

Registro e Mosaico de Imagens Obtidas por Câmera Digital a bordo de VANT

Marcos Eduardo Gomes Borges *
Marina Laís da Silva Nascimento *
Juliano E. C. Cruz *
Leila Maria Garcia Fonseca †

Instituto Nacional de Pesquisas Espaciais – INPE
Programa de Mestrado em Computação e Matemática Aplicada *
Divisão de Processamento de Imagens †
São José dos Campos, Brasil
{marcoseborges, marina.lsnascimento, juliano.ecc}@gmail.com, leila@dpi.inpe.br

Resumo—Neste trabalho foi pesquisado soluções para registro e mosaico de imagens adquiridas por câmera digital a bordo de VANT. A ideia é apresentar soluções para dois tipos de problemas que ocorrem ao mosaicar sequências de imagens aéreas: i) distorções geométricas inseridas na imagens devido às variações de altitude, ii) distorções (escala, projeção e ângulo de visada) nas imagens de baixas altitudes e que possuem cenas de objetos altos, tais como prédios e montanhas.

Palavras-chave—Registro de Imagens, Mosaico, VANT, TerraLib, SIFT.

I. INTRODUÇÃO

A utilização de Veículos Aéreos não Tripulados (VANTs) tem apresentado grande crescimento nos últimos anos devido a diversos fatores, tais como ausência de tripulação em tarefas tediosas, cansativas ou que envolvem riscos à tripulação, baixo custo operacional e de fabricação comparados às aeronaves convencionais, entre outros. Imagens aéreas obtidas através de VANTs possuem grandes aplicabilidades [1] e o objetivo geral deste trabalho é encontrar solução para distorções geométricas que surgem ao realizar registro e mosaico de imagens adquiridas por câmera digital a bordo de aeronaves não tripuladas.

O mosaico de sequências de imagens aéreas apresenta alguns problemas de distorções geométricas devido às variações de altitudes da aeronave e distorções devido as diferenças de escala, projeção e ângulo de visada em cenas de baixa altitude e que apresentam prédios e montanhas. Outro problema também enfrentado ao gerar o mosaico nesse tipo de imagem, é que quando a aeronave realiza curvas para seguir o plano de voo traçado captura imagens com sistema de coordenadas rotacionadas em ângulos diferentes e desconhecidos, e isso também gera distorções que buscamos resolver neste trabalho.

Na Figura 1 pode-se observar o exemplo de um plano de voo para um VANT. Neste trabalho, as imagens adquiridas pela câmera digital a bordo da aeronave não possuem georeferenciamento e são coletadas a cada um segundo. Após o término da aquisição das imagens, as mesmas necessitam ser mosaicadas. O procedimento inicial para essa tarefa é o registro de imagens, que inicia a busca por correspondências entre imagens diferentes que representam a mesma cena [2]. Neste trabalho, a busca por correspondências entre pontos de imagens diferentes é realizada e comparada entre os algoritmos: SIFT proposto por [3] e pelos algoritmos de registro implementados na biblioteca TerraLib.

II. REGISTRO E MOSAICO DE IMAGENS

O registro de imagens pode ser entendido como um processo de casamento entre duas imagens que possuem uma área comum. A



Fig. 1. Exemplo de Plano de Voo do VANT

imagem tomada como base de registro é chamada de imagem de referência e a imagem a ser registrada é chamada de imagem de ajuste. O processo basicamente envolve três etapas:

- 1) Obtenção de pontos de controle;
 - a) Extração de feições;
 - b) Casamento das feições extraídas;
- 2) Determinação da função de transformação;
- 3) Sobreposição das imagens.

As imagens a serem registradas podem ser relacionadas através de função de transformação simples se a geometria das imagens for semelhante. Se a geometria das imagens for diferente, as transformações podem ser aproximadas utilizando um função polinomial cujos parâmetros são determinados a partir das coordenadas dos pontos de controle. O número de pontos de controle representa a situação de um sistema de equações determinado. Entretanto, como as coordenadas medidas dos pontos de controle estão sujeitas a erros, convém usar um número de pontos maior que o número mínimo, o que define um sistema de equações sobredeterminado.

