Marcelo Archanjo José

**RECONSTRUÇÃO TRIDIMENSIONAL DE BAIXO CUSTO
A PARTIR DE PAR DE IMAGENS ESTÉREO**

São Paulo
2008

Marcelo Archanjo José

RECONSTRUÇÃO TRIDIMENSIONAL DE BAIXO CUSTO A PARTIR DE PAR DE IMAGENS ESTÉREO

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do título de
Mestre em Engenharia.

Área de Concentração:
Sistemas Eletrônicos

Orientadora:
Profa. Dra. Roseli de Deus Lopes

São Paulo
2008

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de junho de 2008.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

José, Marcelo Archanjo

Reconstrução tridimensional de baixo custo a partir de par de imagens estéreo / M.A. José. -- ed.rev. --São Paulo, 2008. 135 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Visão computacional 2.Reconstrução (Geometria e modelagem computacional) 3.Algoritmos para imagens 4.Visão binocular I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

DEDICATÓRIA

Dedico este trabalho à memória de meu pai Miguel Archanjo José.

AGRADECIMENTOS

Agradeço à minha mãe Josecila Maria de Vasconcelos José pelo apoio e incentivo, ao meu irmão André Archanjo José pelo acompanhamento, dicas, orientações e correções, à minha namorada Audrei Lepski pela paciência, apoio e revisões do texto.

Agradeço também à minha orientadora Profa. Dra. Roseli de Deus Lopes por ter acreditado nas minhas idéias e me apoiado. E ao Prof. Dr. Osvaldo Ramos Tsan Hu pela recomendação para iniciar o mestrado no LSI.

RESUMO

A obtenção e a reconstrução da geometria tridimensional (3D) de objetos e ambientes têm importância crescente em áreas como visão computacional e computação gráfica. As formas atuais de obtenção e reconstrução 3D necessitam de equipamentos e montagens sofisticadas que, por consequência, têm custos elevados e aplicação limitada.

Este trabalho apresenta criticamente os principais algoritmos para a reconstrução 3D a partir de par de imagens estéreo e identifica os mais viáveis para utilização com equipamentos convencionais. Por meio da implementação de alguns destes algoritmos, da comparação dos resultados obtidos em sua execução e também pela comparação com os resultados encontrados na literatura, são identificadas as principais deficiências. São propostas adequações aos algoritmos existentes, em particular, é apresentada a proposta da técnica das faixas que proporciona a redução drástica no consumo de memória para o processamento da geometria 3D e que possui desempenho computacional melhor em relação às técnicas tradicionais.

Foi implementado um protótipo de sistema de reconstrução 3D que permite a reconstrução pelas diferentes técnicas estudadas e propostas, bem como permite visualizar o cenário reconstruído sob diferentes pontos de vista de forma interativa.

Palavras-chave: Correspondência Estéreo. Modelagem 3D. Reconstrução 3D. Visão 3D. Visão Computacional. Visão Estéreo.

ABSTRACT

The acquisition and reconstruction of three-dimensional (3D) geometry of objects and environments have their importance growing in areas such as Computer Vision and Computer Graphics. The current methods to acquire and reconstruct three-dimensional data need sophisticated equipments and assemblies, which have expensive costs and limited applications.

This work presents the main algorithms for 3D reconstruction using a pair of stereo images and identifies which are viable to use with conventional equipments. Through the implementation of some of these algorithms, by comparing the results obtained and comparing with the results presented in the literature, the main limitations were identified. This work proposes adjustments in the existing algorithms, in particular it proposes the stripping technique, which provides a huge memory usage reduction for 3D geometry processing and better computing performance if compared with traditional approaches.

A prototype system for 3D reconstruction was implemented, which allows the reconstruction using the different researched and proposed techniques and allows interactive visualization of the reconstructed scene in different angles.

Keywords: 3D Reconstruction. 3D Modeling. 3D Vision. Computer Vision. Stereo Matching. Stereo Vision.

SUMÁRIO

CAPÍTULO 1	Introdução	1
1.1	Apresentação	1
1.2	Síntese do Problema	8
1.3	Objetivos	8
1.4	Relevância	9
1.5	Materiais e Métodos	10
1.6	Estrutura da Dissertação	10
CAPÍTULO 2	Conceitos e Algoritmos de Visão Estéreo	11
2.1	Introdução	11
2.2	Visão Estéreo	11
2.3	Geometria	13
2.4	Correspondência	15
2.5	Restrição Epipolar	15
2.6	Abordagens Tradicionais para Visão Estéreo	18
2.6.1	Uso do DSI em Métodos Locais	26
2.7	Comparação de Requisitos	31
2.8	Trabalhos Correlatos	34
2.9	Conclusão	35
CAPÍTULO 3	Avaliação dos Principais Algoritmos	36
3.1	Consumo de Memória	41
3.2	Calibração Rígida	42
3.3	Precisão nas Cores	44
3.4	Conclusão	45
CAPÍTULO 4	Soluções Propostas e Resultados Obtidos	46
4.1	Protótipo de Sistema	46
4.1.1	Delimitação do Escopo	47
4.1.2	Desenvolvimento	47
4.2	Consumo de Memória	51

4.2.1	Técnica das Faixas.....	52
4.2.2	Custo de Correspondências	54
4.2.3	Agregação de Evidências.....	55
4.2.4	Seleção de Disparidades.....	56
4.2.5	Consumo de Memória	57
4.3	Calibração Rígida.....	62
4.4	Precisão nas Cores.....	67
4.5	Desempenho em Execução	72
4.6	Geração da Malha 3D	79
4.7	Conclusão	82
CAPÍTULO 5 Resultados em uma Situação Real.....		83
5.1	Experimento.....	83
5.2	Conclusão	91
CAPÍTULO 6 Conclusões		92
6.1	Contribuições	92
6.2	Trabalhos Futuros.....	93
6.3	Conclusões	94
REFERÊNCIAS BIBLIOGRÁFICAS		95
APÊNDICE A - Consumo Total de Memória		99
APÊNDICE B – Documentação Publicada no <i>Site Source Forge</i>		103

LISTA DE ILUSTRAÇÕES

Figura 1 - Reconstrução 3D de face obtida por varredura a <i>laser</i> com o digitalizador Minolta VIVID 300 3D, fonte: http://www.visuallee.com/weblog/2002_12_01_archive.html (acesso em janeiro de 2008).	1
Figura 2 - (a) Sete fotos em ângulos diferentes do Carrossel do Louvre; (b) reconstrução 3D usando Canoma, fonte: http://www.canoma.com/movies.html (acesso em janeiro de 2008).	2
Figura 3 - (a)(b) Respectivamente, foto da esquerda e da direita de um par estéreo; (c) reconstrução 3D do ambiente, fonte (FAUGERAS, 1993).	3
Figura 4 - (a) Reconstrução 3D usando ressonância magnética, fonte: (NAKAZONE, 2005); (b) reconstrução 3D usando tomografia computadorizada, fonte: (SOUZA; CENTENO; PEDRINI, 2003); (c) reconstrução 3D da face de um bebê usando ultrassom 3D, fonte: <i>SPIE Press, Medical Imaging</i>	4
Figura 5 - Exemplo de correspondência entre as imagens de um par estéreo. O ponto que na imagem da esquerda está em $y=202$, na imagem da direita está em $y=193$, indicando uma disparidade de 9 pixels. Par estéreo original publicado em (SCHARSTEIN; SZELISKI, 2002).	5
Figura 6 - Consumo de memória para a carga do DSI.	6
Figura 7 - Par de imagens estéreo (SCHARSTEIN; SZELISKI, 2002).	12
Figura 8 - Mapa de disparidades.	13
Figura 9 - Ilustração da geometria estéreo, representando duas câmeras idênticas e paralelas com distância focal f e distância entre elas b (SCHARSTEIN, 1999).	13
Figura 10 - Ilustração de restrição epipolar. Neste exemplo, o ponto P que gera PL em L deve pertencer à reta definida por CL e PL , sendo que a projeção de P em R deve pertencer à reta eR definida como linha epipolar (SCHARSTEIN, 1999).	16
Figura 11 - Montagem Estéreo com duas câmeras idênticas e paralelas (SCHARSTEIN, 1999).	17

Figura 12 - <i>Disparity Space Image</i> (DSI).....	18
Figura 13 - (a) Par estéreo de entrada (SCHARSTEIN; SZELISKI, 2002); (b)(c)(d) representação em tons de cinza da diferença quadrática entre o par estéreo, para as disparidades 21, 16 e 10, respectivamente.....	20
Figura 14 - (a) Representação do DSI indicando um corte no $y = 151$; (b) fatia (x,d) do DSI para $y = 151$. Fonte das imagens individuais: (SCHARSTEIN; SZELISKI, 2001).....	21
Figura 15 - (a) Corte em um DSI (SCHARSTEIN; SZELISKI, 2001); (b) indicação de um caminho que minimiza a energia.....	21
Figura 16 - (a) Mapa de disparidades ideal; (b) mapa de disparidades método local; (c) mapa de disparidades método global (<i>graph cuts</i>); (d) relação entre (a) e (b) onde a cor branca indica pixels iguais e a cor preta indica pixels diferentes; (e) relação entre (a) e (c); As figuras (a)(b)(c) foram extraídas de (SCHARSTEIN; SZELISKI, 2001). Obs.: Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.	22
Figura 17 - Ponto em questão e sua janela 3x3 na imagem da esquerda, e a linha epipolar na imagem da direita, onde é procurada a correspondência. Par estéreo original publicado em (SCHARSTEIN; SZELISKI, 2002).	23
Figura 18 - Descrição das variáveis para a janela da esquerda e da direita. ...	24
Figura 19 - Cálculos repetidos com o deslocamento da janela da imagem da esquerda.	27
Figura 20 - Planos que armazenam os custos de correspondência.....	28
Figura 21 - Posições envolvidas durante o processo de agregação para o ponto (60,70).....	29
Figura 22 - Simulação de valores nos diversos planos para a posição (40,50).....	30
Figura 23 - Síntese das escolhas e desafios deste trabalho.	37
Figura 24 - Pares estéreo: no alto, imagem conhecida como Venus, e abaixo, Tsukuba (SCHARSTEIN; SZELISKI, 2002).....	39
Figura 25 - (a) Mapa de disparidades da imagem Venus obtido com o código do Middlebury College; (b) mapa de disparidades da imagem	

Venus obtido com o código desenvolvido para este trabalho; (c) relação entre (a) e (b), sendo que os pixels brancos indicam pixels iguais na comparação e os pretos as diferenças; (d) mapa de disparidades da imagem Tsukuba obtido com o código do Middlebury College; (e) mapa de disparidades da imagem Tsukuba obtido com o código desenvolvido para este trabalho; (f) Relação entre (d) e (e).....	40
Figura 26 - Tempo de processamento de ambos os códigos.....	41
Figura 27 - Distorções simuladas nas imagens de entrada.....	42
Figura 28 - (a) Par estéreo original da imagem Vênus (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades; (b) deslocamento de linha na imagem da direita, o mapa de disparidades gerado e relação deste mapa de disparidades com o original de (a) onde a cor preta indica pixels diferentes; (c) rotação na imagem da direita, seu mapa de disparidades e a relação com a imagem (a); (d) curvatura na imagem da direita, seu mapa de disparidade e a relação com a imagem(a).	43
Figura 29 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades; (b) brilho modificado na imagem da direita, mapa de disparidades gerado e à direita comparação entre os dois mapas de disparidade, onde a cor preta indica pixels diferentes.....	44
Figura 30 – (a) Módulos do sistema que foram desenvolvidos; (b) Representação do sistema operacional e alguns de seus módulos.....	48
Figura 31 – Tela do sistema desenvolvido durante o processamento do par de imagens, gerando o mapa de disparidades e a geometria 3D.....	49
Figura 32 – Tela do sistema desenvolvido na opção de reconstrução 3D diretamente do mapa de disparidades.....	50
Figura 33 - Diagrama da Técnica das Faixas.....	53
Figura 34 - Custo de correspondência para o primeiro preenchimento das faixas de agregação para uma janela 3x3.	54
Figura 35 - Custo de correspondência para demais linhas até o fim das imagens de entrada.	55
Figura 36 - No alto a primeira agregação, abaixo as seguintes aproveitam os resultados obtidos anteriormente.	56

Figura 37 - Processo de agregação para as demais faixas de agregação.....	56
Figura 38 - Tamanho das faixas de agregação.....	58
Figura 39 - Consumo de memória para a carga do DSI e das faixas de agregação.	59
Figura 40 - Consumo de memória para a carga das faixas de agregação.....	60
Figura 41 - Ponto selecionado na imagem da esquerda e janela de procura na imagem da direita.....	63
Figura 42 - Par estéreo original da imagem Vênus (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades.....	64
Figura 43 - (a) Ilustração da distorção; (b) mapa de disparidades para uma busca 1D; (c) comparação do mapa de disparidade (b) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças.; (d) mapa de disparidade para uma busca 2D (5 linhas de procura); (e) comparação do mapa de disparidade (d) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças	64
Figura 44 - (a) Ilustração da distorção; (b) mapa de disparidades para uma busca 1D; (c) comparação do mapa de disparidade (b) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças; (d) mapa de disparidade para uma busca 2D (3 linhas de procura); (e) comparação do mapa de disparidade (d) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças.	65
Figura 45 - (a) Ilustração da distorção; (b) mapa de disparidades para uma busca 1D; (c) comparação do mapa de disparidade (b) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças; (d) mapa de disparidade para uma busca 2D (7 linhas de procura); (e) comparação do mapa de disparidade (d) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças.	66
Figura 46 - (a) Imagem com faixa azul clara no centro de uma região azul escura; (b) representação gráfica das intensidades das cores para a linha pontilhada em (a); (c) Primeira derivada.....	68
Figura 47 - Máscaras conhecidas como operadores de Sobel.....	69
Figura 48 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002); (b) mesmo par estéreo processado com o operador gradiente, valores absolutos e limitados a 255.	69

Figura 49 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades; (b) brilho modificado na imagem da direita, mapa de disparidades gerado e à direita comparação com o mapa de disparidades de (a), onde a cor preta indica pixels diferentes; (c) brilho modificado na imagem da direita, mapa de disparidades gerado (obtido utilizando operador gradiente) e à direita comparação com o mapa de disparidades de (a), onde a cor preta indica pixels diferentes.....	70
Figura 50 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades (obtido utilizando Operador Gradiente); (b) brilho modificado na imagem da direita, mapa de disparidades gerado (obtido utilizando Operador Gradiente) e à direita comparação com o mapa de disparidades de (a), onde a cor preta indica pixels diferentes.....	71
Figura 51 - Pares de Imagens Estéreo, Cones (alto), Venus (meio) e Tsukuba (baixo).	75
Figura 52 - Comparação do consumo de memória.	77
Figura 53 - Desempenho para a procura por correspondências.	78
Figura 54 - Malha 3D inicial.....	79
Figura 55 - (a) Origem das coordenadas no <i>bitmap</i> se encontra no canto superior esquerdo; (b) origem das coordenadas se encontra no centro.	80
Figura 56 - (a) Faixas com tons de cinza partindo do branco até o preto; (b) resultado gerado com a técnica de Reconstrução 3D descrita aqui. ...	80
Figura 57 - (a) Imagem da esquerda do para estéreo Tsukuba (SCHARSTEIN; SZELISKI, 2002);(b) mapa de disparidades ideal utilizado; (c) apresentação da geometria 3D reconstruída a partir do mapa de disparidades em modo <i>wireframe</i> ; (d) Reconstrução 3D em modo sólido.....	81
Figura 58 - Calibração das câmeras para obtenção do par de imagens estéreo.	83
Figura 59 - Imagens obtidas em ambiente não controlado.	84

Figura 60 – Tela do sistema executando com imagens de 5.1 MegaPixel, em destaque a resolução das imagens de entrada e o tempo de execução.....	85
Figura 61 - (a) Imagem da esquerda onde é identificada uma linha específica $y=217$; (b) imagem da direita identificando a mesma linha da imagem (a) porém agora ela se encontra na posição $y=216$	85
Figura 62 - Execução usando as intensidades das imagens de entrada, em destaque a resolução das imagens de entrada e o tempo de execução.....	86
Figura 63 - Execução usando operador gradiente.	87
Figura 64 - Mapa de disparidades corrigido.	88
Figura 65 - Execução somente a partir do mapa de disparidades.	88
Figura 66 - Resultados obtidos na reconstrução 3D à esquerda <i>flat</i> e à direita <i>wireframe</i> . No alto, a imagem de entrada e abaixo o mapa de disparidades.....	89
Figura 67 - Tela do 3Ds Max com o arquivo VRML carregado.....	90
Figura 68 - Imagem gerada pelo 3Ds Max a partir do arquivo VRML e da imagem de entrada.	90
Figura 69 - Agregação de evidências usando plano temporário.	100

LISTA DE TABELAS

Tabela 1 - Comparação de requisitos entre aplicações para visão estéreo, Dados extraídos de (SCHARSTEIN, 1999), exceto reconstrução 3D de baixo custo.	32
Tabela 2 – Cálculo de consumo de memória do DSI e da faixa de agregação para algumas resoluções.	60
Tabela 3 - Cálculo de consumo de memória para as demais etapas do processo de procura por correspondências.	61
Tabela 4 – Parâmetros e dados comparativos.	76
Tabela 5 - Comparação das formulas para cada etapa.	101
Tabela 6 – Simulação de consumo de memória para as duas técnicas nas etapas otimizadas.	102

DICIONÁRIO DE TERMOS (INGLÊS - PORTUGUÊS)

Area Based = Baseado na Área

Computer Vision = Visão Computacional

Disparity Selection = Seleção das Disparidades

Epipolar Constraint = Restrição Epipolar

Evidence Aggregation = Agregação das Evidências

Feature Based = Baseado na Característica

Full Calibration = Calibração Completa

Image Based Rendering = Representação Baseada em Imagens

Matching Cost = Custo de Correspondência

View Synthesis = Síntese Visual

Stereo Matching = Casamento Estéreo

Stereo Correspondence = Correspondência Estéreo

Stereo Vision = Visão Estéreo

Stripping Technique = Técnica das Faixas

Sum of Absolute Differences (SAD) = Soma das Diferenças Absolutas

Sum of Squared Differences (SSD) = Soma das Diferenças Quadráticas

Visual Servoing = Controle baseado em Visão

View Synthesis = Síntese Visual

Winner-take-all (WTA) = Vencedor-leva-tudo

NOTAÇÃO

- Termos que já são de uso comum em inglês serão apresentados em inglês destacados em itálico, exemplo: *wireframe*.
- Quando o termo é conhecido em inglês, mas uma tradução literal não é adequada, na sua primeira apresentação no texto será apresentado em inglês com uma tradução explicativa, em seguida será apresentado em inglês destacado em itálico; exemplo: *Disparity Space Image (DSI)* (volume com as disparidades das imagens de entrada).

