

Relatório do Projeto Demonstrativo 3: Múltiplas Vistas

Aluno: Jessé Barreto de Barros
Matrícula: 17/0067033
Sistemas Mecatrônicos - PPMec-UnB
Disciplina: Visão Computacional
Turma: A
Data: 07/05/2017

1. Objetivos

O principal objetivo deste projeto demonstrativo é a avaliação dos aspectos envolvidos no desenvolvimento e utilização de algoritmos para a extração de mapas de profundidade obtidos a partir de pares estéreo de imagens.

2. Introdução

2.1. Reconstrução Estéreo

A reconstrução estéreo possui como objetivo a reconstrução da informação tridimensional, perdida através da projeção dos pontos no espaço tridimensional para o espaço bidimensional das imagens, através da utilização de múltiplas imagens de vistas diferentes da mesma cena.

A visão estéreo simples utiliza apenas duas vistas e é baseada no sistema visual binocular humano. Nesse sistema, geralmente, um par de imagens tiradas de câmeras paralelas com uma distância horizontal entre si que é conhecida.

Para facilitar a obtenção da informação de distância dos pontos da imagem ao centro da lente das câmeras, ou seja, a profundidade dos objetos na cena, é necessário que, para isso, ambas as câmeras possuam seus parâmetros intrínsecos conhecidos e ambas as imagens estejam retificadas.

O principal princípio por trás da reconstrução estéreo é a da triangulação de um ponto na cena e aos as suas projeções nas duas imagens. Conforme Figura 1.

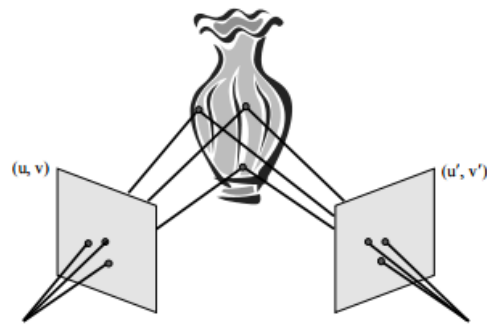


Figura 1. Triangulação de pontos correspondentes em uma cena utilizando duas imagens de vistas diferentes.

2.2. Correspondência Estéreo

O problema de correspondência é encontrar em um par de imagens os pontos que correspondem ao mesmo ponto na cena.

Utilizando a geometria epipolar para a projeção das imagens obtém-se que pontos correspondentes da cena estão sobre o mesmo plano epipolar e após a projeção esse plano se manifesta como um linha em ambas as imagens, a linha epipolar.

Esse processo de restrição de busca apenas as linhas epipolares tornam a busca de pontos correspondentes menos trabalhosa e utiliza menos recursos computacionais, Figura 2.

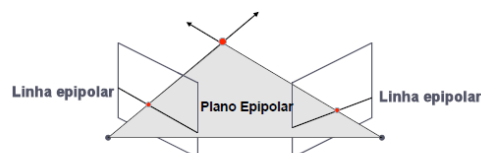


Figura 2. Representação do plano horizontal e a sua intersecção nos planos das imagens estéreo formando as linhas epipolares.

Para facilitar esse processo e torná-lo ainda mais efí-

ente utiliza-se imagens retificadas. Nessas imagens o plano da imagem é reprojetoado sobre um plano comum. Como resultado obtém-se que linhas epipolares correspondentes estão sobre a mesma reta, ou seja, o que deslocamento entre esses pontos correspondentes é horizontal. Figura 3.

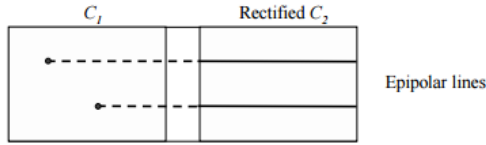


Figura 3. Linhas epipolares após o processo de retificação.

Mesmo após a simplificação na busca dos pontos correspondentes ainda é necessário utilizar um algoritmo que consiga identificar quais são os pontos correspondentes através de um parâmetro de correlação entre os pontos presentes nas duas imagens.

