ALGORITMOS PARA CONSTRUÇÃO DE PANORAMA DE IMAGENS 360 E VISUALIZAÇÃO

Alan Kazuo Hiraga[1], Francisco Assis da Silva[2], Almir Olivette Artero[3]

[1]Discente, [2]Docente da Faculdade de Informática da Universidade do Oeste Paulista (UNOESTE), Presidente Prudente, SP; [3]Docente da Faculdade de Ciências e Tecnologia da UNESP, Presidente Prudente, SP.

RESUMO

Este trabalho apresenta uma metodologia de construção de mosaico de imagens para a formação de panorama 360 e um aplicativo para visualização. Foram utilizados os algoritmos SIFT e RANSAC comumente encontrados na literatura para a correspondência entre imagens. Os algoritmos de projeção, de ajustamento de imagens e Blend para suavização das junções foram implementados neste trabalho com o uso da biblioteca de Visão Computacional OpenCV. Na metodologia deste trabalho foram aplicadas técnicas para diminuir as distorções causadas nas junções sucessivas de imagens, assim como aumentar a qualidade do panorama final e obter um melhor desempenho. O aplicativo de visualização desenvolvido em C# mostra a imagem 360 resultante da junção de imagens em um cilindro com ponto de vista no seu interior. Os resultados dos experimentos realizados utilizando a técnica proposta tem se mostrado satisfatórios, o mosaico formado pela junção de várias imagens sofre pequenas distorções, no entanto, essas distorções não interferem na formação final do panorama 360, o que apresenta uma boa qualidade visual.

Palavras-chave: Panorama 360, retificação de imagens, ajustamento de imagens, SIFT, RANSAC.

ALGORITHMS FOR CONSTRUCTION OF 360 PANORAMIC IMAGES AND VISUALIZATION

ABSTRACT

This paper presents a methodology for the construction of mosaic images to form 360 panorama and an application for viewing. We used the SIFT and RANSAC algorithms commonly found in the literature for matching images. The algorithms for projection, adjustment of images and Blend for smoothing the joints were implemented in this paper using the OpenCV Computer Vision library. The methodology of this research techniques were applied to reduce the distortions caused in the joints of successive images, as well as increase the quality of the final panorama and get better performance. The viewer application developed in C# shows the 360 image resulting from images stitching into a cylinder with the point of view inside. The results of the experiments performed using the proposed technique has proved satisfactory, the mosaic formed by the joining of multiple pictures suffers little distortion, however, these distortions do not interfere with the final formation of the panorama 360, which provides good visual quality.

Keywords: 360 panoroma, images retification, images adjustment, SIFT, RANSAC.

1. INTRODUÇÃO

A utilização de imagens panorâmicas vem se tornando cada vez mais comum, podendo obter estas por meio de câmeras ou lentes especiais. Estas imagens são construídas principalmente para retratar um ambiente com maior ângulo de visão, podendo ser utilizadas, por exemplo, na construção de cenários de jogos, onde há a simulação da realidade, como em simuladores de vôo e de corrida. Imagens panorâmicas são obtidas desde 1839 (CURTIN, 2001), onde nesta época, eram capturadas várias fotos, estas eram reveladas em papel e depois eram recortadas e coladas na sequência que foram capturadas, assim surgiu o termo "*Stitching*" de imagens ou costura de imagens. Atualmente a construção de imagens panorâmicas é feita com o uso de softwares ou por meio do uso de lentes especiais.

Os softwares que fazem a junção de imagens são comumente denominados de software de "*Stitching*" ou de mosaico de imagem. As lentes que são capazes de obter imagens panorâmicas são chamadas de "*Fisheye*" ou olho de peixe (HILL, 2007). Elas possibilitam a construção desse tipo de imagem, pois elas possuem uma grande angular ou uma distância focal maior do que as lentes convencionais, possibilitando assim a captura de imagem do ambiente com uma maior abrangência. A imagem panorâmica deve retratar um ambiente com maior ângulo de visão que o ser humano pode enxergar. Por meio das imagens panorâmicas é possível construir uma imagem em que um ambiente é retratado, esta imagem é denominada panorama 360 (TSAN HU, 2006).

Panoramas 360 são utilizados para retratar um ambiente, empresas como a Google e a Microsoft retratam ruas e rodovias por meio deste recurso, assim o usuário pode "andar" pelas vias de uma cidade que desejar sem sair de casa, ou até mesmo, planejar suas viagens de férias conhecendo melhor os pontos turísticos, restaurantes e hotéis da cidade onde ele pretende conhecer. Outras empresas fazem o uso de panoramas 360 para apresentar melhor seus produtos e serviços, como por exemplo, sites Web de hotéis e pousadas que possibilitam aos clientes fazer um "tour" virtual nos ambientes do estabelecimento.

Visto a importância de imagens panorâmicas, este trabalho propõe a implementação de uma metodologia para realizar a junção de imagem para a criação de um panorama 360. Um visualizador também foi desenvolvido neste trabalho para permitir a interação com imagens panorâmicas, possibilitando a visualização em diferentes ângulos e pontos de vista. Neste trabalho são tratadas as técnicas e algoritmos que foram utilizadas para compor a metodologia desenvolvida, assim como os problemas, soluções e resultados obtidos.