CAPÍTULO 1 Introdução

1.1 Apresentação

A Reconstrução Tridimensional (3D) uma área muito pesquisada tanto na Visão Computacional quanto na Visualização Científica. Ela tem por objetivo obter uma representação geométrica tridimensional de ambientes ou objetos. Suas aplicações são amplas em áreas como arquitetura, medicina, cartografia 3D, robótica e realidade aumentada.

Informações sobre a geometria 3D de um ambiente, objeto ou corpo podem ser adquiridas principalmente por varredura a *laser*, sonares, tomografia, ressonância magnética e fotografias.

Os sistemas de varredura a *laser* medem a distância da fonte baseados no atraso entre a emissão do sinal luminoso e seu retorno (FERNANDES et al. 2004). São muito empregados na reconstrução de fachadas, objetos e faces. A Figura 1 apresenta um exemplo de reconstrução 3D de face com dados obtidos por varredura a *laser*.

Figura 1 - Reconstrução 3D de face obtida por varredura a *laser* com o digitalizador Minolta VIVID 300 3D, fonte: http://www.visuallee.com/weblog/2002_12_01_archive.html (acesso em janeiro de 2008).

Sistemas baseados em fotografias realizam a reconstrução 3D a partir de uma foto (determina-se a face do objeto em questão e realiza-se a extrusão) ou de múltiplas fotos em ângulos distintos como ilustra a Figura 2. No caso de múltiplas fotos, marcam-se pontos de referência para identificar as arestas comuns do objeto em questão, nas diversas fotos e a partir desta informação é produzida a geometria 3D. Softwares como o Canoma (<http://www.canoma.com>) ou o Photomodeler (<http://www.photomodeler.com>) são especializados neste tipo de modelagem 3D.

Figura 2 - (a) Sete fotos em ângulos diferentes do Carrossel do Louvre; (b) reconstrução 3D usando Canoma, fonte: <http://www.canoma.com/movies.html> (acesso em janeiro de 2008).

Também é possível estimar a profundidade por meio de duas imagens em par estéreo (SCHARSTEIN; SZELISKI, 2001), (FAUGERAS, 1993) (Figura

3) ou múltiplas imagens em pares estéreo (KANADE et al., 1996), (KIMURA et al., 1999). As imagens utilizadas podem ser tanto em perspectiva quanto panorâmicas (GRASSI; OKAMOTO, 2006), (SVODOBA; PAJDLA; HLAVAC, 1997).

Figura 3 - (a)(b) Respectivamente, foto da esquerda e da direita de um par estéreo; (c) reconstrução 3D do ambiente, fonte (FAUGERAS, 1993).

Um trabalho de validação de reconstrução 3D é apresentado em (EID et al., 2004). Nele é comparada a reconstrução 3D com dados obtidos através de varredura laser e fotografia estéreo.

Em técnicas como ressonância magnética (Figura 4a) e tomografia computadorizada (Figura 4b), obtêm-se fatias da região em questão a partir de informações unidimensionais obtidas em diversos ângulos, que é a base para a construção de uma imagem bidimensional desta fatia. A reconstrução 3D vem da ligação da seqüência de fatias.

Sonares 3D são utilizados em atividades navais como a reconstrução 3D do fundo de lagos. Na medicina a ultra-sonografia 3D pode ser usada para a reconstrução 3D da face de bebês ainda no útero (Figura 4c).

Figura 4 - (a) Reconstrução 3D usando ressonância magnética, fonte: (NAKAZONE, 2005); (b) reconstrução 3D usando tomografia computadorizada, fonte: (SOUZA; CENTENO; PEDRINI, 2003); (c) reconstrução 3D da face de um bebê usando ultrassom 3D, fonte: *SPIE Press, Medical Imaging*.

Os equipamentos necessários para cada uma destas formas de reconstrução 3D têm custos elevados, sendo que entre todos os equipamentos de aquisição, os mais acessíveis são as câmeras digitais utilizadas nos sistemas baseados em fotografias.

Entre os sistemas que utilizam fotografias, destacam-se os baseados em visão estéreo (imagens em perspectiva), utilizados na visão computacional. Estes sistemas permitem a automação completa do processamento, pois o posicionamento das câmeras facilita a identificação de correspondências entre um par de imagens, o que não acontece com outras técnicas que exigem a intervenção manual para identificação de pontos de correspondências.

O problema básico da visão estéreo é a procura por correspondências num par de imagens de entrada, e uma vez encontrada esta correspondência, pode-se medir a disparidade entre as imagens (deslocamento horizontal) como ilustra a Figura 5.

Figura 5 - Exemplo de correspondência entre as imagens de um par estéreo. O ponto que na imagem da esquerda está em $y=202$, na imagem da direita está em $y=193$, indicando uma disparidade de 9 pixels. Par estéreo original publicado em (SCHARSTEIN; SZELISKI, 2002).

Os principais aspectos que precisam ser apresentados para a compreensão do problema são consumo de memória, calibração rígida e precisão nas cores.

a) Consumo de Memória

Em linhas gerais, os algoritmos de visão estéreo podem ser divididos em dois grupos principais: locais e globais. Ambos utilizam um volume conhecido como *Disparity Space Image (DSI)* (volume com as disparidades das imagens

de entrada). Maiores detalhes serão apresentados no CAPÍTULO 2, no item 2.6.

O tamanho ocupado pelo DSI está diretamente ligado ao tamanho da imagem de entrada, pois ele é um volume (x,y,d). Considerando que o resultado da diferença quadrática (entre cada pixel das imagens) seja armazenado em uma variável do tipo *float* (4 bytes), o número de níveis de disparidade **d** seja 4%¹ da largura da imagem **x** e que a altura de uma imagem **y** siga o padrão de 3/4 da largura ($y = 3x/4$), obtém-se:

$$\begin{aligned} DSI &= 4xyd \\ DSI &= 4x \frac{3}{4} x 0,04x \\ DSI &= 0,12x^3 \end{aligned} \quad (1)$$

Na Figura 6, é apresentada a curva com a estimativa do consumo de memória do DSI para diferentes resoluções. Note que se trata de uma curva cúbica.

O consumo de memória somente do DSI (usando as técnicas tradicionais) é bastante elevado. Por exemplo, para uma resolução de 2592 x 1944 (5.1 Megapixel), o consumo é de aproximadamente 1,9 GB de memória RAM. Se a resolução é elevada para 5200 x 3900 (20 Megapixel), o consumo cresce para aproximadamente 15,7 GB de memória RAM. Este consumo de memória nos dias atuais é proibitivo para o contexto de uma solução de baixo custo.

b) Calibração Rígida

Dentro do contexto de laboratório é normal a utilização da restrição epipolar em Visão Estéreo (mais detalhes no CAPÍTULO 2, em 2.5 Restrição Epipolar). Esta restrição facilita muito o processo de procura por correspondências, pois limita o processo em comparar uma linha de uma imagem com uma linha específica da outra (procura unidimensional - 1D). Por outro lado, as câmeras precisam ser iguais, precisas e a calibração das câmeras tem que ser bastante rígida.

A calibração rígida e a precisão das câmeras exigem equipamentos com custo elevado, além de um ambiente controlado.

c) Precisão nas cores

O processo de procura por correspondências normalmente é realizado considerando-se apenas as intensidades das cores obtidas nas fotos, exigindo-se precisão nas duas câmeras utilizadas. Por outro lado, é difícil manter a mesma luminosidade de um ambiente não controlado entre duas fotos tiradas em momentos diferentes com a mesma câmera.

1.2 Síntese do Problema

Quanto à redução de custos na reconstrução 3D, fica claro que duas etapas são críticas: a aquisição e o processamento. Na aquisição, os sistemas baseados em fotografias se mostram como os mais viáveis. No processamento, há três problemas que precisam ser resolvidos: consumo de memória, calibração rígida e precisão nas cores.

1.3 Objetivos

O objetivo principal deste trabalho é pesquisar, conceber e desenvolver um sistema de reconstrução 3D de baixo custo, funcional, flexível e baseado em áreas de pesquisa com perspectivas futuras de expansão. Para atingir este objetivo principal é necessário cobrir os seguintes objetivos específicos:

- pesquisar sobre reconstrução 3D e visão estéreo;
- implementar, testar e comparar os algoritmos da literatura;
- identificar os principais desafios;
- conceber soluções para os desafios encontrados, em especial, o consumo de memória, a calibração rígida e a precisão das cores;
- desenvolver e testar novas soluções;
- comparar com os existentes.

Pretende-se implementar um protótipo de sistema que permita gerar reconstruções 3D e visualizações imediatas dos resultados em equipamentos de baixo custo, a partir de pares de fotografias obtidas com câmeras convencionais.

1.4 Relevância

Os projetos de pesquisa e os sistemas comerciais disponíveis em reconstrução 3D exigem ambientes complexos e onerosos. A criação de um conjunto de algoritmos que permita baratear e popularizar o processo de reconstrução 3D facilitaria a difusão destes conhecimentos para inúmeros estudantes, despertando o interesse nesta área, bem como em visão estéreo e visão computacional.

A reconstrução 3D a partir de par de fotografias estéreo permite modelar ambientes, objetos ou criar avatares 3D a partir de objetos e cenários reais.

Em um país como o Brasil, a redução de custos proposta por este trabalho é fundamental para viabilizar a disseminação mais ampla de conhecimentos em reconstrução 3D e visão computacional.

A motivação deste trabalho veio da pesquisa da literatura e da constatação experimental da viabilidade de realizar reconstrução 3D em equipamentos convencionais, mesmo com algumas limitações. Corrigir estas limitações permitiria a reconstrução 3D em equipamentos de baixo custo, viabilizando não só a sua aplicação a uma maior faixa de usuários e diferentes aplicações, mas também a difusão do conhecimento nesta área a fim de estimular novos estudos e desenvolvimentos.

1.5 Materiais e Métodos

Para realizar as pesquisas, testes e implementações deste trabalho foi utilizado um computador pessoal (PC) comum (Pentium III de 900MHz, com 384MB de RAM e *Windows 2000 Professional*) e uma câmera digital Sony®. A estratégia utilizada foi a de desenvolver um código próprio das abordagens tradicionais em visão estéreo, para identificar claramente suas deficiências e, posteriormente, propor soluções, implementando o protótipo de um sistema com as soluções propostas.

1.6 Estrutura da Dissertação

Esta dissertação está organizada em seis capítulos. O CAPÍTULO 1 introduz e apresenta os objetivos e a relevância deste trabalho. O CAPÍTULO 2 descreve os conceitos de visão estéreo, e detalha os principais algoritmos relacionados ao trabalho. O CAPÍTULO 3 apresenta e avalia criticamente os principais algoritmos encontrados na literatura. O CAPÍTULO 4 apresenta o sistema desenvolvido, detalha a implementação das soluções propostas e mostra os resultados alcançados para cada situação incluindo desempenho em execução e o consumo de memória. O CAPÍTULO 5 apresenta informações sobre um caso real de utilização do sistema desenvolvido, demonstrando as funcionalidades obtidas através dos conceitos explorados. O CAPÍTULO 6 apresenta a síntese das contribuições do trabalho e as principais conclusões e sugestões de trabalhos futuros.

CAPÍTULO 2 Conceitos e Algoritmos de Visão Estéreo

2.1 Introdução

A reconstrução 3D visa recuperar as informações tridimensionais de ambientes reais (ou virtuais, a partir de imagens sintéticas de teste).

Os sistemas baseados em fotografias naturalmente já trabalham com informações bidimensionais (x,y), faltando a profundidade (z). Para a obtenção da profundidade, a chave é encontrar as correspondências entre duas ou mais imagens. Em abordagens que usam câmeras estereo, a estratégia se baseia no funcionamento do sistema visual humano (e de muitos animais), obtendo informações tridimensionais a partir de duas imagens, a chamada visão binocular (estereoscópica). Cada olho projeta na retina uma imagem que é transmitida para o cérebro, onde são obtidas as informações tridimensionais a partir da disparidade entre as imagens. É claro que o processo é mais complexo que isso, incluindo outros fatores como raciocínio e conhecimento prévio de objetos.

2.2 Visão Estéreo

A visão computacional faz uso de conhecimentos da Inteligência Artificial (RUSSELL; NORVIG, 2004) e do Processamento de Imagens e tem por objetivo extrair informações para o reconhecimento de características. Ela possui diversos ramos, entre eles a visão estereo, que atua especificamente na

situação de extrair informações de profundidade a partir de duas imagens de entrada com deslocamento lateral.

A visão estéreo, como Daniel Scharstein (SCHARSTEIN, 1999) descreve, é a inferência geométrica da cena a partir de duas ou mais fotos digitais tiradas simultaneamente de posições ligeiramente diferentes. Isto é feito sem esforço algum pelo sistema visual humano, que traduz boa parte das informações tridimensionais das diferenças entre as visões de cada olho.

Para o caso de duas imagens, a informação de profundidade é obtida por meio da determinação da disparidade entre os pontos correspondentes em ambas as imagens. A Figura 7 mostra um exemplo de imagens estéreo.

Figura 7 - Par de imagens estéreo (SCHARSTEIN; SZELISKI, 2002).

Quando se trabalha com mais de duas imagens, o cálculo da disparidade é feito aos pares, isto é, o processo é realizado sucessivamente entre cada par de imagens.

Um resultado comumente apresentado é o mapa de disparidades, no qual os tons de cinza claros representam objetos próximos e os tons escuros representam objetos distantes (Figura 8). Trata-se de uma representação bidimensional, de um cenário tridimensional, onde a profundidade é apresentada na forma de tons de cinza.

Figura 8 - Mapa de disparidades.

2.3 Geometria

Seja um ponto P e suas duas imagens: P_L na imagem da esquerda e P_R na imagem da direita. Seja f a distância focal e b a distância entre as duas câmeras. Considerando que o ponto P tem uma profundidade Z e um deslocamento lateral X (em relação à câmera da esquerda) e ainda P_L está na coordenada X_L e P_R na coordenada X_R , a Figura 9 ilustra a situação descrita.

Figura 9 - Ilustração da geometria estéreo, representando duas câmeras idênticas e paralelas com distância focal f e distância entre elas b (SCHARSTEIN, 1999).

Conclui-se por semelhança de triângulos que:

$$\frac{X_L}{f} = \frac{X}{Z} \quad (2)$$

e

$$\frac{X_R}{f} = \frac{X + b}{Z} \quad (3)$$

Para a disparidade **d**, tem-se:

$$d = X_R - X_L = \frac{f * b}{Z} \quad (4)$$

Note que a disparidade é diretamente proporcional à distância focal **f** e à distância entre as câmeras **b**, e que é inversamente proporcional à profundidade **Z**. Uma vez que a distância focal e a distância entre as câmeras são constantes para um determinado par de imagens, um mapa de disparidades provê uma relação inversa das profundidades da cena. É muito fácil provar isto, observe sua mão direita à frente do seu olho direito (com o olho esquerdo fechado) numa posição próxima ao centro da visão; agora sem mover a mão, feche o olho direito e abra o esquerdo, note que a mão não está no centro da visão e sim bastante deslocada para a direita, este é um exemplo de grande disparidade. Continuando, agora olhando para um objeto muito distante, como a Lua, fazendo o mesmo processo: centralize a visão com o olho direito enquanto o esquerdo está fechado, em seguida feche o direito e abra o esquerdo; você vai notar que a Lua (ou qualquer objeto distante) continua no centro da visão apesar de ser outro olho que está vendo; este é um exemplo de pouca (ou nenhuma) disparidade. Logo, se conclui que há uma grande disparidade para objetos próximos (como a mão) e que há pouca ou nenhuma disparidade para objetos distantes (como a Lua).

2.4 Correspondência

Mas, como se pode concluir que em um par de imagens, um ponto P_L corresponde ao mesmo ponto P , tendo-se o ponto P_R ? Resolver este problema de procura por correspondências é a parte difícil da visão estéreo. As dificuldades incluem ambigüidades devido a superfícies com texturas e/ou estruturas repetitivas, regiões com intensidades uniformes, incertezas causadas por ruído, bem como oclusão parcial (visibilidade de um ponto somente para uma câmera, devido à paralaxe).

Basicamente, existem duas formas de procurar por correspondências. Uma, trata pontos com maior grau de certeza, é a idéia que está por trás dos algoritmos estéreo baseados na característica *feature-based*, onde a procura por correspondência fica restrita a partes como bordas e cantos, gerando desta forma um mapa de disparidades esparso, que precisa ser interpolado caso haja a necessidade de informações de profundidade de pontos não tratados. A outra forma de procura trata a imagem por regiões (ou áreas) que contêm suficiente informação para prover correspondências não ambíguas. Esta forma é conhecida como baseada na área, *area-based* que possui a vantagem de produzir um mapa de disparidades cheio, porém com possíveis erros, uma vez que pontos distantes das bordas e cantos possuem um menor grau de certeza na correspondência.

2.5 Restrição Epipolar

Com o desafio de encontrar a correspondência de pontos entre duas imagens, surge uma pergunta: onde se deve procurar por esta correspondência? Supondo duas câmeras cujos centros de projeção são C_L e

C_R e seus planos de imagem L e R . Considerando que P_L é a projeção de um ponto P em L , este ponto P deve estar na reta definida por C_L e P_L , logo sua projeção em R deve pertencer a reta e_R , esta reta é conhecida como linha epipolar (Figura 10).

