

MC940 - Análise de Imagem

Felipe Santana Dias - 215775

24 de maio de 2022

Sumário

Visando extrair ROIs (Regions of Interesting) relevantes para a comparação entre imagens de impressões digitais, neste relatório foram utilizadas técnicas de IFT combinadas ao uso de filtros morfológicos para realizar essa seleção.

As informações neste documento foram separadas em seções como se segue: A caracterização do problema e o embasamento teórico proposto são descritos nas seções 1 e 2. As seções 3, 4 e 5 são dedicadas à explicação dos algoritmos utilizados, a execução realizada e os resultados encontrados. Finalizando o relatório, as seções 5, 6 e 7 apresentam os resultados, as conclusões realizadas e a bibliografia utilizada.

1. Problema

Para que impressões digitais sejam reconhecidas e comparadas entre si, é necessário realizar o processamento dessas imagens para que os pares sejam corretamente reconhecidos. Essas imagens nem sempre são uniformes e da mesma maneira. No banco de dados utilizado para esse projeto (*imagens.01*) as impressões digitais possuem assinaturas, marcas de caneta e outras informações sobrepostas às linhas dos dedos, podendo se tornar um impeditivo para o reconhecimento da pessoa correta.

O objetivo desse projeto, portanto, é extrair uma região de interesse (ROI) contendo a melhor área da impressão digital com o mínimo de ruído possível para futuramente realizarmos a análise e comparação com as ROIs de outras impressões digitais.

2. Literatura proposta

Para solucionar essa situação, as imagens das impressões digitais precisam passar por um processamento de imagem que reconheça as áreas e traços de maior importância para a identificação dos indivíduos. A partir desse reconhecimento, é realizada a transformação dessas imagens para que seja possível a comparação entre as diferentes digitais, tornando possível a identificação dos indivíduos,

Como proposta e baseando-se nos estudos de Alexandre X. Falcão, Jorge Stolfi, e Roberto de Alencar Lotufo¹, essas transformações podem ser criadas a partir da conectividade entre os pixels. Os pontos que possuem um caminho entre eles são considerados conexos e formam os objetos da imagem. Esses caminhos, por sua vez, podem ser transformados em caminhos ótimos e que podem ser reduzidos a uma floresta de caminhos ótimos.

Dessa forma, o Image Forest Transformation (IFT) nos possibilita selecionar as linhas das digitais como uma floresta de grafos. Para encontrarmos o centro da impressão digital e selecionar o melhor ROI possível, é necessário transformar os múltiplos componentes gerados da análise inicial no maior componente conexo possível, estimando assim o melhor centro geométrico possível. Esse centro geométrico é utilizado no cálculo do ROI.

3. Algoritmos implementados

A partir dos grafos encontrados na leitura da imagem, podemos aplicar filtros morfológicos para conectar os componentes distantes. A construção desses filtros são baseados nas funções de Dilatação e Erosão.

3.1 Função de Dilatação

```
1. Para Cada  $p \in D_I$  Faça  $C(p) \leftarrow +\infty$  e  $D(p) \leftarrow I(p)$ .
2. Se  $S = \emptyset$  Então  $S \leftarrow \{p \in D_I \mid I(p) \neq 0 \text{ e } \exists q \in A_1(p), I(q) = 0\}$ .
3. Enquanto  $S \neq \emptyset$  Faça
4.   | Remova  $p$  de  $S$ .
5.   | Atribua  $C(p) \leftarrow 0$ ,  $R(p) \leftarrow p$ , e insira  $p$  em  $Q$ .
6. Enquanto  $Q \neq \emptyset$  Faça
7.   | Remova  $p$  de  $Q$  tal que  $p = \arg \min_{q \in Q} \{C(q)\}$ .
8.   | Se  $C(p) \leq \gamma^2$  Então
9.     | Atribua  $J(p) \leftarrow I(R(p))$ .
10.    | Para Cada  $q \in A_{\sqrt{2}}(p) \mid q \in D_I \text{ e } C(q) > C(p)$ 
11.      | e  $I(q) = 0$  Faça
12.        | Atribua  $tmp \leftarrow \|q - R(p)\|^2$ .
13.        | Se  $tmp < C(q)$  Então
14.          | Se  $q \in Q$  Então Remova  $q$  de  $Q$ .
15.          |  $C(q) \leftarrow tmp$  e  $R(q) \leftarrow R(p)$ .
16.          | Insira  $q$  em  $Q$ .
17.    | Senão Atribua  $S \leftarrow S \cup \{p\}$ .
18. Retorne  $\hat{D}$  e  $S$ .
```

