

Trabalho 05 de Introdução ao Processamento de Imagem Digital

MARIANNA DE PINHO SEVERO

RA: 264960

I. INTRODUÇÃO

O processamento de imagens é uma importante atividade desempenhada dentro da área de Visão Computacional, sendo responsável pela captura, representação e transformação de imagens. Ele permite a extração de informações e a transformação de certas características de imagens, de maneira a facilitar tanto a percepção humana como a interpretação realizada por computadores [1].

Com respeito à transformação de imagens, diversas técnicas podem ser utilizadas, de acordo tanto com as necessidades do domínio de aplicação em que as imagens serão empregadas, como com os objetivos das transformações.

Neste trabalho, empregamos técnicas de processamento de imagens para o propósito de Registro de Imagens. Mais especificamente, aplicamos uma série de operações sobre as imagens de maneira a obtermos pontos de interesse, seus descritores e matrizes de homografia com o objetivo de criarmos imagens panorâmicas a partir de pares de imagens.

O registro de imagens é uma técnica de processamento de imagens que tem como objetivo estabelecer a correspondência e alinhar imagens obtidas em diferentes intervalos de tempo, a partir de diferentes dispositivos de aquisição ou sob perspectivas distintas. Ele tem sido aplicado em diversas áreas, como o sensoriamento remoto e a área médica, para tarefas como a detecção de mudanças em imagens, a integração de fotografias com mapas, a fusão de dados obtidos a partir de diferentes técnicas de aquisição e a recuperação de informações.

Na Seção 2 são apresentadas as bibliotecas utilizadas neste trabalho; na Seção 3 são descritas a saída e a entrada de dados; na Seção 4 apresenta-se as técnicas empregadas para a detecção de pontos de interesse, para o cálculo dos respectivos descritores, das matrizes de homografia, e para a construção das imagens panorâmicas; na Seção 5 são apresentados os resultados e discussões; por fim, na Seção 6, apresenta-se a conclusão.

II. DEPENDÊNCIAS

Para a implementação dos algoritmos utilizados neste trabalho, as seguintes bibliotecas foram empregadas:

- numpy: usada para a manipulação dos *arrays* que representam as imagens e para a criação de estruturas de dados auxiliares.
- opencv: utilizada para a leitura e escrita de imagens e para a aplicação das técnicas de registro.
- matplotlib: empregada para a apresentação das imagens.

Além dessas bibliotecas, foram empregadas funções para a apresentação (**plotCorrespondingLines**) e para o armazenamento (**saveCorrespondingLines**) das imagens, baseadas nos códigos disponibilizados em **Feature Matching**.

III. ENTRADA E SAÍDA DE DADOS

As imagens utilizadas para os testes neste trabalho foram retiradas do [Site da Disciplina](#) e estão no formato JPEG (*Joint Photographic Experts Group*). Elas foram armazenadas em um diretório chamado **input_images**.

Para utilizá-las, elas foram carregadas utilizando o trecho de código 1, em que o primeiro parâmetro indica o caminho da imagem e o segundo informa o formato para o qual queremos carregá-las. Neste caso, como não utilizamos o segundo parâmetro, seu valor padrão é usado, e carregamos imagens no formato BGR. A função empregada retorna um *array* numpy de três dimensões, em que uma delas representa a largura, a outra representa a altura do *array*, e a terceira dimensão representa os três canais que compõem uma imagem BGR. Assim, cada elemento em um determinado canal representa a intensidade do *pixel* da imagem naquele canal.

```
image = cv.imread('input_images/name_image.jpg')
```

Código 1: Carregar uma imagem com OpenCV.

Por sua vez, as imagens geradas – que também são representadas por *arrays* numpy no programa – são armazenadas, também no formato JPEG, em subdiretórios do diretório **output_images**, de acordo com a operação de cálculo de descritores utilizada, e do tipo de informação que representam. As imagens panorâmicas resultantes de processamentos que empregaram o método ORB (*Oriented FAST and Rotated BRIEF*) são armazenadas no caminho **output_images/orb/panoramic/** e aquelas resultantes de processamentos que empregaram o BRIEF (*Binary Robust Independent Elementary Features*) são salvas em **output_images/brief/panoramic/**. Também, as imagens que mostram as linhas de correspondência entre os pares de imagens foram salvas em **output_images/orb/lines/** e **output_images/brief/lines/**.

Para salvar as imagens, utilizamos o trecho de código 2, em que o primeiro argumento é o caminho onde salvaremos a imagem e o segundo é a estrutura de dados que a representa. Ademais, os nomes das imagens panorâmicas são compostos dos seguintes campos: o algoritmo de cálculo dos descritores, o nome da primeira imagem, o nome da segunda imagem, o limiar de correspondência e o limiar

de erro. As imagens que apresentam as linhas de correspondência possuem quase todos esses campos, exceto o de limiar de erro. Um exemplo de nome de imagem panorâmica é **orb_lake1_lake2_0.4_2.5.jpg**, e de imagem de correspondências é **brief_lake1_lake2_0.5.jpg**.

```
cv.imwrite('output_images/subdir/name_image.jpg',
           image)
```

Código 2: Salvar uma imagem com OpenCV.

Os algoritmos implementados neste trabalho foram escritos utilizando-se a plataforma **Jupyter Notebook**. Dessa forma, o arquivo onde os algoritmos foram escritos possui a extensão **.ipynb**.