O produto gerado através das técnicas de registro de imagens é o mosaico, que nada mais é do que uma composição de imagens

adquiridas de diferentes pontos de vista visando contruir uma imagem maior, permitindo assim uma visão global da cena.[4]

III. FERRAMENTA UTILIZADA

A. TerraLib

TerraLib é uma biblioteca *open source* de classes e funções SIG disponível na Internet, possibilitando portanto, um ambiente colaborativo e seu uso para o desenvolvimento de novas ferramentas. Atualmente é desenvolvida pelo INPE, PUC-Rio e FUNCATE.[5]

A utilização desta biblioteca se deu em todas as etapas necessárias para a realização do registro, ou seja, na obtenção de pontos de controle, na transformação espacial da imagem e na sobreposição das imagens.

IV. DETECÇÃO DE PONTOS DE CONTROLE

Ponto de interesse é como é chamado qualquer ponto de uma imagem em que o sinal tenha uma mudança bidimensional. Cantos ou ângulos no formato L, T e Y obecem essa definição assim como pontos pretos em um fundo branco, o final de ramificações ou qualquer textura bidimensional significante.[6] Ponto de controle são, portanto, os pontos de interesse de imagens de sensoriamento remoto que são utilizados para aplicar transformações geométricas afim de realizar o processo de registro entre as determinadas imagens.[7]

A. Detector Moravec

O detector Moravec foi uma dos primeiros detectores de ponto de interesse a serem desenvolvidos, sendo descrito primeiramente em 1977 por Hans Moravec[8]. Esse detector é baseado na função de autocorrelação do sinal. Ele compara as diferenças de nível de cinza entre a janela atual e de janelas deslocadas em quatro direções paralelas à colunas e linhas. Se o mínimo dessas quatro diferenças é superior a um determinado limiar, então um ponto de interesse foi encontrado.[6]

O operador Moravec pode ser definido matematicamente como:

$$\begin{aligned} V_1 &= \frac{1}{p(q-1)} \sum_{i=-k}^k \sum_{j=-l}^{l-1} (g(i, j) - g(i, j+1))^2 \\ V_2 &= \frac{1}{(p-1)q} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (g(i, j) - g(i+1, j))^2 \\ V_3 &= \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (g(i, j) - g(i+1, j+1))^2 \\ V_4 &= \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (g(i, j+1) - g(i+1, j))^2 \\ V &= \min(V_1, V_2, V_3, V_4) \end{aligned}$$

Onde $p = 2k + 1$ e $q = 2l + 1$, sendo que $k \times l$ é o tamanho da janela utilizada.

Na biblioteca TerraLib a classe `TePDIMMIOMatching` implementa o método Moravec.

B. Fluxo Ótico

Fluxo ótico é a distribuição de velocidades aparentes do movimento de padrões de brilho em uma imagem.[9] Assim, pode-se obter informações importantes a respeito da distribuição espacial dos objetos visualizados e da taxa de mudança dessa distribuição. Discontinuidades do fluxo ótico podem ajudar a segmentar uma determinada imagem em regiões que correspondem a diferentes a diferentes objetos. Esse conceito começou a ser estudado na década de 1940 e foi publicado primeiramente pelo psicólogo James Gibson. [10] [11]

Um pixel tendo a localização (x, y, t) com intensidade I será movido em $\Delta x, \Delta y$ and Δt entre os quadros de uma mesma imagem. Assim, chega-se a equação:

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} = 0$$

simplificando tem-se que,

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0$$

Onde V_x e V_y são os componentes x e y do fluxo ótico.

Na biblioteca TerraLib a classe `TePDIOFMatching` implementa o método de fluxo ótico.

C. SIFT

O algoritmo SIFT, acrônimo para *Scale Invariant Feature Transform*, foi publicado por David Lowe [12] em 1999 e tem função de construir descritores de pontos-chaves de uma imagem, sendo este descritores independentes das mudanças de escala, rotação, translação e luminosidade que uma imagem pode sofrer. Utilizou-se neste trabalho a implementação em C++ obtida em [13]. O SIFT é utilizado na busca por correspondências entre sequência de imagens diferentes que contenham partes da mesma cena. A busca é feita através de pontos-chave correspondentes, utilizando-se seus descritores. Nesta pesquisa a distância euclidiana é utilizada em três abordagens diferentes para avaliar a mais apropriada para imagens obtidas por VANTs. Os algoritmos de cada abordagem são: *DistEuclidConvencional*, o qual aplica a função de busca diretamente, sem tratar seus resultados. *DistEuclidRedundante*, que chama a função de busca duas vezes, a segunda chamada é feita invertendo-se os parâmetros da função, somente correspondências que ocorram em ambas são guardadas. *DistEuclidEsc* onde observa-se a continuidade de escala entre segmentos de reta traçados entre os pontos pertencentes e às correspondências geradas por esta função.

