4
Multi-Resolução com Cortes de Grafo para Estéreo

As abordagens para o problema de estéreo que utilizam algoritmos baseados em cortes de grafo geralmente apresentam excelentes resultados em termos de acurácia. Entretanto, dada a grande quantidade de pixels, rótulos e arestas manipulados pelo algoritmo, definindo um espaço de busca intratável em um contexto em que o tempo de processamento é um fator que deve ser levado em consideração, o desempenho apresentado por esses algoritmos quase sempre torna seu uso inviável. Dessa forma, a redução do número de pixels e de rótulos utilizados pelo algoritmo é um fator importante para a implementação de um algoritmo mais eficiente.

Worby propõe em (19) e (18) um método de multi-resolução para estéreo e fluxo óptico com o propósito de reduzir o número de variáveis (nós e arestas do grafo) manipuladas pelo algoritmo de cortes de grafo. Nesse contexto propõe três algoritmos para reduzir o número de rótulos utilizados. Entretanto seus algoritmos usam necessariamente o algoritmo de troca-$\alpha\beta$, já que para combinar o uso de estéreo e motion em único algoritmo foi necessário que V (o termo de suavidade) fosse definido como uma função que não é uma métrica (veja seção 3.2.2). Dessa forma a complexidade do algoritmo é $\mathcal{O}(mn^2)$ (sendo m o número de pixels e n o número de rótulos).

Neste capítulo vamos explorar a ideia proposta por Worby visando somente sua aplicação para o uso em estéreo. Dessa forma poderemos alterar o método proposto para que use o algoritmo de expansão-α, obtendo assim um algoritmo mais eficiente com complexidade $\mathcal{O}(mn)$. Apresentaremos também os algoritmos LDNR, EL e EAC, propostos por Worby para a redução do número de rótulos, e discutiremos as alterações que são necessárias nos mesmos para fazerem uso do algoritmo de expansão-α.

4.1
Multi-Resolução para Estéreo

Nesta seção vamos apresentar a abordagem de multi-resolução com cortes de grafo para o problema de estéreo proposta nesta dissertação e descrever as etapas envolvidas. A Figura 4.1 mostra uma visão geral do método e seus

