Projeto 2 de Princípios de Visão Computacional

Gabriel Martins de Miranda
130111350
Universidade de Brasília
Email:gabrielmirandat@hotmail.com

***Resumo*—A presente demonstração se baseia no algoritmo de M.Brown e D.Lowe, implemetados na classe invólucro image stitching. Dado um conjunto de imagens da torre de tv de brasília, tiradas apenas rotacionando a câmera num eixo fixo em 180 graus, criou-se um panorama robusto, invariante a ordem das imagens de entrada, orientação, escala e iluminação.**

I. Introdução

Um panorama caracteriza-se pelo concatenaccão de algumas/várias imagens de forma que dada esta união tem-se uma visão mais ampla do que com cada imagem em separado.

De uma maneira geral, para se criar uma imagem panorâmica simples, realizamos os seguintes passos:

1) Tirar sequência de fotos a partir da mesma posição, rotacionando a câmera sobre seu eixo óptico.
2) Computar a transformação T da segunda para a primeira imagem.
3) Moldar a segunda imagem para se sobrepor à primeira.
4) Repetir para todas as outras.

Acontece que com esta abordagem, não conseguimos obter um panorama tão robusto e invariante como o proposto neste projeto demonstrativo.

Uma abordagem mais convincente foi proposta e já vem implementada no módulo $stitching. Images\ stitching$, presente no software OpenCV. De maneira geral, segue os passos utilizados pelo módulo:

1) Encontrar features invariantes.
2) Eliminar features desnecessárias.
3) Encontrar relacionamento entre todas as imagens através do casamento de features comuns (vetores descritores).
4) Relizar casamentos para construção do panorama propriamente dito.
5) Retificar efeito ondulado do panorama.
6) Otimizar o resultado homogeneizando áreas próximas e eliminando o efeito de costura.

II. Metodologia

O primeiro passo no algoritmo do panorama é encontrar características que possam definir/diferenciar uma imagem de outra. Estas características, chamadas de $features$, geralmente são cantos retirados do cruzamento entre duas bordas.

Abordagens tradicionais para localização de cantos, tais como o de $Harris$, apesar de serem algoritmos robustos, caem em um dos problemas de invariância. No caso de $Harris$, mudanças na escala podem impedir a localização destas características.

Atraés do algoritmo $Scale\ Invariant\ Feature\ Transform$, $SIFT$, é possível obter $features$ invariantes. Estas estão localizadas na relação escala-espacial de diferença máxima/mínima da função Gaussiana. Em cada localização de $feature$, características de escala e orientação são estabelecidas. Isto nos dá um quadro de similaridade-invariância no qual podemos fazer medidas. O método então realiza o histograma das orientações dos gradientes locais. Este acumulo espacial é muito importante para a invariância do SHIFT.

Isto permite que as bordas se desloquem sem alterar os vetores descritores (aqueles que indicam o relacionanto entre features comuns nas diferentes imagens que compõem o panorama), dando grande robustez para as posteriores transformações afins.

A invariância à iluminação é adquirida pelo uso de gradientes e pela normalização dos vetores descritores.

Uma vez que as $features$ foram extraídas de todas as imagens, elas devem ser relacionadas umas com as outras. Cada $feature$ é casada com seus k vizinhos mais próximos em seu espaço de atuação, através do algoritmo $k - d\,tree$.

O segundo passo se trata do descarte de features desnecessárias e do casamento das imagens, que futuramente se tornarão um panorama. Do passo anterior, nós identificamos imagens que tem um grande número de $matches$ entre si. Consideramos um número fixo de m imagens que possuem os maiores números de casamentos de features com a imagem atual, como potenciais para a concatenação entre elas. Primeiro, usamos o método $RANdom\ SAmple\ Consensus$, $RANSAC$, para selecionar um conjunto de $inliers$, (ou features boas, contidas num intervalo de mínimo-máximo numa reta de transição) que são compatíveis com uma homografia entre as imagens. Os outliers (features ruins, ou que ficaram fora do intervalo) são descartados. Depois disto, aplicamos um método probabilístico para verificar o match.

O descarte proposto pelo $RANSAC$ é muito importante pois diminui muito a complexidade e custo de processamento, permitindo rápida estimação das matrizes de homografia entre as imagens que possuem correspondência através dos inliers encontrados. Desta maneira, as imagens são alinhadas e unidas de acordo com a homografia encontrada.

