MAC499 – Trabalho de Formatura Supervisionado

Verificação de Impressões Digitais utilizando Morfologia Matemática[1]

Daniel André Vaquero[2]

Orientador: **Prof. Dr. Junior Barrera**[3]

Departamento de Ciência da Computação[4]
Instituto de Matemática e Estatística
Universidade de São Paulo

8 de dezembro de 2003

Resumo

Este texto é a monografia final da disciplina MAC499. Nela são apresentados os aspectos técnicos de um projeto de iniciação científica desenvolvido durante o último ano do curso de Bacharelado em Ciência da Computação, além de um pouco da experiência pessoal adquirida pelo autor durante a participação no projeto e durante o curso do BCC.

[1] O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – Brasil

[2] `daniel@vision.ime.usp.br`

[3] `jb@ime.usp.br`

[4] Departamento de Ciência da Computação – IME - USP. Rua do Matão, 1010, São Paulo, SP, 05508-900 Brasil.

Sumário

1 O Projeto de Iniciação Científica

1.1 Introdução e Objetivos

As impressões digitais, marcas presentes nas pontas dos dedos dos seres humanos, podem ser utilizadas para identificação de pessoas, já que dois seres humanos nunca possuem as mesmas digitais, nem mesmo irmãos gêmeos, e estes sinais em geral permanecem inalterados durante a vida de um indivíduo, exceto por ocorrência de lesões ou degeneração [6]. Esse fato possibilitou a utilização das impressões digitais para a identificação de pessoas. Uma aplicação clássica é a identificação de criminosos pela polícia, através das digitais deixadas em locais de crime. Podemos também utilizar as digitais em sistemas de segurança, para liberar ou negar acesso a uma área restrita (um centro de alta segurança ou um edifício, por exemplo), ou para efetuar uma operação de *login* num sistema computacional.

Quando as impressões digitais começaram a ser utilizadas para identificação de pessoas, o reconhecimento era feito apenas manualmente, o que tornava o processo muito custoso, pois era necessário um especialista nesse tipo de aplicação para realizar o reconhecimento e as impressões digitais são estruturas com muitos detalhes, o que exigia muito tempo para verificar as particularidades existentes. Devido a essas dificuldades, iniciaram-se diversos estudos de técnicas para acelerar este reconhecimento com auxílio computacional, e fazê-lo de forma automática, sem a necessidade de um especialista na área [5]. Isto serviu como motivação para a criação de diversos sistemas para identificação automática de pessoas.

O processo de *verificação* de uma impressão digital consiste em, dadas duas impressões digitais, verificar se elas são provenientes do mesmo dedo. Em particular, pode-se representar uma impressão através de uma imagem digital, e utilizar técnicas computacionais para realizar a verificação.

A Morfologia Matemática [1, 2], teoria que estuda mapeamentos entre reticulados completos, é bastante útil para modelar transformações entre imagens, e tem sido utilizada com muito sucesso para resolver diversos problemas de processamento de imagens.

A iniciação científica teve como foco o estudo de técnicas para verificação automática de impressões digitais. O objetivo do projeto é o desenvolvimento de um software de verificação de impressões digitais, utilizando técnicas da Morfologia Matemática, a ser usado em conjunto com um sensor de captura desenvolvido por uma empresa brasileira, formando assim um sistema completo de verificação de impressões digitais. Durante os últimos meses, desenvolvemos um protótipo do sistema. As próximas seções mostram a metodologia utilizada para fazer a verificação, uma análise dos resultados obtidos, alguns problemas encontrados no processo e alguns pontos que podem ser explorados em continuidade à pesquisa.

1.2 Metodologia Utilizada

1.2.1 O processo de verificação

O processo de verificação de impressões digitais pode ser dividido em algumas etapas principais:

- Aquisição das imagens: as imagens que representam as digitais a serem verificadas são obtidas através de algum dispositivo de captura, como, por exemplo, um sensor que gera imagens digitais a partir do contato de um dedo com uma superfície de leitura;

- Filtragem da imagem: nesta etapa a imagem adquirida é processada a fim de melhorar a sua qualidade e prepará-la para a extração de características;

- Extração de características: são extraídas das imagens filtradas algumas características que serão utilizadas na próxima etapa para realizar a comparação entre as digitais;

- Comparação (*Matching*): as características obtidas na etapa anterior são usadas para fazer a comparação entre as duas impressões digitais e decidir se elas são provenientes ou não do mesmo dedo.

A figura 1 proporciona uma visão geral das etapas descritas acima.

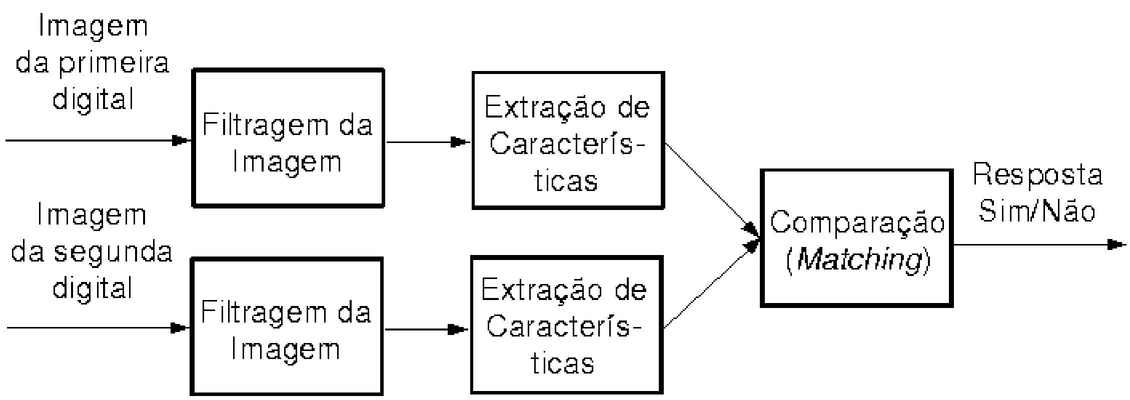

Figura 1: Visão geral do processo de verificação

1.2.2 Filtragem da imagem

Para realizar a filtragem das imagens, elaboramos uma abordagem baseada em filtros morfológicos (operadores da Morfologia Matemática). Nesta seção é feita uma descrição dos filtros utilizados, após a apresentação dos conceitos básicos necessários para a sua compreensão.

