

Utilização de Ferramentas Livres em um Curso de Visão Computacional

Hemerson Pistori e Mauro Conti Pereira
Universidade Católica Dom Bosco, UCDB
Av. Tamandaré, 6000 - Jd. Seminário, 79117-900 Campo Grande, MS, Brasil
(pistori,mauro)@ucdb.br

Resumo

Este artigo apresenta os resultados de uma experiência de três anos no oferecimento da disciplina de visão computacional em um curso de Engenharia de Computação. O programa do curso é inteiramente baseado em ferramentas livres, como o ImageJ e o Scilab, com todas as aulas sendo ministradas em laboratórios de computação. Além de apresentar as ferramentas utilizadas, este trabalho expõe a importância desta disciplina como elemento integrador entre diversas áreas da engenharia de computação, matemática e estatística. A metodologia de avaliação utilizada durante a disciplina também é discutida neste artigo.

1. Introdução

O curso de graduação em engenharia de computação¹ da Universidade Católica Dom Bosco², UCDB, no Mato Grosso do Sul, está entre os mais antigos do país, tendo sido implantado em 1996. A proposta do curso busca um equilíbrio entre aspectos de hardware e software, tendo como eixo central o desenvolvimento tecnológico da região. Em 2004, foi iniciado o oferecimento da disciplina optativa em visão computacional, para acadêmicos do penúltimo semestre do curso, que tem duração de 5 anos. A disciplina nasceu em um momento em que o Grupo de Pesquisa em Engenharia de Computação³, GPEC, da UCDB, começava a fortalecer as pesquisas nesta linha, com o retorno de alguns professores que estavam em pós-graduação.

A evolução do plano de ensino da disciplina, além de considerar a literatura didática disponível [11, 9, 15, 20, 3, 14] e outros programas já consolidados no país e no exterior, reflete o conhecimento adquirido e produzido no desenvolvimento de projetos de pesquisa e desenvolvimento do GPEC. Boa parte dos acadêmicos que freqüentam o

curso acabam sendo inseridos ou já estão inseridos em projetos do GPEC, através do programa de iniciação científica e da realização de trabalhos de conclusão de curso integrados aos projetos de pesquisa. O resultado é uma integração harmoniosa e bastante produtiva entre ensino e pesquisa.

A UCDB tem sido também um importante pólo de formação de profissionais e transferência de tecnologia envolvendo software livre. Esta característica refletiu-se também na disciplina de visão computacional, que utiliza intensamente software livre como recurso pedagógico. Na próxima seção será apresentado e discutido o conteúdo programático da disciplina. Em seguida, descreve-se algumas das ferramentas de apoio didático. A seção 4 explica a metodologia de avaliação, enquanto a última seção apresenta as conclusões e sugestões para aprimoramento do plano de ensino da disciplina.

2. Conteúdo Programático

A disciplina de visão computacional, com uma carga horária de 72 horas/aula divididas em 18 semanas, tem dois objetivos principais: (1) oferecer ao acadêmico uma visão abrangente dos principais conceitos, teorias e métodos da área, destacando possíveis aplicações e buscando motivá-lo para estudos mais avançados na pós-graduação e (2) ilustrar a essencial importância dos fundamentos matemáticos estudados nos primeiros anos do curso, nas disciplinas de álgebra, cálculo, estatística e matemática discreta.

Para oferecer uma visão abrangente de uma área tão vasta e em pleno desenvolvimento, optamos por aulas bastante práticas, que buscam despertar no aluno uma compreensão intuitiva e significativa dos conceitos fundamentais da disciplina. Esta abordagem permite que o conteúdo programático da disciplina, apresentado a seguir, seja explorado satisfatoriamente pelos alunos. Optamos também, na medida do possível, e sem prejudicar o primeiro objetivo da disciplina, por um aprofundamento maior nos conteúdos diretamente relacionados com projetos de pesquisa desenvolvidos na instituição, na área da visão computacional. Isto permite uma maior integração entre o ensino e a pesquisa,

1 <http://www.ec.ucdb.br>

2 <http://www.ucdb.br>

3 <http://www.gpec.ucdb>

com algumas atividades diretamente relacionadas com experimentos reais.

