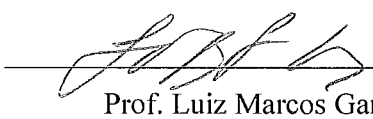
AS ABORDAGENS VOLUMÉTRICAS PARA O ALGORITMO ESTÉREO

Ítalo de Oliveira Matias

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

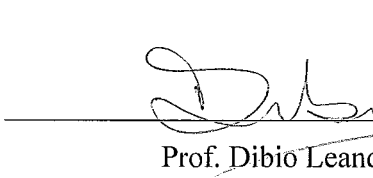
Aprovada por:

Prof. Antônio Alberto Fernandes de Oliveira , D.Sc.



Prof. Luiz Marcos Garcia Gonçalves , D.Sc.

Prof. Claudio Esperança, Ph.D.



Prof. Dibio Leandro Borges, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2001

MATIAS, ÍTALO DE OLIVEIRA

As Abordagens Volumétricas para o Algoritmo Estéreo [Rio de Janeiro] 2001

VII, 67 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2001)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Visão Estéreo
2. Algoritmo Cooperativo
3. Programação Dinâmica

I. COPPE/UFRJ II. Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

AS ABORDAGENS VOLUMÉTRICAS PARA O ALGORITMO ESTÉREO

Ítalo de Oliveira Matias

Março/2001

Orientador: Antonio Alberto Fernandes de Oliveira
Luiz Marcos Garcia Gonçalves

Programa: Engenharia de Sistemas e Computação

O objetivo do trabalho é descrever um algoritmo estéreo volumétrico para a obtenção de mapas de disparidade com a detecção de oclusões. O trabalho foi realizado no espaço lcd (linha x coluna x disparidade) em que a inicialização dos voxels é gerada pelo par estéreo a partir de uma medida de similaridade (fase de similaridade), e esses valores são refinados a partir de um processo iterativo no qual favorece os mais altos e inibe os mais baixos valores em uma mesma linha de visão (Fase de Competição). Para suspender o processo de atualização foi possível diminuir o tempo de performance sem influenciar no resultado final, obtido em todos os exemplos testados. Para gerar mapas de disparidade suaves é aplicada uma técnica de Programação Dinâmica.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

THE VOLUMETRIC APPROACH TO STEREO MATCHING

Ítalo de Oliveira Matias

March/2001

Advisor: Antonio Alberto Fernandes de Oliveira
Luiz Marcos Garcia Gonçalves

Department: Computing and Systems Engineering

The goal of this work is to describe a volumetric stereo algorithm for obtaining disparity maps and identifying the presence of occlusions. It works in the rcd (rows x columns x disparities range) space initializing the voxels with a measure of the similarity between the stereo pair which they represent (Similarity Phase), and refining these values through an iterative process which inhibits all voxels but one placed along the same line of sight (Competition Phase). By ceasing the update of little promissing voxels it was possible to enhance the time performance without influencing the results in all tested examples. Also, to get smoother maps, a Dynamical Programming process is applied to obtain the final solution.

Dedicatória:

Á Deus, Meus Pais, minha irmã Roberta e minha noiva Márcia

Agradecimentos:

Aos meus irmãos: João, Frederico, Antonio e Roberta.

Ao meu tio Paulo e aos meus sobrinhos: Rafael e Ana Carolina.

Ao meu orientador, Prof. Antônio Alberto Fernandes de Oliveira, pelo total apoio, incentivo e orientação em todo o tempo e pela confiança em mim depositada.

Ao co-orientador e amigo, Prof. Luiz Marcos Garcia Gonçalves, pela sua ajuda, compreensão e as valiosas contribuições para este trabalho .

Aos professores e amigos do LCG e da COPPE: Cláudio Esperança, Paulo Sérgio, Gilson Giraldi, Rômulo, Walter, Antônio, Mara, Nelma, Mexas, Fernando Wagner, José Valentim, Luis Fernando, Roque, Margareth, e outros.

Aos colegas de trabalho: Fábio Ponte, Edison Yoshino, Mauro Dias, Cleir Araújo, Cynthia Mattoso e Débora Jardim.

À PETROBRAS – Petróleo Brasileiro SA., representado pelo eng. eletrônico Paulo Roberto dos Santos Pereira.

À Empresa Tecnocoop Sistemas em nome de Henrique Pegado e Vitor Zenha.

À CAPES que contribuiu com parte do apoio financeiro.

Capítulo 1

Introdução

Em Visão Computacional, o processo de reconstrução estéreo também conhecido comumente como stereópsis, consiste na determinação da profundidade a partir de duas (ou mais) imagens de uma cena que contenham uma área de recobrimento em comum, obtidas por câmeras separadas espacialmente. A base fundamental da reconstrução estéreo é o processo de correspondência estéreo, mais conhecido como "matching", cujo resultado é a determinação do que se denomina de mapa de disparidade. Um mapa de disparidade contém o deslocamento de cada *pixel* em coordenadas de imagem, em relação aos seus correspondentes na outra imagem, dadas duas imagens de uma mesma cena com as características citadas acima. O mapa de disparidade é gerado a partir da determinação de pares de *pixels* homólogos, isto é, pares de *pixels* correspondente ao mesmo ponto objeto nas duas imagens da cena. A determinação de todos os pares de *pixels* homólogos nas imagens permitirá reconstruir as coordenadas tridimensionais das regiões na cena que originaram a projeção daqueles *pixels*, contanto que o processo de aquisição das imagens seja bem controlado. Geralmente, os métodos tradicionais aplicam técnicas de triangulação e integração numérica sobre o mapa de disparidade para gerar as coordenadas tridimensionais para todos os pontos da cena.

A determinação de *pixels* homólogos ocorre geralmente com a utilização de funções que medem a similaridade entre os *pixels* nas imagens, tais como o cálculo de valores de correlação ou outras medidas de similaridade e, geralmente a grande maioria dos métodos envolve o uso de operações de convolução. Operações desse tipo envolvem um tempo de execução alto e grande quantidade de memória. Assim, o processo de correspondência estéreo ou "matching" é o gargalo em aplicações computacionais que exigem o processo de reconstrução estéreo, devido à grande quantidade de cálculos necessários à determinação dos *pixels* homólogos. Neste

trabalho, tentamos diminuir este tempo de processamento, aplicando técnicas de programação dinâmica, visando acelerar o processo de determinação da correspondência.

Idealmente, o processo de reconstrução estéreo deve produzir um mapa de disparidade denso e que seja também suave e detalhado, o que geralmente não é possível na prática. Outro problema comumente encontrado nos métodos existentes, com umas poucas exceções, é que os algoritmos não tratam a de detecção de oclusão. Este problema é tratado no algoritmo oferecido por Kanade e Zitnick em [01] que será discutido em detalhes no próximo capítulo. Basicamente, esse algoritmo aplica um processo iterativo a partir de um mapa de disparidade inicial grosseiro que produz como resultado um mapa de disparidade com zonas de oclusões explicitamente determinadas.

Nossa proposta de trabalho consiste em introduzir melhorias no algoritmo proposto por Kanade e Zitnick em [01], visando acelerar o processo, sem nos preocuparmos tanto com os aspectos de detecção explícita de oclusão, mas sim com a determinação de um método que seja eficiente e robusto.

Como veremos, o algoritmo desenvolvido neste trabalho determina mapas de disparidade com um tempo bem aquém do algoritmo de Zitnick e Kanade em [01] e mantém ainda uma certa coerência e precisão na determinação de zonas de oclusões. No presente trabalho, buscamos exatamente definir regiões na imagem onde a disparidade possa ser determinada corretamente. Para gerar correspondências corretas, utilizamos uma função de atualização iterativa que utiliza duas hipóteses (unicidade e/ou continuidade) que serão introduzidas no próximo capítulo, num espaço tridimensional **LCD** (**linha** x **coluna** x **disparidade**), onde cada elemento é denominado de voxel.

Construímos uma matriz tridimensional de valores de similaridade no espaço da disparidade, onde cada elemento dessa matriz corresponde a um *pixel* na imagem de referência e uma disparidade relativa a outra imagem. A função de atualização consiste em determinar a disparidade dos outros pontos cuja disparidade não pôde ser corretamente determinada no mapa inicial, incluindo os denominados "pontos ocultos". Assim, ao invés de uma interpolação explicita, determinamos um mapa completo

posterior pelo espalhamento da disparidade na vizinhança de cada ponto que possua os valores da função de similaridade corretamente determinados. Para o nosso algoritmo os *pixels* em zonas de oclusão são determinados a partir de valores muito baixos dessa função de atualização.

Assim, o nosso principal objetivo é propor melhorias no algoritmo do Kanade [01], usando técnicas distintas para acelerar a função de atualização e a obtenção de melhores resultados. Conseguimos introduzir melhorias significativas no tempo de processamento do algoritmo deles aplicando técnicas de Programação Dinâmica (PD) para fornecer uma convergência rápida a valores finais, refinados. Convém ressaltar que a detecção de oclusões de forma explícita ocorre em nosso algoritmo, ainda que não com a mesma exatidão que no algoritmo deles [01], porém o ganho em tempo de processamento é da ordem de duas a cinco vezes (mais rápido), segundo os experimentos conduzidos que serão apresentados no decorrer do trabalho.

Para testar o nosso algoritmo, testamos exaustivamente usando vários pares de imagens estéreo. Para mostrar os resultados, escolhemos alguns pares cedidos cordialmente pelo C. Lawrence Zitnick [01] e por Gonçalves e Oliveira [02], que serão vistos no decorrer do trabalho. Os mapas de disparidade resultantes são suaves e detalhados, e poderão ser notadas zonas de oclusões explicitamente determinadas.

1.1 Organização do Trabalho

Além desta breve introdução, o restante desta dissertação está organizado de maneira apresentada a seguir:

- **Capítulo 2 - Premissas e trabalhos relacionados**

Neste capítulo, descrevemos as premissas do processo de reconstrução estéreo, definindo o que é disparidade e mapa de disparidade, além de descrevermos sobre os tipos e os principais métodos de reconstrução estéreo existentes na literatura. Ao final, introduzimos brevemente nossa proposta.

- **Capítulo 3 – O algoritmo**

Neste capítulo, descrevemos sucintamente a formulação do problema de reconstrução estéreo num espaço LCD (linhas x colunas x disparidade) formulada por Zitnick e Kanade em [01] e introduzimos detalhadamente nossa proposta, com uso de programação dinâmica.

- **Capítulo 4 – Experimentos e resultados**

Neste capítulo, mostramos a aplicação prática do algoritmo proposto a vários pares de imagens estéreo, com características diferentes. Fazemos comparações com o algoritmo originalmente introduzido por Zitnick e Kanade, para cada tipo de experimento.

- **Capítulo 5 – Conclusão e Trabalhos Futuros**

Neste capítulo falamos sobre como foi o nosso trabalho proposto, incluindo algumas dificuldades de implementação, e introduzimos algumas propostas do que poderia ser feito para melhorá-lo a posteriori.

Capítulo 2

Premissas e trabalhos relacionados

As técnicas de visão computacional visam a obtenção de informação sobre objetos, ambientes ou cenas, a partir de dados digitalizados dos mesmos. Normalmente, essa informação é obtida com o uso de algoritmos computacionais (baseados em formas matemáticas ou estatísticas) que operam sobre dados digitais, captados a partir de sensores. Um exemplo típico é o uso de imagens que podem ser obtidas a partir de câmeras filmadoras, equipamentos de ressonância magnética e de ultra-sonografia.

Um sistema de aquisição de imagens possui dispositivos sensíveis à luz refletida ou emitida pelos objetos. Esses (foto)-sensores determinam valores para a luminância de pequenas regiões da superfície. Os valores de luminância são geralmente quantizados através da divisão do espaço de percepção em vários níveis, cuja quantidade depende da capacidade de sensibilidade do dispositivo. Para sistemas monocromáticos os níveis de intensidade da luminância são normalmente denominados de níveis de cinza ou tons de cinza, sendo 256 níveis suficientes para a maioria das aplicações. Geralmente estas pequenas regiões são definidas por formas retangulares ou quadrados agrupados lado a lado num plano de projeção, sendo a imagem resultante composta por uma matriz 2D. O *pixel* (de "pixmap element") é a denominação usual para cada pequena região deste plano. Assim, cada *pixel* nada mais é que uma codificação ou uma quantificação dos atributos de luminância, cor, ou brilho de uma pequena porção da superfície de um objeto discretamente amostrada.

De forma análoga à visão biológica, o processamento das imagens fornecidas por sistemas de aquisição em geral é complexo. O processamento a tempo real ainda não está bem resolvido, sendo o ponto crítico no projeto de protótipos de visão computacional e robótica.

Características de objetos e cenas, representativas da sua textura, forma e posição, podem ser extraídas das imagens, pela sua análise, através da manipulação computacional dos valores de níveis de cinza. Com estas características torna-se possível construir um modelo computacional de objetos ou cenas representadas pelos dados discretos contidos nas imagens, a partir do qual outras tarefas de mais alto nível, tais como identificação ou outras envolvendo decisão, possam ser executadas. Este processo é denominado usualmente de "reconstrução", não obstante o tipo de característica usada.

Fatores como a posição, a forma dos objetos e a direção de iluminação influem no processo de obtenção dos dados sensoriais referentes a um objeto em uma cena imageada. Considerando estes e também outros fatores como a distância a que se encontram os objetos, Marr [03] propôs em 1975 um caminho para a reconstrução e identificação de objetos a partir de imagens, resumido a seguir:

a) Retirada ou realce de componentes característicos das imagens para estimar a orientação da superfície do objeto e recuperação da forma através de métodos que utilizam o sombreamento, padrões de textura e detecção de contornos, resultando em uma função que Marr denominou de "esquema 2D e meio". O resultado desta etapa normalmente é algo intermediário, como um mapa de disparidade ou um diagrama de agulhas (normais em cada ponto), que será usado posteriormente para geração ou definição da terceira dimensão.

b) Segmentação do resultado obtido na etapa anterior, gerando uma imagem simbólica. Nesta etapa, usando propriedades inerentes aos objetos e técnicas de integração numérica, é reconstituída a terceira dimensão, tendo geralmente como resultado um mapa relativo de alturas, uma imagem de alturas ou uma imagem simbólica representando a reconstrução da profundidade.

c) Re-segmentação com a medição das propriedades. Elementos com mesmas propriedades são agrupados, sendo identificadas regiões com mesmo padrão, elementos lineares, cantos, ou outras características. Esta re-segmentação será usada com técnicas de agrupamento perceptual para a etapa seguinte, de identificação ou reconhecimento.

d) Reconhecimento, comparando com padrões, usando restrições. Os elementos re-segmentados são comparados com modelos preexistentes de objetos, sendo identificados.

Os problemas de visão computacional podem ser divididos em algumas subclasses de processamento, definidas pelo tipo da informação recebida e fornecida pelo sistema. Uma classe particular toma informações sobre a intensidade de uma imagem e fornece informações sobre as propriedades de superfícies visíveis que afetam diretamente aquela intensidade. Esta classe de problemas é muitas vezes referida como visão primitiva e pode ser dividida de acordo com o mecanismo físico distinto pelo qual as propriedades de superfícies visíveis podem afetar a intensidade das imagens. Alguns exemplos dessa classe são os problemas de reconstrução da forma conhecida na literatura como problemas do tipo "shape from x", onde x pode ser (binocular) estéreo, estéreo-fotométrico, movimento, sombreamento e, ainda, com o uso de cor.

Neste trabalho, tratamos especificamente o problema de "shape from stereo", onde o objetivo é a reconstrução tridimensional de uma cena a partir de imagens estéreo da mesma, ou seja, duas (ou mais) imagens que contenham uma área de recobrimento em comum (mesma cena), obtidas por sistemas de aquisição com geometria (mais ou menos) controlada, a partir de pontos de vista diferentes. Um exemplo bem usual e comum é o emprego de imagens aéreas contendo detalhes do terreno, obtidas por uma câmera embarcada em um avião. Seqüências de imagens estéreo deste tipo são muito usadas em processos de restituição digital ou aerofotogrametria para fins de mapeamento.

De um modo geral, os algoritmos de reconstrução a partir de imagens estereoscópicas consistem em três passos:

a) extração ou realce de feições ou características das imagens;
b) estabelecimento de correspondência (matching) entre as feições extraídas;
c) reconstrução tridimensional.

Em outras palavras, numa primeira fase, ocorre o realce de características com o uso de filtros específicos ou processos de realce. Então, para cada par de pontos correspondentes determinados pelo processo de correspondência estéreo, a disparidade

pode ser calculada e coordenadas tridimensionais para o ponto objeto correspondente podem ser reconstruídas usando equações de inversão de projeção que serão apresentadas a seguir.

2.1 Reconstrução Estéreo

A obtenção de visão 3D a partir de seqüências de fotografias, com as fotos contendo imagens de cenas reais, idealizou-se no final do último milênio, com a invenção da máquina fotográfica. O modelo utilizado é semelhante ao olho humano. De forma similar ao modelo biológico, a partir das diferenças de posição das projeções de pontos nas imagens, por triangulação, pode-se determinar a posição relativa de um ponto objeto. Se o sistema estiver calibrado (conhecidas localizações de alguns poucos pontos objetos nas imagens), isto permite a reconstituição das condições (posições absolutas) que as câmeras possuíam no momento da aquisição ou tomada das imagens (a este processo se denomina usualmente de orientação absoluta ou externa). Desta forma, pode-se determinar a distância a que outros pontos objetos se encontram do sistema de aquisição e a conseqüente reconstrução tridimensional da superfície ou cena em questão.

A partir do final do penúltimo e do início do último século, surgiram as técnicas fotogramétricas convencionais que passaram a aplicar esses princípios de reconstrução estéreo para a obtenção de informação a partir de seqüências de fotografias da superfície do globo terrestre, obtidas por uma câmera aero-transportada, selecionadas duas a duas, visando o mapeamento da superfície do terreno em questão para fins cartográficos. Essas câmeras fotográficas com propósito específico foram transportadas inicialmente em balões e depois em aviões, após a invenção destes últimos. Com o avanço tecnológico, o surgimento dos computadores e as técnicas de digitalização de imagens, na década de 70, essas técnicas fotogramétricas convencionais passaram a sofrer processos de automatização, surgindo processos automáticos de reconstrução a partir de seqüências de imagens estéreo ("shape from stereo"). O objeto principal de automatização ou gargalo foi o estabelecimento do processo de correspondência estéreo (matching) para que, a partir do qual, se pudesse obter a terceira dimensão (profundidade). Convém notar que na maioria dos esforços realizados nesse sentido, de

mesma forma que no processo convencional, também eram usadas duas imagens digitais tomadas de uma mesma cena e obtidas de pontos de vista diferentes. Ainda, surgiram também outros processos automáticos como, por exemplo, a identificação e/ou o reconhecimento automático a partir de modelos dos objetos presentes na cena. Estas técnicas de reconstrução estão bem definidas, possuindo vasta bibliografia em artigos e livros texto como Vision [03], Robot Vision [04], From Images to Surfaces [05] e Computer Vision [06]. A seguir, daremos uma repassada no processo de reconstrução estéreo.

A menos de distorções ou erros devidos ao processo de aquisição, pode-se considerar uma imagem como uma transformação projetiva (projeção radial) de um conjunto de feições ou características no espaço (R^3) em um plano (R^2). É um esquema discreto, refletindo ou representando os objetos contínuos na cena. Este esquema pode ser considerado como uma amostra estatística controlada, uma vez que as projeções de pontos nas imagens guardam uma relação bem definida com pontos na cena. Desta forma, pode-se estabelecer algumas restrições e implementar algumas regras de reconstrução, com a determinação dos parâmetros das transformações inversas, de modo que se possa mapear pontos de imagem (ou *pixels* 2D) em seus correspondentes na cena (regiões 3D), reconstruindo a forma dos objetos em três dimensões a partir de suas projeções nas imagens. Estas aplicações, cujos parâmetros são procurados, podem ser aproximadas por transformações lineares, sendo elas basicamente transformações de rotação, translação e de escala. Uma vez determinados os coeficientes destas transformações e conhecidas as posições de objetos nas imagens, pode-se determinar suas posições na cena.