IV. QUESTÕES E SOLUÇÕES

Neste trabalho, foi pedido que criássemos imagens panorâmicas a partir de pares de imagens diferentes que representam o mesmo local. Nas próximas subseções, descrevemos os passos realizados para atingir esses objetivos.

A. Conversão das imagens para níveis de cinza.

O primeiro passo realizado neste trabalho foi a conversão das imagens para níveis de cinza. Isso foi feito de maneira a facilitar o emprego dos algoritmos de detecção de pontos de interesse e de cálculo de descritores, que utilizam os níveis de cinza dos *pixels* para o cálculo dessas informações.

Primeiramente, as imagens foram carregadas para um dicionário chamado **images**, cujas chaves receberam os seguintes valores, de acordo com os pares de imagens: 'building1', 'building2', 'lake1', 'lake2', 'road1', 'road2', 'desertroad1', 'desertroad2', 'field1', 'field2'.

Então, empregamos o método **cvtColor** da biblioteca *opencv* para transformá-las em imagens monocromáticas, armazenando-as em um novo dicionário, chamado **gray_images**, com as mesmas chaves do dicionário *images*.

B. Detecção de pontos de interesse.

O segundo passo para a realização do registro de imagens foi a identificação de pontos de interesse nos pares de imagens a serem registradas. Pontos de interesses são características da imagem que nos permitem identificar unicamente determinadas regiões, como cantos, junções e bordas.

Para identificá-los, utilizamos o método **detect** da classe ORB, implementada na biblioteca *opencv*. Este método emprega uma versão modificada da abordagem FAST (*Features from Accelerated and Segments Test*), que melhora o desempenho desta com relação à escala e a rotação de imagens.

Um detalhe importante é que tentamos empregar o método STAR para a detecção de pontos de interesse para o BRIEF, conforme sugerido no [Tutorial do Opencv](#), mas não conseguimos obtê-los para as imagens armazenadas nas chaves 'field1' e 'field2' do dicionário *gray_images*. Dessa maneira, para obtermos pontos de interesse para todas as imagens, independentemente da metodologia de obtenção de descritores empregada, utilizamos o método *detect* do ORB.

C. Cálculo de descritores.

Descritores são estruturas que armazemam informações sobre os pontos de interesse, como, por exemplo, área, assinatura, energia de deformação e muitas outras [1]. Além disso, bons descritores devem ser invariantes a transformações geométricas, tais como escala, translação e rotação.

Neste trabalho, empregamos dois algoritmos para o cálculo dos descritores dos pontos de interesse, o BRIEF e o ORB, ambos aplicados através do método **compute** de suas respectivas classes.

O BRIEF descreve os pontos de interesse através de *strings* binárias, cujos tamanhos podem variar entre 128, 256 ou 512 bits. Para obter essas *strings*, o BRIEF seleciona, de acordo com uma determinada distribuição, pares de *pixels* dentro de uma região próxima ao ponto de interesse que ele deseja descrever e atribui valores 1 ou 0 para cada par, de acordo com o resultado da comparação entre as intensidades de seus *pixels*. Ademais, a quantidade de pares escolhidos depende do tamanho desejado da *string*.

O ORB, por sua vez, utiliza uma versão modificada do BRIEF, que busca melhorar o desempenho deste com relação à rotação, mantendo uma alta variância e baixa correlação entre os testes realizados para a obtenção dos valores binários.

D. Cálculo das correspondências.

Para o cálculo das correspondências entre os descritores dos pares de imagens, adotamos duas etapas:

Primeiro, utilizamos o algoritmo BFMatcher (*Brute Force Matcher*), que compara os descritores de uma imagem com todos os descritores da outra imagem e retorna apenas os *matches* – pares de pontos de interesse – que possuem a menor distância. Como utilizamos métodos que geram descritores que são *strings* binárias, para calcularmos as distâncias entre os descritores escolhemos a *Distância de Hamming*, como sugerida em [Feature Matching](#). Mais um detalhe importante sobre esse algoritmo é que ele possui um parâmetro booleano, chamado **crossCheck**, que quando verdadeiro faz com que apenas os *matches* nos quais ambos os descritores são o mais próximo um do outro sejam retornados. Neste trabalho, atribuímos valor verdadeiro a esse parâmetro.

A segunda etapa foi a seleção dos *matches*, dentre aqueles retornados pelo BFMatcher, cujos valores de correspondência eram maiores do que um determinado limiar. Para calcularmos as correspondências, normalizamos as distâncias de todos os *matches* pelo maior valor encontrado entre eles – já que não sabemos qual o maior valor de distância possível entre dois descritores – e subtraímos de 1 essas distâncias normalizadas. Assim, quanto menor a distância entre dois descritores, maior é sua correspondência. Para observarmos a influência do limiar de correspondência sobre o processo de registro, selecionamos *matches* para diferentes valores de limiar, os quais foram {0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9}.

E. Cálculo de matrizes de homografia.

Uma matriz de homografia nos permite realizar o mapeamento entre um conjunto de variáveis que está em um

determinado plano, para outro conjunto que está em outro plano. Em outras palavras, dadas duas imagens, a homografia representa o conjunto de transformações que devemos aplicar às coordenadas dos *pixels* em uma das imagens para que cheguemos às coordenadas dos *pixels* correspondentes na outra imagem. Essas transformações causam a deformação da imagem original, cujos *pixels* são a entrada do sistema de transformação.