1. Introdução à visão computacional
 - (a) História, abrangência e inter-relações (processamento digital de imagens, reconhecimento de padrões, aprendizagem de máquina, inteligência artificial, robótica, etc)
 - (b) Ferramentas de apoio ao desenvolvimento de sistemas de visão computacional
 - (c) Áreas e exemplos de aplicações
2. Formação da imagem e modelos de representação
 - (a) Introdução à visão humana e dispositivos de captura de imagens
 - (b) Iluminação e cor
 - (c) Textura
 - (d) Modelos para representação de imagens
 - (e) Ruído
3. Filtros
 - (a) Convolução em duas dimensões
 - (b) Domínio espacial e domínio das frequências
 - (c) Realce e suavização
 - (d) Detecção de bordas
4. Morfologia Matemática
 - (a) Definições elementares
 - (b) Dilatação e Erosão
 - (c) Abertura e Fechamento
 - (d) Exemplos de algoritmos baseados em morfologia matemática
5. Extração e Seleção de Atributos
 - (a) Momentos Estatísticos
 - (b) Bancos de Filtros
 - (c) Matrizes de Co-ocorrência
 - (d) Análise de componentes principais e análise discriminante
6. Segmentação
 - (a) Limiarização
 - (b) Segmentação por agrupamento (clustering)
 - (c) Segmentação por ajuste de modelos (model fitting)
 - (d) Tópicos em segmentação baseada em análise de textura
7. Rastreamento
 - (a) Conceitos elementares
 - (b) Fluxo Óptico

(c) Filtros Preditivos

8. Tópicos complementares

- (a) Visão estereoscópica
- (b) Realidade aumentada
- (c) Aprendizagem automática e reconhecimento de padrões aplicados à visão computacional
- (d) Métodos estatísticos aplicados a segmentação e reconhecimento de objetos

Os fundamentos matemáticos são revisados e abordados durante toda a disciplina. O conceito de espaço vetorial e de transformações lineares são explorados em conexão com os espaços de cores e espaços de vetores de atributos. Auto-vetores e auto-valores formam a essência de diversos algoritmos de seleção de atributos. O cálculo diferencial multivariável, principalmente o importantíssimo conceito de gradiente, é também trabalhado de maneira bastante intuitiva através dos métodos de detecção de borda e de fluxo óptico. A estatística aparece de maneira central em diversos métodos de segmentação e de rastreamento, além de ser trabalhada também como ferramenta para analisar o desempenho de diferentes algoritmos.

Também a teoria dos grafos e a teoria dos autômatos e das linguagens formais são revisitadas através dos algoritmos de reconhecimento sintático de padrões, entre outros. Alguns dos problemas da visão computacional podem ser modelados como problemas de otimização, remetendo o acadêmico à área de pesquisa operacional, e novamente à teoria dos grafos e a álgebra, que aliás, permeia praticamente todos o curso. Entre as outras áreas exploradas durante a disciplina estão a óptica, a geometria, o cálculo numérico, as estruturas de dados, a teoria dos sinais e sistemas, o processamento digital de sinais, o reconhecimento de padrões, a aprendizagem de máquina, a robótica, a inteligência artificial e a análise de algoritmos.

3. Ferramentas de apoio didático

A utilização exclusiva de softwares livres e gratuitos durante a disciplina deve-se, em primeiro lugar, a disponibilidade de bons softwares e em segundo lugar, a possibilidade de acesso e modificação dos códigos-fonte, fundamental para a realização de experimentos e redução no tempo de desenvolvimento. É importante destacar que trata-se de um curso de engenharia, portanto mais voltado a formação de criadores de novas tecnologias. Trabalhamos, principalmente, com a ferramenta ImageJ [1, 17], o Scilab [16, 10] e com applets Java e animações disponíveis na Internet. Links para todas as ferramentas de apoio, classificadas por tópicos, estão disponíveis na página da disciplina, seção “material de apoio”⁴. Nos próximos parágrafos destacaremos alguns destes recursos.

O pacote ImageJ é uma versão multiplataforma do software NIH Image, para Macintosh. Na disciplina, utilizamos o pacote ImageJ para experimentos com diferentes representações para uma mesma imagem, equalização, detecção de bordas, transformadas de Fourier e Hough ⁵, convolução utilizando diferentes núcleos, operações morfológicas, entre outros. Através do software *Color Inspector 3D* [4] é possível, por exemplo, visualizar histogramas 3D em diferentes espaços de cores, por diferentes ângulos e formas. A Figura 1 ilustra o funcionamento desse software em um imagem colorida.

