trabalho pode ser utilizado em aplicações de tempo real de baixa precisão, com erros entre 2 e 5 centímetros nas reconstruções de objetos com distâncias reais inferiores a 40 centímetros.

Como proposta para futuras melhorias no sistema, estaria a opção de salvar a nuvem de pontos em arquivos que pudessem ser lidos pela maioria dos modeladores 3D do mercado. Com isso, alguns processos que consomem muito tempo, por exemplo, a etapa de triangulação de pontos poderia ser realizada nos softwares modeladores. Também, no contexto da robótica, pois permitiria que a reconstrução fosse feita somente depois do mapeamento do ambiente, tirando a sobrecarga de processamento do robô. Além disso, oferece a praticidade e conveniência de poder ver a qualquer momento uma determinada reconstrução salva previamente.

Outra melhoria seria utilizar uma câmera estéreo com maior distância de base e uma menor distância focal. Isso permitiria conseguir valores de disparidade mais exatos e um campo de visão mais ampliado para objetos próximos.

Ainda para trabalhos futuros, deseja-se acoplar o sistema a um robô para mapear o ambiente no qual ele está se movimentando e ao mesmo tempo aplicar a reconstrução de forma a modelar todo o espaço tridimensional visto pelo robô. Isso será um desafio, pois, pretende-se adotar a mesma abordagem desse trabalho, utilizando equipamentos de baixo custo.

REFERÊNCIAS

- ACHARD, Julien. **La Focale**. Disponível em: <<http://www.commentcamarche.net/contents/l-objectif/la-focale>>. Acesso em: 15 ago. 2012.
- AIRES, Guilherme de Souza. **Estimação de Movimento de Câmera Via Geometria Epipolar**. 2010. 99 p. – Dissertação de Mestrado em Engenharia Elétrica – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.
- ALBURNIO, Ronaldo dos Santos. **Métodos Passivos de Reconstrução 3D voltados á preservação Digital de Acervos Culturais**. 2012. 73 p. – Dissertação de Mestrado em Informática – Universidade Federal do Paraná, Curitiba, 2012.
- ALVES, Elaine Gracindo. **Desenvolvimento de Algoritmos de Calibração de Câmeras para Aplicações em Robótica**. 2007. 52 p. – Dissertação de Mestrado em Engenharia Elétrica – Universidade Estadual de Londrina, Londrina, 2007.
- ARAÚJO, Allan D. G de. **Uma Proposição para o Cálculo de Mapas de Disparidade de Imagens Estéreo Usando um Interpolador Neural Baseado em Funções de Base Radial**. 2010. 52 p. – Tese (Mestrado em Engenharia Elétrica) – Universidade Federal do Estado do Rio Grande do Norte, Natal, 2010.
- AYDIN, Tarkan; AKGUL, Yusuf Sinan. **A New Adaptive Focus Measure for Shape From Focus**. In M. Everingham and C. Needham, editors, Proceedings of the British Machine Conference, pages 8.1-8.10. BMVA Press, doi: 10.5244/C.22.8, 2008.
- AZEVEDO, Teresa C. S. **Reconstrução Tridimensional de Estruturas Anatômicas Externas Utilizando Métodos Volumétricos**. 3^o Congresso Nacional de Biomecânica, p. 557-562, ISBN: 978-989-96100-0-2, Bragança - Portugal, fev. 2009.
- AZEVEDO, Tereza C. S.; TAVERES, João Manuel R. S.; VAZ, Mário A. P. **Análise do Método de Calibração de Câmeras Proposto por Zhang**. 5^o Congresso Luso-Moçambicano de Engenharia, Maputo - Moçambique, set. 2008.
- BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. 1.ed. Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly, 2008.
- CAMARGO, Marcelo A. Ribeiro. **Geração de Objetos Virtuais a partir de Imagens**. 2008. 63 p. – Dissertação de Mestrado em Ciência da Computação – Universidade Metodista de Piracicaba, Piracicaba, 2008.
- FRANÇA, José A. de. **Desenvolvimento de Algoritmos de Visão Estereoscópica para Aplicações em Robótica Móvel**. 2003. 103 p. – Relatório Técnico (Mestrado em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis, 2003.
- GALO, Maurício; TOZZI, Clésio L. Série em Ciências Geodésicas “30 anos da pós-graduação em ciências Geodésicas”: **A Representação de Matrizes de Rotação e o uso de Quartérnios em Ciências Geodésicas**. Imprensa Universitária da Universidade Federal do Paraná, Curitiba, v. 1, p. 214-331, 2001.

GEIGER, Andreas; ROSER, Martin; URTASUN, Raquel. **Efficient Large-Scale Stereo Matching**. 10th Asian Conference on Computer Vision, Queenstown, New Zealand, Nov. 2010.

GUERRA, Rodrigo da Silva. **Calibração Automática de Sistemas de Visão Estéreo a partir de Movimentos Desconhecidos**. 2004. 109 p. – Tese (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

HOLANDA, Guilherme Costa et al. **Sistema de Mediação de Distância para Robôs Móveis Utilizando o Algoritmo SURF e Visão Estéreo**. V CONNEPI-2010, Maceió, nov. 2010.

JOSÉ, Marcelo Archanjo. **Reconstrução Tridimensional de Baixo Custo a partir de Par de Imagens Estéreo**. 2008. 135 p. – Tese (Mestrado em Engenharia de Sistemas Eletrônicos) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2008.

