

TÉCNICAS DE PDI E INTELIGÊNCIA ARTIFICIAL APLICADAS AO RECONHECIMENTO DE PLACAS DE CARRO NOS PADRÕES BRASILEIROS

EDSON CAVALCANTI NETO*, PEDRO PEDROSA REBOUÇAS FILHO†

* *Av. Treze de Maio, 2081*

Benfica

Fortaleza, Ceará, Brasil

† *Av. Contorno Norte, 10*

Parque Central Distrito Industrial

Maracanaú, Ceará, Brasil

Emails: edsoncavalcanti@ifce.edu.br, pedrosarf@ifce.edu.br

Abstract— Techniques of Digital Image Processing techniques together with Pattern Recognition and Computational Intelligence appear to be necessary in recognizing characters in car plates. This happens due to the need to monitor the rotation of cars based on the plates, with the large number of drivers who break the caster. In addition, other applications are also possible, such as identification exceeding speed limits with photosensors, and even popular car tracking on highways. In this sense, this paper proposes a new system for detection, segmentation and recognition of license plates in the Brazilian pattern. Then, is used Digital Image Processing techniques to extract the characters and Pattern Recognition techniques to identify the numbers and letters in the board. As results, the system achieved a 97.3% accuracy rate in the extraction and recognition of numbers and a rate of 96.16% in the extraction and recognition of letters. Therefore, the final system has an accuracy rate of approximately 94%, approaching the minimum standards required by DENATRAN which is 95%. Thus, it can be concluded that the system developed demonstrates the potential for the development of commercial applications.

Keywords— Computer Vision, Image Processing, car plate.

Resumo— Técnicas de Processamento Digital de Imagens juntamente com técnicas de Reconhecimento de Padrões e Inteligência Computacional mostram-se necessárias no reconhecimento dos caracteres de placas de carro. Isto acontece devido a necessidade de fiscalizar o rodízio dos carros baseados nas placas, frente ao grande número de motoristas que infringem o rodízio. Além disto, outras aplicações também são possíveis, como identificação dos carros que ultrapassam a velocidade em fotosensores, e até mesmo rastreamento de carro procurados em rodovias. Neste sentido, o presente trabalho propõe um novo sistema de Detecção, Segmentação e Reconhecimento de placas de carros no padrão brasileiro. Para isto, são utilizada técnicas de Processamento de Imagens para extrair os caracteres e técnicas de Reconhecimento de Padrões para identificar os números e as letras presentes na placa detectada. Nos resultados encontrados, o sistema obteve uma taxa de acerto 97,3% na extração e reconhecimento dos números e uma taxa de 96,16% na extração e reconhecimento das letras. Com isso, o sistema final possui uma taxa de acerto de aproximadamente 94%, aproximando-se dos padrões mínimos exigidos pelo DENATRAN que é 95%. Deste modo, pode-se concluir-se que o sistema desenvolvido demonstra potencial de desenvolvimento de aplicações comerciais.

Palavras-chave— Visão computacional, Processamento Digital de Imagens, placa de carro.

1 Introdução

Os países desenvolvidos já utilizaram, com sucesso, a maioria das técnicas e soluções de curto prazo e baixo custo, na tentativa de controlar o problema dos acidentes de trânsito. Atualmente, melhorias significativas são difíceis de serem alcançadas se não houver uma compreensão mais ampla e profunda do processo que leva à ocorrência de um acidente. Para os países como o Brasil, que ainda enfrentam problemas decorrentes de uma infraestrutura viária deficiente, já é chegado o momento de buscar novas abordagens com relação a segurança de trânsito (Aguiar et al., 2007).

Em virtude do constante crescimento da frota de veículos, principalmente nos grandes centros urbanos, surgem problemas cada vez mais complexos e difíceis de serem prontamente corrigidos e solucionados. Os setores que tratam da engenharia de trânsito têm dificuldades em conseguir

informações rápidas e precisas sobre os veículos em trânsito, não só para efetuar o controle e o monitoramento do trânsito, mas também para fins de segurança e planejamento (Guingo et al., 2002).

Nos últimos anos, sistemas inteligentes de controle de trânsito vêm sendo utilizado em pedágios, monitoramento em tempo real e sistemas de estacionamento (Kato et al., 2002). Estes sistemas podem utilizar câmeras para identificação de veículos e placas utilizando Reconhecimento de Caracteres.

