

Trabalho de Formatura Supervisionado - MAC
499

Fábio Takashi Hirano
Supervisor: Ronaldo Fumio Hashimoto

1 de dezembro de 2011

Sumário

I	Parte Técnica	5
1	Introdução	7
1.1	Reconhecimento de Caracteres	7
1.1.1	Segmentação	8
1.1.2	Extração de Características	8
1.1.3	Classificação	9
2	Conceitos e Tecnologias Estudadas	11
2.1	Representação de Imagens	11
2.2	Algoritmos de Extração de Características	11
2.2.1	<i>Zoning</i>	12
2.2.2	Projeções	12
2.2.3	<i>Thinning</i>	13
2.3	Pré-Processamento das Imagens	16
2.3.1	Histogramas RGB	16
3	Atividades Realizadas e Resultados	19
3.1	Testes com Projeções	20
3.1.1	Projeções com Histogramas RGB	20
4	Conclusão	23
II	Parte Subjetiva	25
5	O Curso de Ciência da Computação	27
5.1	Desafios e Frustrações	27
5.2	Disciplinas relevantes ao Trabalho	27
5.3	Daqui pra Frente...	28
6	Agradecimentos	29

Parte I

Parte Técnica

Capítulo 1

Introdução

Muitas pessoas se utilizam dos cartões de crédito e débito para efetuar suas compras. Além de mais práticos, a utilização dos cartões tornou as transações financeiras mais seguras, pois dependem de uma senha e de uma combinação de números. Com isso, os pontos de venda estão se adaptando a esse cenário, e existem cada vez mais estabelecimentos comerciais que permitem o pagamento com o uso do cartão.

Entretando, a utilização dos cartões ainda possui algumas restrições. Em geral, as máquinas utilizadas para recebimento dos pagamentos desse tipo estão atreladas a uma linha telefônica. Isso é um problema, especialmente para prestadores de serviço que não possuem posto de atendimento fixo (por exemplo, taxistas).

Esse problema poderia ser solucionado se, ao invés de usar uma máquina de cartões, o prestador se utilizasse de algum dispositivo de acesso a Internet, como um celular. Tendo acesso a rede, bastaria ter em mãos os números do cartão de crédito para, por exemplo, efetuar um pagamento. Visando facilitar esse processo, pensou-se na possibilidade de obter o número do cartão apenas com uma foto do mesmo, poupando o trabalho de ter que digitar uma sequência de números, o que poderia aumentar a possibilidade de erros.

Buscamos, para executar essa tarefa, utilizar técnicas de Visão Computacional, área da Ciência da Computação que estuda a forma como o computador “vê”, isto é, como o computador extrai informações de uma imagem e como ele as interpreta.

1.1 Reconhecimento de Caracteres

O Reconhecimento de Caracteres é uma área da Ciência da Computação que reúne aspectos de Visão Computacional, Reconhecimento de Padrões e Inteligência Artificial. Várias aplicações hoje em dia utilizam-se do reconhecimento de caracteres, tais como scanners e conversores de arquivos (por exemplo, de um arquivo *.pdf* para um *.odt*).

O Reconhecimento Óptico de Caracteres (*Optical Character Recognition*, ou OCR) é o nome dado a qualquer tecnologia que reconheça caracteres a partir de imagens computacionais. Através de um OCR, conseguimos, por exemplo, digitalizar um texto impresso. Exemplos de OCR disponíveis são o Tesseract, o

OCRFeeder e o CuneiForm.

Cada OCR se utiliza de uma sequência de algoritmos para identificar os caracteres dentro de uma imagem. O processo de reconhecimento pode ser entendido em 3 grandes etapas: Segmentação, Extração de Características e Classificação.

1.1.1 Segmentação

Por conta da complexidade das imagens digitais, é necessário um processo que identifique a localização e a forma dos objetos presentes nela. Esse processo é o que chamaremos de *segmentação*. Na segmentação, cada objeto é identificado e tem suas características extraídas na próxima etapa. Para os OCR, os objetos são os caracteres. Diversos algoritmos podem ser usados para segmentar uma imagem, entre eles Detecção de Bordas[4], Limiarização [8] e Segmentação por Regiões [8].

Para os propósitos desse trabalho, a fase da Segmentação não foi considerada. Apesar de ser uma área da Visão Computacional com muitos desafios a serem enfrentados, no escopo do problema aqui exposto a Segmentação é um problema relativamente fácil: na maior parte dos cartões de crédito e débito, a numeração está localizada na mesma região. Dessa forma, identificar a área onde estão os números do cartão, nosso objeto de estudo, é, aparentemente, uma tarefa bem comportada, de modo que os algoritmos existentes devem funcionar bem nesse caso.