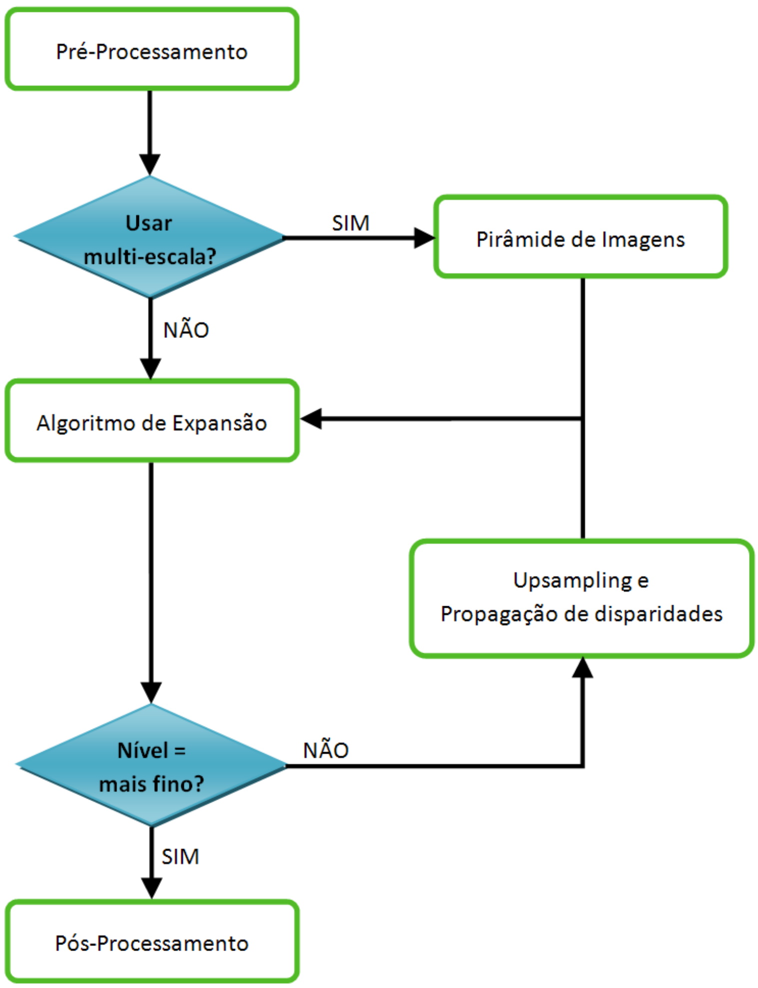

Figura 4.1: Fluxograma geral do método de Multi-Resolução com Cortes de Grafo para estéreo.

estágios. Primeiramente as imagens de entrada alimentam o algoritmo no estágio de *Pré-Processamento*, o qual realiza outras tarefas de inicialização e configuração de parâmetros..

Após o estágio de pré-processamento estão os módulos principais do algoritmo, os quais são responsáveis pela implementação de multi-resolução e cortes de grafo. No estágio *Pirâmide de Imagens* ocorre a criação das pirâmides das imagens de entrada. Em seguida, no estágio *Algoritmo de Expansão* ocorre a construção do grafo do qual vamos encontrar um corte mínimo, como discutimos no Capítulo 3

Na seção 3.2.2 foi discutido que o algoritmo de expansão-α só poderia ser usado se V fosse uma métrica. Worby em (19) utilizou o método de Multi-Resolução com cortes de grafo mas fazendo uso do algoritmo de troca-$\alpha\beta$. Isso era necessário pois Worby propunha um método que envolvia estéreo e fluxo óptico e desta forma V não era uma métrica. Assim, era mandatório

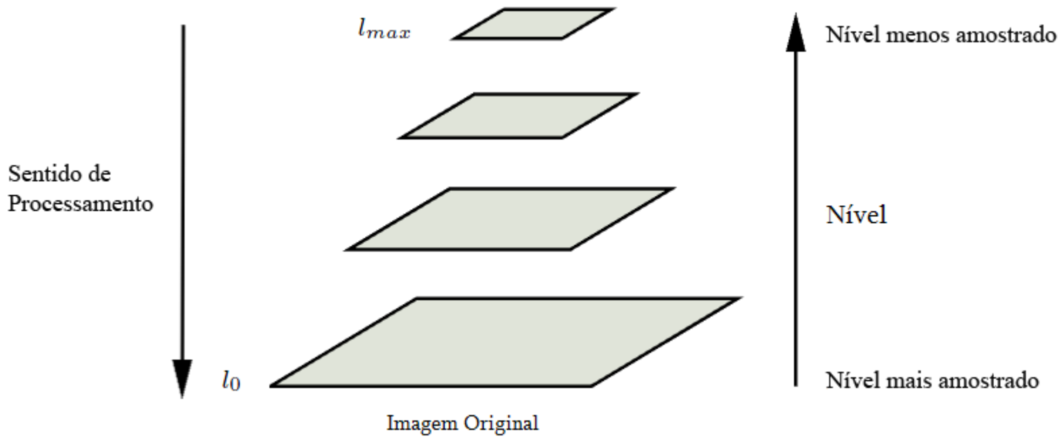

Figura 4.2: Exemplo de uma pirâmide de imagens. O nível 0 (l_0) representa a imagem original, na escala mais amostrada, enquanto l_{max} a escala menos amostrada.

utilizar o algoritmo de troca. No nosso caso, como não temos a pretensão de explorar fluxo óptico, podemos usar V como métrica e fazer uso do mais eficiente algoritmo de expansão.

No estágio *Upsampling e Propagação de disparidades* nós reamostramos o mapa de disparidades obtido em um nível com menor resolução (menos amostrado) para que ele sirva de inicialização para os próximos níveis da pirâmide.