Morfologia Matemática

Inicialmente, vamos definir matematicamente uma imagem. Seja $I \subseteq \mathbb{Z}^2$ um subconjunto retangular finito de pontos, $L = [0, l]$ um intervalo fechado em $\mathbb{Z}$, e $\mathcal{F}[I, L] = \{f : I \to L\}$. Uma das formas de se definir uma imagem é através de uma função $f \in \mathcal{F}[I, L]$. Se $l = 1$, temos o caso particular das imagens binárias, onde cada pixel (um dos pontos de I) só pode assumir os valores 0 e 1.

Uma imagem binária pode também ser definida como um subconjunto $E \in \mathcal{P}(I)$ tal que E contém apenas os pixels p de I para os quais $f(p) = 1$, onde $\mathcal{P}(I) = \{X : X \subseteq I\}$, ou seja, $\mathcal{P}(I)$ é o conjunto dos subconjuntos de I. Além disso, é possível demonstrar que existe uma correspondência um-por-um entre $\mathcal{P}(I)$ e $\mathcal{F}[I, L]$ para $L = [0, 1]$, sendo então equivalentes as representações de imagens binárias como funções ou como subconjuntos [1].

A linguagem formal denominada *linguagem morfológica* é composta pelos operadores de *erosão* e *dilatação*, e pelas operações de *supremo*, *ínfimo*, *negação* e *composição*. Um *operador morfológico* pode ser então definido por uma frase nessa linguagem [1].

Sejam $f, g \in \mathcal{F}[I, L]$. Os operadores e operações da linguagem morfológica são então definidos:

- Operadores:
  - Erosão: $\epsilon_B(f(x)) = \bigwedge_{y \in B} f(x + y)$
  - Dilatação: $\delta_B(f(x)) = \bigvee_{y \in B} f(x - y)$

Onde $B \subseteq \mathbb{Z}^2$ é denominado elemento estruturante.

- Operações:
  - Supremo: $f \vee g = max\{f(x), g(x)\}$
  - Ínfimo: $f \wedge g = min\{f(x), g(x)\}$
  - Negação: $\neg f(x) = l - f(x)$, onde $l = max(L)$
  - Composição: $\psi_1\psi_2(f) = \psi_1(\psi_2(f))$, onde $\psi_1 : \mathcal{F}[I, L] \rightarrow \mathcal{F}[I, L]$ e $\psi_2 : \mathcal{F}[I, L] \rightarrow \mathcal{F}[I, L]$ são dois operadores.

No caso de imagens binárias, a operação de supremo é equivalente à união de conjuntos, a operação de ínfimo é equivalente à intersecção de conjuntos e a operação de negação é equivalente ao complemento de um conjunto.

Para a compreensão das etapas apresentadas nas próximas seções, também são importantes as definições dos seguintes operadores [1, 2]:

- Abertura: $\gamma_B(f) = \delta_B(\epsilon_B(f))$;
- Fechamento: $\phi_B(f) = \epsilon_B(\delta_B(f))$;
- Hit-miss: $\lambda_{A,B}(f) = \epsilon_A(f) \wedge \epsilon_{\neg B}(\neg f)$, com $A \subseteq B$ (usado para reconhecer padrões em imagens);
- Afinamento: $\sigma_{A,B}(f) = f - \lambda_{A,B}(f)$.

Binarização

O filtro *threshold* de parâmetro t aplicado a uma imagem f é definido como

$$T(f,t)(x) = \begin{cases} 1, & se\ f(x) \geq t \\ 0 & c.c. \end{cases}$$

Seja o operador morfológico de *abertura por reconstrução* [2] aplicado à imagem f. Tal operador é composto por:

- erosão de f, resultando numa imagem g;
- inf-reconstrução da imagem f a partir de g, isto é, a dilatação infinita de g condicionada à imagem f (a cada iteração, faz-se uma dilatação na imagem seguida de uma intersecção com f, até que a dilatação não modifique mais a imagem).

O operador *top-hat da abertura por reconstrução* aplicado a uma imagem f é definido como sendo a subtração $f - \gamma_{rec}(f)$, onde $\gamma_{rec}(f)$ denota a abertura por reconstrução de f.

A aplicação do *top-hat da abertura por reconstrução* à imagem em níveis de cinza gerada pelo sensor tem um efeito que pode ser explicado pela seguinte analogia: suponha que a imagem em níveis de cinza seja equivalente a uma superfície onde, quanto maior for o nível de cinza, maior é a altitude. A aplicação do operador sobre tal superfície resulta numa superfície rebaixada, onde todas as altitudes ficam próximas de zero mas os pontos de maior altitude continuam mais altos que os demais.

Assim, a subseqüente aplicação de um operador *threshold* de parâmetro baixo (1, por exemplo) resulta numa imagem binária que representa as linhas da impressão digital.

Extração da região de interesse

As imagens geradas pelo sensor de captura nem sempre são ocupadas em sua totalidade pela impressão digital. Por exemplo, os cantos da imagem podem estar vazios (o que é bastante comum, devido à forma dos dedos). É denominada *região de interesse* da imagem a região que efetivamente contém as linhas que representam a impressão digital. Nesta etapa do processamento da imagem, um filtro é aplicado com o propósito de detectar automaticamente a região de interesse. A determinação da máscara (uma imagem binária que define a região de interesse) é feita a partir de g, a imagem em níveis de cinza da impressão digital:

- $op = \gamma_{B_e(r)}(g)$, onde $B_e(r)$ é um disco euclidiano de raio r, ou seja, $B_e(r) = \{(x_1, x_2) \in \mathbb{Z}^2 : \sqrt{x_1^2 + x_2^2} \leq (r + 0.5)\}$;
- th = Threshold de op com parâmetro t;
- n = Negação de th;
- $asf = \phi_{B_e(q)}(\gamma_{B_e(q)}(n))$;
- $op_2 = \gamma_H(\gamma_V(asf))$, onde H é uma linha na horizontal e V uma linha na vertical;
- $mask$ = Intersecção entre a negação da moldura da imagem e op_2. A eliminação da moldura da imagem é uma heurística que traz bons resultados, já que a probabilidade de termos regiões do fundo da imagem próximas às bordas é muito maior do que no restante da imagem.

Tendo então a máscara, basta fazer uma intersecção entre esta e a imagem binária gerada na etapa anterior para obter a região de interesse. Para as nossas imagens de teste, os valores $r = 6$, $t = 127$, $q = 7$, comprimento de H e V igual a 40 e largura da moldura considerada igual a 10 pixels produzem excelentes resultados.

