

ESTUDO DE TÉCNICAS PARA IDENTIFICAÇÃO DE MODELOS DE VEÍCULOS

Trabalho de Conclusão de Curso

Engenharia da Computação

Fernando José Lourenço da Rocha
Orientador: Prof. Wellington Pinheiro dos Santos



UNIVERSIDADE
DE PERNAMBUCO

Fernando José Lourenço da Rocha

**ESTUDO DE TÉCNICAS PARA
IDENTIFICAÇÃO DE MODELOS DE
VEÍCULOS**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, junho de 2010.

*Dedico este trabalho a minha amada esposa, que sempre me ajudou em tudo
o que estava ao seu alcance.*

Agradecimentos

Agradeço a todos os professores que ao longo do curso se esforçaram em cultivar continuamente a semente do conhecimento, em especial ao professor Wellington Santos, que compartilhou sua experiência e me incentivou a desenvolver este trabalho. Agradeço também à minha família, que sempre apoiou minhas escolhas e me encorajou a concluir o curso.

Resumo

Diversas aplicações utilizam técnicas de reconhecimento de padrões para resoluções de problemas de forma automatizada. O presente trabalho faz uso destas técnicas para mostrar que é possível realizar a identificação de modelos de veículos por um sistema computacional. Para atingir este objetivo foram estudadas formas de representação e métodos de descrição, que são responsáveis por extrair características relevantes das imagens. Além disso, é necessário o uso de um classificador que, através de comparações dos dados obtidos com dados previamente armazenados, reconhece a que classe (modelo) pertence cada objeto (veículo). Os momentos invariantes de Hu são utilizados como descritor de forma e a classificação é feita calculando a distância mínima entre os vetores de atributos. Os resultados obtidos indicam que a metodologia utilizada pode ser aplicada na resolução do problema em aplicações reais, mas que é necessário a análise de outras técnicas de classificação para se diminuir a taxa de erro.

Palavras-chave: Reconhecimento de padrões. Descritores de forma. Momentos invariantes.

Abstract

Several applications using pattern recognition techniques for solving problems automatically. This research makes use of these techniques to show that you can make the identification of vehicle by a computer system. To achieve this objective we have studied the forms of representation and description methods that are responsible for extracting relevant features of images. Furthermore, it is necessary to use a classifier that, by comparing the data obtained from previously stored data, which recognizes the class (model) belongs to each object (vehicle). Hu invariant moments are used as a descriptor of shape and the classification is done by calculating the minimum distance between the vectors of attributes. The results indicate that the methodology can be applied in solving the problem in real applications, but it is necessary to analyze other classification techniques to reduce the error rate.

Keywords: Pattern recognition. Shape descriptors. Invariant moments.

Sumário

Resumo	i
Abstract	ii
Sumário	iii
Índice de Figuras	v
Índice de Tabelas	vii
Capítulo 1 Introdução	8
Capítulo 2 Métodos de Descrição de Forma em Imagens Digitais	10
2.1 Esquemas de Representação	11
2.1.1 Código de Cadeia	11
2.1.2 Aproximações Poligonais	12
2.1.3 Assinaturas	12
2.1.4 Esqueleto de um Objeto	13
2.2 Descritores de Borda	13
2.2.1 Descritores de Fourier	14
2.3 Descritores de Região	15
2.3.1 Projeções Horizontal e Vertical	15
2.3.2 Propriedades Topológicas	16
2.3.3 Momentos	17
Capítulo 3 Metodologia e Experimentos Realizados	20
3.1 Aquisição da Base de Imagens	20
3.2 Segmentação e Pré-processamento	22

iii

3.2.1	Detectores de Bordas	22
3.2.2	Diferença entre imagens com fundo fixo	24
3.2.3	Método Utilizado	27
3.3	Extração de Características	28
3.4	Classificação	28
Capítulo 4 Resultados Obtidos		29
4.1	Representação pelo contorno	29
4.2	Representação pelo contorno e lanternas	31
Capítulo 5 Conclusão e Trabalhos Futuros		35
Bibliografia		36
Anexo A Código Scilab para extração de momentos invariantes de Hu		37

Índice de Figuras

Figura 1.	Direções do código de cadeia para conectividade-de-4 (a) e para conectividade-de-8 (b) [1].....	11
Figura 2.	Representação 8-direcional do código de cadeia [6].....	12
Figura 3.	Exemplo de assinatura da borda de uma circunferência.....	13
Figura 4.	Diâmetro de um borda.....	13
Figura 5.	Projeções horizontal e vertical.....	16
Figura 6.	Exemplo de variação de escala na amostra utilizada.....	21
Figura 7.	Exemplo de variação de escala na amostra utilizada.....	21
Figura 8.	Detecção de borda através da função <i>Edge</i> , usando o método <i>Canny</i>	23
Figura 9.	Operações de preenchimento do fundo e erosão.....	23
Figura 10.	Destaque da falha na detecção da borda.....	24
Figura 11.	Imagens com fundo fixo.....	25
Figura 12.	Diferença absoluta (esquerda) e Máscara binária (direita).....	26
Figura 13.	Imagem original multiplicada pela máscara.....	26
Figura 14.	Formas de representação do objeto.....	27
Figura 15.	Representação do contorno da traseira de 4 classes/modelos.....	29

Figura 16.	Taxa de erro na classificação	30
Figura 17.	Segunda forma de representação utilizada	31
Figura 18.	Taxa de erro na classificação	32
Figura 19.	Imagens da classe 6.....	33

Índice de Tabelas

Tabela 1.	Vetores de atributos das imagens de referência.	30
Tabela 2.	Vetores de atributos das imagens de referência.	31
Tabela 3.	Vetores de atributos das imagens usadas na classificação da classe 6.	33
Tabela 4.	Destaque da semelhança entre vetores de classes distintas.	34

Capítulo 1

Introdução

O reconhecimento de padrões é uma área de pesquisa que tem por objetivo a classificação de objetos em um número de categorias ou classes [9].

Um computador consegue reconhecer padrões através da comparação de atributos extraídos com outros já armazenados na memória. Um sistema completo é tipicamente formado por um sensor que obtém a informação primária; um mecanismo de extração de características; e um ferramenta para classificação dos padrões.

Através da análise e processamento de imagens digitais, o reconhecimento de padrões pode ser utilizado nas mais diversas aplicações. Na medicina, sistemas de reconhecimento podem classificar tomografias ou radiografias de acordo com características extraídas das imagens. Outras aplicações típicas são reconhecimento de caracteres, identificação biométrica e análises de imagens geológicas.

No presente trabalho, o problema em análise é a identificação do modelo de veículos por um sistema computacional.

Geralmente os sistemas utilizados no reconhecimento de veículos estão focados na identificação de suas placas. Mas esses softwares não garantem que a placa reconhecida é realmente daquele veículo, podendo assim, um veículo roubado ter sua placa trocada a fim de enganar os órgãos de fiscalização, dificultando sua recuperação.

