

5 Resultados

Neste capítulo vamos apresentar os resultados obtidos com o método proposto neste trabalho. Para os testes foram utilizados diferentes pares de imagens presentes no conjunto de dados disponível no site do Middlebury (01), de maneira que pudéssemos realizar uma comparação coerente com outros métodos publicados que também utilizam esse conjunto padrão de testes.

Os métodos que foram comparados foram o utilizado por Kolmogorov em (09), cujos resultados são mostrados na Figura 5.1, o método proposto por Worby em (19), cujos melhores resultados são apresentados na Figura 5.2, e o nosso método, cujos mapas de disparidades obtidos para os pares de imagens Tsukuba, Teddy, Venus e Cones usando o método de multi-resolução LDNR são apresentados na Figura 5.3. Os resultados visuais obtidos utilizando os outros métodos são praticamente iguais, como analisaremos mais à frente e, assim, apresentamos somente as imagens obtidas com o método LDNR.

Na primeira coluna são exibidas as imagens de referência esquerda de cada par considerado, na segunda coluna o mapa de disparidades obtido utilizando o nosso método, na terceira coluna os mapas de disparidades “*groundtruth*” e na quarta coluna são exibidos os mapas de erro obtidos pelo avaliador do Middlebury após os mapas obtidos com nosso método serem submetidos a ele.

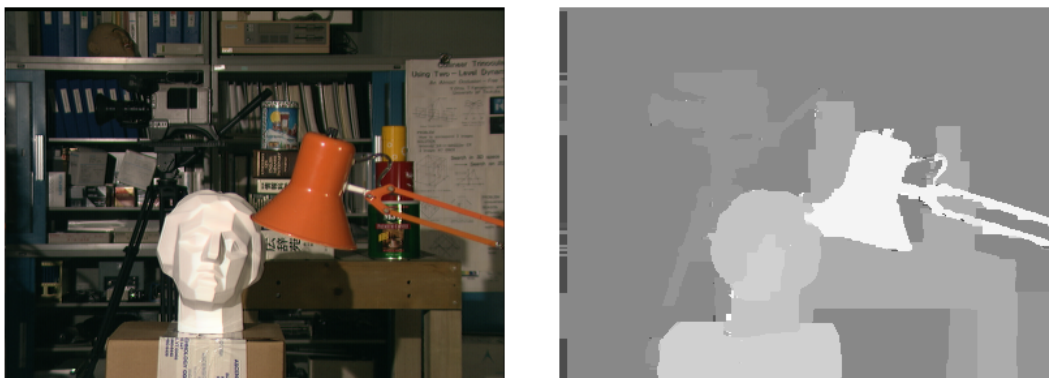


Figura 5.1: Resultados do método desenvolvido por Kolmogorov (09).

A aplicação foi testada em ambiente Windows XP SP3, Dual Quad Core 2.5 GHz, 3GB de RAM. Os rótulos (disparidades) α usados na expansão são

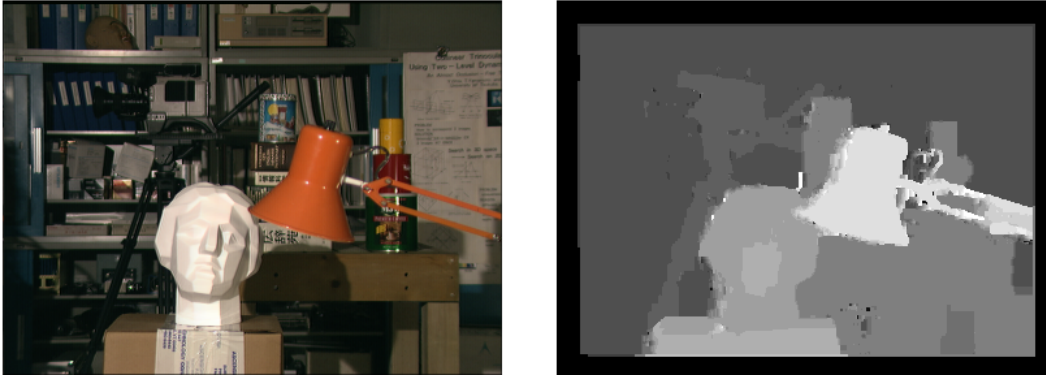


Figura 5.2: Resultados obtidos com o método desenvolvido por Worby (19).

escolhidos em ordem crescente no espaço de disparidades calculado em cada etapa de cada algoritmo. A construção da pirâmide foi feita utilizando filtros gaussianos com $\sigma = 2$ e as imagens são redimensionadas pela metade conforme subimos na pirâmide, ou seja, o fator de amostragem utilizado foi 2. O número máximo de níveis N da pirâmide gaussiana é calculado baseado no tamanho do espaço de disparidades considerado d como sendo $N = \log_2 d$. O espaço de disparidades considerado para cada nível da pirâmide é obtido dividindo-se o espaço de disparidades do nível anterior pela taxa de amostragem. As vizinhanças de cada valor de disparidade d são calculadas numa vizinhança $d + \delta$, onde $\delta = \pm 1$.