A função de Dilatação, implementada pela função *MyDilateBin*, propaga o valor máximo do brilho (intensidade) da relação de adjacência de vizinhança-8. O algoritmo inicia percorrendo um conjunto S de sementes que correspondem aos pixels de borda interna do objeto. Esse conjunto, se não passado através dos parâmetros, é calculado através da função *MyObjectBorder*, que verifica se, para cada pixel da imagem, existem vértices que possuem brilho diferente de zero (correspondente aos valores de preenchimento do objeto) com vizinhança-4 à pelo menos um pixel com valor igual a zero, demonstrando que esse ponto está no limite entre o objeto e o fundo.

A partir desses pixels é formado uma fila de prioridade Q ordenada pelos menores custos atribuídos a cada pixel. Inicialmente, todos os elementos da fila Q possuem custo igual a zero, enquanto todos os outros elementos possuem custo infinito - especificamente, atribuímos o valor de 10.000 para esses casos.

Se este custo é menor que o quadrado do raio de dilatação, um dos parâmetros da nossa função, o brilho da raiz desse pixel p é atribuído à máscara da imagem dilatada D . A partir deste ponto, para cada vizinho-8 q que possui brilho igual a zero e custo maior que o

de p , é calculado a distância euclidiana entre ele e a raiz de p . Para facilitar os cálculos, esses custos são elevados ao quadrado, sendo portanto, todos valores inteiros. Se o valor encontrado for menor que o custo prévio deste pixel q , então esse pixel é removido da fila Q , seu custo é atualizado para o valor encontrado e a raiz de q passa a ser a raiz de p . Com os valores atualizados, o pixel q é inserido novamente na fila até que Q esteja vazio.

Se o custo de p é maior que o quadrado do raio de dilatação, p é inserido em S e quando finalizado, essa lista corresponde aos pixels de borda externa do objeto, podendo ser utilizados como sementes na função de Erosão.

3.2 Função de Erosão

```

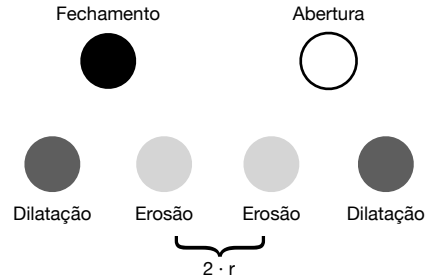
1. Para Cada  $p \in D_I$  Faça  $C(p) \leftarrow +\infty$  e  $E(p) \leftarrow I(p)$ .
2. Se  $S = \emptyset$  Então  $S \leftarrow \{p \in D_I \mid I(p) = 0 \text{ e } \exists q \in A_1(p), I(q) \neq 0\}$ .
3. Enquanto  $S \neq \emptyset$  Faça
4.   Remova  $p$  de  $S$ .
5.   Atribua  $C(p) \leftarrow 0$ ,  $R(p) \leftarrow p$ , e insira  $p$  em  $Q$ .
6. Enquanto  $Q \neq \emptyset$  Faça
7.   Remova  $p$  de  $Q$  tal que  $p = \arg \min_{q \in Q} \{C(q)\}$ .
8.   Se  $C(p) \leq \gamma^2$  Então
9.     Atribua  $J(p) \leftarrow I(R(p))$ .
10.    Para Cada  $q \in A_{\sqrt{2}}(p) \mid q \in D_I, C(q) > C(p)$ 
11.      e  $I(q) \neq 0$  Faça
12.        Atribua  $tmp \leftarrow \|q - R(p)\|^2$ .
13.        Se  $tmp < C(q)$  Então
14.          Se  $q \in Q$  Então Remova  $q$  de  $Q$ .
15.           $C(q) \leftarrow tmp$  e  $R(q) \leftarrow R(p)$ .
16.          Insira  $q$  em  $Q$ .
17.        Senão Atribua  $S \leftarrow S \cup \{p\}$ .
18. Retorne  $\hat{E}$  e  $S$ .