Neste trabalho, a matriz de homografia nos permite mapear os pontos de interesse em uma imagem para os pontos de interesse correspondentes na outra imagem, de acordo com os *matches* explicados da subseção anterior. Para calculá-la, utilizamos o método **findHomography** da biblioteca *opencv*, que recebe como parâmetros os pontos de interesse das duas imagens, a técnica a ser empregada para o cálculo da matriz e um limiar de erro.

A técnica empregada para o cálculo das matrizes foi a RANSAC (*RANdom SAmple Consensus*), a qual obtém uma matriz através de um processo iterativo, de modo que, ao final, uma determinada quantidade de pontos considerados *inliers* seja obtida, de acordo com um critério de erro. Pontos são considerados *inliers* se suas distâncias aos valores obtidos pelo modelo criado pelo RANSAC forem menores do que um determinado limiar de erro. Para analisarmos a influência desse limiar sobre o registro, calculamos matrizes de homografia para diferentes de valores de erro, os quais foram $\{1; 1,5; 2; 2,5; 3; 3,5; 4; 4,5; 5; 5,5; 6; 6,5; 7; 7,5; 8; 8,5; 9; 9,5; 10\}$.

Um detalhe importante é que, como estamos calculando a homografia, precisamos de pelo menos quatro pontos correspondentes nas duas imagens para que consigamos obter as matrizes. Dessa maneira, as matrizes só foram calculadas para os pares de imagens cujas quantidades de *matches* eram maiores ou iguais a quatro.

F. Alinhamento e construção de imagens panorâmicas.

A etapa final do processo de registro de imagens foi a aplicação das transformações representadas pelas matrizes de homografia e a junção das imagens transformadas àquelas mantidas intactas, de maneira a formar as imagens panorâmicas.

Para a aplicação das transformações, utilizamos o método **warpPerspective** da biblioteca *opencv*. Ele recebe como entradas a imagem que queremos transformar, a matriz de homografia e o tamanho que a imagem de saída deve ter. Neste trabalho, escolhemos como tamanho da imagem de saída a soma dos tamanhos das duas imagens que compõem um par.

Para a formação da imagem panorâmica, substituímos, na imagem resultante da transformação de perspectiva, a imagem que permaneceu intacta. Dessa maneira, os pontos correspondentes em ambas as imagens são sobescritos e as regiões diferentes nas duas imagens são mantidas, formando a ideia de imagem panorâmica desejada.

Por fim, é importante destacar que, ao longo de todas as operações descritas nessas subseções, a ordem das imagens é importante para alcançarmos os resultados desejados.

V. RESULTADOS E DISCUSSÕES

Nos próximos parágrafos, apresentamos os testes realizados e os resultados obtidos para cada abordagem utilizada para o registro de imagens.

Na Figura 1, podemos observar os pares de imagens utilizados para os testes. Ao todo foram geradas 1311 imagens panorâmicas, sendo 665 delas obtidas ao utilizarmos o BRIEF e 646 ao empregarmos o ORB. Essa quantidade de imagens é resultante do número de algoritmos de cálculo de descritores (2), da quantidade de pares de imagens (5), de limiares de correspondência (8) e de limiares de erro (19), menos a quantidade de imagens que não foram geradas por não restarem correspondências (*matches*) acima de determinados limiares de correspondência.

A. Pontos de interesse, descritores e matches.

Nas Figuras 2, 3 e 4, podemos observar as linhas de correspondência entre alguns pares de imagens, para diferentes limiares de correspondência e algoritmos de cálculo de descritores.

Podemos observar que, conforme aumentamos o limiar de correspondência, menor se torna a quantidade de linhas, o que é esperado já que estamos selecionando correspondências cada vez maiores. Também observamos que, para algumas imagens, como as da Figura 2, o BRIEF mantém mais *matches* do que o ORB, e para outras, como as da Figura 3, acontece o contrário. Isso demonstra a influência dos diferentes métodos de construção de descritores na correspondência dos pontos de interesse em imagens com determinadas características.

Também podemos observar que, dependendo das diferenças entre as duas imagens que formam um par, um método de descrição pode ter desempenho melhor do que o outro. Podemos observar um exemplo na Figura 3, em que as linhas de correspondência obtidas ao aplicarmos o BRIEF ligam diversos pontos de interesse que não são realmente correspondentes, mas do que acontece ao aplicarmos o ORB. Uma das possíveis motivações é o fato de o BRIEF não apresentar grande invariância com relação à rotação – o que é tratado pelo ORB –, e as duas imagens diferirem por uma rotação considerável.

Ademais, podemos perceber que, para certas imagens, obtemos desempenhos semelhantes ao aplicarmos qualquer um dos dois algoritmos de cálculo de descritores, e o BRIEF pode inclusive apresentar correspondências melhores, como pode ser visto na Figura 4, em que o ORB apresenta algumas linhas que ligam pontos que não são realmente correspondentes.

B. Erro fixo e diferentes limiares de correspondência.

Nas Figuras 5, 6, 7, 8 e 9, podemos observar as imagens panorâmicas geradas a partir dos cinco pares de imagens, para diferentes limiares de correspondência, para erro igual a 5, e para descritores gerados tanto pelo BRIEF como pelo ORB. É importante destacar que as imagens representadas nessas figuras foram cortadas, para melhorar sua visualização no relatório.