Portanto, a motivação principal do trabalho é possibilitar que as autoridades de segurança pública possam identificar veículos roubados e clonados, tornando mais fácil rastrear e apreender tais veículos.

Para resolver este problema, em conjunto com um sistema que identifique a placa, pode ser utilizada uma ferramenta que faça o reconhecimento do veículo através da análise da sua forma, sendo capaz de identificar a marca e o modelo. Os

dados adquiridos com a placa podem ser comparados com os obtidos com a ferramenta de análise de forma, tornando identificação do veículo mais precisa.

Definido o problema, é importante destacar que este trabalho está focado no estudo de técnicas capazes de reconhecer o modelo do veículo. A extração da placa seria função de outro sistema. As duas soluções integradas poderiam ser usadas para analisar em tempo real, por exemplo, as imagens captadas pelo sistema de monitoramento de tráfego de grandes cidades. O que possibilitaria, também em tempo real, o rastreamento de veículos ilegais.

No capítulo 2 são estudadas algumas técnicas de representação de imagens digitais, além de alguns métodos de descrição de forma, que são responsáveis por extrair as características relevantes das imagens.

O capítulo 3 descreve a metodologia utilizada e os experimentos realizados no trabalho. É explicado como foram adquiridas as imagens, quais as técnicas de segmentação estudadas, como as imagens foram representadas e caracterizadas e qual o método de classificação utilizado.

No capítulo 4 são apresentados e analisados os resultados obtidos. Por fim, o capítulo 5 apresenta as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Métodos de Descrição de Forma em Imagens Digitais

Após a etapa de segmentação, onde o objeto ou região a ser analisada é isolada do restante da imagem, é preciso extrair um conjunto de características para proceder com a classificação deste objeto. Para proceder com a classificação, o conjunto de *pixels* resultantes da segmentação deve ser representado e descrito de forma apropriada.

Uma região pode ser representada de acordo com suas características externas (fronteiras) ou em termos de suas características internas (*pixels* que a compõe). A escolha de um esquema de representação é apenas parte da tarefa de tornar os dados adequados para o computador. A próxima etapa é descrever a região baseado na representação escolhida [1].

A descrição é o processo de extração de um número reduzido de medidas e atributos de uma imagem e deve possibilitar a caracterização da forma do objeto em estudo sem ambigüidade.

Na maioria das aplicações práticas é importante que um descritor de forma seja pouco afetado por transformações geométricas, como mudança de escala, rotação e translação.

Nas seções seguintes serão apresentados alguns esquemas de representação e descritores de formas.

2.1 Esquemas de Representação

Estruturas adequadas de representação facilitam o armazenamento e a manipulação dos objetos segmentados da imagem, além de simplificarem o cálculo de certos descritores de forma [6].

2.1.1 Código de Cadeia

É um esquema proposto para representar a borda de um objeto de forma mais simplificada. A curva da borda é aproximada por uma seqüência de segmentos de linha retos e conectados, dos quais são conhecidos o comprimento e a direção.

Cada segmento é codificado de acordo com um número que corresponde à sua direção, e esta seqüência forma o chamado código da cadeia [7]. Esta representação pode ser baseada na conectividade-de-4 ou na conectividade-de-8, conforme mostra a Figura 1.

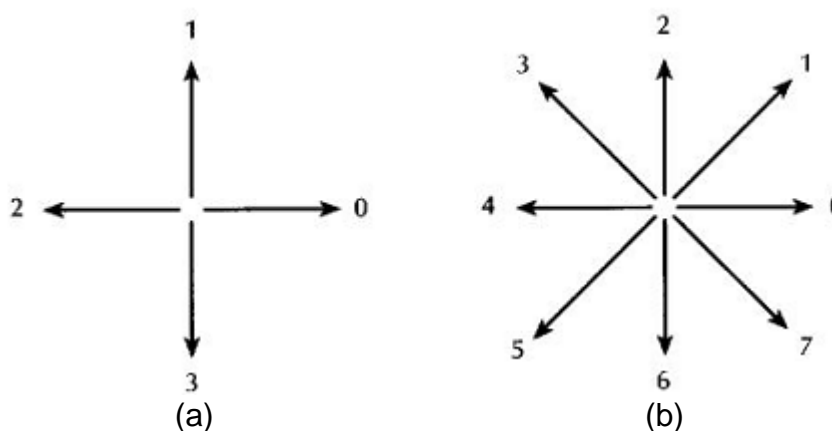


Figura 1. Direções do código de cadeia para conectividade-de-4 (a) e para conectividade-de-8 (b) [1]

Um exemplo de representação usando a conectividade-de-8 pode ser visto na Figura 2, quando o código de cadeia gerado é 1170012344443556670, considerando como ponto de partida a coordenada (3,6) e seguindo o sentido anti-horário.

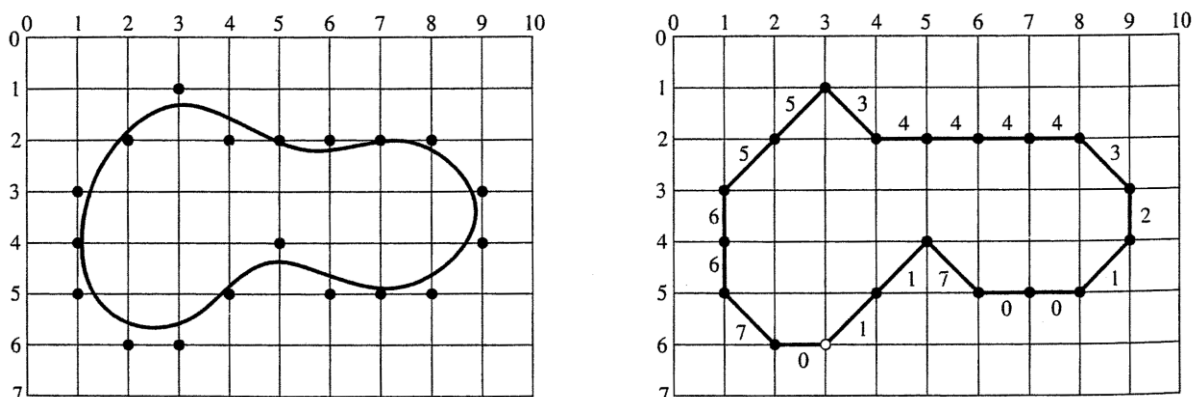


Figura 2. Representação 8-direcional do código de cadeia [6]

2.1.2 Aproximações Poligonais

Consiste em aproximar a borda de um objeto a um polígono, conforme uma precisão definida arbitrariamente. No caso de uma curva fechada, a aproximação será exata quando o número de segmentos do polígono for igual ao número de pontos da borda. Ou seja, cada par de pontos adjacentes vai definir um segmento do polígono.

O principal objetivo da aproximação poligonal é extrair a essência da forma da borda com o menor número de segmentos. Tal problema não é simples e pode ter um alto custo computacional, mas sua solução pode ser simplificada através da busca de polígonos de perímetro mínimo, que se ajustam à geometria estabelecida pela grade da imagem.

