

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 comparamos dois algoritmos de registro de imagens implementados na biblioteca TerraLib: Optical Flow e Operador Moravec Modificado. Através da comparação desses algoritmos implementamos uma solução híbrida para registrar e mosaicar imagens adquiridas por câmera digital a bordo de VANT de modo automático.

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

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 aplicações [1] e o objetivo geral deste trabalho é realizar registro e mosaico de imagens adquiridas por aeronaves não tripuladas. O mosaico de seqüência 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 buscou-se 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 foi realizada e comparada entre dois algoritmos: Optical Flow e Operador Moravec Modificado, ambos implementados na biblioteca TerraLib.



Figura 1-Exemplo de Plano de Voo do VANT

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 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 uma 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 construir uma imagem maior, permitindo assim uma visão global da cena.[3]

Neste trabalho a biblioteca utilizada para a realização do registro e mosaico foi a TerraLib, que 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 [4]. A utilização desta biblioteca se deu em todas as etapas necessárias para a realização do registro e mosaico das imagens, ou seja, na obtenção de pontos de controle, na transformação espacial da imagem e na sobreposição das imagens.

A. 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 obedecem a essa definição assim como pontos pretos em um fundo branco, o final de ramificações ou qualquer textura bidimensional significativa [5]. Ponto de controle são, portanto, os pontos de interesse em imagens de sensoriamento remoto que são utilizados para aplicar transformações geométricas afim de realizar o processo de registro entre as determinadas imagens.[6]

1) 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 [7]. Esse detector 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 à colunas e linhas. Se o mínimo dessas quatro diferenças é superior a um determinado limiar, então um ponto de interesse foi encontrado [5].

O operador Moravec pode ser definido matematicamente como:

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

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

Na biblioteca TerraLib, o método Moravec é implementado pela classe TePDIMMIOMatching.

2) Optical Flow: 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. Descontinuidades do fluxo ótico podem ajudar a segmentar uma determinada imagem em regiões que correspondem a diferentes objetos. Esse conceito começou a ser estudado na década de 1940 e foi publicado primeiramente pelo psicólogo James Gibson [9] [10].

Um pixel tendo a localização $(x; y; t)$ com intensidade I será movido em Δx , Δy e 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, o método Optical Flow ou é implementado pela classe TePDIOFMatching.

A. 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.

1) Correlação cruzada normalizada: O funcionamento do método estatístico de correlação cruzada normalizada 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 [3][11][12].

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

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. O TerraLib implementa esse método de casamento em suas classes.

B. 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 adequadamente geometricamente e espacialmente a fim 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. Abaixo temos as 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 [13].

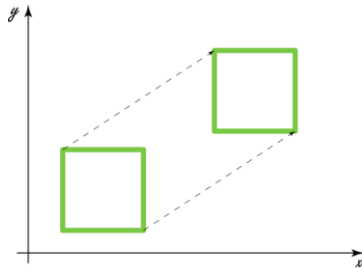


Figura 2-Exemplo de translação em um espaço bidimensional

2) Variação de escala: A variação de escala é o fato de se esticar ou encolher uma determinada figura em relação ao 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 [13].

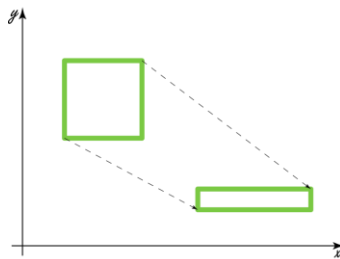


Figura 3-Exemplo de variação de escala em um espaço bidimensional

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 [13].

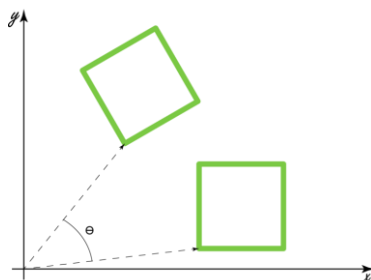


Figura 4-Exemplo de rotação em um espaço bidimensional

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.[13]

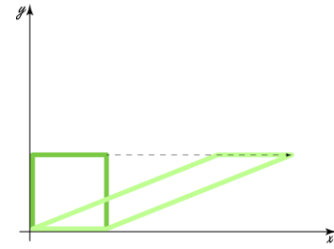


Figura 5- Exemplo de cisalhamento em um espaço bidimensional

5) Projeção: Projeção é o processo no qual se obtém uma figura bidimensional a partir de uma cena tridimensional.[13]

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

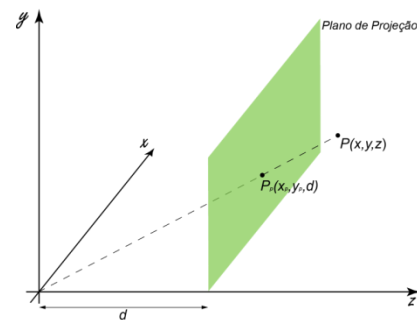


Figura 6- Exemplo de projeção

C. 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 [13].

2) Afim: A transformação afim é uma transformação que compreende as operações de uma 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 [13].