As demais seções deste trabalho estão organizadas da seguinte maneira: na seção 2 é apresentado o problema inerente à junção de imagens e as distorções causadas nas imagens, assim como a retificação de imagens como solução para diminuir esses efeitos; na seção 3 é descrito suscintamente o algoritmo SIFT utilizado para a extração dos descritores de pontos chave e o algoritmo para correlação entre os pontos chave entre duas imagens; na seção 4 é apresentado o algoritmo RANSAC que elimina os pontos chave não condizentes entre os conjuntos de pontos chave de duas imagens e gera a matriz homográfica baseada nos pontos chave realmente correspondentes; na seção 5 são apresentados algoritmos de *Blend*, que são utilizados para suavizar as bordas das junções entre as imagens; a seção 6 mostra os problemas encontrados para a formação de panorama 360

devido ao desalinhamento das extremidades esquerda e direita da imagem panorâmica, bem como a solução empregada; na seção 7 é apresentada a construção do visualizador baseado no *PTViewer*; e por fim, na seção 8 são apresentadas as conclusões e considerações finais do trabalho.

2. RETIFICAÇÃO DE IMAGENS

A retificação de imagens é um estágio prévio da computação necessário para a construção de panoramas. Em Roberto et al. (2010) são discutidas duas formas de retificação de imagens, a retificação planar e a retificação cilíndrica. Ambas determinam que o par de imagens a ser retificado deva ser reorganizado a partir de uma re-projeção.

2.1. Geometria Epipolar

A geometria *epipolar* descreve as relações que existem entre duas imagens. Cada ponto em um plano que passa por dois centros de projeção serão projetados em cada imagem na intersecção deste plano com o plano da imagem correspondente. Portanto, essas duas linhas de intersecção contêm uma correspondência *epipolar* (POLLEFEYS, 1999).

Desse modo, existem relações importantes que ocorrem entre os pares de imagens. A primeira delas é a reta que liga o centro C_1 da primeira imagem com o centro C_2 da segunda, que é chamado de *baseline*. O ponto de intersecção desta reta com o plano de imagem é chamado de *epipolo*. Para a primeira imagem têm-se o *epipolo* e_1 e para a segunda imagem tem-se o *epipolo* e_2. Se existem dois pontos m_1 e m_2 na primeira e na segunda imagem, respectivamente, que são a projeção de um ponto M em coordenadas de mundo, pode-se dizer que M, C_1, C_2 são coplanares, formando o plano *epipolar* (ROBERTO et al., 2010). A Figura 1 ilustra a geometria *epipolar*.

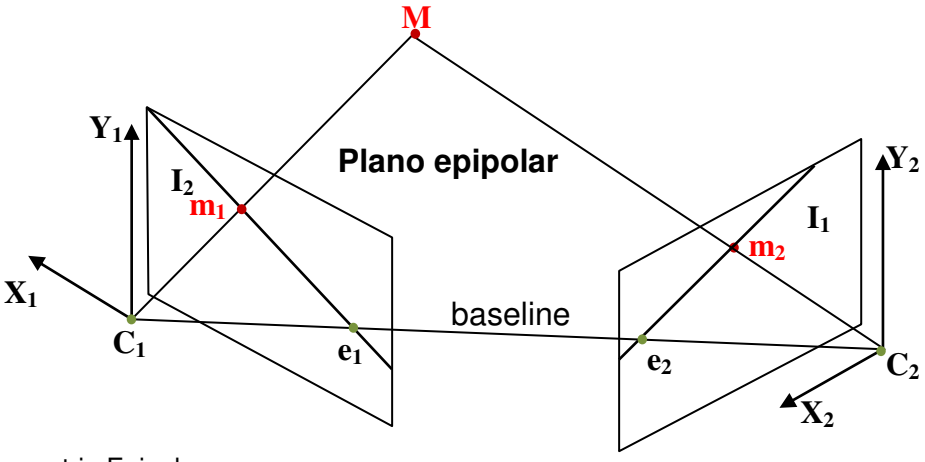


Figura 1. Geometria Epipolar.
Fonte: Tsan Hu (2006)

2.2. Retificação Planar

Utilizando-se do conceito de geometria *epipolar*, a retificação planar deve ajustar as imagens no *baseline* para assim uni-las, dessa forma garante-se que as linhas epipolares correlatas estarão na mesma altura, porém quanto mais próximo da imagem o *epipolo* estiver, maior será o tamanho da imagem retificada. Caso o *epipolo* se localizar dentro da imagem, a junção resultará em uma imagem que tende ao infinito (ROBERTO et al., 2010). A Figura 2 mostra um exemplo de junção de imagens utilizando a Retificação Planar evidenciando esse efeito de esticar as extremidades laterais.



Figura 2. Resultado de junção de imagens usando a Retificação Planar.

2.3. Retificação Cilíndrica

No entanto, em uma retificação cilíndrica ou projeção cilíndrica, o *epipolo* não se localiza dentro da própria imagem. Essa retificação consiste em determinar um cilindro de raio unitário que tem o *baseline* com eixo de revolução e, em seguida, mapear cada pixel da imagem em uma coordenada (v, θ) de um sistema de coordenadas cilíndricas, que pode ser usado normalmente, como um ponto (x, y) na imagem (ROBERTO et al., 2010).

A projeção cilíndrica se baseia em projetar cada pixel da imagem em um cilindro de raio Z de mesma altura da imagem (AGAPITO, 2001 apud TSAN HU, 2006).