2.1.3 Assinaturas

Uma assinatura é uma representação unidimensional de uma borda e pode ser gerada de várias formas. Uma das mais simples é através do gráfico da distância da borda ao centróide em função do ângulo, como mostra o exemplo da Figura 3.

O objetivo principal do uso de assinaturas é reduzir a representação da borda a uma função unidimensional, que é mais simples de ser descrita que a borda bidimensional original.

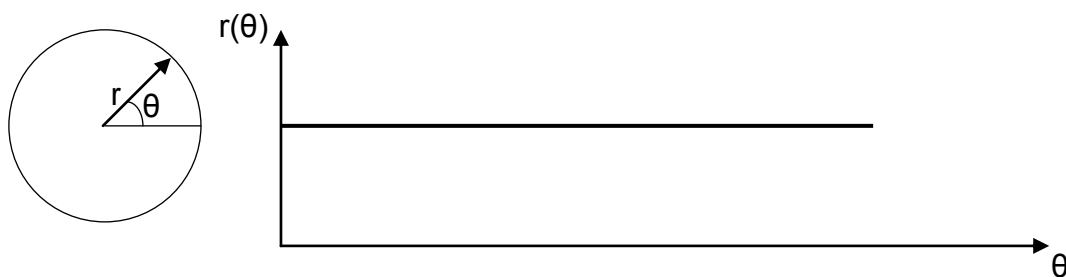


Figura 3. Exemplo de assinatura da borda de uma circunferência

2.1.4 Esqueleto de um Objeto

O esqueleto é uma representação reduzida de um objeto e pode ser obtido por meio de técnicas com transformada do eixo médio, transformada de distância, diagrama de Voroni e afinamento. Consiste basicamente em reduzir uma região planar em um grafo.

2.2 Descritores de Borda

Alguns descritores são bastante simples, porém sofrem influência de operações de rotação, translação ou mudança de escala.

O Perímetro é um exemplo de um descritor de borda. Um valor aproximado pode ser obtido com a contagem dos *pixels* ao longo da borda do objeto.

Outro descritor de borda útil é o Diâmetro, que é definido como a maior distância entre dois pontos pertencentes à borda, conforme ilustrado na Figura 4. O Diâmetro de uma borda B, pode ser expresso por:

$$\text{Diâmetro}(B) = \max [D(p_i p_j)] \quad (1)$$

Onde D é a distância entre os pontos p_i e p_j .

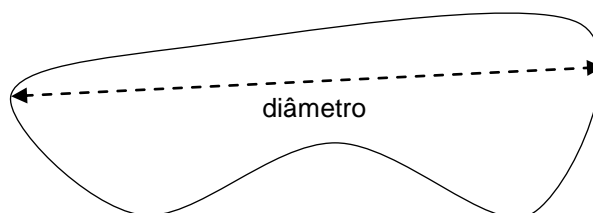


Figura 4. Diâmetro de um borda

Tanto o Perímetro como o Diâmetro são invariantes quanto à rotação e translação, mas não são invariantes quanto à mudança de escala.

Também é usada como descritor a Curvatura de uma borda, que mede a taxa de variação da inclinação ao longo do contorno do objeto. Pontos situados ao longo de um segmento de reta, por exemplo, apresentam curvatura igual a zero, enquanto que uma circunferência tem curvatura constante.

2.2.1 Descritores de Fourier

A Transformada de Fourier baseia-se no teorema de Fourier, pelo qual toda função contínua pode ser decomposta em uma soma de funções senoidais de frequências variadas. Para o caso discreto, que é aquele que interessa à aplicação em imagens digitais, usa-se a Transformada Discreta de Fourier (DFT) [7].

Seja uma fronteira digital de N pontos no plano xy . Considerando (x_0, y_0) , $(x_1, y_1), \dots, (x_{N-1}, y_{N-1})$, os pontos da borda obtidos através do percurso anti-horário. Para $x(k)=x_k$ e $y(k)=y_k$, temos então a borda expressa como:

$$s(k)=[x(k),y(k)] \quad (2)$$

Pode-se tratar cada elemento de 's' como um número complexo:

$$s(k)=x(k)+j y(k), \text{ para } k=0, 1, 2, \dots, N-1 \quad (3)$$

A transformada de Fourier de $s(k)$ então pode ser definida como:

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp\left(\frac{-j2 \Pi uk}{N}\right) \quad (4)$$

Suponha que apenas os M primeiros coeficientes sejam utilizados. Isto é equivalente a zerar todos os valores de $a(u)$ para $u > M-1$. Então temos:

$$s^*(k) = \sum_{u=0}^{M-1} a(u) \exp\left(\frac{j2 \Pi uk}{N}\right) \quad (5)$$

As baixas frequências guardam informações de forma, enquanto altas frequências representam os detalhes finos. Portanto, quanto maior o valor de M utilizado, mais detalhada será a descrição da imagem. Para descrever eficientemente caracteres, por exemplo, o uso dos 15 primeiros coeficientes já são suficientes [6].

2.3 Descritores de Região

A Área de uma região é definida pela quantidade de *pixels* contidos dentro de seu contorno. Da mesma forma que o Perímetro, já descrito na seção 2.2, a Área é comumente usada como descritor em casos onde não há variação no tamanho dos objetos, pois ambos são invariantes apenas quanto à rotação e translação.

Para obter a invariância também quanto à escala, pode-se usar a Área e o Perímetro em conjunto, através da definição da Compacidade ou Circularidade de uma região, que é uma medida sem dimensão.

A Compacidade de uma região em forma de círculo, por exemplo, é mínima. Para uma região qualquer, a Compacidade é definida como o quadrado do perímetro sobre área:

$$C = p^2 / A \quad (6)$$

2.3.1 Projeções Horizontal e Vertical

A Figura 5 ilustra um exemplo de projeções horizontal e vertical. A projeção horizontal de uma imagem binária é obtida com o somatório dos *pixels* com valor igual a 1 em cada linha da região.

De forma análoga se obtém a projeção vertical, desta vez com a soma dos *pixels* existentes em cada coluna.