Finalmente, no estágio de *Pós-Processamento* o algoritmo encerra, os resultados obtidos são armazenados e os parâmetros são reconfigurados.

4.1.1 Criação da Pirâmide de Imagens

O estágio *Pirâmide de Imagens* calcula pirâmides gaussianas de ambas as imagens de entrada, como mostrado na Figura 4.2 (para uma imagem). A imagem original está na parte de baixo da pirâmide, no nível l_0. Cada nível da pirâmide representa uma versão borrada e menos amostrada da imagem abaixo dela. Worby (18) utiliza um filtro gaussiano de duas dimensões, tirando vantagem da separabilidade para aumentar o desempenho e evitar o efeito de alisasing.

Estamos escolhendo um fator de amostragem 2. Vamos também utilizar um filtro gaussiano com raio mínimo de 2. Em todos os casos, escolhemos $\sigma = 2$, resultando em um filtro de raio 6. Segundo Worby, isso garante que não haverá aliasing na imagem, proporcionando o efeito de suavidade desejado.

Essa etapa de construção da pirâmide é muito rápida e não compromete o desempenho do método de maneira significativa.

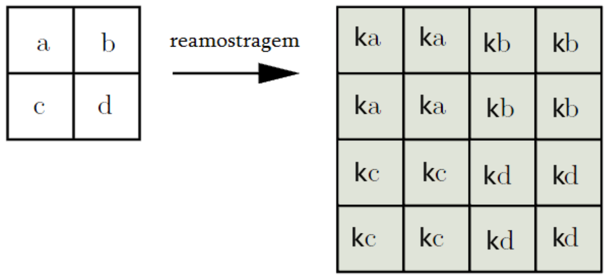

Figura 4.3: Reamostragem de um nível menos amostrado l_i para um nível mais amostrado l_{i-1}.

4.1.2
Reamostragem e Propagação de Disparidades

Para que o algoritmo de multi-resolução funcione corretamente os mapas de disparidade devem ser propagados entre os níveis da pirâmide. O mapa de um nível inicializa o próximo nível da pirâmide, dando assim um bom ponto de partida para o algoritmo no novo nível. Ao passar de um nível menos amostrado (com menor resolução) para um nível mais amostrado (com maior resolução) na pirâmide, para cada pixel p o algoritmo de reamostragem cria $N \times N$ pixels, onde N representa o fator de escala utilizado, como mostrado na Figura 4.3. Os valores de disparidade para os pixels com disparidade conhecidas numa escala menor são então multiplicados pelo fator de amostragem utilizado, ou seja, pixels com rótulos conhecidos no nível l_i são multiplicados pelo fator de amostragem (no nosso caso, $k = 2$) para determinar os rótulos no nível l_{i-1} na vizinhança daquele pixel.

4.2
Redução do Espaço de Disparidades

Um dos grandes desafios enfrentados pela maioria dos algoritmos de estéreo é estabelecer o equilíbrio entre tempo computacional e acurácia. Uma maneira de melhorar a acurácia é impor condições mais rígidas ao algoritmo e aumentar o número de disparidades possíveis. Entretanto num contexto de cortes de grafo essa abordagem pode trazer consequências indesejáveis para o tempo computacional. Por outro lado podemos também reduzir o número de rótulos utilizados ou o número de pixels para reduzir o tempo computacional mas isso reduziria também a acurácia do método.

Nesta seção vamos apresentar uma proposta de multi-resolução para o cálculo de correspondências em estéreo que foi inicialmente proposta por Worby (19). Os benefícios de um algoritmo de multi-resolução são:

– redução no número de pixels manipulados no cálculo dos níveis intermediários: as dimensões da imagem são reduzidas por um fator de amostragem k em cada nível da pirâmide;

– redução no número de rótulos manipulados: cada dimensão do espaço de disparidades é reduzido por um fator de amostragem k em cada nível da pirâmide.