Estamos considerando que, como resultado da segmentação das imagens, temos cada algarismo do cartão separado, para que possamos processá-los um a um. Isso é possível pois não existe, nos cartões de crédito, uma “fórmula” que os defina, como acontece com os números de CPF, por exemplo. Assim, podemos classificar os algarismos separadamente pois não é necessário (nem possível) fazer a verificação da autenticidade do número completo após o final do processamento.

1.1.2 Extração de Características

Extração de Características é definida em [5] como o problema de “extrair de dados brutos a informação mais relevante para propósitos de classificação, com a intenção de minimizar a variação do padrão dentro de uma mesma classe e tentando maximizar a variação do padrão entre as classes”. Nessa etapa, devemos processar uma imagem de maneira a passar ao classificador informações suficientemente relevantes para que ele possa decidir que caracter foi passado na entrada. Os algoritmos de Extração de Características visam, principalmente, minimizar o ruído, isto é, as partes da imagem que são irrelevantes (por exemplo, sombras e brilho excessivo) e eliminar possíveis ambiguidades, advindas do excesso de ruído, por exemplo.

Dentro de um OCR, a Extração de Características é a etapa que mais influencia em sua eficiência, sendo alvo de diversos estudos e pesquisas. Essa etapa será tema central desse trabalho. Em [10] são discutidos diversos algoritmos de Extração de Características, dando um bom panorama do histórico dos estudos nessa área. Alguns algoritmos discutidos no artigo foram considerados para esse trabalho, como o *Zoning*. No livro de Pitas[9] há vários algoritmos de extração de características, junto com suas implementações computacionais.

1.1.3 Classificação

Ao processar uma imagem num OCR, queremos saber qual (ou quais) caracteres estão presentes nela. Ou seja, para um OCR as *classes* são todos os caracteres possíveis no escopo do trabalho. Uma vez extraídas as características da imagem, o OCR passa esses dados para um classificador, que deve tomar a decisão sobre qual classe (no nosso caso, qual algarismo) mais se aproxima do objeto. Essa decisão é tomada após uma etapa chamada de *Treinamento*. Para treinar um classificador, passamos por ele vários vetores de características devidamente extraídas na etapa anterior e, sabendo de antemão a resposta desejada, ajustamos seus parâmetros de modo que a confiabilidade da classificação seja a maior possível. Para que o treinamento seja bem sucedido, são necessárias várias entradas em diferentes condições. Assim, o classificador conseguirá determinar quais características são relevantes para sua seleção.

Capítulo 2

Conceitos e Tecnologias Estudadas

2.1 Representação de Imagens

Para representar uma imagem computacionalmente, Gonzales [6] sugere encarar uma imagem como um vetor bi-dimensional de números. Isso permite que os algoritmos de processamento gráfico possam ser aplicados corretamente, além de facilitar sua descrição. Assim, denotaremos uma imagem como uma matriz A com m linhas e n colunas e $A(i, j)$ o elemento na linha i e coluna j , como representado abaixo:

$$A = \begin{bmatrix} A(1,1) & A(1,2) & \dots & A(1,n) \\ A(2,1) & A(2,2) & \dots & A(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ A(m,1) & A(m,2) & \dots & A(m,n) \end{bmatrix}$$

Cada *pixel* recebe um número k entre 0 e M , sendo M o número de níveis de intensidade que a imagem tem. Em geral, os computadores reservam 8 bits para a resolução de intensidade, o que nos dá um total de 256 níveis de intensidade. Assim, para a maior parte dos casos, $0 \leq k < 256$, onde 0 representa a cor preta e 255, a cor branca, com os níveis intermediários representando os tons de cinza, do mais escuro para o mais claro. Essas imagens serão chamadas de *Imagens em tons de Cinza*. Podemos ter, ainda, *Imagens Binárias*, onde cada pixel de A só pode receber dois valores: 0 para um pixel preto e 1 para um pixel branco.

2.2 Algoritmos de Extração de Características

A escolha do algoritmo de Extração de Características dentro de um OCR é um fator determinante de sua eficiência. Existem diversos métodos para extrair características, alguns dos quais foram estudados no decorrer desse trabalho.

2.2.1 Zoning

Utilizado no OCR Calera, é citado e descrito em [2] para imagens binárias e foi generalizado para imagens em tons de cinza em [10].

Dada uma imagem A , o algoritmo divide os elementos de A em k^2 zonas retangulares de mesmo tamanho. Essas zonas são dispostas sobre A de modo que cada uma tenha $\frac{m}{k}$ linhas e $\frac{n}{k}$ colunas. Podemos pensar nessa divisão em zonas como uma grade de $k \times k$ retângulos de tamanho igual colocada sobre A . Em seguida, calculamos a média dos elementos de cada zona e guardamos num novo vetor. Esse vetor será nosso vetor de características, que serão avaliadas pelo classificador.