O algoritmo utilizado foi o uso da disparidade entre os pontos das imagens e a função de correlação (**matching**) foi a utilização da função SAD (*Sum of Absolute Difference*) [1] entre uma janela de pixels da imagem esquerda e outra janela de pixels da imagem direita.

Nesse caso os pontos correspondentes das imagens possuem o valor mínimo da função SAD. Além disso a função SAD se destaca por ser mais rápida e utilizar menos recursos computacionais do que outras funções custo, como por exemplo, o SSD (*Sum of Squared Differences*).

2.3. Profundidade a partir de imagens de um par de imagens estéreo

A disparidade entre os pontos correspondentes é a diferença da posição de um mesmo ponto da cena projetado nas duas imagens.

A disparidade é inversamente proporcional a profundidade real do ponto da cena. Para obtenção da profundidade temos que (1).

$$Z = \frac{Baseline \cdot focalDistance}{disparity} \quad (1)$$

3. Metodologia

O projeto possui o seguinte requisito:

- **Requisito 1:** Assumindo 2 pares de imagens capturadas por uma câmera estereo calibrada e retificada. Desenvolva um algoritmo que para cada par de pontos x,y na imagem esquerda encontre os pontos respectivos na imagem direita utilizando o conceito de *Template Matching* utilizando a função custo SAD. Para esses pontos obtenha os valores X, Y, Z e uma imagem normalizada entre 0 e 255 das informações de profundidade obtidas.

A aplicação implementada para o requisito 1 em C++ utilizando o *OpenCV* abre um par de imagens estéreo padrão (AloeL.png e ALoeR.png), que deve estar armazenado junto ao binário da aplicação, ou o caminho de ambas as imagens estéreo como parâmetros da execução da aplicação.

Ao inicializar a aplicação irá calcular o mapa de profundidade para uma janela de tamanho 5. No entanto, após o resultado haverá um *trackbar* para a seleção do tamanho da janela.

Nas Figuras 4 e 5 há um exemplo da execução da aplicação.

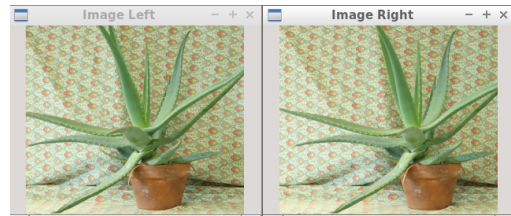


Figura 4. Após o cálculo do mapa de profundidade o algoritmo mostra as imagens estéreo em duas janelas separadas.

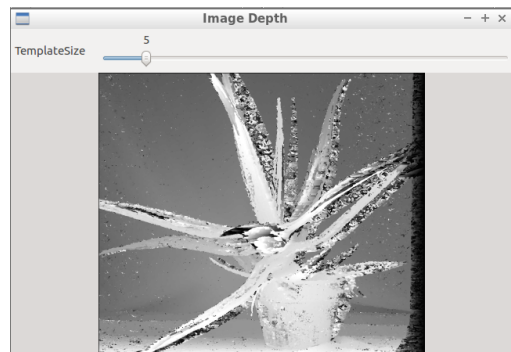


Figura 5. Após o cálculo do mapa de profundidade o algoritmo mostra o mapa de profundidade em uma janela com um *trackbar* para selecionar o tamanho da janela de template matching.

Após a seleção do tamanho da janela de template basta pressionar qualquer tecla para iniciar o cálculo do mapa de profundidade ou apertar esc para finalizar a aplicação. Além de mostrar o mapa de profundidade normalizado a aplicação também o salva como um arquivo *png*.

Primeiramente, o algoritmo converte ambas as imagens para a escala de cinza por que apenas a intensidade luminosa é necessária para a função SAD.

Para calcular o mapa de disparidade o algoritmo busca em um espaço de 128 colunas da imagem direita pelos pontos que minimizam a função SAD das janelas de *matching* que comparam as imagens da direita e da esquerda. Esses pontos então são salvos em uma matriz com os valores mínimos dos mapas de custo e também salva em uma matriz as disparidades para o cálculo da profundidade.