Para calcular a posição dos pixels projetados no cilindro usa-se trigonometria. Considerando que o plano da imagem está em um espaço tridimensional, pode-se construir um cilindro encostado ao plano da imagem. Considerando a origem dos eixos no centro deste cilindro, e assim coincidente com o centro óptico da lente, o raio do cilindro (Z) será a distância focal da lente. Para a imagem de uma câmera, o eixo Z deverá coincidir com o eixo óptico.

É necessária uma conversão das coordenadas dos pontos. Normalmente considera-se a origem dos eixos da imagem em seu canto superior esquerdo. Para efetuar a projeção considera-se o centro da imagem, coordenada (0,0) no ponto principal do plano da imagem. Cada ponto da imagem terá uma coordenada no sistema global dada como (X, Y, Z). Esta posição projetada no cilindro terá a coordenada (v, θ), sendo que v é a altura do ponto de intersecção entre cilindro e a reta que liga a origem ao ponto (X, Y, Z). O ângulo θ é aquele formado entre o eixo Z e esta reta (SZELISKI; SHUM, 1997; SHUM, 2001 apud TSAN HU, 2006, p.26).

A Figura 3 mostra a representação de uma projeção cilíndrica em duas perspectivas.

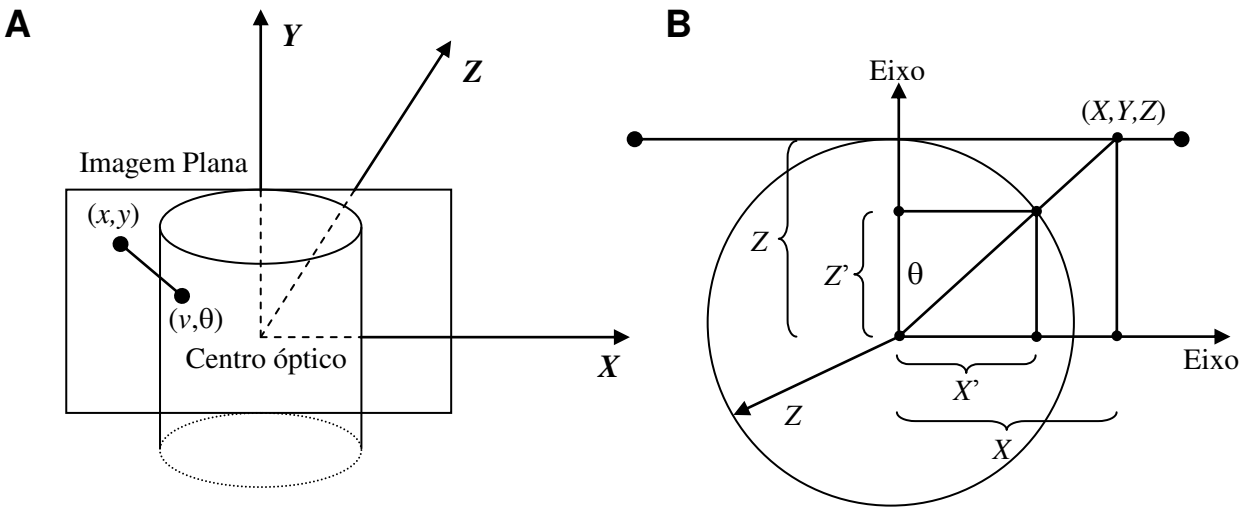


Figura 3. (a) Vista da imagem plana no cilindro. (b) Vista superior do cilindro.

Fonte: Tsan Hu (2006).

Para calcular o ângulo θ, tem-se:

$$\theta = tg^{-1}\left(\frac{x}{y}\right) \quad (1)$$

para calcular *v* tem-se que calcular antes *x'*:

$$x' = z.\sin(\theta) \quad (2)$$

a altura *v* é calculada pela relação de triângulos:

$$\frac{x}{v} + \frac{x}{x'} \Rightarrow v = \frac{y.x'}{x} \quad (3)$$

substituindo *x'* e isolando *v*, tem-se:

$$v = \frac{y.z.\sin(\theta)}{x} \quad (4)$$

A Figura 4 mostra um exemplo contendo o resultado da junção de imagens usando a Retificação Cilíndrica.



Figura 4. Resultado da junção de imagens usando a Retificação Cilíndrica.

3. ALGORITMO SIFT

SIFT (*Scale Invariant Feature Transform*) é um algoritmo de visão computacional definido e publicado por David G. Lowe em 1999 (LOWE, 1999) que permite a detecção e extração de descritores que tem muitas propriedades que são desejáveis para correspondência (*matching*) de diferentes imagens de um objeto ou cena. Esses descritores são razoavelmente invariantes a mudanças de iluminação, ruído de imagem, rotação, escala e pequenas mudanças de ponto de vista. Os descritores são altamente distintos, permite que uma determinada característica possa ser corretamente combinada (*matched*) com alta probabilidade em um banco de dados de descritores de diversas imagens, provendo uma base para o reconhecimento de um objeto ou cena. A obtenção de descritores de uma imagem é feita por meio de quatro estágios, onde os dois primeiros descrevem a parte do detector e as duas seguintes a formação do descritor. O detector SIFT é baseado nos cálculos de diferenças de Gaussianas e o descritor SIFT utiliza histogramas de gradientes orientados para descrever a vizinhança local dos pontos de interesse. Esses estágios são descritos a seguir (LOWE, 2004).

Detecção de extremos no espaço de escala: Esse primeiro estágio da computação faz a busca em todas as escalas e locais de uma imagem. Isto é feito utilizando-se a diferença de filtros gaussianos de modo a se identificar pontos de interesse invariáveis à escala e orientação. Este é o estágio mais custoso computacionalmente do algoritmo.

