New courseware modules and software for digital image processing.

Mariusz Jankowski

University of Southern Maine

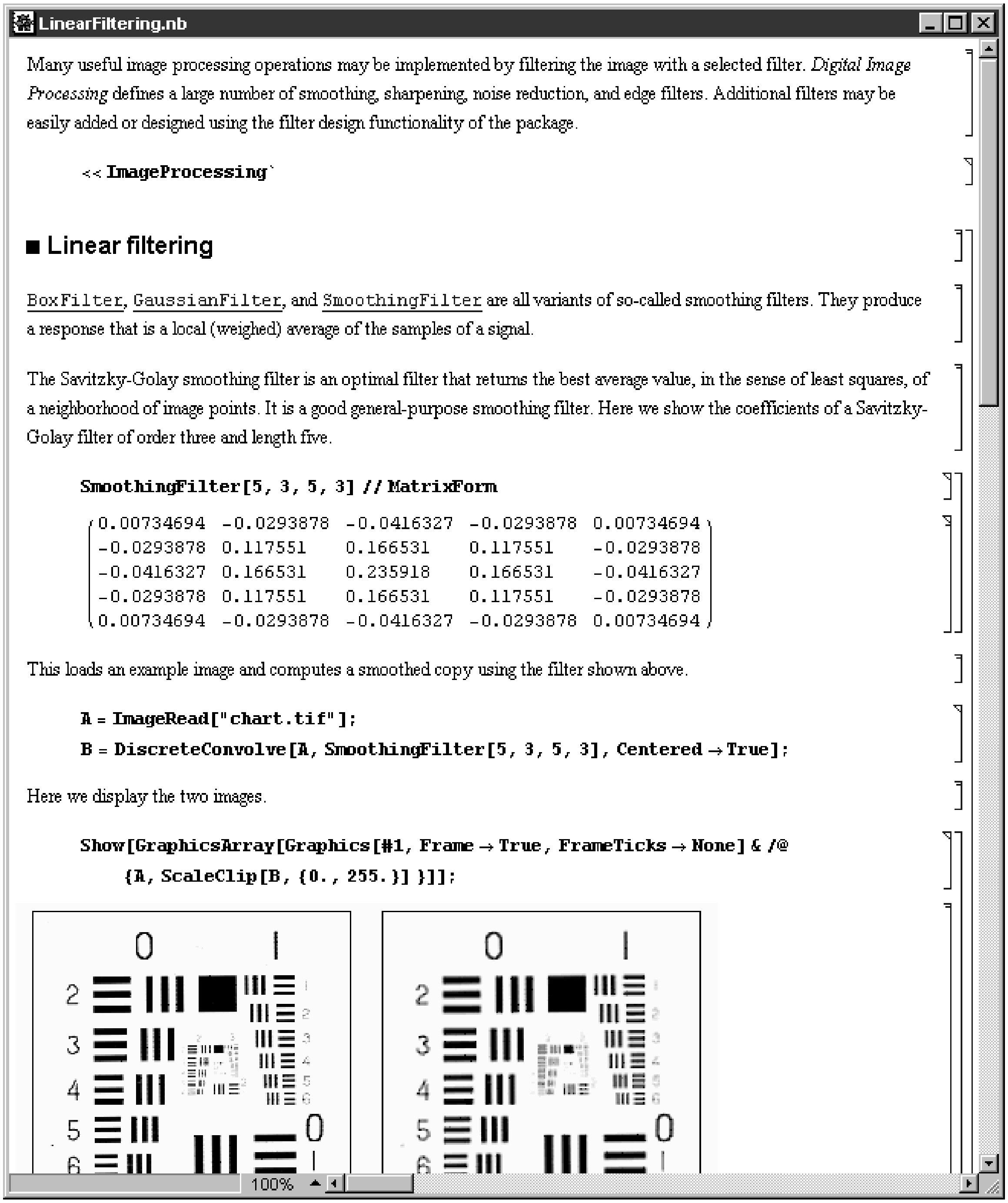
Introduction

Processing of two-dimensional signals in the form of grayscale or color images has become an important research and investigation tool in many areas of science and engineering. The study of image processing has become an integral part of the education of electrical engineering, computer engineering, biomedical engineering and even many computer science students. This paper describes advanced electronic courseware materials for a senior undergraduate electrical engineering technical elective on digital image processing and the software used to support the computational needs of the course.