```

A função de Erosão *MyErodeBin*, por sua vez, é muito semelhante a de Dilatação. As diferenças consistem basicamente na utilização da borda externa do objeto como sementes, podendo este ser fornecido pela função *MyBackgroundBorder* ou na seleção encontrada na função anterior.

Com essas duas funções formamos os filtros de Abertura e Fechamento. O primeiro é a combinação da função de erosão seguida da dilatação, enquanto o segundo é seu oposto - dilatação e erosão. Em nosso projeto, o filtro responsável por transformar os diversos componentes separados no maior possível é *MyAsfCOBin*. Este filtro é a combinação entre Fechamento e Abertura. Com isso esperamos encontrar uma componente maior que corresponda a impressão digital.

Para a implementação dessa função, é possível combinar os algoritmos de modo a otimizar a chamada das funções auxiliares. Como mostrado no grafo abaixo, ao combinar ambas funções é possível utilizar somente as funções dilatação e erosão sendo utilizado nessa última duas vezes o valor do raio.



Essa operação tem como consequência o alargamento das bordas externas, gerando deformações que podem prejudicar o reconhecimento das figuras. Para solucionar essa situação é possível aplicar um filtro de erosão que removerá esses excessos. Como essa é uma função antagônica à realizada anteriormente, se já aplicarmos ela, pode acontecer de componentes que estavam conexas voltarem a se separar. Por isso, utilizamos primeiro uma função de fechamento de buracos para que os pixels que não possuem conexões fortes entre si possam ser completos, permitindo que a erosão aconteça somente nas bordas externas.

3.3 Função de Fechamento de Bacias

```

1. Para Cada  $p \in D_I$  Faça
2.   Faça  $C(p) \leftarrow +\infty$ 
3.   Para Cada  $q \in A_1(p)$  Faça
4.     Se  $q \notin D_I$  Então
5.       Faça  $C(p) \leftarrow I(p)$ , insira  $p$  em  $Q$  e break.
6. Enquanto  $Q \neq \emptyset$  Faça
7.   Remova  $p$  de  $Q$  tal que  $p = \arg \min_{q \in Q} \{C(q)\}$ .
8.   Para Cada  $q \in A_1(p) \mid q \in D_I, C(q) > C(p)$  Faça
9.     Atribua  $tmp \leftarrow \max\{C(p), I(q)\}$ .
10.    Se  $tmp < C(q)$  Então
11.       $C(q) \leftarrow tmp$ .
12.      Insira  $q$  em  $Q$ .
13. Retorne  $\hat{C}$ .

```

A função de Fechamento de Bacias (*MyCloseBasins*) percorre todos os pixels p da imagem e atribui um custo infinito em cada um deles. Cada vez que a função encontra um vizinho q de vizinhança-4 que não faz parte do domínio da imagem, o custo de p é atualizado para o valor de sua intensidade e é adicionado à fila de prioridade Q .

Percorrida toda a imagem, iterativamente é removido o elemento de menor custo da fila Q , visita-se cada vizinho q de vizinhança-4 presente no domínio da imagem e se seu custo for maior que de p , o máximo valor entre o custo de p e a intensidade de q será atribuído ao novo custo $C(q)$ desde que seja menor que o valor existente previamente. O pixel q é inserido novamente na fila e o processo se repete até que Q esteja vazio.

Concluído esse processo, é feita então a erosão, encontrando assim o maior componente da imagem que é selecionado e, por fim, extraído o ROI.

4. Execução

Todos os algoritmos foram implementados no arquivo *project01.c* presente na pasta *fingerprints*. Para executá-lo, primeiro é realizado a compilação com o comando *make project01* seguido do comando *./bin/project01 images_01 pasta_destino*. Os ajustes e alterações nos parâmetros são realizados na função *main* dentro desse mesmo arquivo.