O algoritmo SIFT, acrônimo para *Scale Invariant Feature Transform*, foi publicado por David Lowe em 1999 e mostrou ser robusto no reconhecimento de objetos parcialmente cobertos e espalhados pela imagem. O algoritmo transforma uma imagem em uma coleção de vetores de feições locais, que são invariáveis a translação, variação de escala e rotação e são parcialmente invariáveis a mudanças de iluminação e a projeções afim e 3D. Esses vetores são obtidos através da função de diferenças gaussianas.

[12]

$$\begin{aligned} M_{ij} &= \sqrt{(I_{ij} - I_{i+1,j})^2 + (I_{ij} - I_{i,j+1})^2} \\ R_{ij} &= \text{atan2}(I_{ij} - I_{i+1,j}, I_{i,j+1} - I_{ij}) \end{aligned}$$

V. CASAMENTO DE PONTOS DE CONTROLE

Após se obter os pontos de controle de ambas as imagens em que deseja-se realizar o registro, é necessário descobrir qual é o ponto da imagem de ajuste correspondente à um determinado ponto na imagem de referência.

Dentre diversos métodos existentes para se descobrir a correspondência entre os pontos, as classes de casamento do TerraLib utilizam o método estatístico de correlação cruzada normalizada. Seu funcionamento se dá ao extrair pequenas janelas ao redor de cada ponto de controle nas duas imagens. Aplica-se, então, aos pares o método(descrito abaixo) entre todas as janelas obtidas na imagem de referência com as obtidas na imagem de ajuste.[4][14][15]

$$R(i, j) = \frac{\sum_{l=0}^{K-1} \sum_{m=0}^{L-1} W_z(l, m) \cdot S_{ij}(l, m)}{\sqrt{\sum_{l=0}^{K-1} \sum_{m=0}^{L-1} W_z^2(l, m) \sum_{l=0}^{K-1} \sum_{m=0}^{L-1} S_{ij}^2(l, m)}}$$

Onde S_{ij} é a janela da imagem de referência e W_z a janela da imagem de ajuste. Assim, quanto maior valor de $R(i, j)$ maior é a semelhança entre as duas janelas comparadas.

VI. TRANSFORMAÇÕES GEOMÉTRICAS

Com o casamento dos pontos de controle já realizado, agora é necessário aplicar funções de transformação geométrica para que se possa ter a imagem de ajuste adequada geometricamente e espacialmente afim de sobrepor as imagens obtendo o mosaico.

Transformação geométrica é o nome que se dá à aplicação de uma determinada função matemática em uma determinada figura geométrica em que o resultado é geometricamente igual ou semelhante à figura original.

A. Transformações elementares

1) *Translação*: A translação desloca um determinado ponto ou conjunto de pontos uma determinada distância em um determinado sentido.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Onde t_x e t_y são respectivamente as taxas de translação no eixo x e y . [16]

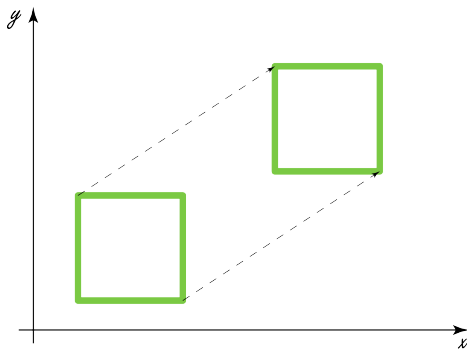


Fig. 2. Exemplo de translação em um espaço bidimensional

2) *Variação de escala*: A variação de escala é o fato de se esticar ou encolher uma determinada figura em relação aos eixos x e y .