Os passos seguidos até então nos dá a rotação relativa entre as imagens, mas não permite saber como são no mundo 3D. Se simplesmente assumirmos que R (rotação) = I (identidade)

para uma das imagens, tipicamente nos deparamos com um efeito onludalado na saída do panorama. Podemos corrigir esta ondulação usando a heurística do modo como as pessoas geralmente tiram fotos panorâmicas. É raro as pessoas torcerem a câmera relativamente ao horizonte, então os vetores horizontais da câmera recaem num plano comum. Assim é possível corrigir este efeito aplicando uma rotação global derivada desta suposição.

Através dos parâmetros geométricos da câmera, (orientação e distância focal), podemos encontrar um parâmetro fotométrico, o ganho geral entre as imagens. Através deste ganho global e do uso do algoritmo de Burt e Adelson chamado $multi-band\ blending$, que suaviza as intensidades de pixels diferentes entre $features$ iguais (que foram correspondidas na saída do panorama), podemos realizar um acabamento e otimizar o resultado final do panorama.

Diante destes passos, pode-se simular uma situção fictícia de como o algoritmo de obtenção de panoramas funciona:

Assumindo que se quer unir 4 imagens, $I_0,\ I_1,\ I_2,\ I_3$, o objetivo é computar as matrizes de homografia $H_0,\ H_1,\ H_2,\ H_3$.

1) Computar todos os pares de homografias $H_{01},\ H_{02},\ H_{03},\ H_{12},\ H_{13},\ H_{23}$, onde homografia H_{01} molda a imagem I_0 na I_1, etc.
2) Selecionar uma imagem âncora, por exemplo I_1 cuja posição ficará fixa. Para isto, podemos escolher H_1 como sendo a matriz identidade.
3) Encontrar imagem que melhor se alinha com I_1 baseado no número máximo de $matches$ consistentes, por exemplo I_3.
4) Atualizar $H_3 = H_1 \text{x} H_{13}^{-1} = H_{13}^{-1} = H_{31}$
5) Encontrar imagem que melhor se encaixe com I_1 ou I_3, por exemplo I_2 em I_3.
6) Atualizar $H_2 = H_3 \text{x} H_{23}$
7) Idem para I_0
8) Fazer ajuste para otimizar o alinhamento global.

III. Resultados

1) Imagens tiradas de cima da torre de TV(câmera rotacionando só na horizontal):

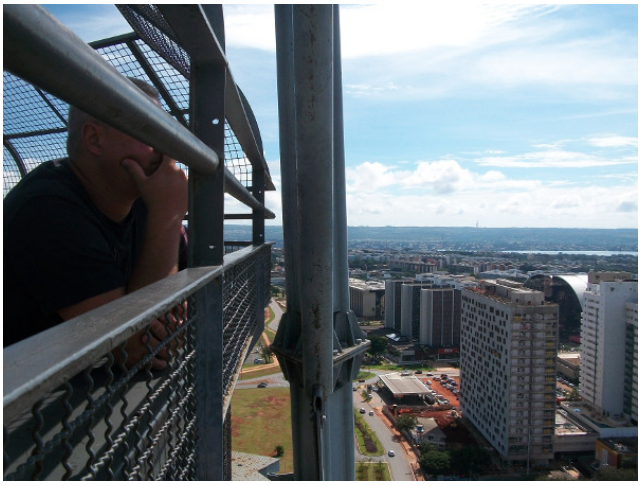

Fig. 1: imagem 1/6. Vista em cima da torre.

Fig. 2: imagem 2/6. Vista em cima da torre.

Fig. 3: imagem 3/6. Vista em cima da torre.

Fig. 4: imagem 4/6. Vista em cima da torre.

Fig. 5: imagem 5/6. Vista em cima da torre.

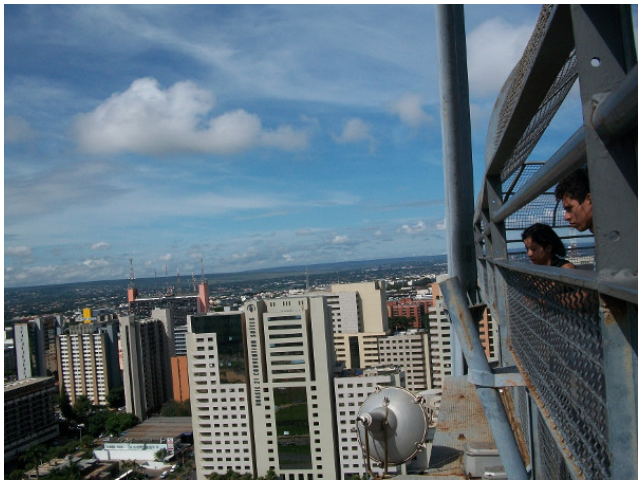

Fig. 6: imagem 6/6. Vista em cima da torre.