Definindo mais precisamente, sejam $Z = \{z_1, z_2, \dots, z_{k^2}\}$ as k^2 zonas de A e seja $V[1 \dots k^2]$ o vetor de características resultante da aplicação do algoritmo de Zoning. Então, V será preenchido com

$$V[l] = \frac{1}{K} \sum_{(i,j) \in z_l} A(i,j)$$

onde $K = \frac{m}{k} \frac{n}{k}$ é o número de *pixels* por zona e $l = 1, 2, \dots, k^2$.

2.2.2 Projeções

Uma projeção de uma imagem pode ser definida, numa dada direção, como a soma dos valores de seus pixels em cada ponto da direção desejada. Por exemplo, uma X -Projeção (ou seja, uma projeção na direção do eixo X) de uma imagem $m \times n$ pode ser obtida com a fórmula

$$X[k] = \sum_{i=1}^m A(i, k)$$

para $k = 1, 2, \dots, n$.

Já uma Y -Projeção é calculada com a fórmula

$$Y[l] = \sum_{j=1}^n A(l, j)$$

para $l = 1, 2, \dots, m$.

Podemos representar as projeções através de histogramas, cuja abscissa varia entre 0 e o tamanho da imagem na direção desejada (largura para X -Projeções e altura para Y -Projeções). Na imagem 2.1, o histograma ao lado da imagem representa sua Y -Projeção. A área preta do histograma representa a soma dos níveis de cinza da imagem em cada linha. Note que a região ao centro da imagem, onde estão as cartas de baralho, é mais clara. Essa área corresponde à área do histograma com maior intensidade. Assim, se estivéssemos procurando pelas cartas de baralho na imagem, poderíamos nos concentrar na região onde o histograma possui valores maiores.

As projeções são úteis para perceber concentrações de determinados tons em regiões da imagem. Por exemplo, uma projeção com valor próximo de zero indica que a região possui alta concentração da cor preta. Picos de projeção podem indicar regiões com alto contraste. Essas regiões de alto contraste são



Figura 2.1: Exemplo de imagem e sua Y -Projeção

especialmente úteis para identificar as bordas de uma determinada imagem. Por isso, é interessante calcular não só a projeção da imagem como um todo, mas também de partes dela.

Por exemplo, na figura 2.2, as projeções foram calculadas dentro dos retângulos destacados, e seus resultados são mostrados através de um histograma. Nos retângulos amarelo e preto, foram calculadas X -Projeções, enquanto nos retângulos verde e vermelho, foram calculadas Y -Projeções.

No retângulo amarelo, é possível notar um grande pico no lado direito do histograma. Comparando com a imagem, vemos que esse pico coincide com a parte do retângulo que contém uma parte do algarismo 9. No retângulo vermelho, também há uma região mais elevada no histograma, que também coincide com a região onde se encontra o algarismo. Tendo apenas as informações dadas pelas projeções, poderíamos, por exemplo, descartar a possibilidade de o algarismo da imagem ser 0, pois suas projeções embaixo e à esquerda não poderiam ter picos tão discrepantes, uma vez que o algarismo 0 ocuparia mais espaço nos retângulos destacados.

Apesar dessa análise não ser muito elucidativa em geral, no problema dos cartões de crédito a abordagem parece interessante: há apenas 10 possibilidades para cada caractere (0 a 9) e cada um deles possui alguma característica que o distingue dos demais. Isso pode ser notado, por exemplo, nos mostradores de relógios digitais, como o da figura 2.3. Neles, existem posições pré-determinadas para as luzes que formam os algarismos. Assim, ao estudar as projeções de determinadas posições, onde sabemos que podem existir partes do algarismo estudado, podemos decidir qual o algarismo presente na imagem dada.

2.2.3 *Thinning*

Para reconhecer os algarismos de um cartão de crédito, não é necessário todo o corpo do algarismo: se tivermos apenas uma pequena linha para cada região conexa da imagem, teremos algo próximo ao esqueleto da figura. Com essa linha em mãos, podemos classificar o algarismo, eliminando possíveis erros decor-