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 . [6]

4) Projeção: A projeção é nada mais que uma transformação linear em um espaço projetivo [13]. A forma geral da transformação projetiva é descrita como:

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

III. 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 .

Pode-se verificar o quão a transformação é correta através do cálculo do erro RMSE (*Root-Mean-Square Error*).

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

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.[3]

IV. METODOLOGIA

O sistema implementado é dividido em duas partes baseadas em algoritmos da biblioteca TerraLib, conforme mostra a Figura 7.

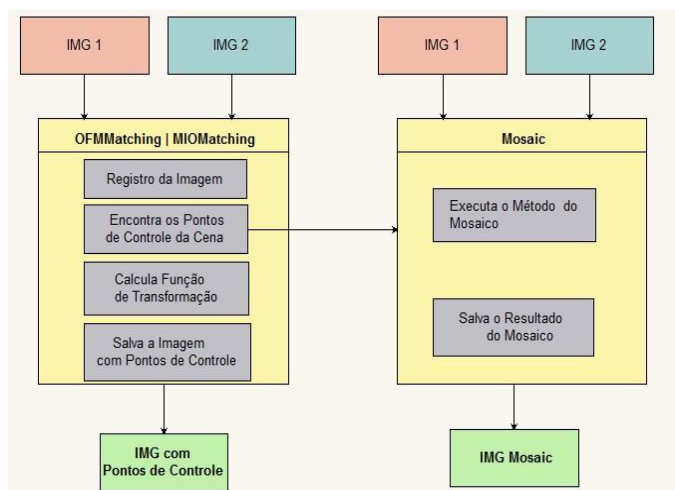


Figura 7- Estrutura Lógica do Sistema

A primeira parte se refere ao bloco MMIOMatching e OFMMatching, onde duas imagens são passadas como parâmetros de entrada e como resultado são apresentados os pontos de controle e a transformação geométrica entre as imagens. Essa parte do sistema primeiramente executa o método OFMMatching por ser mais rápido, e caso esse

método não encontre pontos de controle suficientes, então é executado o método MMIOMatching, que é mais lento, porém consegue encontrar mais pontos de controle. Também, durante a execução dessa primeira etapa, há ainda a opção de salvar os pontos de controle encontrados em duas novas imagens somente para conferência visual.

A Segunda parte se refere ao bloco Mosaic, que recebe como parâmetros de entrada a saída do primeiro bloco, ou seja, os vetores que representam os pontos de controle e a transformação geométrica. Além disso, são parâmetros de entrada as imagens que serão mosaicadas e o nome da nova imagem que apresentará o resultado do mosaico.

A. Configuração do Sistema

Foram realizados vários testes empíricos, e para alcançarmos resultados satisfatórios desenvolvemos um sistema híbrido utilizando os dois métodos de registro. O método MMIOMatching apresentou resultados melhores para os testes realizados, todavia é mais lento que o método OFMMatching. Este último método, foi o mais rápido para realização do registro, e em algumas cenas, embora apresentasse o valor de erro RMSE baixo, encontrava poucos pontos de controle, e por consequência o mosaico utilizando esses pontos apresentavam grandes distorções. Através dos testes realizados, chegou-se a seguinte configuração:

1) MMIOMatching

- Tipo de transformação geométrica: Afim.
- Estratégia de Remoção de pontos divergentes: Remoção aleatória.
- Valores máximos para os erros de mapeamento (direto e inverso): 5.
- Relação de resolução entre pixels: 1 para 1.
- Número máximo de pontos de controles a ser localizados: 4000 pontos.
- Tamanho da janela de correlação: 45x45 pixels.
- Tamanho da janela do Operador Moravec: 11x11 pixels.

2) OFMMatching

- Tipo de transformação geométrica: Afim.
- Estratégia de Remoção de pontos divergentes: Remoção aleatória.
- Valores máximos para os erros de mapeamento (direto e inverso): 5.
- Relação de resolução entre pixels: 1 para 1.
- Número máximo de pontos de controles a ser localizados: 529.
- Tamanho da janela de correlação: 45.
- Sensibilidade da correlação: 0.5.
- Máxima sensibilidade de detecção: 0.02.