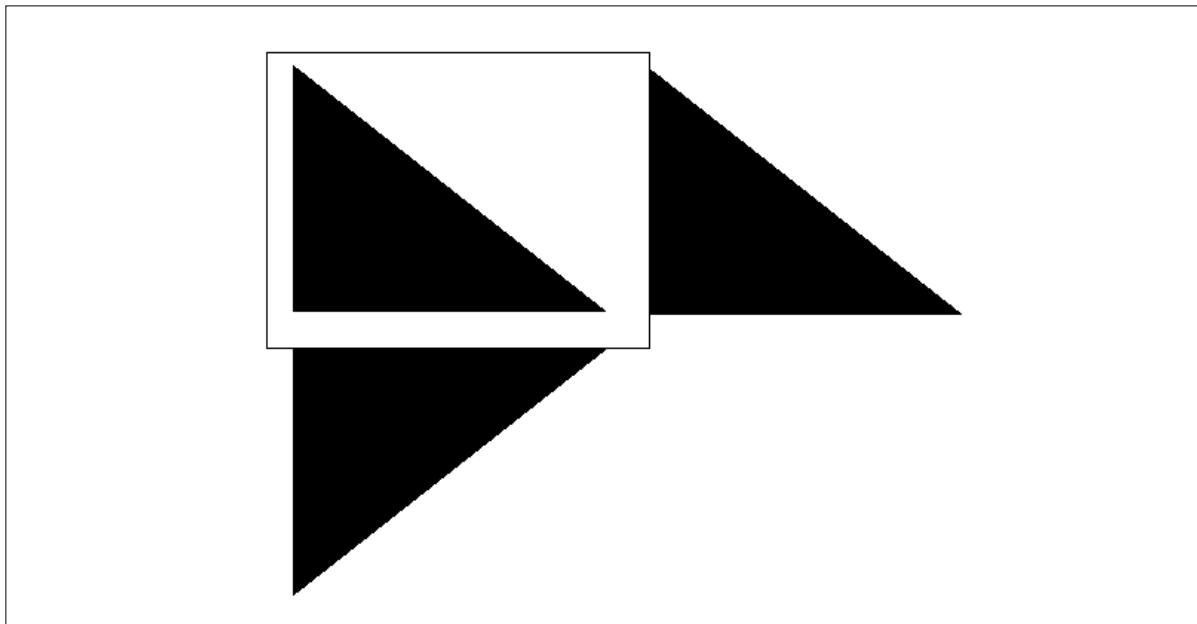


Figura 5. Projeções horizontal e vertical

2.3.2 Propriedades Topológicas

As propriedades topológicas são úteis para descrições globais de regiões no plano das imagens. A topologia é o estudo das propriedades de um objeto que não são afetadas por transformações geométricas [1].

Duas possíveis descrições topológicas podem ser obtidas pela contagem do número de buracos na região (H) e do número de componentes conexos (C). O número de Euler (E), que também é uma propriedade topológica, é definido através dessas medidas, conforme expresso a seguir:

$$E = C - H \quad (7)$$

2.3.3 Momentos

Momentos descrevem quantidades numéricas em alguma distância de um ponto de referência ou eixo e podem ser usados para caracterizar uma imagem binária ou em tons de cinza, considerando esta como uma função bidirecional de densidade [4].

A representação de toda informação presente em uma imagem utilizaria um número infinito de valores de momento. Assim, para a implementação prática de sistemas de reconhecimento que utilizem momentos, torna-se necessário selecionar um conjunto representativo de valores de momentos que possam trazer informações suficientes para caracterizar a imagem sem ambigüidade.

A definição de momentos geométricos regulares tem a forma de uma projeção de função $f(x,y)$, que representa a imagem em uma função polinomial do tipo $x^p y^q$. O momento $(p+q)$ é definido de acordo com a equação a seguir. Sendo $p,q = 0,1,\dots,\infty$.

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad (8)$$

Onde $m_{0,0}$ é a área da região e $m_{0,1}$ e $m_{1,0}$ são as coordenadas do centro de massa da região.

O chamados momentos centrais são momentos centralizados em regiões, e podem ser expressos como a seguir.

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x,y) dx dy \quad (9)$$

Onde $\bar{x} = \frac{m_{10}}{m_{00}}$ e $\bar{y} = \frac{m_{01}}{m_{00}}$, que são as coordenadas do centro de massa

normalizadas pela área.

Para um imagem digital temos:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (10)$$

A partir desta equação, para valores de p,q entre 0 e 3, são obtidos os momentos de centrais de 1ª a 3ª ordem, que são invariantes quanto a translação.

1ª ordem:

$$\mu_{10} = \mu_{01} = 0 \quad (11)$$

2ª ordem:

$$\begin{aligned} \mu_{20} &= m_{20} - \bar{x}m_{10} \\ \mu_{02} &= m_{02} - \bar{y}m_{01} \\ \mu_{11} &= m_{11} - \bar{y}m_{10} \end{aligned} \quad (12)$$

3ª ordem:

$$\begin{aligned} \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} - 2\bar{y}^2 m_{10} \\ \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} - 2\bar{x}^2 m_{01} \\ \mu_{30} &= m_{30} - 3\bar{x}m_{20} - 2\bar{x}^2 m_{10} \\ \mu_{03} &= m_{03} - 3\bar{y}m_{02} - 2\bar{y}^2 m_{01} \end{aligned} \quad (13)$$

Os momentos centrais normalizados pela área são também invariantes quanto à escala. Eles são definidos por $\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^\gamma}$, onde $\gamma = \frac{p+q+2}{2}$, para $(p+q) \geq 2$.

Com a combinação dos momentos centrais de segunda e terceira ordem, normalizados pela área, [Hu] calculou um conjunto de sete momentos que, além de invariantes quanto à translação e escala, são também invariantes quanto à rotação. Estes sete momentos são listados a seguir:

$$\begin{aligned}
 \phi_1 &= \eta_{20} + \eta_{02} \\
 \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
 \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
 \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
 \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) \\
 & \quad (\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
 \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
 \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30}) \\
 & \quad (\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
 \end{aligned} \tag{14}$$

Outros tipos de momentos são também utilizados, nas mais diversas aplicações, para extração de características invariantes em imagens digitais. Entre os mais utilizados estão momentos ortogonais, momentos de Legendre, momentos de Zernike, momentos rotacionais e momentos complexos. Cada um tem características próprias com relação à capacidade de descrição de imagens, sensibilidade ao ruído e redundância de informações que carrega [4].

Capítulo 3

Metodologia e Experimentos

Realizados

O objetivo principal deste trabalho é o estudo de descritores de forma e a realização de experimentos para analisar a eficiência, em reconhecimento de padrões, de métodos de representação e descrição de veículos. Foi escolhido um classificador simples, pois o foco não será a análise de desempenho do classificador.

A seguir serão expostos os procedimentos realizados em cada etapa do processo de reconhecimento de padrões. As etapas são: aquisição da base de imagens, segmentação e pré-processamento, extração de características e classificação.

3.1 Aquisição da Base de Imagens

Nas bases de dados públicas, utilizadas em pesquisas na área de processamento de imagem, não foi possível encontrar um número suficiente de imagens de veículos para realizar os experimentos deste trabalho.

Portanto, foi necessário criar a base de dados utilizada no trabalho, que é formada por imagens da traseira de veículos. Este processo foi manual e foram tiradas várias fotos coloridas da traseira de veículos em estacionamentos públicos com uma máquina fotográfica digital com sensor CCD-*charge-coupled device*. As imagens foram obtidas no formato JPEG-*Joint Photographic Experts Group*, na resolução de 1600x1200 *pixels* (2.0 *megapixels*).

As fotos não foram tiradas de uma distância fixa pré-estabelecida, portanto, há variações de escala dos objetos de análise. Isso pode ser observado na Figura 6, onde a área ocupada pelo veículo da esquerda é de aproximadamente 212.000

pixels, enquanto o da direita é de 185.000 *pixels*, apesar das duas imagens possuírem a mesma dimensão (600x600 *pixels*).