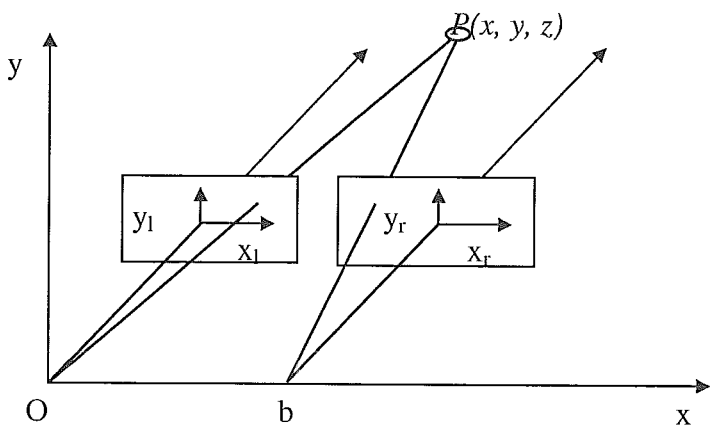


Figura 2.1 - Modelo estéreo

A Figura 2.1 representa um modelo estéreo não convergente, no qual cada plano de imagem está rebatido pelo seu ponto focal (centro de perspectiva). O sistema visual representado na Figura 2.1 está referenciado a um sistema de coordenadas de mundo **XYZ** (sistema de mão esquerda), sendo (x, y, z) as coordenadas tridimensionais de um ponto neste sistema. A imagem esquerda tem a origem do seu sistema de coordenadas (x_l, y_l) nas coordenadas de mundo (0, 0, f) e seu ponto focal correspondente em (0, 0, 0). De forma similar, o sistema de coordenadas da imagem direita (x_r, y_r) tem sua origem em (b, 0, f) e seu ponto focal correspondente em (b, 0, 0). A distância b do segmento que une os pontos nodais que se localizam nos centros do conjuntos de lentes das duas câmeras é também chamada de linha de base e a distância focal f é a distância do ponto nodal ao plano imagem em cada sistema de aquisição. Considera-se que é a mesma para ambos. Convém ressaltar que o sistema global definido aqui poderia ter uma origem qualquer, e que assim o problema poderia ser generalizado para situações não tão ideais como se vê na Figura 2.1, porém com um certo controle geométrico.

2.2 Cálculo da Disparidade e Profundidade

Em um sistema de aquisição bem controlado, cada ponto-objeto que esteja dentro do campo de vista é radialmente projetado, através do ponto nodal (ou centro ótico) de cada uma das câmeras, nos planos de imagem esquerdo e direito. Na Figura 2.1, pode ser visto um ponto típico P(x, y, z) e suas projeções, que, neste exemplo, são visíveis nas duas imagens. Considerando-se os princípios básicos de geometria, em particular a semelhança de triângulos, as equações de projeção perspectiva de um dado ponto, a menos de erros devido à distorção e outros erros inerentes ao processo de aquisição, podem ser aproximadas por:

$$x_l = \frac{fx}{z}$$

$$x_r = \frac{f(x-b)}{z}$$

$$y_l = y_r = \frac{fy}{z}$$

Estas três equações podem ser manipuladas e invertidas para se encontrar x, y e z. Sendo a disparidade definida como $d = x_l - x_r$, as equações de inversão da projeção perspectiva podem ser escritas como:

$$x = \frac{bx_l}{d} \qquad (2.1)$$

$$y = \frac{by_l}{d} \qquad (2.2)$$

$$z = \frac{bf}{d} \qquad (2.3)$$

Estas equações proporcionam a base para determinar a estrutura tridimensional de uma cena a partir de imagens estéreo da mesma.

A equação encontrada para determinar a incógnita z mostra que esta é inversamente proporcional à disparidade *d* e diretamente proporcional ao comprimento b da linha de base. Assim, fixado um erro na determinação da disparidade, a precisão na determinação da profundidade *z* cresce de forma direta com *b*. Porém, com o crescimento de b, mesmo com a existência de movimentos de vergência do conjunto, as imagens tendem a ser muito diferentes uma da outra, ou seja, um ponto que é visível em uma imagem pode não o ser na outra, causando problemas na determinação da correspondência. Uma coerência deve ser procurada visando encontrar a melhor relação entre estes fatores. Verifica-se também, da relação 2.3 que a disparidade é proporcional à distância focal *f*. Na prática, à medida que *f* cresce, as imagens também crescem, aumentando a distância do ponto projetado nas imagens ao centro destas e em conseqüência a disparidade.

2.3 Correspondência estéreo (matching)

Na reconstrução a partir de imagens estéreo, o processo fundamental é a identificação, nas duas imagens, das projeções correspondentes a um mesmo ponto objeto. Esta correspondência é também conhecida como "matching" de elementos, sendo ainda hoje o gargalo em sistemas artificiais de visão ativa.

Algumas pesquisas realizadas nos anos 70 [Julesz 72, Marr 79], usando estereogramas de pontos randômicos (aleatórios), mostraram claras evidências da existência de uma fase onde se dá o estabelecimento de correspondências entre os objetos presentes nas imagens formadas nas duas retinas, não estando ainda hoje bem claro como isto é realizado ou a que precisão. Computacionalmente, a correspondência estéreo pode ser calculada com utilização de vários tipos de processos, algébricos ou estatísticos [07], incluindo correlação de áreas ou de características (ou elementos), relaxação com uso de diferenças de níveis de cinza entre elementos vizinhos, e uso de programação dinâmica com detecção de arestas determinadas por operadores morfológicos. Normalmente, as imagens são pré-filtradas, usando técnicas que podem ser encontradas em [08], visando eliminação dos efeitos das altas freqüências e visando realçar ou separar algum tipo de característica. Após esta filtragem, é então calculado o matching. Atualmente, com as tecnologias paralelas, o processo de matching pode ser calculado em tempo real [09,10,11,12,13,14], ou seja, à medida que as imagens são adquiridas. Após o estabelecimento da correspondência, poderão ser usadas as técnicas de triangulação e integração numérica para a reconstrução da profundidade e conseqüente determinação da forma de objetos.

Note que o ideal é estabelecer a correspondência para todos os *pixels* das imagens. Na prática, isto não ocorrerá. Geralmente não é possível relacionar todos os pontos devido a oclusão ou ambigüidade de elementos que podem ocorrer entre uma imagem e outra. No final do processo, geralmente se tem a determinação de correspondência para vários elementos, a partir dos quais podem ser interpolados valores de profundidade para o restante.

Como vimos acima, a diferença de disparidade entre *pixels* vizinhos nas imagens, representando a diferença de altura entre dois pontos, é o princípio básico da reconstrução estéreo. Com a determinação da disparidade em toda a imagem (mapa de disparidade), pode-se então construir um modelo 3D para a superfície ou cena que deu origem às imagens. No mapa de disparidade, para cada *pixel* de uma imagem encontra-se o deslocamento relativo ao seu correspondente na outra imagem.

Determinar o mapa de disparidade não é trivial. Podemos pensar, a princípio, em determinar detalhes que sejam inconfundíveis nas imagens, tais como contornos de

objetos, certos ângulos, linhas, etc., em uma imagem e tentar sua localização na outra, ou usar as diferenças observadas nos tons de cinza entre *pixels* vizinhos (textura) para tentar estabelecer a correspondência. Por outro lado, os valores da luminância dos *pixels* correspondentes a um mesmo ponto na superfície podem ser diferentes nas duas imagens. Isto pode ocorrer devido a vários fatores, como diferenças observadas na determinação dos valores discretos (quantizados) da luminância em posições diferentes do plano de projeção, características diferentes entre sistemas de aquisição, diferentes pontos de vista com que as imagens são obtidas gerando diferentes ângulos de iluminação, distorções ocorridas no processo de aquisição, má localização dos elementos ou até mesmo ruídos, o que é muito comum em sistemas de aquisição devido ao tempo de uso. Uma das piores situações que pode ainda ocorrer é a ocultação de um elemento numa das imagens. Nesse caso, não será possível determinar o correspondente. Todos esses problemas devem ser tratados de forma a se evitar ou minimizar os seus efeitos prejudiciais ao processo de reconstrução.

Há várias maneiras de se restringir o problema de correspondência entre duas imagens estereoscópicas. Geralmente, leva-se em consideração o conceito de linhas epipolares, procurando as correspondências entre primitivas em locais mais ou menos definidos, estabelecendo assim restrições que facilitam uma solução para o problema. Em um sistema de aquisição com duas câmeras, as linhas epipolares correspondentes a um ponto objeto, nas duas imagens, são dadas pela interseção de cada plano de imagem com o plano determinado pelo ponto objeto e pelos dois centros de projeção (centros óticos ou pontos nodais de cada câmera). Conseqüentemente, a restrição de epipolaridade diz que se um ponto é visível nas duas imagens, suas projeções deverão estar sobre as linhas epipolares correspondentes. Esta condição (determinação das linhas epipolares) fica assegurada após a orientação interna (ou relativa) do conjunto, que restabelece as posições relativas que o sistema possuía no momento de tomada das imagens. Atualmente, há vários exemplos de sistemas (cabeças estéreo) que possuem os eixos óticos no mesmo plano (e/ou em paralelo), permitindo que as linhas epipolares sejam paralelas ao eixo X nas imagens (direção horizontal). Assim, a busca pelo correspondente de um *pixel* é executada em uma única direção.

2.4 Classificação dos métodos de matching

Podemos classificar os métodos de matching existentes segundo duas dicotomias. Na primeira, temos os métodos baseados em áreas versus os métodos baseados em elementos (ou características), e na segunda, temos os métodos diretos (baseados em correlação) versus os métodos iterativos (baseados em relaxação). Convém notar que pode haver quaisquer combinações entre as duas dicotomias. Por exemplo, pode ser usada uma técnica de relaxação baseada em área ou baseada em elemento. Neste último caso, deverão ser levadas em consideração restrições de diferenciabilidade para os *pixels* vizinhos dos *pixels* componentes do elemento.

2.4.1 Matching baseado em áreas

A idéia principal dos métodos baseados em área é que se realize o matching para todos os elementos componentes das imagens, e não apenas para um subconjunto de elementos bem definidos, tais como arestas ou cantos. Ao final do matching deverão estar determinados valores de disparidade em todos os locais das imagens (exceto onde não se tenha correspondência devido à oclusão ou ruídos). Os métodos baseados em área normalmente consideram cada *pixel* da imagem ou a média de um conjunto de *pixels* para o matching. Podem ser usadas tanto técnicas de correlação quanto de relaxação. Não são realizadas operações para detecção explícita de arestas, cantos ou outras estruturas bem definidas. O que pode ocorrer geralmente é uma pré-filtragem das imagens visando apenas realçar estes elementos.

Idealmente, para se evitar um processamento posterior para interpolação ou espalhamento da disparidade, um algoritmo de visão estéreo deve produzir um mapa de disparidade denso, com esta última determinada em todos os *pixels* nas imagens. Temos que levar em consideração ainda que este mapa de disparidade resultante deve ser suave (contínuo) e detalhado. Ou seja, objetos ou regiões na cena que possuam uma superfície contínua devem gerar uma região no mapa de disparidade onde esta última varia de forma suave, enquanto pequenos elementos de superfícies, onde ocorre a descontinuidade da disparidade, devem ser detectados como pertencendo a intervalos

entre regiões separadamente distintas (detalhes). Na prática, é muito difícil senão impossível para um algoritmo estéreo satisfazer a hipótese de suavidade ou de continuidade e ao mesmo tempo representar a cena com detalhes. Algoritmos que gerem mapas detalhados tendem a ser ruidosos, perdendo a suavidade ou continuidade, e algoritmos que gerem mapas suaves tendem a perder detalhes da cena.

Um dos problemas principais para métodos baseados em área como em Horn [04], Gonçalves e Oliveira [02], Wood [15] e Kanade & Okutomi [16], é a seleção de uma janela com dimensões apropriadas. A escolha das dimensões deve ser tal que se consiga obter um mapa de disparidade suave e detalhado. Note que isto depende também da variação local da intensidade dos *pixels* (textura) nas imagens. Em geral, em áreas com boa textura, uma pequena janela é suficiente para manter a suavidade. Em áreas de baixa textura, contudo, uma janela maior é necessária para conter bastante variação de intensidade, e assim poder obter uma correspondência confiável. Ainda, ocorrem problemas quando a disparidade varia dentro da janela, isto é, a superfície correspondente não é frontal-paralela. Neste caso, valores de intensidade em píxels dentro da janela podem não ter um correspondente, devido à distorção projetiva. Muitas tentativas foram feitas para tentar contornar este problema que ocorre nos métodos baseados em área. Em Panton [17], a janela de busca é deformada de acordo com a orientação estimada da superfície para reduzir o efeito da distorção projetiva, porém este método não trata a oclusão de bordas. Um método mais recente e sofisticado é o método que usa janelas com dimensões adaptativas elaborado por Kanade & Okutomi [16]. O tamanho e a forma da janela são modificados iterativamente, baseados na variação local e em estimativas correntes da profundidade. O problema de métodos com janela adaptativa é que eles são computacionalmente caros.

Um problema que é considerado fundamental nos métodos baseados em área, descritos acima, é que eles tomam decisões muito localmente e não consideram o fato de que a correspondência de um ponto restringe globalmente outras correspondências, o que é um resultado da geometria inerente ao processo estéreo e também da consistência da cena.

2.4.2 Matching Baseado em Elementos

Em métodos baseados em elementos, tenta-se encontrar em uma das imagens elementos bem definidos que possuam características semelhantes a outros na imagem a corresponder. Isto pode ser feito através do cômputo de valores de correlação ou de outra medida qualquer de similaridade. Estes elementos podem ser *pixels* ou grupos de *pixels*, formando objetos bem definidos tais como arestas, linhas ou regiões cuja estrutura possua determinadas características. Geralmente, numa fase de pré-processamento, as imagens são filtradas usando-se operadores morfológicos para realçar os elementos desejados, e logo após são aplicados operadores de detecção para determinação dos elementos. No trabalho de Ohta e Kanade em [18], foi realizada uma correspondência entre arestas com uso de uma técnica de programação dinâmica para diminuir a complexidade do processo. Este método provê ótimos resultados para a determinação da disparidade, sendo muito preciso, porém, o mapa de disparidade calculado após a determinação da correspondência não é completo, sendo necessárias interpolações para uma densificação desse mapa.

O problema fundamental dos métodos baseados em elementos é a geração de um mapa de disparidade esparso, sendo necessário realizar interpolações de valores para densificar ou completar o mapa. Isto poderá gerar um modelo tridimensional ideal que não represente fielmente a superfície ou cena amostrada, além da necessidade da fase posterior de interpolação. Outro problema refere-se à necessidade de um pré-processamento para extração dos elementos desejados. Numa comparação entre várias implementações realizadas para testar vários tipos de métodos estéreos baseados em áreas X elementos, Gonçalves e Oliveira mostram em [02] que a precisão dos métodos baseados em elementos não compensa a perda de tempo, não sendo uma prática comum o uso destes em aplicações de tempo real e visão robótica. Porém, há que se notar que são precisos, sendo uma boa alternativa se a aplicação é para fins de mapeamento ou uso posterior.

2.5 Tratamento de oclusões e descontinuidades da disparidade

Como vimos nas seções anteriores, estabelecer um mapa de disparidade completo não é trivial devido a possíveis oclusões e erros sistemáticos que podem ocorrer na aquisição e digitalização das imagens. Ainda, isto depende do tipo de método usado, se for um método baseado em elementos, obtém-se um mapa esparso de disparidade. Na prática, o processo de matching deve tentar determinar um conjunto máximo de pontos homólogos e, a partir da disparidade destes, usar algum processo de espalhamento para determinar a disparidade aos pontos vizinhos que não puderam ser correspondidos.

Muitos trabalhos realizados em visão estéreo nas últimas décadas foram guiados pela clara percepção de superfícies completamente preenchidas, notadas quando se olha para os estereogramas, mesmo quando os pontos são esparsos. Julesz, em [19], para criar superfícies mais desafiantes, exibiu estereogramas com arestas aneladas ou agudas em profundidade, conjecturando que nestes casos o processo estéreo era realizado em resolução espacial muito fina. Isto resultou numa grande atividade de experimentos e pesquisas realizadas na década de 80 no sentido de desenvolver técnicas de interpolação que fossem capazes de tomar medidas esparsas de disparidade, em elementos bem definidos tais como arestas, e preenchê-las para criar um mapa de disparidade suave com descontinuidades agudas, tal qual se vê nos estereogramas. Convém ressaltar que este não é o único caminho para resolução do problema de visão estéreo e que não está claro, pelas experiências realizadas, a existência de uma representação preenchida no sistema humano.

Nishihara, em [10], adotou posição contrária, concluindo que a percepção estéreo, surpreendentemente, tem uma resolução espacial pobre, mas uma tolerância a ruídos excelente. Neste modelo alternativo, a percepção de arestas agudas em profundidade pode ser explicada como uma ilusão criada ou favorecida pela presença de fronteiras com luminância diferente, o que ocorre muito em visão colorida. A percepção compelida de uma superfície preenchida é explicada simplesmente pela ligação de uma variável de estado (flag) indicando consistência com uma superfície suave. Um exemplo de tal indicação pode ser um pico de correlação alto e agudo no processo estéreo. Não parece ser necessária uma representação de superfície preenchida (um modelo completo) para explicar a performance psicofísica. Esta posição reforça a idéia da correspondência baseada em áreas, em localizações esparsas, resultando numa ferramenta de medida mínima.

Note que em certos casos, ao invés do cálculo de um modelo para a descrição completa de uma superfície visível, pode ser plausível realizar algumas medidas simples em poucas localizações de uma cena, sendo estas medidas mais do que suficientes para prover informação a ser usada na realização de uma dada tarefa.

Nesse sentido, o algoritmo oferecido por Kanade e Zitnick em [01] determina a correspondência em vários pontos e depois usa uma função de espalhamento para determinar um mapa completo, com detecção explícita de áreas onde ocorre a oclusão de píxels em uma das imagens. Basicamente, a partir de um mapa de disparidade inicial não totalmente preenchido, isto é, um mapa que pode ser obtido por um processo qualquer e onde nem todos os pontos possuam valores da disparidade conhecidos, aplica-se um algoritmo iterativo que produz como resultado um mapa de disparidade onde se possa detectar zonas de oclusões explicitamente. Convém ressaltar que no algoritmo deles, descrito em [01], o processo de determinação do mapa inicial deve fornecer um mapa que contenha valores de similaridade corretos para os *pixels* que não estejam ocultos, o que não é trivial. Geralmente, ocorrem problemas ao se usar quaisquer métodos, seja baseado em áreas ou baseado em elementos. Outro problema encontrado no algoritmo [01] refere-se ao tempo de processamento gasto (número de iterações necessárias) no refinamento do mapa de disparidade inicial (da ordem de 30 minutos, numa estação Indigo 2ex da Silicon Graphics com um processador de aproximadamente 150 Mhz).

Neste trabalho, buscamos exatamente definir regiões na imagem onde a disparidade possa ser determinada corretamente e a partir dessas correspondências corretas, usamos um método iterativo para determinar a disparidade dos outros pontos cuja disparidade não pode ser corretamente determinada no mapa inicial, incluindo os denominados "pontos ocultos". Assim, ao invés de uma interpolação explícita, determinamos um mapa completo posterior pelo espalhamento da disparidade na vizinhança de cada ponto que possua valores de uma função de similaridade corretamente determinados. Mais especificamente, nosso propósito é introduzir melhorias no algoritmo proposto por Kanade em [01], porém sem nos preocuparmos com os aspectos de detecção explícita de oclusão, mas sim com a determinação de um método que seja eficiente e robusto. Os experimentos realizados demonstram que o

algoritmo que será descrito nos próximos capítulos possui robustez suficiente e também velocidade compatível para ser aplicado em sistemas de tempo real.