3) Mosaic

- Método de mistura (*Blender*): Sem *blender*.
- Método de interpolação: Vizinhos mais próximos.
- Opção de auto-equalização (com ou sem): com equalização.

O sistema registra e mosaica as imagens de maneira automática, e para isso é apenas informado o caminho da pasta que contém as imagens a serem registradas e

mosaicadas. Cada mosaico gerado é salvo dentro dessa mesma pasta.

A heurística para a escolha do algoritmo de registro começa primeiramente executando o registro utilizando o método OFMatching por ser mais rápido. Ao final do registro é analisado o valor do erro RMSE e a quantidade de pontos encontrados. Se a quantidade de pontos e o valor do erro RMSE satisfazer um determinado limiar, então realiza-se o mosaico de duas imagens utilizando os pontos de controle e a transformação geométrica calculada. O processo é reiniciado até que acabe a lista de imagens selecionadas para a realização de registro e mosaico.

No entanto, se a quantidade de pontos de controle e o erro RMSE não atenderem ao limiar definido, então é executado o algoritmo MMIOMatching para encontrar os novos pontos de controle e em seguida o mosaico das duas imagens através dos novos valores calculados, e o processo é reiniciado.

V. RESULTADOS

As imagens analisadas foram coletadas por uma câmera digital abordo de um VANT, e apresentam muito ruído devido ao formato de compressão ser JPEG. Cada figura possui tamanho de 4000x3000 pixels. Os resultados foram conseguidos executando a implementação em um laptop com processador Intel Core i5 de 2,4 GHz com dois núcleos de processamento e 4 GB de memória RAM 1333 MHz DDR3.

As Figuras 8 e 9 apresentam o registro e mosaico de duas imagens utilizando o algoritmo OFMatching.

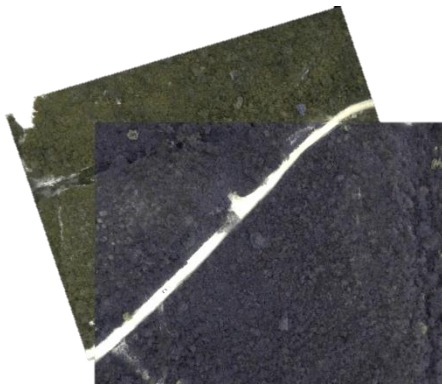


Figura 8 - Mosaico de duas imagens utilizando OFMatching

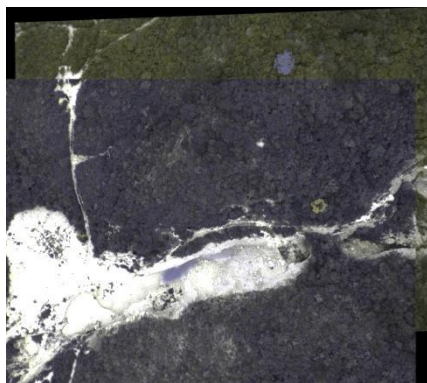


Figura 9 - Mosaico de duas imagens utilizando OFMatching

As Figuras 10 e 11 apresentam o registro e mosaico das mesmas imagens utilizadas para a composição da Figura 8 e 9, todavia utilizando o algoritmo MMIOMatching.

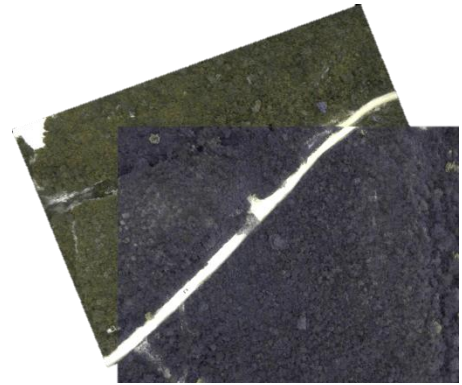


Figura 10 - Mosaico de duas imagens utilizando MMIOMatching

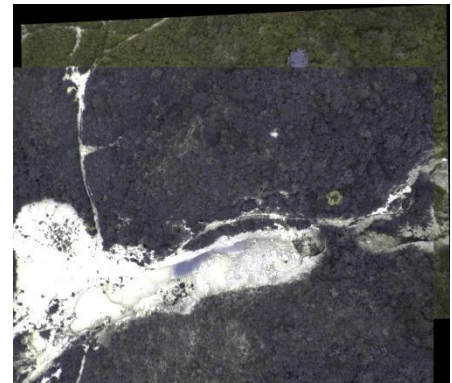


Figura 11 - Mosaico de duas imagens utilizando MMIOMatching

A Tabela 1 apresenta a quantidade de pontos de controle, o erro RMSE de mapeamento direto e mapeamento inverso e o tempo de processamento para os registros apresentados nas Figuras 8, 9, 10 e 11.