Filtragem de ruído

As imagens geradas na etapa de binarização apresentam uma quantidade razoável de ruído na forma de pequenos grãos brancos ou pequenos poros pretos. Para filtrar este ruído, utilizamos os filtros morfológicos *area open* e *area closing* [2], que eliminam componentes conexos de área menor que um parâmetro a ou fecham poros de área menor que um dado parâmetro b, respectivamente.

Extração do esqueleto

Nesta etapa, o objetivo é transformar a imagem num mapa de linhas de espessura 1 que represente fielmente as linhas da imagem original, de modo que a homotopia [1, 2] (disposição relativa dos componentes conexos da imagem) seja preservada. Tal representação denomina-se *esqueleto*. Para chegar a este resultado, primeiramente é aplicada uma seqüência infinita de afinamentos (a aplicação do operador é interrompida quando a imagem não é mais modificada):

$$\sigma^{\infty}_{\Delta\theta,A,B}(f) = \sigma_{\Delta\theta,A,B}(\sigma_{\Delta\theta,A,B}(\cdots\sigma_{\Delta\theta,A,B}(f)))$$

$$\sigma_{\Delta\theta,A,B}(f) = \sigma_{315}(\sigma_{270}(\sigma_{225}(\sigma_{180}(\sigma_{135}(\sigma_{90}(\sigma_{45}(\sigma_0(f))))))))$$

$$\sigma_\theta(f) = \sigma_{A_\theta,B_\theta}$$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Onde A_θ e B_θ denotam as rotações de A e B em θ graus no sentido horário.

Porém, a imagem obtida após a aplicação do operador acima ainda pode possuir linhas de espessura maior que 1. Para resolver isto, em seguida utilizamos o operador denominado *esqueleto por afinamento filtrado* [1].

Filtragem do esqueleto

O esqueleto gerado na etapa anterior apresenta algumas "rebarbas" que não fazem parte da geometria original das linhas da imagem. A filtragem realizada nesta etapa tem o objetivo de eliminar tais "rebarbas".

A abordagem adotada para filtrar o esqueleto consiste em primeiro aplicar o filtro de poda (*pruning*) [2] com n iterações:

$$\sigma^n_{\Delta\theta,A,B}(f) = \underbrace{\sigma_{\Delta\theta,A,B}(\sigma_{\Delta\theta,A,B}(\cdots\sigma_{\Delta\theta,A,B}(f)))}_{n}$$

$$\sigma_{\Delta\theta,A,B}(f) = \sigma_{315}(\sigma_{270}(\sigma_{225}(\sigma_{180}(\sigma_{135}(\sigma_{90}(\sigma_{45}(\sigma_0(f))))))))$$

$$\sigma_\theta(f) = \sigma_{A_\theta,B_\theta}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Onde A_θ e B_θ denotam as rotações de A e B em θ graus no sentido horário.

Em seguida os pontos isolados são eliminados pela aplicação de um afinamento usando elementos estruturantes

$$C = D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Grande parte do ruído é filtrado com este procedimento, porém pedaços importantes do esqueleto também são perdidos com o *pruning*. Para contornar isso, é realizada uma reconstrução do esqueleto original a partir dos pontos extremos detectados no esqueleto podado (esta detecção pode ser feita através da aplicação de um filtro *hit-miss* com os elementos estruturantes A e B usados durante a poda). A reconstrução é uma inf-reconstrução, já citada anteriormente, porém com um número limitado (n) de iterações de dilatação.

Um problema que surge nesta etapa está relacionado com o elemento estruturante utilizado para fazer a reconstrução. Se é utilizada uma cruz (representada pela matriz P abaixo), nas regiões do esqueleto que são 8-conectadas [1], isto é, nas regiões onde os pixels vizinhos podem estar tanto na horizontal quanto na diagonal não haverá reconstrução se existirem apenas vizinhos na diagonal. Por outro lado, se é utilizada uma caixa (representada pela matriz Q abaixo), uma quantidade maior de ruído já eliminado pela poda pode ser reconstruída.

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Neste ponto, termina a etapa de filtragem da imagem. A figura 2 ilustra as etapas do processo.