4. Resultados

Foram obtidos os seguintes resultados para as imagens estéreo presentes na Figura 4. Os resultados estão presentes nas Figuras 6, 7, 8 e 9.

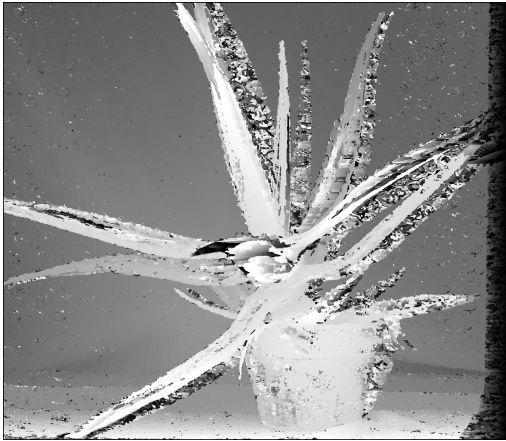


Figura 6. Mapa de profundidade para uma janela de matching de tamanho = 5.

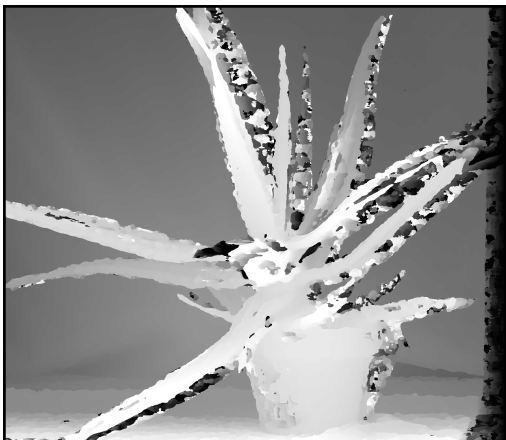


Figura 7. Mapa de profundidade para uma janela de matching de tamanho = 15.



Figura 8. Mapa de profundidade para uma janela de matching de tamanho = 31.



Figura 9. Mapa de profundidade para uma janela de matching de tamanho = 51.

5. Discussão e Conclusões

O projeto demonstrativo concluiu os objetivos propostos e com esse projeto foi possível aprender mais sobre o processo de reconstrução de profundidade utilizando imagens obtidas de um câmera estéreo.

Foi possível verificar as dificuldades inerentes de algoritmos de visão computacional e o fácil aumento de complexidade computacional e o custo de tempo que o mesmo possui conforme as imagens aumentam de resolução.

Com o intuito de cobrir todos os requisitos o algoritmo desenvolvido possui complexidade $O(N \cdot M \cdot D \cdot W^2)$ sendo: (a) N - o número de linhas da imagem; (b) M - o número de colunas na imagem; (c) D - o número de colunas percorridas para buscar as correspondências; (d) W - O tamanho da janela da função SAD. É possível verificar que a alta complexidade implica em uma aplicação que consome bastante tempo para parâmetros com tamanhos grandes.

Como o esperado percebe-se que a qualidade do mapa de profundidade obtido depende bastante do tamanho da ja-

nela utilizada durante a etapa de *template matching*. Valores pequenos da janela levam a um mapa mais "fino", mas que possui alta taxa de ruídos e para valores muito grandes da janela levam a um mapa que possui menos detalhes porém menos ruídos. No entanto, no resultado obtido parece haver um taxa alta de ruído também na imagem com janela maior. Isso se deve ao fato que a textura da imagem é muito semelhante em algumas regiões e esse baixo contraste leva a uma imagem mais ruidosa.

Para uma visualização mais fácil utilizou-se do inverso dos mapas de profundidade para as Figuras.

6. Bibliografia

Referências

- [1] Sad - sum of absolute differences. http://en.wikipedia.org/wiki/Sum_of_absolute_differences. acessado em 06/04/2016.