



UNIVERSIDADE FEDERAL
DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO PROCESSAMENTO DIGITAL DE IMAGENS

RELATÓRIO EXTENSÃO DO PROJETO ORIGINAL

Nome: Lucas Mendes Massa

Matrícula: 18112211

Nome: Jhonnye Gabriel de Oliveira Farias

Matrícula: 18112203

RECAPITULANDO O TRABALHO ORIGINAL.

O trabalho teve como objetivo abordar a força que os métodos clássicos de processamento de imagem, quando bem estruturados, conseguem superar os algoritmos direcionados a dados, como redes neurais. O paper aborda um algoritmo simples e fácil para o preenchimento de mapas de profundidade, o qual roda na CPU e não necessita de treinamento. O algoritmo foi avaliado seu *benchmarking* na base de dados de preenchimento de profundidade KITTI, atingindo, na data de sua submissão, a primeira colocação.

O algoritmo proposto segue o seguinte pipeline:

- Inversão de profundidade: visa facilitar a separação entre os pixels válidos e inválidos do mapa de profundidade;
- Dilatação com kernel personalizado: aplica uma operação morfológica de dilatação através de um kernel definido pelo usuário;
- Fechamento de pequenos buracos: aplica uma operação morfológica de fechamento com um kernel de valores unitários 5x5.
- Preenchimento de pequenos buracos: aplica uma nova dilatação com um kernel de valores unitários 7x7 e atualiza apenas os valores dos pontos inválidos do mapa através de uma máscara;
- Extensão até o topo da imagem: para contabilizar objetos altos como árvores, postes e edifícios que se estendem acima do topo dos pontos LIDAR, o valor superior ao longo de cada coluna é extrapolados;
- Preenchimento de grandes buracos: cuida de buracos maiores no mapa de profundidade que não foram totalmente preenchidos nos passos anteriores. Aplica uma nova dilatação com um kernel de valores unitários 31x31 e atualiza apenas os valores dos pontos inválidos do mapa através de uma máscara;
- Borramento mediano e gaussiano: visa eliminar outliers que possam

ter ocorrido como resultado das etapas anteriores. Em ambas as operações é aplicado um kernel 5x5.

- Inversão de profundidade: visa transferir o mapa para o formato existente antes da primeira inversão.

TESTES PROPOSTOS

Levando em conta que o trabalho original não explicita nenhum teste a respeito do formato dos kernels utilizados, exceto para a etapa de dilatação inicial, nós aplicamos diferentes kernels para as etapas restantes do algoritmo original, visando obter resultados melhores. Além disso, também foi testado o uso de uma abordagem de preenchimento guiado em conjunto com o algoritmo original. Para as etapas morfológicas para o preenchimento de buracos menores testou-se o uso dos diferentes kernels propostos pelos autores para a etapa de dilatação inicial, sendo eles kernels cheios, em cruz ou diamante, de tamanhos 3x3, 5x5 ou 7x7. Já para as etapas borramento e morfológica para o preenchimento de buracos maiores foram testados kernels unitários de tamanhos diferentes.

No que concerne ao algoritmo guiado, o mesmo foi proposto no artigo *Intensity Guided Depth Upsampling by Residual Interpolation* de Konno *et al.* O mesmo faz uso do chamado “filtro guiado”, o qual é proposto em outro artigo intitulado *Guided Image Filtering*, de HE *et al.* De acordo com os autores, o filtro guiado é derivado de um modelo linear local e gera a saída considerando o conteúdo de uma imagem guia, a qual pode ser a própria imagem de entrada ou outra imagem diferente. Dessa forma, o kernel é uma função que recebe a imagem guia como entrada, no seguinte formato:

$$q_i = \sum_j W_{ij}(I) p_j,$$

onde q é a imagem de saída, W o kernel que está em função da imagem guia I e p a imagem de entrada. Para maiores detalhes a respeito do funcionamento do kernel, recomendamos a leitura do artigo original. Dessa forma, o algoritmo proposto por Konno *et al.* faz uso do filtro guiado através do seguinte pipeline:

- Aplica-se o filtro guiado tendo como entrada um mapa de profundidade esparso e como imagem guia uma imagem de intensidade da mesma cena. O resultado é considerado um mapa de profundidade estimado;
- Calcula-se os resíduos a partir de uma subtração entre o mapa estimado e o esparso dado como entrada;
- Aplica-se uma interpolação bicúbica aos resíduos;
- Obtém-se o mapa final a partir da adição dos resíduos interpolados com o mapa previamente estimado.