Como cada nível tem um número menor de rótulos e pixels e parte de uma condição inicial produzida pela etapa anterior a convergência em cada nível é muito mais rápida. Dessa forma, os mapas de disparidade encontrados em cada nível são encontrados mais rapidamente e funcionam como inicialização para a configuração inicial de rótulos do próximo nível, fazendo que o método inicie no novo nível com uma estimativa para alimentar a otimização, que espera-se estar próximo do resultado final. Quando o nível mais amostrado da pirâmide é alcançado a configuração está mais próxima do mínimo global geral do que a inicialização normal dos métodos de cortes de grafo iniciadas sem nenhuma estimativa de rotulaçao. Assim a convergência é alcançada mais rapidamente.

Podemos descrever a forma geral de um algoritmo de multi-resolução é apresentada no Algoritmo 1, chamado de algoritmo de Level Seeding (LS). $Dmap_i$ representa o mapa de disparidades obtido no nível i e $Dmap_{final}$ se refere à solução final obtida no nível com mais amostras.

Algoritmo 1: Forma geral dos métodos de multi-resolução.

```
1  Criar pirâmide gaussiana das imagens
2  Determinar espaço de disparidades para cada nível da pirâmide
3  for i ← numLevels − 1 to 0 do
4      if i = (numLevels − 1) then
5          Dmap_i = algoritmo de cortes de grafo normal
6      else
7          Aumente a amostragem do mapa de disparidades
8          Dmap_i = EMA(Dmap_{i+1})
9      end
10 end
11 Dmap_final = Dmap_0
```

Para determinar o espaço de disparidades para cada nível Worby (19) propõe que o número máximo de disparidades permitidas em cada nível seja dividido pelo fator de amostragem. Por exemplo, se o espaço de disparidades para o nível mais amostrado (nível 0) é [16,16] ele passa a ser [8,8], [4,4] e [2,2] para os níveis 1, 2 e 3, respectivamente. O algoritmo de minimização de energia (EMA na linha 8) é o único passo que varia para os algoritmos que serão descritos nessa seção.

Vamos agora descrever os três algoritmos propostos por Worby que usam o framework de multi-resolução, os quais visam reduzir o número de rótulos usados durante a etapa *EMA* (linha 8 do Algoritmo 1). Como já foi mencionado, ao propor estes algoritmos Worby se concentrou em criá-los utilizando o algoritmo de troca-$\alpha\beta$, já que lhe era condição necessária por estar usando uma função de energia não classificável como métrica para fluxo óptico. Como o propósito do nosso trabalho é utilizar o algoritmo mais eficiente e preciso de expansão-α e não estamos interessados em considerar fluxo óptico, temos que alterar os algoritmos propostos para trabalherem com este novo algoritmo. As contribuições propostas serão descritas em detalhes nas seções 4.2.1, 4.2.2 e 4.2.3.

4.2.1
LDNR (Label Disparity Neighbourhood Restricted)

O principal problema do algoritmo LS descrito anteriormente é que, apesar do algoritmo ter uma inicialização de níveis bem eficiente, o conjunto de rótulos manipulados pelo algoritmo é muito grande, impedindo que haja uma redução mais significativa do tempo computacional. Para resolver este problema o algoritmo *Label Disparity Neighbourhood Restricted (LDNR)* usa a noção de vizinhanças de rótulos para restringir o tamanho do espaço de disparidades (18).

A um rótulo só lhe é permitido expandir para um rótulo em sua vizinhança, ou seja, para cada rótulo L presente no mapa reamostrado definimos uma vizinhança a uma distância de $\pm\delta$ ao redor de L. As vizinhanças são determinadas no início, de maneira que os valores de disparidade possíveis sejam fixos em todas as iterações. A Figura 4.4 mostra um exemplo em que $\delta = 1$. Nesse caso a idéia é que os pixels com rótulo z_1 só possam expandir para rótulos que estejam dentro de sua vizinhança, $L_N^{z_1}$.