(a) Building1.



(b) Building2.



(c) Lake1.



(d) Lake2.



(e) Road1.



(f) Road2.



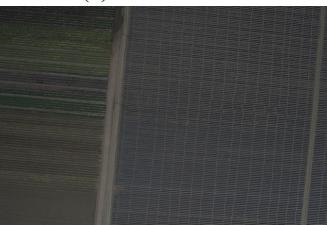
(g) DesertRoad1.



(h) DesertRoad2.



(i) Field1.



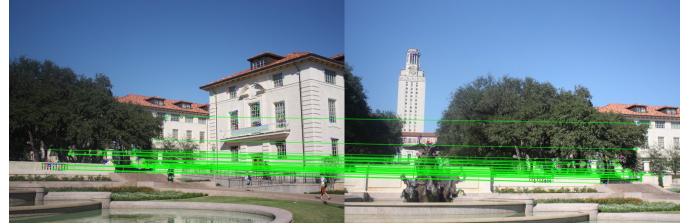
(j) Field2.

Figura 1: Pares de imagens originais.

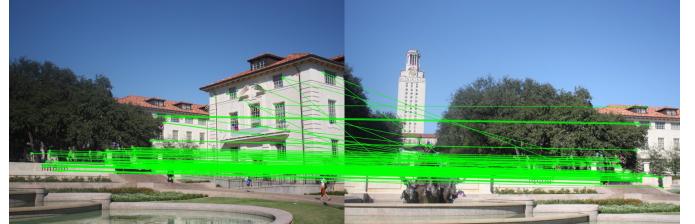
Podemos observar que os resultados, ao empregarmos o ORB, pioraram ao aumentarmos o limiar de correspondência. Já os resultados, ao utilizarmos o BRIEF, melhoraram em algumas regiões das imagens, enquanto outras pioraram. Também, na Figura 5, ao empregarmos um limiar de correspondência de 0,9 para o BRIEF e 0,8 para o ORB, ambos produziram resultados ruins. Isso pode ser decorrência do pequeno número



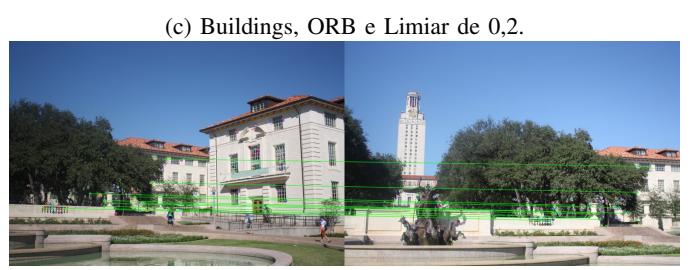
(a) Buildings, BRIEF e Correspondência 0,2.



(b) Buildings, BRIEF e Correspondência 0,8.



(c) Buildings, ORB e Limiar de 0,2.



(d) Buildings, ORB e Correspondência 0,8.

Figura 2: Linhas de correspondência para as imagens Building1 e Building2.

de *matches* que foram mantidos.

Além disso, ao empregarmos o BRIEF, não conseguimos resultados bons para as imagens panorâmicas geradas a partir das Figuras 1g e 1h. Provavelmente por causa da grande quantidade de correspondências erradas que comentamos na subseção anterior. Também, tanto ao usarmos o BRIEF ou ORB, não conseguimos gerar imagens panorâmicas, a partir de alguns pares de imagens, para alguns limiares de correspondência, pois não restaram *matches* com correspondências acima desses limiares.

Para as imagens panorâmicas geradas a partir das Figuras 1c e 1d, o aumento do limiar de correspondência não fez grande diferença nos resultados e estes foram bons. Já para as imagens panorâmicas geradas a partir das Figuras 1e e 1f, o aumento do limiar de correspondência melhorou os resultados. É importante destacar que julgamos como bons resultados aquelas imagens em que as figuras que as compõem ficaram bem alinhadas e conseguimos perceber pouco ou nenhum detalhe



(a) DesertRoads, BRIEF e Correspondência 0,2.



(b) DesertRoads, BRIEF e Correspondência 0,6.



(c) DesertRoads, ORB e Correspondência 0,2.



(d) DesertRoads, ORB e Correspondência 0,6.

Figura 3: Linhas de correspondência para as imagens DesertRoad1 e DesertRoad2.

que mostre que duas imagens foram unidas.

Ademais, acreditamos que os resultados ruins decorrentes de certos limiares de correspondência estejam relacionados à diminuição do número de *matches* entre as imagens. Dessa forma, para aqueles pares de imagens que continuaram com uma quantidade considerável de correspondências após a aplicação do limiar, os resultados mostraram-se bons. É o que observamos, por exemplo nas Figuras 8 e 9.



(a) Lakes, BRIEF e Correspondência 0,2.



(b) Lakes, BRIEF e Correspondência 0,9.



(c) Lakes, ORB e Correspondência 0,2.



(d) Lakes, ORB e Correspondência 0,9.

Figura 4: Linhas de correspondência para as imagens Lake1 e Lake2.

Por fim, destacamos que observamos esses comportamentos para erros de 1, 5 e 10 e escolhemos mostrar apenas as imagens para o erro 5 para não sobrecarregar de imagens o relatório.