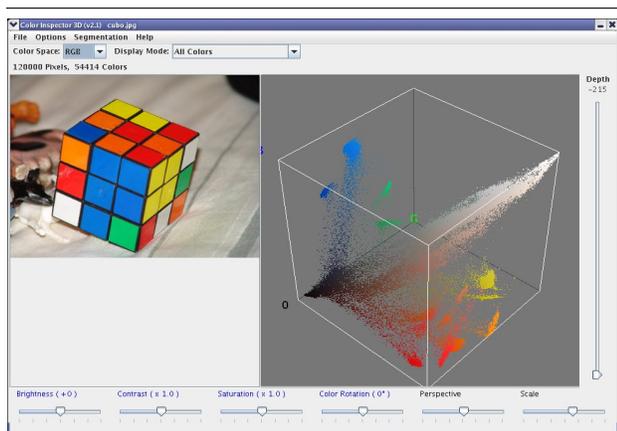


Figura 1. Histograma 3D, no espaço de cores RGB, produzido pelo software *Color Inspector 3D*

Todas as aulas ocorrem no laboratório, de forma que os conceitos podem ser explorados na prática. O ImageJ permite também a inserção de diferentes tipos de ruídos e uma série de ajustes de parâmetros em diversos algoritmos. O aluno é encorajado a experimentar as diversas configurações possíveis, buscando os melhores resultados. Este processo é fundamental para uma compreensão mais profunda dos problemas e das possibilidades para solução. A Figura 2 reproduz a utilização do recurso *plot profile*, do ImageJ, em experimentos com adição de ruídos.

Entre os websites utilizados durante a disciplina, destacamos o do projeto LITE [5], que oferece uma série de applets que permitem a exploração de fenômenos visuais interessantes e que ajudam a motivar o acadêmico pelo assunto. Nos primeiros 30 minutos de cada aula, os alunos,

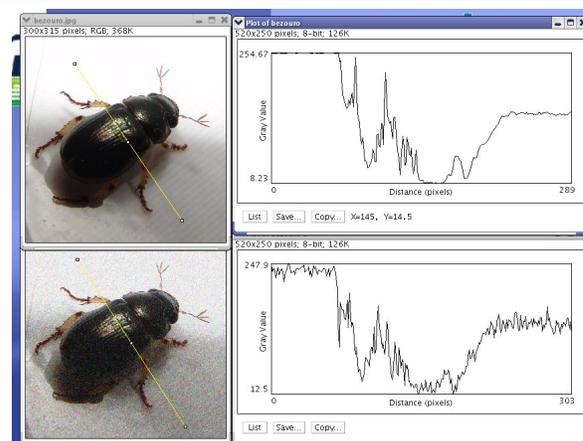


Figura 2. Visualização do efeito de ruído Gaussiano utilizando a ferramenta *Plot Profile* do ImageJ. Na parte superior temos a imagem do bezouro sem adição de ruído, juntamente com o perfil gerado para a linha em amarelo. Na parte inferior temos a imagem e o perfil após adição de ruído.

após serem informados sobre os conteúdos que serão trabalhados durante as aulas, são motivados a buscar na Internet informações relacionadas. Assim, quando a explicação é iniciada, eles já estão “mergulhados” no tema e mais receptivos. Além disso, muitos “descobrem” recursos e links interessantes que passam a fazer parte da página de apoio da disciplina. Entre os applets utilizados na disciplina estão o *Converging Lens* [13], que permite a realização de experimentos simulados de formação de imagens através de lentes convergentes (Figura 3) e o *Visualizing the Gradient (2D) Applet* [12], interessante na revisão do conceito de gradiente, utilizado em diversas áreas da visão computacional, como a detecção de bordas, por exemplo (Figura 4).