Este trabalho tem como objetivo o desenvolvimento de técnicas de Processamento Digital de Imagem e Inteligência Computacional com o foco em sistema de detecção e reconhecimento de placas de veículos nos padrões brasileiros.

2 Revisão Bibliográfica

Um sistema de Visão Computacional típico possui 6 etapas. A etapa de aquisição de imagens, que consiste no processo de captura da imagem, enquanto a etapa de pré-processamento consiste em retirar falhas e imperfeições da mesma (Gonzalez et al., 2009). A segmentação é a etapa responsável por dividir a imagem em regiões de interesse (Gonzalez et al., 2009). Posteriormente, a região segmentada é utilizada para extrair atributos, utilizados para identificar padrões previamente estabelecidos. Por fim, é gerado um relatório com as análises e informações obtidas (Gonzalez et al., 2009). Uma ilustração das etapas de um sistema típico de Visão Computacional pode ser observado na Figura 1.

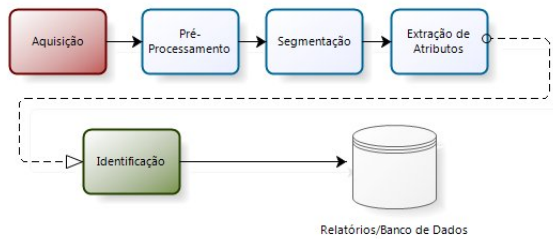


Figura 1: Etapas de um sistema de Visão Computacional típico.

Uma imagem é uma matriz de *pixels*, em que o valor de cada *pixel* pode ser obtido por uma função bidimensional $f(x,y)$, onde os valores de x e y representam a posição do ponto no plano cartesiano e a amplitude f é chamada de intensidade ou nível de cinza (Gonzalez et al., 2009).

As outras etapas de um sistema de Visão Computacional são a segmentação da imagem, ou seja, a etapa na qual o objeto de interesse é obtido a partir da imagem, a extração de atributos que consiste na extração de dados a partir da imagem que servirão como entrada no sistema de reconhecimento e por último a identificação do objeto através da técnica de Reconhecimento de Padrões.

Este trabalho utiliza técnicas de Processamento Digital de Imagens aplicadas nas etapas de pré-processamento, segmentação, pós-processamento e extração de atributos. Dentre estas técnicas, destacam-se os filtros operadores gradientes, limiarização, filtro de Canny e a transformada de Hough.

Após a extração de atributos, são utilizadas técnicas de Inteligência Computacional Aplicada (ICA) ou Reconhecimento de Padrões (RP) para identificar o padrão analisado. Neste trabalho, é utilizada uma Rede Neural Artificial (RNA) do tipo *Multi Layer Perceptron* (MLP).

2.1 Filtros operadores gradiente

Dentre as formas de detectar as bordas de objetos em imagens, destaca-se a utilização de operadores gradiente, podendo-se citar os operadores de Roberts, Prewitt e Sobel (Siegwart and Nourbakhsh, 2004). Uma forma de implementar estes operadores é aplicar uma convolução bidimensional na imagem utilizando máscaras específicas. Neste trabalho é usado o operador Sobel, e a máscara utilizada em sua convolução é dada por (Sonka et al., 2007)

$$g(x, y) \cong \sqrt{(s_1^2 + s_2^2)}; \quad (1)$$

em que

$$s_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; s_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \quad (2)$$

Um exemplo de aplicação do operador gradiente Sobel está ilustrado na Figura 2, em que a imagem original é apresentada na Figura 2(a), a imagem em tom de cinza na Figura 2(b) e o resultado do Sobel na Figura 2(c).

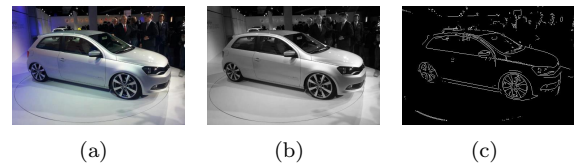


Figura 2: Exemplo de aplicação do operador Sobel para detecção de bordas: a) imagem em RGB, b) imagem em tom de cinza e c) resultado do operador Sobel.