A geometria epipolar é a relação entre pontos em uma imagem e sua correspondente linha epipolar na outra imagem. Ela pode ser facilmente computada se as configurações de ambas as câmeras (a posição dos centros de projeção C_L e C_R e dos planos de imagem L e R) são conhecidas em algum sistema de coordenadas globais. Esta situação é conhecida como configuração estéreo completamente calibrada (*full calibration*), que é extremamente rígida e exige bastante precisão na construção do aparato onde ficam as câmeras (Figura 11).

Figura 10 - Ilustração de restrição epipolar. Neste exemplo, o ponto P que gera P_L em L deve pertencer à reta definida por C_L e P_L , sendo que a projeção de P em R deve pertencer à reta e_R definida como linha epipolar (SCHARSTEIN, 1999).

Uma situação particular de geometria epipolar simples resulta do uso de duas câmeras idênticas e paralelas, onde os planos de imagem coincidem e os

eixos X_s são paralelos à linha que liga seus centros de projeção. Neste caso, linhas epipolares correspondentes são horizontais e têm a mesma coordenada y . Trata-se da situação ilustrada na Figura 9.

Figura 11 - Montagem Estéreo com duas câmeras idênticas e paralelas (SCHARSTEIN, 1999).

Na prática, a restrição epipolar garante que um determinado ponto na imagem da esquerda se encontre na mesma linha na imagem da direita, mudando somente a coluna, reduzindo o problema a uma busca 1D. Ela é amplamente utilizada nos algoritmos de visão estéreo (SCHARSTEIN; SZELISKI, 2001), pois apesar de necessitar de um aparato e uma montagem mais sofisticados permite focar os esforços de pesquisa e desenvolvimento na melhoria da qualidade ou do desempenho dos algoritmos.

A calibração completa é extremamente rígida, tanto na exigência de câmeras precisas, quanto no ajuste e escolha dos parâmetros extrínsecos (disposição física: posição e orientação) e intrínsecos (características ópticas e geometria interna da câmera). Isto causa complexidade e onerosidade aos sistemas de reconstrução 3D. Por exemplo, somente uma câmera digital métrica (seus parâmetros internos são conhecidos e estáveis) custa atualmente (2008) cerca de US\$ 3.000,00.

É importante ressaltar que com o aumento da resolução, os ajustes físicos e ópticos das câmeras se tornam ainda mais delicados, exigindo maior precisão, e, por consequência, resultando em equipamentos ainda mais onerosos.

Não utilizar a restrição epipolar exige do algoritmo uma busca pela correspondência em uma região bidimensional (2D) e não mais em uma linha (1D), o que onera o desempenho do algoritmo. Dependendo da aplicação do algoritmo, este pode ser um fator crítico.

2.6 Abordagens Tradicionais para Visão Estéreo

Em linhas gerais, os algoritmos de visão estéreo podem ser divididos em dois grupos principais: globais e locais. Um método recente, conhecido como Semi-Global (HIRSCHMÜLLER, 2005), (HIRSCHMÜLLER, 2008), mistura algumas características de ambos os métodos.

Antes de apresentar os métodos globais e locais, é necessário apresentar o conceito de *Disparity Space Image* (DSI) [YANG93], (BOBICK; INTILLE, 1999), (SUN, 2002), que é um volume (x, y, d) , onde x é a largura da imagem de entrada, y é a altura e d é a quantidade de níveis de disparidade (Figura 12).

Figura 12 - *Disparity Space Image* (DSI).

Cada plano corresponde à diferença quadrática (ou absoluta), pixel a pixel entre a imagem da esquerda e a imagem da direita, sendo que a imagem da direita é deslocada de 0 (zero) até d . Para cada deslocamento da imagem da direita, o resultado é armazenado em um plano diferente, logo cada plano representa uma disparidade.

A Figura 13 ilustra como é a representação visual em tons de cinza do conteúdo de três planos do DSI. Nota-se na Figura 13d o fundo mais escuro (menor diferença quadrática) indicando que ali houve uma melhor correspondência para a disparidade 10. Já na Figura 13c, vê-se a região escura em uma profundidade intermediária (disparidade 16). Na Figura 13b, a região escura se encontra à frente (objetos próximos) e a disparidade é de 21 (maior que a disparidade da Figura 13c $d=16$ e da Figura 13d $d=10$), demonstrando a relação: quanto mais próximo o objeto maior a disparidade.

Figura 13 - (a) Par estéreo de entrada (SCHARSTEIN; SZELISKI, 2002); (b)(c)(d) representação em tons de cinza da diferença quadrática entre o par estéreo, para as disparidades 21, 16 e 10, respectivamente.

Fazendo um corte no DSI na posição indicada na Figura 14a e visualizando por cima (planta baixa), obtém-se a imagem da Figura 14b.

Figura 14 - (a) Representação do DSI indicando um corte no $y = 151$; (b) fatia (x,d) do DSI para $y = 151$. Fonte das imagens individuais: (SCHARSTEIN; SZELISKI, 2001).

Um dos primeiros métodos que pode ser classificado como global foi criado por Marroquin, Mitter e Poggio (MARROQUIN; MITTER; POGGIO, 1987).

Os métodos globais visam encontrar a disparidade que minimiza a energia, isto é, traçam o percurso da esquerda para a direita no plano (x,d) (para um determinado y do DSI) percorrendo as regiões mais escuras, gerando um perfil como o apresentado na Figura 15b.

Figura 15 - (a) Corte em um DSI (SCHARSTEIN; SZELISKI, 2001); (b) indicação de um caminho que minimiza a energia.

Os métodos globais em geral produzem mapas de disparidade mais precisos se comparados com métodos locais, porém o custo desta qualidade é o tempo de processamento. O mapa de disparidades da Figura 16b foi obtido utilizando um método local tradicional e foi produzido em 1,1s. O mapa de disparidades da Figura 16c foi obtido com um método global (*graph cuts* (BOYKOV; VEKSLER; ZABIH, 1999), [VEKSLER99]) em 662s¹. A Figura 16a apresenta um mapa de disparidades ideal. A precisão dos métodos é ilustrada na Figura 16d (método local) e Figura 16e (método global), onde a cor branca indica pixels iguais aos do mapa de disparidades ideal.

Figura 16 - (a) Mapa de disparidades ideal; (b) mapa de disparidades método local; (c) mapa de disparidades método global (*graph cuts*); (d) relação² entre (a) e (b) onde a cor branca indica pixels iguais e a cor preta indica pixels diferentes; (e) relação entre (a) e (c); As figuras (a)(b)(c) foram extraídas de (SCHARSTEIN; SZELISKI, 2001). Obs.: Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

Os métodos locais foram os primeiros a serem desenvolvidos (HANNAH, 1974). A partir de um ponto na imagem da esquerda se procura o correspondente na imagem da direita. Uma forma de medir esta

¹ Estes resultados são apresentados num relevante trabalho de taxonomia desenvolvido por Daniel Scharstein e Richard Szeliski [SCHARSTEIN01].

² A relação entre as imagens é calculada da seguinte forma:

- Diferença entre as duas imagens (pixels iguais = 0, preto; pixels diferentes qualquer cor).
- Aplicado *threshold*, onde 0 fica com preto e acima de 0 fica como branco
- Inversão das cores para indicar com branco a igualdade e em preto a diferença. Este passo é realizado para facilitar a impressão das imagens, pois há uma tendência em ficar mais branco que preto, sendo necessário somente imprimir os pontos de erro.

correspondência é subtrair os valores das cores (R: vermelho, G: verde, B: azul) de um ponto na imagem da esquerda de um ponto na imagem da direita e, em seguida, realizar a soma dos módulos. Este processo é feito sucessivamente para as possíveis disparidades, sendo que a de menor resultado é considerada a melhor correspondência. Comparar somente um ponto com outro seria uma abordagem muito simplista, que resultaria em muitos erros, entretanto uma forma de contornar estes erros é realizar a comparação não só de um ponto, mas de uma região vizinha ao ponto, algo como uma janela 3x3 (ou 5x5 ou mesmo 7x7) centrada no ponto em questão. A Figura 17 apresenta um exemplo que considera o uso da restrição epipolar. Note que a procura do ponto (247,133) da imagem da esquerda começa exatamente nesta posição na imagem da direita e desloca-se à esquerda partindo do $x=247$ e indo até o $x=231$.

Figura 17 - Ponto em questão e sua janela 3x3 na imagem da esquerda, e a linha epipolar na imagem da direita, onde é procurada a correspondência. Par estéreo original publicado em (SCHARSTEIN; SZELISKI, 2002).

Subtraem-se as cores de cada pixel da janela (no exemplo, 3x3), soma-se o módulo de cada resultado e, em seguida, soma-se o resultado para cada

pixel, este resultado é armazenado e o processo é feito na linha epipolar na imagem da direita. Entre os resultados armazenados, aquele que apresentar o menor valor será considerado a melhor correspondência. Esta abordagem é conhecida como soma das diferenças absolutas ou SAD (*Sum of Absolute Differences*).

Sendo **E** a matriz (fixa) com os valores das cores da janela da esquerda e **D** a matriz (móvel) com os valores das cores da janela da direita. Na Figura 18 é apresentada a distribuição destas variáveis.

E1	E2	E3
E4	E5	E6
E7	E8	E9

D1	D2	D3
D4	D5	D6
D7	D8	D9

Figura 18 - Descrição das variáveis para a janela da esquerda e da direita.

Tem-se a seqüência de cálculos que segue:

$$\text{Disp1} = |\text{E1.vermelho} - \text{D1.vermelho}| + |\text{E1.verde} - \text{D1.verde}| + |\text{E1.azul} - \text{D1.azul}|$$

$$\text{Disp2} = |\text{E2.vermelho} - \text{D2.vermelho}| + |\text{E2.verde} - \text{D2.verde}| + |\text{E2.azul} - \text{D2.azul}|$$

$$\text{Disp3} = |\text{E3.vermelho} - \text{D3.vermelho}| + |\text{E3.verde} - \text{D3.verde}| + |\text{E3.azul} - \text{D3.azul}|$$

$$\text{Disp4} = |\text{E4.vermelho} - \text{D4.vermelho}| + |\text{E4.verde} - \text{D4.verde}| + |\text{E4.azul} - \text{D4.azul}|$$

$$\text{Disp5} = |\text{E5.vermelho} - \text{D5.vermelho}| + |\text{E5.verde} - \text{D5.verde}| + |\text{E5.azul} - \text{D5.azul}|$$

$$\text{Disp6} = |\text{E6.vermelho} - \text{D6.vermelho}| + |\text{E6.verde} - \text{D6.verde}| + |\text{E6.azul} - \text{D6.azul}|$$

$$\text{Disp7} = |\text{E7.vermelho} - \text{D7.vermelho}| + |\text{E7.verde} - \text{D7.verde}| + |\text{E7.azul} - \text{D7.azul}|$$

$$\text{Disp8} = |\text{E8.vermelho} - \text{D8.vermelho}| + |\text{E8.verde} - \text{D8.verde}| + |\text{E8.azul} - \text{D8.azul}|$$

$$\text{Disp9} = |\text{E9.vermelho} - \text{D9.vermelho}| + |\text{E9.verde} - \text{D9.verde}| + |\text{E9.azul} - \text{D9.azul}|$$

$$\text{Disp} = \text{Disp1} + \text{Disp2} + \text{Disp3} + \text{Disp4} + \text{Disp5} + \text{Disp6} + \text{Disp7} + \text{Disp8} + \text{Disp9} \quad (5)$$

Sendo **En** o pixel na imagem da esquerda, **Dn** o pixel na imagem da direita, **Disp_n** a disparidade para aquele pixel e, por último, **Disp** como a disparidade daquela janela 3x3 em questão. A **Disp** é calculada para várias posições na imagem da direita, estas posições são conhecidas como níveis de disparidade, sendo a última posição conhecida como disparidade máxima, ou seja, o máximo de deslocamento horizontal que um objeto pode ter.

Outra maneira de fazer a procura por correspondência é subtrair as cores de cada pixel da janela (no exemplo, 3x3), somar o quadrado de cada resultado (ao invés do módulo) e, em seguida, utilizar o mesmo processo descrito anteriormente. Esta abordagem é conhecida como soma dos quadrados das diferenças ou SSD (*Sum of Squared Differences*). É notada nas diversas publicações, uma maior utilização da SSD em vez da SAD, porém algoritmos de visão estéreo que trabalham em tempo real (KANADE et al., 1996), (KIMURA et al., 1999), (HIRSCHMÜLLER; INNOCENT; GARIBALDI, 2002) utilizam a SAD, principalmente porque a elevação ao quadrado utilizada na SSD é mais onerosa que a escolha do resultado positivo na SAD. Por outro lado, os resultados gerados na SAD permitem o uso de variáveis menores.

Tem-se a seqüência de cálculos que segue:

$$\text{Disp1}=(\text{E1.vermelho}-\text{D1.vermelho})^2+(\text{E1.verde}-\text{D1.verde})^2+(\text{E1.azul}-\text{D1.azul})^2$$

$$\text{Disp2}=(\text{E2.vermelho}-\text{D2.vermelho})^2+(\text{E2.verde}-\text{D2.verde})^2+(\text{E2.azul}-\text{D2.azul})^2$$

$$\text{Disp3}=(\text{E3.vermelho}-\text{D3.vermelho})^2+(\text{E3.verde}-\text{D3.verde})^2+(\text{E3.azul}-\text{D3.azul})^2$$

$$\text{Disp4}=(\text{E4.vermelho}-\text{D4.vermelho})^2+(\text{E4.verde}-\text{D4.verde})^2+(\text{E4.azul}-\text{D4.azul})^2$$

$$\text{Disp5}=(\text{E5.vermelho}-\text{D5.vermelho})^2+(\text{E5.verde}-\text{D5.verde})^2+(\text{E5.azul}-\text{D5.azul})^2$$

$$\text{Disp6}=(\text{E6.vermelho}-\text{D6.vermelho})^2+(\text{E6.verde}-\text{D6.verde})^2+(\text{E6.azul}-\text{D6.azul})^2$$

$$\text{Disp7}=(\text{E7.vermelho}-\text{D7.vermelho})^2+(\text{E7.verde}-\text{D7.verde})^2+(\text{E7.azul}-\text{D7.azul})^2$$

$$\text{Disp8}=(\text{E8.vermelho}-\text{D8.vermelho})^2+(\text{E8.verde}-\text{D8.verde})^2+(\text{E8.azul}-\text{D8.azul})^2$$

$$\text{Disp9}=(\text{E9.vermelho}-\text{D9.vermelho})^2+(\text{E9.verde}-\text{D9.verde})^2+(\text{E9.azul}-\text{D9.azul})^2$$

$$\text{Disp}=\text{Disp1}+\text{Disp2}+\text{Disp3}+\text{Disp4}+\text{Disp5}+\text{Disp6}+\text{Disp7}+\text{Disp8}+\text{Disp9} \quad (6)$$

Sendo **En** o pixel na imagem da esquerda, **Dn** o pixel na imagem da direita, **Dispn** a disparidade para aquele pixel e **Disp** a disparidade da janela 3x3 em questão, uma vez encontrada a melhor correspondência, é medida a disparidade. Por exemplo, se o ponto em questão na imagem da esquerda se encontra nas coordenadas (22,30) e o seu correspondente na imagem da direita está em (12,30) significa que a disparidade é de 10 pixels.

Para a criação de um mapa de disparidades, este processo é realizado em toda a imagem. Uma vez obtidas as disparidades, basta associar tons de cinza a cada uma, sendo que a máxima terá tom mais claro (objeto próximo) e a mínima terá tom mais escuro (objeto distante).

2.6.1 Uso do DSI em Métodos Locais

Atualmente, o DSI é utilizado tanto em métodos globais quanto em locais. Nestes, o DSI tem um aspecto importante na eliminação de cálculos repetitivos, garantindo um excelente desempenho, porém a desvantagem é o elevado consumo de memória.

É possível codificar um método local com alguns laços “for” sem armazenar resultados intermediários, porém diversos cálculos serão repetidos muitas vezes. Suponha a análise do ponto (22,30) na imagem da esquerda e que se trabalha com uma janela 3x3. Logo, os cálculos de correspondência (SAD ou SSD ou qualquer outro) serão realizados nove vezes o número máximo de disparidades esperadas na imagem da direita, ou seja, se o valor máximo de disparidade esperado for de 10 pixels, serão efetuados 90 vezes estes cálculos. Ao passar para o próximo pixel na imagem da esquerda (23,30), todos os 90 cálculos se repetem, sendo que pelo menos 60 já haviam sido calculados diretamente na operação anterior (Figura 19).

Figura 19 - Cálculos repetidos com o deslocamento da janela da imagem da esquerda.

A maneira adotada para aperfeiçoar este processo é armazenar resultados parciais dos cálculos e reaproveitá-los nas diversas etapas da procura por correspondências. Conseqüentemente, há um uso maior de memória, mas com um ganho bastante significativo de desempenho em tempo de execução.

A seqüência de processamento comumente adotada (SCHARSTEIN, 1999), (SCHARSTEIN; SZELISKI, 2001) é:

- custo de correspondência – cálculo da diferença quadrática (ou outra), é nesta etapa que o DSI é criado e carregado;
- agregação de evidências – soma de todas as diferenças para uma determinada janela;
- seleção de disparidades - escolha do melhor resultado.

Para ilustrar esta abordagem num exemplo de uso, considera-se uma situação a seguir em que o valor máximo de disparidade (d) seja de 10 pixels e o par de imagens em questão possua uma resolução x=320 e y=200, cada.

a) Custo de Correspondência

É criado o DSI (*Disparity Space Image*) com 10 planos com 320x200 posições, criando um espaço x,y,d (Figura 20) e realiza-se o processo de cálculo de correspondência (SAD, SSD ou qualquer outro) para cada pixel (sem somar à janela) para cada disparidade, isto é, compara-se o pixel (60,50) da imagem da esquerda com o pixel (60,50) da imagem da direita e armazena-se o resultado na posição (60,50) do plano 0 (equivale a disparidade 0). Em seguida, compara-se o pixel (60,50) da imagem da esquerda com o pixel (59,50) da imagem da direita e armazena-se o resultado na posição (60,50) do plano 1 (equivale à disparidade 1), e assim por diante. Esta etapa ocorre tanto em métodos locais quanto em globais. Este processo é conhecido como “*matching cost*”, aqui traduzido como custo de correspondência.