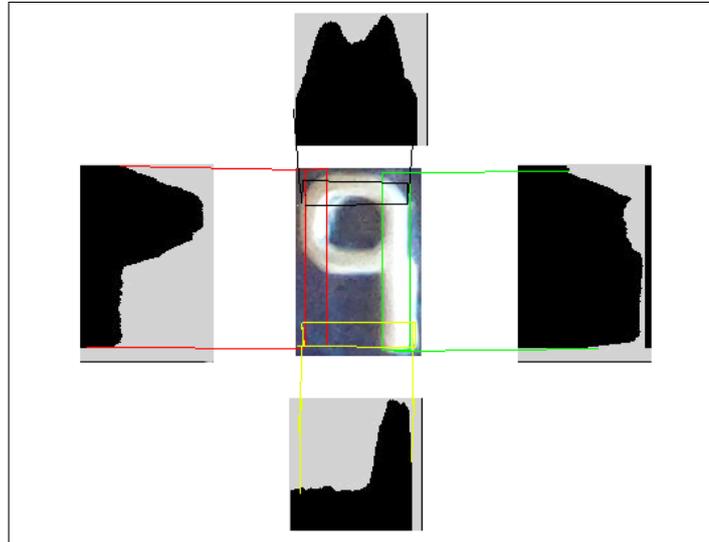


Figura 2.2: Projeções calculadas sobre a imagem



Figura 2.3: Exemplo de mostrador digital

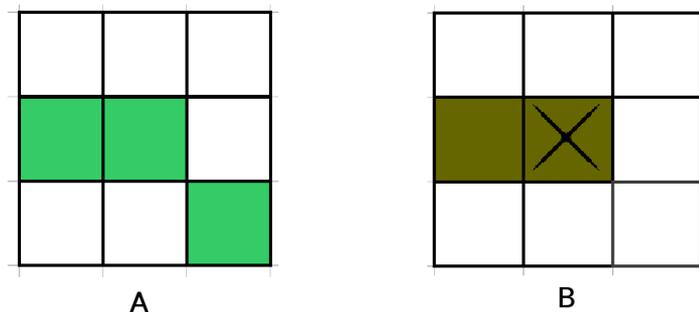


Figura 2.4: Representações de um detalhe de imagem a ser processada pelo *thinning*. Cada quadrado representa um pixel.

rentes de ruído na imagem. Os algoritmos que buscam encontrar essa linha são chamados de *Thinning*.

De acordo com Pitas[9], não existe uma definição matemática precisa para *thinning*. Como consequência, existem inúmeras versões desses algoritmos, mas todos baseiam-se na seguinte idéia: aplicam-se sucessivamente erosões na parte externa da imagem até que se encontre um conjunto de linhas de espessura unitária. Ou seja, os algoritmos de *thinning* vão, iterativamente, tirando pedaços da “casca” da imagem, até que tenha chegado ao seu centro.

Para o *thinning* funcionar corretamente, é importante manter a conectividade na imagem. Então, os pixels da borda são apagados de maneira que não se perca essa propriedade. Para isso, os algoritmos devem:

1. Não remover pixels que possam causar descontinuidade. A figura 2.4-A mostra uma situação em que nenhum dos 3 pixels pode ser removido sem separar uma linha existente na figura em 2 partes.
2. Não remover pixels que diminuam o comprimento de uma linha. A figura 2.4-B representa essa situação. Se removermos o pixel marcado, estaremos diminuindo o comprimento da linha a que ele pertence. Com isso, podemos deformar a imagem e alterar sua classificação.

Em seu livro, Pitas apresenta 2 tipos de algoritmo *Thinning*: Simple e em 2 etapas. Ambos são descritos sob a hipótese de que conhecemos o fundo da imagem e sabemos seu tom predominante. Algumas técnicas para determinar esse tom serão descritas adiante. Conhecendo a cor de fundo, os pixels cujas intensidades diferem mais que um determinado valor de sua intensidade serão os objetos de estudo, uma vez que são candidatos a pixels formadores da imagem desejada.

O *Thinning* simple percorre toda a imagem verificando a vizinhança de cada pixel, isto é, o quadrado 3×3 com o pixel a ser verificado ao centro. Se na região existem mais de 2 pixels candidatos, isto é, os pixels não se encaixam no caso B da figura 2.4, verificamos se a remoção do pixel central quebrará alguma das regras impostas para o algoritmo. Se isso não ocorre, percorremos a vizinhança no sentido horário, contando o número de transições dos tons de imagem para

o tom da cor de fundo. Se houver no máximo uma transição, então é seguro remover o pixel central sem afetar a conectividade das componentes da imagem. Uma desvantagem desse método é que a limpeza da imagem é feita de maneira não simétrica: ela depende da ordem em que percorremos os pixels.

Essa desvantagem é atacada no *Thinning* em 2 etapas. Nesse método, percorremos a imagem duas vezes, de maneiras diferentes, marcando os pixels para remoção. Apenas os pixels marcados mais de uma vez serão, de fato, removidos. Pitas sugere mais modificações lógicas para tornar o *Thinning* mais eficiente, descrevendo em detalhes sua implementação.