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} v_x & 0 \\ 0 & v_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Onde v_x e v_y são respectivamente as taxas de escala no eixo x e y . [16]

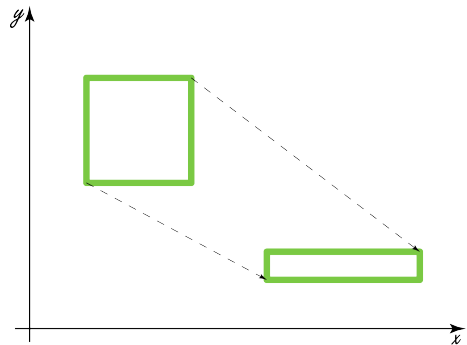


Fig. 3. Exemplo de variação de escala em um espaço bidimensional

3) *Rotação*: Na rotação rotaciona-se a figura em torno de um determinado eixo.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Onde θ é o ângulo que a figura será rotacionada em relação a posição original levando em consideração a origem como eixo. [16]

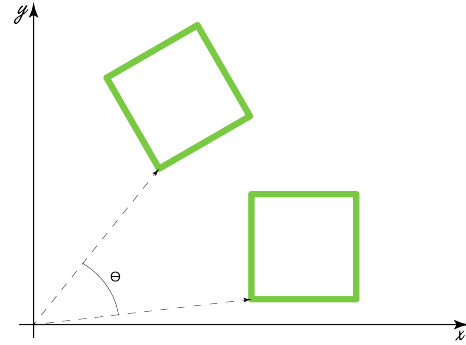


Fig. 4. Exemplo de rotação em um espaço bidimensional

4) *Cisalhamento*: O cisalhamento resulta em um movimento translacional na direção de um eixo no qual a magnitude aumenta ao longo do outro eixo.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Onde c é o coeficiente de cisalhamento. [16]

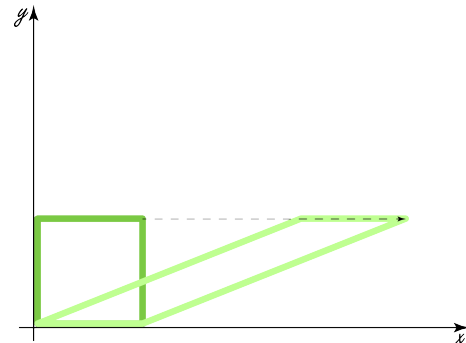


Fig. 5. Exemplo de cisalhamento em um espaço bidimensional

5) *Projeção*: Projeção é o processo no qual se obtém uma figura bidimensional a partir de uma cena tridimensional. [16]

$$\begin{bmatrix} x \\ y \\ z \\ \frac{z}{d} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

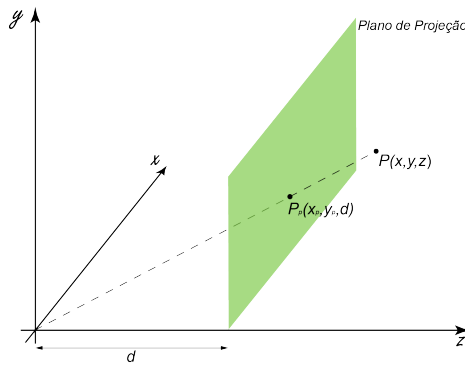


Fig. 6. Exemplo de projeção

B. Tipos de transformação

Existem basicamente quatro tipos de transformações geométricas, mas somente três estão disponíveis no TerraLib: afim, polinomial de segundo grau e projetiva.

1) *Linear*: A transformação linear ou também chamada de euclidiana, pode ser definida como:

$$F(x) = Qx$$

Sendo que $Q = n \times m$. Ou seja, transformações do tipo lineares são aquelas obtidas através de multiplicações matriciais. Assim, através das transformações lineares pode-se realizar operações de rotação, cisalhamento e variação de escala.[16]

2) *Afim*: A transformação afim é uma transformação que compreende as operações que a transformação linear consegue executar mais a operação de translação. A transformada em questão pode ser definida como:

$$F(x) = Qx + q$$

Sendo que $Q = n \times m$ e q tem tamanho m . [16]

3) *Polinômio de Segundo Grau*: Existe também transformações de segundo grau que são descritas como:

$$F(x) = Wa$$

$$W = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n y_n & x_n^2 & y_n^2 \end{bmatrix}$$

Onde $W = n \times m$ e a tem tamanho m . [7]