Figura 6. Exemplo de variação de escala na amostra utilizada

Também houve variação quanto à rotação das imagens, pois, além da aleatoriedade gerada pelo senso de orientação do fotógrafo, em algumas situações havia inclinação do piso. A Figura 7 ilustra este fato.



Figura 7. Exemplo de variação de rotação na amostra utilizada

Outro detalhe que pode modificar um pouco a forma do contorno obtido é a posição da câmera em relação ao veículo, tanto na vertical (altura), quanto na horizontal (mais à direita ou mais à esquerda).

3.2 Segmentação e Pré-processamento

Para realizar a maior parte das operações sobre as imagens, foi utilizado o software Scilab [8], em conjunto com a biblioteca SIVP 0.4.3. O software ImageJ [3], uma ferramenta de processamento e análise de imagens desenvolvida em Java, também foi utilizada em alguns momentos para operações manuais e cálculos rápidos.

A etapa de Segmentação tem a função de isolar os objetos de interesse para a resolução do problema, retirando o fundo da imagem, o chão e outras informações que não são relevantes. Neste trabalho, os objetos de interesse são os automóveis. Portanto, é necessário isolá-los do restante da imagem.

A segmentação automática de objetos, na maioria das aplicações reais, é um problema complexo e demanda um estudo para definição de quais técnicas podem ser aplicadas.

No presente trabalho, serão utilizadas, como entrada para o descritor de forma, imagens binárias contendo apenas o contorno da traseira dos veículos. Para se obter tais imagens, inicialmente tentou-se utilizar funções de detecção de bordas.

3.2.1 Detectores de Bordas

Em testes preliminares, dentre os métodos para extração de bordas disponíveis na biblioteca do Scilab utilizada, o que apresentou melhor resultado quanto ao destaque dos contornos do veículo foi o método *Canny*, utilizado pela função *Edge*. A Figura 8 apresenta um exemplo desta operação.



Figura 8. Detecção de borda através da função *Edge*, usando o método *Canny*

Em seguida, o fundo é preenchido pela cor branca e a imagem sofre uma operação de erosão¹, para eliminar os destaques dos contornos internos da região, pois tais destaques variam muito, dependendo das condições de luminosidade, e também gerariam variações na descrição. Desta forma se obtém uma imagem que representa apenas o contorno da região do modelo em análise, conforme ilustrado na Figura 9. Esta imagem, à direita, já está apta a ser utilizada como entrada do descritor de forma.

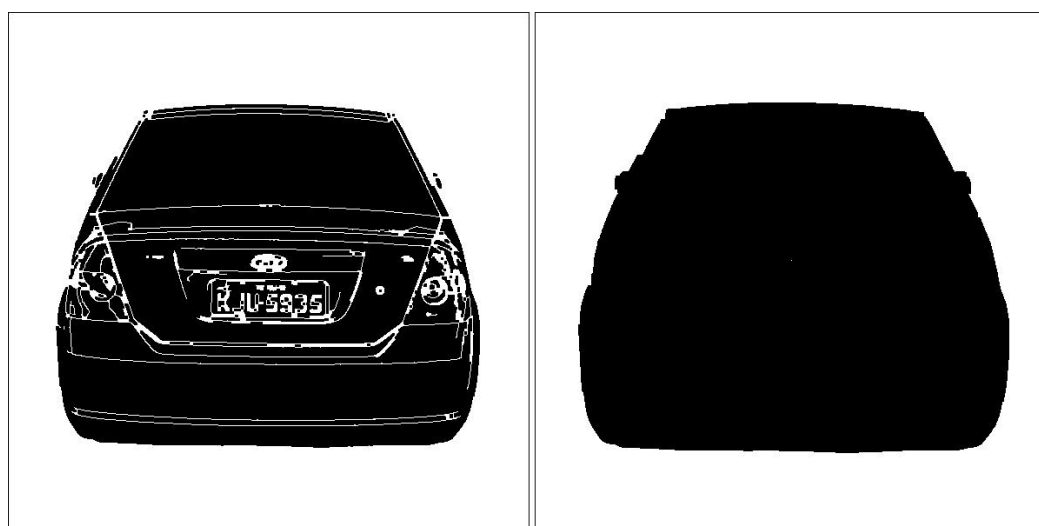


Figura 9. Operações de preenchimento do fundo e erosão

¹ Transformação morfológica que gera encolhimentos na imagem.

Para o exemplo apresentado, o método descrito apresentou um bom resultado, obtendo ao final a representação desejada do objeto em estudo. Mas não é eficiente o suficiente para grande parte das imagens utilizadas neste trabalho. O problema ocorre porque nem sempre a função reconhece a borda do objeto, pois em alguns casos, nas áreas de fronteira entre o objeto e o fundo, há pouca variação de tonalidade. A Figura 10 destaca tal comportamento.



Figura 10. Destaque da falha na detecção da borda

Outro problema observado é que também são destacados os contornos de elementos que pertencem ao fundo da imagem, o que prejudica o processo de isolamento do contorno do veículo. Este problema poderia ser resolvido com o operações para atenuar os contornos, mas isso agravaria o problema anterior, já que também iria prejudicar a detecção das bordas do objeto de interesse.

3.2.2 Diferença entre imagens com fundo fixo

Outra abordagem interessante para resolver o problema da segmentação é o uso de imagens com o fundo fixo. Para isto seria necessário o uso de uma câmera fixa. Com esta abordagem o processo de segmentação pode ser simplificado e o algoritmo consiste em comparar uma imagem primária, que não tem nenhum carro, com as imagens capturadas posteriormente. Assim, todos os *pixels* estranhos à imagem primária são considerados como pertencentes ao veículo.

Num exemplo simples pode-se demonstrar o uso desta técnica. A Figura 11 mostra uma imagem com a rua vazia e outra com alguns veículos. Através do cálculo da diferença absoluta entre as duas imagens e algumas operações adicionais, é possível isolar os objetos de interesse.



Figura 11. Imagens com fundo fixo

A imagem da diferença absoluta (Figura 12 – esquerda), não representa bem os veículos, pois é o resultado do quanto cada *pixel* variou de uma foto para a outra. As imagens parecem fantasmas porque seus valores refletem tanto o que estava representado na primeira como na segunda imagem. Esta imagem da diferença pode ser binarizada (Figura 12 – direita) e usada como máscara para obter os *pixels* originais da imagem que contém os carros.



Figura 12. Diferença absoluta (esquerda) e Máscara binária (direita)

Através da multiplicação da máscara binária pela imagem original, recupera-se os objetos de interesse, isolados do restante da imagem (Figura 13).



Figura 13. Imagem original multiplicada pela máscara

Esta abordagem não foi usada no presente trabalho, pois não foi possível adquirir imagens de uma câmera fixa. Mas aplicações reais para identificação de veículos, como as que motivam este trabalho, freqüentemente usam este tipo de câmera e, portanto, poderiam utilizar esta técnica de segmentação.