2.2 Filtro de Canny

O filtro detector de bordas Canny realiza dois efeitos, a filtragem de ruídos e encontra os *pixels* que compõem as bordas de objetos nas imagens (Canny, 1986). Para desenvolver este algoritmo, primeiramente concentraram seus estudos em bordas ideais, cuja representação pode ser feita por funções em uma dimensão (1-D). Ele mostrou que o melhor filtro para iniciar o seu algoritmo era um algoritmo de suavização, operador Gaussiano, seguido por um Gaussiano derivativo, que em uma dimensão pode ser dado por (McAndrew, 2004):

$$d = \left(-\frac{x}{\sigma^2} \right) e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

onde σ^2 consiste na variância dos dados e x o dado de entrada.

O algoritmo do Canny foi projetado para possuir estas três propriedades:

- Minimização do erro na detecção. O algoritmo deve marcar muitas bordas reais quanto possível na imagem;

- Localização das bordas. A distância entre as bordas reais da imagem e as bordas encontradas pelo algoritmo devem ser minimizadas;
- Resposta mínima: uma borda dada na imagem deve ser marcada uma única vez, e uma imagem com ruído não deve criar falsas bordas.

2.3 Transformada de Hough

A transformada de Hough é uma técnica de extração de características utilizadas na análise de imagens, Visão Computacional e Processamento Digital de Imagem. O objetivo da técnica é encontrar pontos imperfeitos do objeto comparando com classe de formas pelo processo de votação. Este procedimento de votação é levado em um espaço de parâmetros, a partir do qual os candidatos de objetos são obtidos como máximos locais num espaço acumulador assim chamado, que está explicitamente construído pelo algoritmo para o cálculo da transformada de Hough.

A clássica transformada de Hough estava preocupada com a identificação de linhas na imagem, mas depois a transformada de Hough foi estendida para identificar posições de formas arbitrárias, como círculos ou elipses.

As coordenadas polares podem ser utilizadas para representação de uma linha (Duda and Hart, 1972). Desta forma foi sugerido que linhas poderiam ser completamente parametrizadas usando o comprimento, r , e a orientação, q , do vetor normal para a linha da imagem original. Usando esta parametrização, todo o ponto (x,y) na linha satisfará a equação 4.

$$r = x \cdot \cos(q) + y \cdot \sin(q) \quad (4)$$

Computa-se a transformada de Hough dos *pixels* da imagem, registrando os resultados em um histograma, matriz bidimensional. Cada *pixel* (x,y) do espaço real produz uma linha senoidal no espaço de Hough (Yuen et al., 1989).

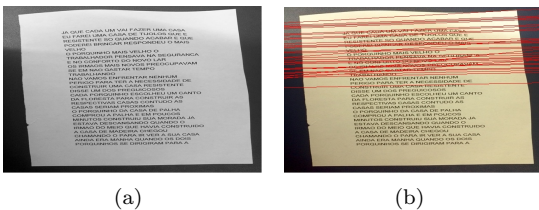


Figura 3: Exemplo de aplicação da transformada de Hough, a) imagem original b) resultado da transformada de Hough.

2.4 Limiarização adaptativa por média

Limiarização é uma técnica na qual se determina um valor, limiar, em que a partir deste valor a

imagem é percorrida e caso o valor do *pixel* $f(x,y)$ atual for menor do que o limiar ele recebe um valor. Caso ele seja maior, recebe outro valor onde estes valores serão determinados em cada aplicação. Existem formas de se definir este limiar, algumas adaptativas e outras fixas. Visando tornar o algoritmo automático e menos dependente das condições do ambiente, todos os limiares implementados e testados são adaptativos. A seguir, é feito um comparativo entre os métodos de limiarização adaptativa a fim de selecionar o mais adequado para a aplicação.

A limiarização por média é um método local e adaptativo que utiliza-se dos valores de níveis de cinza presentes na vizinhança de cada *pixel* $P(x,y)$, para obter o valor de limiar de cada *pixel* $P(x,y)$ através da equação:

$$\mu(x,y) = \frac{1}{N \cdot M} \sum_{i=0}^N \sum_{j=0}^M p(i,j) \quad (5)$$

em que M é a quantidade de vizinhos por coluna, N a quantidade de vizinhos por linha, x a posição horizontal relativa ao início da imagem, y a posição horizontal relativa ao início da imagem, μ o valor limite obtido para cada ponto (x,y) da imagem, p o valor do nível de cinza de um dado ponto (x,y) da imagem (Glasbey, 1993).

