

Aplicativo para composição de imagens digitais em dispositivos móveis

Joelmir José Justino

UNIFAL-MG

Alfenas, Brasil

Email: a11014@bcc.unifal-mg.edu.br

Luiz Eduardo da Silva

UNIFAL-MG

Alfenas, Brasil

Email: luizedu@bcc.unifal-mg.edu.br

Resumo—Composição de imagens é a tarefa de combinar imagens com regiões que se sobrepõem em uma imagem maior, sendo um dos ramos mais antigos e estudados em processamento digital de imagens, com uma grande variedade de algoritmos propostos. Este trabalho apresenta um aplicativo para dispositivos móveis que implementa um desses algoritmos, baseado nas transformações *watershed* (Linha divisora de águas) e reconstruções morfológicas, para o Sistema Operacional Android. Os testes demonstraram que na imagem resultante a linha de costura que combina as imagens é praticamente invisível. Porém, obteve-se um baixo desempenho ao combinar imagens de alta resolução (8 MP). A partir desses resultados pode-se concluir que o algoritmo possui alta eficiência na escolha da linha de costura.

Abstract—Image composition is the task of combining images with regions that overlap in a bigger image, being one of the oldest and studied branches in digital image processing, with a great variety of proposed algorithms. This work presents a mobile application that implements one of these algorithms, based on the watershed transformations and morphological reconstructions for the Android Operating System. Tests have shown that in the resulting image the stitching line that combines the images is virtually invisible. However, poor performance was achieved by combining high-resolution (8 MP) images. From these results it can be concluded that the algorithm has high efficiency in the selection of the seam line.

I. INTRODUÇÃO

Algoritmos para composição de imagens que buscam uma linha de costura invisível estão entre os mais antigos e amplamente utilizados em processamento digital de imagens. Estes são os responsáveis pelas imagens de alta resolução presentes em mapas digitais e imagens de satélites, além de estarem presentes em praticamente todos os dispositivos móveis e câmeras digitais vendidas atualmente [1].

Uma das técnicas que se utiliza para a composição de imagens é a utilização de algoritmos baseados em *watershed* (Linha Divisora de águas - LDA) e reconstrução morfológica, para encontrar uma linha de costura que conecte as imagens de forma a minimizar a visibilidade da linha de costura que as unem.

O algoritmo proposto por SILVA, L. E. em sua tese de mestrado [2], originalmente implementado em computadores de mesa, utilizando a linguagem de programação Java, implementa as transformações LDA para a criação de imagens panorâmicas. Este trabalho apresenta uma implementação de tal algoritmo para dispositivos móveis que utilizam o Sistema Operacional Android.

Os dispositivos móveis tais como *smartphones* (celulares inteligentes) e *tablets* não são apenas ferramentas de comunicação eficientes, mas também dispositivos computacionais capazes, equipados com câmeras digitais de alta resolução, telas coloridas de alta qualidade, e processadores gráficos (*GPU*) com grande poder de processamento. No entanto, em comparação com os computadores de mesa, os dispositivos móveis, ainda apresentam algumas desvantagens, incluindo quantidade limitada de memória e capacidade computacional.

II. TRABALHOS RELACIONADOS

Esta seção descreve brevemente alguns trabalhos relacionados a este artigo.

Um algoritmo para a obtenção de imagens panorâmicas utilizando aritmética de inteiros em dispositivos móveis com poder de processamento e memória limitados é apresentado por [3]. Para se alcançar uma projeção eficiente sem a estimação de transformações são capturadas sucessivas imagens através de um janela semitransparente e, para o caso de uma imagem panorâmica horizontal, a câmera deve ser rotacionada no eixo y com um ângulo fixo, de forma a obter-se áreas de sobreposição. Então, cada imagem é projetada em uma superfície cilíndrica utilizando uma equação de projeção e aritmética de inteiros. Este trabalho, no entanto, requer a captura das imagens, para uma imagem panorâmica horizontal, que a câmera se mova no eixo x , mantendo-se fixa no eixo y .