Para a realização dos testes utilizamos diferentes espaços de disparidades, a saber, 16, 32, 64 e 128. Para cada um dos três métodos comparados e para cada valor do espaço de disparidades foram realizadas cinco execuções do programa de maneira a se obter o desempenho médio de cada um dos três métodos.

A tabela 5.1 mostra o tempo de execução médio observado para o método desenvolvido por Kolmogorov no sistema considerado para calcular o mapa de disparidades de um par de imagens retificadas.

No nosso método implementamos um algoritmo de corte de grafos com multi-resolução que utiliza a proposta de Worby para a criação de pirâmides gaussianas para as imagens de entrada, onde o método de corte de grafos é executado em cada nível da pirâmide de imagens geradas e o resultado de cada

Tabela 5.1: Resultados obtidos com o método utilizado por Kolmogorov

Espaço de Disparidades	Tempo Médio de Execução
16	13 s
32	29 s
64	56 s
128	120 s

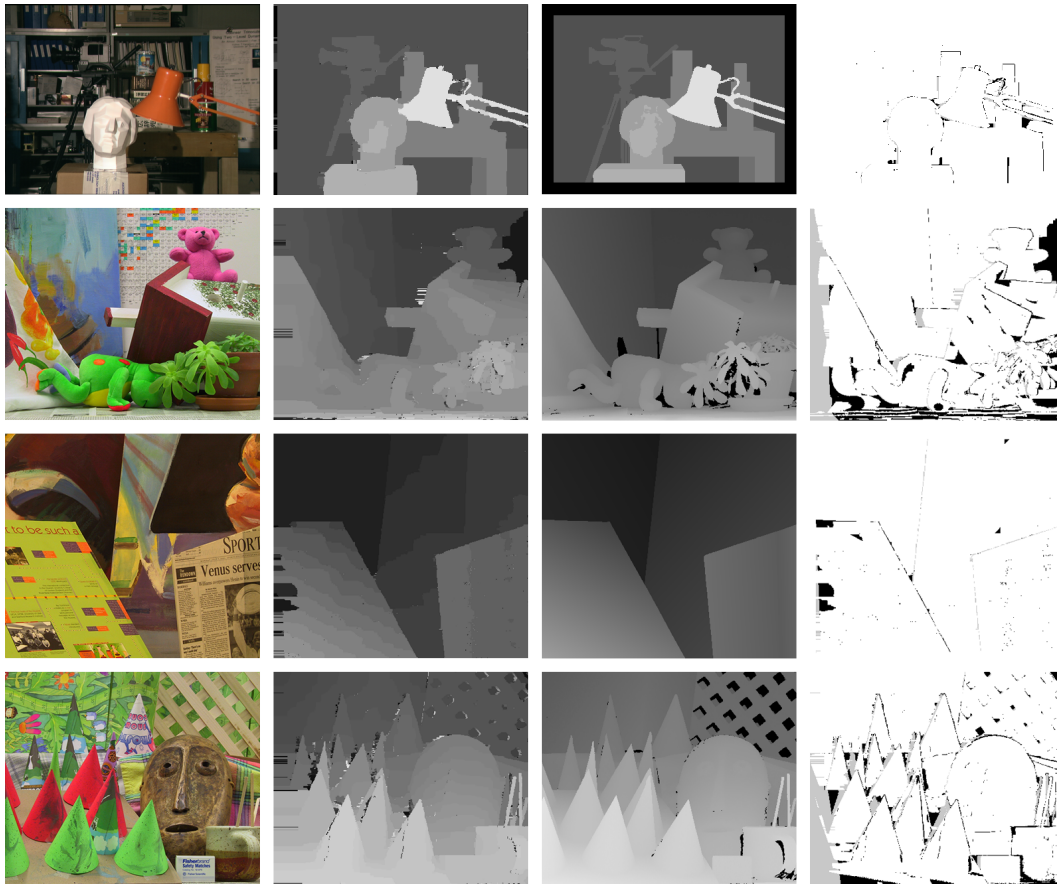


Figura 5.3: Resultados obtidos com nosso método usando o algoritmo LDNR.

nível inicializa o passo seguinte, até que alcancemos o mapa de disparidades final (no nível mais refinado). O algoritmo de expansão- α utilizado é o mesmo proposto por Kolmogorov em (09).