Background

A few years ago the Department of Electrical Engineering embarked on a plan of integrating a leading general-purpose computational software system into the teaching of many core courses. Thanks to modern mathematical computing systems, it is now possible to replace the typical “chalkboard” lecture with closely supervised interactive, “hands-on” sessions in a computer equipped classroom. These systems have the potential of improving the learning and teaching environments in many sciences and engineering disciplines. The essential feature is the systematic use of a powerful mathematical computing environment that simultaneously forces and empowers the student to be an active participant in the lecture. Ultimately, the goal is to enhance understanding of fundamental theoretical concepts within the discipline by significantly increasing the use of computation and visualization in the learning process. We also use the software to help students overcome deficiencies in their mathematics preparation and to decrease the tedium of frequently needed rote calculations.

In 1996 several software programs were evaluated resulting in the selection of *Mathematica* from Wolfram Research[1]. *Mathematica* was selected because it is uniquely capable in combining excellent documentation features with a powerful programming language and advanced general-purpose symbolic and numeric functionality[2]. This allows the same medium, a *Mathematica* notebook, to be used both for presentation and the student’s own computational work. The notebook is a complete interactive document combining text, tables, graphics, calculations, and other elements. Below is a screenshot of a typical notebook showing text (Times New Roman font), *Mathematica* input (bold **`Courier`** font), output (`Courier` font), and graphics. For a full discussion of *Mathematica* and many example notebooks go to Wolfram Research[1] and related sites. The Courseware Catalog[3] may be of particular interest to engineering educators.



LinearFiltering.nb

Many useful image processing operations may be implemented by filtering the image with a selected filter. *Digital Image Processing* defines a large number of smoothing, sharpening, noise reduction, and edge filters. Additional filters may be easily added or designed using the filter design functionality of the package.

```
<< ImageProcessing`
```

■ Linear filtering

BoxFilter, GaussianFilter, and SmoothingFilter are all variants of so-called smoothing filters. They produce a response that is a local (weighed) average of the samples of a signal.

The Savitzky-Golay smoothing filter is an optimal filter that returns the best average value, in the sense of least squares, of a neighborhood of image points. It is a good general-purpose smoothing filter. Here we show the coefficients of a Savitzky-Golay filter of order three and length five.

```
SmoothingFilter[5, 3, 5, 3] // MatrixForm
```

$$\begin{pmatrix} 0.00734694 & -0.0293878 & -0.0416327 & -0.0293878 & 0.00734694 \\ -0.0293878 & 0.117551 & 0.166531 & 0.117551 & -0.0293878 \\ -0.0416327 & 0.166531 & 0.235918 & 0.166531 & -0.0416327 \\ -0.0293878 & 0.117551 & 0.166531 & 0.117551 & -0.0293878 \\ 0.00734694 & -0.0293878 & -0.0416327 & -0.0293878 & 0.00734694 \end{pmatrix}$$

This loads an example image and computes a smoothed copy using the filter shown above.

```
A = ImageRead["chart.tif"];
B = DiscreteConvolve[A, SmoothingFilter[5, 3, 5, 3], Centered → True];
```

Here we display the two images.

```
Show[GraphicsArray[Graphics[#1, Frame → True, FrameTicks → None] & /@
   {A, ScaleClip[B, {0., 255.}] }]];
```

100%

Since 1996, the software technology within *Mathematica* has continued to improve validating its selection over other capable but not as fully featured products. New features are being continually added and performance has improved with every new version of the software. A recent major release has been shown to perform at levels required for processing large arrays commonly found in speech, audio, or image processing[4]. Furthermore, its already superb presentation features have been enhanced by the development of automatic translation capabilities to web formats such as html, MathML, gif, and other, as well as development of web-server technologies. The latter, in particular, is opening a whole new modality to the way *Mathematica* may be used to deliver instructional materials to a distributed or classroom-based audience.

Courseware materials

The existence of powerful computational environments such as *Mathematica* is necessary but not sufficient to realize the potential of modern computer technologies to improve learning and teaching. There is a clear need for pedagogically sound courseware modules that make use of the existing software tools. The digital image processing courseware materials were designed in support of a senior-level technical elective and currently consist of 15 interactive lecture modules covering the major topics of an advanced undergraduate course on the subject. The modules, which are in the form of *Mathematica* notebooks, are used in a computer-equipped classroom during sessions that mix a traditional instructor-led lecture format with laboratory-based problem solving. This mix of lecture and laboratory combines the best features of these two traditional instructional styles.