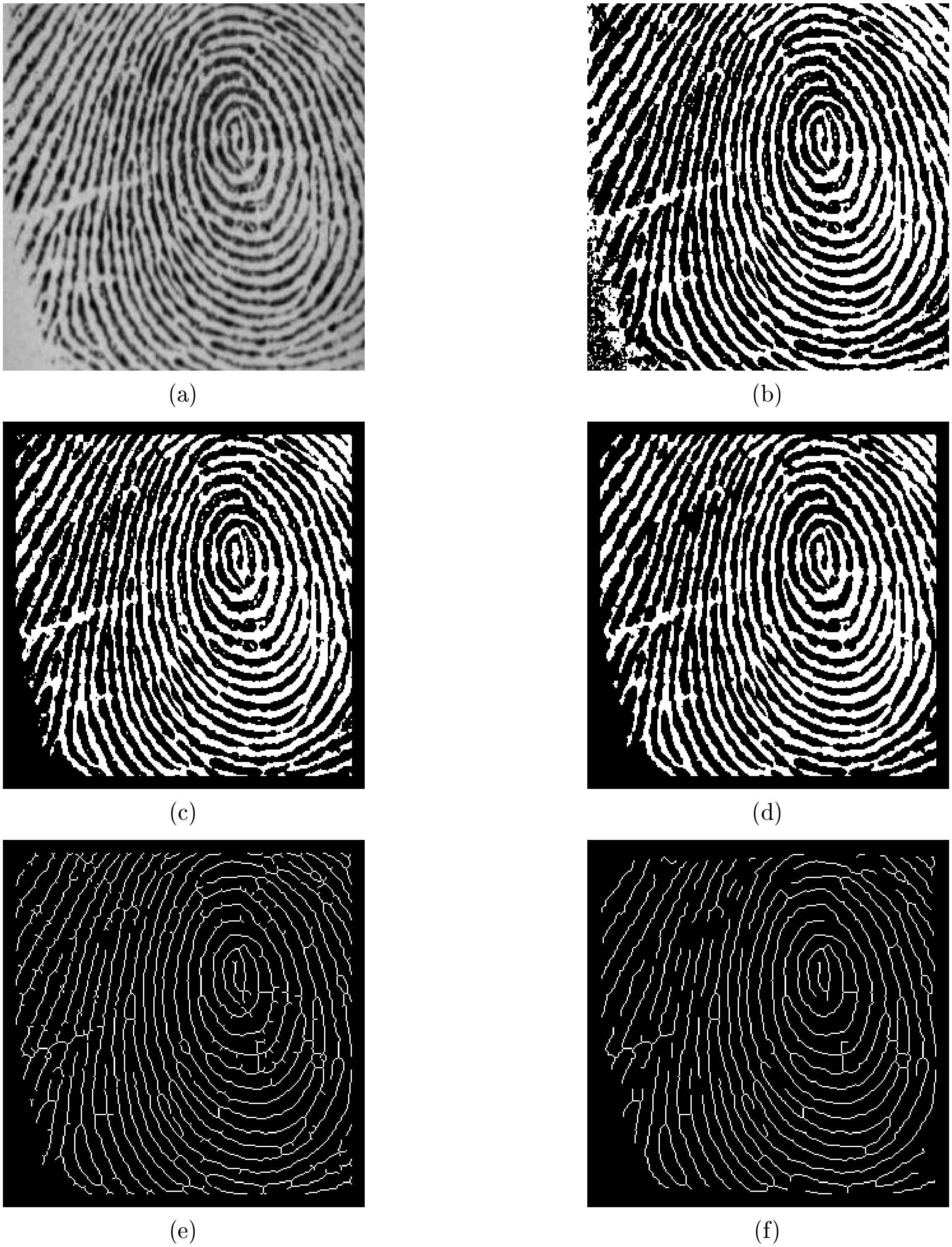

Figura 2: Etapas da filtragem: (a) Imagem em níveis de cinza; (b) Imagem binária; (c) Região de interesse; (d) Filtragem de ruído; (e) Esqueleto; (f) Esqueleto filtrado.

1.2.3 Extração de Características

Nosso protótipo utiliza as estruturas conhecidas como *minúcias* [5], que podem ser definidas como descontinuidades locais no padrão de uma impressão digital, para determinar se duas digitais são iguais.

Na literatura são encontrados muitos tipos de *minúcias* [4], mas os mais usados são as terminações de linha e as bifurcações, já que os outros tipos podem ser expressos como combinações desses dois tipos. Uma minúcia pode ser caracterizada pelo seu tipo (terminação de linha ou bifurcação), suas coordenadas (x, y) na imagem, onde x é o número da linha e y o número da coluna onde está localizada, e pela sua orientação θ (ângulo da direção da minúcia em relação à horizontal). A figura 3 mostra os tipos de minúcias mais usados.

Figura 3: Principais tipos de minúcias

Nesta etapa do processo de verificação, operadores *hit-miss* são usados para detectar os pontos onde ocorre uma terminação de linha ou uma bifurcação. É necessário criar uma família de pares de elementos estruturantes (A_i, B_i) tais que cada configuração possível de pixels numa janela 3 x 3 da imagem que forme uma bifurcação ou uma quebra de linha esteja representada por um dos pares da família, a menos de rotações. Então, as ocorrências de terminações de linha e bifurcações no esqueleto serão dadas por $\bigcup \lambda_{A_i,B_i}$, considerando todos os valores de i e todas as rotações em intervalos de 45 graus dos pares (A_i, B_i).

Por exemplo, o operador definido por $\bigcup \lambda_{A_1^\theta, B_1^\theta}$,para $\theta = 0, 45, ..., 315$, e

$$A_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

com A_1^θ e B_1^θ denotando as rotações de A_1 e B_1 em θ graus no sentido horário detecta o tipo de bifurcação representado por A_1 e B_1.

A figura 4 mostra os pontos de bifurcação detectados pelo nosso protótipo numa imagem contendo um esqueleto.

1.2.4 Comparação (*Matching*)

Na última etapa do processo de verificação, as minúcias detectadas são utilizadas para realizar a comparação entre as duas impressões digitais. Utiliza-se um algoritmo de comparação similar ao proposto em [7], que é invariante à translação, rotação e a pequenas distorções entre as impressões digitais. Resumidamente, o algoritmo é composto por uma fase de normalização e uma fase de reconhecimento. Na fase de normalização, a idéia é determinar parâmetros de normalização (um vetor de translação, um ângulo de rotação e dois pontos centrais, um na primeira imagem e um na segunda imagem, que serão usados para "equiparar" uma imagem à outra). Para isto, para cada minúcia presente nas imagens é definida uma *característica local* composta pelo tipo da própria minúcia (que será o ponto central da característica) além dos tipos das 5 minúcias mais próximas e dos ângulos relativos entre os 5 vetores determinados pelas minúcias mais próximas e a minúcia central. A figura 5 ilustra uma *característica local*.

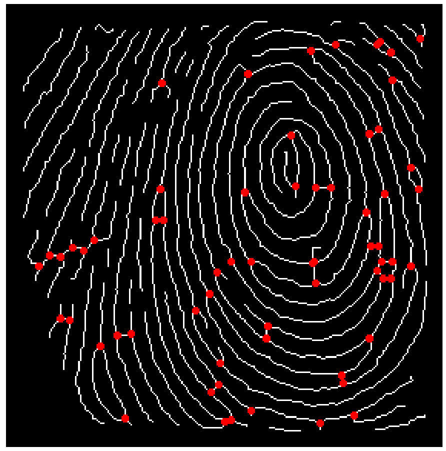

Figura 4: Pontos de bifurcação

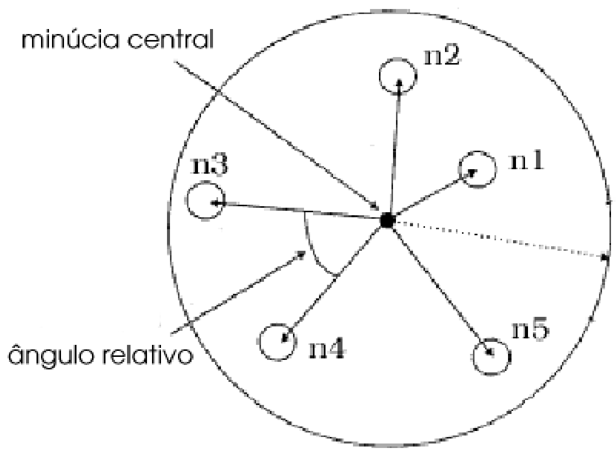

Figura 5: Característica local

Então, para comparar as características locais das imagens deve-se considerar os pares (s_i, t_j), tais que $1 \leq i \leq m$ e $1 \leq j \leq n$, onde m e n são, respectivamente, os números de minúcias presentes na primeira e na segunda imagem, s_i é a característica local cujo centro é a i-ésima minúcia da primeira imagem e t_j é a característica local cujo centro é a j-ésima minúcia da segunda imagem. Para cada um desses pares, os tipos de minúcias e os ângulos relativos são comparados e uma pontuação (*matching score*) é atribuída ao par. Se o tipo das minúcias centrais for igual, o *matching score* é igual ao número máximo de ângulos relativos seqüenciais iguais para o par. Caso contrário, é zero. As minúcias centrais dos pares com maior *matching score* serão usadas como centro da imagem na etapa de reconhecimento. Pode-se filtrar o conjunto de pares com maior *matching score* para diminuir o número de candidatos para a etapa do reconhecimento, comparando outras características das minúcias mais próximas (distância ao centro e orientação, por exemplo) e eliminando os pares que não coincidirem usando tais características.