Localização de pontos chave: Para cada local candidato, em que foi detectado um extremo (máximo e mínimo), um modelo detalhado é ajustado para se determinar a localização e escala. Pontos chave, ou pontos de interesse, são então selecionados baseando-se em suas medidas de estabilidade.

Determinação de orientação: Uma ou mais orientações são associadas para cada localização de ponto chave, baseadas em direções do gradiente da imagem local. As operações seguintes são realizadas nos dados da imagem relativamente transformados em relação

à orientação, escala e localização de cada ponto chave, assim provendo a invariância a estas transformações.

Descritor dos pontos chave: Neste estágio da computação, os gradientes locais da imagem são medidos na escala selecionada, na região ao redor de cada ponto chave. Estas medidas são então transformadas para uma representação que permite observar níveis significantes de distorção de forma e mudança na iluminação. São então criados histogramas de orientação para compor o descritor.

Para se encontrar a correspondência (*match*) entre duas imagens, devem-se encontrar pontos em comum entre as duas. Os pontos de interesse (pontos chave) são detectados utilizando o algoritmo SIFT, onde se obtém para cada ponto, um vetor de descritores com 128 dimensões. Tendo-se descritores de duas imagens, a tarefa de se encontrar a correspondência de uma imagem em outra é resumida por se encontrar entre os descritores de uma imagem, os melhores candidatos a serem seus equivalentes na outra imagem. Segundo Lowe (2004), a melhor correspondência candidata para cada ponto chave é encontrada pela identificação de seu vizinho mais próximo. O vizinho mais próximo é definido como o ponto chave com menor distância Euclidiana para o vetor de descritores invariantes.

Para exemplificar a tarefa de encontrar a correspondência de forma simples entre duas imagens I_1 e I_2, cujos descritores são definidos respectivamente por di_1 e dj_2, i, j os índices para cada um dos descritores de cada imagem, k o tamanho de cada descritor e m a magnitude de cada valor dos vetores, tem-se a seguinte representação:

$$\begin{aligned} di_1 &= (m_1 i_1, m_1 i_2, m_1 i_3, ..., m_1 i_k) \\ dj_2 &= (m_2 j_1, m_2 j_2, m_2 j_3, ..., m_2 j_k) \end{aligned} \quad (5)$$

A correspondência é realizada encontrando-se o melhor candidato dj_2 para determinado di_1, isto é feito procurando-se o vizinho mais próximo de di_1 entre todos os possíveis candidatos k em dj_2. O melhor candidato é definido por dj_2 que possui a menor distância Euclidiana em relação à di_1, ou seja, deseja-se encontrar j para todo i que minimize a função:

$$|di_1 - dj_2| = \sqrt{(m_1 i_1 - m_2 j_1)^2 + (m_1 i_2 - m_2 j_2)^2 + ... (m_1 i_k - m_2 j_k)^2} \quad (6)$$

Dessa forma são encontrados todos os pares de descritores correspondentes nas duas imagens I_1 e I_2. É utilizada uma estrutura de dados *k-d tree* (BEIS; LOWE, 1999), que é uma forma de árvore de busca binária balanceada, para encontrar o vizinho mais próximo de descritores. O algoritmo denominado *Best-Bin-First* (BBF) é utilizado para a busca, pois retorna o vizinho mais próximo com alta probabilidade e uma maior eficiência se comparada com uma busca sequencial.

4. RANSAC

O algoritmo RANSAC (*RANdom SAmple Consensus*) proposto por Fischler e Bolles (1981) é um método de estimação robusto projetado para extração dos *inliers*[1] do conjunto total de dados de entrada. Esses dados de entrada seriam os pontos chave detectados pelo algoritmo SIFT aplicado às imagens. Além de ser amplamente empregado para o reconhecimento de objetos, o algoritmo RANSAC possibilita encontrar as correspondências características geometricamente consistentes usadas para resolver a homografia entre pares de imagens. Esta atividade é realizada para a construção de panoramas de imagens utilizando os descritores extraídos das imagens com o algoritmo SIFT (BROWN; LOWE, 2007).

[1] *inliers*: pontos de dados que se ajustam com um determinado modelo desejado dentro de uma certa tolerância de erro.

A principal vantagem no uso do RANSAC é que ele possui a propriedade de ser um estimador robusto. Estimadores de parâmetros clássicos, como mínimos quadrados, se baseiam em todos os dados que tomam como entrada. Desse modo, se grande parte dos dados vier de uma fonte em que haja erros inerentes, ou ruidosos, (isto é, algum tipo de *outlier*[2]) o modelo final estimado será prejudicado. O RANSAC, porém, foi desenvolvido com o intuito de evitar que *outliers* atrapalhem a computação do modelo final.

Como mencionado por Fischler and Bolles (1981), ao contrário das técnicas convencionais que usam muitos dados para obter uma solução inicial, e assim, eliminar os *outliers*, o RANSAC utiliza apenas um conjunto com número mínimo e suficiente de pontos necessários para uma primeira estimativa, e procede aumentando o conjunto com pontos de dados consistentes. Uma amostra mínima depende do modelo a ser gerado. Geralmente, é o número *s* de dados que será utilizado para inicializar um modelo. Por exemplo, no caso do modelo a ser estimado ser uma reta, uma amostra mínima consiste de apenas dois elementos.