Figura 20 - Planos que armazenam os custos de correspondência.

b) Agregação de Evidências

Em seguida, são alocados mais 10 planos com 320x200 posições onde serão registrados os resultados das somas das janelas. Trata-se da adição (considerando uma janela quadrada 3x3) dos nove resultados de cada plano de custo de correspondência na posição central da janela deste novo plano. Por exemplo, para um plano de custo de correspondência, na posição (60,70) somam-se os resultados que estão nas posições indicadas na Figura 21.

(59,69)	(60,69)	(61,69)
(59,70)	(60,70)	(61,70)
(59,71)	(60,71)	(61,71)

Figura 21 - Posições envolvidas durante o processo de agregação para o ponto (60,70).

É armazenado o resultado desta soma na posição (60,70) do plano 3, recém alocado.

Este processo é conhecido como agregação de evidências (nesta descrição, trabalhou-se com janela quadrada).

Dentro do laço “for”, que faz esta varredura, costuma-se realizar a soma em 3 colunas separadas. Neste exemplo, as colunas são 59, 60 e 61; quando o laço “for” avançar da posição (60,70) para a posição (61,70) é subtraída a coluna 59 e acrescentada a coluna 62. Esta técnica é amplamente utilizada visando a melhoria de desempenho e é conhecida como *box filtering* (MCDONNELL, 1981). Originalmente foi desenvolvida para uso em filtros de imagens, de onde surgiu o nome, posteriormente, passou a ser empregada em visão estéreo no processo de agregação.

Em vez de alocar mais 10 planos, pode-se alocar somente 1 plano que armazena temporariamente o resultado da agregação. Quando o processo termina de preencher o plano inteiro, ele sobrepõe o plano de custo de correspondência.

Além da janela quadrada tradicional, existem outras abordagens. Uma muito citada, adapta o tamanho da janela à região em questão. Esta abordagem é conhecida como janela adaptativa (KANADE; OKUTOMI, 1994). Outra forma de trabalho é conhecida como *shiftable window* (ARNOLD, 1983), a qual desloca a janela, mantendo-a ancorada em diferentes pontos da janela principal.

c) Seleção de disparidades

Por último, é criado mais um plano com 320x200. Nele é armazenado, em cada posição, o valor da disparidade que obteve a melhor correspondência, isto é, o menor valor. Por exemplo, se para o ponto (40,50) tem-se nos planos de agregação de evidências os resultados da Figura 22, o menor resultado encontra-se no plano 6, o que significa que a melhor correspondência foi encontrada com disparidade de 6 pixels.

Plano	0	1	2	3	4	5	6	7	8	9
Valor	101	232	184	609	900	143	7	240	78	93

Figura 22 - Simulação de valores nos diversos planos para a posição (40,50).

Então, é armazenado na posição (40,50) o valor 6. Este processo é realizado para os 320x200 pontos e é conhecido como seleção de disparidades. A partir deste plano, é criada uma imagem em escalas de tons de cinza, onde para o exemplo com disparidade máxima de 10 pixels o 0 seria (0,0,0) (preto em RGB) e o 9 seria (255,255,255) (branco em RGB) e os demais valores os níveis intermediários distribuídos uniformemente.

Escolher o menor número como melhor correspondência é conhecido como “*winner-take-all*” (WTA), ou aqui traduzido como “vencedor-leva-tudo”. A desvantagem desta abordagem é que não garante a unicidade da correspondência, isto é, um ponto da imagem da direita pode ser considerado uma boa correspondência por vários pontos da esquerda.

Eventualmente, se faz necessária alguma preparação do par de imagens antes destes processamentos, ou seja, uma etapa de pré-processamento. Caso se deseje obter uma resolução maior no mapa de disparidades, inclui-se uma etapa posterior à seleção de disparidades onde são interpolados pixels

entre os já existentes. Este novo arranjo (SCHARSTEIN, 1999), (SCHARSTEIN; SZELISKI, 2001) no algoritmo de visão estéreo, compreende:

- pré-processamento (opcional),
- custo de correspondência,
- agregação de evidências,
- seleção de disparidades,
- estimação de disparidade sub-pixel (opcional).

Métodos locais podem ser codificados de maneira bastante eficiente conseguindo inclusive atuar em tempo real (KANADE et al., 1996), (KIMURA et al., 1999), (HIRSCHMÜLLER; INNOCENT; GARIBALDI, 2002). Para conseguir tal desempenho são utilizadas algumas otimizações, dentre elas a que merece destaque é a técnica de *box filtering* (MCDONNELL, 1981).

2.7 Comparação de Requisitos

Agora que os detalhes sobre visão estéreo foram vistos, é possível analisar suas aplicações e requisitos, em especial, a deste trabalho: reconstrução 3D de baixo custo. A Tabela 1 apresenta um comparativo entre os requisitos de diferentes aplicações.

Tabela 1 - Comparação de requisitos entre aplicações para visão estéreo, Dados extraídos de (SCHARSTEIN, 1999), exceto reconstrução 3D de baixo custo.

	Cartografia	Navegação	Reconhecimento	Controle Baseado em Visão	Síntese Visual	Reconstrução 3D	Reconstrução 3D de baixo custo
Entrada							
Limitada	Sim (a)	Não	Não	Sim (b)	Não	Não	Não
Sempre com Textura	Sim	Sim (c)	Não	Não	Não	Não	Não
Possui Oclusão	Não (d)	Sim	Sim	Sim	Sim	Sim	Sim
Requisitos							
Saída Densa	Sim	Não	Não (e)	Não	Sim	Sim	Sim
Trata Oclusão	Não	Não	Sim	Não	Sim	Sim	Não
Calibração Completa	Sim	Não (f)	Sim	Sim	Não (g)	Sim	Não
Precisão na Profundidade	Sim	Não	Sim	Sim	Não	Sim	Não (h)
Geometria correta	Sim	Sim	Sim	Sim	Não	Sim	Não
Tempo Real	Não	Sim	Não	Sim	Não	Não	Não

(a) Imagens aéreas.

(b) Normalmente em ambiente de laboratório.

(c) Em ambiente fechado existem regiões sem textura.

(d) Exceto a oclusão causada por pontes, viadutos e prédios altos.

(e) Depende da abordagem.

(f) Exceto para permitir a projeção da direção de pilotagem em uma imagem.

(g) Exceto quando é feita a estimativa dos parâmetros visuais de referência.

(h) Depende da precisão na obtenção das imagens estéreo.

Segue abaixo, o detalhamento de cada item, bem como a explicação da resposta para a reconstrução 3D de baixo custo:

ENTRADA:

Limitada - significa que as imagens de entrada têm características conhecidas.

(Não) - O objetivo é trabalhar com imagens Lambertianas¹ de qualquer ambiente.

Sempre com textura - significa que normalmente para aquela aplicação as fotos são ricas em detalhes (de uma forma geral, imagens da natureza têm esta característica).

(Não) – Não se espera que todos os objetos tenham textura, cabendo sempre a possibilidade existir objetos com superfícies homogêneas.

¹ Superfícies cuja refletância não possui variação significativa com a variação do ponto de vista.

Possui oclusão - sempre ocorre oclusão em fotos de objetos próximos.

(Sim) – Ambientes livres sempre tem oclusão, por outro lado, o uso de visão estéreo com somente duas imagens não trata a oclusão.

REQUISITOS:

Saída Densa - necessita que o mapa de disparidades seja denso, isto é, obtido com técnicas baseadas na área (enquanto aplicações que não possuem esta exigência podem utilizar técnicas baseadas na característica).

(Sim) – Como o resultado da aplicação é uma malha 3D do ambiente fotografado, não há espaço para apresentar somente algumas características da imagem.

Trata Oclusão - a aplicação exige que a oclusão causada pela paralaxe seja resolvida.

(Não) – Para tratar a oclusão, seria necessária uma abordagem com mais de um par estéreo (KANADE et al., 1996), (KIMURA et al., 1999), que não produz uma solução de baixo custo, logo não é o objetivo deste trabalho.

Calibração completa - exige rigidez na calibragem das câmeras.

(Não) – A calibração completa introduz muito rigor ao sistema, como a finalidade deste trabalho é flexibilidade, não é adotada a calibração completa.

Precisão na Profundidade - exigência de que os dados de profundidade em todo o mapa de disparidades sejam precisos.

(Não) – Por não usar calibração completa e não exigir equipamentos precisos, o nível de precisão será variável, dependendo da precisão no processo de obtenção das imagens estéreo.

Geometria Correta - equivale à precisão na profundidade, mas considera todas as coordenadas.

(Não) – Devido ao uso de câmeras convencionais e calibração flexível, não há garantias de que a geometria esteja completamente correta.

Tempo Real – o processamento do mapa de disparidades em tempo real somente é necessário na navegação e no controle baseado na visão (*visual servoing*).

(Não) – Como o objetivo é apresentar um sistema de reconstrução 3D de baixo custo, não há exigência de alto desempenho.

2.8 Trabalhos Correlatos

Existem diversas pesquisas que relacionam a visão estéreo com a reconstrução 3D. De fato, o produto normalmente utilizado pelas técnicas de visão estéreo, o mapa de disparidades, é uma representação tridimensional do ambiente, sendo que a profundidade é apresentada na forma de tons de cinza.

Um trabalho que merece destaque foi desenvolvido por Daniel Scharstein no livro *View Synthesis Using Stereo Vision* (SCHARSTEIN, 1999) (desenvolvido a partir de sua tese de doutorado) e aborda a utilização de visão estéreo em síntese visual.

A síntese visual busca produzir uma imagem sintética (artificial) em um determinado ângulo a partir de outras imagens reais em ângulos diferentes. As técnicas comumente utilizadas procuram fazer uma torção (*“warping”*) das imagens reais produzindo uma nova, simulando o ângulo desejado. A proposta de Daniel foi aperfeiçoar a síntese visual, com as informações tridimensionais obtidas no mapa de disparidades produzido pelas técnicas de visão estéreo, produzindo assim, imagens sintéticas mais realistas.

O trabalho de Daniel Scharstein está relacionado com este trabalho por se tratar de uma pesquisa muito profunda em visão estéreo com o objetivo de relacioná-la com outra área, no caso dele, a síntese visual. Esta pesquisa foi inspiradora para a confecção desta dissertação, pois ambas focam alternativas para modificar as técnicas tradicionais e não simplesmente utilizá-las.

Por outro lado, um software de uso gratuito (versão executável apenas), o Looxis Faceworx® (<http://www.looxis.com/>) realiza especificamente a modelagem de cabeças a partir de duas fotos (uma frontal e outra lateral). Ele demanda do usuário a colocação manual de pontos de referência do rosto e dos contornos para a reconstrução 3D. O modelo 3D gerado da face pode ser enviado para a Looxis a fim de produzir um bloco de cristal com este modelo 3D gravado internamente. O objetivo da disponibilização gratuita do software é justamente promover a venda destes blocos de cristal. Este software está relacionado com este trabalho, não pelas técnicas utilizadas, mas sim pelo fato de ser um sistema que a partir de fotografias produz a reconstrução 3D.

2.9 Conclusão

Neste capítulo, foram apresentados os detalhes geométricos das principais técnicas de visão estéreo foram apresentados os métodos globais e locais. Foi aprofundada a compreensão do funcionamento dos métodos locais. Foi também apresentado um comparativo entre diferentes aplicações de visão estéreo e foram destacados os principais trabalhos correlatos.

Neste capítulo, as técnicas selecionadas são descritas e avaliadas frente aos objetivos deste trabalho.

A visão estéreo é uma escolha funcional, pois pode ser automatizada. Sua perspectiva de continuar sendo um tópico bastante pesquisado é forte, pois na natureza os sistemas visuais estéreo são muito bem sucedidos, o que indica que este é um caminho promissor. A reconstrução 3D usando visão estéreo apresenta a maior viabilidade para atingir os objetivos deste trabalho.

Como se apresentou no CAPÍTULO 2, os métodos locais são bastante rápidos. Tendo em vista que a intenção deste trabalho é a utilização de equipamentos de baixo custo, é adequado que não sejam exigidos computadores com capacidade de processamento elevada, logo a utilização de métodos locais se faz necessária. Como a aplicação deste trabalho é a reconstrução 3D, se faz necessário o mapa de disparidades denso.

A Figura 23 ilustra o caminho percorrido durante as pesquisas e o posterior desenvolvimento deste trabalho, visando o objetivo da reconstrução 3D de baixo custo, funcional, flexível e baseada em pesquisas com boas perspectivas futuras.

Figura 23 - Síntese das escolhas e desafios deste trabalho.

Existem ainda aspectos dentro das escolhas realizadas que precisam ser tratados e resolvidos: consumo de memória, calibração rígida e precisão nas cores.

Existem ainda decisões a cerca das três etapas principais:

- custo de correspondência,
- agregação de evidências,
- seleção de disparidades.

Para o custo de correspondência, existem muitas aplicações publicadas que utilizam a SSD, por outro lado aplicações em tempo real utilizam SAD com hardware especializado (KANADE et al., 1996) (KIMURA et al., 1999) dificultando a comparação de resultados. As técnicas que utilizam SSD possuem ótimas referências de tempo publicadas (SCHARSTEIN; SZELISKI, 2001), inclusive com o código fonte à disposição na página do Middlebury

College (<http://vision.middlebury.edu/stereo/>). Visando a produção de um trabalho integrado aos dados de diversas publicações, foi escolhida a SSD para a etapa de custo de correspondência.

Para a agregação de evidências, foi escolhido trabalhar com a janela quadrada fixa, ao invés de janelas variáveis (como a janela adaptativa), para codificar a técnica de *box filtering* visando obter o melhor desempenho possível. As técnicas com janela variável privilegiam a qualidade em detrimento do desempenho de tempo de execução.

Na etapa de seleção de disparidades, a técnica do vencedor-leva-tudo (*winner-take-all*: WTA) foi escolhida, pois ela é simples, amplamente utilizada e não onera o desempenho.

Então, tem-se:

- custo de correspondência – SSD,
- agregação de evidências – janela quadrada e uso de *box filtering*,
- seleção de disparidades – WTA.

Uma vez definidos todos os algoritmos envolvidos no processo de procura por correspondências em visão estéreo, foi desenvolvido o código em C/C++ (material para estudo destas linguagens pode ser encontrado em: (KERNIGHAN; RITCHIE, 1990), (PUGH, 1990) e (STROUSTRUP, 1991)), visando produzir o mapa de disparidades denso (base para a reconstrução 3D).

Foram utilizadas imagens padrão da literatura (Figura 24). Estas imagens obedecem à restrição epipolar e seus histogramas estão normalizados.

Figura 24 - Pares estéreo: no alto, imagem conhecida como Venus, e abaixo, Tsukuba (SCHARSTEIN; SZELISKI, 2002).

Os resultados do código desenvolvido neste trabalho foram muito similares àqueles obtidos pelo código disponível no site do Middlebury College (este foi o código utilizado no trabalho de taxonomia disponível em (SCHARSTEIN; SZELISKI, 2002)), ambos executados no mesmo equipamento. Na Figura 25 são apresentados os mapas de disparidade de ambos os códigos e um comparativo entre eles (Figura 25c e Figura 25f). São destacadas na cor preta as diferenças que estão concentradas nas bordas, pois no código desenvolvido neste trabalho foi adotado o critério de não calcular a disparidade quando não houver todos os níveis de disparidade disponíveis (esta situação ocorre somente na lateral esquerda). Também foi adotado não calcular a disparidade se a janela não puder ser preenchida integralmente, o que ocorre nas demais bordas. Também existem alguns pixels (pretos) dispersos no meio da imagem (Figura 25c e Figura 25f), os quais indicam diferenças, causadas por diferentes abordagens na decisão de empate no processo de seleção de

disparidades (WTA), mas não representam diferença significativa. Para a imagem Tsukuba (384x288) tem-se (descontando a borda) 42 pixels diferentes (0,038%) e para a imagem Venus (434x383) tem-se (descontando a borda) 53 pixels diferentes (0,032%).

Figura 25 - (a) Mapa de disparidades da imagem Venus obtido com o código do Middlebury College; (b) mapa de disparidades da imagem Venus obtido com o código desenvolvido para este trabalho; (c) relação entre (a) e (b), sendo que os pixels brancos indicam pixels iguais na comparação e os pretos as diferenças; (d) mapa de disparidades da imagem Tsukuba obtido com o código do Middlebury College; (e) mapa de disparidades da imagem Tsukuba obtido com o código desenvolvido para este trabalho; (f) Relação entre (d) e (e).

O tempo de processamento das imagens para ambos os códigos é apresentado no gráfico da Figura 26. O código implementado foi 21% mais rápido para a imagem Tsukuba e 15% mais rápido para a imagem Vênus. Isto se deve ao fato do código do Middlebury College estar preparado para executar diversos algoritmos de visão estéreo, o que introduz código de controle. Por outro lado, o código implementado aqui é muito especializado e, conseqüentemente, tem melhor desempenho em tempo de execução.

Figura 26 - Tempo de processamento de ambos os códigos.

3.1 Consumo de Memória

Durante o desenvolvimento do código, é notável o excessivo consumo de memória para o DSI, pois a alocação é feita com variáveis ligadas diretamente ao tamanho da imagem e do número de disparidades (conforme evidenciado no CAPÍTULO 1 em Consumo de Memória). Esta situação merece uma proteção, para evitar que o usuário utilize uma imagem com resolução que necessite de mais memória do que está disponível no equipamento.

3.2 Calibração Rígida

Considerando que o objetivo do trabalho é permitir certo grau de flexibilidade na calibração das câmeras, foram simuladas pequenas distorções (comuns num ambiente não controlado) com o objetivo de medir a sensibilidade do algoritmo. A Figura 27 ilustra as distorções consideradas.

Figura 27 - Distorções simuladas nas imagens de entrada.

Os resultados da procura 1D (tradicional) não foram satisfatórios, (Figura 28b, c e d últimas imagens à direita) comparando com o mapa de disparidades gerado a partir das imagens originais. Nota-se que há predominância da cor preta, indicando muita diferença. De fato, isto já era esperado, uma vez que o processo de procura por correspondências ocorre somente na linha epipolar e, sem dúvida, a correspondência não estará nesta linha, devido à distorção introduzida.