Capítulo 3

Algoritmo estéreo

O objetivo principal do presente trabalho é o desenvolvimento de um algoritmo estéreo para a obtenção de mapas de disparidade, a partir de pares de imagens estéreo de uma cena, que seja ao mesmo tempo eficiente e robusto. Como visto no capítulo anterior, no algoritmo de Zitnick e Kanade, descrito em [01], a partir de um mapa de disparidade inicial não totalmente preenchido, isto é, um mapa que pode ser obtido por um processo qualquer e onde nem todos os pontos possuam valores da disparidade conhecidos, aplica-se um algoritmo iterativo que produz como resultado um mapa de disparidade onde se possa detectar zonas de oclusões explicitamente. Um problema encontrado naquele algoritmo [01] refere-se ao tempo de processamento gasto (número de iterações necessárias) no refinamento do mapa de disparidade inicial. Em linhas gerais, o algoritmo aqui proposto segue a mesma metodologia (a partir de um mapa grosseiro inicial, refinar iterativamente até atingir padrões satisfatórios).

Basicamente, propomos melhorias no algoritmo deles, usando técnicas distintas para acelerar o processamento, obtendo assim bons resultados. A diferença básica é que são aplicadas técnicas de Programação Dinâmica (PD) para fornecer uma convergência rápida a valores finais, refinados. A detecção de oclusões de forma explícita não ocorre com o mesmo grau de precisão, porém o ganho em tempo de processamento é da ordem de 2 a 5 vezes, segundo os experimentos conduzidos que serão apresentados no decorrer do trabalho.

Neste capítulo, apresentamos as premissas que servirão de base para a nossa proposta de melhoria no algoritmo estéreo. Ele está organizado em cinco seções principais onde os seguintes tópicos são apresentados e discutidos: algoritmo estéreo cooperativo, o sistema **XYZ** e o sistema **LCD**, suporte local, valores de similaridade, detecção de oclusão e programação dinâmica.

3.1 Algoritmo Estéreo Cooperativo

Marr e Poggio [20,21] apresentaram duas restrições básicas, levadas em consideração no desenvolvimento do algoritmo de visão estéreo deles. A primeira restrição diz que só pode existir um único correspondente para cada *pixel*. Em outras palavras, isto significa que cada *pixel*, em cada imagem, corresponde a um ponto (objeto) único na superfície representada nas imagens em questão. Note que isto é de certa forma dependente do grau de opacidade da cena. Para cenas com superfícies ou objetos transparentes, por exemplo, esta condição pode não ser respeitada. Em imagens desse tipo, ao usarmos o valor da intensidade dos *pixels* em uma determinada vizinhança para tentar, por similaridade, encontrar o correspondente de um determinado *pixel*, pode ocorrer que mais de um *pixel* atenda aos critérios de similaridade. A segunda restrição é que os valores da disparidade são geralmente contínuos, isto é, a disparidade varia, globalmente, de forma suave e pode ser considerada constante em uma vizinhança local. Nota-se que a hipótese da continuidade da disparidade é válida de uma maneira geral (na maioria das cenas). Geralmente, as descontinuidades da disparidade ocorrem somente na região da borda dos objetos. Assim, nestes locais de bordos, deve-se tomar alguns cuidados com relação a esta última restrição ou hipótese.

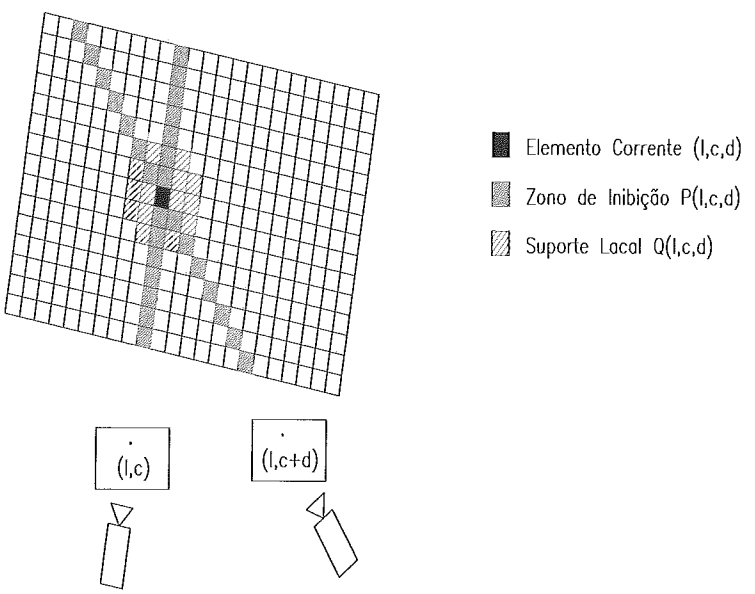


Figura 3.1 – Projeção Estéreo

No presente trabalho, propomos uma técnica que utiliza as duas hipóteses acima descritas, num espaço tridimensional ***LCD*** (**linha** x **coluna** x **disparidade**). Formalmente, sendo fornecidas duas imagens no espaço ***LC***, consideramos a disparidade ***d*** como sendo a terceira dimensão, restrita num espaço de dimensão ***D***, ou seja, $d=0, \ldots, d_{max}$. Note que essa representação no espaço ***LCD*** é diferente de métodos volumétricos tradicionais (isto é, [22, 23, 24]), que também trabalham no espaço 3D, porém usando um sistema de coordenadas de mundo ***XYZ***, onde ***Z*** é a terceira dimensão.

Neste trabalho, consideramos um modelo ideal, onde a câmera da esquerda é colocada na origem do sistema e a câmera da direita só é transladada no eixo horizontal. Assume-se em princípio, sem perda de generalidade, que as imagens sejam retangulares. Cada elemento *(l,c,d)* do espaço ***LCD*** projeta no *pixel (l,c)* na imagem da esquerda e no *pixel (l,c+d)* na imagem da direita, conforme ilustrado na Figura 3.1.

Para se obter um mapa de disparidade suave e detalhado (isto é, detectando também as áreas de oclusão), usa-se uma função de atualização da disparidade, definida no espaço ***LCD***, para refinar iterativamente os valores de disparidade ***d***. Seja $L_n(l,c,d)$ essa função de atualização da disparidade, e o seu valor atribuído ao elemento *(l,c,d)* na n-ésima iteração. Valores iniciais, para $L_0(l,c,d)$, são calculados a partir das imagens estéreo originais, usando a fórmula abaixo:

$$L_0(l,c,d) = \partial\left(I_{esquerda}(l,c), I_{direita}(l,c+d)\right) \qquad \textbf{(3.1)}$$

Na Equação 3.1, ∂ pode ser uma função qualquer de medida de similaridade, tal como correlação normalizada. Idealmente, esta função de similaridade dos *pixels* das imagens deve produzir valores ótimos (altos) para *pixels* correspondentes determinados corretamente nas imagens. Entretanto, note que o que se espera para o oposto pode não ocorrer, isto é, muitos pontos correspondentes determinados erroneamente podem também ter valores altos iniciais, dados pela função de similaridade acima. Este problema é de certa forma dependente de parâmetros como o suporte local desejado (dimensões da janela de similaridade e dimensões da janela de busca por um correspondente), e de outros fatores como erros sistemáticos no processo de aquisição.

No presente trabalho, usa-se o somatório das diferenças absolutas como função de similaridade ∂, para determinar $L_0(l,c,d)$. Considerando-se determinar o grau de similaridade para um dado voxel (l,c,d), para uma janela de dimensões 2M+1 x 2N+1, esta função pode ser dada simplesmente por:

$$\partial(l,c,d)=L_0(l,c,d)=\sum_{m=-M}^{M}\sum_{n=-N}^{N}\left|I_e(l+m,c+n)-I_r(l+m,c+n+d)\right| \quad (3.2)$$

Usando esta função (3.2), o algoritmo para determinar os valores iniciais $L_0(l,c,d)$ em todos os voxels (l,c,d) do espaço ***LCD*** pode ser melhor entendido da seguinte forma:

- Centra-se uma janela de dimensões 2M+1 x 2N+1 em cada *pixel (l,c)* da imagem de referência, seja esta por exemplo a imagem esquerda;

- Para todos os *pixels* (l,c+d) da imagem direita, onde d são todos os valores de disparidade admitidos, utiliza-se o operador ∂ dado acima, que calcula o somatório dos módulos das diferenças entre todos os *pixels* que estejam dentro das janelas consideradas nas imagens direita e esquerda (a Figura 3.2 ilustra o caso de janelas 3x3).

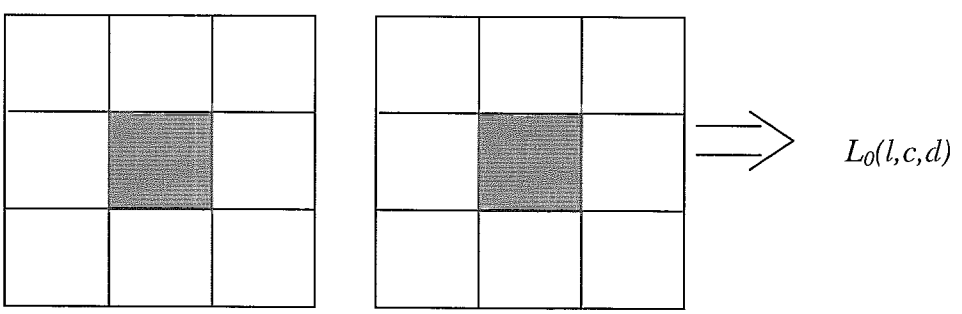


Figura 3.2 – Representação da função $\mathbf{L_0}$

Faz-se ainda uma normalização sobre a própria função $L_0(l,c,d)$ para que os valores de $L_0(l,c,d)$ estejam entre 0 e 1:

$$L_0(l,c,d) = \frac{I_{max}}{L_0(l,c,d) + I_{max}} \quad (3.3)$$

onde I_{max} corresponde ao valor máximo de intensidade que um *pixel* pode apresentar, ou seja, 255. Aqui, na prática, faz-se esta normalização dos valores de L_0 descritos pela Equação 3.3 no intervalo [0.1, 1.0], isto é, os valores calculados a partir da função $L_0(l,c,d)$, que estejam mais próximos de 1.0 implicam que o voxel atual já seja um possível candidato a ser escolhido como resultado final do processo de determinação da correspondência. E, por outro lado, voxels cujos valores de L_0 estejam próximos de 0.1 já são praticamente descartados nas iterações seguintes.

Em uma primeira análise, vemos que a escolha da função L_0 deve ser feita adequadamente. Se esta escolha for bem feita, de tal forma que pelo menos 50% dos valores de similaridade já estejam corretos, o resultado parcial já será bem parecido com o resultado final. Isto poderá não acontecer caso a função $L_0(l,c,d)$ não seja bem escolhida, e, neste caso, conforme será visto posteriormente, a função a $L_0(l,c,d)$ calculada através da similaridade da intensidade poderá restringir os valores finais.

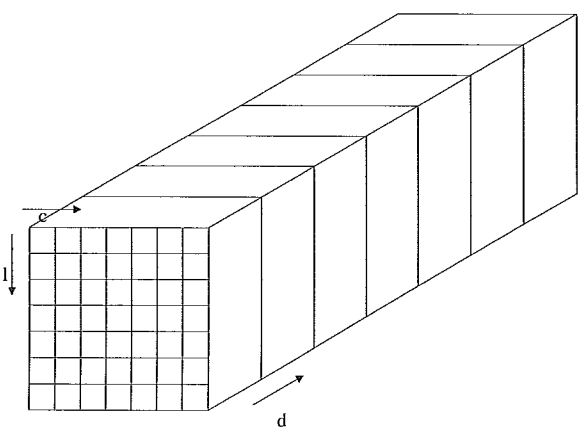


Figura 3.3 – Forma de armazenamento dos L_0

A Figura 3.3 mostra a forma de armazenamento da função L_0. Cada fatia (ou plano de corte) paralela ao plano formado pelas linhas e colunas contém o resultado da função de similaridade $L_0(l,c,d)$ naquele plano. Note que o valor da disparidade é mantido constante para cada elemento da mesma fatia, ou seja, a primeira fatia corresponde ao plano $d = 0$, a segunda fatia corresponde ao plano $d = 1$, e assim sucessivamente até que a última fatia corresponde ao plano $d = D_{max} - 1$. Idealmente, para um determinado *pixel (l,c)*, os valores da função $L_0(l,c,d)$ só estariam definidos num único plano *d=constante*, em que a similaridade é máxima, o que não ocorre na prática devido a erros. Neste trabalho, busca-se exatamente descartar essas falsas correspondências, usando a hipótese de unicidade.

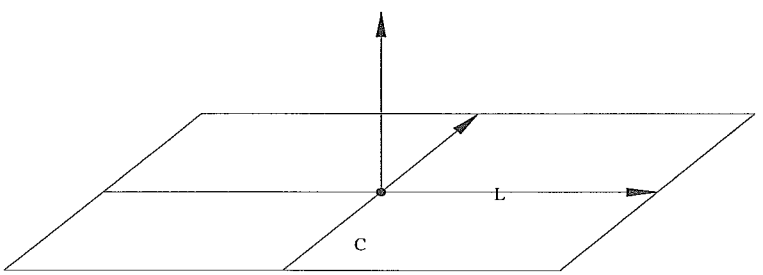


Figura 3.4 – Representação dos centros das imagens

3.2 O sistema XYZ e o sistema LCD

Quando se trabalha no sistema ***XYZ***, normalmente coloca-se a origem do sistema de coordenadas no centro de projeção da imagem esquerda. No sistema ***LCD***, aqui usado, o centro de projeção localiza-se na reta vertical que passa pelo centro das imagens (ver Figura 3.4). Considerando-se que este centro possui coordenadas (X/2, Y/2), esses valores podem ser usados para se estabelecer uma relação entre **x, c, y** e **l**. Os voxels do paralelepípedo (***LCD***) que possuem coordenadas (l=L/2,c=C/2,d) projetam no centro da imagem da esquerda, conforme pode ser visualizado na Figura 3.5.

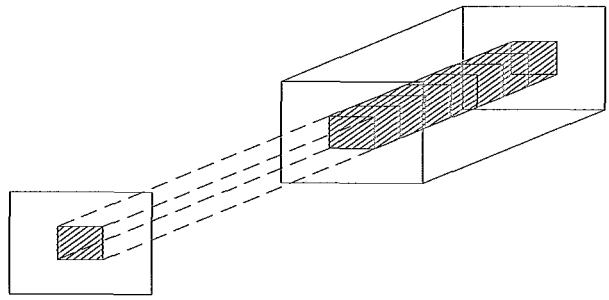


Figura 3.5 – Projeção do paralelepípedo(L/2,C/2,d) na imagem

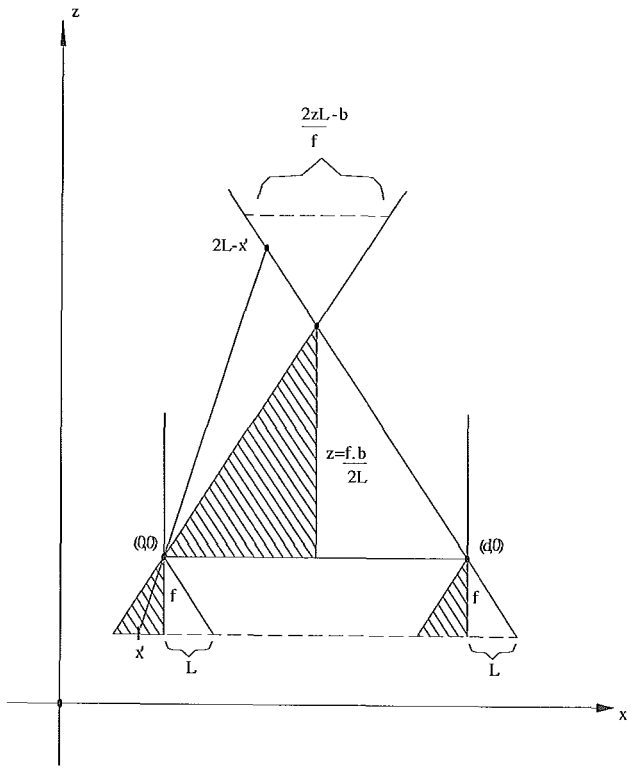


Figura 3.6 – Modelo ideal usado para converter do sistema ***XYZ*** para o ***LCD***.

3.3 Representando Equações de Retas no Sistema LCD

Para apresentar o modelo matemático usado neste trabalho, é necessário primeiramente introduzir as representações para equações de retas no sistema **LCD**. Para tal, usa-se o modelo estéreo representado na Figura 3.6, que foi derivado do modelo ideal para estereoscopia, visto no capítulo anterior, com algumas redefinições descritas a seguir:

f: distância focal ou distância do ponto nodal (centro) de cada conjunto de lentes;

b: Comprimento do segmento que une os pontos nodais (ou centros dos conjuntos de lentes) das duas câmeras (este segmento é também denominado de linha de base);

z: profundidade do ponto em relação ao sistema global definido;

L: deslocamento em x do ponto em relação ao eixo das abscissas.

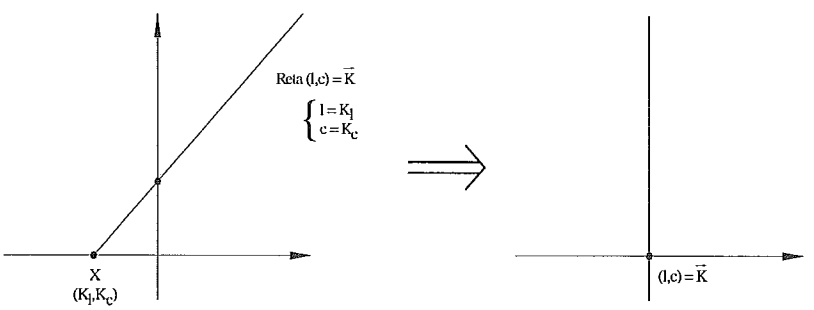


Figura 3.7 a) **Figura 3.7 b)**

Segundo o modelo apresentado acima, torna-se possível transportar a equação de uma reta do sistema ***LCD*** para o sistema ***XYZ***. Para tal, seja r uma reta oblíqua descrita no sistema ***XYZ*** e seja *k* a sua correspondente no sistema ***LCD*** (ver Figura 3.7). Conforme ilustrado na Figura 3.8, sabemos que uma reta oblíqua, ao ser representada em um computador em forma de imagem ("raster"), pode interceptar mais de um *pixel* em uma mesma coluna. A reta *(l, c) = k*, expressa no sistema *XYZ*, possui a expressão paramétrica dada por:

$$\left(\frac{z}{f}\left(\frac{X}{2}-K_c\right)+\frac{L}{2}, K_l-\frac{Y}{2}+\frac{C}{2}, z\right) \qquad \textbf{(3.4)}$$

Considerando a Equação A.2.2 (para o modelo ideal de estereoscopia descrito no Apêndice 2), a reta que possui o ponto (k_l, k_c, k_d) tem a seguinte expressão no sistema ***XYZ***:

$$\left(\frac{z}{f}\left(\frac{X}{2}-(K_c+K_d)+b+\frac{L}{2}, K_l-\frac{Y}{2}+\frac{C}{2}, z\right)\right) \qquad \textbf{(3.5)}$$

No sistema ***LCD***, a reta acima é descrita por:

$$(c+d)=(k_c+k_d) \qquad \textbf{(3.6)}$$

Pode-se chegar a expressão acima (3.6), primeiro substituindo-se z por $f\left(\frac{b}{d}\right)$ na expressão 3.5 acima, onde D descreve a linha de base:

$$x=\left(\frac{b}{d}\left(\frac{X}{2}-(K_c+K_d)+b+\frac{L}{2}\right)\right) \qquad \textbf{(3.7)}$$

Como c e x se relacionam pela expressão:

$$x=\frac{b}{d}\left(\frac{X}{2}-c\right)+\frac{L}{2}, \qquad \textbf{(3.8)}$$

igualando-se, portanto, os lados direitos da Equação (3.7) e da Equação (3.8), obtemos o resultado desejado (ou seja, a Equação 3.6).