A Figura 5 mostra um exemplo apresentado por Hartley e Zisserman (2003) que pode ser facilmente visualizado. Nele é estimada uma reta ajustada a um conjunto de pontos bidimensionais. Isso pode ser feito estimando uma transformação unidimensional $x' = ax + b$ entre os pontos correspondentes em duas linhas.

O problema que é ilustrado na Figura 5 (a) é encontrar a reta que minimiza a soma das distâncias perpendiculares quadradas (regressão ortogonal) a partir de um conjunto de pontos de dados 2D, sujeito à condição que nenhum dos pontos válidos diverge dessa linha por mais de *t* unidades. Deseja-se então a linha ajustada aos dados e a classificação de dados em *inliers* (pontos válidos) e *outliers*. O limiar *t* é fixo de acordo com a medida de ruído (por exemplo, $t = 3\sigma$). Esse exemplo mostra que o ajuste dos pontos de dados é severamente afetado pelos *outliers*.

Já com o uso do algoritmo RANSAC (Figura 5 (b)), dois pontos são selecionados aleatoriamente, esses pontos definem uma reta. Essa seleção aleatória é repetida um número de vezes e a reta com maior número de pontos que ficam dentro de um limiar de distância é julgada como ajuste robusto. Os pontos dentro da distância do limiar são os *inliers* (e constitui o conjunto dos consensos).

Na Figura 5 (a) e (b) os pontos sólidos representam os *inliers*, e os pontos abertos representam os *outliers*. Na Figura 5 (b) as linhas pontilhadas indicam a distância do limiar, a linha passando pelos pontos *a* e *b* constitui a linha ajustada, o ponto *c* é um *outlier*.

A partir do conjunto mínimo necessário de pontos selecionados aleatoriamente, o RANSAC mede a precisão do modelo para todos os outros pontos, classificando-os em *inliers* ou *outliers*. Para tal, é definido um valor de limiar *t*, que é a distância máxima do modelo que um dado pode estar para ser considerado um *inlier*. O conjunto de *inliers* associado a um modelo é o conjunto de consenso (*consensus set*) do modelo. Assim, com várias iterações, o algoritmo é capaz de encontrar um modelo preciso para o problema em questão. Uma vantagem do RANSAC é a sua habilidade de realizar a estimativa de parâmetros de um modelo de forma robusta, ou seja, ele pode estimar parâmetros com um alto grau de acerto mesmo quando um número significativo de *outliers* esteja presente nos dados analisados (FISCHLER; BOLLES, 1981).

[2] *outliers*: pontos de dados que não se ajustam ao modelo correspondente ao objeto desejado, estão fora de uma certa tolerância de erro.

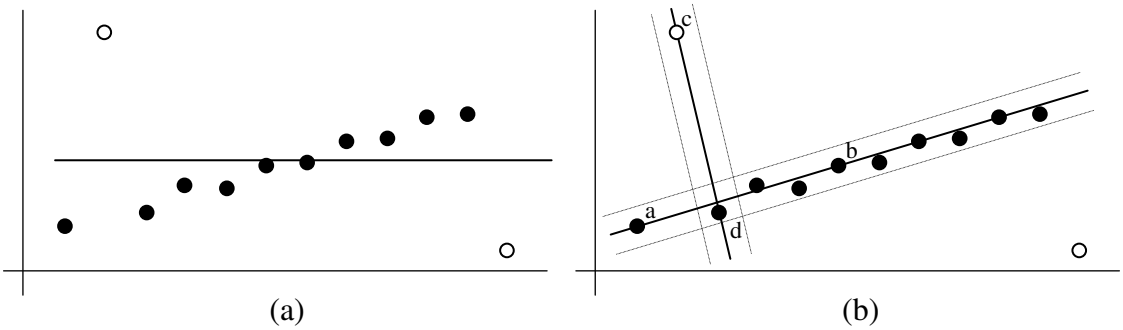


Figura 5. Estimação de uma linha reta ajustada a um junto de pontos 2D.
Fonte: Hartley e Zisserman (2003).

As etapas de execução são resumidas como segue no Algoritmo 1 e foram obtidas em Fischler e Bolles (1981) e Hartley e Zisserman (2003):

Algoritmo 1 – RANSAC

Objetivo: Ajuste robusto de um modelo para um conjunto *S* que contêm *outliers*.

Passos:

1: Selecione aleatoriamente o número mínimo de pontos *s* requeridos de um conjunto *S* para determinar os parâmetros do modelo.

2: Determine o conjunto de pontos de dados S_i que estão dentro de um limiar de distância *t* do modelo (matriz fundamental). O conjunto S_i é o conjunto de consensos da amostra e define os *inliers* de *S*.

3: Se o tamanho de S_i (número de *inliers*) é maior que *T* (número necessário a ser alcançado), os parâmetros modelo são re-estimados usando todos os *inliers* identificados (S_i) e termina.

4: Se o tamanho de S_i é menor que *T*, um novo subconjunto é selecionado e repete os passos anteriores.

5: Após *N* tentativas o maior conjunto de consensos de S_i é selecionado, e o modelo é re-estimado usando todos os pontos no subconjunto S_i.