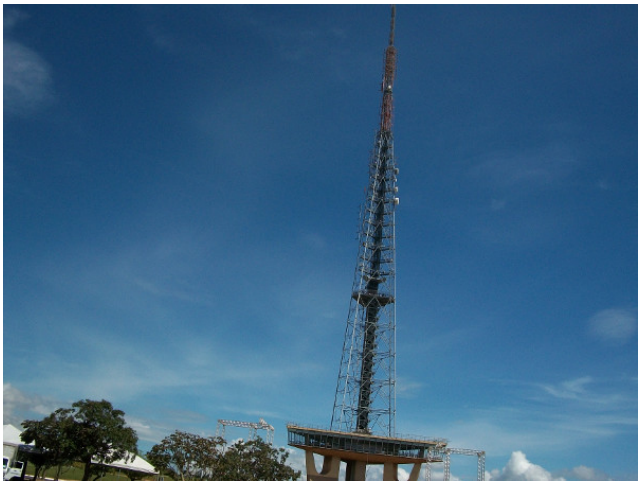

Fig. 7: imagem 1/14. Vista em baixo da torre.

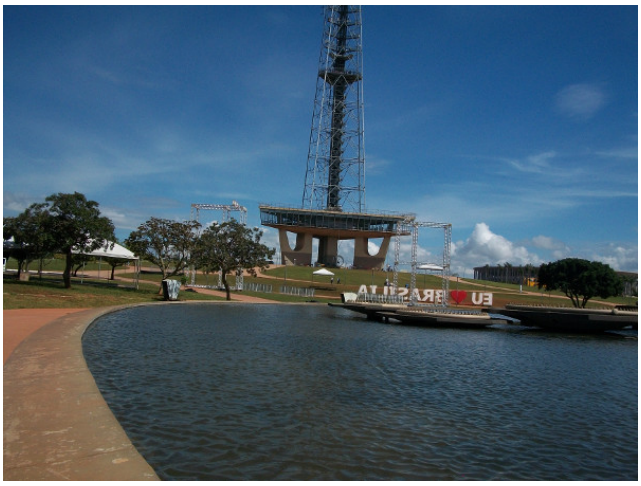

Fig. 8: imagem 2/14. Vista em baixo da torre.

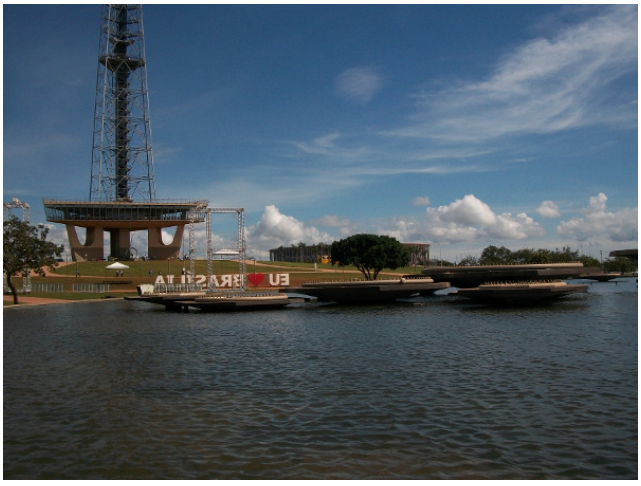

Fig. 9: imagem 4/14. Vista em baixo da torre.

2) Imagens tiradas de baixo da torre de TV(câmera rotacionando na horizontal e vertical):

Fig. 10: imagem 7/14. Vista em baixo da torre.

Fig. 11: imagem 10/14. Vista em baixo da torre.

Fig. 12: imagem 11/14. Vista em baixo da torre.

Fig. 13: imagem 12/14. Vista em baixo da torre.

Fig. 14: imagem 13/14. Vista em baixo da torre.

Fig. 15: imagem 14/14. Vista em baixo da torre.

Fig. 16: imagem resultado. Vista em cima da torre.

Fig. 17: imagem resultado. Vista em baixo da torre.

IV. Conclusão

Através do algoritmo aqui proposto, torna-se possível a criação de panoramas robustos e invariantes à questões referentes à escala, zoom, iluminação e orientação. Os resultados são surpreentes, graças as etapas de otimização, torna-se difícil supor que houve junção de imagens, já que não se pode ver as bordas das imagens que compõem o panorama.

V. Referências

[1] Automatic Panoramic Image Stitching using Invariant Features, Matthew Brown and David G. Lowe.

[2] Eric Yuan's Blog Perstando et Praestando RANSAC.

[3] ramsrigoutham A wayfarer's monologue! Panorama Image Stitching in OpenCV

[4] Project4: Image Warping and Mosaicing Danielle Millett

[5] OpenCV Stitching example (Stitcher class, Panorama) - MARE's Computer Vision Study