Figura 3.8 "Rasterização" de uma reta

3.4 O SUPORTE LOCAL

A hipótese de continuidade implica que elementos vizinhos possuam valores de similaridade consistentes. Nesse trabalho, é utilizada uma média dos valores vizinhos, em um espaço 3D, para aumentar a consistência. Essa média de valores é calculada dentro de uma região volumétrica que denominamos de suporte local para um único elemento.

O suporte local determina quais elementos da vizinhança que devem contribuir para a média. Na verdade, o suporte local deve incluir somente os elementos da vizinhança que possam vir a contribuir com um valor de similaridade correto, isto é, se o elemento atual estabelecer uma correspondência correta para dois *pixels* nas imagens.

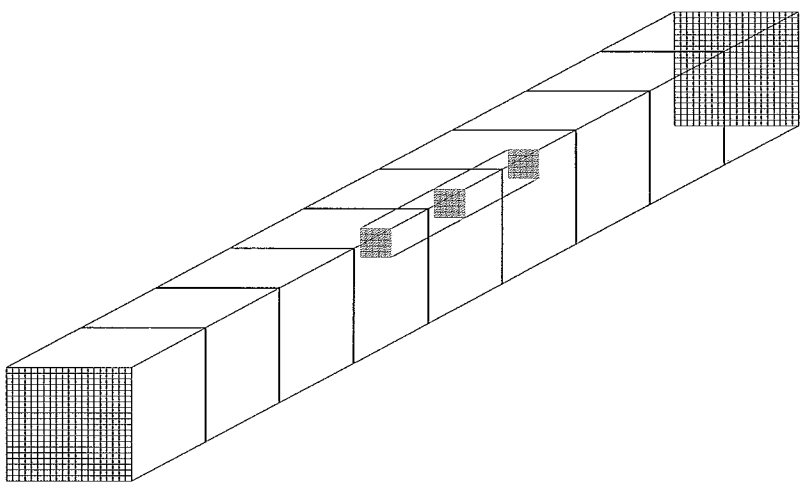


Figura 3.9 – O prisma mais interno representa um suporte local de 5x5x3

As dimensões do suporte local influenciam no resultado final do algoritmo, pois ele é uma média entre os vizinhos de um determinado voxel. Note que não é adequado utilizar um suporte local de grandes dimensões, devido a possível inclusão de voxels que influenciam erroneamente no somatório final para o cálculo da média. No algoritmo proposto, obtivemos os melhores resultados com um suporte local de dimensões 5x5x3, isto é, varia em cinco unidades no plano definido pelas linhas e colunas e varia em três unidades para a disparidade (eixo da profundidade).

Marr e Poggio [20, 21], por exemplo, usaram elementos em uma área bidimensional onde o valor da disparidade é constante, isto é seu suporte local é uma área 2D (onde d = constante) no espaço ***LCD***. O suporte local, usado por eles [20, 21], corresponde a uma fatia do prisma (Figura 3.10) que é um plano paralelo ao plano da imagem de referência. Contudo, devido a certas restrições da superfície a ser descrita, entre elas a declividade representada nas imagens pela diferença de disparidade entre *pixels* vizinhos, em nosso trabalho usamos uma área 3D como suporte local. Muitas hipóteses de suporte local 3D foram propostas em [25], [26], [27], [16]. Kanade e Okutomi [16] apresentam uma análise detalhada dos diversos relacionamentos incluindo as diferenças principais entre eles. No presente trabalho, utiliza-se um suporte local com dimensões fixas para todos os voxels, isto é, mesma largura, altura e profundidade.

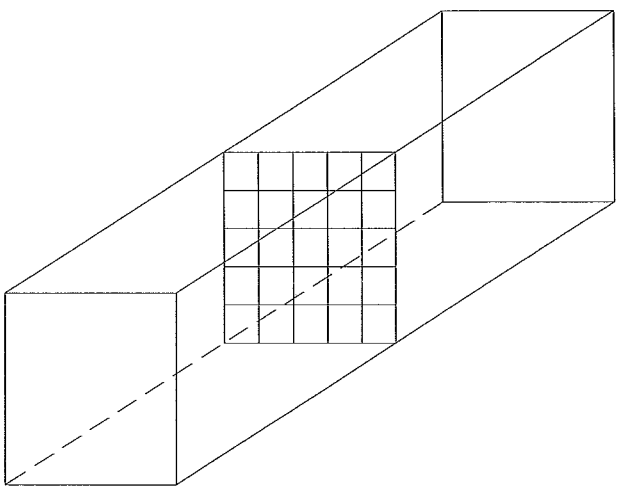


Figura 3.10 – A fatia quadriculada corresponde a um plano *d = constante*.

Seja $S_n(l,c,d)$ o valor do suporte local representado em cada voxel (l,c,d) do espaço ***LCD***, isto é, o somatório de todos os valores de similaridade dentro de uma região **Q** de suporte local 3D sobre os valores de $L_n(l,c,d)$ que descrevem os valores de similaridades correntes ou atuais.

$$S_n(l,c,d) = \sum_{(l',c'd')\in Q} L_n(l+l', c+c', d+d') \qquad (3.9)$$

Convém ressaltar que, no algoritmo proposto, a utilização de um suporte local 3D está baseada na hipótese da continuidade da disparidade, descrita anteriormente.

3.5 Fase de Competição entre Voxels

A hipótese da unicidade implica que só pode existir um único correspondente dentro de um conjunto de elementos que projetam no mesmo *pixel* em uma imagem. Conforme ilustrado na Figura 3.2 pelos quadrados preenchidos (mais escuros), seja P(l,c,d) a zona de inibição, isto é, o conjunto de elementos que se sobrepõe ao elemento (l,c,d) quando projetados em uma imagem. Sabemos que cada voxel dentro da área P(l, c, d) projeta no *pixel* (l,c) na imagem da esquerda ou no *pixel* (l,c+d) na imagem da direita. Seja $R_n(l,c,d)$ uma média decorrente do resultado da inibição nos elementos de $S_n(l,c,d)$ em função dos elementos em P(l,c,d), que pode ser dada por:

$$R_n(l,c,d) = \left(\frac{S_n(l,c,d)}{\sum_{(l',c',d') \in P} S_n(l',c',d')} \right)^{\alpha} \qquad (3.10)$$

Arguímos que exista uma competição entre os voxels de uma mesma coluna c, uma vez que a cada iteração, os voxels com valores de similaridade mais altos na área de inibição P(l,c,d) tendem a competir entre eles. Ainda, devemos considerar que o somatório dos valores de similaridade dentro do suporte local, representado na Equação 3.9, pode acarretar numa sobrecarga da suavidade, e assim sendo, pode ocorrer uma perda dos detalhes da cena.

Na Equação 3.10, o expoente α controla a convergência da função R_n. Para garantir que um único voxel dentro da área de inibição P(l,c,d) tenha uma convergência direcionada para 1, α deve ser muito maior que 1.

Ainda, de acordo com a Equação 3.10, o denominador da função $R_n(l,c,d)$ é calculado sobre a zona de inibição P(l,c,d), descrita na Figura 3.1. É válido observar que, na zona de inibição, o voxel sobre o qual estão sendo calculadas todas as operações possui uma influência duplicada, devido estar contido tanto na retas oblíquas (retas que são projetadas na imagem da direita) quanto nas retas verticais (retas que são projetadas na imagem da esquerda).

Assim, o somatório dos valores de similaridade dentro da Equação 3.9 pode resultar em perda de detalhes devido a grande quantidade de suavização da imagem, o

que pode também acarretar em uma má detecção das bordas dos objetos juntamente com um efeito da justaposição entre objetos distintos, pois pode ocorrer a junção entre os mesmos.

No algoritmo aqui apresentado, propomos restringir paulatinamente os valores da correspondência, de acordo com a similaridade entre o *pixel (l,c)* da imagem esquerda e o *pixel (l,c+d)* da imagem direita. Deste modo, só permitimos a elementos que projetam em *pixels* com intensidade similar possuírem altos valores de similaridade, embora possam haver *pixels* com intensidades não similares e com altos valores de correspondência. Os valores iniciais $L_0(l,c,d)$, calculados anteriormente, são usados para restringir os valores atuais da função $L_n(l,c,d)$. Seja $T_n(l,c,d)$ a função que denota o valor $R_n(l,c,d)$ restrita pelos valores de $L_0(l,c,d)$, isto é :

$$T_n(l,c,d) = L_0(l,c,d) * R_n(l,c,d) \tag{3.11}$$

A função de atualização usada no algoritmo proposto é construída a partir das equações 3.9, 3.10 e 3.11 nessa ordem, como sendo:

$$L_{n+1}(l,c,d) = L_0(l,c,d) * \left(\frac{S_n(l,c,d)}{\sum_{(l'',c'',d'') \in P} S_n(l'',c'',d'')} \right)^{\alpha} \tag{3.12}$$

Enquanto o método usado neste trabalho usa as mesmas hipóteses de Marr e Poggio, a função de atualização usada aqui difere substancialmente da função usada por eles. Usando a notação corrente, a função de atualização de Marr e Poggio é dada por:

$$L_{n+1}(l,c,d) = \sigma\left(S_n(l,c,d) - \varepsilon \sum_{(l',c',d') \in P(l,c,d)} L_n(l',c',d') + L_0(l,c,d) \right), \tag{3.13}$$

onde σ é a chamada função sigmóide em [08], ε é uma constante de inibição sobre os valores de $L_n(l,c,d)$, $S_n(l,c,d)$ é o nosso suporte local e $L_0(l,c,d)$ contém todos os valores iniciais, incluindo o que chamamos de falsas correspondências (também denominados de alvos falsos ou "false targets").

Possivelmente, Marr e Poggio [20, 21] usaram valores de similaridades discretos e um suporte local 2D para $Q(l,c,d)$, devido às restrições de tecnologia (memória e processamento) não permitirem melhores condições na época. Convém ressaltar que os resultados por eles obtidos, através de imagens sintéticas de pontos aleatórios ("random dot stereograms"), foram muitos bons. Porém, devido às imagens estéreo não sintéticas (reais) poderem apresentar variações no nível da intensidade e também variações na disparidade, é necessário a criação de um suporte local 3D para que se obtenha valores contínuos . A Equação 3.12 tem duas vantagens principais em relação à Equação 3.13 no uso em imagens reais.

Primeiro, os valores de similaridade na Equação 3.12 são restritos pelos valores iniciais devido à manutenção dos detalhes. Na Equação 3.13 os valores iniciais são adicionados aos valores correntes, preponderando dessa maneira os valores que forem inicialmente altos. Com isso pode ocorrer a sobrecarga da suavidade e a perda de detalhes.

Segundo, a função de inibição na Equação 3.12 é mais simples que a função sigmóide. Para os experimentos de Marr e Poggio, eles preferiram usar uma função limiar em vez de uma função sigmóide devido a restrições de processamento.

O algoritmo aqui proposto tem ainda outra vantagem sobre o algoritmo de Marr e Poggio, e também sobre o de Kanade e Zitnick, que é o uso da Programação Dinâmica para a aceleração do processo de convergência da função $L_{n+1}(l,c,d)$ e também para a suavização dos mapas de disparidade do processo final. A técnica da Programação Dinâmica utilizada no nosso algoritmo será descrita posteriormente neste mesmo capítulo.

3.5.1 Detecção de Oclusão

A oclusão é um dos aspectos mais críticos, sendo difícil de ser detectada e tratada em algoritmos estéreo. Em qualquer imagem razoavelmente complexa, existem *pixels* que podem estar oclusos em uma das imagens, consequentemente não são detectados. Isto significa a não obtenção de um correspondente (ou casamento) para o mesmo, possivelmente com valores baixos para a função de similaridade.

Muitos algoritmos estéreo não consideram o caso da detecção explícita de oclusão. Devido a isso, esses algoritmos produzem erros grosseiros em áreas de oclusão ou encontram valores falsos de similaridade para o que chamamos de foreground (objeto de interesse) e background (fundo). Recentemente, foram publicados alguns algoritmos em estéreo (Belhumeur e Mumford [28], Geiger, Ladenford et al. [29], e, Intile e Bobick [30]) que consideram a detecção de oclusão e identificação das descontinuidades da disparidade.

Em nosso algoritmo, identificamos oclusões, a partir dos valores de similaridade finais e usando a restrição de unicidade. Desde que não ocorra falsas correspondências em áreas de oclusão, ou seja, todos os valores correspondentes a *pixels* oclusos devem ser baixos. Em nosso algoritmo consideramos um *pixel* como ocluso se o mesmo contiver um valor de similaridade menor que 0.11, mas em algumas imagens esse limite pode ser um pouco maior. Convém ressaltar que, os próprios valores da função L_0 já determinam para um dado *pixel* se ele tem tendência a ser ocluso.

Considere um *pixel* p na imagem da esquerda, cujo correspondente na imagem da direita não seja visível. De acordo com a Figura 3.10, para um elemento v ao longo da linha de visão de p, há dois casos que podem ocorrer para a sua projeção q na imagem da direita. No primeiro caso, mostrado na Figura 3.10(a) q, que está na imagem da direita, tem um ponto correspondente que é visível na imagem da esquerda. Então, existe um elemento v´que é o correspondente correto entre q e p´. Havendo elementos v e v´que projetem em um mesmo *pixel* q, seus valores de similaridade inibirão qualquer ponto que estiver na mesma linha de visão de q devido a hipótese de unicidade. Geralmente, o elemento corrente v´terá um alto valor de similaridade, isto é, um valor alto para o seu respectivo $L_n(l,c,d)$ e isto faz com que o valor de similaridade para o elemento v diminua. Assim sendo, o elemento v terá um valor baixo para $L_n(l,c,d)$. O segundo caso, representado na Figura 3.10(b), é mais difícil de ser detectado. Ele ocorre quando o ponto q (que é o correspondente correto) estiver ocluso na imagem da esquerda. Deste modo, nem p nem q terão um valor correto de similaridade. O valor de similaridade entre p e q não receberá a contribuição dos valores na zona de inibição $P(l,c,d)$ e correspondências falsas poderiam ter valores altos. Nos dois casos descritos acima, hipóteses adicionais devem ser feitas para podermos encontrar as áreas de

oclusão. As restrições devem ser elaboradas de tal maneira que se possa realmente descrever um *pixel* com sendo ocluso. Em geral, as áreas de oclusão são representadas dentro da função de atualização $L_{n+1}(l,c,d)$, como regiões que possuem valores de intensidade não similares no intervalo da disparidade considerado.

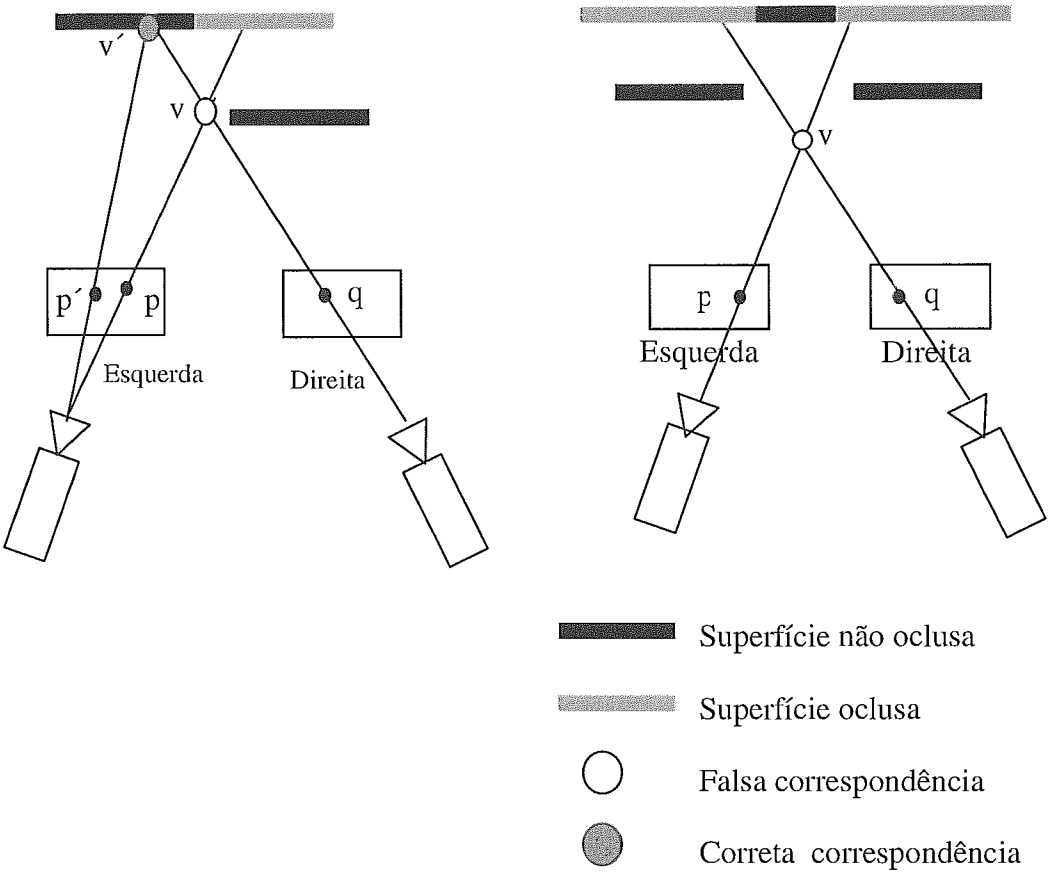


Figura 3.10 – a) Se q não é ocluso existe um correto match com p´ o qual inibirá o falso match com p;

Figura 3.10 - b) Se q é ocluso então, é possível que haja um falso match com p.

No algoritmo proposto, após os valores de similaridade terem convergido, pode-se determinar se um *pixel* é ocluso, ao se encontrar um elemento com o valor de similaridade superior (Figura 3.11) ao longo de sua linha de visão. A técnica da Programação Dinâmica usada aqui e que será descrita na próxima seção, explora de

alguma forma a detecção de *pixels* oclusos, apesar de não ser esta a finalidade do seu uso, mas sim acelerar o processamento.

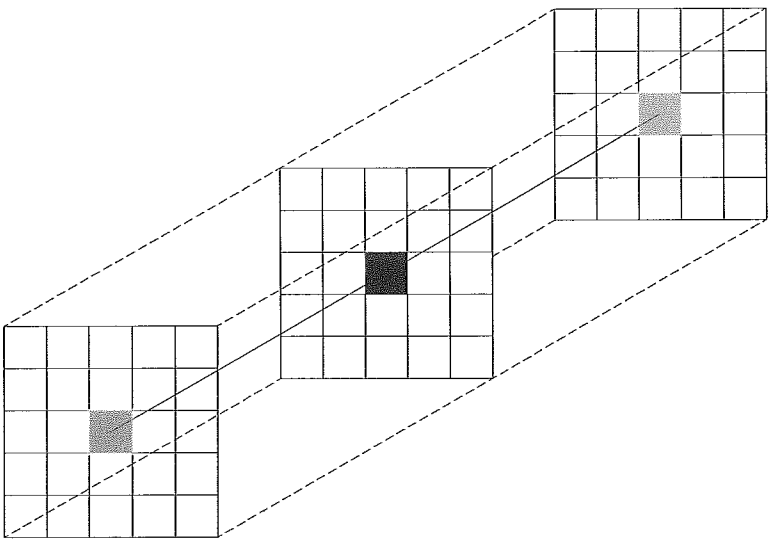


Figura 3.11 – Representação gráfica de um pixel ocluso de acordo com o algoritmo.