Figura 28 - (a) Par estéreo original da imagem Vênus (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades; (b) deslocamento de linha na imagem da direita, o mapa de disparidades gerado e relação deste mapa de disparidades com o original de (a) onde a cor preta indica pixels diferentes; (c) rotação na imagem da direita, seu mapa de disparidades e a relação com a imagem (a); (d) curvatura na imagem da direita, seu mapa de disparidade e a relação com a imagem(a)¹.

¹ Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

3.3 Precisão nas Cores

Ainda simulando outra situação esperada, existe o caso onde as imagens de entrada não têm cores exatamente iguais. Para ilustrar esta distorção, foi modificado o brilho de uma das imagens de entrada e realizado o processo de procura por correspondências, a Figura 29b apresenta um mapa de disparidades bastante distorcido, sua comparação (à direita) com o mapa de disparidades original destaca esta distorção com pixels na cor preta.

Figura 29 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades; (b) brilho modificado na imagem da direita, mapa de disparidades gerado e à direita comparação entre os dois mapas de disparidade, onde a cor preta indica pixels diferentes¹.

Nota-se assim, a não disposição das técnicas tradicionais para tratar este tipo de distorção.

¹ Foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

3.4 Conclusão

Foi demonstrado com estes testes que mudanças significativas na abordagem tradicional de visão estéreo são necessárias, e que os três maiores desafios são: redução do consumo de memória, calibração flexível e redução da sensibilidade à variação das cores.

CAPÍTULO 4 Soluções Propostas e Resultados Obtidos

Este capítulo apresenta as soluções propostas para os desafios descritos no CAPÍTULO 3. Como existem desafios específicos, seus resultados serão apresentados em cada subseção para propiciar maior clareza.

4.1 Protótipo de Sistema

Além da questão conceitual, é realizado também o desenvolvimento do sistema de reconstrução 3D, que possui diversos detalhes funcionais tais como:

- reconstrução 3D usando visão estéreo (tradicional);
- reconstrução 3D usando visão estéreo (técnica das faixas);
- geração de arquivo VRML com a malha 3D reconstruída;
- produção do mapa de disparidades na tela e *bitmap*;
- navegação 3D, visualização imediata do ambiente reconstruído;
- ajuste de parâmetros na ferramenta e armazenamento dos últimos parâmetros utilizados;
- reconstrução 3D a partir do mapa de disparidades: permite manipular o mapa de disparidades e carregá-lo como fonte de informações;
- procura por correspondências usando as intensidades das cores ou imagem processada com operador gradiente;
- procura por correspondências em 2D e 1D ao em vez de somente 1D;

- redução no consumo de memória para viabilizar o processamento de imagens de alta resolução em computadores comuns.

4.1.1 Delimitação do Escopo

Algoritmos de visão estéreo, normalmente apresentam erros causados por:

- falsas correspondências causadas por imagens com regiões uniformemente coloridas ou mesmo padrões repetitivos;
- correspondências não encontradas, isto é, para uma das câmeras um determinado ponto é visível, mas para a outra este mesmo ponto não está visível (oclusão parcial).

Existem diversas abordagens que procuram minimizar estas situações (KANADE et al., 1996), (KIMURA et al., 1999), mas este não é o objetivo deste trabalho, pois erros como estes não serão completamente tratados, de maneira que precisarão ser corrigidos manualmente no mapa de disparidades ou na malha 3D.

4.1.2 Desenvolvimento

Todo o desenvolvimento de código neste trabalho foi orientado a não exigir a aquisição de software algum (além do Sistema Operacional). Todo o código foi desenvolvido em C/C++ (material para estudo destas linguagens pode ser encontrado em: (KERNIGHAN; RITCHIE, 1990), (PUGH, 1990) e (STROUSTRUP, 1991)) e somente utiliza componentes fornecidos gratuitamente pelo fabricante do Sistema Operacional.

Os módulos que compõe o sistema estão ilustrados na Figura 30.

Figura 30 – (a) Módulos do sistema que foram desenvolvidos; (b) Representação do sistema operacional e alguns de seus módulos.

O sistema desenvolvido permite duas maneiras de trabalho. Uma baseada na carga do par estéreo, gerando mapa de disparidades (tanto na tela quanto *bitmap*) e gerando malha 3D (tanto no visualizador 3D quando em VRML) como ilustra a Figura 31. Outra, baseada na carga do mapa de disparidades e de textura para gerar a malha 3D (tanto no visualizador 3D quando em VRML) como ilustra a Figura 32. Esta segunda opção surgiu do fato de que é mais fácil editar e corrigir um mapa de disparidades do que editar e corrigir a malha 3D.

A Figura 31 e a Figura 32 apresentam a tela do sistema em que está indicada em ordem numérica a seqüência em que devem ser preenchidos os campos e selecionadas as opções.

Figura 31 – Tela do sistema desenvolvido durante o processamento do par de imagens, gerando o mapa de disparidades e a geometria 3D.

Figura 32 – Tela do sistema desenvolvido na opção de reconstrução 3D diretamente do mapa de disparidades.

4.2 Consumo de Memória

A abordagem tradicional de procura por correspondências nos métodos locais como o desenvolvido aqui (SSD/Janela quadrada/WTa) seguem um padrão quanto à alocação e uso da memória:

1. carga das imagens de entrada;
2. alocação do DSI e carga de dados;
3. alocação de um plano para o armazenamento temporário dos resultados da agregação, quando totalmente preenchido sobrepõe o plano correspondente do DSI;
4. alocação da memória para o mapa de disparidades e o seu preenchimento, selecionando a melhor disparidade (melhor plano do DSI) para cada pixel;
5. liberação da memória das imagens de entrada, DSI, plano temporário e do mapa de disparidades.

Como se pode notar, o DSI é preenchido na segunda etapa e utilizado até a seleção de disparidades. Com esta abordagem, o óbvio seria pensar em alguma solução que envolvesse paginação em disco para conseguir a redução do consumo de memória, porém esta não seria uma boa alternativa, uma vez que onera fortemente o desempenho do algoritmo.

Uma alternativa que se mostrou viável foi modificar completamente o comportamento do algoritmo: ao invés de processar em cada etapa toda a imagem, é possível carregar as imagens de entrada, fatiá-las e realizar o processamento somente da faixa selecionada de forma a produzir uma única linha no mapa de disparidades, mas qual seria o tamanho desta faixa?

Analisando de trás para frente, o menor produto de cada rodada poderia ser uma linha no mapa de disparidades. Para construí-la são necessários todos os planos do DSI, mas estes planos não precisam ter a altura da imagem de entrada, basta que tenham a altura da janela quadrada mais 1 (este 1 permite o uso do *box filtering*), estas faixas do DSI seriam preenchidas a partir de faixas da imagem de entrada com a altura da janela na primeira carga e,

posteriormente, acrescentando uma linha abaixo e retirando uma linha acima (como se a faixa estivesse deslizando sobre as imagens de entrada). Esta técnica foi batizada como Técnica das Faixas e é uma contribuição significativa deste trabalho.

4.2.1 Técnica das Faixas

Com a técnica das faixas, as imagens de entrada já carregadas são tratadas em faixas e o processo de procura por correspondências ocorre somente na faixa, estas devem ter o menor tamanho possível, sem deixar de manter os resultados intermediários para reuso durante o processo.

A menor altura da faixa é a própria altura da janela mais 1, pois durante o processo de agregação é necessário ter os resultados de custo de correspondência disponíveis, sendo que uma altura menor não permitiria isso. Outra conclusão é que a faixa deve ter a mesma largura da imagem de entrada, pois larguras menores eliminariam dados úteis para reuso durante o deslocamento lateral da janela.

O processo de custo de correspondência somente trata as faixas das imagens de entrada, preenchendo as **faixas de agregação** (pequeno bloco do DSI) com a altura da janela (uma linha abaixo ficará livre na primeira rodada). Estas são trabalhadas no processo de agregação de evidências produzindo as **linhas de seleção de disparidade**, que são tratadas no processo de seleção de disparidade produzindo uma linha no mapa de disparidades (Figura 33).

Figura 33 - Diagrama da Técnica das Faixas.

Depois da primeira rodada, é realizado o cálculo do custo de correspondência somente em uma linha (imediatamente abaixo) das imagens de entrada, o resultado é armazenado naquela linha que sobrou nas **faixas de agregação**. O processo de agregação de evidências agora ignora a primeira linha das **faixas de agregação**, produzindo as **linhas de seleção de disparidades** e seguindo normalmente o processo de seleção de disparidades. Depois da segunda rodada, são eliminados os dados da primeira linha das **faixas de agregação** e no lugar, são colocados os dados de custo de correspondência da linha imediatamente inferior à faixa nas imagens de entrada, e assim por diante, até o fim da imagem. Este processo reaproveita os dados que já haviam sido calculados nas rodadas anteriores, somente remove os desnecessários e adiciona os novos.

Visando facilitar o entendimento na análise das etapas de custo de correspondência, agregação de evidências e seleção de disparidades, são utilizados como exemplo os dados de um par de imagens de entrada com 320x200 com disparidade máxima de 10 e com janela de 3x3.

4.2.2 Custo de Correspondências

Para a primeira rodada são alocadas 10 **faixas de agregação** de 320 de largura e 4 de altura e realiza-se o processo de cálculo de correspondências (SSD) para cada pixel das linhas 0, 1 e 2 (janela 3x3). Isto é, compara-se o pixel (45, 1) da imagem da esquerda com o pixel (45,1) da imagem da direita e armazena-se o resultado na posição (45,1) da **faixa de agregação** 0 (equivale à disparidade 0). Em seguida, compara-se o pixel (45,1) da imagem da esquerda com o pixel (44,1) da imagem da direita e armazena-se o resultado na posição (45,1) da **faixa de agregação** 1 (equivale à disparidade 1), e assim por diante (Figura 34).

Figura 34 - Custo de correspondência para o primeiro preenchimento das faixas de agregação para uma janela 3x3.

Na segunda rodada, são usadas as mesmas 10 **faixas de agregação**, porém agora é realizado o cálculo do custo de correspondência de somente uma linha abaixo das imagens de entrada e o resultado é armazenado na linha que sobrou da primeira rodada da **faixa de agregação** (Figura 35).

Figura 35 - Custo de correspondência para demais linhas até o fim das imagens de entrada.

Na terceira rodada a linha 1 das **faixas de agregação** é substituída pelo cálculo do custo de correspondência da próxima linha abaixo das imagens de entrada. O processo segue assim, substituindo a linha desnecessária das *faixas de agregação* e adicionando novas linhas recentemente calculadas.

4.2.3 Agregação de Evidências

Para o processo de agregação são alocadas 10 linhas com 320 pixels cada, estas linhas são chamadas **linhas de seleção de disparidade**. Nestas linhas, são armazenados os resultados das somas dos valores contidos nas **faixas de agregação**, lembrando que as faixas têm a altura da janela mais 1. Logo na primeira rodada devem ser consideradas somente as 3 primeiras linhas. Seguindo o mesmo princípio do *box filtering*, cada deslocamento da janela à direita aproveita o resultado da agregação anterior adicionando uma coluna à direita e subtraindo uma coluna à esquerda (Figura 36).

Figura 36 - No alto a primeira agregação, abaixo as seguintes aproveitam os resultados obtidos anteriormente.

Na segunda rodada, a técnica muda. Daqui em diante, é aproveitado o resultado da **linha de seleção de disparidade** da rodada anterior, é subtraída a linha da janela anterior e adicionada a nova linha da janela (Figura 37). E assim, segue até o fim da varredura nas imagens de entrada.

Figura 37 - Processo de agregação para as demais faixas de agregação.

4.2.4 Seleção de Disparidades

Por último, é criado mais um plano com 320x200, onde é armazenado linha a linha o valor da disparidade que obteve a melhor correspondência. A seleção da melhor correspondência ocorre na comparação da mesma posição das **linhas de seleção de disparidade**, por exemplo, para a posição 70 entre todas as linhas a linha 4 obteve o menor resultado (WTA). Logo, significa que no mapa de disparidades será armazenado na posição (70, y) o tom de cinza correspondente à disparidade 4 (onde o y representa o ponto da varredura principal).

4.2.5 Consumo de Memória

Durante o processo de procura por correspondências, além das imagens de entrada, são armazenados na memória os resultados intermediários dos cálculos para reuso, o que dá desempenho aos algoritmos tradicionais. Com a técnica das faixas, as imagens de entrada já carregadas são tratadas em faixas e o processo de procura por correspondências ocorre somente na faixa, sendo que a mesma é deslocada de cima para baixo nas imagens de entrada, permitido assim, o reuso dos resultados dos cálculos, o que proporciona bom desempenho. Esta economia de memória é fundamental para permitir o uso de computadores convencionais.

É importante quantificar o uso de memória em cada técnica. Sabendo-se que a carga das imagens e do mapa de disparidades ocorre em ambas as técnicas e o principal foco da técnica das faixas foi o DSI, logo será enfatizado o cálculo do consumo de memória no DSI.

Como descrito no CAPÍTULO 1 em 1.1 Apresentação, o tamanho ocupado pelo DSI nas técnicas tradicionais é de:

$$\begin{aligned}
 DSI &= 4xyd \\
 DSI &= 4x \frac{3}{4} x 0,04x \\
 DSI &= 0,12x^3
 \end{aligned}
 \tag{7}$$

Onde **x** é a largura de imagem de entrada, **y** a altura e **d** a disparidade.

Para a técnica das faixas, não se tem o DSI no mesmo sentido das técnicas tradicionais, mas sim as **faixas de agregação** (FA). Considerando que o resultado da diferença quadrática (entre cada pixel das imagens) seja armazenado em uma variável do tipo *float* (4 bytes), também se estima que número de níveis de disparidade **d** seja 4%¹ da largura da imagem, que a altura de uma imagem segue o padrão de 3/4 da largura ($y = 3x/4$) e que a altura da janela **j** seja 3%² ($j = 0,03y$) da altura da imagem **y** (Figura 38).

Figura 38 - Tamanho das faixas de agregação.

1 A disparidade está mais ligada à distância entre as câmeras (mais detalhes no CAPÍTULO 2 no item 2.3). Foi adotado este percentual de 4% da largura, pois ele é bastante coerente se comparado com o valor usado em diversas imagens. Esta estimativa é necessária para o cálculo do consumo de memória para diferentes resoluções.

2 O tamanho da janela tem mais relação com as características visuais das imagens de entrada, porém quanto maior a resolução de uma imagem maior a tendência de crescimento da janela. Foi adotado este percentual de 3% da altura, pois ele é bastante coerente se comparado com o valor usado em diversas imagens. Esta estimativa é necessária para cálculo de consumo de memória para diferentes resoluções.

$$\begin{aligned}
 FA &= 4x(1+j)d \\
 FA &= 4x\left(1 + \frac{0,09x}{4}\right)0,04x \\
 FA &= \left(1 + \frac{0,09x}{4}\right)0,16x^2 \\
 FA &= 0,16x^2 + \frac{0,09x0,16x^2}{4} \\
 FA &= 0,16x^2 + 0,0036x^3
 \end{aligned} \tag{8}$$

Uma vez definidas as fórmulas que regem o consumo de memória para ambas as técnicas, pode-se traçar as curvas para diversas resoluções (Figura 39). Como o consumo de memória das **faixas de agregação** é uma pequena fração do consumo do DSI, a visualização no mesmo gráfico fica prejudicada. Na Figura 40 é apresentada somente a curva das **faixas de agregação**.

Figura 39 - Consumo de memória para a carga do DSI e das faixas de agregação.

Figura 40 - Consumo de memória para a carga das faixas de agregação.

Para facilitar a comparação dos dados são apresentados na Tabela 2 os resultados numéricos destas curvas, apenas para algumas resoluções.

Tabela 2 – Cálculo de consumo de memória do DSI e da faixa de agregação para algumas resoluções.

				Tradicional	Técnica das Faixas
x	y	d	j	MB	MB
320	240	12,8	7,2	3,750	0,128
640	480	25,6	14,4	30,000	0,963
800	600	32,0	18,0	58,594	1,855
1024	768	41,0	23,0	122,880	3,846
1600	1200	64,0	36,0	468,750	14,453
2592	1944	103,7	58,3	1992,904	60,812
3200	2400	128,0	72,0	3750,000	114,063
3600	2700	144,0	81,0	5339,355	162,158
4800	3600	192,0	108,0	12656,250	383,203
5200	3900	208,0	117,0	16091,309	486,865

x= Largura da imagem
y= Altura da imagem
d= Níveis de disparidade
j= Altura da janela

Nota-se que a técnica das faixas utiliza apenas uma pequena fração de memória se comparada às técnicas tradicionais (que usam o DSI). Para uma resolução de 2592 x 1944 (5.1 Megapixel), por exemplo, o consumo é de

aproximadamente 1992,9MB de memória RAM para o DSI e apenas 60,8MB (3%) para as **faixas de agregação**. Elevando a resolução para 5200 x 3900 (20 Megapixel), o consumo vai para aproximadamente 16091,3MB de memória RAM para o DSI e apenas 486,8MB (novamente 3%) para as **faixas de agregação**.

Os mapas de disparidade produzidos pela técnica das faixas são exatamente os mesmos produzidos pelo código tradicional desenvolvido neste trabalho. Estes são apresentados na Figura 25b e na Figura 25e.

Além do consumo de memória necessário para as faixas de agregação FA (técnica das faixas) e o DSI (abordagem tradicional), existem outras etapas que exigem memória. É necessário espaço para a carga das imagens, para o processo de agregação e para o mapa de disparidades, mais detalhes no Anexo 1.

Para demonstrar a ordem de grandeza do consumo de memória destas etapas, é apresentada, na Tabela 2, uma coluna adicional com o cálculo do consumo.

Tabela 3 - Cálculo de consumo de memória para as demais etapas do processo de procura por correspondências.