5. Resultados

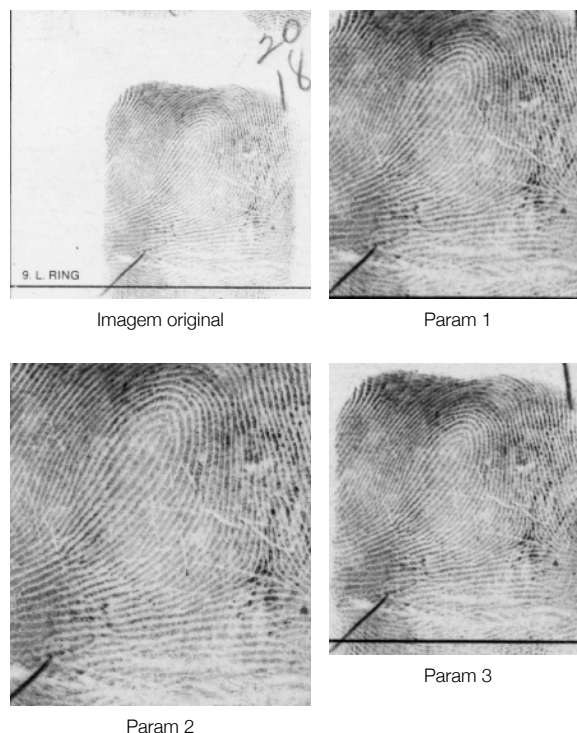
Resultados e discussões do encontrado

Os parâmetros disponíveis para encontrarmos os melhores resultados são os raios utilizados nas funções *MyAsfCOBin* ($r1$) e *MyErodeBin* ($r2$). Para testarmos como os raios combinados impactavam no resultado final, todos os testes realizados mantiveram os valores proporcionais entre eles. As imagens foram processadas de três maneiras diferentes:

1. $r1 = 15.0$ e $r2 = 30.0$, sendo esse o padrão inicial;
2. $r1 = 30.0$ e $r2 = 60.0$, o dobro dos valores iniciais;
3. $r1 = 2.0$ e $r2 = 4.0$, valores extremamente menores que os testados anteriormente.

Para avaliar os resultados iremos analisar o conjunto de 4 imagens gerados pelas transformações.

Imagem b101-9_t



Analisando o primeiro caso da imagem *b101-9_t*, é possível perceber que o parâmetro 3, por possuir um raio muito menor, tem menor exatidão em localizar o centro da impressão, implicando no aparecimento da borda inferior da caixa onde está localizada a impressão digital.

No caso dos parâmetros 2, a seleção é mais correta ao remover as linhas que não fazem parte da impressão digital. Contudo, a área de abrangência é menor, perdendo informações como a alta densidade de linhas no canto superior direito da imagem, que podem ser fatores cruciais na hora do reconhecimento.

O parâmetro 1 também considerou a borda inferior da caixa como parte da imagem, mas de maneira muito mais discreta. Ao que parece, apesar de não remover completamente as linhas intrusas, ele compreende uma região maior da impressão digital, podendo ser um fator positivo no resultado gerado.

Imagem b106-8_t

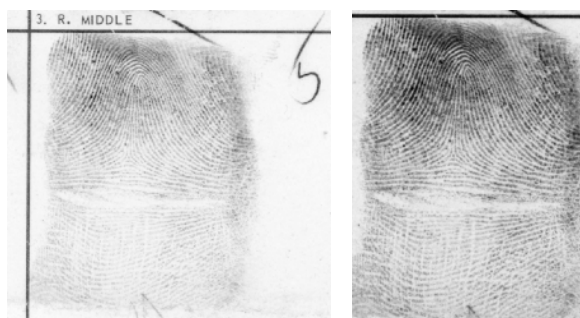
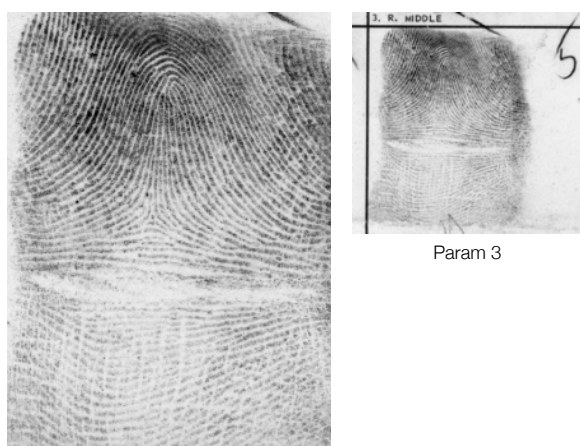