2.3 Pré-Processamento das Imagens

Dentre os algoritmos de Extração de Características estudados, uma característica em comum pode ser observada: todos tem resultados melhores se a imagem possuir forte contraste entre o objeto e o fundo. Para o *Zoning*, um contraste alto permite que as zonas onde a maior parte dos pixels pertence ao fundo da imagem se separem bem daqueles que contém partes do objeto. Uma imagem com intensidades muito homogêneas dificultam sua execução, pois o vetor de características resultante terá valores muito próximos entre si. Para as Projeções, o aumento do contraste aumenta a discrepância entre os valores do vetor de características, aumentando os picos do histograma e facilitando a diferenciação dos objetos. Finalmente, para o *Thinning*, o aumento do contraste ajuda a evitar a remoção de pixels errados.

2.3.1 Histogramas RGB

Chen[3] sugeriu um algoritmo relativamente simples para aumentar o contraste entre fundo e objeto. Em seu artigo, Chen desenvolveu um método para identificar textos em placas de trânsito. Para isso, ele usou a decomposição das imagens em matrizes RGB. Nessa decomposição, queremos encontrar 3 matrizes A_r , A_g e A_b de dimensão $m \times n$, cujos elementos, compostos entre si, formam a cor do pixel correspondente na imagem A . Cada uma das matrizes representa, respectivamente, a intensidade da cor Vermelha (R), Verde (G) e Azul (B), utilizados pelo computador para formar todas as cores que ele consegue exibir. A tabela abaixo mostra algumas cores possíveis, junto com os valores em cada matriz que a compõem.

Cor	$A_r(i, j)$	$A_g(i, j)$	$A_b(i, j)$
Branco	255	255	255
Preto	0	0	0
Azul	0	0	255
Amarelo	255	255	0

Com as matrizes A_r , A_g e A_b , o próximo passo é calcular os histogramas de frequência. Neles, indicamos a quantidade de pixels de cada intensidade presentes na matriz. Um valor alto indica uma grande quantidade de pixels com a intensidade. Como, em geral, a cor de fundo é predominante nas imagens, o algoritmo de Chen elimina os pixels que possuem as intensidades presentes no pico de cada histograma. Dessa maneira, ao compor as três matrizes já

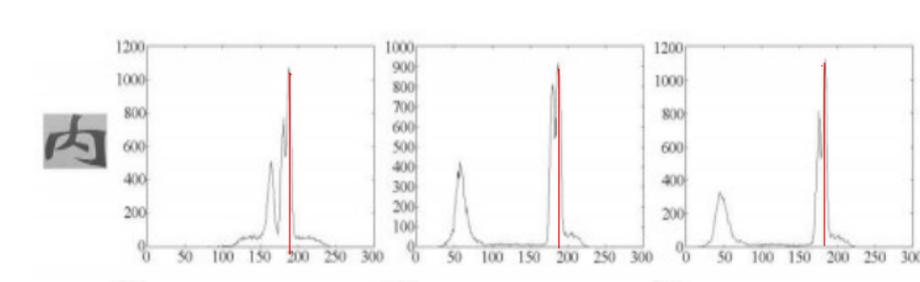


Figura 2.5: Exemplo de histogramas RGB

processadas, teremos eliminado a cor de fundo da imagem, aumentando seu contraste com a imagem restante.

Por exemplo, observe a figura 2.5. Os histogramas apresentados foram calculados sobre, respectivamente, A_r , A_g e A_b . A linha destacada apresenta o pico mais alto de cada histograma. Esses valores, provavelmente, são os valores que formam a cor de fundo da imagem original (apresentada ao lado dos histogramas). Sendo assim, se removermos os pixels cuja decomposição se aproxima dos valores máximos em cada matriz, estaremos removendo o fundo da figura, dando maior destaque para o objeto de estudo.

Capítulo 3

Atividades Realizadas e Resultados

Durante o decorrer do trabalho, o problema foi se mostrando um desafio maior que o esperado. À priori, foram testados alguns algoritmos de detecção e casamento de características. Esperávamos que, dessa maneira, pudessemos encontrar os algoritmos facilmente. O algoritmo do SIFT[7] (*Scale-Invariant Feature Transform*) foi escolhido para os teste iniciais. Esse algoritmo, porém, se mostrou complexo demais para o problema. Por sua natureza, o SIFT exigia uma correspondência muito grande entre os algoritmos para determiná-los como sendo da mesma classe. Sendo muito preciso, o SIFT acabava não identificando corretamente os números. Alterar o algoritmo para ser mais permissivo o tornaria inútil, já que a classificação seria prejudicada, uma vez que ele poderia identificar erroneamente dois algoritmos diferentes como sendo da mesma classe.