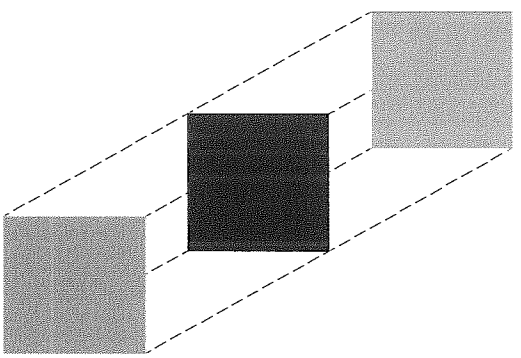


Figura 3.12– Representação visual do pixel ocluso.

Na Figura 3.11, os voxels são pintados de acordo com o critério descrito a seguir. Levando em consideração que o nível de cinza varia do preto (intensidade 0) ao branco (intensidade 255), quanto maior o nível de cinza, maior será o valor obtido pela função $L_{n+1}(l,c,d)$. Os voxels que contêm a intensidade mais próxima do preto (intensidade 0) são aqueles que possuem os menores valores de $L_{n+1}(l,c,d)$. Os voxels que contêm altos valores de $L_{n+1}(l,c,d)$ são representados por alta intensidade (mais próximos do branco). Então, de acordo com a Figura 3.12, o plano que está pintado de

preto contém um voxel que deve ter um valor de similaridade baixo para a função $L_{n+1}(l,c,d)$.

3.6 Programação Dinâmica

A técnica de programação dinâmica já foi usada anteriormente em algoritmos estéreo. Tzovaras e Grammalidis [31], em seu algoritmo estéreo, consideram a dinâmica proporcionada por seqüências de imagens que contém juntos movimento e disparidade. Uma estimativa da disparidade para cada *pixel* é calculada utilizando um algoritmo de programação dinâmica baseado na hierarquia de cada *pixel*. Em Ohta e Kanade [18], a programação dinâmica é utilizada em duas varreduras, interna e externa (intra- e inter-scanline) de arestas, de forma progressiva para a execução do matching .

Neste trabalho, desenvolvemos uma técnica de PD, também descrita em Matias e Oliveira [32], que calcula uma função de custo mínimo para cada plano de linha (l=constante) contido nos valores de similaridade. Nesse caso, a programação dinâmica é usada, explorando-se as restrições de suavidade e continuidade dos mapas de disparidade, descartando-se desta maneira a criação de buracos na imagem resultante. A Figura 3.13 fornece uma visualização em 3D de uma fatia (2D) sobre a qual é aplicada a função de custo mínimo.

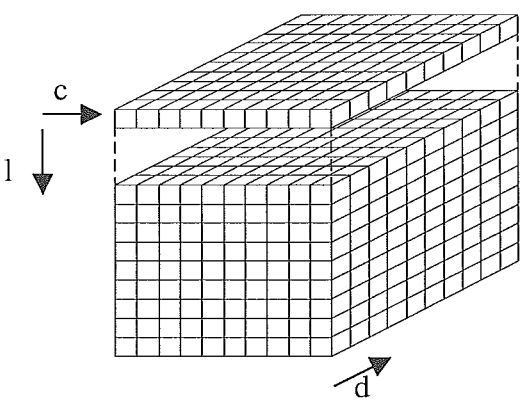


Figura 3.13 – Mostra a divisão em fatias (*l = constante*), no sistema lcd

Neste caso, quebramos o problema no espaço 3D em dois subproblemas no espaço bidimensional. Em cada plano de linha constante, tentamos manter a condição de

diferenciabilidade da disparidade de um *pixel* para seu vizinho. Isto é, consideramos que a diferença de disparidade entre um elemento $L_{n+1}(l,c,d)$ e $L_{n+1}(l,c+1,d)$ varia de forma suave, ou seja:

$$|L_{n+1}(l,c+1,d) - L_{n+1}(l,c,d)| < \varepsilon$$

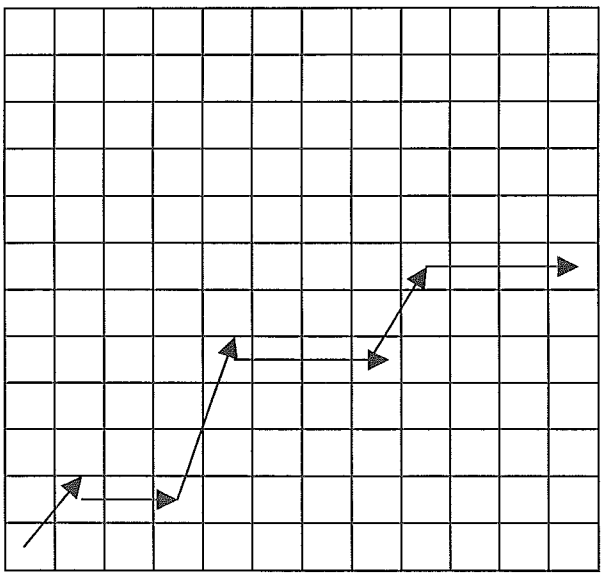


Figura 3.14 – Representação em corte na linha

A Figura 3.14 exibe o caminho ótimo da disparidade em uma fatia 2D (plano l = *constante*), na representação da função $L_{n+1}(l,c,d)$. Para encontrar este caminho ótimo através da função custo mínimo, utilizamos a metodologia descrita a seguir:

- determina-se os voxels que possuem os maiores valores de $L_{n+1}(l,c,d)$ dentro de uma certa coluna;
- calcula-se a função de custo mínimo entre as colunas $c = 1$ até $c =$ *máximo*;
- em cada coluna, o voxel (l,c,d) escolhido determinará o *pixel* (l,c) da imagem resultante.

3.7 O Algoritmo e sua complexidade

Conforme foi citado no início deste capítulo, para gerar os mapas de disparidade, primeiro calculamos os valores iniciais, que denominamos pela função $L_0(l, c, d)$, e logo após calculamos a função $L_{n+1}(l, c, d)$ que dependem dos valores iniciais. O algoritmo para a geração dos mapas de disparidades é o seguinte:

1. Para cada voxel (l, c, d) construir os valores de similaridade iniciais dados pela função $L_0(l, c, d)$;
2. Iterativamente, atualizar os valores da função $L_{n+1}(l, c, d)$ usando a expressão (3.10), até que todos os vóxels satisfaçam a regra de terminação. Vóxels com valores extremamente baixos não são mais atualizados;
3. Utilizar a Programação Dinâmica sobre os valores de $L_{n+1}(l, c, d)$, incluindo somente vóxels com os mais altos valores, que determinam um voxel V^* (l,c,d) correspondente ao *pixel* (l,c);
4. Se o valor de match para o voxel V* é muito baixo, classificar o *pixel* correspondente como sendo ocluso, caso contrário a sua saída gera a disparidade *d*.

O tempo para a execução dos passos de 1 a 4 (do algoritmo descrito acima) é da ordem de N^2*D*I, onde N^2 é o tamanho da imagem, D é o intervalo máximo de disparidade admitido, e I é o número de iterações. O espaço de memória requerido é da ordem de N^2*D.

Em termos práticos, este algoritmo é cerca de duas a cinco vezes mais rápido do que o algoritmo de Kanade. Basicamente, com três iterações atingimos a precisão ideal, conforme será visto no próximo capítulo.

Capítulo 4

Experimentos e resultados

4.1 Introdução

Como visto nos capítulos anteriores, a implementação de nosso trabalho está baseada no algoritmo do Zitinick e Kanade [01, 33, 34] no que se refere ao seu esquema básico, ou seja, à metodologia seguida de refinar iterativamente a partir de um mapa de disparidade grosseiro inicial, porém diferindo no que se refere às técnicas empregadas para o processamento, a partir daí, para alcançar a suavização dos mapas de disparidade de forma mais rápida. Vale ressaltar que o objetivo deles [01, 33, 34] é a detecção da oclusão e o nosso objetivo é a suavização dos mapas de disparidade utilizando a programação dinâmica, e também a aceleração do processo de convergência, o que acarreta numa possível perda de detalhes da imagem. Convém ainda ressaltar que a nossa proposta não deixa totalmente de lado a detecção de oclusões, sendo esta uma conseqüência do processo final, com um certo grau de precisão, claro.

Em nossos experimentos foi utilizado como hardware um micro PC AT da IBM com processador Pentium II de 300 Mhz da Intel com 64 Megabytes de memória RAM. O nosso aplicativo foi construído utilizando a API gráfica "OpenGL" juntamente com a linguagem de programação C e com o "GLUT" como sendo o gerenciador de menus. Convém ressaltar que a aplicação é independente da plataforma computacional, pois foram feitos testes nos sistemas Operacionais Windows98, Linux, Unix da Solaris e AIX da IBM.

Em uma segunda etapa, foi também construída uma interface para nosso aplicativo usando a biblioteca gráfica "MOTIF", voltada para o ambiente Linux.

A parte prática desta tese incluiu ainda a o desenvolvimento e implementação de um programa para a reconstrução tridimensional e visualização a partir dos dados de disparidade, denominado de "DTM" (veremos alguns resultados neste capítulo fornecidos por ele). Este programa foi elaborado por nós em parceria com o Programador da Petrobrás Francisco Fábio de Ponte e cuja interface foi construída no ambiente "C++ Builder" voltado para plataformas que usam o sistema operacional Windows.

Usamos vários pares de imagens estéreo para testar o algoritmo e foram selecionados dentre estes alguns pares para mostrar os resultados que serão apresentados a seguir. Estas imagens selecionadas foram fornecidas por Zitnick e Kanade [01, 33, 34] via e-mail e por Gonçalves e Oliveira [02]. O formato de imagens utilizado como entrada e saída de nosso programa é o pgm (portable graymap). Só utilizamos imagens em grayscale (tons de cinza), uma vez que o objetivo era comparar com outros algoritmos existentes. Convém ainda ressaltar que a resolução da imagem pode ser qualquer uma, porém, quanto maior esta resolução, maior será o tempo para a geração dos mapas de disparidade.

Finalmente, é válido ressaltar ainda que, em relação aos ambientes Linux e Windows, obteve-se um pequeno ganho de rapidez no ambiente Linux, devido a este ter um melhor gerenciamento de memória.

4.2 Resultados experimentais

Para demonstrar a validade do algoritmo proposto, selecionamos três pares de imagens estéreo distintos, e cada um deles com resolução diferente. A função utilizada para calcular os valores iniciais $L_0(l,cd)$ foi o somatório das diferenças absolutas apresentado na Equação (3.1) com a normalização inicial efetuada pela Equação (3.2), de modo que seus valores estejam entre 0.1 e 1. O limite de oclusão foi constante para todas as imagens com valor 0.111, e, nos experimentos, o fator de convergência α variou entre 1.01 a 1.05. Na maioria das imagens o fator alfa não teve muita influência, pois a Programação Dinâmica acelerou o processo de convergência. A Programação Dinâmica

é usada em todos os passos de iteração, desde os valores iniciais até os valores de similaridade atuais.

A Figura 4.1 mostra um dos pares estéreo selecionado para visualização do resultado do algoritmo nesta seção. Ambas as imagens foram tomadas de uma cena contendo uma mina de carvão.

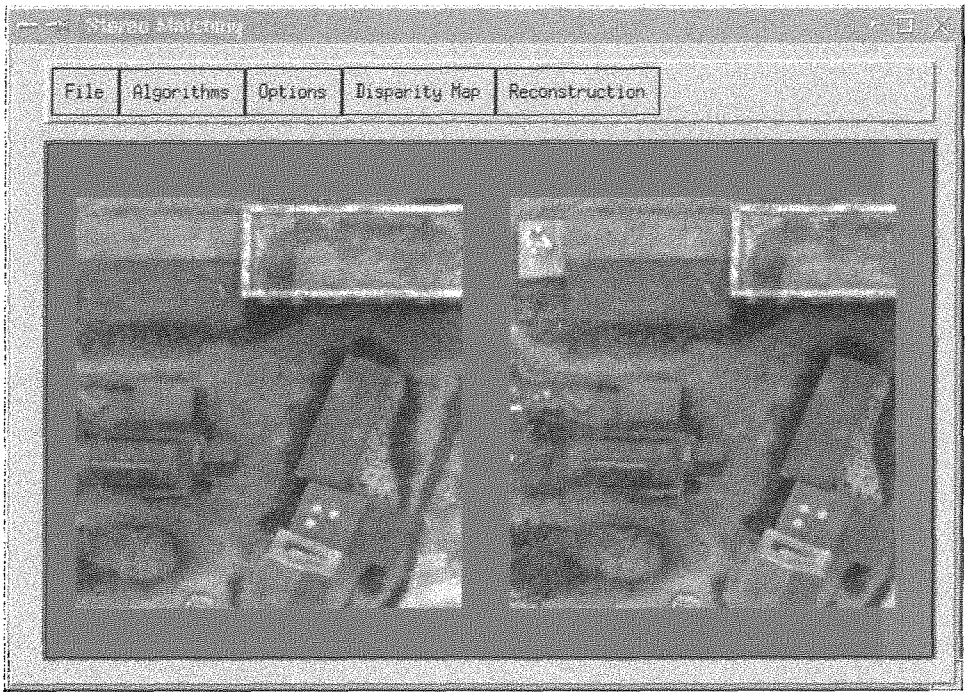


Figura 4.1 – Par estéreo de tamanho 240x256

Aplicamos o algoritmo proposto neste trabalho nas imagens apresentadas na Figura 4.1 e com poucas iterações, obtivemos um resultado bem parecido com o resultado obtido pelo Zitnick e Kanade [01, 33, 34], consumindo um tempo de processamento muito menor. O resultado deste primeiro experimento pode ser visualizado na Figura 4.2. A imagem da esquerda representa o mapa de disparidade inicial dado pela função L_o e o mapa da direita foi obtido após o processo iterativo com o uso de Programação Dinâmica. Para a determinação de L_o, foi utilizada uma janela de tamanho 3x3 no cálculo do somatório das diferenças absolutas (Equação 3.1). O suporte local utilizado foi de dimensões 5x5x3. O resultado representado pela imagem da direita para a obtenção do mapa de disparidade foi conseguido com apenas três iterações.

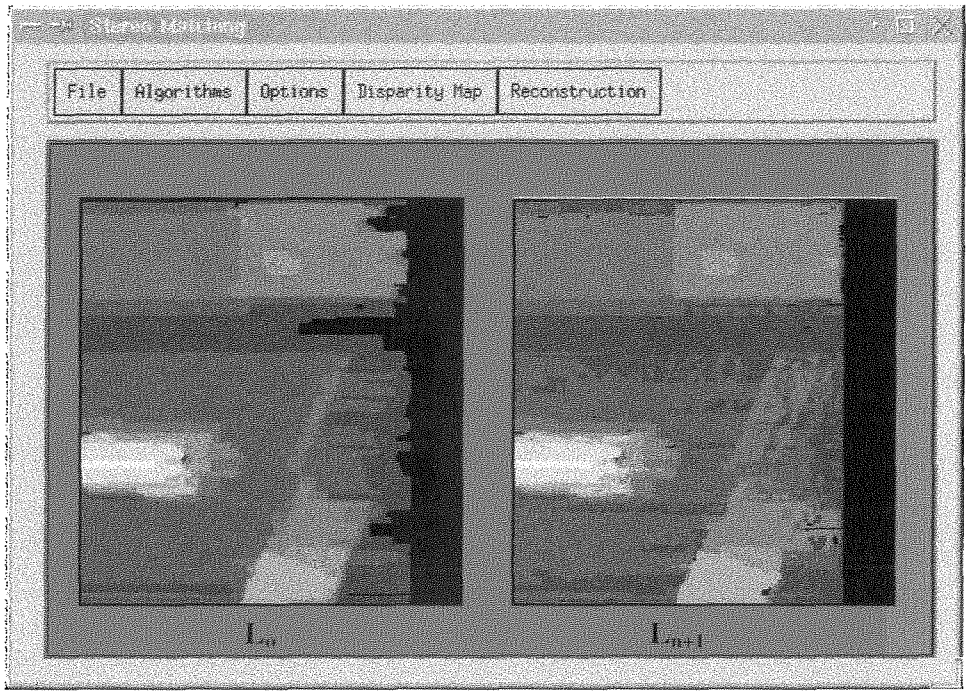


Figura 4.2 – À esquerda estão os L_o e à direita está o mapa de disparidade obtido.

O tempo para a determinação dos valores iniciais para L_o foi de 1.93 segundos e o resultado final, com o mapa de disparidade, foi obtido em 6.91 segundos. Para a convergência com o uso da Programação Dinâmica, foi usado como ponto de corte valores de similaridade menores que 0.75. É válido observar que o uso da Programação Dinâmica, além de acelerar o processo, apresenta um efeito suavizador, resultando num mapa de disparidade com continuidade notada de um ponto para outro.

Deve-se observar ainda que é possível acelerar o processo de convergência aumentando-se o ponto de corte para a Programação Dinâmica, mas que isso pode acarretar em uma perda de detalhes e também no aparecimento de falsas correspondências e/ou, conseqüentemente, falsas oclusões. A Figura 4.3 ilustra bem o efeito desse aumento. A imagem de disparidade da esquerda corresponde aos valores iniciais de L_o, e a imagem de disparidade da direita corresponde aos valores finais após aplicação da PD. Ressalta-se o aparecimento de regiões onde pode ser notada a existência de uma certa margem de erros, ao comparar este resultado com o mostrado a Figura 4.2.

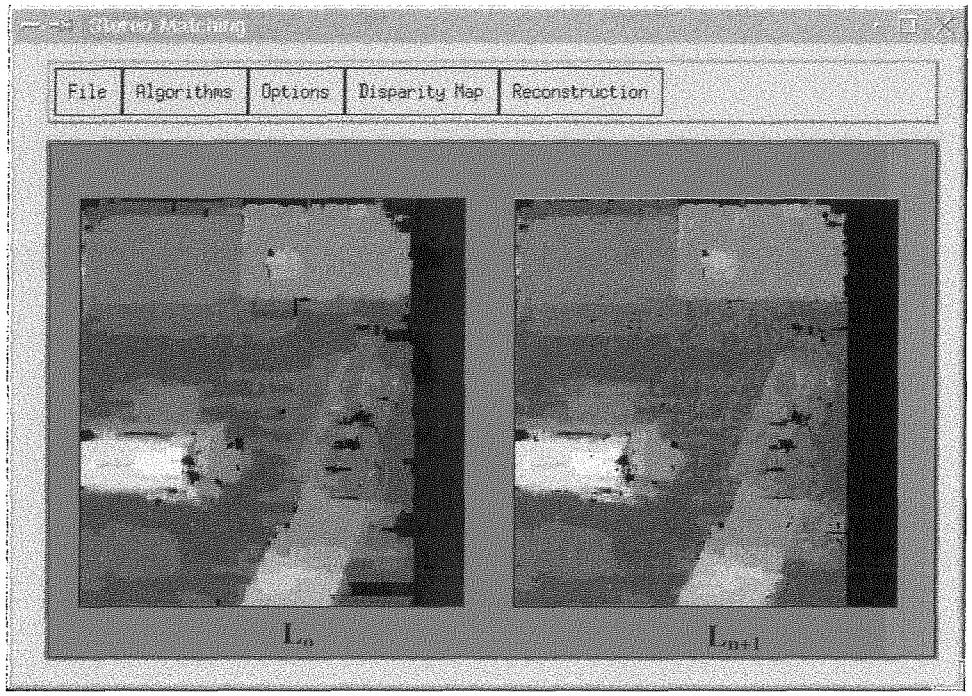


Figura 4.3 – Aparecimento de falsas correspondências e de falsas oclusões

Para alcançar o resultado mostrado a Figura 4.3, foi utilizada uma janela de tamanho 3x3 para o cálculo do somatório das diferenças absolutas (ou seja, para a determinação dos valores de L_o). As dimensões do suporte local utilizado foram 5x5x3, e com apenas três iterações conseguiu-se o resultado mostrado pela imagem da direita para a obtenção do mapa de disparidade.

