

# 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 nas 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.

**Index Terms**—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 no 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 às diferenças de escala, projeção e ângulo de visada em cenas de baixa altitude e que apresentam prédios e montanhas. Outro problema também enfrentado ao mosaicar esse 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. As imagens adquiridas pela câmera digital 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. ALGORITMO SIFT

O algoritmo SIFT - Scale Invariant Feature Transform foi desenvolvido por [1] em 1999 e sua função é construir descritores de pontos-chaves de uma imagem, sendo este descritores independentes das



Fig. 1. Exemplo de Plano de Voo do VANT

mudanças de escala, rotação, translação e luminosidade que uma imagem pode sofrer. Utilizamos neste trabalho a implementação em C++ obtida em [4]. 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.

### III. 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. Assim, deseja-se possibilitar uma nova geração de aplicações SIG baseada nos avanços tecnológicos dos bancos de dados espaciais. A TerraLib é desenvolvida pela DPI, divisão pertencente ao INPE, Tecgraf, PUC-Rio e FUNCATE.

### IV. TRANSFORMADAS GEOMÉTRICAS

**T**ransformação 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  $y$ . [5]

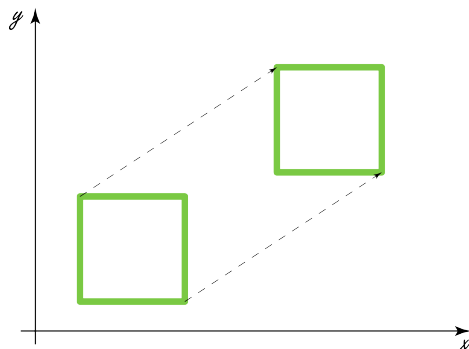


Fig. 2. Exemplo de translação em um espaço bidimensional

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  $y$ . [5]

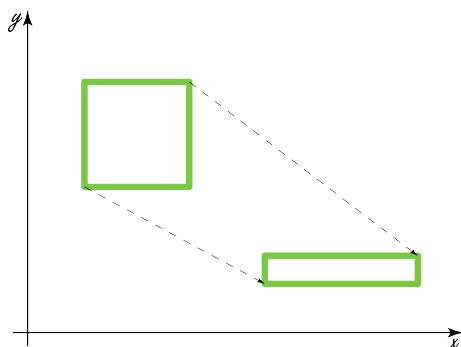


Fig. 3. Exemplo de variação de escala em um espaço bidimensional

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  $\theta$  é o ângulo que a figura será rotacionada em relação a posição original levando em consideração a origem como eixo. [5]

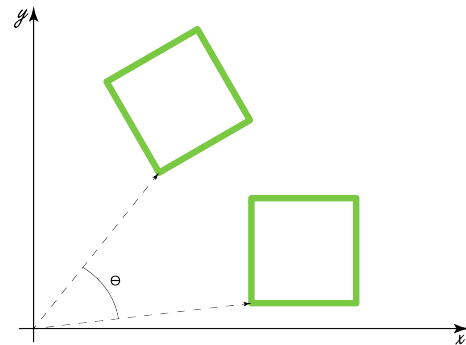


Fig. 4. Exemplo de rotação em um espaço bidimensional

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. [5]