O algoritmo acima foi aplicado a diferentes trechos do algoritmo de preenchimento de mapa de profundidade do artigo base para o presente trabalho.

RESULTADOS

Fechamento de pequenos buracos

	MAE (m)	RMSE (m)	Execution time (ms)
Original	0.305	1.345	10.5
Diamond 3x3	0.316	1.395	11.4
Diamond 5x5	0.314	1.369	11.2
Diamond 7x7	0.319	1.357	11.5
Full 3x3	0.301	1.349	10.8
Full 7x7	0.314	1.364	11.2
Cross 3x3	0.316	1.395	11.6
Cross 5x5	0.311	1.360	11.1
Cross 7x7	0.317	1.348	11.1

Preenchimento de pequenos buracos

	MAE (m)	RMSE (m)	Execution time (ms)
Original	0.305	1.345	10.5
Diamond 3x3	0.318	1.446	10.7
Diamond 5x5	0.310	1.395	11.1
Diamond 7x7	0.307	1.366	11.3
Full 3x3	0.316	1.431	10.9
Full 5x5	0.308	1.371	10.9
Cross 3x3	0.318	1.446	11.0
Cross 5x5	0.310	1.395	10.9
Cross 7x7	0.307	1.366	10.8

Preenchimento de grandes buracos

	MAE (m)	RMSE (m)	Execution time (ms)
Original	0.305	1.345	10.5
Kernel 15x15	0.305	1.341	10.5
Kernel 19x19	0.305	1.338	10.3
Kernel 23x23	0.305	1.339	10.4
Kernel 27x27	0.305	1.342	10.5
Kernel 35x35	0.305	1.349	10.8
Kernel 39x39	0.306	1.354	10.7
Kernel 43x43	0.306	1.358	11.0
Kernel 47x47	0.306	1.362	11.0

Borramento mediano e gaussiano

	MAE (m)	RMSE (m)	Execution time (ms)
Original	0.305	1.345	10.5
Median 3x3	0.307	1.351	8.3
Gaussian 3x3	0.304	1.368	10.6
Gaussian 7x7	0.313	1.331	10.8

Abordagem de preenchimento guiado

Foi testada a inserção do algoritmo mencionado anteriormente em diferente etapas do algoritmo original. Em nenhuma delas ocorreram melhorias. Em sua grande maioria o desempenho foi muito similar ao original. Nas ocasiões restantes o desempenho foi degradado. Em todas as ocasiões o tempo de execução teve um aumento significativo, o que já era esperado, tendo em vista que foi adicionada uma nova rotina a pipeline.

Combinações

Fazendo combinações a partir dos melhores resultados das tabelas acima, foi

possível obter mais alguns resultados interessantes. Com uma configuração de kernel 19x19 para o preenchimento de grandes buracos e 7x7 para o kernel gaussiano foi possível atingir um RMSE de 1.326. No entanto, o MAE aumentou para 0.312. Acrescentando um kernel Full 3x3 a essa configuração na etapa de fechamento de pequenos buracos chegou-se a um RMSE de 1.328 com MAE de 0.309.

Conclusão

Dessa forma, pode-se concluir que a melhor configuração possível se dá pela modificação da configuração do artigo original para utilizar um kernel 19x19 na etapa de preenchimento de grandes buracos. Essa configuração apresentou melhorias no tempo de execução e no RMSE, mantendo o mesmo MAE. Outras configurações testadas não apresentaram melhorias ou só apresentaram melhorias em algumas métricas com pioras nas demais. Isso ressalta a qualidade do trabalho realizado pelos autores com relação ao algoritmo original, tendo em vista que não foi possível encontrar facilmente uma forma de trazer melhorias significativas aos resultados.