Decidimos, em seguida, partir para o estudo dos OCR. Definimos excluir desse estudo a Segmentação e a Classificação, focando os trabalhos na Extração de Características, ponto crucial para os mecanismos de OCR. Infelizmente, a grande gama de algoritmos dessa natureza acabou dificultando a escolha dos algoritmos mais úteis ao nosso trabalho. Não pudemos testar todos os algoritmos disponíveis e acabamos definindo alguns para estudo. A partir daí, foram realizados inúmeros testes com um bando de números de cartão de crédito e alguns dos algoritmos de Extração de Características.

O *Zoning* não apresentou resultados favoráveis. Sua simplicidade foi prejudicial, já que para entradas mais complicadas seu resultado não era confiável. Além disso, em imagens de cores muito homogêneas, o *Zoning* teve como resultado um vetor de características em que a diferença do valor de cada elemento do vetor para os outros não era muito grande, tornando-se inútil para fins de classificação.

O *Thinning* foi estudado mas não foi implementado e, portanto, não obtivemos resultados sobre sua eficiência ou não na solução do problema. A implementação apresentada por Pitas[9] é dependente de uma binarização da imagem e, durante o período desse trabalho, não conseguimos binarizar as imagens satisfatoriamente.

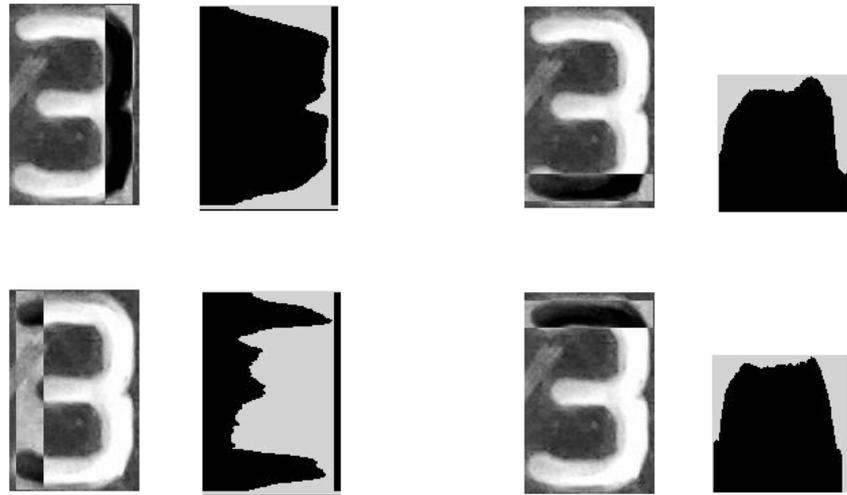


Figura 3.1: Testes com projeções locais no algoritmo 3

3.1 Testes com Projeções

As projeções foram o foco principal desse trabalho. A idéia de calcular as projeções em alguns retângulos pré-determinados, como se colocássemos máscaras sobre os números (como no visor de um relógio digital, por exemplo) surgiu durante uma das reuniões. Fizemos muitos testes e seus resultados são os que seguem.

Para imagens bem comportadas, com nítida separação entre o fundo e o algarismo, a técnica das projeções funcionou bem. A figura 3.1 mostra os resultados obtidos com o algoritmo 3. Na figura, são mostrados os histogramas e a imagem do número 3 em tons de cinza, com o retângulo alvo da projeção em destaque. O retângulo da esquerda mostra como a técnica funciona bem para algarismos nítidos: Há 3 picos e 2 vales, exatamente o esperado para uma região que deve ter 3 regiões pertencentes ao algarismo e 2 não pertencentes. Os outros histogramas mostram-se quase constantes, pois estão inteiramente preenchidos com a presença do algarismo 3.

Já em imagens menos nítidas, a técnica pareceu funcionar também, porém com bem menos eficácia. A figura 3.2 mostra a projeção da esquerda em outro algarismo 3, bem menos nítido que o anterior. Ainda podemos perceber 3 regiões mais altas e 2 mais baixas, mas o excesso de ruído na imagem criou alguns outros picos menores que podem atrapalhar a classificação.

3.1.1 Projeções com Histogramas RGB

O algoritmo de Chen pareceu uma boa saída para aumentar o contraste dos algarismos com as imagens de fundo, além de limpar o ruído presente nas imagens. Infelizmente, não houve tempo para testar a abordagem de seu algoritmo