KARLSTROEM, Adriana. **Estimação de Posição e Quantificação de Erro Utilizando Geometria Epipolar entre Imagens**. 2007. 156 p. – Tese (Mestrado em Engenharia Mecânica) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2007.

KIRNER, Claudio; TORI, Romero. **Realidade Virtual: Conceitos e Tendências**. 1.ed. São Paulo: Mania de Livro, 2004.

LIBELAS. **LIBELAS (Library for Efficient Large-scale Stereo Matching)**. Disponível em: <<http://www.cvlibs.net/software/libelas.html>>. Acesso em: 16 ago. 2012.

LIBV4L2CAM. **Libv4l2cam**. Disponível em: <<http://code.google.com/p/libv4l2cam/>>. Acesso em: 16 ago. 2012.

MARQUES, Pedro Henrique; CABRAL, Allyson; PEIXOTO, Adailson. **Reconstrução de modelos 3D e Edição de Imagens a partir de Fotografias Arquitetônicas**. Workshop of Undergraduate Work. XX Brazilian Symposium on Computer Graphics and Image Processing, Maceió – Alagoas, jan. 2009.

MEDEIROS, Adelardo A. D. de. **Robótica Experimental: Material Didático**. Universidade Federal do Rio Grande do Norte – Departamento de Engenharia de Computação e Automação, Natal, 2008.

MINORU. **Minoru**. Disponível em: <<http://www.minoru3d.com/>>. Acesso em: 02 jul. 2012.

NEVES, André C. das; TSUZUKI, Marcos de S. Guerra. **Reconstrução 3D a partir de Sequência de Imagens**. Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, dez. 2012.

NIEDERÖST, Markus; NIEDERÖST, Jana; SKUCKA, Jirí. **Shape from focus. fully automated 3D reconstruction and visualization of microscopic objects**. ETH, Eidgenössische Technische Hochschule Zürich, Institute of Geodesy and Photogrammetry, 2003.

NOGUEIRA, Fernando M. de Almeida; TOZZI, Clésio Luis. **Geração Automática de Mapas de Disparidade em Visão Estéreo**. Campinas - São Paulo, Brasil, 1999.

OLIVEIRA, Janaina Souza. **Reconstrução Tridimensional de Imagens Obtidas por Tomografia Computadorizada**. 2007. 88 p. – Monografia (Bacharel em Sistema de Informação) – Universidade Estadual de Montes Claros, Montes Claros, 2007.

OLIVEIRA, Marco A. F. de. **Correlacionamento Estéreo de Complexidade Linear Baseado em Indexação de Regiões**. 2006. 53 p. – Dissertação de Mestrado em Ciência da Computação – Universidade Federal de Santa Catarina, Florianópolis, 2006.

OPENCV. **Camera Calibration and 3d Reconstruction**. Disponível em: <http://opencv.willowgarage.com/documentation/cpp/calib3d_camera_calibration_and_3d_reconstruction.html#index-7867>. Acesso em: 15 ago. 2012.

_____. **OpenCV**. Disponível em: <<http://opencv.org/>>. Acesso em: 15 ago. 2012.

PCL. **PCL (Point Cloud Library)**. Disponível em: <www.pointclouds.org>. Acesso em: 21 jul. 2012.

ROCHA, Lourena; GONÇALVES, Luiz. **An Overview of Three-Dimensional Videos: 3D Content Creation, 3D Representation and Visualization, Current Advancements in Stereo Vision**, Asim Bhatti (Ed.), ISBN: 978-953-51-0660-9, InTech, Available from: <http://www.intechopen.com/books/current-advancements-in-stereo-vision/an-overview-of-three-dimensional-videos-3d-content-creation-3d-representation-and-visualization>, 2012.

SASDELLI, Duílio Campos. **Extração de Pontos de Interesse de Imagens Utilizando o Algoritmo SHIFT no ambiente Android**. 2010. 47 p. – Monografia de projeto da disciplina de Projeto Orientado em Computação I – Universidade Federal de Minas Gerais, Belo Horizonte, 2010.

SOBRE ARTE. **Elementos da Perspectiva**. Disponível em: <http://www.sobrearte.com.br/desenho/perspectiva/elementos_da_perspectiva.php>. Acesso em: 22 mai. 2012.

SOUZA, Anderson A. de S; GONÇÁLVES, Luiz M. Garcia. **Mapeamento de Ambientes em Grade de Ocupação Probabilística a partir de Visão Estéreo**. SBAI – Simpósio Brasileiro de Automação Inteligente, São João Del-Rei – Minas Gerais, 2011.

STEMMER, Marcelo R. et al. **Apostila de Sistema de Visão**. Florianopolis, 2005.

V4L2spec. **V4L2 API Specification (Revision 0.24)**. Disponível em: <<http://v4l2spec.bytesex.org/>>. Acesso em: 16 ago. 2012.

VOGEL, Oliver et al. **Making Shape from Shading Work for Real-World Images**. 31st DAGM Symposium on Pattern Recognition, Heidelberg – Berlim, p. 191-200, ISBN: 978-3-642-03797-9, 2009.

XBOX. **Kinect**. Disponível em: <<http://www.xbox.com/pt-BR/Kinect/>>. Acesso em: 15 ago. 2012.