Seja (s_k, t_l) um dos pares de maior *matching score*. Calcula-se então um vetor de translação que é a diferença entre as coordenadas do ponto central de s_k e as coordenadas do ponto central de t_l, e um ângulo de rotação baseado em como os ângulos relativos de s_k e t_l coincidiram durante o *matching*. Aplica-se essas transformações aos pontos da segunda imagem, tomando o cuidado de realizar a rotação em torno do ponto central de t_l, o que fará com que a segunda imagem se "sobreponha" à primeira tendo os pontos centrais de s_k e t_l como referenciais centrais. A este processo dá-se o nome de normalização.

Após a normalização, é feita uma intersecção entre as duas imagens para detectar os pontos cujas coordenadas coincidem. Para que exista uma tolerância a pequenas distorções nas imagens, antes da intersecção é feita uma dilatação por um disco euclidiano na primeira imagem, de modo que um ponto na primeira imagem cujas coordenadas sejam diferentes mas muito próximas das de um ponto na segunda imagem seja incluído na intersecção. O reconhecimento consiste em, para cada ponto p presente na intersecção, comparar as características do par de pontos (q, r), com q pertencente à primeira imagem e r pertencente à segunda imagem, que fez com que p estivesse presente na intersecção. Tais características podem ser, por exemplo, o tipo da minúcia, sua orientação ou o número de transições entre preto e branco que ocorrem no esqueleto da imagem, ao longo da reta determinada pela minúcia em questão e a minúcia central. O número de pontos cujas características coincidem é então determinado, e se tal número for maior que um determinado limiar de aceitação o algoritmo responde que as duas impressões digitais são provenientes do mesmo dedo, caso contrário responde que não. Diferentes países definiram o número mínimo de minúcias coincidentes necessárias de acordo com seus próprios padrões [8]. Por exemplo, nos Estados Unidos o limiar vai de 7 a 12, já em Cingapura o valor do limiar utilizado é 16.

O algoritmo de *matching* descrito acima foi inspirado na maneira como os especialistas em reconhecimento de impressões digitais abordam o problema [7]. Primeiramente, eles tentam encontrar correspondências entre pequenas porções das imagens (o equivalente ao *matching* usando características locais, resultando nos parâmetros de normalização), para depois de estabelecer um ponto central de referência para as duas imagens, fazerem o reconhecimento observando a imagem de forma global e comparando outras características (tipo de minúcia, orientação, etc.)

A figura 6 ilustra em linhas gerais o funcionamento do algoritmo de *matching*.

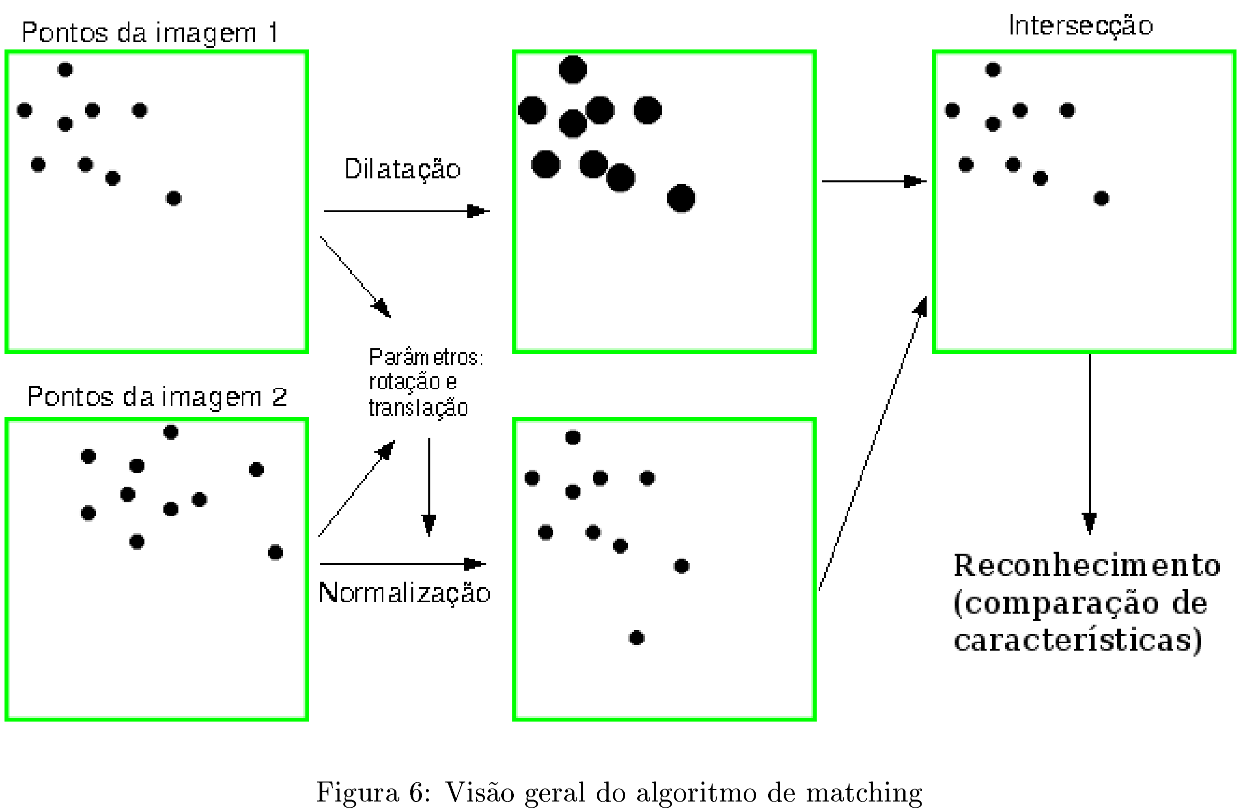

Figura 6: Visão geral do algoritmo de matching

1.3 O protótipo desenvolvido

O protótipo do sistema de verificação de impressões digitais foi implementado na linguagem MATLAB, usando a SDC Morphology Toolbox [3] nas etapas que envolvem a utilização de ferramentas da Morfologia Matemática.