O número de iterações *N* é escolhido grande o suficiente para assegurar uma probabilidade *p* (normalmente estabelecido para 0.99), que pelo menos uma das amostras aleatórias do conjunto de *s* pontos não inclui um *outlier*. Considere *w* a representação da probabilidade que algum ponto de dado selecionado é um *inlier* e $\varepsilon = 1 - w$ a probabilidade de observar um *outlier*. Pelo menos *N* iterações (cada uma de *s* pontos) são requeridos, onde (HARTLEY; ZISSERMAN, 2003):

$$(1-w^s)^N = 1-p \tag{7}$$

e assim o número de iterações necessárias é dado por:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{8}$$

5. BLEND

Na construção de imagens panorâmicas a costura (junção) entre as imagens ocorre com a sobreposição entre duas imagens, isso faz com que o panorama tenha uma diferenciação nas costuras, por fatores como diferentes ângulos de

captura da imagem e iluminação da cena. Isso faz com que o panorama fique com um aspecto de corte entre as junções como mostrado na Figura 7a.

O objetivo de um algoritmo de "*Blending*" é possibilitar a um mosaico de imagens a não identificação onde ocorrem as junções, ou pelo menos tornar isso menos evidente, produzindo assim um panorama onde é impossível identificar o inicio ou o fim, que é o caso do panorama 360.

Para este trabalho foram investigados alguns algoritmos de *Blend* como *Optimal Seam* e *Feathering* (LEVIN, 2006).

5.1. Optimal Seam

Este método se baseia na construção de um mosaico onde a costura ou junção entre duas imagens tenha a costura perfeita, assim existem requisitos que devem ser atendidos, são eles:

- A diferença de intensidade entre os pixels das duas imagens deve ser pequena;
- As estruturas geométricas dos pixels das imagens vizinhas devem ser similares.

Atender a esses dois requisitos não é uma tarefa trivial, pois os valores de intensidades luminosas e estruturas geométricas nem sempre podem ser valores mínimos simultaneamente, no entanto pode-se encontrar uma costura em que a soma de ambos os requisitos tenha um menor valor possível (XING et al., 2010).

5.1. Feathering

No entanto, para este trabalho, somente o algoritmo de *Feathering* (UYTTENDAELE, 2001) (LEVIN, 2006) foi implementado, por se tratar de um algoritmo que gera ótimos resultados com imagens que são capturadas num mesmo intervalo de tempo. *Feathering* é uma técnica usada na computação gráfica para suavizar as bordas ou borrá-las, essa técnica é muito usada para unir ou sobrepor imagens.

Deste modo, o algoritmo implementado neste trabalho faz a junção das imagens considerando as tonalidades que serão sobrepostas. Portanto, o valor de cada pixel é determinado pela região que o mesmo se encontra, efetuando assim uma mistura de pixels calculada pela média ponderada. Quanto mais perto da imagem de origem maior será sua relevância para a imagem resultante. A Figura 6 mostra uma representação gráfica do *Blend Feathering*.

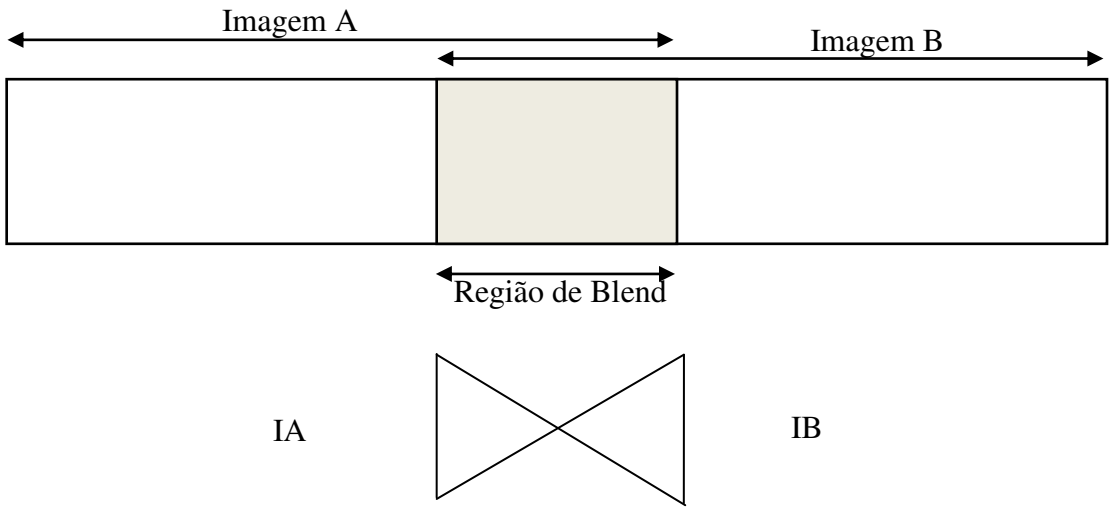


Figura 6. Representação do *Blend Feathering.*

A equação implementada no algoritmo do *Blend Feathering* é representada por:

$$PB(i, j) = (1 - w).PA(i, j) + w.PB(i, j) \qquad (9)$$

A Figura 7 (b) mostra o resultado do algoritmo *Blend Feathering* aplicado em um par de imagens.



A B

Figura 7. Imagens sem *Blend* (a) e com a aplicação do *Blend Feathering* (b).

6. AJUSTE NAS IMAGENS

Na construção de um panorama ocorre um fenômeno que é causado pelas junções de múltiplas imagens, assim ao decorrer das junções o erro da junção inicial é propagado para as demais, desse modo à medida que a homografia é calculada e aplicada novamente o erro vai se propagando pelo mosaico.

Para Brown e Lowe (2007) este erro deve ser ajustado, ele descreve um algoritmo de ajustamento denominado "*Bundle Adjustment*", que tem por objetivo ajustar esses erros propagados para formar um panorama mais condizente com a realidade.