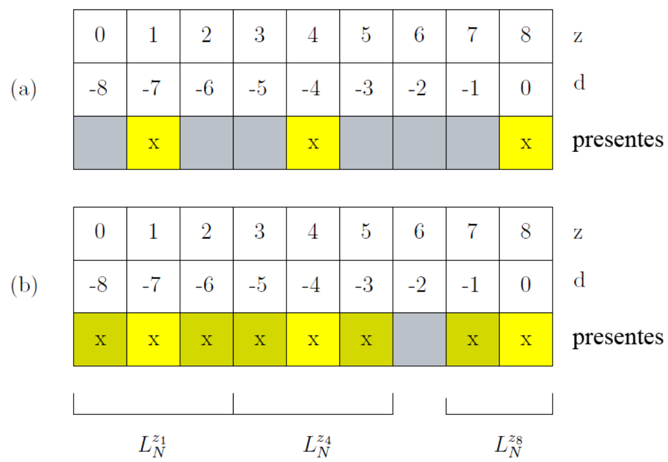

Figura 4.4: Em (b) a vizinhança de rótulos de (a). Os rótulos em questão (*presentes*) estão em amarelo. (Reproduzido de (19))

A justificativa para restringir a vizinhança de rótulos é que os pixels, em cada iteração, já estão bem próximos do seu rótulo ideal, e desta forma, só precisam expandir um pouco mais. Assim realizamos as expansões num conjunto reduzido de rótulos.

A implementação deste método é apresentada no Algoritmo 2, que é simplesmente o passo *EMA* (linha 8) do Algoritmo 1. A entrada deste algoritmo é um mapa de disparidades $Dmap_{up}$ reamostrado a partir do nível anterior da pirâmide. A condição de convergência testa se a energia da iteração atual é menor que a energia da iteração anterior. Se a função não convergir o algoritmo começa determinando todos os rótulos presentes em $L_{present}$ no mapa de disparidades reamostrado a partir do nível anterior (linha 1). Esse conjunto de rótulos não é alterado no resto do algoritmo. Após isso ele realiza uma iteração de expansão. Uma iteração neste algoritmo itera pelos rótulos em $L_{present}$. Para cada rótulo l_a ele encontra a vizinhança de rótulos $L_N^{l_a}$ e realiza expansões-α para cada rótulo l_b em $L_N^{l_a}$.

Algoritmo 2: Algoritmo de minimização de energia LDNR.

1 Determinar o conjunto de rótulos $L_{present}$ em $Dmap_{up}$
2 **while** $E_{cur} < E_{prev}$ **do**
3     $E_{cur} = E_{prev}$
4   **for** $\forall l_a \in L_{present}$ **do**
5   Determinar a vizinhança de rótulos L_N de l_a
6   **for** $\forall l_b \in L_N$ **do**
7             $Dmap_{cur} =$ expansão(l_b)
8   **end**
9   Calcular E_{cur} de $Dmap_{cur}$
10   **end**
11 **end**
12 $Dmap_{final} = Dmap_{cur}$

4.2.2
EL (Expanding Label Disparity Neighbourhood at Every Iteration)

O algoritmo LDNR possui como característica importante uma vantagem em relação ao método original que é a redução do número de rótulos manipulados pelo algoritmo. Possui, entretanto, uma desvantagem que é a propagação de erros (valores de disparidade inválidos) entre os níveis da pirâmide, o que diminui a acurácia do método, dada a grande redução que é realizada. Isso acontece devido à combinação das etapas de reamostragem e da restrição dos rótulos só poderem expandir para outros rótulos dentro da sua vizinhança.

Nesse âmbito o algoritmo *Expanding Label Disparity Neighbourhood at Every Iteration (EL)* se propõe a resolver o problema da redução excessiva de rótulos. A ideia é basicamente permitir que o conjunto de rótulos cresça dinamicamente em cada iteração do algoritmo (18).