O programa em MATLAB implementa todas as etapas do processo descrito acima, com exceção da etapa de aquisição das imagens. Para tal etapa, existe um programa escrito em C que executa em ambiente Windows e um driver em formato DLL, fornecidos pela empresa que desenvolveu o sensor, que são responsáveis pela interação com o dispositivo de captura e digitalização das impressões digitais.

A metodologia implementada foi em essência a descrita anteriormente, com alguns ajustes de parâmetros feitos para que o algoritmo se adaptasse melhor às imagens geradas pelo sensor, já que futuramente o protótipo poderá evoluir para um software a ser utilizado em conjunto com este sensor. Como exemplo, podemos citar a escolha de parâmetros do algoritmo de detecção da região de interesse.

Em relação ao problema apontado na etapa de reconstrução do esqueleto, optamos por utilizar a cruz como elemento estruturante. Esta decisão deveu-se ao fato de na etapa de *matching* estarmos utilizando, em caráter experimental, apenas bifurcações como características. As terminações de linhas não estão sendo utilizadas porque a quantidade de minúcias espúrias (isto é, presentes no esqueleto mas não presentes na imagem original) deste tipo mostrou-se muito grande em nossos experimentos. Desta forma, reduz-se o número de bifurcações espúrias nos esqueletos gerados pela etapa de filtragem da imagem.

O número mínimo de coincidências entre minúcias necessárias na etapa de reconhecimento foi fixado com o valor 7, já que, como estamos utilizando apenas as bifurcações como características, o número de *matchings* necessários numa verificação deve ser menor.

1.4 Análise dos resultados obtidos

Foi criado um pequeno banco de dados de imagens de impressões digitais capturadas pelo sensor, com a finalidade de realizarmos testes e experimentações com algoritmos. O sensor gera imagens em níveis de cinza (com 8 bits de profundidade) com resolução de 224 x 288 pixels e as transmite a um computador através de uma porta USB.

Obtivemos as impressões digitais de 29 indivíduos. Para cada pessoa, foram lidas 16 impressões, todas do mesmo dedo e em momentos diferentes.

Para estimar a porcentagem de acerto na verificação executamos, para cada pessoa, o seguinte procedimento:

- escolhe-se uma das 16 imagens;
- realiza-se a verificação da imagem escolhida com cada uma das 16 imagens da mesma pessoa e conta-se o número de acertos.

Portanto, no total foram feitas 464 verificações. Destas 464 verificações, 263 retornaram a resposta correta, resultando assim numa taxa estimada de acerto de 56,68%.

Para estimar a taxa de falsa aceitação (responder que duas digitais são provenientes do mesmo dedo quando na verdade são de dedos diferentes), foram escolhidos aleatoriamente no banco de dados 406 pares de digitais, onde cada par é composto por duas digitais de pessoas diferentes. O número de respostas “sim” obtido foi igual a 185, portanto a taxa de falsa aceitação estimada foi igual a 45,57%.

Através da análise manual das imagens, verificamos que esqueletos contendo muitas minúcias espúrias interferem negativamente no resultado, podendo levar o algoritmo a fornecer uma resposta incorreta.

Em especial, pontos onde as linhas se apresentam muito próximas nas imagens podem gerar pontos de conexão nas imagens filtradas, e conseqüentemente bifurcações espúrias serão detectadas na etapa de extração de características.

1.5 Atividades realizadas

A iniciação científica teve seu início em agosto de 2002. No período de agosto a dezembro de 2002, eu ainda não estava envolvido com o projeto de reconhecimento de impressões digitais, e o foco da iniciação científica estava no estudo dos conceitos básicos de aprendizado computacional, com aplicações em processamento de imagens. Em janeiro de 2003, iniciei minha participação no projeto sobre reconhecimento de impressões digitais. O período de janeiro a julho foi composto principalmente por estudos teóricos, enquanto o período de agosto a novembro foi destinado ao desenvolvimento do protótipo do sistema de verificação de impressões digitais. As principais atividades realizadas foram:

- Agosto a dezembro de 2002:
  - Leitura de material sobre aprendizado computacional, em especial [9].
- Janeiro a julho de 2003:
  - Levantamento bibliográfico de material específico sobre reconhecimento de impressões digitais;
  - Familiarização com o laboratório de visão e processamento de imagens do IME e com os programas disponíveis;
  - Leitura de artigos sobre reconhecimento de impressões digitais;
  - Levantamento de características observadas nas imagens nos métodos de reconhecimento;
  - Leitura de livros sobre Morfologia Matemática: [1] e [2];
  - Estudo das funções da SDC Morphology Toolbox for MATLAB [3].
- Agosto a novembro de 2003:
  - Familiarização com a linguagem de programação MATLAB;
  - Elaboração da metodologia utilizada;
  - Aquisição de imagens para o banco de dados de teste;
  - Desenvolvimento do protótipo: implementação, experimentação com algoritmos, ajustes na metodologia.

1.6 Conclusão e Considerações Finais

No atual estágio de nossa pesquisa, temos um protótipo de um sistema de verificação de impressões digitais que implementa o processo completo de verificação. Porém, sua acurácia ainda não é apropriada para utilização em aplicações reais, devido à alta taxa de erro apresentada nos testes.

Em continuidade à pesquisa, alguns pontos ainda podem ser explorados a fim de melhorar os índices de acerto do protótipo. Como os resultados do *matching* dependem diretamente dos pontos detectados na etapa de extração de características, que por sua vez dependem dos resultados da filtragem da imagem, um refinamento nas estratégias empregadas para filtrar a imagem no sentido de diminuir o número de minúcias espúrias presentes no esqueleto gerado teria forte impacto na qualidade dos resultados obtidos no *matching*. Se os pontos de terminação de linhas encontrados no esqueleto corresponderem aos verdadeiros pontos encontrados na imagem original, ao contrário do que ocorre no protótipo atual, tais pontos poderão ser usados também para fazer o reconhecimento, o que resultará numa maior discriminabilidade entre as impressões digitais. Outra possível alteração envolveria modificações no procedimento de *matching* para que este seja menos sensível ao ruído proveniente de etapas anteriores.

A utilização de técnicas de *aprendizado computacional* [9] para projetar filtros melhores também é uma alternativa a ser considerada.