				Tradicional		Técnica das Faixas	
				Demais Etapas	DSI	Demais Etapas	FA
x	y	d	j	MB	MB	MB	MB
320	240	12,8	7,2	1,172	3,750	0,895	0,128
640	480	25,6	14,4	4,688	30,000	3,578	0,963
800	600	32,0	18,0	7,324	58,594	5,591	1,855
1024	768	41,0	23,0	12,000	122,880	9,160	3,846
1600	1200	64,0	36,0	29,297	468,750	22,363	14,453
2592	1944	103,7	58,3	76,887	1992,904	58,690	60,812
3200	2400	128,0	72,0	117,188	3750,000	89,453	114,063
3600	2700	144,0	81,0	148,315	5339,355	113,214	162,158
4800	3600	192,0	108,0	263,672	12656,250	201,270	383,203
5200	3900	208,0	117,0	309,448	16091,309	236,212	486,865

x= Largura da imagem
y= Altura da imagem
d= Níveis de disparidade
j= Altura da janela

Nota-se que o DSI representa o grande problema na questão do consumo de memória, e por isso ele foi o foco do trabalho.

4.3 Calibração Rígida

A forma tradicional de trabalho com visão estéreo vinculada à restrição epipolar está bem estabelecida e apresenta resultados muito bons no âmbito de desempenho e na construção do mapa de disparidades. Como já descrito, o uso da restrição epipolar exige a construção de aparatos fotográficos com precisão, tanto em sua disposição física quanto nos ajustes ópticos e também nas características internas da câmera. Isto faz a pesquisa sobre visão estéreo ou Reconstrução 3D se tornar onerosa e em alguns casos até inacessível. Não utilizar a restrição epipolar provoca um aumento da complexidade e perda de desempenho do algoritmo de busca por correspondências. Por outro lado, reduz os custos dos equipamentos e facilita a disposição dos aparatos fotográficos na obtenção do par de imagens, devido a não exigência de precisão e ao fato de permitir trabalhar somente com uma câmera, o que preserva os mesmos ajustes ópticos em ambas as fotos.

Por não utilizar a restrição epipolar (que permite a busca 1D) será necessária uma procura mais sofisticada que exigirá maior tempo de execução. É realizada uma busca 2D numa região de maior probabilidade de encontrar a correspondência, uma vez selecionado o ponto na imagem da esquerda, é definida uma janela de procura na imagem da direita (Figura 41).

Figura 41 - Ponto selecionado na imagem da esquerda e janela de procura na imagem da direita.

Nesta procura, o volume de dados aumenta drasticamente. Se para 10 níveis de disparidade for feita a procura por correspondências em cinco linhas serão varridos 50 pixels, 5 vezes mais do que no processo normal, lembrando que para cada pixel, existe um conjunto de cálculos, os quais envolvem sua vizinhança. Esta procura exaustiva faz com que o consumo de memória cresça. Se um DSI já ocupava um espaço de memória significativo com a busca 2D (em apenas 5 linhas) geraria o consumo de memória de 5 DSIs. Como a Técnica das Faixas reduziu drasticamente o uso de memória, torna-se viável a procura por correspondências em uma região bidimensional.

A procura 2D significa que para um ponto na imagem da esquerda é procurada sua correspondência em uma região 2D da imagem da direita, podendo ser acima ou abaixo da linha da imagem da esquerda. Não importa a linha onde é encontrada a correspondência, mas sim a posição em x que representa a disparidade. Isto deixa claro que a procura 2D não significa liberdade no posicionamento e calibração das câmeras, mas sim uma flexibilização.

Para ilustrar o comportamento do algoritmo modificado para a busca 2D, serão retomadas as mesmas imagens de entrada da Figura 28, porém agora realizando uma busca 2D. A Figura 42, apresenta como referência as imagens originais e o mapa de disparidades gerado (aqui considerado como ideal).

Figura 42 - Par estéreo original da imagem Vênus (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades.

Para a distorção do tipo deslocamento de linha, por exemplo, uma linha horizontal que na imagem da esquerda está em $y=12$ na imagem da direita é encontrada na linha $y=10$.

A Figura 43a, ilustra a distorção, b apresenta o mapa de disparidades para uma busca 1D e a Figura 43c apresenta a comparação com o mapa de disparidades ideal da Figura 42. Para a busca 2D produziu-se o mapa de disparidades da Figura 43d muito semelhante ao mapa de disparidades ideal (o que pode ser comprovado pela comparação apresentada na Figura 43e, onde ocorre a predominância de pixels brancos).

(a)

(b)

(c)

(d)

(e)

Figura 43 - (a) Ilustração da distorção; (b) mapa de disparidades para uma busca 1D; (c) comparação do mapa de disparidade (b) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças¹; (d) mapa de disparidade para uma busca 2D (5 linhas de procura); (e) comparação do mapa de disparidade (d) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças

¹ Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

Para a distorção do tipo rotação da câmera, por exemplo, uma linha horizontal que está na linha $y=12$ na imagem da esquerda, mas na imagem da direita começa na linha $y=10$, passa pelo $y=11$, 12 e termina na linha 13.

A Figura 44a, ilustra a distorção, b apresenta o mapa de disparidades para uma busca 1D e a Figura 44c apresenta a comparação com o mapa de disparidades ideal Figura 42. Para a busca 2D, produziu-se o mapa de disparidades da Figura 44d, o qual é visualmente parecido com o mapa de disparidades ideal.

Na comparação pixel-a-pixel, (Figura 44e), nota-se a igualdade atingida principalmente numa faixa horizontal central, mas as faixas horizontais superior e inferior, apesar de não serem iguais, produziram (no mapa de disparidades Figura 44d) pixels com valores muito próximos ao ideal. Isto significa que foi encontrada uma boa correspondência próxima da localização onde deveria estar a correspondência ideal. Isto foi causado porque a rotação da imagem mudou o x dos pontos e estes têm uma relação direta com a disparidade. Nota-se que este efeito ocorre principalmente na faixa superior e inferior da imagem, o que justifica a não igualdade em relação ao mapa de disparidades ideal, mas produz um resultado visualmente parecido.

(a)

Figura 44 - (a) Ilustração da distorção; (b) mapa de disparidades para uma busca 1D; (c) comparação do mapa de disparidade (b) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças¹; (d) mapa de disparidade para uma busca 2D (3 linhas de procura); (e) comparação do mapa de disparidade (d) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças.

¹ Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

Para distorções na lente, por exemplo, uma linha horizontal que na imagem da esquerda está na linha $y=12$, na imagem da direita começa na linha $y=12$, sobe para linha $y=13$, vai até a $y=14$, depois volta para $y=13$ e por fim na $y=12$.

A Figura 45a, ilustra a distorção, b apresenta o mapa de disparidades para uma busca 1D e a Figura 45c apresenta a comparação com o mapa de disparidades ideal Figura 42. Para a busca 2D, produziu-se o mapa de disparidades da Figura 45d, o qual é muito parecido com o mapa de disparidades ideal (o que pode ser comprovado pela comparação apresentada na Figura 45e, onde ocorre a predominância de pixels brancos).

Figura 45 - (a) Ilustração da distorção; (b) mapa de disparidades para uma busca 1D; (c) comparação do mapa de disparidade (b) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças¹; (d) mapa de disparidade para uma busca 2D (7 linhas de procura); (e) comparação do mapa de disparidade (d) com o mapa de disparidades ideal Figura 42, pixels pretos indicam diferenças.

A codificação do programa exigiu um cuidado extra na elaboração da procura 2D, pois todas as etapas do processo da Correspondência Estéreo consideraram diversas linhas de variação, e especialmente a Seleção de

¹ Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

Disparidade define qual é a melhor correspondência entre todas as linhas em questão.

Nota-se que a procura 2D atingiu seus objetivos, conseguindo produzir mapas de disparidade com um alto nível de similaridade em relação ao ideal, mesmo para as distorções que provocaram uma queda na qualidade tão significativa na busca 1D (linha epipolar).

4.4 Precisão nas Cores

Algoritmos de visão estéreo geralmente são muito sensíveis às variações de luminosidade, pois utilizam a intensidade das cores para realizar a procura por correspondências, pequenas variações entre as duas fotos podem apresentar resultados bastante distintos no mapa de disparidades, como já descrito no CAPÍTULO 3 em 3.3 Precisão nas Cores.

Em um ambiente aberto ou improvisado, munido de câmeras convencionais, há dificuldade em se obter duas fotos com intensidades de cores iguais. Quando se utiliza uma câmera, pode ocorrer uma pequena variação de luminosidade no intervalo de tempo entre cada foto. Por outro lado, quando se utiliza duas câmeras, as fotografias dificilmente terão as mesmas intensidades das cores, devido à imprecisão das câmeras convencionais.

Os algoritmos que utilizam as intensidades das cores no processo de procura por correspondências são sensíveis, pois cores ligeiramente diferentes induzem a falsas correspondências. Por exemplo, quando há um degradê e o par de imagens não está normalizado, a correspondência pode começar em um ponto errado e se propagar por todo degradê. Isto pode ser observado na Figura 49b na luminária laranja. Nota-se, no mapa de disparidades, que esta região deveria ter a mesma disparidade (Figura 49a), porém a variação no brilho fez parecer que a disparidade variava, gerando trocas substanciais nos tons de cinza no mapa de disparidades da Figura 49b.

Uma forma conhecida de anular a sensibilidade às variações de intensidade das cores é realizar um pré-processamento nas imagens de entrada, e especificamente, a detecção de bordas utilizando o operador gradiente (SCHARSTEIN, 1999) se mostra muito eficiente.

O operador gradiente representa a primeira derivada de um sinal (Figura 46).

Figura 46 - (a) Imagem com faixa azul clara no centro de uma região azul escura; (b) representação gráfica das intensidades das cores para a linha pontilhada em (a); (c) Primeira derivada.

Para realizar esta operação em uma imagem são utilizadas máscaras conhecidas como operadores de Sobel (Figura 47) (mais detalhes podem ser encontrados em (GONZALEZ; WOODS, 2000)).

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figura 47 - Máscaras conhecidas como operadores de Sobel.

Não havendo valores negativos nas imagens, foi adotada a estratégia de somente utilizar o módulo do gradiente. Além disso, também há um limite superior para os valores das operações nas imagens que é 255, foi adotado um corte nos valores superiores a 255. O resultado visual desta operação é mostrado na Figura 48b.

Figura 48 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002); (b) mesmo par estéreo processado com o operador gradiente, valores absolutos e limitados a 255.

Estas estratégias não oneram o desempenho e o corte simples em 255 iguala bordas com valores altos, facilitando desta forma a procura por correspondência.

Retomando as mesmas imagens da Figura 29, na qual a imagem da direita teve seu brilho modificado, agora é apresentado na Figura 49c o mapa de disparidades processado utilizando o operador gradiente. Nota-se a grande similaridade entre os mapas de disparidade da Figura 49a e da Figura 49c, que é comprovada na comparação pixel-a-pixel à direita na Figura 49c.

Figura 49 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades; (b) brilho modificado na imagem da direita, mapa de disparidades gerado e à direita comparação com o mapa de disparidades de (a), onde a cor preta indica pixels diferentes¹; (c) brilho modificado na imagem da direita, mapa de disparidades gerado (obtido utilizando operador gradiente) e à direita comparação com o mapa de disparidades de (a), onde a cor preta indica pixels diferentes.

Apesar da similaridade obtida com o operador gradiente, foi realizada uma comparação pouco adequada: um mapa de disparidades baseado na

¹ Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

intensidade das cores Figura 49a com um mapa de disparidades baseado nas bordas Figura 49c.

Esta comparação foi necessária para prover um paralelo com a Figura 49b. Uma maneira mais adequada de demonstrar a insensibilidade da técnica às variações na iluminação é comparar dois mapas de disparidade processados com operador gradiente, um proveniente das imagens originais e outro proveniente do par que sofreu distorção no brilho. A Figura 50 apresenta esta comparação. Nota-se que o mapa de disparidade produzido tornou-se ainda mais preciso. A comparação pixel-a-pixel é apresentada na Figura 50b, à direita.

Figura 50 - (a) Par estéreo original da imagem Tsukuba (SCHARSTEIN; SZELISKI, 2002) e seu correspondente mapa de disparidades (obtido utilizando Operador Gradiente); (b) brilho modificado na imagem da direita, mapa de disparidades gerado (obtido utilizando Operador Gradiente) e à direita comparação com o mapa de disparidades de (a), onde a cor preta indica pixels diferentes¹.

Foi demonstrado que o uso do operador gradiente, na maneira aqui codificada, reduz muito a sensibilidade às variações na iluminação. Superando mais um dos desafios apresentados neste trabalho.

¹ Para imagens que têm bordas brancas, foi incluído um contorno azul (fora da imagem) para facilitar a visualização.

4.5 Desempenho em Execução

No CAPÍTULO 3, foi realizada a comparação de desempenho entre duas codificações tradicionais (SSD / Janela quadrada / WTA) com o objetivo de validar o desenvolvimento feito neste trabalho, tanto no tempo de execução, quanto no resultado visual.

Retomando a comparação de desempenho, mas agora colocando à prova a técnica das faixas. Será avaliado se a redução de consumo de memória interferiu de forma positiva ou negativa no tempo de execução do algoritmo. Serão apresentados os dados referentes a três códigos:

- tradicional Middlebury College;
- tradicional com código implementado pelo autor deste trabalho;
- técnica das faixas.

O objetivo é retomar os dados já apresentados e compará-los com a técnica das faixas. O código tradicional aqui implementado compartilha boa parte do código com a técnica das faixas, logo as diferenças de desempenho entre as duas fornecerão uma boa referência da influência desta nova técnica.

Um programa que executa a procura por correspondências, normalmente exige as etapas:

1. leitura dos arquivos de entrada;
2. alocação de memória de trabalho;
3. procura por correspondências;
 - a. pré-processamento;
 - b. cálculo dos custos;
 - c. agregação das evidências;
 - d. seleção das disparidades;

- e. refinamento;
- 4. gravação dos arquivos com os resultados;
- 5. liberação da memória alocada.

Todas as medidas de tempo se referem à etapa de procura por correspondências, sendo utilizada a função C: clock() antes e depois da função ou das funções responsáveis pela procura por correspondências. Os retornos desta função foram armazenados e o final é subtraído do inicial de forma a apresentar o tempo de execução.

A título meramente informativo, no código do Middlebury College, a função RunMatcher(); invoca a função ComputeCorrespondence(); que realiza todas as alocações de memória necessárias e depois possui as chamadas:

```
clock_t time0 = clock();    // record start time (can't use global
                             timer...)

// Invoke all of the stereo matching processing steps
PreProcess();    // see StcPreProcess.cpp
RawCosts();      // see StcRawCosts.cpp
Aggregate();     // see StcAggregate.cpp
Optimize();      // see StcOptimize.cpp
Refine();        // see StcRefine.cpp

clock_t time1 = clock();    // record end time
total_time = (float)(time1-time0)/(float)CLOCKS_PER_SEC;

Copyright © Daniel Scharstein and Richard Szeliski, 2001.
```

Note que a função clock() é chamada antes e depois das funções que realizam a procura por correspondências.

Tanto no código tradicional aqui implementado, quanto na técnica das faixas ocorre somente o cálculo dos custos, agregação das evidências e seleção das disparidades. Não foi necessário realizar pré-processamento e nenhum refinamento das imagens para produzir os mesmos resultados (exceto as bordas) (SCHARSTEIN; SZELISKI, 2001). Para a comparação ocorrer de forma adequada, foi modificado o código de (SCHARSTEIN; SZELISKI, 2001)

de maneira que somente fossem contabilizados os tempos das mesmas etapas:

RawCosts() realiza o cálculo dos custos;

Aggregate() realiza a agregação das evidências;

Optimize() realiza a seleção das disparidades.

```
// Invoke all of the stereo matching processing steps
PreProcess();    // see StcPreProcess.cpp

clock_t time0 = clock();    // record start time (can't use global
timer...)
RawCosts();       // see StcRawCosts.cpp
Aggregate();      // see StcAggregate.cpp
Optimize();       // see StcOptimize.cpp

clock_t time1 = clock();    // record end time
total_time = (float)(time1-time0)/(float)CLOCKS_PER_SEC;

Refine();         // see StcRefine.cpp
Copyright © Daniel Scharstein and Richard Szeliski, 2001.
```

O processo de gravação da imagem com o mapa de disparidades é realizado pela função WriteData(); que é posterior à execução da função RunMatcher();, deixando claro que as operações de gravação de arquivos não são computadas no contador total_time.

É importante ressaltar que no código do Middlebury College, não ocorrem operações de vídeo, pois o programa é executado puramente em linha de comando.

O equipamento utilizado foi um *Notebook* Pentium III 900MHz com 384MB RAM e *Windows 2000 Professional*, sendo que no momento da execução dos programas, havia memória livre suficiente (para evitar paginação).

São utilizadas as imagens da Figura 51: Venus e Tsukuba (SCHARSTEIN; SZELISKI, 2002) e Cones (SCHARSTEIN; SZELISKI, 2003), a procura por correspondência usa os parâmetros da Tabela 4 e são utilizados os mesmos critérios de comparação.

Figura 51 - Pares de Imagens Estéreo, Cones (alto), Venus (meio) e Tsukuba (baixo).

Tabela 4 – Parâmetros e dados comparativos.

	Tsukuba	Venus	Cones
Largura x Altura	384 x 288	434x383	900 x 750
Níveis de Disparidade	16	20	26
Janela	9 x 9	9 x 9	15 x 15
Imagem da Esquerda (nome original):	scene1.row3.col1.ppm	im0.ppm	im2.ppm
Imagem da Direita (nome original):	scene1.row3.col2.ppm	im5.ppm	im3.ppm
disp_scale ¹ :	15	12	9
Consumo de Memória			
Técnica das Faixas (MB)	0.23	0.33	2.10
Abordagem Tradicional (MB)	6,75	12.68	66,95
Desempenho			
Técnica das Faixas (s)	0.240	0.470	3.165
Abordagem Tradicional código implementado aqui (s)	0.441	0.861	4.907
Abordagem Tradicional código do Middlebury College (s)	0.558	1.008	4.927

O consumo de memória (Tabela 4) pode ser mais bem visualizado na Figura 52. Para as três imagens, a técnica das faixas consumiu com as faixas de agregação aproximadamente 3% da memória quando comparada com o DSI da abordagem tradicional, o que significa uma redução de 97%.