O tempo para a determinação dos valores iniciais para L_o foi de 1.93 segundos e o resultado final foi obtido em 5.76 segundos. Para a convergência com o uso da Programação Dinâmica, foram descartados valores de similaridade menores que 0.93. O tempo de processamento para a criação do mapa de disparidade mostrado na imagem da direita da Figura 4.3 é bem mais rápido (mais de um segundo) que o gasto para a criação do mapa de disparidade mostrado na imagem da direita da Figura 4.2, isto devido ao valor de corte ser maior o que acarreta uma aceleração no processo. Porém, pode-se notar na Figura 4.3, a presença de falsas correspondências e de falsas zonas de oclusão.

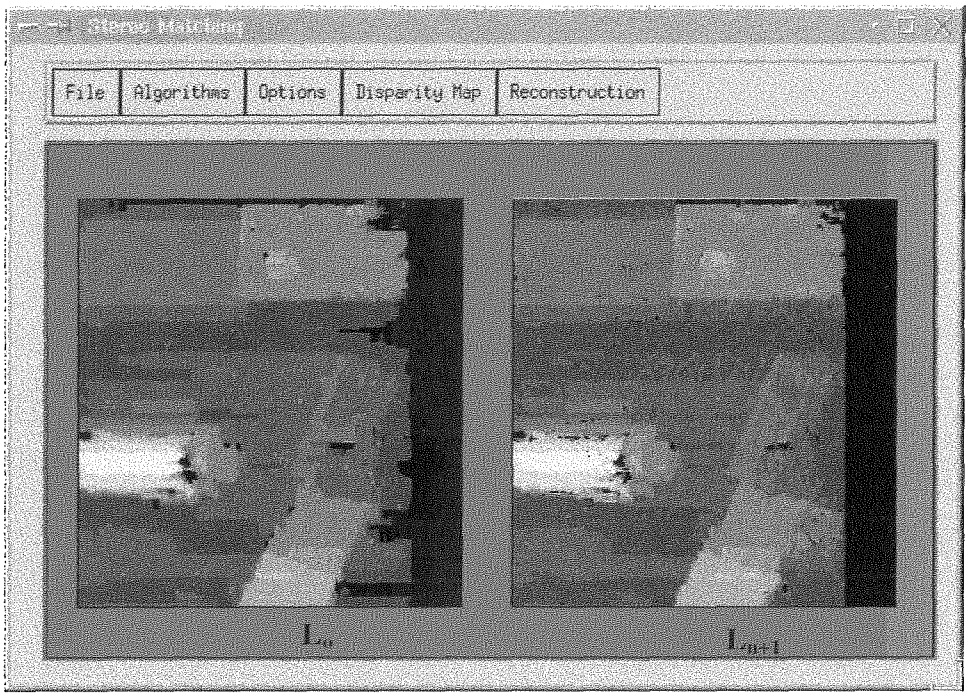


Figura 4.4 – Exibida com suporte local 3x3x3

Outra maneira de se melhorar o desempenho do processo de convergência com uso da PD é através da diminuição do suporte local. No resultado mostrado na Figura 4.4, foi utilizada uma janela de tamanho 3x3 para o cálculo do somatório das diferenças absolutas (determinação dos valores de L_o na imagem da esquerda), as dimensões do suporte local utilizado foram 3x3x3. Com apenas três iterações foi conseguido o resultado representado pela imagem da direita para a obtenção do mapa de disparidade.

O tempo para a determinação dos valores iniciais para L_o foi de 1.93 segundos e o resultado final foi obtido em 5.02 segundos. Para o processo de convergência com uso da Programação Dinâmica, utilizamos como valor de corte valores de similaridade menores que 0.9. Nota-se neste experimento um ganho em desempenho pela diminuição do suporte local, porém pode-se observar na imagem da direita da Figura 4.4 que ocorre também o aparecimento de algumas falsas correspondências bem como de falsas oclusões.

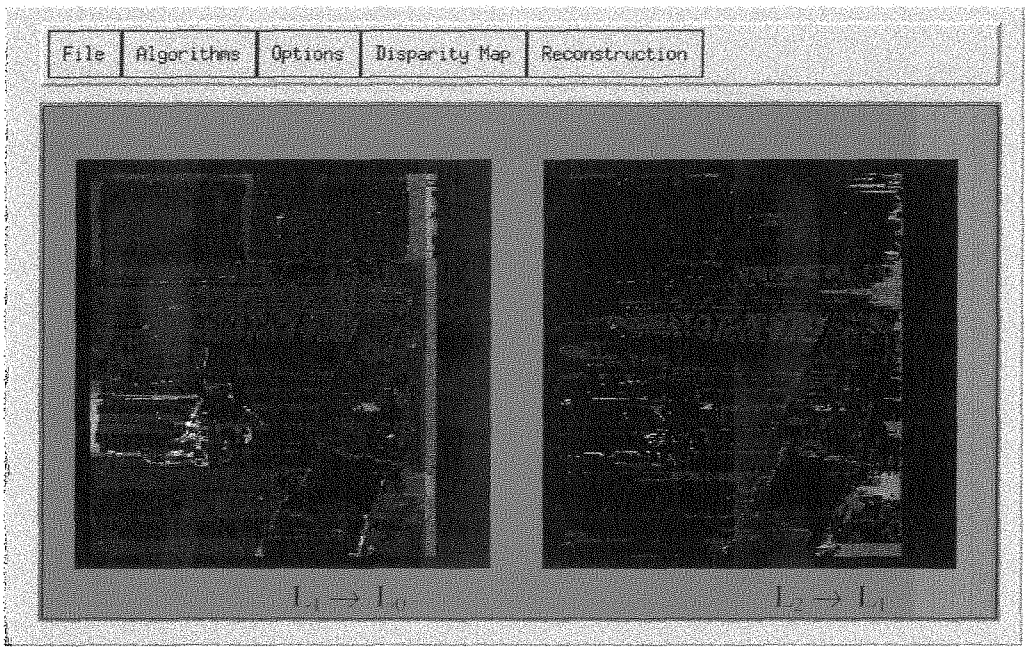


Figura 4.5 – Representação da diferença de imagens entre duas iterações consecutivas

Como forma de testar e validar a convergência do algoritmo, foram realizados vários experimentos onde calculamos valores comparativos da variação da disparidade (ou da variação dos valores de similaridade atualizados pela função L_n) entre uma iteração e outra. Na figura 4.5, a imagem da esquerda representa um resultado (imagem) ilustrativo da diferença entre os mapas de disparidade encontrados com os valores da função $L_1(l,c,d)$ e os valores da função $L_o(l,c,d)$. A imagem da direita representa um resultado (imagem) ilustrando a diferença de disparidade encontrada com os valores da função $L_2(l,c,d)$ e os valores da função $L_1(l,c,d)$.

A partir de cada uma destas imagens foi calculado o desvio padrão que mostra que a diferença entre iterações, a posteriori, é decrescente. Na imagem da esquerda que é a diferença entre L_1 e L_0 o desvio padrão encontrado foi de 9.79, enquanto que na imagem da direita que é a diferença entre L_2 e L_1, o desvio padrão encontrado foi de 6.14.

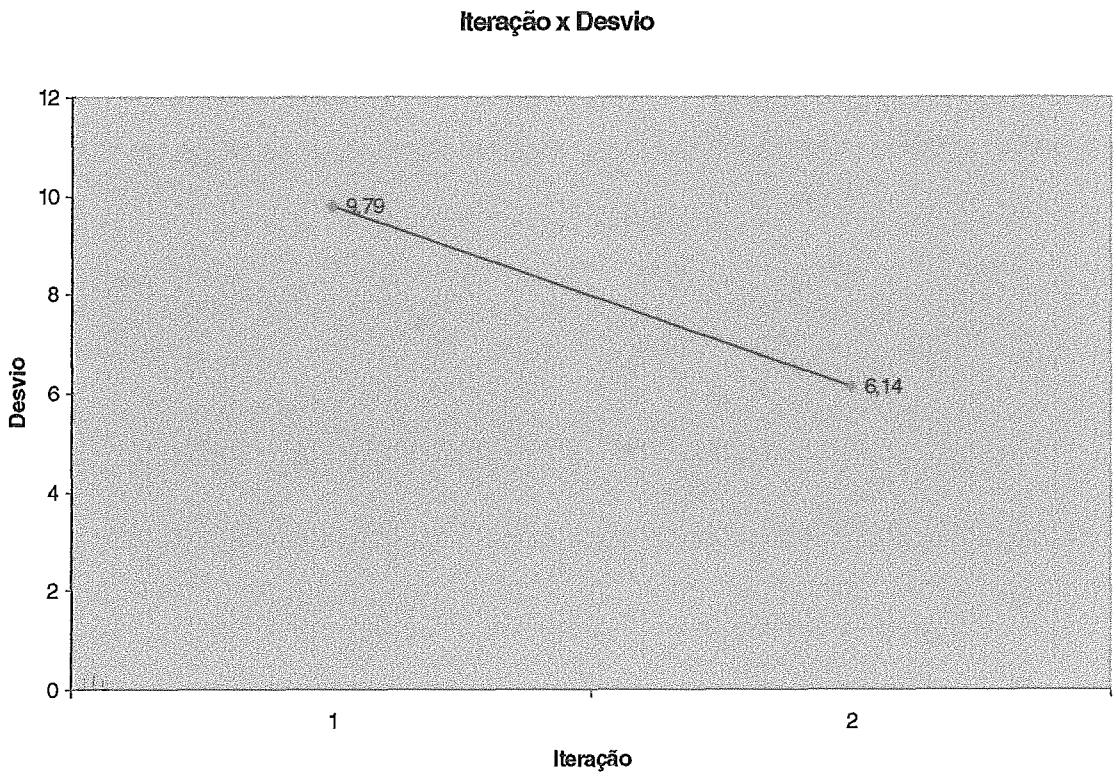


Gráfico 4.1 – Representação desvio x intervalo de iteração

O Gráfico mostrado na Figura 4.1 mostra este resultado. O ponto (1, 9.79) equivale ao resultado descrito pela imagem da esquerda na Figura 4.5, na qual a abscissa corresponde a transição da iteração da função $L_n \rightarrow L_{n+1}$, isto é a transição de $L_0 \rightarrow L_1$. O ponto (2, 6.14) equivale ao resultado descrito pela imagem da direita na Figura 4.5 na qual a abscissa corresponde a transição da iteração de $L_1 \rightarrow L_2$.

Este gráfico representa o decaimento do desvio padrão, conseqüentemente do erro, na determinação dos valores de disparidade entre duas iterações consecutivas.

Figura 4.6 – Comparação com a imagem ideal

Em seqüência, foi realizado um experimento mais elaborado com a finalidade de comparar os resultados do algoritmo proposto nos Capítulos anteriores com resultados supostamente ideais. Este mesmo resultado "ideal", que pode ser visualizado na imagem da direita na Figura 4.6, também foi utilizado por Kanade e Zitnick [01, 33, 34] na determinação do desempenho do algoritmo deles. Podem ser notadas explicitamente as áreas de oclusão, dadas pelas pequenas regiões mais escuras naquela imagem.

Para obter o resultado apresentado na imagem da esquerda da Figura 4.6, usando o algoritmo proposto neste trabalho, foi utilizada uma janela de dimensões 9x9 para o cálculo do somatório das diferenças absolutas. As dimensões do suporte local utilizado foram de 7x7x3 e com quatro iterações foi conseguido o resultado final para a obtenção do mapa de disparidade.

O tempo total, gasto para computar os $L_n(\ l,\ c,\ d\ )$ finais, foi de 13.02 segundos. Para a determinar a convergência com o uso da Programação Dinâmica, utilizamos como valor de corte valores de similaridade menores que 0.88. Vê-se que o resultado obtido (imagem da esquerda) é bem próximo do resultado idealizado (imagem da

direita). Pôde-se notar que ocorreu a formação de alguns elementos correspondentes falsos, na imagem da esquerda, determinados basicamente pelo fator de corte utilizado na PD, no cálculo da função de custo mínimo. É válido comentar que o aumento substantivo no tempo de processamento decorreu do aumento das dimensões do suporte local e da janela para cálculo do somatório das diferenças absolutas. Ainda assim, estamos bem abaixo do tempo de processamento dispendido pelo algoritmo de Kanade [01, 33, 34], da ordem de 20 minutos.

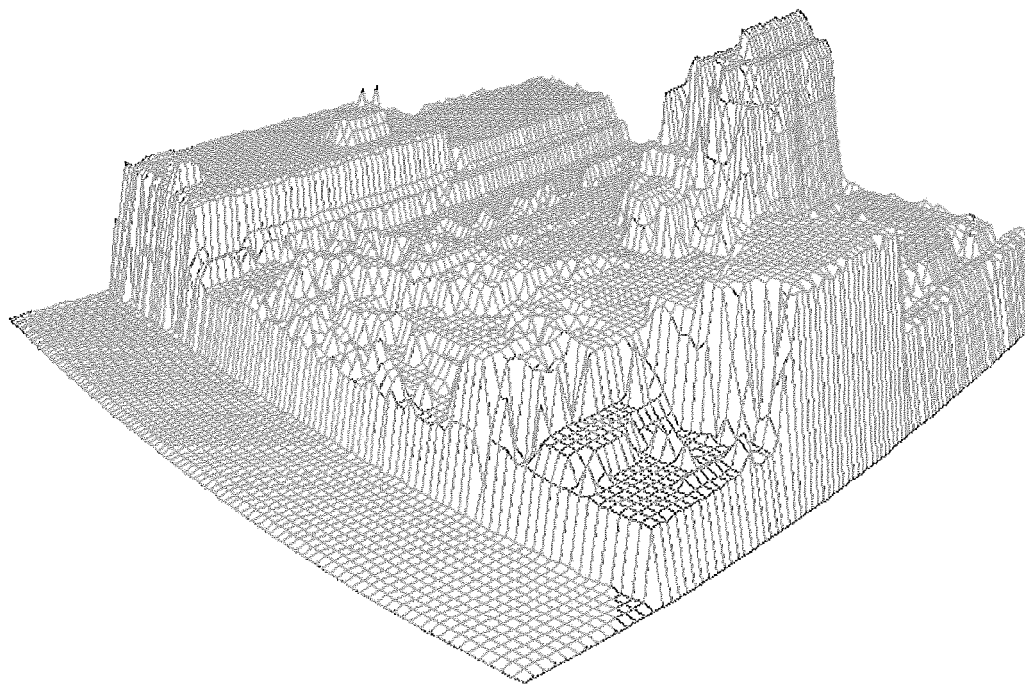


Figura 4.7 – Reconstrução estéreo da mina de carvão

Para ilustrar visualmente os resultados do algoritmo, a Figura 4.7 mostra a reconstrução tridimensional do mapa de disparidade mostrado na Figura 4.4, obtido como resultado do processo estéreo. A reconstrução estéreo mostrada foi obtida com uso do programa DTM que foi brevemente descrito anteriormente. Ela é formada a partir de um arquivo de entrada na forma xyz, isto é o x corresponde à coordenada x do pixel na imagem, o y corresponde à coordenada y do pixel na imagem e o z corresponde à disparidade na imagem, que é o valor do tom de cinza correspondente àquele pixel devidamente equalizado para que se possa notar as diferenças no mapa de alturas.

Convém ressaltar que o processo acima é uma aproximação grosseira do que seria a reconstrução tridimensional a partir de um mapa de disparidade. Ela não

considera aspectos relacionados com calibração de câmera e orientação absoluta em relação à cena real, bem como outros aspectos como distorções, e erros no processo estéreo. Notamos que, no presente trabalho, não nos detemos sobre aspectos relacionados à reconstrução tridimensional em sí, mas sim aos aspectos relacionados à construção dos mapas de disparidade.

Além dos experimentos descritos acima, outros experimentos foram realizados ainda no sentido de validar a nossa proposta. A Figura 4.8 representa um par estéreo de uma cena onde aparece uma sala de estudos, usado nestes experimentos complementares.

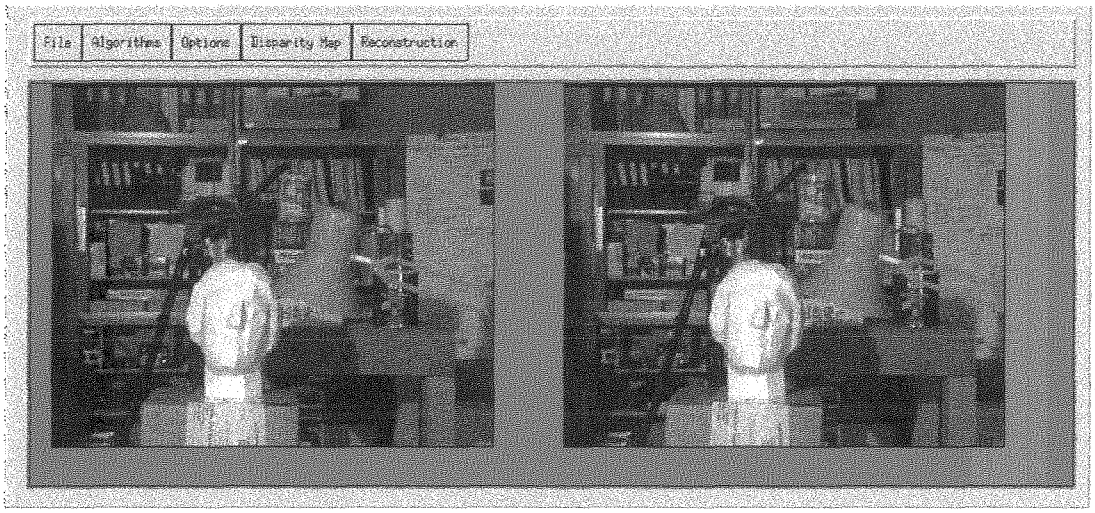


Figura 4.8 – Par estéreo de tamanho 385x289

As imagens mostradas na Figura 4.8 possuem diversos elementos ao fundo ("background"), fazendo que seja difícil a representação detalhada destes no mapa de disparidade final. Segundo o algoritmo proposto, a tendência é que todos esses elementos do background se aglutinem em uma mesma escala de cinza, isto devido à hipótese de continuidade da disparidade. Ainda, devido a esse par estéreo possuir objetos de forma geométrica bastante heterogênea, usamos, numa fase de pré-processamento um operador morfológico, visando evitar ou diminuir os efeitos maléficos causados pelas muitas arestas e objetos com textura variada existentes na cena. Do mesmo modo que a imagem da mina de carvão, os valores iniciais de $L_n(l,c,d)$, para este par estéreo (ver imagem da esquerda da Figura 4.9), já nos fornece uma boa visão de como será o mapa de disparidade resultante.

Figura 4.9 - À esquerda estão os L_o e à direita está o mapa de disparidade obtido.

Para obtenção do resultado mostrado pela imagem da direita da Figura (4.9), foi utilizada uma janela de tamanho 3x3 para o cálculo do somatório das diferenças absolutas. As dimensões do suporte local utilizado foram 5x5x3, e com apenas três iterações conseguimos o resultado final (imagem da direita) para a obtenção dos mapas de disparidade.

O tempo para a determinação dos valores iniciais foi de 3.91 segundos e o resultado final foi obtido em 8..82 segundos. Para o cálculo da função de custo mínimo na aplicação da Programação Dinâmica, utilizamos como valor de corte valores de similaridade menores que 0.9. Como citado acima, nesse par de imagens foi utilizado um operador morfológico para conectar pixels correspondentes ao mesmo objeto, numa fase de pré-processamento.

Note que os valores iniciais $L_0(l, c, d)$, já nos dão um bom indicativo de como deverá ser o mapa de disparidade resultante do processo de PD. Notamos que neste par estéreo o algoritmo não funcionou adequadamente para a detecção de oclusão, embora as bordas dos objetos da cena estejam bem delimitadas.