Tabela 1 – Resultados de Processamento

Fig	Algoritmo	Pontos de Controle	RMSE direto	RMSE inverso	Tempo (s)
8	OFMatching	6	2,32	2,31	44
9	OFMatching	8	1,49	1,53	42
10	MMIOMatching	55	2,91	3,04	305
11	MMIOMatching	245	2,96	2,96	329

Observando as Figuras 8, 9, 10 e 11, percebemos que há pouca diferença entre os resultados, porém analisando a Tabela 1 é notável que o algoritmo OFMatching é muito mais rápido se comparado com o algoritmo MMIOMatching. Dessa forma, através de várias análises empíricas, implementamos um sistema para registro automático híbrido utilizando esses dois algoritmos, e conseguindo assim diminuir o tempo de processamento necessário para realizar registro e mosaico de sequências de imagens aéreas.

A Figura 12 apresenta o registro e mosaico de 14 figuras que foram selecionadas para compor os testes realizados neste trabalho por apresentarem diversas rotações devido serem captadas durante o momento em que o VANT efetua curvas para seguir o plano de voo. O registro e mosaico dessas imagens foi conseguido de forma automática através do sistema implementado. O processamento para registrar e mosaicar as imagens levou 3,29 horas para ser concluído.

Somente com o algoritmo OFMatching, não conseguimos realizar mosaico de sequências de imagens, isso devido ao reduzido número de pontos de controle encontrados. Realizando registro somente com o algoritmo

MMIOMatching, o tempo de processamento para registrar e mosaicar as 14 imagens foi de 3,59 horas para ser concluído.



Figura 12 - Mosaico de 14 imagens utilizando solução híbrida

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] L. M. G. Fonseca, "Registro automático de imagens de sensoriamento remoto baseado em multiplas resoluções," Ph.D. dissertation, Instituto Nacional de Pesquisas Espaciais, Sao Jose dos Campos, 1999-02-26 1999.
- [3] D. Fedorov, "Sistema semi-automático de registro e mosaico de imagens," Master's thesis, INPE, 2002.
- [4] DPI/INPE, "Terralib website," <http://www.terralib.org/>, 2012.
- [5] C. Schmid; R. Mohr; C. Bauckhage, "Evaluation of interest point detectors," International Journal of Computer Vision, 2000.
- [6] Remote Sensing: Models and Methods for Image Processing. Elsevier Inc, 2007.
- [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. 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.
- [11] Fonseca L. M. G; Manjunath B. S., "Registration techniques for multisensor remotely sensed imagery," PE&RS, 1996.
- [12] F. Zhao; Q. Huang; W. Gao, "Image matching by normalized cross-correlation," ICASSP, 2006.
- [13] Computer Graphics: Principles and Practice 2nd Edition in C. Addison-Wesley, 1995.

VI. CONCLUSÃO

No trabalho foram apresentados os resultados obtidos com a comparação dos algoritmos de registro de imagens do TerraLib: *Optical Flow* e detector Moravec modificado. Os testes apresentaram resultados satisfatórios, onde o algoritmo *Optical Flow* foi mais eficiente em imagens que apresentavam pouca rotação, e o algoritmo utilizando detector Moravec apresentou melhores resultados nas situações onde o VANT realizou curvas para seguir o plano de voo e capturou imagens muito sobrepostas e com rotação em ângulos desconhecidos.

Devido ao fato de obter um maior conjunto de pontos de controle e pelo próprio método de operação, o detector Moravec é cerca de dez vezes mais demorado em relação ao *Optical Flow*, no entanto na maioria dos casos se mostrou mais preciso, encontrando mais pontos de controle, o que garantiu mosaicos aceitáveis.

A implementação e os testes gerados através desse trabalho poderão ser utilizados para realizar o processo de registro e mosaico de imagens áreas de VANTs de uma maneira mais eficiente e com um produto final com menos distorções. Há ainda a vantagem de a biblioteca TerraLib ser de código aberto e *software* livre, podendo assim, facilmente se reproduzir os resultados obtidos neste trabalho.

Embora os resultados sejam aceitáveis, ainda temos o problema do tempo de processamento gasto para registrar e mosaicar as imagens ser muito alto. Conseguimos minimizar esse problema através da implementação do sistema híbrido, porém o tempo de processamento ainda continua elevado. Para trabalhos futuros seria importante comparar e testar outros métodos para registro, tais como os algoritmos SIFT e SURF e buscar alguma heurística para minimizar ainda mais o tempo de processamento.