Os autores Xiong and Pulli [4] apresentam uma abordagem para combinar um conjunto de imagens com uma região de interseção previamente alinhadas, buscando uma linha de costura ótima na área de sobreposição das imagens, de forma rápida e com baixo consumo de memória. Um caminho mínimo é encontrado na área de sobreposição das imagens através de programação dinâmica e utilizando uma linha de costura ótima para rotular as imagens. A utilização de programação dinâmica garante que a costura encontrada será ótima, permitindo que as imagens sejam combinadas de forma rápida e problemas como fantasmas e borrões causados por objetos que se movem na cena e pequenos erros no momento da captura da imagem são evitados.

Buscando melhorar a performance do processo de costura das imagens o trabalho apresentado por Yan et al. [5] propõem que primeiro se encontre as correspondências utilizando-se *SIFT* (*Scale Invariant Feature Transform*) e então se faça o

uso de *RANSAC* (*Random Sample Consensus*) para filtrar as correspondências erradas antes de estimar as transformações as projeções desejadas de forma a otimizar o processo de costura.

Com o objetivo de reduzir o poder computacional necessário para se processar uma imagem panorâmica em dispositivos móveis o trabalho apresentado por Wang et al [6] utiliza *Mobile offloading*, que consiste em transferir o processamento pesado que seria realizado no dispositivo móvel para grandes servidores, reduzindo o tempo de processamento e o consumo elétrico do dispositivo.

O trabalho apresentado por Lin and Yang [7] propõem o uso colaborativo de *smartphones* para a captura das imagens, onde mais de um *smartphones* participa da captura das imagens, onde um é utilizado como servidor e os outros como clientes. Todas as imagens capturadas são enviadas para o servidor onde o processo de costura é realizado gerando a imagem final.

III. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados conceitos importantes para o entendimento do algoritmo utilizado. Na primeira subseção alguns conceitos sobre Imagens Digitais serão explicados. Em seguida, é introduzido o conceito de Morfologia Matemática e Linha Divisora de Águas.

A. Imagens Digitais

De maneira geral, imagens digitais representam uma cena bidimensional onde cada ponto da imagem é codificado usando um *string* de *bits*, permitindo sua utilização por computadores. Em uma definição mais formal, uma imagem digital é uma função $f(x, y)$ que representa a intensidade luminosa da cor em um determinado ponto (x, y) em que x e y representam coordenadas espaciais no plano [8].

As imagens digitais do tipo rastreamento (*raster*) e vetorial possuem um conjunto finito de valores digitais chamados *picture elements* ou *pixels*. Os *pixels* são a menor unidade individual de uma imagem, contendo valores quantizados que representam o brilho em um ponto da cena.

B. Imagens panorâmicas

Uma imagem panorâmica é a captura de uma cena com um campo de visão horizontal tão grande quanto o olho humano é capaz de enxergar. A obtenção de uma imagem panorâmica pode ser realizada utilizando-se um conjunto de imagens com menor ângulo de visão ou a partir de uma sequência de vídeo.

A principal vantagem em se construir imagens panorâmicas a partir de um conjunto de fotografias é a possibilidade de se capturar as imagens individuais com um maior grau de precisão. Porém, um método de para realizar a costura das imagens em sua região de sobreposição, formando uma linha que combine as imagens de tal forma que não se perceba que a imagem final foi formada por mais de uma imagem [9].

C. Morfologia Matemática

A morfologia matemática estuda a estrutura geométrica das entidades presentes em uma imagem, permitindo processar imagens com o objetivo de realce, segmentação, detecção de

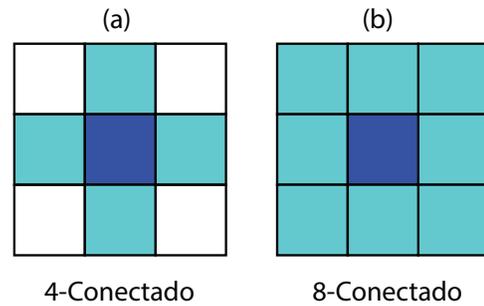


Figura 1. Conectividade de pixel. (a) pixel 4-conectado, (b) pixel 8-conectado

bordas, esqueletização, afinamento, análise de formas, compressão e etc. [10].

O poder da morfologia matemática está no elemento estruturante que consiste em um conjunto completamente definido e conhecido (forma, tamanho) que é comparado, a partir de uma transformação, ao conjunto desconhecido da imagem, podendo determinar se o elemento estruturante está ou não presente na imagem.