C. Correspondência fixa e diferentes limiares de erro.

Nas Figuras 10, 11, 12, 13 e 14, podemos observar as imagens panorâmicas geradas, para diferentes limiares de erro do algoritmo RANSAC e para uma correspondência de 0,5, ao empregarmos o BRIEF e o ORB. Novamente, as imagens foram cortadas para melhorarmos a visualização no relatório.



Figura 5: Imagens panorâmicas Buildings, para BRIEF e ORB, diferentes limiares de correspondência e erro fixo.

Para a Figura 10, em que utilizamos uma correspondência de 0,5, podemos perceber que os resultados do BRIEF melhoram ao aumentarmos o limiar de erro até 3 e, a partir desse valor, não há grandes mudanças. Por sua vez, os resultados do ORB melhoram até um erro de 4 e, então, pioram ao aumentarmos mais o limiar.

Ao observarmos a Figura 11, vemos que novamente não conseguimos bons resultados ao empregarmos o BRIEF. Ou seja, não houve um valor de limiar de erro que melhorasse os resultados. Já, ao empregarmos o ORB, percebemos que, conforme aumentamos o limiar de erro, os resultados melhoram.

Para tentarmos explicar esse comportamento, devemos lembrar que o RANSAC cria um modelo que gera valores que, ao serem comparados aos dados reais, permitem a classificação desses dados como *inliers* ou *outliers*, e o modelo final deve ser construído de modo a termos uma determinada quantidade mínima de *inliers*. Também, devemos lembrar que a quanti-

dade de correspondências que possuem valores maiores do que 0,5, ao empregarmos o ORB, na Figura 11 é bem maior do que aquela na Figura 10. Logo, podemos relacionar a quantidade de correspondências ao desempenho do RANSAC em construir as matrizes de homografia. Especificamente, podemos pensar que um grande número de correspondências aumenta a probabilidade de termos a quantidade mínima de *inliers* e gerar um bom modelo. Ao passo que, quando temos uma pequena quantidade de correspondências, ainda tentaremos obter esse número mínimo de *inliers*, mas o modelo gerado pode não ser tão bom, ocorrendo uma espécie de *overfitting*.

Já ao observarmos a Figura 12, vemos que aumentar o limiar de erro piorou os resultados, tanto ao empregarmos o BRIEF como o ORB. Por fim, vemos que o aumento do limiar de erro não alterou muito os resultados mostrados nas Figuras 13 e 14, e eles foram bons. É importante lembrarmos que estas foram as figuras que também apresentaram bons resultados



Figura 6: Imagens panorâmicas DesertRoads, para BRIEF e ORB, diferentes limiares de correspondência e erro fixo.

nas análises realizadas em seções anteriores, o que demonstra que o BRIEF e o ORB são bons algoritmos para cálculo de descritores para imagens com características como as dessas.

VI. CONCLUSÃO

Conforme pôde ser observado, conseguimos fazer o que foi pedido no Trabalho 05. Foram aplicados algoritmos para a identificação de pontos de interesse, para o cálculo de descritores, para o estabelecimento de correspondências entre pares de imagens, além da construção de matrizes de homografia, a

criação de imagens panorâmicas e a apresentação das linhas de correspondências entre os pares.

Também, podemos perceber que os algoritmos e limiares que geram os melhores resultados dependem das características de cada imagem, de maneira que determinadas configurações de parâmetros funcionam melhor para algumas imagens e para outras não.

Por fim, também notamos que existem relações entre a quantidade de *matches* obtidos entre pares de imagens e os limiares de erro empregados no RANSAC, contribuindo para