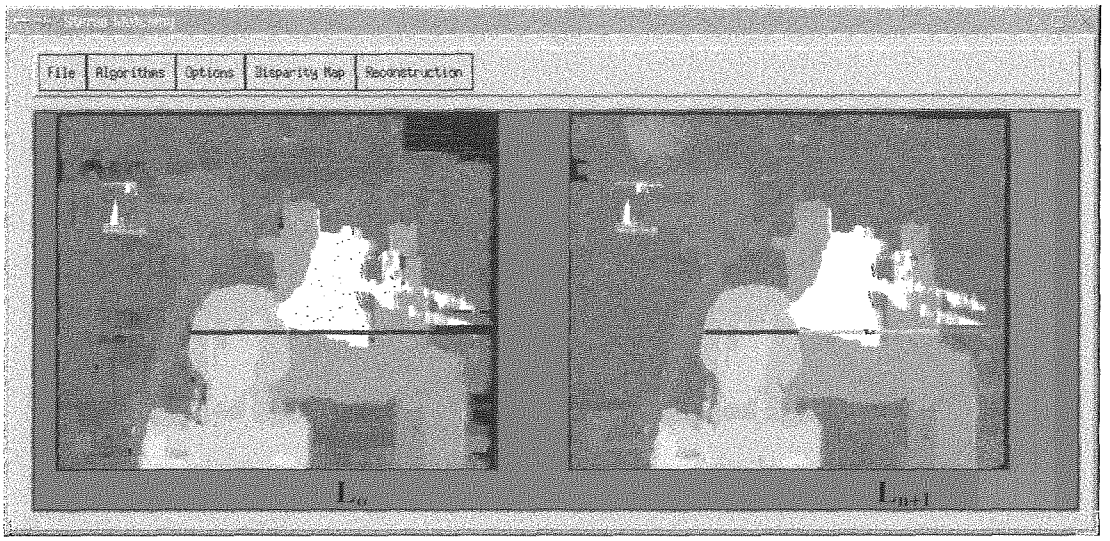


Figura 4.10 - Aparecimento de falsas correspondências e/ou oclusões

Da mesma forma que nos experimentos anteriores, aumentamos o valor de corte usado pela função de custo mínimo na PD para mostrar os efeitos desse aumento. Para obter o resultado inicial para $L_0(\ l,\ c,\ d\ )$, visualizado na imagem esquerda da Figura 4.10, foi utilizada uma janela de dimensões 3x3 para o cálculo do somatório das diferenças absolutas. O suporte local foi de 5x5x3, e com apenas três iterações conseguimos o resultado representado pela imagem da direita para a obtenção dos mapas de disparidade.

O tempo para a determinação dos valores iniciais foi de 3.91 segundos e o resultado final foi obtido em 8.82 segundos. Usamos como valor de corte para a Programação Dinâmica valores de similaridade menores que 0.95. Nessa imagem (da direita) foi também utilizado um operador morfológico para conectar pixels correspondentes ao mesmo objeto, numa fase de pré-processamento.

Nota-se uma linha horizontal de tonalidade escura, que aparece tanto para o mapa determinado pelos valores iniciais $L_0(l,\ c,\ d)$ quanto pelos valores finais. Estas linhas correspondem aos valores em que o Custo Mínimo, encontrado a partir da Programação Dinâmica, naquela faixa do plano tende a um valor baixo próximo de zero, ou seja, uma correspondência e/ou oclusão falsa é determinada.

Da mesma forma, nos resultados mostrados na Figura 4.11, houve uma diminuição na área do suporte local, o que acarretou num ganho de desempenho, porém também gerou algumas falsas correspondências.

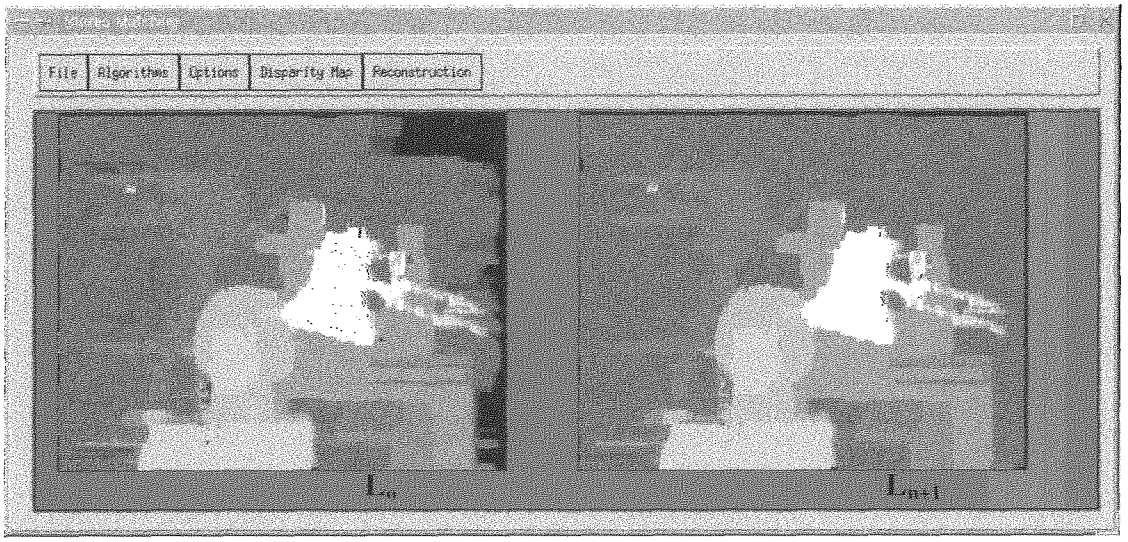


Figura 4.11 – Exibida com suporte local 3x3x3

Neste experimento (Figura 4.11), foi utilizada uma janela de dimensões 3x3 para o cálculo do somatório das diferenças absolutas na determinação de $L_0(l, c, d)$. O suporte local usado foi de 3x3x3 e com apenas três iterações conseguimos o resultado representado pela imagem da direita para a obtenção do mapa de disparidade final.

O tempo para a determinação dos valores iniciais foi de 3.91 segundos e o resultado final foi obtido em 8.03 segundos. Para a Programação Dinâmica, utilizamos como valor de corte valores de similaridade menores que 0.9.

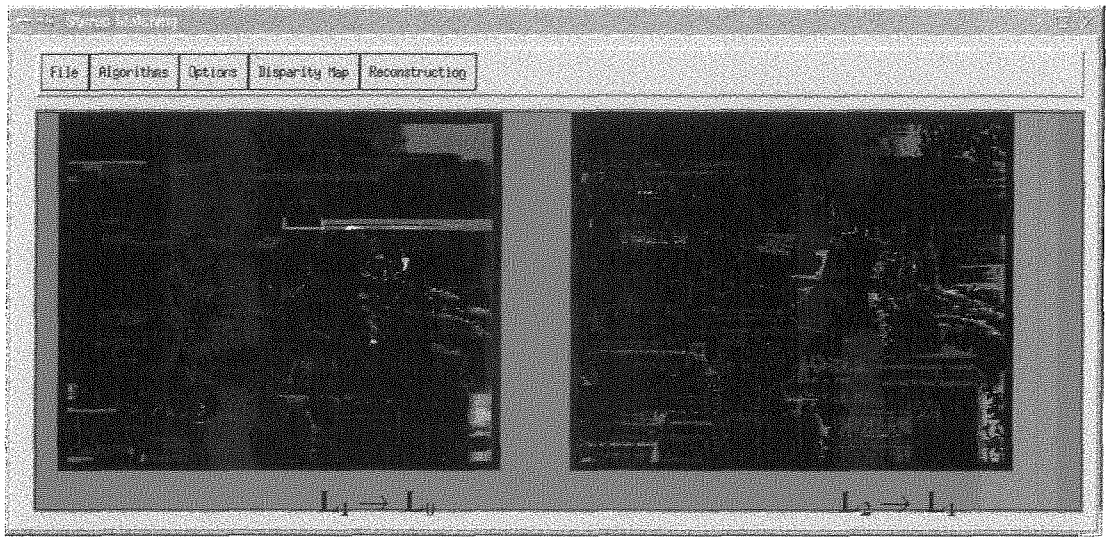


Figura 4.12 – Representação da diferença de imagens entre duas iterações consecutivas

Analogamente, mostramos o erro que determina a convergência do processo, para este par. A imagem da esquerda da Figura 4.12 representa um resultado a diferença entre o mapa de disparidade encontrado com os valores da função $L_1(l,c,d)$ e os valores da função $L_o(l,c,d)$. A imagem da direita representa a diferença de disparidade encontrada com os valores da função $L_2(l,c,d)$ e os valores da função $L_1(l,c,d)$.

A partir de cada uma destas imagens foi também calculado o desvio padrão, que, entre iterações a posteriori, é decrescente. Na imagem da esquerda (ou primeira imagem diferença equivalente a L_1 - L_0), o desvio padrão encontrado foi de 8.73, enquanto que na imagem da direita (L_2 - L_1) o desvio padrão encontrado foi de 5.39.

Estes resultados foram transcritos no gráfico mostrado na Figura 4.13. No gráfico, o ponto (1, 8.73) equivale ao resultado dado pela imagem da esquerda na Figura 4.12, na qual a abscissa corresponde a transição da iteração $L_0 \rightarrow L_1$. O ponto no gráfico dado por (2, 5.39) equivale ao resultado descrito pela imagem da direita na Figura 4.12, na qual a abscissa corresponde a transição da iteração da função $L_1 \rightarrow L_2$.

Este gráfico representa também o decaimento do desvio padrão entre duas iterações consecutivas.

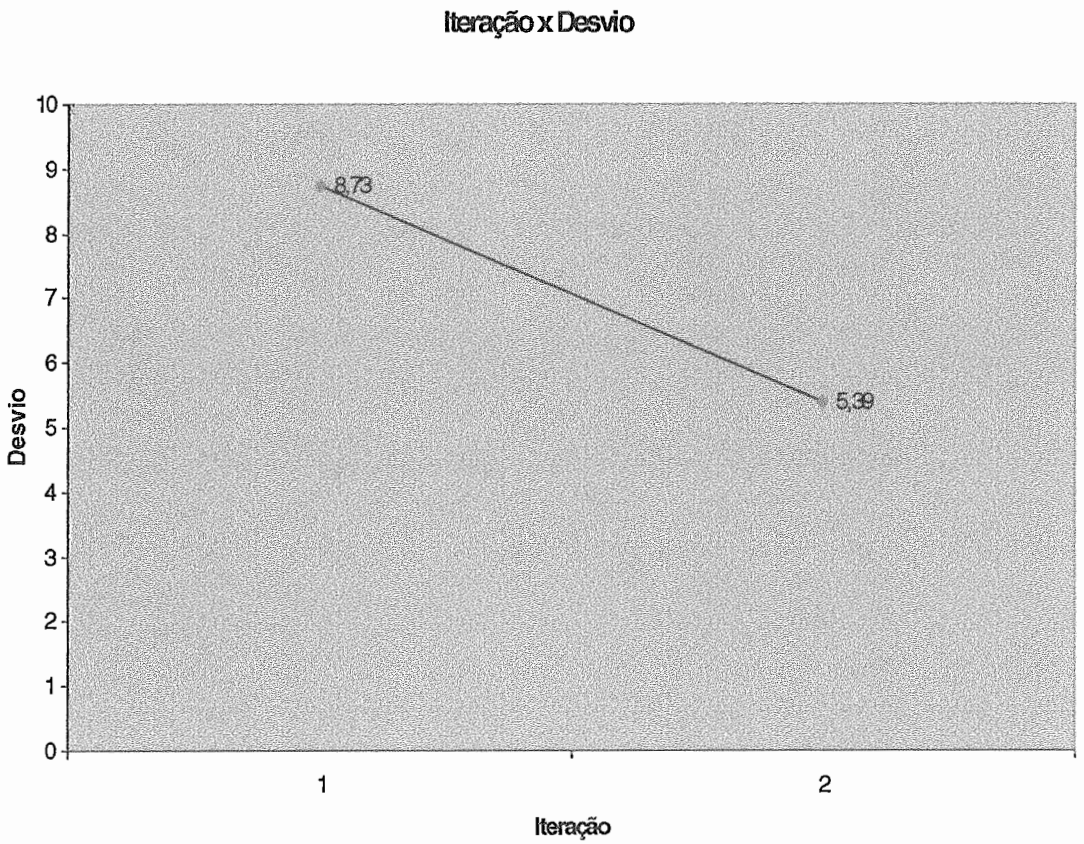


Gráfico 4.2 – Representação desvio x intervalo de iteração

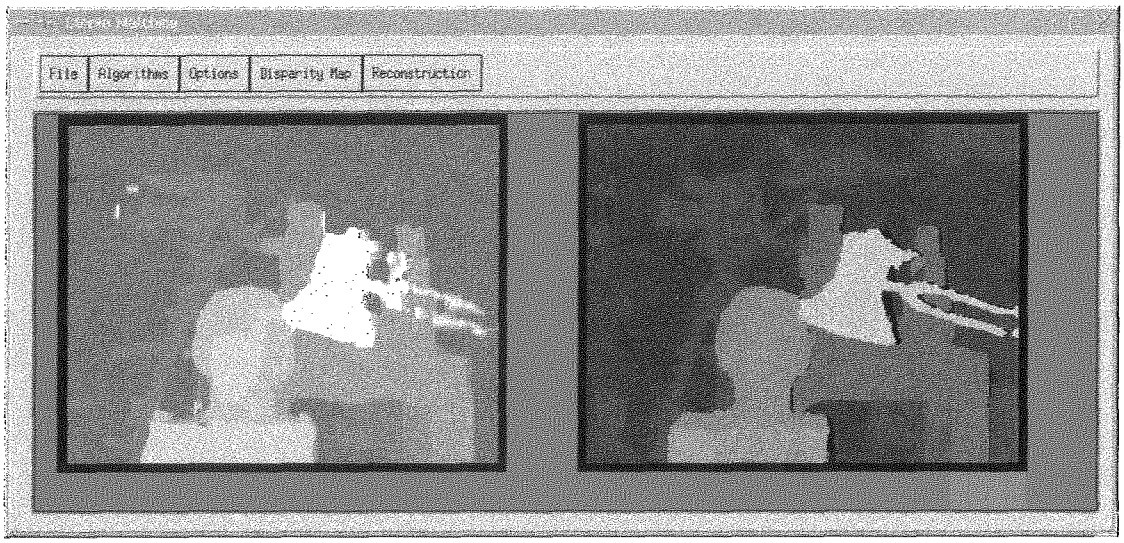


Figura 4.13 – Comparação com outro resultado

Podemos fazer uma comparação do algoritmo proposto por nós com o de Kanade e Zitnick [01, 33, 34] analisando a Figura 4.13. Uma comparação entre mapas de disparidade resultantes dos dois métodos mostra que o algoritmo proposto neste trabalho é quase tão robusto, porém muito mais rápido que o algoritmo deles. A imagem a esquerda representa o mapa de disparidade resultante do nosso método e a da direita o do método deles. Podemos a esta altura ressaltar que há algumas vantagens e também algumas desvantagens de nosso método com relação ao algoritmo deles [01, 33, 34]. A vantagem do nosso resultado é que com apenas 3 iterações já conseguimos obter um resultado bastante adequado com o que poderia ser chamado de resultado "ideal", enquanto no resultado deles [01, 33, 34] foram efetuadas 15 iterações. Logo, nosso algoritmo torna-se bem mais rápido. A desvantagem entre nosso resultado e o deles [01] é que não detectamos oclusões de forma explícita, embora isto ocorra com um certo nível para algumas imagens não ruidosas.

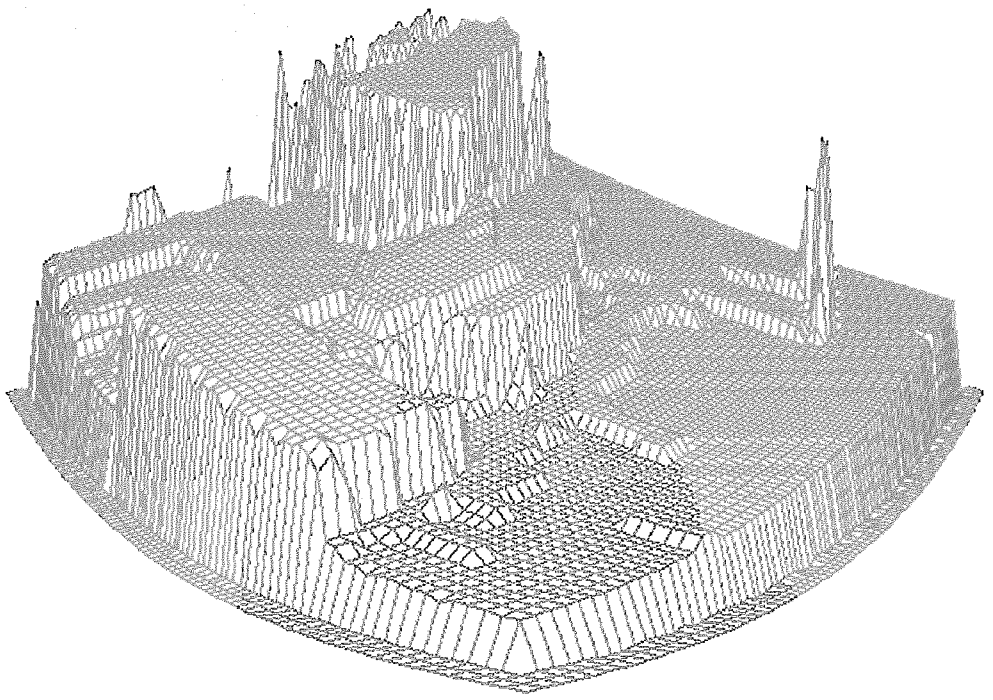


Figura 4.14 – Reconstrução estéreo da sala de estudos

A Figura (4.14) mostra a reconstrução estéreo do resultado obtido por nós, mostrado na Figura 4.9. A malha mostrada foi também determinada a partir de um arquivo de entrada na forma xyz, onde o x corresponde a coordenada X do pixel na imagem, o y corresponde a coordenada Y do pixel na imagem e o z corresponde a disparidade na imagem devidamente equalizado.

Finalmente, para confirmar os resultados obtidos nos experimentos anteriores, selecionamos o par estéreo do vulcão Santa Helena, localizado em Seattle, WA. As imagens originais exibidas na Figura 4.15 foram obtidas bem após a erupção do vulcão em 1982, quando em um dos raros dias em que ele não está encoberto por nuvens, e possuem dimensões de 256 x 256 pixels. Nota-se a textura bem diferente das imagens usadas anteriormente. Nota-se também que a distância de projeção é bem maior que a das cenas anteriores. Estes aspectos podem acarretar em problemas para o processo estéreo.