O *The Color Spectrum Applet* permite a manipulação do espectro eletromagnético visível, oferecendo em tempo real a cor correspondente ao espectro [19]. A interface deste applet é apresentada na Figura 5. O *The Joy of Convolution Applet* é um ambiente gráfico para experimentos com convolução em 1D [6]. Também são interessantes os applets interativos para experimentos com Snakes [2] e Watershed [18], ambos com opções para alteração de parâmetros e inserção de diferentes níveis e tipos de ruídos. Além dos applets, incentivamos durante a disciplina a utilização da ferramenta Scilab para experimentos com os conceitos de álgebra e estatística. O Scilab é uma alternativa livre e gratuita ao ambiente Matlab. Um toolbox de processamento de imagens, para o Scilab, o SIP [7, 8], está em fase avançada de construção, com diver-

4 A página da disciplina pode ser acessada através do link <http://www.ec.ucdb.br/pistori>, seção “teaching”
5 O módulo para transformadas de Hough foi desenvolvido pelo primeiro autor deste artigo

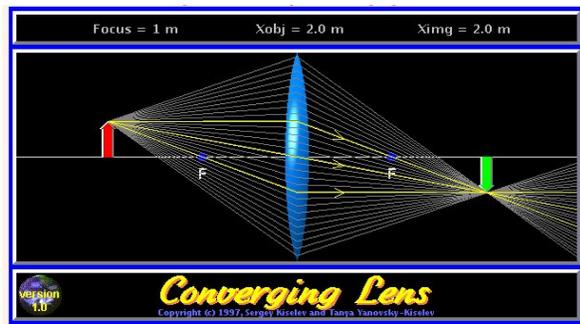


Figura 3. Janela do applet para experimentos com lentes convergente

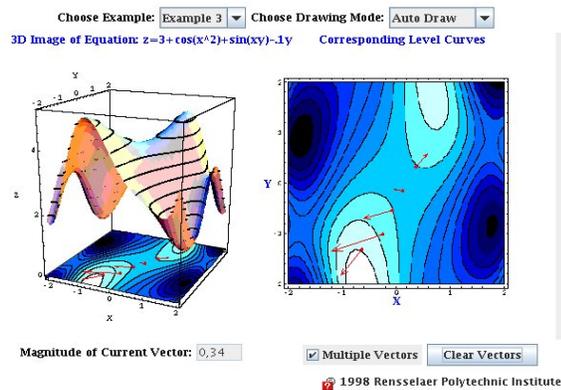


Figura 4. Vetores gradiente, em vermelho, desenhados pelo applet *Visualizing the Gradient 2D*

soos recursos já disponíveis.

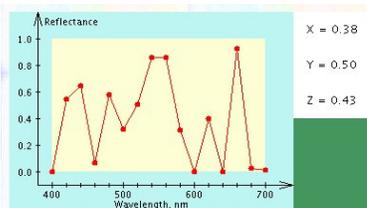


Figura 5. Interface do applet para criação de cores a partir do espectro

A Figura 6 ilustra a utilização do ImageJ em experimentos com transformada de Fourier em duas dimensões.

O acadêmico é instruído a aplicar e visualizar o resultado da transformada em imagens preparadas para acentuar as diferenças no domínio das frequências. Com isso, é possível perceber, na prática, a importância destas transformadas, por exemplo, na análise de textura.

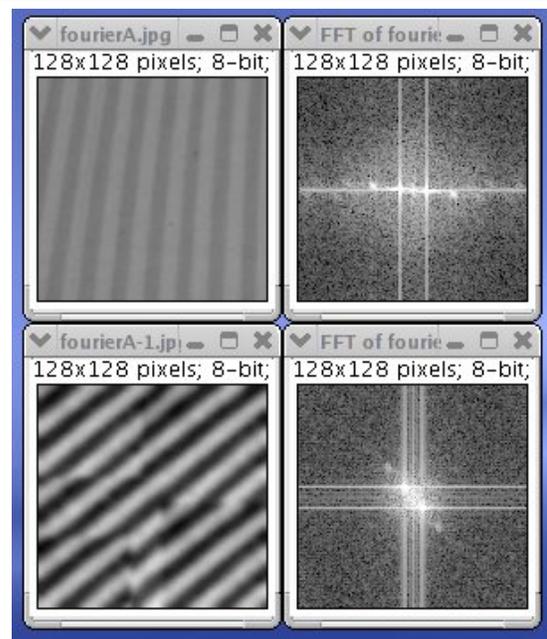


Figura 6. Experimentos com transformadas de Fourier 2D

4. Metodologia de avaliação

Como a disciplina de visão computacional é oferecida no último ano do curso de engenharia, paralelamente ao projeto de conclusão de curso, optamos por uma forma de avaliação que pudesse reforçar as habilidades do acadêmico para conduzir um trabalho de pesquisa e desenvolvimento tecnológico. Sendo assim, o acadêmico é avaliado quanto à capacidade de envolvimento em atividades que exigirão: levantamento bibliográfico, produção de textos técnicos e científicos, implementação de técnicas relacionadas ao conteúdo da disciplina, realização de experimentos e discussão de resultados. Para que o desenvolvimento do acadêmico possa ser avaliado progressivamente, e para que possamos oferecer uma retro-alimentação apropriada, a avaliação prevê as seguintes etapas:

Semana 4 Entrega da primeira versão de um artigo contendo, principalmente, uma boa revisão bibliográfica da área envolvendo o tema escolhido e uma introdução,

descrevendo, em linhas gerais, a justificativa, objetivos e metodologia do trabalho a ser realizado. O instrumento de avaliação nesta etapa é a cópia digital do artigo (de três a seis páginas), seguindo o formato em Latex da Sociedade Brasileira de Computação, SBC⁶.

Semana 10 Entrega de uma versão intermediária do artigo, agora com os objetivos e metodologia bem especificados e detalhados. Esta versão deve conter também alguma informação sobre o desenvolvimento do trabalho, que já deve ter sido iniciado neste ponto. A versão digital do artigo, agora um pouco mais aprimorado, serve de instrumento de avaliação.

Semana 18 Entrega da versão final do artigo, arquivos fontes e material de apoio do software implementado. Além do artigo, utiliza-se nesta etapa os códigos-fonte e material de apoio para reprodução dos trabalhos desenvolvidos.

Para tornar viável o acompanhamento dos trabalhos, e também para fortalecer as habilidades de trabalho em equipe, essencial após a conclusão do curso, recomendamos a divisão da turma em grupos de 2 ou 3 acadêmicos, dependendo do tamanho da turma. Temos trabalhado na UCDB com turmas pequenas, de no máximo 12 alunos, o que tem facilitado o trabalho nesta disciplina. É também fundamental que o retorno das considerações realizadas pelo professor seja rápido (não mais que 7 dias) e detalhado. As duas primeiras etapas são especialmente importantes, pois muitos acadêmicos chegam ao último ano tendo pouca experiência com a metodologia científica e com a escrita de artigos. Temos buscado a reversão deste quadro, na UCDB, através da utilização de avaliação por artigos já a partir do quinto semestre.

Também é importante auxiliar o acadêmico na escolha dos temas, para que a execução do projeto seja viável. Temos tido mais sucesso em projetos envolvendo estudos comparativos de desempenho, aplicados a problemas reais, com bases de imagens disponíveis ou possíveis de serem preparadas, em que implementações já estão disponíveis e que permitam a utilização de ferramentas automáticas para preparação dos resultados. Assim, o acadêmico pode se concentrar na análise dos resultados. Incentiva-se a inserção do projeto da disciplina em um projeto de pesquisa real, de forma que os resultados possam ser utilizados de fato. Esta inserção é facilitada pela existência, na UCDB, de um grupo de pesquisa com uma linha na área de visão computacional⁷. Listamos abaixo alguns dos títulos de projetos, para a disciplina, desenvolvidos nestes três anos:

- Extração de Características Comportamentais de Camundongos em Ambientes Controlados
- Filtro de Partículas para Rastreamento de Múltiplos Camundongos
- Estudo de Técnicas de Rastreamento das Mãos para o Desenvolvimento de Interfaces Homem-Máquina
- Rastreamento da Posição dos Olhos para Detecção da Direção do Olhar

Além do projeto, os acadêmicos podem obter conceitos positivos, através da (1) participação diferenciada em sala de aula e listas de discussão; de (2) participação em atividades extra-classe diretamente relacionadas com os conteúdos da disciplina; de (3) contribuições significativas para a página da disciplina e outros materiais didáticos; e (4) realização de pequenas implementações de alguns métodos necessários para os projetos de pesquisa em desenvolvimento pelo GPEC. Devido a subjetividade inerente a utilização destes conceitos, limitamos a pontuação a um máximo de 5. Instruções detalhadas sobre a forma de entrega e apresentação dos trabalhos são mantidas na página da disciplina.