Conectividade de *pixels* em processamento digital de imagens é o modo em que os *pixels* em uma imagem bidimensional se relacionam com seus vizinhos [11]. Tipos de conectividade:

- *Pixels* 4-conectados são os que estão conectados horizontalmente e verticalmente, são vizinhos à todos os *pixels* que tocam uma de suas bordas, que possuem as coordenadas $(x \pm 1, y)$ ou $(x, y \pm 1)$ é conectado ao *pixel* em (x, y) , figura 1 (a).
- *Pixels* 8-conectados são os que estão conectados horizontalmente, verticalmente e diagonalmente com as coordenadas $(x \pm 1, y \pm 1)$, juntamente com os *pixels* 4-conectados é conectado ao *pixel* em (x, y) , figura 1 (b).

A ideia da transformação *watershed* (LDA) é concebida de uma analogia topográfica. Segue um exemplo intuitivo da LDA: considere uma imagem onde os níveis de cinza representam a altitude dos objetos presente nesta. Se cada valor mínimo dessa imagem contiver um orifício, ao a afundarmos em uma região com água em uma velocidade vertical constante, as regiões da imagem serão preenchidas e, com isso, dois ou mais fluxos vindos de mínimos diferentes podem se unir. Os diques construídos na superfície para evitar tal junção das águas constituem a LDA da imagem Figura 2. A transformação LDA é a principal ferramenta para a segmentação de uma imagem.

IV. ALGORITMO MORFOLÓGICO PARA COMPOSIÇÃO DE IMAGENS

Antes de adentrar-se ao conteúdo, frise-se que esta seção descreve o algoritmo utilizado nesse trabalho.

O algoritmo para composição de imagens proposto por SILVA, L. E. [2], utilizado nesse trabalho, busca encontrar uma linha de costura conexa e irregular que descreva um caminho 4-conectado ou 8-conectado na região de interseção, em que

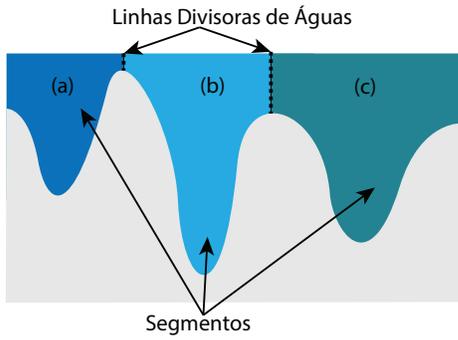


Figura 2. LDA - (a), (b) e (c) são segmentos formados pelo preenchimento das bacias topográficas

um ponto presente na linha deve possuir a maior similaridade em relação ao seu vizinho não pertencente a linha, com a região de interseção sendo atravessada de um extremo ao outro. A figura 3 mostra a sequência de passos para a execução do algoritmo.



Figura 3. Pipeline com a divisão dos passos para a execução do algoritmo.

Para a geração da imagem panorâmica é indispensável que as imagens selecionadas estejam em uma mesma escala e orientação, além de possuírem uma região de interseção. Considera-se que correções de escala e orientação já tenham sido realizados. O posicionamento para a identificação da região de interseção deve ser realizado manualmente.

Com a definição da região de interseção, o próximo passo é a definição das bordas e dos extremos da imagem de correlação. As bordas representam a costura natural ao efetuar-se uma simples sobreposição das imagens. Já os extremos, representam o início e o fim da linha de costura. A figura 4 mostra um exemplo de região de correlação com suas bordas e extremos. Onde a região cinza representa as duas imagens a serem combinadas e a região verde (centro da imagem) representa a região de interseção.

Com base na região de interseção, deve-se calcular o índice de similaridade (IC) entre os *pixels*. Para isso utiliza-se o esquema de cores *RGB*, onde R_1 , G_1 e B_1 as três bandas (*red*, *green*, *blue*) da imagem 1, e R_2 , G_2 e B_2 as três bandas da imagem 2.