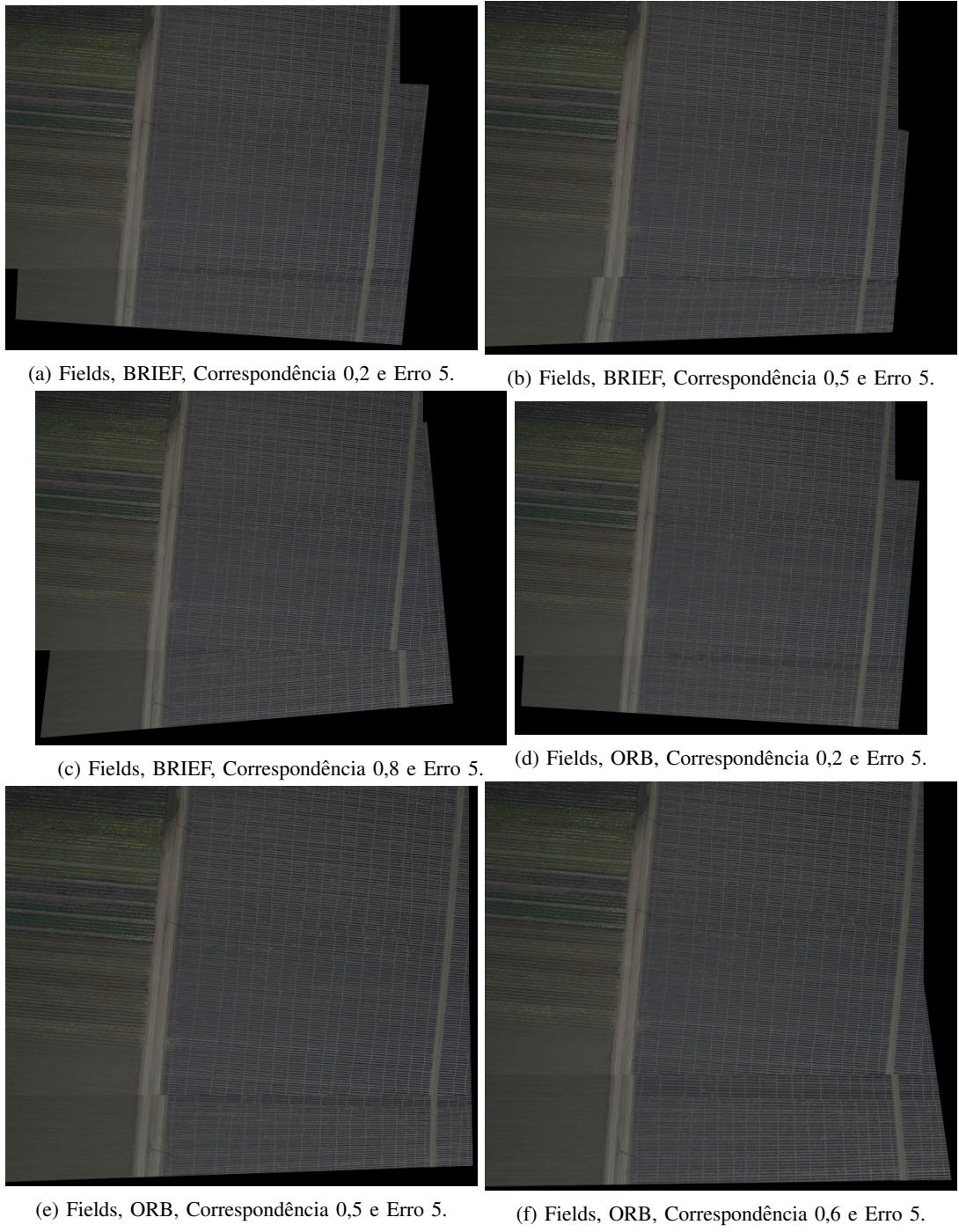


Figura 7: Imagens panorâmicas Fields, para BRIEF e ORB, diferentes limiares de correspondência e erro fixo.

a geração de melhores ou piores matrizes de homografia e, consequentemente, de imagens panorâmicas com maior ou menor qualidade.

REFERÊNCIAS

- [1] H. Pedrini and W. R. Schwartz, *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, 2008. [1](#), [2](#)



(a) Lakes, BRIEF, Correspondência 0,2 e Erro 5.



(b) Lakes, BRIEF, Correspondência 0,5 e Erro 5.



(c) Lakes, BRIEF, Correspondência 0,9 e Erro 5.



(d) Lakes, ORB, Correspondência 0,2 e Erro 5.



(e) Lakes, ORB, Correspondência 0,5 e Erro 5.



(f) Lakes, ORB, Correspondência 0,9 e Erro 5.

Figura 8: Imagens panorâmicas Lakes, para BRIEF e ORB, diferentes limiares de correspondência e erro fixo.



(a) Roads, BRIEF, Correspondência 0,2 e Erro 5.



(b) Roads, BRIEF, Correspondência 0,5 e Erro 5.



(c) Roads, BRIEF, Correspondência 0,9 e Erro 5.



(d) Roads, ORB, Correspondência 0,2 e Erro 5.



(e) Roads, ORB, Correspondência 0,5 e Erro 5.



(f) Roads, ORB, Correspondência 0,8 e Erro 5.

Figura 9: Imagens panorâmicas Roads, para BRIEF e ORB, diferentes limiares de correspondência e erro fixo.



Figura 10: Imagens panorâmicas Buildings, para BRIEF e ORB, diferentes limiares de erro e correspondência fixa.