5. Conclusão

Este artigo apresentou alguns aspectos do plano pedagógico da disciplina de visão computacional, em um curso de graduação em engenharia de computação. Entre os principais pontos deste plano estão a utilização de software livre como ferramenta de apoio didático, a avaliação através de artigos e a integração com a pesquisa. A introdução desta disciplina, no último ano, juntamente com a criação do grupo de pesquisa em visão computacional, tem encorajado os acadêmicos a desenvolver trabalhos de mestrado nessa mesma área. Já temos 5 egressos trabalhando nessa área e 4 acadêmicos da turma atual já demonstraram interesse na pós-graduação em visão computacional. Todo o material de apoio e o plano de ensino da disciplina estão disponíveis através da página da disciplina na Internet.

Na metodologia proposta, é essencial que as aulas aconteçam em laboratórios com equipamentos apropriados para a execução dos softwares de processamento de imagens e com acesso à Internet. Por ser uma disciplina que permite a exploração intuitiva de uma diversidade de conceitos matemáticos importantes para a engenharia, estamos propondo para o futuro o adiantamento do oferecimento desta disciplina para o terceiro ano do curso, aproximando-o assim das disciplinas de fundamentos matemáticos.

Agradecimentos

Este trabalho recebeu apoio financeiro da Universidade

⁶ O modelo da SBC pode ser obtido em <http://www.sbc.org.br/template>

⁷ Informações sobre o Grupo de Pesquisa em Engenharia e Computação da UCDB podem ser obtidas em <http://www.gpec.ucdb.br>

Católica Dom Bosco, UCDB. Os projetos de pesquisa com impacto direto no ensino da disciplina tiveram apoio da Fundação de Apoio ao Desenvolvimento do Ensino, Ciência e Tecnologia do Estado de Mato Grosso do Sul, FUNDECT, e da Agência Financiadora de Estudos e Projetos, FINEP. Alguns dos acadêmicos que participaram da disciplina receberam bolsas PIBIC/CNPQ e da própria UCDB.

Referências

- [1] M. D. Abramoff, P. J. Magelhaes, and S. J. Ram. Image processing with imagej. *Biophotonics International*, 11(7):36–42, 2004.
- [2] M. Andersson. Snakes applet. <http://www.s2.chalmers.se/research/image/Java/NewApplets/ACM/>, Junho 2006.
- [3] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.
- [4] K. U. Barthel. Color inspector 3d. Available at <http://rsb.info.nih.gov/ij/plugins/color-inspector.html>, Junho 2006.
- [5] K. Brecher. Project lite. Available at <http://lite.bu.edu/>, Junho 2006.
- [6] S. Crutchfield. The joy of convolution applet. <http://www.jhu.edu/signals/convolve/>, Junho 2006.
- [7] R. Fabbri. Desenvolvimento de um pacote de processamento de imagens para o scilab. *Projeto de Conclusão de Curso. Departamento de Ciências de Computação e Estatística da Universidade de São Paulo*, 2003.
- [8] R. Fabbri. Sip toolbox. Available at <http://siptoolbox.sourceforge.net/>, Junho 2006.
- [9] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.
- [10] C. Gomez. Scilab. Available at <http://www.scilab.org/>, Junho 2006.
- [11] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Pub. Co., 2002.
- [12] R. P. Institute. Visualizing the gradient 2d applet. <http://www.ibiblio.org/links/applets/appindex/gradientvis.html>, Junho 2006.
- [13] S. Kiselev. Converging lens applet. http://www.physics.uoguelph.ca/applets/Intro_physics/kisalev, Junho 2006.
- [14] D. Marr. *Vision*. Freeman & Co., 1982.
- [15] M. S. Nixon and A. S. Aguado. *Feature Extraction and Image Processing*. Butterworth-Heinemann, 2002.
- [16] P. S. M. Pires and D. A. Rogers. Free/open source software: An alternative for engineering students. In *Proceed. of FIE - Frontiers in Education*, November 2002.
- [17] W. Rusband. Imagej. Available at <http://rsb.info.nih.gov/ij/>, Junho 2006.
- [18] D. Stadelmann. Watershed applet. <http://bigwww.epfl.ch/demo/watershed/index.html>, Junho 2006.
- [19] E. Vishnevsky. The color spectrum applet. <http://www.cs.rit.edu/ncs/color/>, Junho 2006.
- [20] P. F. Whelan and D. Molloy. *Machine Vision Algorithms in Java: Techniques and Implementation*. Springer-Verlag, 2000.