¹ Somente usado na abordagem do tradicional do Middlebury College, para obter exatamente o mesmo mapa de disparidade.

Figura 52 - Comparação do consumo de memória.

Os dados de desempenho demonstram que as duas abordagens tradicionais têm um desempenho parecido, sendo que a diferença se reduz significativamente com o aumento da quantidade de níveis de disparidade. Estes dados podem ser mais bem visualizados na Figura 53.

Figura 53 - Desempenho para a procura por correspondências.

Como já comentado no CAPÍTULO 3, o código tradicional aqui implementado apresenta resultados melhores se comparado com o código do Middlebury College. Mas a técnica das faixas apresenta resultados ainda mais significativos. Para a imagem Tsukuba, ela executou em apenas 43% do tempo em relação a Código do Middlebury College e executou em 55% do tempo em relação ao código tradicional aqui implementado. Para a imagem Venus, estas percentagens foram 47% e 75%, respectivamente. Para a imagem Cones, a técnica das faixas executou em 64% do tempo para ambas, pois o tempo de execução das duas abordagens tradicionais foi bastante similar.

A técnica das faixas apresentou desempenho melhor que a técnica tradicional para todas as imagens. Isto acontece, pois ocorre a varredura de cima para baixo em y , apenas uma vez. Na abordagem tradicional esta varredura ocorre pelo menos três vezes: uma no cálculo dos custos de correspondências, outra na agregação de evidências e mais uma na seleção de disparidades. Mais uma varredura pode ocorrer se for utilizado plano temporário (CAPÍTULO 2) durante a agregação de evidências.

Estas varreduras são dispendiosas, pois estão ligadas diretamente à altura da imagem, além de ocorrerem para todos os níveis de disparidade.

4.6 Geração da Malha 3D

O mapa de disparidade é uma forma de apresentação de informações tridimensionais, onde se tem a disposição espacial (x, y) e a informação da disparidade (inverso da profundidade) apresentada na forma de tons de cinza. O objetivo aqui é gerar uma malha tridimensional aberta com a textura aplicada sobre ela, permitindo tanto a visualização do ambiente, quanto a geração de arquivos com estas informações.

Para gerar a superfície 3D aberta, a base é o mapa de disparidades. A abordagem aqui adotada, foi construir uma malha uniforme de vértices como a apresentada na Figura 54.

Figura 54 - Malha 3D inicial.

É criada uma relação direta entre os vértices desta malha com os pixels do mapa de disparidades. Por exemplo, para um mapa de disparidade de 80 de largura e 80 de altura e uma malha de 8 x 8 vértices sobreposta ao *bitmap*, haverá um vértice a cada $80/8=10$ pixels na horizontal e um vértice a cada $80/8=10$ pixels na vertical. Como no *bitmap* a origem das coordenadas está no canto superior esquerdo e na malha 3D a origem está no centro, foi adotado

que a posição $(0, 0)$ do *bitmap* é equivalente à $(-40, 40, z)$ da malha do exemplo (Figura 55).

Figura 55 - (a) Origem das coordenadas no *bitmap* se encontra no canto superior esquerdo; (b) origem das coordenadas se encontra no centro.

Para cada vértice da malha é colhida a cor do pixel equivalente à sua posição, seu valor é subtraído de 255 gerando o valor de z . Para cada tom de cinza há um nível de profundidade, utilizando uma imagem com uma escala de tons de cinza como da Figura 56a, é demonstrado o posicionamento de cada faixa da escala em uma profundidade produzindo o efeito visual da Figura 56b.

Figura 56 - (a) Faixas com tons de cinza partindo do branco até o preto; (b) resultado gerado com a técnica de Reconstrução 3D descrita aqui.

Dentro do mesmo processo de definição dos vértices é realizado também o mapeamento da textura. A mesma se origina de um *bitmap* com dimensões iguais ao mapa de disparidades, portanto seguiu-se a mesma

distribuição de pixels sobre a malha 3D. Utilizando como textura a imagem da esquerda do par estéreo Tsukuba (Figura 57a) e utilizando o mapa de disparidades ideal (Figura 57b), obter-se-á o resultado visual apresentado na Figura 57c e na Figura 57d, respectivamente, a apresentação *wireframe* e sólida do ambiente reconstruído.

(a)

(b)

(c)

(d)

Figura 57 - (a) Imagem da esquerda do para estéreo Tsukuba (SCHARSTEIN; SZELISKI, 2002);(b) mapa de disparidades ideal utilizado; (c) apresentação da geometria 3D reconstruída a partir do mapa de disparidades em modo *wireframe*; (d) Reconstrução 3D em modo sólido.

4.7 Conclusão

Este capítulo apresentou detalhadamente as soluções desenvolvidas e aplicadas para solucionar as deficiências encontradas.

Como cada solução resolve um problema isoladamente, são apresentados os resultados para cada situação separadamente.

Para ilustrar as distorções na calibração e na precisão das câmeras são utilizadas imagens da literatura com estas distorções aplicadas.

Para a medida de desempenho em execução são utilizadas as imagens da literatura sem alterações e são apresentados os dados de consumo de memória específicos para o tamanho das imagens e para os parâmetros utilizados.

CAPÍTULO 5 Resultados em uma Situação Real

Neste capítulo é apresentada a aplicação das soluções desenvolvidas a uma situação real onde não há precisão na calibração, o ambiente não é controlado e foram utilizados equipamentos convencionais.

5.1 Experimento

Duas fotografias foram obtidas com uma câmera digital comum, Sony® Cyber-shot DSC-W1 de 5.1 MegaPixels, em um ambiente não controlado. A câmera estava simplesmente apoiada em uma mesa e para que a fotos ficassem alinhadas a câmera foi guiada por uma régua. O deslocamento foi de aproximadamente 60mm (Figura 58).

Figura 58 - Calibração das câmeras para obtenção do par de imagens estéreo.

O equipamento utilizado para o processamento foi um *Notebook* Pentium III de 900MHz, com 384MB de RAM e *Windows 2000 Professional*.

As imagens obtidas são apresentadas na Figura 59.

Figura 59 - Imagens obtidas em ambiente não controlado.

Mantendo a resolução original (2592x1944) o processamento foi de ~691 segundos (Figura 60). Reduzindo a resolução das imagens para (648x486), visando facilitar os ajustes dos parâmetros e agilidade nos testes, o processamento foi de ~17 segundos (Figura 62).

Figura 60 – Tela do sistema executando com imagens de 5.1 MegaPixel, em destaque a resolução das imagens de entrada e o tempo de execução.

Observando-se estas imagens em detalhe (Figura 61), nota-se o resultado da falta de rigidez na calibração, neste caso, um deslocamento de linha como descrito no CAPÍTULO 4 em 4.3 Calibração.

Figura 61 - (a) Imagem da esquerda onde é identificada uma linha específica $y=217$; (b) imagem da direita identificando a mesma linha da imagem (a) porém agora ela se encontra na posição $y=216$.

Apesar do ambiente não ser controlado, não houve variação de luminosidade significativa, capaz de degradar a procura por correspondências utilizando as intensidades das cores. Com isso, foi executado o processo tanto com as intensidades das cores (Figura 62), quanto com o operador gradiente (Figura 63). Foram utilizados em ambos os processamentos 50 níveis de disparidade, 3 linhas de variação (busca 2D), janela de 21x21 e malha 3D de 64x64 vértices.

Figura 62 - Execução usando as intensidades das imagens de entrada, em destaque a resolução das imagens de entrada e o tempo de execução.

Figura 63 - Execução usando operador gradiente.

O consumo de memória para as **faixas de agregação** é de $FA=4x(1+j)d$ como são 50 níveis de disparidade em 3 linhas serão 150 faixas, logo o consumo de memória será: $FA=4*648*(1+21)*150=8,16\text{MB}$. Se fosse usada a abordagem tradicional, sendo o $DSI=4xyd$, resultaria em: $DSI=4*648*486*150=180,20\text{MB}$.

Comparando a Figura 62 com a Figura 63, nota-se que os mapas de disparidade estão diferentes. Usando as intensidades (Figura 62), o fundo distante ficou com melhor qualidade, porém a borda esquerda da mesa ficou distorcida. Já com o operador gradiente (Figura 63), a mesa ficou bem definida, mas o fundo distante distorcido.

O melhor resultado é obtido pela combinação manual dos mapas de disparidade e realizando-se pequenas correções. O mapa de disparidades

corrigido manualmente é apresentado na Figura 64. A facilidade de correção de falhas diretamente no mapa de disparidades (*bitmap*), em vez de atuar na geometria 3D (utilizando um editor 3D), foi uma observação empírica interessante.

Figura 64 - Mapa de disparidades corrigido.

Além do processamento das imagens de entrada, da geração do mapa de disparidades e da malha 3D, o sistema aqui desenvolvido permite a carga do mapa de disparidades e a obtenção da malha 3D. A tela do programa, contendo o mapa de disparidades corrigido, é apresentada na Figura 65.

Figura 65 - Execução somente a partir do mapa de disparidades.

A partir deste momento, é possível visualizar o ambiente 3D reconstruído dentro do próprio programa. O quadro no alto à esquerda é uma janela 3D com navegação on-line que foi desenvolvida usando Microsoft DirectX®. O programa permite a visualização interativa do ambiente reconstruído, tanto em modo sólido, quanto em *wireframe*, tanto com a imagem de entrada, quanto com o mapa de disparidades (Figura 66).

O arquivo VRML (.WRL) gerado pode ser importado em qualquer software de edição 3D e ser visualizado. A Figura 67 apresenta a tela do 3Ds Max® com o arquivo VRML carregado e a Figura 68 apresenta a imagem reconstituída a partir do arquivo VRML com a imagem de entrada como textura.

Figura 66 - Resultados obtidos na reconstrução 3D à esquerda *flat* e à direita *wireframe*. No alto, a imagem de entrada e abaixo o mapa de disparidades.

Figura 67 - Tela do 3Ds Max com o arquivo VRML carregado.

Figura 68 - Imagem gerada pelo 3Ds Max a partir do arquivo VRML e da imagem de entrada.

5.2 Conclusão

O experimento apresentado, neste capítulo, mostrou que o sistema desenvolvido se adaptou muito bem a uma situação real (ambiente não controlado, imagens não normalizadas, calibração flexível e uso de equipamentos convencionais), de forma que o objetivo principal deste trabalho foi atingido.

CAPÍTULO 6 Conclusões

6.1 Contribuições

Neste trabalho, foi concebida a técnica das faixas, uma contribuição significativa em visão estéreo, pois apresenta uma solução eficiente para o consumo de memória do DSI, o qual é utilizado amplamente, tanto para métodos locais, quanto para métodos globais. Além da redução drástica no consumo de memória, também apresentou desempenho melhor, se comparado com as abordagens tradicionais.

O código fonte do protótipo aqui desenvolvido está disponível sob a GPL (*General Public License*) como software de código aberto no site <http://sourceforge.net/projects/reconststereo/>.

Além do código fonte publicado no site *Source Forge* está disponível também documentação (em inglês) sobre o uso do protótipo, um tutorial da instalação das ferramentas necessárias para a programação (usando somente softwares gratuitos) e outro tutorial que demonstra a compilação do programa, esta documentação é apresentada no APÊNDICE B – Documentação Publicada no Site Source Forge.

Foi publicado o artigo: Sistema de Reconstrução 3D de Baixo Custo (JOSÉ, 2007a) na VII International Conference on Graphics Engineering for Arts and Design, Graphica 2007. Este artigo aborda a questão da redução de custos dos equipamentos de aquisição e processamento para a reconstrução 3D visando facilitar o acesso aos conhecimentos nesta área.

Foi também publicado o artigo: Sistema de Auxílio à Modelagem 3D usando Visão Estéreo (JOSÉ, 2007b) no Workshop de Realidade Virtual e Aumentada, WRVA'07. Este artigo aborda as dificuldades do processo de

modelagem e os benefícios que o sistema de reconstrução 3D aqui desenvolvido pode proporcionar.

6.2 Trabalhos Futuros

Sugere-se como trabalhos futuros, investir em pesquisas e desenvolvimentos de soluções que proporcionem auto-calibração, persistência e multi-resolução.

Auto Calibração - Usar a flexibilidade do sistema para se auto-calibrar, considerando duas câmeras fixas, o sistema deve realizar a procura por correspondências, identificar as distorções de calibração e ajustar os parâmetros de procura para as próximas tomadas.

Persistência 3D – Utilizando-se câmeras *web*, reconstruir o ambiente 3D e manter a geometria 3D obtida (não reconstruir a cada *frame*).

Desta forma, novos objetos que entrem em cena são reconstruídos sem alterar os dados do fundo. Esta abordagem pode ser útil em soluções similares ao cromaqui (*chroma key*), por outro lado, esta pode ser uma alternativa à reconstrução 3D em tempo real, pois tanto na entrada de novos objetos em cena, quanto na movimentação das câmeras é necessário somente reconstruir as novas informações, preservando as iniciais, reduzindo-se a carga de processamento.

Multi - resolução – Tratar a imagem com baixa resolução e em pontos de interesse usar toda a informação (alta resolução), assim como funciona a visão humana. Em geral, somente são tratados os detalhes dos objetos de interesse, mas um caso particular ocorre no momento em que for tratado somente o centro da imagem com alta resolução (simulação de cones e bastonetes).

6.3 Conclusões

Os objetivos deste trabalho foram atingidos. Foi realizada uma ampla pesquisa sobre reconstrução 3D (CAPÍTULO 1) e uma pesquisa profunda sobre visão estéreo (CAPÍTULO 2). Foram escolhidas linhas de pesquisa ativas e que apresentam perspectivas promissoras de continuidade no futuro, sendo selecionados para aprofundamento os algoritmos e técnicas que mais se aproximavam do objetivo principal do trabalho: a reconstrução 3D de baixo custo.

Foram codificados os algoritmos tradicionais (CAPÍTULO 3), sendo identificadas e medidas suas deficiências. Pesquisaram-se soluções conhecidas para estas deficiências (como o uso do operador gradiente em visão estéreo), por outro lado foram aplicadas soluções heterodoxas¹ como a procura 2D. E ainda foi concebida uma técnica inovadora para solucionar o desafio do consumo excessivo de memória, a grande contribuição técnica deste trabalho, a técnica das faixas.

Um sistema de reconstrução 3D e visualização interativa foi desenvolvido. Seus resultados foram medidos e apresentados (CAPÍTULO 4). Este sistema desenvolvido foi submetido a uma situação real de uso (CAPÍTULO 5), mostrando aplicabilidade, eficiência, flexibilidade e funcionalidade. Neste caso real, foram utilizados equipamentos convencionais, comprovando que é possível realizar reconstrução 3D de baixo custo.

Foi estudado, pesquisado, classificado, desenvolvido e demonstrado um caminho viável para a reconstrução 3D em ambientes com recursos financeiros escassos, possibilitando o acesso e a disseminação de conhecimento nesta área de pesquisa.

¹ Esta é uma solução heterodoxa, pois contraria a idéia comum de uso da restrição epipolar com calibração rígida, para permitir uma procura 1D.

REFERÊNCIAS BIBLIOGRÁFICAS

ARNOLD, R. D. **Automated Stereo Perception**. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University, 1983

BOBICK, A. F.; INTILLE, S. S. **Large occlusion stereo**. International Journal of Computer, Vision, 33(3), 181–200, 1999.

BOYKOV, Y.; VEKSLER, O.; ZABIH, R. **Fast approximate energy minimization via graph cuts**. In Seventh International Conference on Computer Vision (ICCV'99), pages 377–384, Kerkyra, Greece, 1999.

EID, A. H. et al. **Validation of 3-D Reconstruction from Sequence of Images**. Proceeding of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2002), pp. 375-380, Crete, Greece, June 2002

FAUGERAS, O. **Three-dimensional Computer Vision: A Geometric Viewpoint**. The MIT Press, 1993.

FERNANDES, V. O. et al. **Análise do Potencial da Monorestituição Digital Aplicado à Imagem Multiespectral Daedalus ATM e Aerofoto Métrica Integrado ao Laser Scanning**." COBRAC, 2004.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. Editora Edgard Blücher Ltda, 2000.

GRASSI JR, V.; OKAMOTO JR, J. **Development of an omnidirectional vision system**. Journal of the Brazilian Society of Mechanical Sciences and Engineering, Rio de Janeiro, v. XXVIII, n. 1, p. 58-68, 2006.

HANNAH, M. J. **Computer Matching of Áreas in Stereo Images**. Ph.D thesis, Stanford University, 1974.

HIRSCHMÜLLER, H.; INNOCENT, P. R.; GARIBALDI, J. M. **Real-time correlation-based stereo vision with reduced border errors**. International Journal of Computer Vision, 47(1/2/3):229–246, April-June 2002.

HIRSCHMÜLLER, H. **Accurate and efficient stereo processing by semi-global matching and mutual information**. In Proc. CVRP, volume 2, pages 807–814, 2005.

HIRSCHMÜLLER, H. **Stereo Processing by Semiglobal Matching and Mutual Information**. IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 30, No. 2, February 2008.

JOSÉ, M. A. **Sistema de Reconstrução 3D de Baixo Custo**. Graphica'2007 - XVIII Simpósio Nacional de Geometria Descritiva e Desenho Técnico e VII International Conference on Graphics Engineering for Arts and Design Graphica'07, Curitiba, PR, 2007.

JOSÉ, M. A. **Sistema de Auxilio à Modelagem 3D usando Visão Estéreo..** WRVA'07 Workshop de Realidade Virtual e Aumentada, Itumbiara, GO, 2007.

KANADE, T.; OKUTOMI, M. **A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment.** IEEE Transactions On Pattern Anaylsis And Machine Intelligence, Vol. 16, No. 9, September 1994.

KANADE, T. et al. **A stereo machine for video-rate dense depth mapping and its new applications.** In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96), pages 196–202, San Francisco, California, 1996.

KERNIGHAN, B. W.; RITCHIE, D. M. **C A Linguagem De Programação: Padrão ANSI.** Editora Campus, 1990.