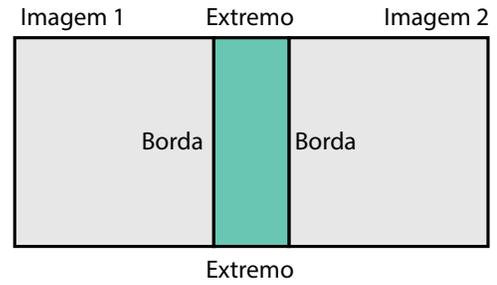


Figura 4. Imagem de correlação - definição das bordas e extremos

O IC pode ser calculado como a máxima diferença nas três bandas:

$$IC(x, y) = \max(|\Delta R_i|, |\Delta G_i|, |\Delta B_i|) \quad (1)$$

ou a diferença ponderada,

$$IC(x, y) = \sqrt{\Delta R_i^2 + \Delta G_i^2 + \Delta B_i^2} \quad (2)$$

onde $\Delta R_i = R_1(x, y) - R_2(x, y)$, $\Delta G_i = G_1(x, y) - G_2(x, y)$ e $\Delta B_i = B_1(x, y) - B_2(x, y)$, ou ainda, pode-se utilizar o modelo $L * u * v^*$, em que L^* , u^* e v^* são definidos a partir dos valores RGB [12], [13],

$$IC(x, y) = \sqrt{(\Delta L_i^*)^2 + (\Delta u_i^*)^2 + (\Delta v_i^*)^2} \quad (3)$$

onde o valor de $\Delta L_i^* = L_1^*(x, y) - L_2^*(x, y)$, $\Delta u_i^* = u_1^*(x, y) - u_2^*(x, y)$, $\Delta v_i^* = v_1^*(x, y) - v_2^*(x, y)$.

Ao aplicar a transformação LDA na imagem de correlação, obtêm-se as linhas mais promissoras para a linha de costura. Porém, obtêm-se uma imagem supersegmentada, sendo que um processo de filtragem visando a definição da melhor linha de costura se faz necessário.

Assim obtêm-se uma imagem numérica, de mesma dimensão da região de interseção, com cada ponto representando a correlação entre os pontos nas duas imagens. Um valor numérico baixo representa uma alta similaridade entre os pontos, bons candidatos à linha de costura. Um valor alto, representa uma baixa similaridade.

Ao aplicar a transformação LDA na imagem de correlação, obtêm-se as linhas mais promissoras para a linha de costura. Porém, obtêm-se uma imagem supersegmentada, sendo que demanda uma filtragem visando a definição da melhor linha de costura.

Para a simplificação da imagem após aplicar-se a LDA e definição de uma única linha de costura, o algoritmo utilizado aplica uma modificação homotópica baseada em reconstrução. Com o objetivo de determinar os mínimos (marcadores) a partir dos quais realiza-se o processo de inundação no cálculo da LDA, serão utilizadas as bordas da imagem de correlação como primeiros marcadores para a modificação homotópica.

Para encontrar uma linha de costura que passe por uma região de alta correlação, o algoritmo proposto por SILVA, L. E. [2] utiliza um processo recursivo em que a linha de costura obtida no passo anterior define os marcadores das próximas

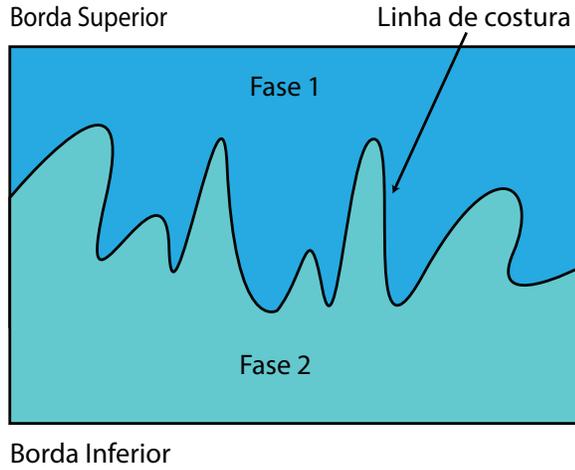


Figura 5. Linha de costura encontrada no passo inicial, considerando as bordas superior e inferior como marcadores

iterrações (Figura 5). Assim, obtém-se uma subdivisão da imagem em duas fases, em que a imagem de mínimos marcadores da próxima fase é definida como:

$$\forall p \in D, I(p) = \begin{cases} 1, & \text{se } p \text{ é um ponto de borda ou } p \\ & \text{pertence a uma fase considerada} \\ 0, & \text{caso contrário.} \end{cases} \quad (4)$$

Assim, pode-se encontrar outras possíveis linhas de costura nas duas fases definidas na etapa corrente do algoritmo.