3.2.3 Método Utilizado

Diante das dificuldades apresentadas, optou-se por fazer a segmentação dos objetos manualmente, pois o uso de um método de segmentação automática inadequado poderia embutir nos resultados finais a precisão do algoritmo, gerando resultados que não avaliariam corretamente a precisão do método de descrição de forma em estudo. Para este propósito foi utilizado o software ImageJ, através das ferramentas de seleção disponíveis.

Foram geradas duas formas de representação dos objetos. Na primeira o resultado é uma imagem binária que contem a região delimitada pelo contorno do veículo, conforme pode ser observado na Figura 14 - centro. Este tipo de representação pode gerar ambigüidade no processo de descrição, pois muitos modelos de carros possuem um contorno traseiro semelhante.

Visando aumentar a discriminação entre as classes foi gerada uma segunda forma de representação, também formada pelo contorno, mas desta vez subtraída a região correspondente às lanternas traseiras. O exemplo está apresentado na Figura 14 - direita.



Figura 14. Formas de representação do objeto

3.3 Extração de Características

A etapa de extração de características consiste na obtenção de medidas relevantes que possam ser usada na classificação de padrões. É também uma forma de compressão dos dados, já que reduz a quantidade de informação, devendo eliminar redundâncias e aumentar a distinção entre classes.

No presente trabalho foi utilizada uma função que extrai os sete momentos invariantes de Hu. A entrada da função é a imagem binária que representa o contorno do veículo. A saída é um vetor de sete elementos, contendo os valores absolutos normalizados dos momentos. O código Scilab do algoritmo encontra-se no Anexo A.

A amostra de imagens utilizada apresenta variações quanto a escala, rotação e translação. O motivo de se escolher usar momentos invariantes como descritor de forma é o fato deste método ser capaz de representar propriedades, de imagens, que são invariantes quanto à escala, rotação e translação, conforme exposto no Capítulo 2.

3.4 Classificação

Para a etapa de classificação foram usadas 60 imagens da traseira de veículos, sendo utilizados 10 modelos diferentes. Portanto, há 6 imagens diferentes de cada uma das 10 classes.

Uma imagem de cada classe foi selecionada como imagem de referência. O descritor foi aplicado a todas as imagens, incluindo as de referência. Os vetores de características gerados pelo descritor (v_i) foram comparados com o conjunto de vetores das imagens de referência (v_{rj}), através do cálculo da distância destes vetores: $d_j = \sum \| v_{rj} - v_i \|$, onde i é i -ésima imagem e j é a j -ésima classe. Desta forma, a menor distância definirá a qual classe a imagem pertence.

Capítulo 4

Resultados Obtidos

Os resultados na classificação dos padrões foram divididos em duas partes isoladas. Inicialmente serão apresentados os resultados derivados da primeira forma de representação utilizada, que considera somente o contorno das traseiras dos veículos. Em seguida, os resultados considerando também a forma da lanternas traseiras.

4.1 Representação pelo contorno

A seguir, na Figura 15, são expostos exemplos deste tipo de representação, que como pode ser observado, possui características geométricas semelhantes para diferentes modelos.



Figura 15. Representação do contorno da traseira de 4 classes/modelos

O resultado da classificação utilizando esta abordagem não foi bom, pois os vetores de característica extraídos de classes diferentes apresentam valores próximos, gerando muita confusão entre as classes.

Este comportamento pode ser visto, por exemplo, através da análise dos vetores extraídos das imagens de referência das classes CELTA e CLASSIC, ou das classes PRISMA e FIT (Tabela 1).

Tabela 1. Vetores de atributos das imagens de referência.

CLASSE							
1-CELTA	0.7623	0.0649	4.1766	6.1077	11.9294	3.1429	12.1709
2-CIVIC	0.6717	0.1160	4.4594	7.2735	13.2430	5.0509	14.2221
3-CLASSIC	0.7520	0.0345	4.3099	6.3702	12.1553	3.2263	13.0902
4-COROLA	0.4685	2.7722	1.3790	3.6168	7.1867	3.6137	6.4568
5-FIESTA SEDAN	0.7077	0.2067	3.1587	5.2382	9.9950	2.6976	11.0893
6-FIT	0.6994	0.3770	3.0759	5.3795	10.9177	2.8051	9.6808
7-MERIVA	0.8056	0.4473	6.4635	8.0303	15.6896	4.1100	16.7168
8-P206	0.6541	0.0265	2.5543	4.9623	9.6078	4.0832	9.3347
9-PRISMA	0.6926	0.2031	3.7177	5.8519	11.1555	2.9864	11.6839
10-UNO	0.7628	0.2787	4.3111	5.9884	11.8298	4.7428	11.9337

O erro médio computado pela classificação através da distância mínima dos vetores foi de 88%, variando entre 80% e 100%, dependendo da classe, conforme gráfico a seguir.

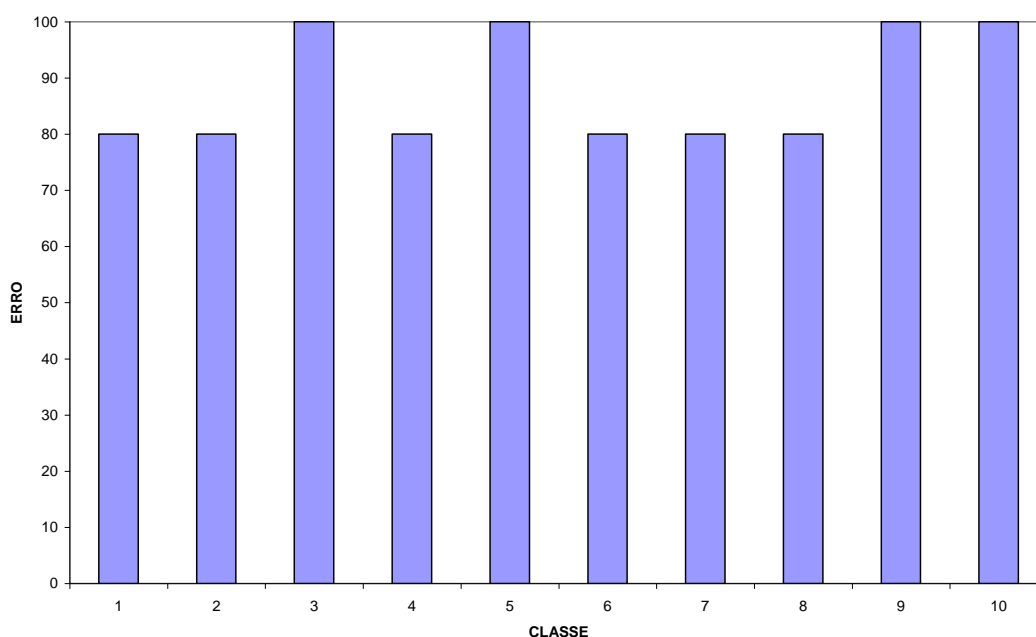


Figura 16. Taxa de erro na classificação

4.2 Representação pelo contorno e lanternas

A segunda forma de representação adotada também é formada pelo contorno do veículo, mas desta vez é subtraída a região correspondente às lanternas traseiras. A Figura 17 mostra alguns exemplos.