O resultado da limiarização adaptativa da média, apesar de uma boa visibilidade dos caracteres, continuidade dos traços, o algoritmo apresenta falhas no processo de segmentação. É visível a ocorrência dos problemas, como um alto grau de ruído no fundo da imagem, que torna inviável a utilização deste método (Glasbey, 1993).

2.5 RNA MLP

A partir de um sistema dinâmico, no qual a caracterização matemática é desconhecida, sabe-se sobre o sistema somente o conjunto de dados de entrada-saída gerados por este em instantes tempos discretos a uma taxa uniforme (Lathi, 2008).

O problema de Reconhecimento de Padrões está ligado a caracterizar matematicamente o sistema dinâmico de uma forma que a saída do sistema na simulação em cada instante i possua um menor erro em relação a saída da rede real (Haykin, 2008). A caracterização matemática do sistema pode ser abordado por diversas métricas, nas quais estas possuem métricas de transformação linear e não linear. Desta forma a rede *Multi-layer Perceptron* (MLP), é uma das métricas que abordam esta caracterização matemática.

A MLP é uma rede supervisionada, ou seja, os pesos de cada camada são reajustados de acordo com o valor da saída da rede comparada com o valor real da saída (Arbib, 2003). A MLP é subdividida em entrada do sistema, pesos da camada oculta, saída da camada oculta que consiste na

entrada da camada de saída, pesos da camada de saída e saída do sistema, podendo ser mais de uma camada oculta (Ruck et al., 1990; Haykin, 2008).

Nesse sentido, a fase de treinamento da MLP pode ser dividida em duas etapas, uma chamada de sentido direto na qual o trânsito de informações flui da entrada até a saída, gerando o valor de saída da rede. Enquanto isso na segunda etapa, chamada de sentido inverso, é utilizado o valor do erro gerado no sentido direto, e este é retropropagado para as camadas intermediárias para ajustar os pesos destas (Bishop, 2006; Widrow and Winter, 1988).

3 Metodologia

Na Figura 4 é ilustrado o fluxograma das etapas do SVC desenvolvido para detecção e reconhecimento de placas de carros nos padrões brasileiros. Para o desenvolvimento deste sistema é utilizada a Linguagem C, e a biblioteca livre openCv. Esta biblioteca foi utilizada nas etapas de PDI devido ser compatível com a linguagem utilizada, além de acelerar o desenvolvimento do sistema, visto que muitas das técnicas utilizadas já estavam implementadas de forma otimizada nes biblioteca.

Os testes foram realizados em duas plataformas diferentes, na qual uma era um PC laptop com processador core i7, 8gb de memoria RAM, placa de vídeo dedicada, a outra plataforma foi um sistema embarcado Linux Debian em plataforma ARM.

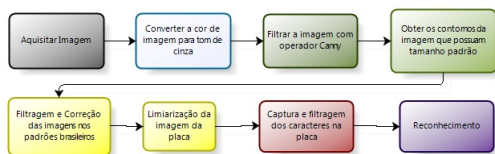


Figura 4: Etapas do SVC desenvolvido para detecção e reconhecimento de placas de carros nos padrões brasileiros.

As etapas de um SVC típico estão ilustradas por cores no fluxograma apresentado na Figura 4, sendo cinza para aquisição, azul claro para o pré-processamento, verde para a segmentação, amarelo para o pós-processamento, vermelho para extração de atributos e roxo para o reconhecimento. Cada etapa deste fluxograma é exemplificada na Figura 5 e executado na seguinte sequência:

1. A primeira etapa do algoritmo é a aquisição da imagem digital. Nesta etapa é utilizada uma câmera USB da marca Logitech de referência C210 (Logitech, 2013). Esta câmera possui sua resolução de captura de 640x480 pixels em uma frequência de 30 imagens por segundo. A vantagem da câmera utilizada é



Figura 5: Resultado dos testes das etapas do SVC desenvolvido.

que possui autofocus e ajuste da imagem em ambientes externos.

2. A segunda etapa do processo é a conversão da escala da imagem originalmente em RGB que utiliza 3 canais de cores para uma imagem que possui 1 canal de cor, chamada de imagem em tom de cinza. Esta conversão da

escala RGB para tom de cinza é realizada através da média dos valores dos *pixels* dos canais R, G e B (Gonzalez et al., 2009).