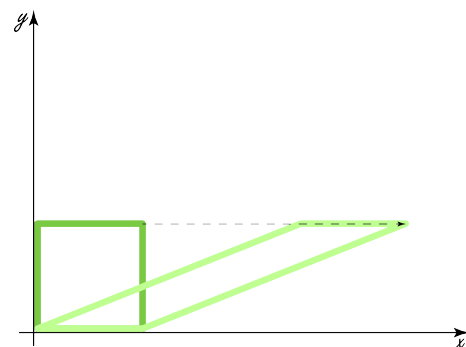


Fig. 5. Exemplo de cisalhamento em um espaço bidimensional

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

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

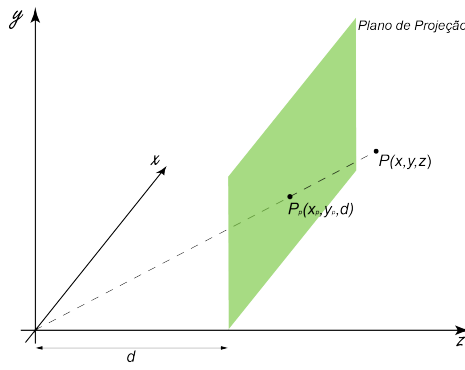


Fig. 6. Exemplo de projeção

### B. Tipos de transformação

É necessário primeiramente

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.[5]

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  $m$ . [5]

3) *Polinômio Segundo Grau*:  $F(x) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2$

4) *Projeção*: A projeção é nada mais que uma transformação linear em um espaço projetivo.[5] A forma geral da transformação projetiva é descrita como:

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

### V. DETECÇÃO DE PONTOS DE INTERESSE

**P**onto de interesse é como é chamado qualquer ponto de uma imagem em que o sinal mude bidimensionalmente. Cantos ou ângulos no formato L, T e Y obedecem essa definição assim como pontos pretos em um fundo branco, o final de ramificações ou qualquer textura bidimensional significativa.[6]

#### A. Detector Moravec

O operador Moravec foi uma dos primeiros detectores de ponto de interesse a serem desenvolvidos, sendo descrito primeiramente em 1977 por Hans Moravec.[7] Esse operador é baseado na função de auto-correlação do sinal. Ele compara as diferenças de nível de cinza entre a janela atual e de janelas deslocadas em quatro direções paralelas às 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:

$$V_1 = \frac{1}{p(q-1)} \sum_{i=-k}^k \sum_{j=-l}^{l-1} (g(i,j) - g(i,j+1))^2$$

$$V_2 = \frac{1}{(p-1)q} \sum_{i=-k}^{k-1} \sum_{j=-l}^l (g(i,j) - g(i+1,j))^2$$

$$V_3 = \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (g(i,j) - g(i+1,j+1))^2$$

$$V_4 = \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (g(i,j+1) - g(i+1,j))^2$$

$$V = \min(V_1, V_2, V_3, V_4)$$

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

### B. Fluxo Ótico

Fluxo ótico é a distribuição de velocidades aparentes do movimento de padrões de brilho em uma imagem.[8] 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 americano James Gibson.[9] [10]

Um pixel tendo a localização  $(x, y, t)$  com intensidade  $I$  será movido em  $\Delta x, \Delta y$  and  $\Delta t$  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.

### C. TerraLib

Os algoritmos descritos acima são utilizados por duas classes do TerraLib: `TePDIMMIOMatching` utiliza o operador Moravec e `TePDIOFMatching` utiliza o fluxo ótico.

## VI. 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 mosaicaadas 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)

## VII. RESULTADOS

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

## VIII. COMPARAÇÃO DOS ALGORITMOS

Comparação do SIFT e dos algoritmos do TerraLib

## IX. CONCLUSÃO

Conclusão aqui.

## REFERÊNCIAS

- [1] M. A. P. Domiciano A. Canhoto, E. H. Shiguemori, "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] A. Vedaldi, "Sift++: A lightweight c++ implementation of sift," <http://www.vlfeat.org/~vedaldi/code/siftpp.html>, 2011.
- [5] *Computer Graphics: Principles and Practice 2nd Edition in C*, Addison-Wesley, 1995.
- [6] C. Schmid; R. Mohr; C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, 2000.
- [7] H. P. Moravec, "Towards automatic visual obstacle avoidance," *5th International Joint Conference on Artificial Intelligence*, 1977.
- [8] *The Perception of the Visual World*, Riverside Press, 1950.
- [9] J.J Gibson, "On the analysis of change in the optic array," *Scandinavian J. PsychoL* 18, 1977.
- [10] B. Horn; B. Schunck, "Determining optical flow," *Artificial Intelligence* 17, 1981.