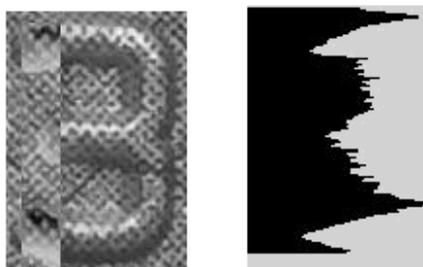


Figura 3.2: Uma entrada mais problemática

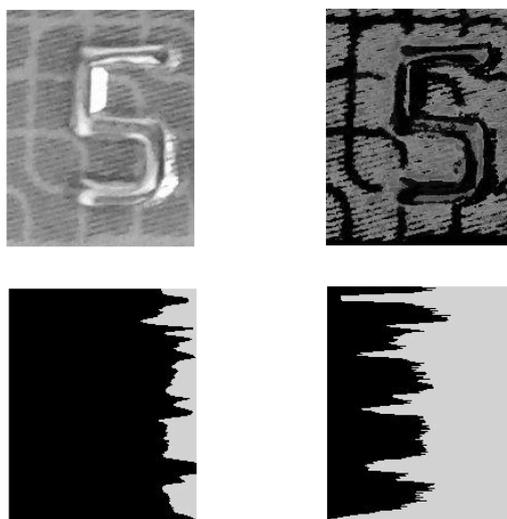


Figura 3.3: Projeções antes e depois da aplicação do algoritmo de Chen

para mais entradas. Um dos testes realizados pode ser visto na figura 3.3. Nesse teste, foi calculada a projeção de um retângulo que passa ao longo do eixo Y , no meio da imagem, passando sobre o algarismo 5. À esquerda estão a imagem original e sua Y -projeção mediana. Podemos notar que, apesar de o algarismo não estar posicionado sobre o retângulo alvo, sua projeção calculada gerou um histograma quase constante. Para fins de classificação, essa característica, se passada para o classificador, poderia gerar como resposta um número 1. Aplicado o algoritmo de Chen e recalculada a Y -projeção mediana, podemos ver que, agora, a projeção indica que há pontos de descontinuidade sobre o retângulo mediano. Os vales da projeção representam as regiões do retângulo que intersectam a superfície do algarismo 5. Dessa maneira, passamos para o classificador uma característica mais fiel, e minimizamos sua possibilidade de erro.

Infelizmente, não houve tempo para realizar mais testes que confirmassem (ou não) a utilidade do algoritmo de Chen na solução do problema.

Capítulo 4

Conclusão

Apesar de não termos conseguido desenvolver um software que, de fato, reconheça números em imagens de cartão de crédito, acredito que os avanços realizados na área indicam um bom caminho a ser seguido. A junção das projeções sobre X e Y com o algoritmo de Chen pareceu promissora. Além disso, poderiam ser testados mais algoritmos que trabalham com o aumento do contraste dentro de imagens como, por exemplo, o *Ultimate Opening*[1]. Um próximo passo para resolver esse problema poderia ser testar o *Ultimate Opening* nas entradas do problema e verificar se as características extraídas após sua aplicação são mais ou menos relevantes.

Acredito que, apesar do objetivo final não ter sido alcançado, os erros e desvios que o problema causou foram bem proveitosos no intuito de refutar teorias não aplicáveis aos problemas e nos fazer buscar abordagens ainda não tentadas para a solução de problemas da área de Visão.

Pessoalmente, foi um grande aprendizado. Além de entender como funciona, de fato, a produção científica acadêmica, ainda tive contato com um problema bem palpável e real e com uma área de pesquisa (Visão Computacional) com que tive pouco contato durante a graduação.

Parte II

Parte Subjetiva

Capítulo 5

O Curso de Ciência da Computação

5.1 Desafios e Frustrações

Quando fiz minha transferência para o curso de Ciência da Computação, vindo da Licenciatura em Matemática do IME USP, tive receio de que, por minha pouca experiência em programação (que, na época, resumia-se ao curso de Introdução à Computação) fosse um obstáculo à minha formação. Percebi, porém, que isso não era verdade. O viés teórico da Ciência da Computação do IME, às vezes tão hostilizado pelos seus alunos, é, justamente, o que me atraiu no curso. Por gostar muito de matemática, meus cursos favoritos na graduação foram os puramente teóricos, mesmo que fora do escopo da computação. Cursei Teoria dos Jogos (na Estatística) e Análise Numérica (na Matemática Aplicada), por exemplo, e estes estão entre os cursos que mais gostei de ter feito.

Acredito que um dos grandes desafios do BCC seja fazer a transição do ambiente acadêmico para o mercado de trabalho. Acho que o nível de conhecimento exigido pelo mercado de trabalho seja muito baixo comparado ao que aprendemos no BCC, e isso pode acabar desestimulando seus alunos, que vêm a oportunidade de começar a trabalhar muito cedo e, com isso, acabam se desligando do curso. Não vejo como isso possa mudar, mas vejo com tristeza alguns bons alunos do BCC não investirem na carreira acadêmica seduzidos pelo Mercado de Trabalho.

Algumas frustrações no BCC foi justamente o Trabalho de Formatura. Por ser feito juntamente com outras disciplinas, infelizmente não consegui dedicar tanto tempo quanto desejaria ter feito a esse trabalho. Vejo o trabalho de formatura como uma oportunidade única, na graduação, de enfrentar um problema sem ter um enunciado pronto, como fazemos durante todo o curso com os Exercícios de Programação.