Finalmente, quando o protótipo estiver apresentando taxas aceitáveis de erro, para que este se transforme num sistema completo de verificação de impressões digitais pelo menos duas etapas seriam necessárias:

- Otimização: a razão principal de escolha do MATLAB como plataforma de implementação do protótipo foi a flexibilidade proporcionada para a experimentação e teste de novas idéias de forma bastante rápida. Porém, em termos de eficiência computacional o MATLAB não é a opção mais apropriada para a nossa aplicação. Portanto, seria necessário fazer uma tradução do código em MATLAB para alguma linguagem mais adequada, como C, por exemplo.

- Integração: o software responsável pela etapa de aquisição das imagens precisará ser integrado ao software das outras etapas. Entre as funcionalidades previstas, deverão estar presentes um módulo de cadastramento de usuários autorizados, cuja finalidade seria registrar num banco de dados as impressões digitais ou informações extraídas delas, e um módulo de reconhecimento, responsável por adquirir uma imagem de uma digital usando o sensor e fazer a verificação com a digital apropriada pré-armazenada no banco de dados.

2 Experiência Pessoal

2.1 Desafios e frustrações

Durante a realização do projeto de iniciação científica, muitos desafios foram encontrados. A própria participação no projeto já constitui um grande desafio, por ser um projeto complexo e por eu estar entrando em contato pela primeira vez com o ambiente acadêmico de pesquisa.

Ao iniciar o projeto, eu não tinha conhecimento sobre o problema. Foi então necessário fazer um levantamento bibliográfico de material específico sobre reconhecimento de impressões digitais, para se ter uma idéia do estado da arte. Felizmente, muitos artigos foram encontrados, visto que a identificação por impressões digitais é o tipo de identificação mais estudado até hoje em biometria, embora o problema ainda não esteja resolvido completamente.

O estudo mais aprofundado de morfologia matemática (durante as disciplinas da área de visão e processamento de imagens os professores falam um pouco sobre isso, mas apenas os conceitos básicos, já que existe uma disciplina de pós-graduação que trata exclusivamente do assunto), o aprendizado de uma linguagem de programação que eu quase não havia utilizado durante o curso (o MATLAB) e a familiarização com a SDC Morphology Toolbox for MATLAB também foram desafios que precisaram ser vencidos antes de iniciar a implementação do sistema. A elaboração de uma metodologia de verificação de impressões digitais baseada quase que exclusivamente em morfologia matemática também é uma tarefa bastante desafiadora, visto que na literatura são poucos os artigos que fazem uso de operadores morfológicos para resolver o problema, e ainda assim os utilizam em apenas algumas partes do processo.

Em aspectos não técnicos, um grande desafio enfrentado foi a dificuldade de organização do tempo para conciliar as atividades das disciplinas do BCC com as atividades da iniciação científica. Muitas vezes era necessário interromper as atividades do projeto para me dedicar às disciplinas, ou deixar um pouco as disciplinas de lado para tirar o atraso no projeto. Felizmente, a existência das semanas de "break" do BCC ajuda muito nesse aspecto.

No planejamento inicial do projeto, estava previsto que ao final do trabalho um sistema completo de verificação de impressões digitais estaria finalizado. Porém, durante o andamento do trabalho, o projeto mostrou ser de complexidade maior do que a esperada, e o tempo disponível para a sua realização não seria suficiente para chegar ao objetivo proposto inicialmente. Então, meu objetivo passou a ser o de ter um protótipo completo funcionando com taxas de acerto e falsa aceitação adequadas para uma aplicação real, mas ainda sem a integração discutida anteriormente na parte técnica e sem muitas preocupações com eficiência computacional. Isto foi, de certa forma, uma frustração.

Atualmente, temos um protótipo que implementa o processo completo de verificação, como mostrado anteriormente. Porém, as taxas de acerto e de falsa aceitação do protótipo ainda estão longe de serem ideais. Outra frustração que tive foi a de não ter conseguido melhorar o protótipo de forma que as taxas sejam aceitáveis para utilização em aplicações reais, devido ao pouco tempo disponível para isso.

2.2 As disciplinas do BCC e o projeto

Várias disciplinas do curso de Bacharelado em Ciência da Computação tiveram importância na realização do projeto de iniciação científica. Dentre as mais importantes, podem ser citadas:

- MAC417 - Visão e Processamento de Imagens - Parte I: neste curso pude aprender os conceitos básicos da área de Visão e Processamento de Imagens, na qual o projeto se enquadra;
- MAC447 - Análise e Reconhecimento de Formas: diversos conceitos abordados no curso foram úteis no decorrer do projeto. Cursando esta disciplina também pude compreender melhor alguns conceitos vistos na disciplina MAC417;
- MAC420 - Introdução à Computação Gráfica: embora o projeto não envolva computação gráfica, diversos algoritmos geométricos vistos no curso foram úteis na implementação do algoritmo de *matching*;
- MAC329 - Álgebra Booleana e Aplicações: disciplina de fundamental importância para a compreensão da teoria da Morfologia Matemática;
- MAC414 - Autômatos e Linguagens Formais: nesta disciplina também são vistos alguns conceitos importantes em Morfologia Matemática, como as gramáticas e linguagens formais;

Além destas disciplinas, também foram bastante importantes os Laboratórios de Programação I e II, por exigirem projetos maiores que os de outras disciplinas do BCC, fornecendo ao aluno a prática necessária para trabalhar num projeto maior e que envolva bastante programação; e as disciplinas introdutórias de computação, cujos conceitos formam a base para os estudos nas outras disciplinas do BCC.

Quanto à minha formação, considero que todas as disciplinas do curso tiveram alguma importância, com exceção de Física I e II, que têm integração questionável com as demais disciplinas. O BCC procura oferecer uma formação genérica, provendo uma visão geral das várias áreas da Ciência da Computação e preparando o aluno para ter a capacidade de se aprimorar nas áreas em que houver maior interesse ou necessidade. Sendo assim, considero excelente a formação oferecida pelo BCC.