4) *Projeção*: A projeção é nada mais que uma transformação linear em um espaço projetivo.[16] A forma geral da transformação projetiva é descrita como:

$$F(x) = \left(\frac{a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + a_{1n+1}x_{(n+1)}}{b_1x_1 + b_2x_2 + \dots + b_nx_n + b_{n+1}}, \dots, \frac{a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + a_{mn+1}x_{(n+1)}}{b_1x_1 + b_2x_2 + \dots + b_nx_n + b_{n+1}} \right)$$

VII. AVALIAÇÃO DA PRECISÃO DO REGISTRO

Uma operação importante para um sistema de registro é a avaliação da transformação computada. Sejam g e G as imagens de ajuste e referência respectivamente, e x , y e X , Y os conjuntos de pontos de controle casados que definem uma transformação de distorção T . Podemos verificar o quão a transformação é correta através do cálculo do erro RMSE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - T(X_i))^2 + (y_i - T(Y_i))^2}$$

onde:

$(x_i, y_i), (X_i, Y_i), i = 1..n$ é o conjunto de pares de pontos de controle obtidos no processo de casamento. T é a função de distorção entre as imagens obtida através do casamento de pontos de controle.

VIII. FLUXOGRAMA PARA GERAR O MOSAICO

A montagem do mosaico se baseia em duas classes do TerraLib, o arquivo *MMIOMatching* recebe duas imagens e apresenta como resultado os pontos de controle da cena, que são representados através de dois vetores para cálculo da função de transformação. Esses pontos localizados são acrescentados na imagem que é salva com outro nome. (Talvez colocar algo da dissertação do Dmitri Fedorov) Já o segundo arquivo *Mosaic* recebe como entrada a saída do arquivo *MMIOMatching*, ou seja, os vetores que representam os pontos de controle, além das imagens que serão mosaicadas e o nome da nova imagem. (Devemos definir as configurações nesse item, qual foi a função de transformação que usamos entre outros detalhes, seria interessante fazer um desenho com explicando esse cliço)

IX. RESULTADOS

Devemos descrever quanto tempo levou para executar, quantas imagens foram utilizadas para gerar o mosaico, quais as vantagens.

X. COMPARAÇÃO DOS ALGORITMOS

Comparação do SIFT e dos algoritmos do TerraLib

XI. CONCLUSÃO

Conclusão aqui.

REFERÊNCIAS

- [1] A. Canhoto; E. H. Shiguemori; M. A. P. Domiciano, "Image sequence processing applied to autonomous aerial navigation," *Signal and Image Processing Applications*, 2009.
- [2] E. H. Shiguemori G. A. M. Goltz, "Aplicação do algoritmo sift em imagens de navegação autônoma," *Workshop Anual de Pesquisa e Desenvolvimento do IEAv*, 2008.
- [3] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.
- [4] D. Fedorov, "Sistema semi-automático de registro e mosaico de imagens," M.S. thesis, INPE, 2002.
- [5] DPI/INPE, "Terralib website," <http://www.terralib.org/>, 2012.
- [6] C. Schmid; R. Mohr; C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, 2000.
- [7] *Remote Sensing: Models and Methods for Image Processing*, Elsevier Inc, 2007.
- [8] H. P. Moravec, "Towards automatic visual obstacle avoidance," *5th International Joint Conference on Artificial Intelligence*, 1977.
- [9] *The Perception of the Visual World*, Riverside Press, 1950.
- [10] J.J Gibson, "On the analysis of change in the optic array," *Scandinavian J. Psyehol* 18, 1977.
- [11] B. Horn; B. Schunck, "Determining optical flow," *Artificial Intelligence* 17, 1981.
- [12] David G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, 1999.
- [13] A. Vedaldi, "Sift++: A lightweight c++ implementation of sift," <http://www.vlfeat.org/~vedaldi/code/siftpp.html>, 2011.
- [14] Fonseca L. M. G; Manjunath B. S., "Registration techniques for multi-sensor remotely sensed imagery," *PE&RS*, 1996.
- [15] F. Zhao; Q. Huang; W. Gao, "Image matching by normalized cross-correlation," *ICASSP*, 2006.
- [16] *Computer Graphics: Principles and Practice 2nd Edition in C*, Addison-Wesley, 1995.