Figura 17. Segunda forma de representação utilizada

Os vetores gerados a partir deste forma de representação resultaram numa maior distinção entre as classes. Apesar dos vetores de referência das classes CELTA e CLASSIC ainda guardarem um certa semelhança, houve menos confusão entre as classes na classificação, pois as demais classes apresentaram maior variação quanto aos valores dos atributos extraídos, conforme pode ser observado na Tabela 2.

Tabela 2. Vetores de atributos das imagens de referência.

CLASSE							
1-CELTA	0.4833	0.3228	6.7799	8.4419	16.8111	4.3213	16.7831
2-CIVIC	0.1960	0.6655	4.9287	7.3299	14.1012	4.8252	14.2028
3-CLASSIC	0.4276	0.5402	6.2685	8.1606	15.7548	4.1152	16.9045
4-COROLA	0.2594	0.5708	7.2352	11.9631	25.0571	6.7701	21.7234
5-FIESTA SEDAN	0.3686	0.3705	5.6818	7.6142	14.8366	3.8523	15.1654
6-FIT	0.3836	0.2297	4.8917	7.2500	15.7847	3.6878	13.3734
7-MERIVA	0.4153	0.8677	4.9240	6.8748	13.3797	3.4922	13.6762
8-P206	0.3538	0.5458	3.5785	5.9742	13.2201	4.3684	11.1877
9-PRISMA	0.2481	0.8414	5.3969	7.9128	15.0027	5.0459	15.7150
10-UNO	0.4197	1.0043	7.9745	9.1931	18.6925	7.1058	18.0082

Utilizando esta abordagem, o erro médio de classificação foi de 36%. A variação do erro de acordo com a classe pode ser observada a seguir.

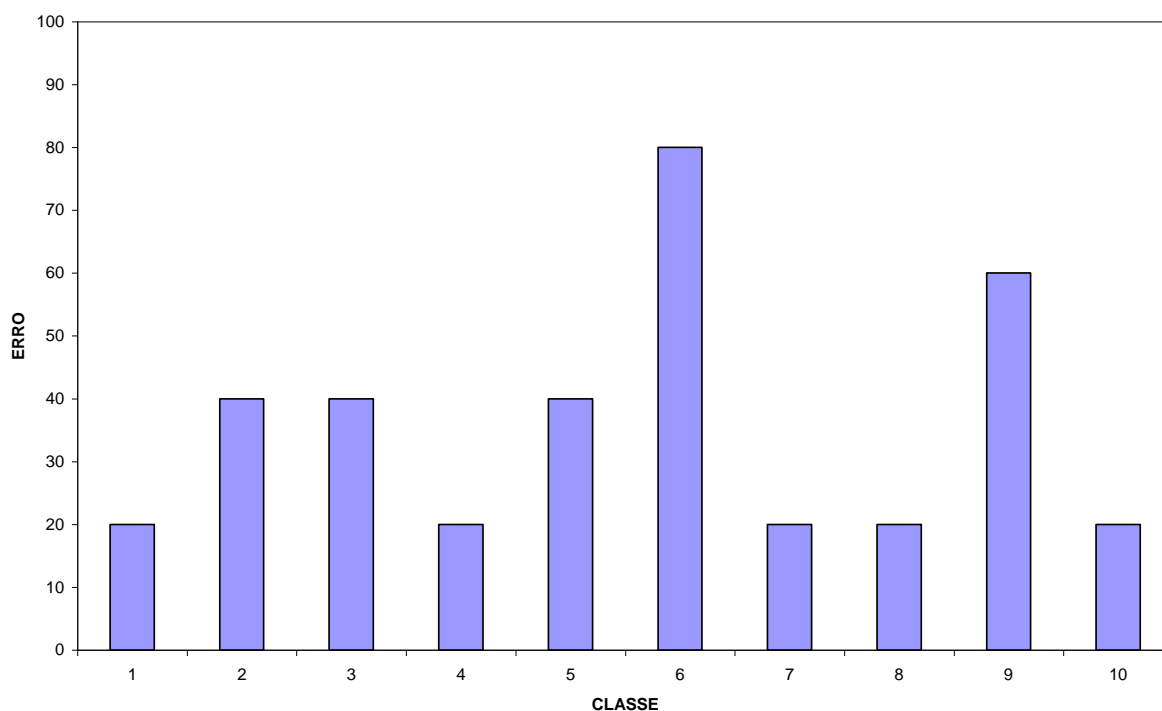


Figura 18. Taxa de erro na classificação

Analisando o resultado e os vetores de características, pode-se observar que o algoritmo de extração dos momentos não foi eficiente na descrição de algumas classes.

Na classe 6 (FIT), que teve erro de 80%, houve muita variação de valores nos vetores gerados dentro da própria classe, o que indica que o descritor utilizado é sensível a pequenas variações de forma, que podem resultar dos processos aquisição das imagens ou de seleção do objeto.

Este problema pode ser observado na Figura 19 e na Tabela 3, onde são expostos os objetos da classe 6 e os seus vetores de atributos, respectivamente.

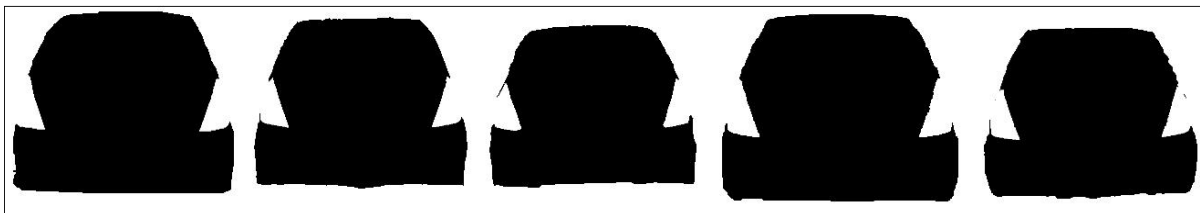


Figura 19. Imagens da classe 6

Como pode ser observado o formato geral dos objetos é muito semelhante. Porém, pequenas diferenças podem ter resultado em atributos com valores bastante diferentes, como, por exemplo, o atributo 7 dos vetores extraídos das imagens FIT3 e FIT4. Considerando que todos pertencem à mesma classe, isso não deveria ocorrer.

Tabela 3. Vetores de atributos das imagens usadas na classificação da classe 6.

	1	2	3	4	5	6	7
FIT2	0.4648	0.0101	4.1994	6.584	12.465	3.4332	12.2686
FIT3	0.3299	0.7402	5.7515	8.3611	15.7446	5.222	17.7226
FIT4	0.3546	0.6553	3.6875	5.8848	13.0173	4.3617	10.6575
FIT5	0.3871	0.2048	4.7993	7.1435	14.9281	3.6442	13.1741
FIT6	0.3598	0.5888	5.6119	7.4738	14.0552	5.3177	15.7155

Outra classe que não obteve um bom resultado na classificação (60% de erro) foi a classe 9 (PRISMA). Desta vez o problema identificado foi a confusão com as classes 1 e 3. Algumas imagens foram classificadas como pertencentes a estas classes devido à semelhança com os seus vetores de referência.