Para encontrar uma linha de costura na fase 1 da (Figura 5), por exemplo, executa-se o mesmo processo do passo inicial, considerando a fase 2 e a borda superior como marcadores. Este processo resulta em uma linha de costura unindo um extremo a outro da região de interseção, passando, eventualmente, por pontos de alta correlação na fase 1, inclusive por pontos da linha de costura do passo inicial, três critérios são utilizados como condição de parada da recursão os seguintes critérios:

- 1) A qualidade da nova linha calculada é inferior a da linha obtida no passo anterior, segundo algum critério de avaliação. Seja $L = \{p_0, p_1, \dots, p_2\}$ a sequência de pontos que formam a linha de costura e IC' o negativo da imagem de correlação. Critérios utilizados nesse trabalho são:

- $$\frac{1}{n+1} \sum_{i=0, p_i \in L}^n IC'(p_i) \quad (5)$$

- $$\frac{\frac{1}{n+1} \sum_{i=0, p_i \in L}^n IC'(p_i)}{\sum_{i=1, p_i \in L}^n |IC'(p_i) - IC'(p_{i-1})|} \quad (6)$$

- $$\frac{\frac{1}{n+1} \sum_{i=0, p_i \in L}^n IC'(p_i)}{\max_{p \in L} (IC'(p_i)) - \min_{p \in L} (IC'(p_i))} \quad (7)$$

A equação 5 considera o somatório da correlação dos pontos na linha de costura como medida de qualidade desta. A equação 6 valoriza as linhas onde não ocorrem muitas diferenças abruptas entre pontos vizinhos. A equação 7 leva em conta a dinâmica da linha, calculando o somatório das correlações pela diferença entre o maior e o menor valor da linha.

- 2) a linha de costura passa por uma região onde a diferença da imagem reconstruída X' , e a imagem de correlação original, IC , é igual a zero.
- 3) a linha de costura obtida na iterração atual é igual à linha encontrada na recursão precedente.

Em termos algorítmicos, o critério 1) é utilizado para avaliar a qualidade da linha de costura, indicando a convergência do resultado (fim das subdivisões recursivas). O critério 2) é utilizado para evitar que a linha de costura passe por regiões de baixa correlação, e o critério 3) detecta a aparição na chamada atual do algoritmo recursivo, da mesma linha definida na recursão anterior, evitando, assim, um número infinito de chamadas recursivas.

O último passo é gerar a imagem final unindo-se às imagens através da linha de costura calculada no passo anterior.

V. APLICATIVO

Nesta seção será apresentada uma descrição de como o aplicativo foi desenvolvido.

O desenvolvimento do aplicativo se deu a partir do código original escrito por SILVA, L. E. [2] na linguagem de programação Java. Em primeiro lugar realizou-se uma extensiva leitura do código para posteriormente realizar um trabalho de *refactor* (reescrever o código fonte existente, a fim de melhorar sua legibilidade, capacidade de reutilização ou estrutura sem afetar seu significado ou comportamento).

O primeiro passo do *refactor*, se deu em portar a interface gráfica do usuário (GUI) que utilizava AWT (*Abstract Window Toolkit*) para o utilizar JavaFX. Essa foi uma importante fase pois permitiu identificar como a aplicação original estava estruturada.

Um pacote java chamado base foi criado. Neste pacote foi colocado todas as classes referentes a implementação do algoritmo. Além de novas classes que permitem o uso da implementação original utilizando-se um modelo orientado a eventos, assim facilitando o uso do algoritmo utilizando um modelo *multi thread*, permitindo que a interface gráfica não congelasse durante o processo de execução do algoritmo.

O conceito de imagem no algoritmo foi abstraído em uma interface chamada *ImageWrapper*, permitindo que no fim do *refactor* da aplicação original para utilizar JavaFX, o pacote base pudesse ser diretamente utilizando no aplicativo Android, diferenciando se apenas na implementação da interface *ImageWrapper*.