3. O terceiro passo é a filtragem da imagem com o operador Canny, sendo utilizado neste caso com a finalidade de identificar as bordas dos objetos na imagem.
4. As bordas encontradas na etapa anterior são usadas para segmentar os retângulos na imagem utilizando a Transformada de Hough (Suzuki and Abe, 1985). Os quadrados da imagem que obedecerem a relação altura/largura, com tolerância de $\pm 10\%$ dos padrões brasileiros de placas de carro com dimensões de 40cm x 13cm, são armazenados e processados posteriormente.
5. As placas detectadas na etapa anterior são processadas novamente visando detectar as linhas presentes na mesma utilizando a Transformada de Hough. O coeficiente angular médio de todas as linhas é usado para corrigir a inclinação da placa. Isto é realizado para facilitar a etapa de reconhecimento, mantendo o padrão o mais constante possível.
6. Após a obtenção da placa, a próxima etapa é a identificação e segmentação dos caracteres. Para isto é necessário realizar uma limiarização adaptativa da média sobre a imagem da placa, visando destacar os caracteres do fundo da mesma.
7. De posse dos contornos que existem na placa, o próximo processo é de filtrar estes contornos com a finalidade de encontrar os caracteres de uma placa de carro. Os caracteres, antes de serem enviados para o próximo processo, passam por 4 processos de filtragem, nesta sequência:

- **filtragem pela altura:** verifica a média de todos os possíveis caracteres e filtra todos que estão acima ou abaixo da média com uma tolerância de $\pm 15\%$.
- **filtragem pela largura:** funciona da mesma forma que a filtragem da altura, sendo que este utiliza o parâmetro da largura.
- **filtragem pelo espaçamento:** nesta etapa os objetos, que restaram da etapa anterior, são ordenados utilizando o algoritmo *shell sort* (Azevedo, 1996), possuindo como variável de ordenação o eixo das abscissas. A partir disto, calcula-se o espaçamento entre os objetos, em que uma mediana dos espaçamentos é calculada e aquele vizinho que estiver defasado em $\pm 5\%$ do seu anterior é retirado dos caracteres válidos.

- **filtragem pela quantidade:** após estas filtrações, verifica-se se apenas 7 caracteres existem.

- **filtragem pelo espaçamento entre letras e números:** existindo 7 caracteres, verifica-se se a distância entre o terceiro e quarto caracter é maior que as demais, ou seja, se a distância entre as letras e números é maior.

8. Caso passe por todas as filtrações, a placa é rotulada válida, seus caracteres são redimensionados para um tamanho padrão de 35x35 com fundo preto e caractere branco. Este redimensionamento possui o objetivo de tornar o algoritmo invariante à distância. A altura do caracter é utilizada como referência, sendo redimensionada para 33, e a sua largura permanecerá com a mesma proporção que tinha antes do redimensionamento. Este caracter gerado é centralizado na vertical e na horizontal no padrão 35x35.

4 Resultados

No teste do sistema foi utilizado 700 vídeos e a partir destes foram extraídas 12000 imagens, em que estas imagens foram obtidas nos estacionamentos de instituições de ensino superior de Fortaleza (UFC, UECE e IFCE).

A partir destas imagens, a etapa de PDI extraiu um total de 47808 caracteres dos números de um total de 48000, obtendo assim uma taxa de 99,6%. O PDI extraiu 35895 caracteres das letras de um total de 36000, obtendo assim uma taxa de 99,71%. A etapa do PDI desta forma obteve uma taxa de acerto de 97,02%, pois o processo do PDI é dito válido quando este extrai da imagem os 7 caracteres, onde 3 são das letras e 4 dos números.

A etapa de reconhecimento obteve como entrada um total de 47808 caracteres dos números acertando o reconhecimento de um total de 46751, obtendo uma taxa de acerto de 97,7%. Quanto às letras, a etapa do reconhecimento teve como entrada um total de 35895 acertando um total de 34618 caracteres atingindo uma taxa de acerto de 96,4%.