A Figura 8 retrata um mosaico onde o erro foi propagado, nota-se que o erro fica maior nas extremidades laterais, pois nelas estão presentes a soma dos erros. Por se tratar de uma imagem 360, fica evidente que as extremidades não se unirão, desse modo, faz-se necessário um ajustamento para que o erro não se propague durante a junção.



Figura 8. Panorama sem ajuste.

O algoritmo proposto neste trabalho aplica a matriz homográfica gradativamente em apenas parte de uma das imagens que sofrerá junção. Como um panorama 360 é constituído de várias imagens, quando aplicada a matriz em apenas parte de uma imagem, levando em conta que há no máximo 50% de sobreposição, a junção das imagens seguintes não são afetadas com o erro da junção anterior.

O algoritmo aplica a matriz homográfica na metade da imagem de distorção. Inicialmente é aplicada 100% da matriz homográfica, diminuindo gradativamente até chegar a uma matriz onde só exista translação em *x* e *y*. O algoritmo é representado nas equações na sequência.

A matriz homográfica produzida pelo algoritmo RANSAC representa a seguir é aplicada inicialmente pelo algoritmo:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \quad (10)$$

a matriz homográfica (11) é modificada preservando apenas h_{13} e h_{23} que representam o deslocamento em *x* e em *y*, e aplicada na outra metade da imagem.

$$B = \begin{pmatrix} 1 & 0 & h_{13} \\ 0 & 1 & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad (11)$$

O fator de diminuição gradativa representado em (12) faz com que a matriz homográfica diminua de forma gradativa para que não aconteça uma distorção brusca entre a matriz representada por (10) e por (11).

$$f_{lc} = \frac{b_{lc} - h_{lc}}{tam} \quad (12)$$

onde "*tam*" é a metade do tamanho da imagem que sofrerá distorção.

A matriz de fator de gradação em (13) representa as matrizes que serão geradas a partir da matriz homográfica em (10) até chegar em (11):

$$J = \begin{pmatrix} h_{11} + f_{11}.x & h_{12} + f_{12}.x & h_{13} \\ h_{21} + f_{21}.x & h_{22} + f_{22}.x & h_{23} \\ h_{31} + f_{31}.x & h_{32} + f_{32}.x & 1 \end{pmatrix} \quad (13)$$

Em (14) é representada a verificação para a aplicação da distorção de *J* (13) ou de *B* (11) de acordo com o valor de *x*:

$$Ponto(x,y) = \begin{cases} Ponto(x,y).J, & x < tam \\ Ponto(x,y).B, & x \geq tam \end{cases} \quad (14)$$

Aplicando esse algoritmo nas imagens tem-se o seguinte resultado representado na Figura 9. A imagem panorâmica resultante contempla os 360 graus, possibilitando a visualização em qualquer parte sem apresentar emendas.



Figura 9. Imagem Panorâmica 360 resultante a partir da metodologia proposta.

7. CONSTRUÇÃO DO VISUALIZADOR

O visualizador para as imagens panorâmicas 360 foi implementado no Visual Studio 2010 usando a linguagem de programação C#. Para isso, foi usado um componente visual "*WebBrowser*" para possibilitar a visualização de uma página Web. Página esta que foi construída usando um recurso de um visualizador interativo denominado PTViewer (DERSCH, 2001) que permite o deslocamento do ponto de vista em toda a extremidade da imagem panorâmica.

PTViewer é um recurso para a visualização de panorama equirectangular e cilíndrico, neste trabalho foi usado para visualizar um panorama 360 com uma projeção cilíndrica. É um recurso de código aberto e grande parte de seu código foi alterado por outras pessoas além dos seus criadores, portanto, existem versões gratuitas e comerciais. Desenvolvido utilizando a linguagem Java, pode ser aberto em qualquer navegador que suporte a linguagem a qual foi criado. É um visualizador de panoramas

interativos, que proporciona ao usuário efetuar *Zoom in*, *Zoom out* e movimentar o panorama 360.

A Figura 10 mostra o visualizador, nele existe um menu na parte superior onde o usuário pode selecionar as imagens que ele deseja unir para a formação do panorama 360. Na opção "Abrir Imagens", após o processo de junção das imagens e a formação do mosaico, o visualizador gera uma página Web utilizando o recurso do PTViewer. Uma vez gerada a página Web, o visualizador a exibe no componente "*WebBrowser*". Caso deseje-se salvar essa visualização assim como o panorama, deve-se escolher a opção de "Salvar 360".