A figura 6 apresenta a tela de configuração do aplicativo. Nesta tela podemos selecionar o tamanho que a imagem terá durante o processo de posicionamento manual das imagens que faram parte do panorama. Além de permitir a utilização do

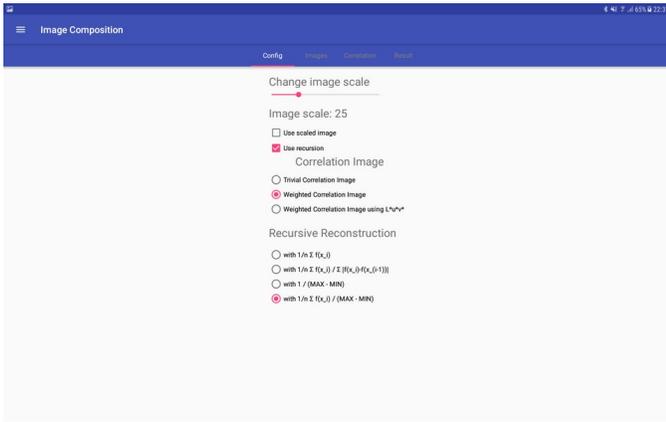


Figura 6. Tela de configuração

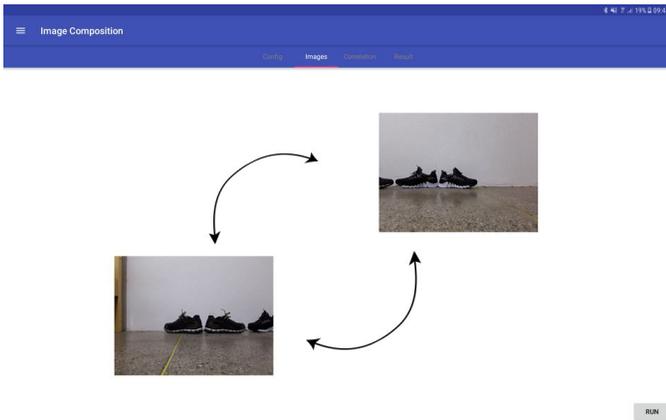


Figura 7. Tela de configuração

tamanho original ou o tamanho reduzido. As outras configurações permitem habilitar ou desabilitar a recursão, selecionar o critério de parada da recursão, e a equação utilizada para gerar a imagem de correlação. Na figura 7 é exibida uma imagem mostrando a tela de posicionamento manual do aplicativo, nesse passo devemos posicionar as imagens com em suas região de sobreposição.

VI. RESULTADOS

O algoritmo foi testado em um Samsung Galaxy Tab A 10.1 com a especificação da Tabela I. A figura 8 demonstra a simples sobreposição das imagens que compõem o panorama. Na figura 9 é mostrado as imagens de correlação, com o índice de similaridade calculado utilizando a equação 1 de diferença máxima, equação 2 de diferença ponderada e 3 utilizando o modelo $L * u * v*$, respectivamente, com a linha de costura escolhida no processo recursivo marcada em vermelho. Assim observa-se que a linha de costura escolhida utilizando tais equações apresentam valores similares. A figura 10 mostra o resultado final da imagem panorâmica, com uma linha de costura que é praticamente invisível. Porém, a diferença na iluminação das duas cenas é visível. A tabela II mostra as a média de desempenho na execução do algoritmo. Como

Tabela I
ESPECIFICAÇÕES DO DISPOSITIVO DE TESTE

CPU	Octa-core (4x1.6 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53)
Memória	2 GB RAM
Tela	10.1 polegadas
Câmera	8 MP, f/1.9, AF, LED flash

Tabela II
TESTE DE DESEMPENHO

Tipo de correlação	Tempo em segundos
Diferença máxima	62
Diferença ponderada	87,83s
Modelo $L * u * v*$	86,5s



Figura 8. Simples sobreposição das imagens para definição do panorama

podemos observar para uma imagem com 8 MP de resolução, ao utilizarmos a equação de diferença máxima obtivemos o melhor resultado