Desta forma o sistema obteve uma taxa de acerto 97,3% na extração e reconhecimento dos caracteres dos números e uma taxa de 96,16% na extração e reconhecimento dos caracteres das letras. Com isso, o sistema final possui uma taxa de acerto de aproximadamente 94%.

Nas tabela 1 são ilustrados os tempos de processamento, em plataforma i386 e ARM, de uma imagem sem placa, uma imagem com placa mas sem reconhecê-la e outra imagem com placa e sendo reconhecida.

Tabela 1: Tempo de processamento nas plataforma i386 e ARM.

Tipo de imagem	Plataforma	T(ms)
sem placa	i386	35
com placa, sem detecção	i386	45
com placa e OCR	i386	50
sem placa	ARM	345
com placa, sem detecção	ARM	395
com placa e OCR	ARM	400

5 Conclusão

O objetivo geral foi atingido, visto que o sistema obteve sucesso na segmentação das placas dos carros e estas foram reconhecidas apresentando uma taxa de acerto aproximadamente de 94%. Adotando os padrões do DENATRAN (2013), em que a taxa mínima de acerto de um sistema de reconhecimento de caracteres deve ser superior a 95%, então conclui-se que o sistema desenvolvido aproxima-se deste padrão, demonstrando potencial de desenvolvimento de aplicações comerciais.

Como sugestões de trabalhos futuros é desejável uma melhora na utilização do sistema quanto à presença intensa de luz ou a ausência desta. Outra sugestão é a implementação deste sistema para a identificação de placas de motos e o estudo da diminuição do tamanho dos caracteres para diminuir o custo computacional.

Agradecimentos

Agradeço ao CNPq pelo apoio no desenvolvimento deste trabalho.

Referências

- Aguiar, H., Junior, O., Caldeira, A. M., Machado, M. A. S., Souza, R. C. and Tanscheit, R. (2007). *Inteligência Computacional Aplicada à Administração, Economia e Engenharia em Matlab*, Thomson Learning.
- Arbib, M. A. (2003). *The handbook of Brain Theory and Neural Networks*, M.I.T Press.
- Azevedo, A. P. (1996). *Métodos de Classificação de Dados e Análise de suas Complexidades*.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer.
- Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on pattern analysis and machine intelligence*, pp. 679–698.
- Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures, *Commun. ACM* **15**(1): 11–15.

- Glasbey, C. A. (1993). Na analisys of histogram-based thresholding algorithms., *CVGIP: Graphical Models and Image Processing* **55**: 532–537.
- Gonzalez, R. C., Woods, R. E. and Eddins, S. I. (2009). *Digital Image Processing*, 3rd edn, Gatesmark Publishing, Knoxville.
- Guingo, B. C., Rodrigues, R. J. and Thomé, A. C. G. (2002). Reconhecimento automático de placas de veículos automotores através de redes neurais artificiais, *Congresso Brasileiro de Computação*.
- Haykin, S. O. (2008). *Neural Networks and Learning Machines*, Pearson Prentice Hall.
- Kato, T., Ninomiya, Y. and Masaki, I. (2002). Preceding vehicle recognition based on learning from sample images, *IEEE International Conference on Intelligent Transportation Systems* **3**: 252–260.
- Lathi, B. P. (2008). *Sinais e Sistemas Lineares*, Bookman.
- Logitech (2013). *Logitech Webcam C210*, <http://www.logitech.com/en-us/support/webcam-C210?osid=14&bit=64>.
- McAndrew, A. (2004). *Introduction do digital image processing with matlab*, Thomson Learning.
- Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E. and Suter, B. W. (1990). The multilayer perceptron as an approximation to a bayes optimal discriminant function, *IEEE TRANSACTIONS ON NEURAL NETWORKS* **1**.
- Siegrwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*, The MIT Press.
- Sonka, M., Hlavac, V. and Boyle, R. (2007). *Image Processing, Analysis, and Machine Vision*, CL Engineering.
- Suzuki, S. and Abe, K. (1985). Topological structural analysis of digitized binary images by border following, *Computer Vision, Graphics and Image Processing* **30**: 32–46.
- Widrow, B. and Winter, R. (1988). Neural nets for adaptative filtering and adaptative pattern recognition, *IEEE Computer* **21**: 25–39.
- Yuen, H. K., Illingworth, J. and Kittler, J. (1989). Detecting partially occluded ellipses using the hough transform, *Image and Vision Computing*.