O objeto PRISMA2 foi classificado como pertencente à classe 3, enquanto que o PRISMA4 e PRISMA5 como pertencentes à classe 1, conforme destaca a Tabela 4.

Tabela 4. Destaque da semelhança entre vetores de classes distintas.

	1	2	3	4	5	6	7
REF-CLASSE1	0.4833	0.3228	6.7799	8.4419	16.8111	4.3213	16.7831
REF-CLASSE3	0.4276	0.5402	6.2685	8.1606	15.7548	4.1152	16.9045
PRISMA2	0.3508	0.3192	6.0073	8.2233	15.8157	4.1256	15.9881
PRISMA3	0.2524	0.8305	5.3911	7.9186	15.0917	5.0535	15.4573
PRISMA4	0.3185	0.4429	6.2357	8.6621	17.3167	4.3397	16.2439
PRISMA5	0.2660	0.8470	6.1846	9.1234	18.2427	5.5650	17.0729
PRISMA6	0.2465	0.8534	5.2789	7.7699	14.6771	4.9858	15.8604

Os erros de classificação nas demais classes são resultantes da combinação destes dois fatores exemplificados com as classes 6 e 9.

As classes 2, 3 e 5 apresentaram erro de classificação de 40%, enquanto as classes 1, 4, 7, 8 e 10 de 20%. O erro médio, considerando as 10 classes, foi de 36%, o que pode ser considerado próximo ao aceitável, mas indica que os métodos utilizados neste trabalho podem ser aprimorados.

Os resultados indicam que o descritor adotado é sensível a pequenas variações de forma e que, portanto, depende de uma segmentação precisa e do uso de classificadores mais robustos quando as classes em análise possuem características semelhantes.

Capítulo 5

Conclusão e Trabalhos Futuros

O presente trabalho mostrou que é possível realizar a identificação de modelos de veículos com o uso de métodos de descrição de forma e técnicas de processamento de imagens.

Os resultados indicam que o descritor utilizado, que extrai os sete momentos invariantes de H_u , não é eficiente o suficiente para classes com elementos semelhantes. Com o objetivo de aumentar a diferenciação entre as classes, trabalhos futuros podem analisar o uso combinado deste descritor com outros, aumentando, assim, a quantidade de atributos extraídos. Além disso, sugere-se também o estudo de um método de classificação mais robusto e adaptável.

Uma outra abordagem que também pode ser estudada é a adaptação de técnicas usadas no reconhecimento de faces, que conseguem caracterizar eficientemente tais objetos, pois tais técnicas são sensíveis aos detalhes de cada face.

É importante destacar as dificuldades encontradas para realizar uma segmentação automática eficiente dos objetos de interesse, considerando que não foram utilizadas condições controladas para a aquisição das imagens. Sugere-se, como trabalho futuro, a realização de experimentos com imagens obtidas por câmeras fixas e suspensas, de forma a reproduzir mais fielmente as condições existentes em aplicações reais de identificação de veículos. Isto possibilitaria o estudo e o teste de técnicas de segmentação direcionadas a este tipo de situação.

Bibliografia

- [1] GONZALEZ, R.C.; WOODS, R.E. Processamento de Imagens Digitais. Edgard Blücher, 2000.
- [2] HU, M. (1961). *Pattern recognition by invariant moments*. Proc. IRE Transactions on Information Theory, 179-187.
- [3] Image Processing and Analysis in Java - ImageJ. Disponível em <http://rsb.info.nih.gov/ij/>. Acessado em 15 de abril de 2010.
- [4] JORGE, G.H.R. Arquitetura para Extração de Características Invariantes em Imagens Binárias Utilizando Dispositivos de Lógica Programável Complexa , 2006, São Carlos.
- [5] KORBES, A. ; FREITAS, G.M. Estudo Comparativo de Descritores para Recuperação de Imagens Binárias , 2008, Campinas.
- [6] PEDRINI, H. ; SCHWARTZ, W.R. Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações. Thomson Learning Edições Ltda, 2007.
- [7] RONDA, T.F; MARQUES P.M.A; FELIPE, J.C. (2008). Avaliação Comparativa de Descritores de Forma Utilizados para Caracterização e Classificação de Nódulos de Mama. Apresentado no XXI Congresso Brasileiro de Engenharia Biomédica.
- [8] Scilab Group. Disponível em <http://www.scilab.org/>. Acessado em 10 de Março de 2010.
- [9] THEODORIDIS, Sergios. Pattern Recognition. Academic Press, 1999.

Anexo A

Código Scilab para extração de momentos invariantes de Hu

```

function [invmoments] = inmoments(im)
    [rows,cols] = size(im);
    x = ones(rows,1)*[1:cols];
    y = [1:rows]'*ones(1,cols);

    // Area e perimetro
    area = sum(sum(im));
    meanx = sum(sum(double(im).*x))/area;
    meany = sum(sum(double(im).*y))/area;

    // Momentos Centrais.
    n20 = sum(sum(double(im).*((x - meanx).^2)))/(area.^2);
    n02 = sum(sum(double(im).*((y - meany).^2)))/(area.^2);
    n11 = sum(sum(double(im).*(x - meanx).*(y - meany)))/(area.^2);
    n30 = sum(sum(double(im).*((x - meanx).^2)))/(area.^(5/2));
    n12 = sum(sum(double(im).*(x - meanx).*(y - meany).^2)))/(area.^(5/2));
    n21 = sum(sum(double(im).*((x - meanx).^2).*(y - meany)))/(area.^(5/2));
    n03 = sum(sum(double(im).*((y - meany).^2)))/(area.^(5/2));

    // Momentos Invariantes
    phi1 = n20 + n02;
    phi2 = ((n20 - n02).^2) + 4.*(n11.^2);
    phi3 = ((n30 - 3.*n12).^2) + ((3.*n21 - n03).^2);
    phi4 = ((n30 + n12).^2) + (n21 + n03).^2;
    phi5 = (n30 - 3.*n12).*(n30 + n12).*((n30 + n12).^2 - 3.*(n21 + n03).^2) + (3.*n21 -
n03).*(n21 + n03).*(3.*(n30 + n12).^2 - (n21 + n03).^2);
    phi6 = (n20 - n02).*((n30 + n12).^2 - (n21 + n03).^2) + 4.*n11.*(n30 + n12) + (n21 + n03);
    phi7 = (3.*n21 - n03).*(n30 + n12).*((n30 + n12).^2 - 3.*(n21 + n03).^2) + (3.*n12 -
n30).*(n21 + n03).*(3.*(n30 + n12).^2 - (n21 + n03).^2);

    invmoments = [abs(log(phi1)), abs(log(phi2)), abs(log(phi3)), abs(log(phi4)),
abs(log(phi5)), abs(log(phi6)), abs(log(phi7))];
endfunction

```