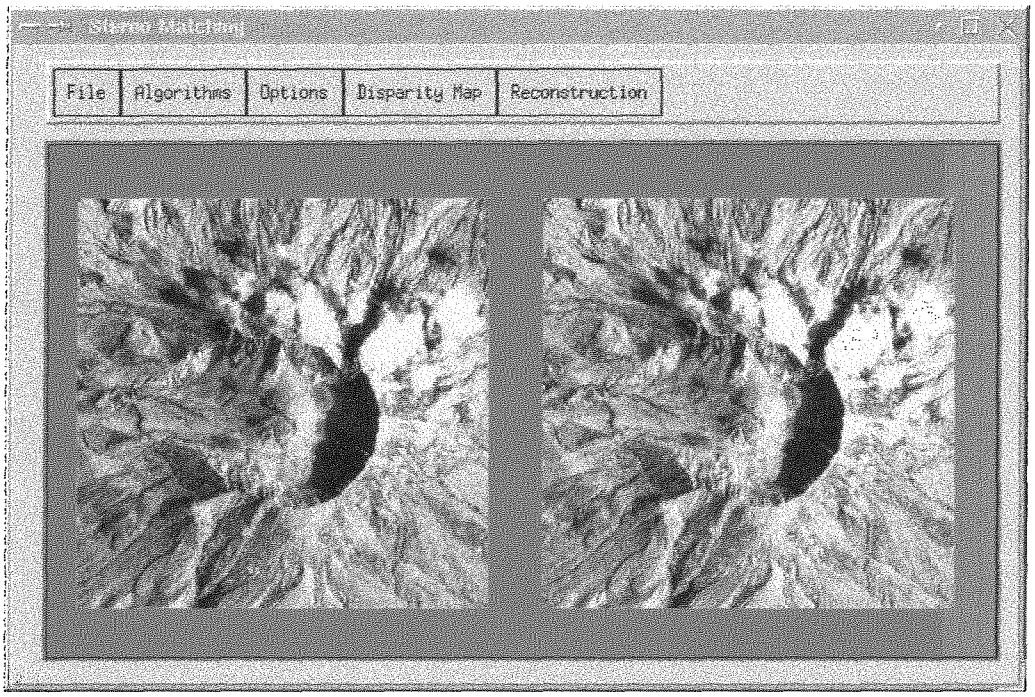


Figura 4.15 – Par estéreo de tamanho 256x256

A Figura 4.16 mostra os resultados conseguidos usando uma janela de 5x5 para determinação dos valores iniciais de similaridade (a imagem da esquerda mostra o resultado inicial conseguido para a função L_0) e usando um suporte local de 5x5x3 no cálculo da função de custo mínimo no processo de PD (a imagem da direita mostra o resultado final). Nota-se algumas falhas de correspondência, determinadas pela qualidade de textura da cena em função dos parâmetros acima.

É possível prever, numa melhor análise visual das imagens originais, que no cálculo de L_0 poderão ser determinados muitos correspondentes falsos caso não se trabalhe com uma janela de similaridade de dimensões razoáveis. Gonçalves e Oliveira, operando sobre as mesmas imagens em [02], descrevem experimentos usando janelas com dimensões variando de 12x12 até 32x32 e eles conseguiram os melhores "matchings" com janelas de 24x24.

Desta forma, em outro experimento, aumentamos as dimensões da janela de similaridade para determinação de L_0 para 9x9 e também o suporte local para 9x9x3, visando observar o comportamento do nosso algoritmo. O resultado, que pode ser observado na Figura 4.17, foi uma melhoria tanto na determinação de valores a função L_0 (imagem da esquerda) quanto para o mapa de disparidade final (imagem da direita)

gerado pelo processo envolvendo PD. Ainda neste experimento, através de uma análise visual, usando um aparelho de estereoscopia sobre as imagens impressas, pôde-se notar, nas imagens finais, a detecção de algumas zonas de oclusão (existem algumas nuvens nas imagens que geram oclusão de píxels nas imagens). Convém ressaltar que o aparelho usado permite uma visão tridimensional melhor da cena do que por simples matching dos olhos no estereograma formado pelas duas imagens.

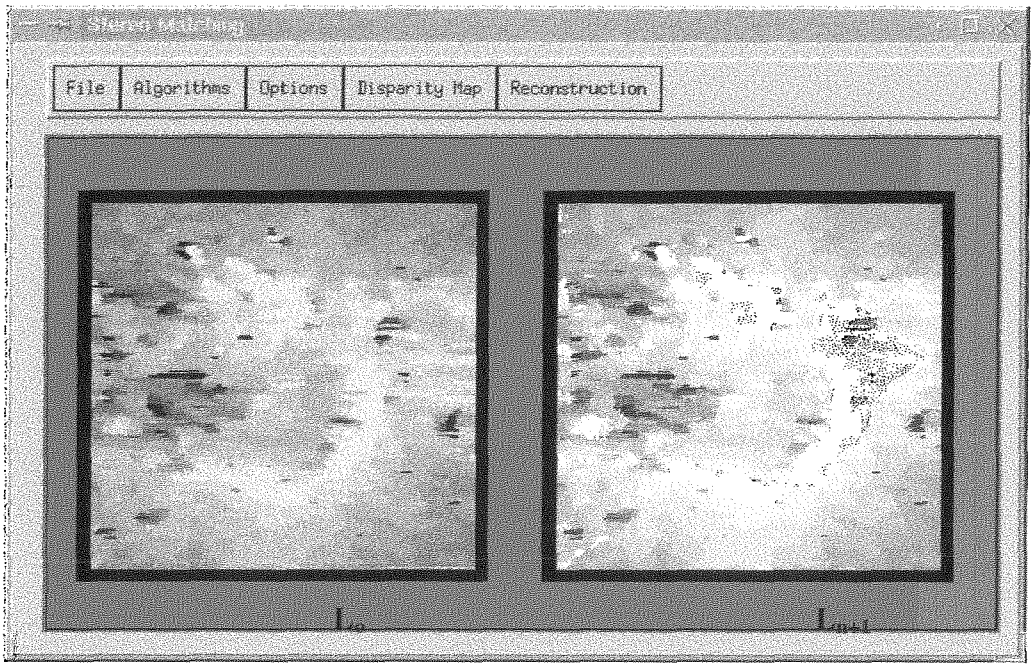


Figura 4.16 – Exibida com suporte local 5x5x3

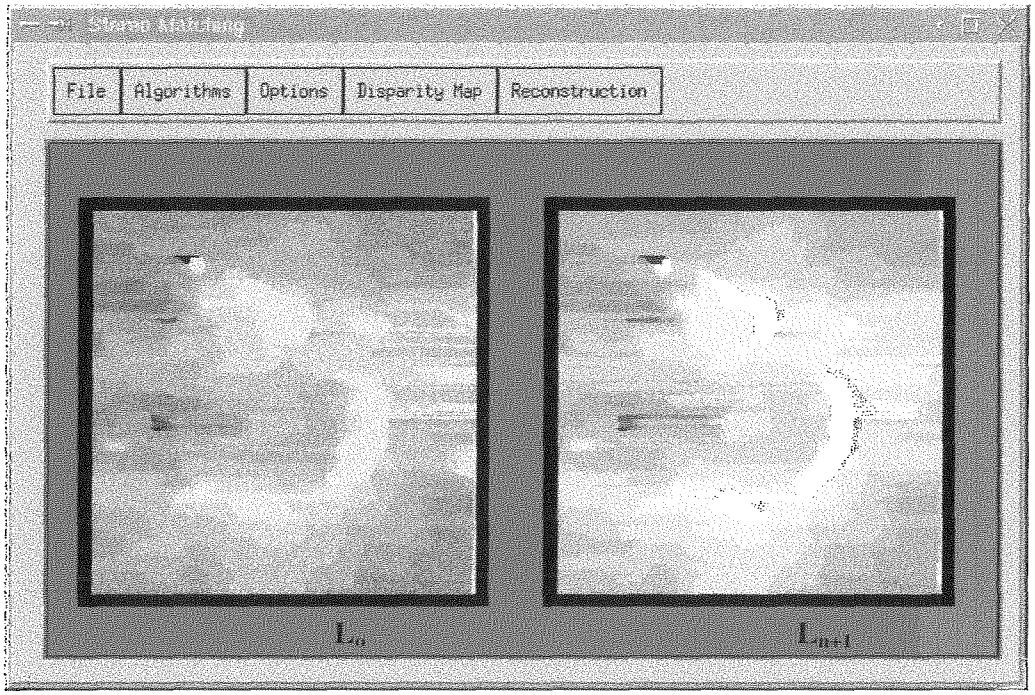


Figura 4.17 – Exibida com suporte local 9x9x3

Podemos verificar pelos resultados mostrados neste Capítulo, que o algoritmo proposto por nós realiza de forma satisfatória um balanceamento entre robustez ou precisão e velocidade de processamento. Convém ressaltar que com apenas um pouco de perda precisão na determinação de áreas de oclusão (suavidade implica em perda de detalhes), conseguimos melhorias substanciais na velocidade de processamento, da ordem de 5 vezes ou mais em relação ao algoritmo anterior [01, 33, 34]. Em termos de cifras, baixamos o número de iterações, que com o algoritmo de Kanade e Zitnick era de 15, para apenas 3, com resultados de precisão satisfatórios.

Capítulo 5

Discussões, Conclusões e Trabalhos Futuros

A intenção deste trabalho foi implementar um algoritmo para determinação da disparidade estéreo usando uma abordagem em que nos empenhamos ao máximo em acelerar o processo, sem contanto perder a qualidade na determinação da disparidade. Se considerarmos que o nosso algoritmo consegue reduzir o tempo de processamento para a obtenção do resultado final (mapa de disparidade refinado) em torno de 2 a 5 vezes em relação ao algoritmo proposto por Zitnick e Kanade [01, 33, 34], podemos considerar que obtivemos excelentes resultados no que se refere a velocidade. Vários foram os fatores que contribuíram para essa otimização. Primeiramente, a escolha de uma função de inicialização, a qual denominamos de $L_o(l,c,d)$, com melhores condições de acerto do que a deles [01, 33, 34] e, em segundo, a inclusão da técnica de programação dinâmica que considera as restrições de unicidade e suavidade (ou continuidade) para acelerar o processo de geração do mapa de disparidade final. Finalmente, para alguns dos resultados apresentados, utilizamos um suporte local menor o que também ajudou a acelerar o processo.

No tocante à precisão, não nos preocupamos muito em detectar oclusões tão precisamente quanto no algoritmo do Zitnick e Kanade em [01, 33, 34], principalmente devido ao nosso primeiro objetivo ser a implementação de um algoritmo que seja eficiente e robusto na prática. Desta maneira podemos dizer que o algoritmo por eles proposto [01, 33, 34] determina oclusões de uma forma mais explícita que o nosso, porém, como já salientado, a perda em tempo de processamento não compensa o ganho em precisão. Ou seja, para que se tenha realmente um bom tratamento das oclusões usando o algoritmo deles, se faz necessário um número bem maior de iterações (em torno de 15). Convém ressaltar que o nosso algoritmo, apesar de não ser tão preciso, ainda assim permite detectar oclusões quase com o mesmo grau de precisão que o deles,

conforme pôde ser verificado nos experimentos apresentados no último capítulo. Além do mais, pôde ser verificado ainda que o nosso algoritmo consome um tempo muito inferior ao deles.

Um dos primeiros trabalhos que realizamos foi a execução de exaustivos testes no sentido de determinar os melhores parâmetros a serem utilizados pelo algoritmo visando diminuir erros primários. Note que se aumentarmos o suporte local, o tempo de execução também aumenta. Então, tentamos encontrar um tamanho de suporte local mais adequado para que não aumentasse muito o tempo de execução e que fornecesse bons valores. Convém ressaltar que as dimensões do suporte local são uma função da quantidade de textura (ou discriminabilidade) das imagens. Ainda, ao iniciarmos nossas implementações, utilizamos a mesma função de inicialização (para determinação de $L_o(l,c,d)$) proposta pelo Zitnick e Kanade em [34]. Observamos após vários testes que esta função não gerava boas estimativas iniciais para a disparidade. A partir de então, formulamos uma outra função de inicialização $L_o(l,c,d)$ com a qual obtivemos valores iniciais altamente satisfatórios, como pôde visto na Figura 4.2. Com esta nossa nova função $L_o(l,c,d)$ os valores utilizados por nosso algoritmo foram restritos entre 0.1 e 1, enquanto que no algoritmo deles [01, 33, 34] os valores são restritos a 0 e 1. Isso nos permitiu alterar o valor limite para a detecção de oclusão, facilitando a convergência do algoritmo.

Podemos ainda citar como uma qualidade do algoritmo aqui proposto, o uso das hipóteses de Marr e Poggio [20, 21] (unicidade e continuidade da disparidade). Estas restrições foram exploradas com o uso da técnica de programação dinâmica, permitindo aos valores da função de atualização convergissem mais rapidamente. Isto permitiu reduzir também a quantidade de memória consumida, cerca de 2 vezes menos que no algoritmo de Zitnick e Kanade [01, 33, 34].

Assim, podemos finalmente ressaltar que, pelos resultados obtidos e mostrados acima, a nossa proposta de algoritmo é tão boa, no que se refere à precisão na detecção de oclusão, quanto a proposta de Zitnick e Kanade [01, 33, 34], com a vantagem principal de consumir um tempo de processamento bem aquém da proposta deles.

5.1 Trabalhos Futuros

Ao contrário da maioria dos algoritmos estéreo que geram mapas de profundidade com tempo de execução bastante altos, o algoritmo aqui apresentado gera mapas de profundidades através de um processo de iteração rápido e eficiente. Como sugestão para aumentar a eficiência do algoritmo aqui apresentado, em termos de tempo de processamento, pode-se tentar adaptar o algoritmo para usar processamento distribuído e/ou paralelo, com memória compartilhada. O uso de processamento distribuído pode permitir ainda uma possível extensão do algoritmo para processar seqüências de pares de imagens estéreo, o que poderia ser imediatamente aplicado em visão ativa ou visão robótica.

Uma outra possibilidade de trabalho futuro, a título de melhoria no algoritmo, seria tentar melhorar a robustez da técnica de Programação Dinâmica aqui descrita. Poderia estudar maneiras mais efetivas e eficientes de se descartar o aparecimento das falsas oclusões. Isto poderia acarretar em uma melhor precisão no tocante à detecção de oclusão.

Finalmente, ressaltamos que o algoritmo apresentado pode ser aplicado à área de visão robótica, devido à possibilidade de geração de mapas de disparidade em um curto espaço de tempo. Claro, os tempos de processamento discutidos estão ainda longe de uma aplicação em tempo real, porém note que aqui não usamos alguma forma de redução ou abstração de dados, o que pode ser feito com algumas adaptações no algoritmo.

REFERÊNCIAS BIBLIOGRAFICAS

[01] ZITNICK, C. L., KANADE, T. "A Cooperative Algorithm for Stereo Matching and Oclusion Detection," *In IEEE Transactions on Pattern Analysis and Machine Intelligence,* v. 22, n. 7, pp. 675-684, July 2000.

[02] GONÇALVES, L., OLIVEIRA, A., "*Pipeline Stereo Matching in Binary Images*". XI International Conference on Computer Graphics and Image Processing (SIBGRAPI'98), pp. 426-433, October 1998.

[03] MARR, D., *Vision, A Computational Investigation into the Human Representation and Processing of Visual Information.* MIT Press, 1982.

[04] HORN, B. K. P., Robot Vision. *MIT Press, 1986.*

[05] GRIMSON, W. E. L., From Images to Surfaces: A Computational Study of the Human Early Visual System., MIT Press. Cambridge, Mass. 1981.

[06] BALARD, D. H., BROWN, C. M., Computer Vision. Prentice-Hall. Englewood Cliffs, NJ, 1982.

[07] PAPOULIS, A., Probability, Random Variables, and Stochastic Proccesses. MacGRAW-HILL, 1991

[08] JAIN, A. K., Fundamentals of Digital Image Processing. Prentice Hall, Inc. New Jersey, 1989.

[09] NISHIHARA, H. K., "Practical Real-Time Imaging Stereo Matcher." *Technical Report. Optical Engeneering.* MIT, Artificial Intelligence Laboratory. 1984

[10] NISHIHARA, H. K., *"Minimal Meaningful Measurements Tools"*. Technical Report. Teleos Research. 1991.

[11] NISHIHARA, H. K., *"Real-Time Tracking of People Using Stereo and Motion."* Technical Report. *MIT*, Artificial Intelligence Laboratory. 1984

[12] HUBER, E., KORTENKAMP, *D., "Using Stereo Vision to Pursue Moving Agents with a Mobile Robot. Metrica Inc. Robotics and Automation Group, NASA Johnson Space Center-ER4. Houston, Texas, 1995."* Proceedings of IEEE Conference on Robotics and Automation, *1995.*

[13] KOLLER, D., LUONG, T., MALIK, J., "Binocular Stereopsis and Lane Marker Flow for Vehicle Navigation: Lateral and Longitudinal Control." *Technical Report.* University of California, 1994

[14] WESSLER, M., *"A Modular Visual Tracking System."* Technical Report. *MIT,* Artificial Intelligence Laboratory. 1996.

[15] WOOD, G., "Realities of automatic correlation problem". *Photogrametric Engineering and Remote Sensing*, v. 49, pp. 537-538, 1983.

[16] KANADE T., OKUTOMI, M., "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, pp. 920-932, 1994.

[17] PANTON, D. J., *"A flexible approach to digital stereo mapping"*. Photogram. Eng. Remote Sensing, v. 44, n. 12, pp. 1499-1512, 1978.

[18] OHTA, Y., KANADE T., *"Stereo by intra- and inter-Scanline search using dynamic programming"*. IEEE Transactions on Pattern Analysis and Machine Intelligence. v. 7, n. 2, pp 139-154, 1985.

[19] JULESZ, B., Foundations of Cyclopean Perception. *University of Chicago* Press. 1971.

[20] MARR, D., POGGIO, *T., "Cooperative computacion of stereo disparity."* Science, v. 194, pp. 209-236, 1976.

[21] MARR, D., POGGIO, T., "A Computational Theory of Human Stereo Vision." *Proceedings of the Royal Society of London B*, v. 204, pp. 301-328, 1979.

[22] COLLINS, R., *"A space-sweep approach to true multi-image matching."* in IEEE Computer Society Conference on Computer Visison and Pattern Recognition, San Francisco, CA, 1996, pp. 358-363.

[23] SHARSTEIN, D., SZELISKI, R., *"Stereo matching with nonlinear diffusion,"* International Journal of Computer Vision, v. 28, n. 2, pp. 155-174, 1998.

[24] SZELISKI, R., GOLLAND, P., *"Stereo matching with transparence and matting,"* in Sixth International Conference on Computer Vision., Bombay, India, pp. 517-524, 1998.

[25] GRINSON, W. E. L., *"Computacional experiments with a feature based stereo algorithm"* IEEE Transactions on Patten Analysis and Machine Intelligence, v. 7, n. 1, pp. 17-34, 1994.

[26] POLLARD, S. B., MAYHEW, J. K. W., FRISBY, J., P. *"Pmf: A stereo correspondence algorithm using a disparity gradient limit,"* Perception, v. 14, pp. 449-470, 1985.

[27] PRAZDNY, K., *"Detection of binocular disparities,"* Biologocal Cybernetics., v. 52, n. 2, pp. 93-99, 1985.

[28] BEULHUMEUR, N. P., MUMFORD, A. D., *"A bayesian treatment of the stereo correspondence problem using half-ocluded regions,"* In Proc. IEEE Conf. On Computer Vision and Pattern Recognition, 1992.

[29] GEIGER, D., LADENDORF, B., YUILLE, A., *"Oclusions and Binocular Stereo,"* International Journal of Computer Vision (IJVC), v. 14, 1995, pp. 211-226.

[30] BOBICK, A., INTILLE, S., *"Disparity –space images and large oclusion stereo,"* European Conference on Computer Vision, J-O. Eklundh (ed), Stockholm, Sweden, v. 801, pp. 179-186, May 1994.

[31] TZOVARAS, D., GRAMMALIDIS, N., STRINTZIS, M. G., *"Object-Based Coding of Stereo Image Sequences Using Joint 3-D Motion/Disparity Compensation,"*

In Proc IEEE Transaction on Circuits and Systems For Video Technology, v. 7, n. 2, pp. 312-327, April 1997.

[32] MATIAS, I. O., OLIVEIRA, A., *"Enhancing the Volumetric Aprooach to Stereo Matching"* XIII International Conference on Computer Graphics and Image Processing (SIBGRAPI'00), pp. 347, 2000.

[33] ZITNICK, C. L., KANADE, *T. "A Cooperative Algorithm for Stereo Matching and Oclusion Detection,"* CMU Technical Report CMU-RI-TR-99-35, 1999.

[34] ZITNICK, C. L., KANADE, T. "A Volumetric Iterative Approach to Stereo Matching and Oclusion Detection," CMU Technical Report CMU-RI-TR-98-30, 1998.