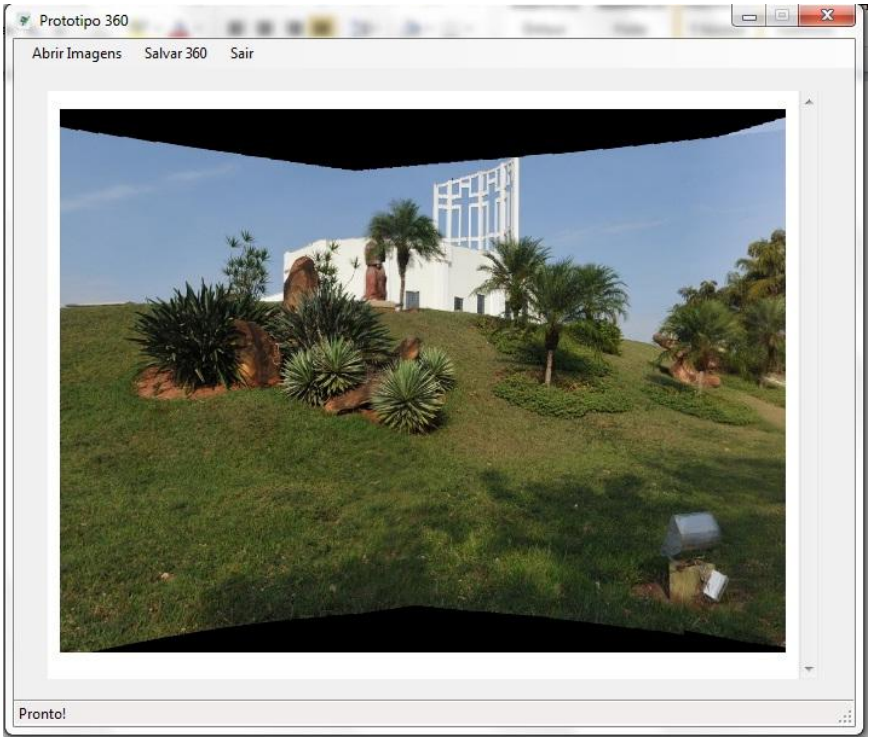


Figura 10. Visualizador de imagem panorâmica 360.

8. CONCLUSÕES

Foram realizados diferentes experimentos para a construção de panoramas de imagens utilizando algumas abordagens. Um primeiro experimento na formação de panorama foi obtido aplicando a matriz homográfica gerada pelo algoritmo RANSAC durante a extração dos *inliers* a partir de cada par de imagens em toda a imagem da junção. Os resultados mostraram que esta abordagem provoca uma grande deformação na imagem resultante esticando-a nas extremidades laterais. Esta deformação inviabiliza a formação de um mosaico de imagens contemplando os 360 graus.

Em um segundo experimento foi aplicada a retificação cilíndrica nos pares de imagens anteriormente ao estágio da computação do algoritmo SIFT para a extração dos descritores de pontos chave. Os resultados apresentaram-se melhores em relação ao experimento sem a retificação cilíndrica, formando uma imagem panorâmica que contempla os 360 graus, porém, existe nessa abordagem o problema da distorção causada pela propagação dos erros. Esses erros afetam os resultados finais, as imagens panorâmicas, onde não é possível juntar as suas extremidades laterais para a formação de um panorama 360 sem emendas.

Os resultados obtidos com a metodologia proposta neste trabalho demonstram-se ser eficientes para a junção de imagem para a formação de panoramas 360. Com a aplicação desta abordagem, conseguiu-se construir imagens panorâmicas com suavizações eficientes nas junções das imagens se comparado com programas comerciais disponíveis na Web para testes que realizam esta mesma atividade.

9. REFERÊNCIAS

BEIS, J.; LOWE, D. G. Shape indexing using approximate nearest-neighbour search in high dimensional spaces. In: **Conference on Computer Vision and Pattern Recognition**. Puerto Rico, pp. 1000-1006, 1997.

BROWN, M.; LOWE, D. G. Automatic Panoramic Image Stitching using Invariant Features **International Journal of Computer Vision**, 74(1):59-73, 2007.

CURTIN, D. P. **Imagem Digital. 2001**. Disponível em: http://www.imagem-digital.com>. Acesso em: 17 de dez. de 2010.

DERSCH, H. **PTViewer Documentation**. 2001. Disponível em: http://webuser.fh-furtwangen.de/~dersch/PTVJ/doc.html. Acesso em: 12 de set. de 2011.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, 1981.

HARTLEY, R.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. University Press, Cambridge, 2nd. ed., 2003.

HILL, R. A lens for whole sky photographs. **Quarterly Journal of the Royal Meteorological Society**, 2007.

LEVIN, A.; ZOMET, A.; PELEG, S.; WEISS, Y. Seamless Image Stitching in the Gradient Domain. In: **9th European Conference on Computer Vision.** Proceedings… Austria, 2006.

LOWE, D. G. Object Recognition from Local Scale Invariant Features. **International Journal of Computer Vision,** 2004.

LOWE, D. G. Object recognition from local scale-invariant features. In: **International Conference on Computer Vision**, Corfu, Greece, pp. 1150-1157, 1999.

POLLEFEYS, M. **Self-calibration and metric 3d reconstruction from uncalibrated image sequences.** Ph.D. Thesis, 1999.

ROBERTO, R. A.; SANTOS, A. G.; TEICHRIEB, V.; KELNER, J. Retificação polar: um método eficaz para retificar pares de imagens e gerar reconstrução densa robusta. **Revista Eletrônica de Iniciação Científica**, v. X, p. 1, 2010.

TSAN HU, O. R. **Contribuições ao desenvolvimento de um sistema de telepresença por meio da aquisição, transmissão e projeção em ambientes imersivos de vídeos panorâmicos**. 2006.128f. Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistema Eletrônicos, São Paulo.

UYTTENDAELE, M.; EDEN, A.; SZELISKI, R. Eliminating ghosting and exposure artifacts in image mosaics. In: **CVPR**, 2001.

XING, C.; WANG, J.; XU, Y. 2010 reported on conference. **An optimal seam line based mosaic method for uav sequence images**. Disponível em http://www.gmat.unsw.edu.au/snap/publications/xing_etal2010b.pdf. Acesso em 06 de julho de 2011.

Recebido em: 05/06/2013

Revisado em: 25/06/2013

Aceito em: 01/07/2013