Um ponto importante a ser observado é a diferença existente entre o trabalho desenvolvido no projeto de iniciação científica e nos projetos e EPs exigidos em disciplinas do curso. A complexidade dos projetos e exercícios-programa é menor, e na sua realização em geral os alunos devem seguir um enunciado bem definido, para permitir que os exercícios sejam feitos em curtos períodos de tempo, adequados à duração das disciplinas, e para que seja possível para os professores ou monitores fazerem a correção de uma grande quantidade de EPs em pouco tempo. Já o projeto de iniciação científica mostrou-se muito mais complexo, tanto na implementação quanto em relação à quantidade de estudo necessário antes de iniciá-la. Além disso, existe uma flexibilidade maior nos caminhos tomados para chegar ao objetivo final, devido à própria natureza da pesquisa, que envolve experimentação e possível redefinição dos métodos com base nos resultados obtidos. Enfim, é natural que existam essas diferenças, já que os objetivos dos dois tipos de atividades são diferentes: nas disciplinas, o objetivo dos EPs e projetos é contribuir para a fixação dos conceitos apresentados e utilizá-los de forma prática, como complemento à teoria vista em sala de aula, enquanto na iniciação científica o objetivo é apresentar ao aluno o ambiente de pesquisa acadêmica, o que envolve a busca de metodologias para a resolução de problemas já existentes ou a apresentação de novos problemas.

Através da participação no projeto de iniciação científica foi também possível notar como diversos conceitos vistos no curso são utilizados em aplicações reais. Em particular, o meu projeto pode futuramente se tornar um produto utilizado em diversas aplicações. Os conceitos envolvidos no projeto foram principalmente os relacionados à área de Visão e Processamento de Imagens; porém, em fases futuras do desenvolvimento, conceitos de outras áreas virão a ser necessários. Por exemplo, para transformar o protótipo num software completo de verificação de impressões digitais, provavelmente conceitos de Engenharia de Software e Sistemas Operacionais seriam importantes, entre outros.

2.3 Interação com o orientador e outros pesquisadores

Na fase inicial de minha iniciação científica, que foi uma fase de estudos, eu tinha reuniões semanais com o professor Junior Barrera, meu orientador neste projeto, para tirar dúvidas e discutir as teorias estudadas. Já na fase de implementação do sistema, inicialmente definimos as primeiras etapas da metodologia e, a cada parte que eu completava, havia uma nova reunião para analisar os resultados obtidos, reformular o método se necessário e definir as próximas etapas. Muitas vezes essas reuniões foram feitas no laboratório de visão e processamento de imagens do IME em frente a um computador, de forma que pudéssemos testar imediatamente novas idéias e visualizar os resultados. Esta interação foi bastante rica e muito importante para o meu trabalho. Nas reuniões, tive a oportunidade de conhecer melhor os detalhes relacionados à pesquisa acadêmica, além de aprender bastante coisa relacionada aos problemas práticos que aparecem em aplicações reais de análise de imagens, aplicações com as quais meu orientador possui bastante experiência. O Junior também me auxiliou bastante na utilização da SDC Morphology Toolbox; tendo participado do desenvolvimento dessa *toolbox*, ele pôde me passar dicas valiosas.

Embora alguns dos professores do grupo de visão e processamento de imagens do IME tenham se disponibilizado para trocar experiências e tirar dúvidas sobre os assuntos relacionados ao projeto, tive poucas conversas com eles, mas as que tivemos foram bastante interessantes. As conversas que tive com outros alunos que desenvolvem projetos no laboratório de visão e processamento de imagens também foram bastante proveitosas para conhecer os diversos trabalhos realizados no instituto, além de trocar informações que de alguma forma foram úteis no decorrer do projeto.

2.4 Considerações finais e outras experiências relevantes

Considero que o período de realização da iniciação científica foi uma etapa muito importante para a minha formação. Na iniciação científica foi possível aprender conceitos que serão bastante úteis no meu projeto de mestrado, além de exercitar os conceitos estudados através da implementação de uma aplicação real. Além disso, pude entrar em contato com o ambiente acadêmico de pesquisa e compreender melhor a metodologia científica, além de conhecer os tipos de desafios que um pesquisador encontra à sua frente.

Outra experiência que tive durante a graduação e que enriqueceu muito a minha formação foi o período de dois anos em que tive a oportunidade de trabalhar como administrador da Rede GNU/Linux dos alunos de graduação do IME. Além de aprender muito sobre sistemas operacionais e redes de computadores, áreas pelas quais também tenho grande interesse, no dia-a-dia da administração fazem-se presentes muitas coisas que são importantes numa empresa. Como exemplos, podemos citar o trabalho em equipe, planejamento de tarefas, políticas administrativas, interação com outros setores do Instituto, etc. Certamente esta experiência contribuiu significativamente para o meu desenvolvimento humano e profissional.

Somando-se a isso a excelente formação oferecida através das disciplinas do BCC, posso afirmar que os quatro anos que passei no IME durante a graduação foram muito bons. Em continuidade aos meus estudos, estarei iniciando em 2004 um mestrado em Ciência da Computação no mesmo instituto, em que trabalharei num projeto de reconhecimento de gestos em vídeo digital.

Os pontos que podem ser explorados em continuidade ao projeto de iniciação científica são os que foram discutidos anteriormente na parte técnica desta monografia. Ainda continuarei trabalhando por um tempo tentando melhorar o protótipo, com o objetivo de obter taxas de erro aceitáveis para aplicações reais.

Referências

[1] G. J. F. Banon and J. Barrera. *Bases da Morfologia Matemática Para a Análise de Imagens Binárias.* IX Escola de Computação, Recife, 1994.

[2] E. R. Dougherty and R. A. Lotufo. *Hands-on Morphological Image Processing.* SPIE, 2003.

[3] *SDC Morphology Toolbox for MATLAB* - http://www.mmorph.com/ (acessado em 01/12/2003)

[4] T. Caohuu and L. Thuy. *Finger Print Recognition: Technology and Application.* http://www.vacets.org/vtic97/thcao.htm (acessado em 01/12/2003)

[5] A. Jain, L. Hong and R. Bolle. *On-Line Fingerprint Verification.* IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 19, no. 4, April 1997.

[6] S. Pankanti, S. Prabhakar and A. Jain. *On the Individuality of Fingerprints.* IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 24, no. 8, August 2002.

[7] A. Wahab, S. H. Chin and E. C. Tan. *Novel approach to automated fingerprint recognition.* IEEE Proc. Vision, Image and Signal Processing, vol. 145, no. 3, June 1998.

[8] D. P. Mital and E. K. Teoh. *An Automated Matching Technique for Fingerprint Identification.* In *Proceedings of the 1997 First International Conference on Knowledge-Based Intelligent Electronic Systems. KES '97*, volume 1, pages 142-147, 21-23 May 1997.

[9] N. S. Tomita. *Programação Automática de Máquinas Morfológicas Binárias baseada em Aprendizado PAC.* Dissertação de Mestrado, IME-USP, fevereiro de 1996.