KIMURA, R. et al. **A convolver-based real-time stereo machine (SAZAN).** In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), pages 457–463, Fort Collins, 1999.

MARROQUIN, J.; MITTER, S.; POGGIO, T. **Probabilistic solution of ill-posed problems in computational vision.** Journal of the American Statistical Association, 82(397), 76–89, 1987

MCDONNELL, M. J. **Box-filtering techniques.** Computer Graphics and Image Processing, vol. 17, pp. 65–70, 1981

NAKAZONE JR, S. **Avaliação Comparativa da Anatomia do Desenvolvimento da Articulação Temporomandibular, por Meio do Uso de Ressonância Magnética, Reconstrução Digital e Cortes Histológicos.** Tese de Doutorado apresentada À Faculdade de Odontologia da Universidade de São Paulo, 2005.

PUGH, K. **Programando Em Linguagem C.** Editora Mcgraw-Hill, 1990.

RUSSELL, S. J.; NORVIG, P. **Inteligência Artificial.** (Second Edition) Editora Campus 2004.

SCHARSTEIN, D. **“View Synthesis Using Stereo Vision”**, Springer, 1999.

SCHARSTEIN, D.; SZELISKI, R. **A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms.** Technical Report, MSR-TR-2001-81, Microsoft Research, November 2001.

SCHARSTEIN, D.; SZELISKI, R. **A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.** International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.

SCHARSTEIN, D.; SZELISKI, R. **High-accuracy stereo depth maps using structured light.** In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.

SOUZA, M. A.; CENTENO, T. M.; PEDRINI, H. **Integrando Reconstrução 3D de Imagens Tomográficas e Prototipagem Rápida Para a Fabricação de Modelos Médicos.** Revista Brasileira de Engenharia Biomédica, v. 19, n. 2, p. 103-115, agosto 2003.

STROUSTRUP, B. **The C++ Programming Language.** Addison-Wesley, Second Edition, Publishing Company, 1991.

SUN, C. **Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques.** International Journal of Computer Vision. vol.47, no.1/2/3, pp.99-117, May 2002.

SVOBODA, T.; PAJDLA, T.; HLAVAC, V. **Central Panoramic Cameras: Geometry and Design.** Research Report K335/97/147, Czech Technical University, Faculty of Electrical Engineering, Center for Machine Perception, available at <ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/TR-K335-97-147.ps.gz>, 1997.

VEKSLER, O. **Efficient Graph-based Energy Minimization Methods in Computer Vision.** Ph.D. thesis, Cornell University, 1999.

YANG, Y.; YUILLE, A.; LU, J. **Local, global, and multilevel stereo matching.** In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93), pages 274–279, IEEE Computer Society, New York, 1993.

SITES CONSULTADOS

Cyberware

Preços de Laser Scanner

<http://www.cyberware.com/pricing/domesticPriceList.html> Acesso em Jan/2008

Middlebury College

Imagens de referência e código fonte para taxonomia

<http://vision.middlebury.edu/stereo/> Acesso em Jan/2008

Looxis

Download do software Faceworx®

http://www.looxis.com/en/k75.Downloads_Bits-and-Bytes-to-download.htm

Acesso em Jan/2008

Visuallee

Reconstrução 3D de face em bloco de cristal

http://www.visuallee.com/weblog/2002_12_01_archive.html Acesso em Jan/2008

Canoma

Software especializado em modelagem 3D a partir de fotos.

<http://www.canoma.com/> Acesso em Jan/2008

Imagens com resultados.

<http://www.canoma.com/movies.html> Acesso em Jan/2008

Photomodeler

Software especializado em modelagem 3D a partir de fotos.

<http://www.photomodeler.com/index.htm> Acesso em Jan/2008

APÊNDICE A - Consumo Total de Memória

Este apêndice tem por objetivo apresentar detalhes sobre o cálculo total do consumo de memória, mesmo de etapas que não são otimizadas. No corpo do trabalho foi enfatizado especificamente o consumo de memória do DSI, por se tratar do ponto mais relevante. Aqui serão tratadas todas as etapas, do processo de procura por correspondências, serão usadas as seguintes notações:

x = Largura

y = Altura

d = Níveis de disparidade

j = Altura da janela

Para realizar o processo de correspondência estéreo é necessário alocar memória para:

1. Ler as imagens de entrada

Para ler as imagens de entrada em ambas as abordagens é necessário o espaço para duas imagens (largura x altura), para cada pixel são necessários 4 bytes (3 para RGB e 1 para Alfa).

Ambas abordagens:

Ler Imagens = $8xy$

2. Carregar os resultados de custo de correspondência

Aqui é que começam as diferenças entre a abordagem tradicional e a técnica das faixas. Na abordagem tradicional é necessário alocar o DSI.

Abordagem tradicional: $DSI = 4xyd$

Na técnica das faixas é necessário alocar uma faixa por nível de disparidade sendo que a faixa tem a altura igual a altura da janela + 1 e tem a largura da imagem de entrada, esta é a **faixa de agregação**.

Técnica das faixas: $FA = 4x(1+j)d$

3. Carregar os resultados da agregação de evidências

Para agregar os valores obtidos nos planos de custo de correspondência na abordagem tradicional, é necessário pelo menos mais um plano temporário com o mesmo tamanho de uma imagem de entrada (Figura 69), este plano armazena os resultados da agregação e após ser preenchido, sobrepõe o plano correspondente de custo de correspondência.

Figura 69 - Agregação de evidências usando plano temporário.

Não usar este plano temporário exigiria a alocação de um conjunto de planos igual ao DSI, isto faria com que a comparação com a técnica das faixas ficasse distorcida, logo aqui será considerado o uso deste plano temporário.

Abordagem tradicional: Plano temporário = $4xy$

Para a técnica das faixas é preciso alocar apenas uma linha para cada nível de disparidade esta linha tem a mesma largura que a imagem original. Esta linha é a **linha de seleção de disparidade**.

Técnica das faixas: Linha de seleção de disparidade = $4xd$

4. Carregar o Mapa de Disparidades

Para gerar o mapa de disparidades em ambas as abordagens é necessário o espaço de uma imagem (largura x altura), para cada pixel são necessários 4 bytes também, apesar do mapa de disparidades ser feito em níveis de cinza aqui será considerado que ambas as abordagens irão gerar o mapa de disparidades igualmente com imagens de 24 bits.

Ambas abordagens: Mapa de disparidades = $4xy$

Na Tabela 5 nota-se que para ler as imagens de entrada e para a construção do mapa de disparidades é ocupado o mesmo espaço de memória, a grande diferença se encontra no custo de correspondências (para a abordagem tradicional é o DSI e para a técnica das faixas são as **faixas de agregação (FA)**).

TABELA 5 - COMPARAÇÃO DAS FORMULAS PARA CADA ETAPA.

	Abordagem Tradicional	Técnica das Faixas
Ler Imagens	$8xy$	$8xy$
Custo de Correspondência	$4xyd$	$4x(1+j)d$
Agregação de Evidências	$4xy$	$4xd$
Mapa de Disparidades	$4xy$	$4xy$
Consumo Total	$16xy + 4xyd$	$12xy + 4xd + 4x(1+j)d$

Uma diferença menor ocorre na agregação de evidências, que na abordagem tradicional representa o tamanho de uma imagem (um plano xy float), mas para a técnica das faixas são as **linhas de seleção de disparidade**

(LSD) que se trata de uma linha com a largura da imagem (x) para cada um dos níveis de disparidade. Como os níveis de disparidade normalmente são menores que a altura (y) da imagem, logo para a agregação de evidências a técnica da faixas também irá consumir menos memória se comparada com a abordagem tradicional. A Tabela 6 apresenta uma simulação de consumo de memória para diversas resoluções.

Tabela 6 – Simulação de consumo de memória para as duas técnicas nas etapas otimizadas.

				Tradicional	Técnica das Faixas	
				Plano Temp.	DSI	LSD
				MB	MB	MB
x	y	d ¹	j ²	MB	MB	MB
320	240	12,8	7,2	0,293	3,750	0,016
640	480	25,6	14,4	1,172	30,000	0,063
800	600	32,0	18,0	1,831	58,594	0,098
1024	768	41,0	23,0	3,000	122,880	0,160
1600	1200	64,0	36,0	7,324	468,750	0,391
2592	1944	103,7	58,3	19,222	1992,904	1,025
3200	2400	128,0	72,0	29,297	3750,000	1,563
3600	2700	144,0	81,0	37,079	5339,355	1,978
4800	3600	192,0	108,0	65,918	12656,250	3,516
5200	3900	208,0	117,0	77,362	16091,309	4,126

Nesta dissertação foi focada a redução de consumo do DSI, pois o mesmo é amplamente utilizado e reconhecido com consumidor excessivo de memória, o que faz da técnica das faixas uma contribuição significativa. Por outro lado o uso do plano temporário para a agregação de evidências pode ser visto como um detalhe de implementação que poderia ser otimizado de outras maneiras, se este fosse o objetivo, por este motivo que a redução de consumo de memória proporcionada pela técnica das faixas ao utilizar as **linhas de seleção de disparidade** em comparação com o plano temporário foi apresentada somente neste anexo.

¹ A disparidade está mais ligada à distância entre as câmeras (mais detalhes no CAPÍTULO 2 no item 2.3). Foi adotado este percentual de 4% da largura, pois ele é bastante coerente se comparado com o valor usado em diversas imagens. Esta estimativa é necessária para o cálculo do consumo de memória para diferentes resoluções.

² O tamanho da janela tem mais relação com as características visuais das imagens de entrada, porém quanto maior a resolução de uma imagem maior a tendência de crescimento da janela. Foi adotado este percentual de 3% da altura, pois ele é bastante coerente se comparado com o valor usado em diversas imagens. Esta estimativa é necessária para cálculo de consumo de memória para diferentes resoluções.

APÊNDICE B – Documentação Publicada no *Site Source Forge*

Este apêndice tem como objetivo apresentar a documentação da maneira como está publicada no site Source Forge (o maior site sobre código aberto do mundo), por este motivo encontra-se em inglês e não segue a formatação desta dissertação.

2008 Marcelo Archanjo

marcelo_archanjo@yahoo.com.br - these source codes are under General Public License.

<https://sourceforge.net/projects/reconststereo>

3D Reconstruction using Stereo Vision How To

First Lesson – Simple Use

- Download the .ZIP file and extract it locally
In \StereoVision\Release execute StereoVision.exe
1- In the combo box:

Type: TSUKUBAL.BMP TSUKUBAR.BMP 16 1 17 I 64 64

- 2- Click in Load Images
- 3- Click in Generate
 - a. Will appear the disparity map (Static bitmap)
- 4- Click in 3D Interface
 - a. Will appear the 3D Interface (interactive 3d view)
 - b. To interact to 3D Interface use the keyboard:
 - i. Up arrow - function: move forward
 - ii. Down arrow - function: move backward
 - iii. Left arrow - function: turn left
 - iv. Right arrow - function: turn right
 - v. Alt + Left arrow - function: strafe left
 - vi. Alt + Right arrow - function: strafe right
 - vii. A - function: move up
 - viii. Z - function: move down
 - ix. Home - function: look up
 - x. End - function: look down
 - xi. Pgdn - function: center view

5- Click in Close

6- You can re-execute the program and use different images

```

VENUSL.BMP VENUSR.BMP 20 1 15 I 64 64
VASOL.BMP VASOR.BMP 40 3 21 I 64 64
SAWTOOTHL.BMP SAWTOOTHR.BMP 20 1 9 I 64 64
MAPL.BMP MAPR.BMP 30 1 9 I 64 64
CONESL.BMP CONESR.BMP 26 1 15 I 64 64

```

7- Execute the steps 1 to 5

Important: this program is a beta version and does not work well if you do not close after execution.

The images Tsukuba, Venus, Sawtooth, Map, Cones are from:

<http://vision.middlebury.edu/stereo/>

SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.

SCHARSTEIN, D.; SZELISKI, R. High-accuracy stereo depth maps using structured light. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.

The images Vaso were taken by the author (Marcelo Archanjo)

Second Lesson – Understand the arguments

VASOL.BMP VASOR.BMP 40 3 21 I 64 64

VASOL.BMP - Left image

VASOR.BMP – Right Image

40 - Maximum disparity

3 - Maximum line variation

21 - Window size

I - Type of search: I= color intensity / G= Gradient Operator

64 64 - 3D mesh resolution

The output of the program will be a bitmap with the disparity map and a VRML file with the 3D mesh.

Maximum disparity is the difference of column (x) for the same pixel of most near object to the camera.

Zoom of left image (VasoL.bmp) nearest point (251, 361)

Zoom of right image (VasoR.bmp), nearest point (211, 361)

The correct max disparity is $251 - 211 = 40$

Max disparity = 20

No comments

Max disparity = 50

Not wrong, but dark.

Max disparity = 40

Best result is achieved with max disparity= 40, the near objects are light.

Maximum line variation is the difference of line (y) for the same pixel of most distant object for the camera.

Max line variation = 1

Some examples of
Correspondence fail

Max line variation = 3

Window size (comparison kernel) is only accepted odd values, high frequency images can use short windows (3x3 or 5x5) these windows make the disparity map sharper.

For low frequency images (or high resolution images) is necessary to use big windows (21x21 or more) these windows make the disparity map smooth.

21 x 21 window

11 x 11 window

False correspondences

Gradient operator

Mesh resolution

64 x 64

16 x 16

Third Lesson – Disparity Map like Input

An easy way to correct fails, like false correspondence is use a image editor directly in the disparity map. After that you can use the disparity map like input, the arguments necessary are:

VASOTRUE.BMP VASOL.BMP 64 64

VASOTRUE.BMP - Disparity Map corrected

VASOL.BMP - Left image (to wrap in the 3D mesh)

64 64 - 3D mesh resolution

The output of the program will be a VRML file with the 3D mesh.

2008 Marcelo Archanjo

marcelo_archanjo@yahoo.com.br - these source codes are under General Public License.

<https://sourceforge.net/projects/reconststereo>

Tutorial 1

This tutorial provide a full description of the installation of the components necessary to change the source code of “3D Reconstruction using Stereo Vision”, available in Source Forge web site.

All software necessary is available free in the Microsoft® web site:

Visual Studio Express 2008

<https://www.microsoft.com/express/download/>

You can download only the Visual C++ 2008:

Or an ISO file with the full Visual Studio 2008 Express:

In this tutorial I will use the ISO image:
VS2008ExpressENUX1397868.iso

DirectX Software Development Kit

<http://www.microsoft.com/downloads/details.aspx?FamilyID=572be8a6-263a-4424-a7fe-69cff1a5b180&DisplayLang=en>

Download the file:

dxsdk_march2008.exe

Windows SDK for Windows Server 2008 and .NET Framework 3.5

<http://www.microsoft.com/downloads/details.aspx?FamilyId=F26B1AA4-741A-433A-9BE5-FA919850BDBF&displaylang=en#Overview>

Download the file:

6.0.6001.18000.367-KRMSDK_EN.iso

This links was accessed in April 2008.

1 - Visual Studio 2008 Express installation

Burn a DVD from the ISO image or access the data inside it, using your preferred tool:

VS2008ExpressENUX1397868.iso

Run: Setup.hta (if not start in auto run)

Click in Visual C++ 2008.

Click in Next.

Read the license terms and if you accept, select “I have read and accept the license terms”, then click in Next.

For the purposes of this tutorial is not necessary install the MSDN or Microsoft SQL. To preserve the disk space, deselect these items and click in Next.

Use the default installation directory, click in Install and wait the installation of the all necessary items.

Wait...

Click in Exit

Open once the Visual C++ 2008 Express Edition.

Now close the Visual C++ 2008 Express Edition.

2 – Installing the DirectX SDK

Run the program:

dxsdk_march2008.exe

Wait the Extracting process...

Click in Unzip

And wait...

Click in OK.

Will appear the following screen

Click in Next.

Read the license terms and if you accept, select “I accept the terms in the license agreement”, then click in Next.

Click in Next and wait

Wait...

Click in Finish

3 - Install the Windows SDK

Burn a DVD from the ISO image or access the data inside it, using your preferred tool:

6.0.6001.18000.367-KRMSDK_EN.iso

Run: Setup.Exe (if not start in auto run)

Click in Next

Read the End-User License Agreement and, if you agree, select “I Agree” and then click in Next.

Use the default installation locations and then click in Next.

Use the default installation options and then click in Next.

Click in Next and wait...

Wait...

Unselect “View the Windows SDK Release Notes.”
Click Finish.

3 – Check the Visual C++ directories

Open the Visual C++ 9.0 Express Edition

Click in Tools / Options

In Options / Projects and Solution / Click in VC++ Directories

In “Show directories for:” select Include files:

Add:

C:\Program Files\Microsoft DirectX SDK (March 2008)\Samples\C++\DXUT\Optional
C:\Program Files\Microsoft DirectX SDK (March 2008)\Samples\C++\DXUT\Core

These two directories are necessary (only see if is already added):

C:\Program Files\Microsoft DirectX SDK (March 2008)\Include
\$(WindowsSdkDir)\include

(If you change the directory during the setup here it will be different)

In “Show directories for:” select Library files:

These two directories are necessary (only see if is already added):

C:\Program Files\Microsoft DirectX SDK (March 2008)\Lib\x86

\$(WindowsSdkDir)\lib

(If you change the directory during the setup here it could be different)

Now you can close the Visual C++ 2008 Express Edition, all configuration is finished.

2008 Marcelo Archanjo

marcelo_archanjo@yahoo.com.br - these source codes are under General Public License.

<https://sourceforge.net/projects/reconststereo>

Tutorial 2

This tutorial provide a full description to generate new binaries using the source code of “3D Reconstruction using Stereo Vision”, available in Source Forge web site.

Open the Visual C++ 2008 Express Edition

Click in File / Open / Project/Solution...

Select the directory where you download the source code for Visual C++ 2008 Express Edition and select the file StereoVision.vcproj

Click in Open

Select in the left pane the Source Files / Main.cpp

Copy a stereo image pair (for example VasoL.bmp and VasoR.bmp in Release directory:

Run StereoVision.exe

Use the arguments

VASOL.BMP VASOR.BMP 40 3 21 I 64 64

Remember that these source codes are under GPL, and enjoy.