VII. CONCLUSÃO

Neste trabalho foi apresentado um aplicativo para composição de imagens implementado para dispositivos móveis que utilizam o Sistema Operacional, utilizado o algoritmo proposto por SILVA, L. E. [2], que ao contrário de outros métodos, define uma linha de costura conexa e irregular para a composição de imagens, utilizando-se essencialmente na operação morfológica de segmentação de imagens, denominada Linha Divisora de Águas (*Watershed*), mostrando ser eficiente em encontrar uma boa linha de costura para as imagens. Todavia, a implementação atual apresenta uma baixa performance, como os teste mostram, além de pode-se observar a diferença de iluminação entre as imagens, sendo estes dois tópico para futuros trabalhos.

REFERÊNCIAS

- [1] R. Szeliski, "Image Alignment and Stitching: A Tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.
- [2] L. E. da Silva, "Algoritmo de Fotomosaico para Imagens Coloridas por Morfologia Matemática," Masters, 1999.
- [3] S. J. Ha, H. I. Koo, S. H. Lee, N. I. Cho, and S. K. Kim, "Panorama mosaic optimization for mobile camera systems," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1217–1225, 2007.
- [4] Y. Xiong and K. Pulli, "Fast Panoramic Stitching for High-Quality Panoramic Images on Mobile Phones," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2592–2600, 2010.

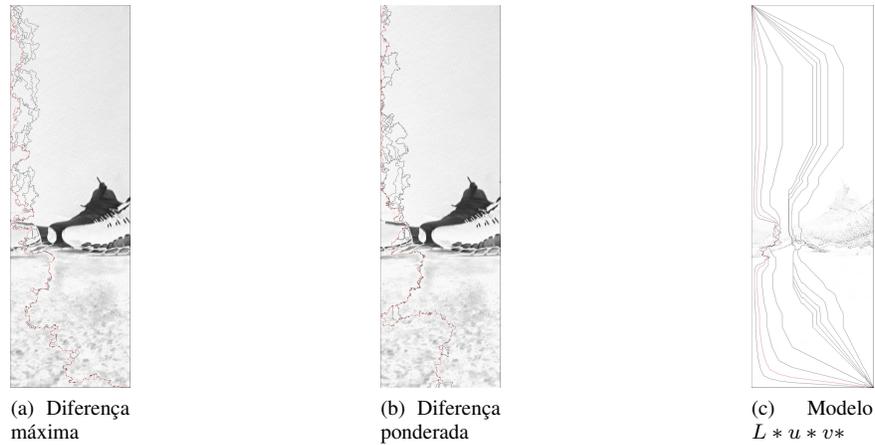


Figura 9. Imagens de correlação.



Figura 10. Imagem final

- [5] W. Yan, C. Liu, and W. Luo, "Fast and low complexity image stitching method on mobile phones," *Control, Automation and Information*, pp. 245–250, 2015.
- [6] Q. Wang, F. Reimeier, and K. Wolter, "Efficient Image Stitching through Mobile Offloading," *Electronic Notes in Theoretical Computer Science*, vol. 327, pp. 125–146, 2016.
- [7] J. Lin and C.-K. Yang, "Collaborative panoramic image generation from multiple mobile phones," *Multimedia and Expo Workshops (ICMEW) 2017*, no. July, pp. 339–344, 2017.
- [8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*, 3rd ed. Pearson; 3 edition (August 31, 2007), 207.
- [9] S. P.-a. Conference, J. Li, and J. Du, "Study on Panoramic Image Stitching Algorithm," no. 408202, pp. 417–420, 2010.
- [10] J. Facon, *Morfologia Matemática Teoria e Exemplos*. Curitiba, Brasil: Editora Universitária Champagnat da Pontifícia Universidade Católica do Paraná, 1996.
- [11] A. C. K. Azriel Rosenfeld, *Digital Picture Processing*, 2nd ed., ser. Computer Science and Applied Mathematics volume 1 and 2. Morgan Kaufmann, 1982.
- [12] J. Tajima, "Uniform Color Scale Applications to Computer Graphics," *Computer Vision, Graphics and Image Processing*, vol. 21, pp. 305–325, 1983.
- [13] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Transactions on Image Processing*, vol. 6, no. 11, pp. 1530–1544, 1997.