5.2 Disciplinas relevantes ao Trabalho

Todas as disciplinas que cursei tiveram sua importância para este trabalho, mas destaco:

- **MAC 110 - Introdução à Computação**
MAC 122 - Princípios de Desenvolvimento de Algoritmos
MAC 323 - Estruturas de Dados
As 3 disciplinas, juntas, nos dão a base de sustentação necessária para aprender a escrever programas de computador. Além disso, despertam nossa atenção para a eficiência dos algoritmos, tema recorrente em todas as disciplinas do BCC.
- **MAC 417 - Visão Computacional**
Sua importância para este trabalho é óbvia. Cursando a disciplina desenvolvi interesse pela área e busquei um trabalho na área.
- **MAC 328 - Algoritmos em Grafos**
MAC 338 - Análise de Algoritmos
Ainda que indiretamente, as disciplinas me ensinaram técnicas que foram úteis na implementação de alguns testes, como Busca em Largura e Profundidade e Programação Dinâmica
- **MAP 131 - Laboratório de Matemática Aplicada**
MAP 2210 - Aplicações de Álgebra Linear
MAP 2220 - Fundamentos de Análise Numérica
As disciplinas de Matemática Aplicada que cursei tiveram importância fundamental e inesperada nesse trabalho: diversas vezes, utilizei-me de entidades matemáticas aprendidas nesses cursos. Por exemplo, os *Splines*, Interpolação Polinomial e Cálculo de Erro. Creio, ainda, que os alunos do BCC deveriam olhar com mais cuidado para essas disciplinas, cujas ementas são muito úteis no meio acadêmico e fora dele.

5.3 Daqui pra Frente...

Não creio que continuarei estudando na Área de Visão Computacional. Apesar de ter gostado muito da área, percebi que meus interesses divergem um pouco dos assuntos estudados nessa área. Deixo isso para os colegas mais interessados em Visão.

Quanto a mim, creio que continuarei estudando. Não sei se Computação, Matemática ou Estatística, mas continuarei estudando.

Capítulo 6

Agradecimentos

Gostaria de deixar registrado alguns agradecimentos a pessoas que, direta ou indiretamente, me ajudaram durante minha estadia no IME-USP.

- A Meus Pais por seu apoio incondicional em todos esses anos
- A meu supervisor, professor Ronaldo Fumio Hashimoto, e ao aluno de Doutorado Wonder Alexandre Luz Alvez, pela ajuda no desenvolvimento do projeto e pela paciência
- Aos colegas do IME, que são muitos e cujos nomes seria injusto citar, sob risco de esquecer alguém. Obrigado pelos momentos inesquecíveis que levarei para sempre comigo
- Aos professores Manuel Valentin Pera Garcia, Sônia Regina Leite Garcia, Sérgio Muniz Oliva Filho, Fábio Armando Tal, Henrique Von Dreifus, Eduardo Colli, Roseli Fernandez, Alegria Gladys Chalom, Cláudia Monteiro Peixoto, Nina Sumiko Tomita Hirata, Marco Dimas Gubitoso e Paulo José Silva e Silva por me ensinarem a gostar de **aprender**, seja Matemática, Estatística, Computação ou simplesmente aprender.

Referências Bibliográficas

- [1] BEUCHER, S. Numerical residues. *Image Vision Comput.* 25 (April 2007), 405–415.
- [2] BOKSER, M. Omnidocument technologies. *Proceedings of the IEEE* 80 (1992), 1066–1078.
- [3] CHEN, X., YANG, J., ZHANG, J., E WAIBEL, A. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on Image Processing* 13, 1 (2004).
- [4] DAVIS, L. S. A survey of edge detection techniques. *Computer Graphics and Image Processing* 4, 3 (1975), 248–270.
- [5] DEVIJER, P. A., E KITTLER, J. *Patter Recognition - A Statistical Approach*. Prentice-Hall, 1982.
- [6] GONZALES, R. C., E WOODS, R. E. *Digital Image Processing*, 3 ed. Pearson Prentice Hall, 2008.
- [7] LOWE, D. G. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2* (Washington, DC, USA, 1999), ICCV '99, IEEE Computer Society, pp. 1150–.
- [8] PEDRINI, H., E SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. Thomsom Learning, 2008.
- [9] PITAS, Y. *Digital Image Processing Algorithms and Applications*. John Wiley and Sons, 2000.
- [10] TRIER, Ø. D., JAIN, A. K., E TAXT, T. Feature extraction methods for character recognition - a survey. *Patter Recognition* 29, 4 (1996), 641–662.