No primeiro passo simplesmente implementamos a criação das pirâmides, a definição do espaço de disparidades para cada nível da pirâmide e a execução do algoritmo de cortes de grafo utilizando o resultado de um nível anterior menos refinado. Os resultados obtidos com essa implementação são apresentados na tabela 5.2, onde mostramos o tempo médio de execução para cada espaço de disparidades considerado e o ganho de desempenho obtido em relação ao método proposto por Kolmogorov.

Em seguida implementamos os algoritmos propostos por Worby para a redução do espaço de disparidades em cada nível. Worby propõe em (19) três métodos para redução do espaço de disparidades utilizando o algoritmo de troca- $\alpha\beta$. Na nossa abordagem implementamos estes métodos utilizando o algoritmo de expansão- α apresentado por Kolmogorov.

A tabela 5.3 mostra os resultados obtidos utilizando a implementação do método LDNR para reduzir o espaço de disparidades utilizado pelo algoritmo

Tabela 5.2: Resultados obtidos com o nosso método

Disparidades	Tempo de Execução	Ganho de Desempenho
16	11 s	15%
32	24 s	17%
64	47 s	16%
128	87 s	28%

em cada etapa. Nela são apresentados o tempo de execução para cada espaço de disparidade considerado e o ganho de desempenho em relação ao método original proposto por Kolmogorov.

Analisando os resultados, observa-se que para um pequeno espaço de disparidades (16) o algoritmo apresenta desempenho bem pouco vantajoso comparado ao algoritmo original, dado o *overhead* adicional de determinar e utilizar as vizinhanças de cada valor de disparidade. Nesse caso, na última etapa, em geral temos um conjunto de valores possíveis de disparidade que é idêntico ao que teríamos se não estivéssemos utilizando o método. Assim, não removemos trabalho do método mas simplesmente acrescentamos, justificando o pouco ganho obtido.

Entretanto, quando aumentamos o espaço de disparidades o algoritmo mostra-se muito eficiente e com desempenho muito melhor quando comparado ao original. Esse comportamento é coerente já que o método consiste justamente em usar somente as disparidades em vizinhanças de disparidades já conhecidas. Dessa forma, quanto maior o espaço de disparidades mais valores de disparidade serão descartados e o desempenho do algoritmo será maior que o do algoritmo original.

Worby afirma que o método LDNR é o que tem o melhor desempenho dentre os três propostos mas sacrifica muito a acurácia como consequência. Os métodos implementados por Worby utilizam o algoritmo de troca- $\alpha\beta$. Como visto na seção 3.2 o algoritmo de troca em geral é utilizado para tratar funções de energia mais genéricas enquanto o de expansão exige que V seja uma métrica mas garante maior aproximação dos resultados obtidos, proporcionando melhores resultados em termos de acurácia.

No nosso caso, o método não sacrifica a acurácia, já que, em contraste

Tabela 5.3: Resultados obtidos utilizando o método LDNR

Disparidades	Tempo de Execução	Ganho de Desempenho
16	12 s	8%
32	20 s	31%
64	34 s	39%
128	32 s	73%

à implementação realizada por Worby, utiliza o algoritmo de expansão- α . Dessa forma as perdas de acurácia sofridas pela utilização de um método mais imediato como o LDNR são compensadas pela precisão do algoritmo de expansão que é utilizado.

Para o segundo método proposto (EL) Worby também utilizou o algoritmo de troca- $\alpha\beta$. Em nossa implementação utilizamos esse método também mas fazendo uso do algoritmo de expansão- α . A tabela 5.4 mostra os resultados obtidos utilizando a implementação do método EL para reduzir o espaço de disparidades utilizado pelo algoritmo em cada etapa.

Tabela 5.4: Resultados obtidos utilizando o método EL

Disparidades	Tempo de Execução	Ganho de Desempenho
16	12 s	8%
32	25 s	14%
64	49 s	13%
128	94 s	22%

A principal característica desse método é que o conjunto de nós presentes é recalculado a cada iteração do algoritmo, em contraste ao método LDNR. Esse conjunto é calculado a partir do último mapa de disparidades obtido, ou seja, no mapa no qual foi aplicado a última expansão. A tendência é que o conjunto total de disparidades presentes cresça mais rapidamente comparado ao LDNR, o que se confirma pelos resultados apresentados na tabela 5.4.