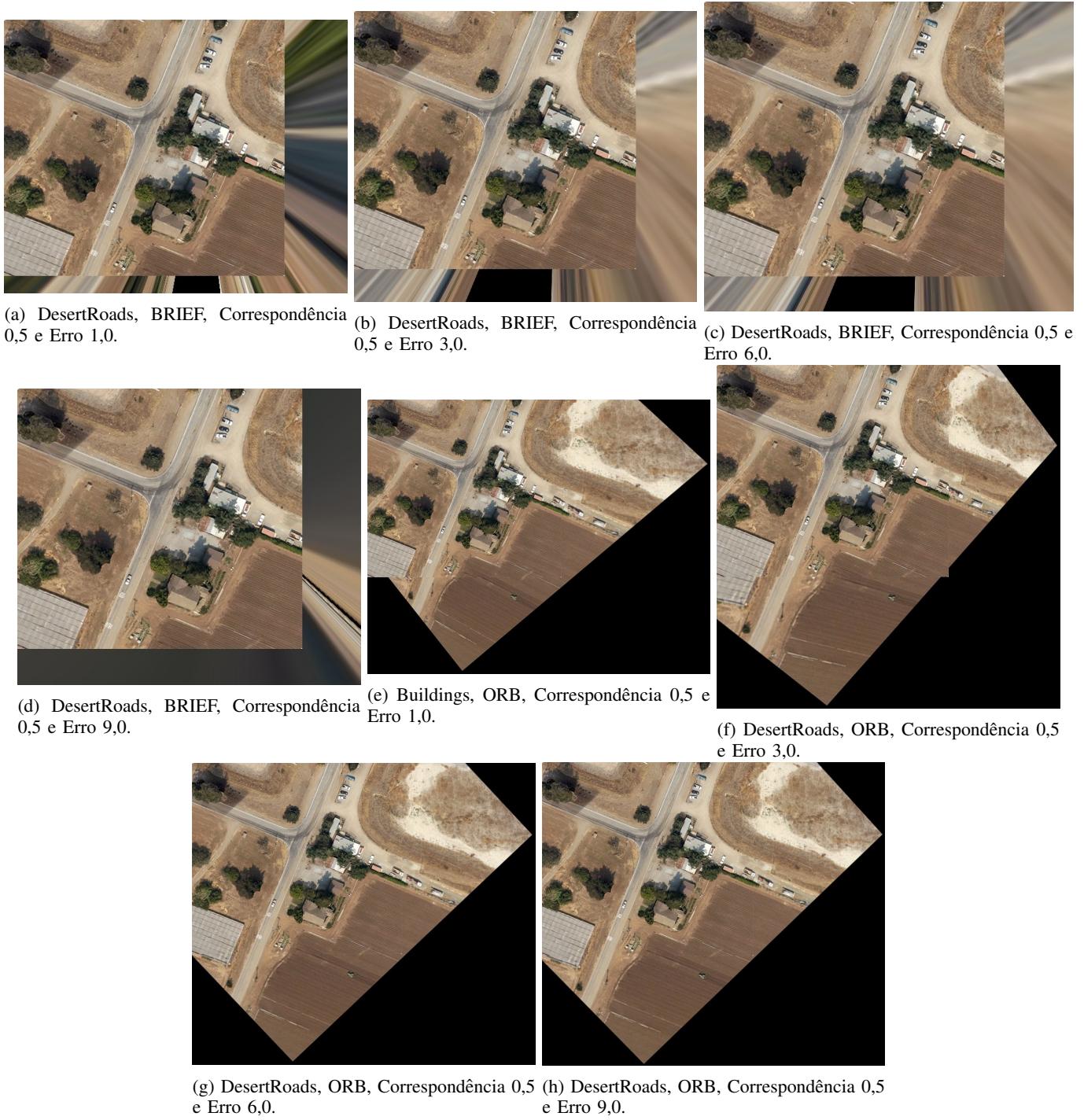
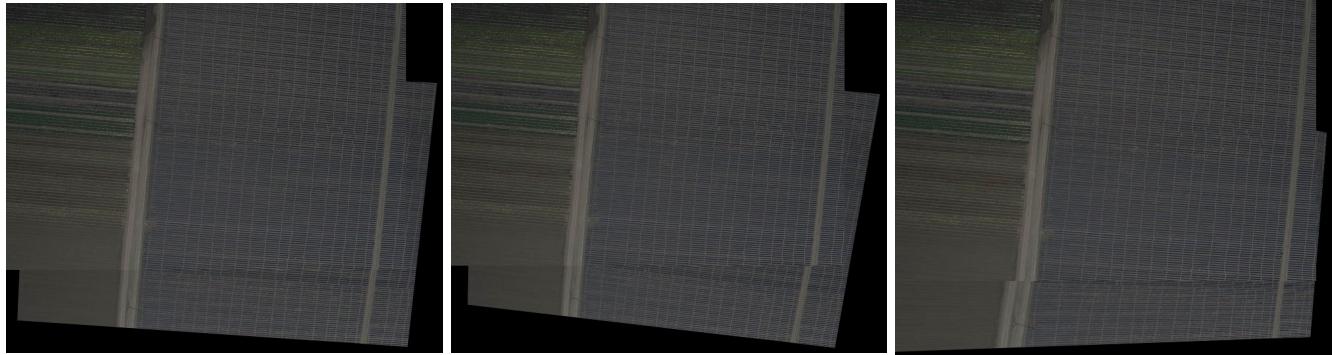


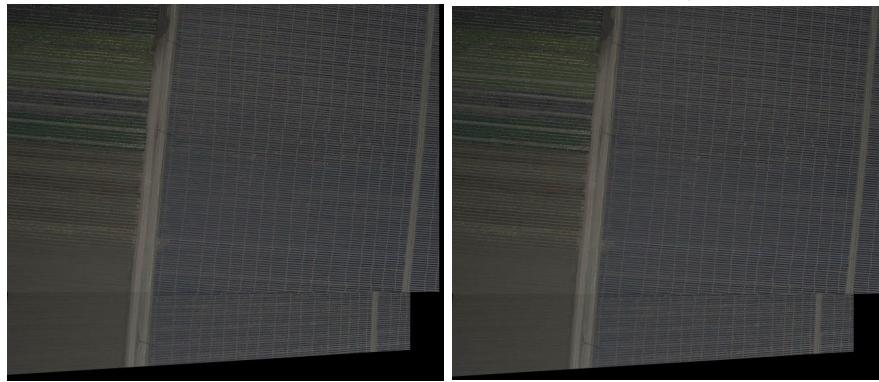
Figura 11: Imagens panorâmicas DesertRoads, para BRIEF e ORB, diferentes limiares de erro e correspondência fixa.



(a) Fields, BRIEF, Correspondência 0,5 e Erro 1,0.
(b) Fields, BRIEF, Correspondência 0,5 e Erro 3,0.
(c) Fields, BRIEF, Correspondência 0,5 e Erro 6,0.