O Algoritmo 3 ilustra esse procedimento. No Algoritmo 2 a determinação do conjunto de rótulos era feita no início do algoritmo e permanecia estática durante toda a execução, fazendo com que os rótulos só pudessem ser expandidos dentro dessa vizinhança pré-determinada. Agora a determinação desse conjunto é feita a cada iteração, dentro do loop mais externo. Dessa forma o conjunto de rótulos pode crescer aos poucos. As expansões ainda são feitas com os rótulos na vizinhança dos rótulos em $L_{present}$, mas nesta abordagem os valores de disparidade são aos poucos propagados até seu valor ideal.

Algoritmo 3: Algoritmo de minimização de energia EL.

$Dmap_{cur} = Dmap_{up}$
while $E_{cur} < E_{prev}$ **do**
  $E_{cur} = E_{prev}$
  Determinar o conjunto de rótulos $L_{present}$ em $Dmap_{cur}$
  for $\forall l_a \in L_{present}$ **do**
    Determinar a vizinhança de rótulos L_N de l_a
    for $\forall l_b \in L_N$ **do**
      $Dmap_{cur}$ = expansão(l_b)
    end
    Calcular E_{cur} de $Dmap_{cur}$
  end
end
$Dmap_{final} = Dmap_{cur}$

Esse aumento progressivo do conjunto de rótulos permite melhorar a acurácia do algoritmo tendo como contra partida o fato de que o tempo computacional também aumenta.

4.2.3 EAC (Expand All Combinations)

Uma outra maneira de aumentar a acurácia do algoritmo é permitir a expansão em mais rótulos. No algoritmo LDNR as expansões-α só podiam acontecer em rótulos dentro de uma vizinhança de disparidade. Para remover essa restrição propomos um novo algoritmo chamado *Expand All Labels(EAC)*[1]. O Algoritmo 4 mostra os passos realizados pelo EAC.

O algoritmo EAC, de maneira semelhante à que é feita pelo LDNR, determina os rótulos presentes em $L_{present}$ no mapa de disparidade reamostrado

[1] O método original proposto por Worby chama-se SAC (*Swap All Combinations*), já que o algoritmo utilizado por ele é o algoritmo de troca-$\alpha\beta$ (*swap*). Como não usamos o algoritmo de troca mas sim o de expansão, alteramos o nome do método para que correspondesse de igual maneira ao que ele faz.

Algoritmo 4: Algoritmo de minimização de energia EAC.

1 Determinar o conjunto de rótulos $L_{present}$ em $Dmap_{up}$
2 **for** $\forall l_i \in L_{present}$ **do**
3   Determinar a vizinhança de rótulos L_N de l_i
4   Adicionar L_N ao conjunto de rótulos L_{todo}
5 **end**
6 **while** $E_{cur} < E_{prev}$ **do**
7   **for** $\forall l_a \in L_{todo}$ **do**
8   expansão(l_a)
9   **end**
10 **end**
11 $Dmap_{final}$ = configuração atual

(linha 2) sem crescer durante o algoritmo. O próximo passo envolve a determinação das vizinhanças de disparidades para cada rótulo presente em $L_{present}$. Após os rótulos de cada vizinhança serem determinados eles são adicionados ao conjunto de rótulos L_{todo}. Em seguida, para cada rótulo presente no conjunto L_{todo}, o algoritmo calcula as expansões-α, permitindo que sejam feitas também de igual maneira para todos os rótulos neste mesmo conjunto.

A desvantagem deste método é que o conjunto de rótulos pode, em algum momento, se tornar o conjunto de rótulos completo. No método proposto por Worby há um grande aumento da acurácia por causa do aumento na quantidade trocas que eram realizadas. Por outro lado, como consequência, também há um aumento considerável no tempo computacional. Como na proposta desenvolvida estamos utilizando o algoritmo de expansão esse tempo é reduzido por um fator de n (onde n é o número de rótulos), implicando em um ganho significativo no desempenho.