Imagem original

Param 1



Param 2

Param 3

No caso da imagem *b106-8_t*, percebemos que a imagem original, apesar de possuir poucos riscos que não correspondem à impressão digital, possui linhas mais finas devido a maneira de coleta da imagem.

Levando isso em conta, vemos que os parâmetros 1 e 3 não conseguiram distinguir muito bem as partes da impressão e as linhas do quadro, indicando que em situações como essas, trabalhar com raios muito maiores pode ser vantajoso, como observado na imagem do parâmetro 3.

Por fim, a imagem *b124-5_t* possui uma forma muito bem definida porém com um grande espaçamento entre duas partes da impressão digital.

O que percebemos ao comparar as imagens geradas pelos diferentes parâmetros é a possibilidade de perda de informações relevantes quando utilizados raios muito pequenos, como é o caso do parâmetro 3. Ao mesmo tempo, raios muito maiores podem realizar cortes desnecessários na imagem, como acontece no parâmetro 2.

Imagem b124-5_t

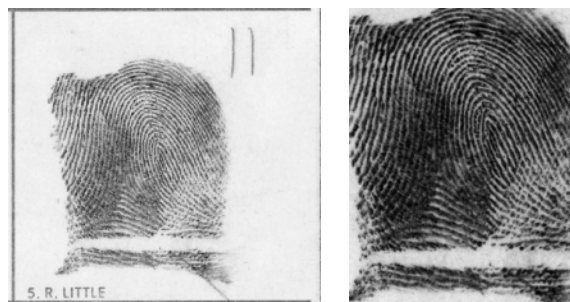
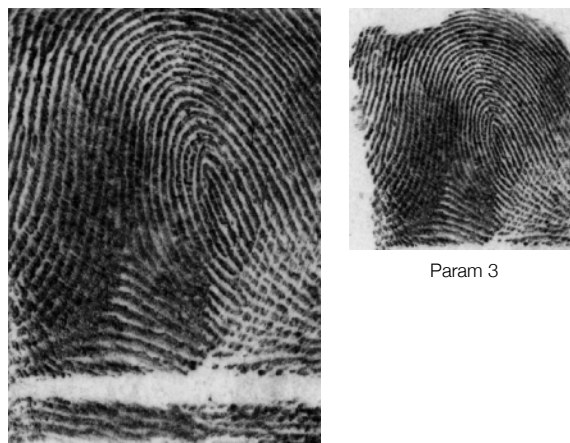


Imagem original

Param 1



Param 2

Param 3

Nessa situação, os parâmetros utilizados na composição 1 geraram resultados muito satisfatórios quando comparado ao proposto pelo desafio.

6. Conclusão

Com base nessas análises e nos códigos implementados, vemos que o uso de IFTs combinado dos filtros de transformação morfológica geram resultados satisfatórios para a seleção de ROIs relevantes.

Comparando os testes realizados, vemos que a utilização dos parâmetros do modo 1 ($r1 = 15$ e $r2 = 30$) podem gerar melhores resultados quando levado em conta as grandes diferenças que existem entre as imagens do banco utilizado.

Possíveis estudos que podem melhorar nossos resultados são relacionados à atribuição dinâmica dos parâmetros, utilizando outros pré-processamentos de imagem para identificar experimentalmente quais deles seriam melhores em cada situação específica.

7. Referências

¹ Falcao, A.X. Stolfi, J. de Alencar Lotufo, R.: "The image foresting transform: theory, algorithms, and applications", In IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 26, NO. 1, JANUARY 2004