(d) Fields, BRIEF, Correspondência 0,5 e Erro 9,0.
(e) Fields, ORB, Correspondência 0,5 e Erro 1,0.
(f) Fields, ORB, Correspondência 0,5 e Erro 3,0.



(g) Fields, ORB, Correspondência 0,5 e Erro 6,0.
(h) Fields, ORB, Correspondência 0,5 e Erro 9,0.

Figura 12: Imagens panorâmicas Fields, para BRIEF e ORB, diferentes limiares de erro e correspondência fixa.

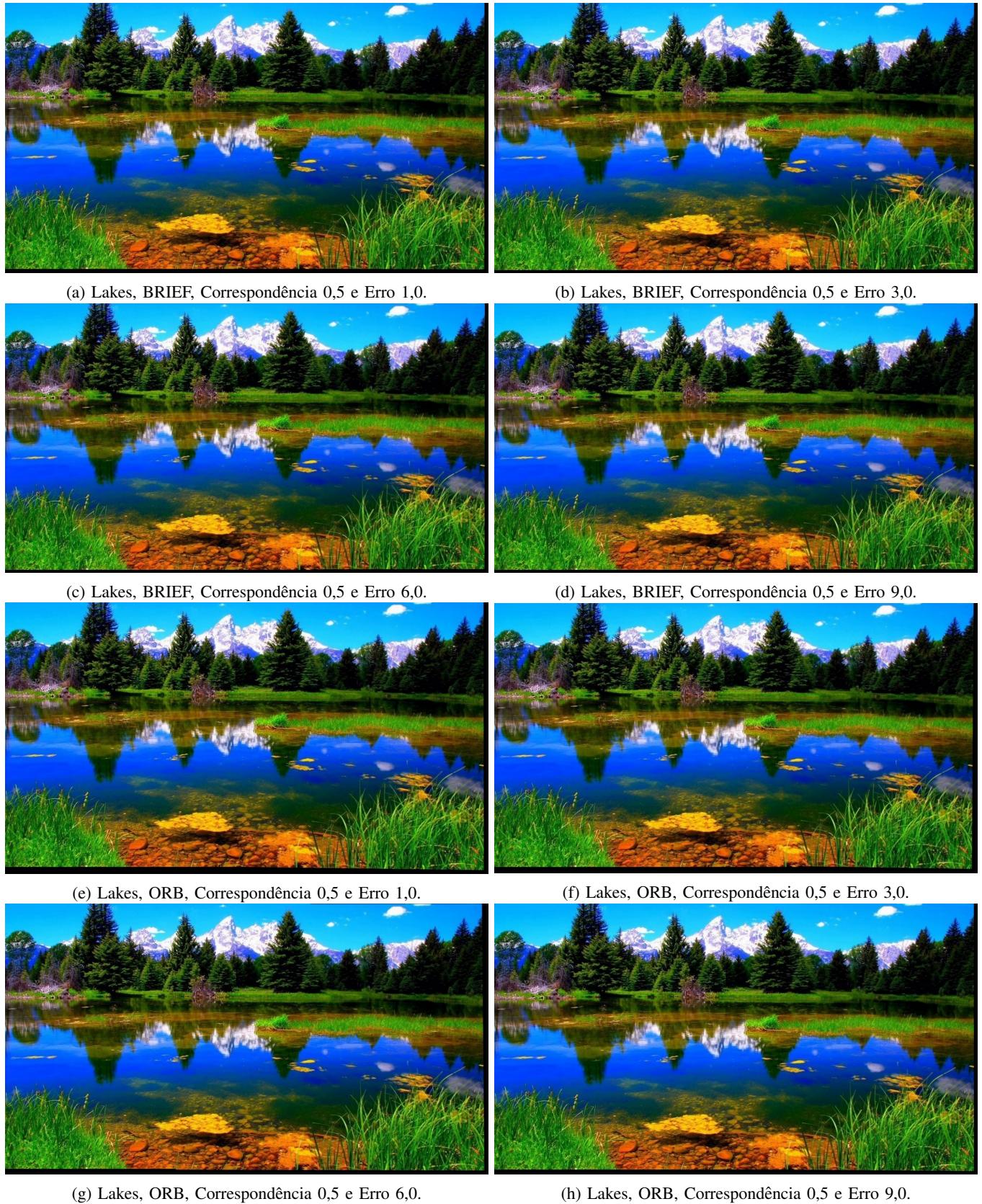


Figura 13: Imagens panorâmicas Lakes, para BRIEF e ORB, diferentes limiares de erro e correspondência fixa.



(a) Roads, BRIEF, Correspondência 0,5 e (b) Roads, BRIEF, Correspondência 0,5 e (c) Roads, BRIEF, Correspondência 0,5 e
Erro 1,0. Erro 3,0. Erro 6,0.



(d) Roads, BRIEF, Correspondência 0,5 e (e) Roads, ORB, Correspondência 0,5 e Erro (f) Roads, ORB, Correspondência 0,5 e Erro
Erro 9,0. 1,0. 3,0.



(g) Roads, ORB, Correspondência 0,5 e Erro (h) Roads, ORB, Correspondência 0,5 e Erro
6,0. 9,0.

Figura 14: Imagens panorâmicas Roads, para BRIEF e ORB, diferentes limiares de erro e correspondência fixa.