Finalmente implementamos o método EAC cujo princípio básico consiste em permitir que o conjunto de nós a serem expandidos seja o mesmo conjunto de nós vizinhos calculados a partir do mapa inicial. A ideia é manter a abordagem do método LDNR de calcular o conjunto de rótulos presentes no mapa reamostrado sem que esse conjunto cresça durante o algoritmo. Após esse processamento as vizinhanças são determinadas e adicionadas a um novo conjunto contendo esses rótulos. A cada iteração serão possíveis expansões de qualquer nó presente nesse novo conjunto para qualquer outro nó presente nesse mesmo conjunto.

Os resultados obtidos em termos de acurácia permaneceram idênticos aos obtidos com os outros dois métodos anteriores. Em termos de desempenho, como poderíamos esperar, temos um método mais rápido que o EL e menos eficiente que o LDNR. Os resultados são apresentados na tabela 5.5 onde o tempo de execução para cada espaço de disparidades considerado é exibido bem como o ganho de desempenho em relação ao algoritmo proposto por Kolmogorov.

Com o propósito de saber precisamente os resultados em termos de acurácia obtidos com o nosso método implementamos um conjunto de testes

Tabela 5.5: Resultados obtidos utilizando o método EAC

Disparidades	Tempo de Execução	Ganho de Desempenho
16	12 s	8%
32	22 s	24%
64	40 s	29%
128	55 s	54%

para comparar nossos resultados com o *groundtruth* de cada par de imagens disponíveis no Middlebury (01). Foram comparados três métodos: o proposto por Kolmogorov (09) utilizando o algoritmo de expansão, o algoritmo implementado por Boykov (04) também utilizando o algoritmo de expansão, e o nosso método.

A tabela 5.6 mostra os resultados obtidos para o par de imagens Tsukuba comparado ao seu respectivo *groundtruth* (primeira linha da Figura 5.3). Foi utilizado um espaço de 16 disparidades, já que é o espaço presente no *groundtruth* disponível. Apresentamos a porcentagem de pixels onde o algoritmo não calculou a disparidade corretamente (pixels incorretos) e a porcentagem de pixels onde a disparidade calculada diferia apenas de ± 1 da disparidade correta. Os valores para os métodos de Kolmogorov e Boykov são os valores fornecidos pelos respectivos autores (09).

Tabela 5.6: Tabela comparativa da acurácia obtida em relação a outros métodos

Método	Pixels Incorretos	Pixels Próximos
Kolmogorov	6.7%	1.9%
Boykov	20.7%	13.6%
Nosso método (LDNR)	6.9%	2.4%

Os resultados apresentados na tabela acima para o nosso método foram obtidos com a utilização do método LDNR em nosso algoritmo. Como já mencionamos os resultados em termos de acurácia para os três métodos implementados foram praticamente idênticos. Quando utilizamos os métodos EL e EAC os valores calculados de pixels incorretos e pixels próximos são bastante semelhantes aos obtidos pelo LDNR, apresentando diferença de aproximadamente $\pm 0.01\%$.

Finalmente os resultados obtidos com nosso método foram avaliados utilizando o conjunto de dados do sistema disponível em (01) e os resultados obtidos são apresentados na Figura 5.4.

Algoritmo	Posição	Tsukuba			Venus			Teddy			Cones		
		<i>nonocc</i>	<i>all</i>	<i>disc</i>	<i>nonocc</i>	<i>all</i>	<i>disc</i>	<i>nonocc</i>	<i>all</i>	<i>disc</i>	<i>nonocc</i>	<i>all</i>	<i>disc</i>
...
OptimizedDP [70]	47.7	1.97	3.78	9.80	3.33	4.74	13.0	6.53	13.9	16.6	5.17	13.7	13.4
ConvexTV [46]	47.9	3.61	5.72	18.0	1.16	2.50	12.4	6.10	15.7	16.8	3.88	14.4	11.5
Nosso Método	48.6	1.40	2.23	7.30	1.31	1.90	6.38	11.1	17.4	20.3	6.10	13.2	13.6
GenModel [20]	50.1	2.57	4.74	13.0	1.72	3.08	16.9	6.68	15.0	19.2	4.64	14.9	11.4
RTCensus [50]	51.3	5.08	6.25	19.2	1.58	2.42	14.2	7.96	13.8	20.3	4.10	9.54	12.2
...

Figura 5.4: Avaliação do algoritmo utilizando o conjunto de dados em Middlebury (01).