The courseware materials presented in this paper were designed to support a senior-level technical elective in an ABET accredited electrical engineering program. They have been used in class since 1996 and have undergone substantial revisions, most recently and significantly, in the spring semester of the 1999/2000 academic year. To the author's best knowledge, the materials which are available online at www.usm.maine.edu/~mjkcc/courses/ele489, are the most comprehensive set of freely available interactive lectures on digital image processing. There are 15 lectures covering the fundamentals of digital image processing. The notebooks supplement a traditional and widely used textbook by R. Gonzalez and R. Woods[5]. The topics covered include:

1. Digital image representation
2. Elements of matrix theory
3. Image histogram
4. Luminance quantization
5. Point operations
6. Geometric operations
7. Linear filtering and convolution
8. Block and region-of-interest processing
9. Image morphology
10. Edge detection
11. Discrete Fourier transform
12. Discrete cosine transform
13. Other image transforms
14. Transform coding
15. Differential coding

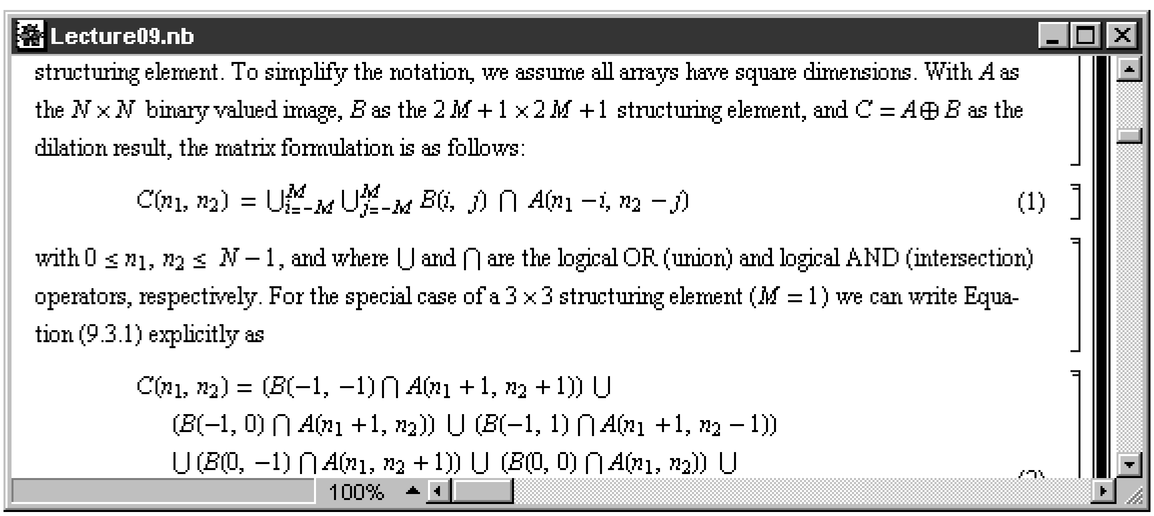
Each lecture consists of introductory material, example calculations, and exercises, and ends with homework problems. The exercises are typically solved in class during the lecture/lab session. They are designed to allow the students to self-test their understanding of the material being discussed. These exercises advance the lecture and allow the students to actively participate in the lecture process. Here are examples of a few typical exercises excerpted from the image processing lectures:

Problem 2.3.1. Verify if a matrix product is commutative, *i.e.* if *X.Y==Y.X*.
Problem 3.2.1. Estimate the histograms of the following simple images:
(a) An all white image. (b) A checkerboard black/white pattern with same number of black and white squares.(c) img/2.
Problem 4.2.2. Calculate the mean squared quantization error for each of the reduced resolution images. Compare with the theoretical result.
Problem 10.2.1. Consider a 2-D signal modeled as a sum of hyperbolic tangents.

```
f[x_,y_]:=(Tanh[3-x]+Tanh[3+x])(Tanh[3-y]+Tanh[3+y])
```

Obtain and plot the gradient magnitude of signal *f(x,y)*.
Problem 11.2.4. Obtain the 32-point DFT of the 1-D sequence defined in Equation (11.2.5) using the matrix formulation.

By repeatedly interacting with *Mathematica* during the course of the semester, the students gain proficiency in using the software to solve common image processing problems. In the process they gain insight into common implementations of many fundamental image processing operators. Typically, with *Mathematica*, these operators may be realized directly from their mathematical definitions, which is particularly significant from a pedagogical perspective. For example, the dilation operator in binary mathematical morphology, can be defined as a convolution-like operator with the algebraic operators of addition and multiplication replaced by the OR and AND operators of Boolean algebra, respectively[5]. This is demonstrated in the following excerpt from the notebook on the subject of image morphology.


Lecture09.nb

structuring element. To simplify the notation, we assume all arrays have square dimensions. With A as the $N \times N$ binary valued image, B as the $2M+1 \times 2M+1$ structuring element, and $C = A \oplus B$ as the dilation result, the matrix formulation is as follows:

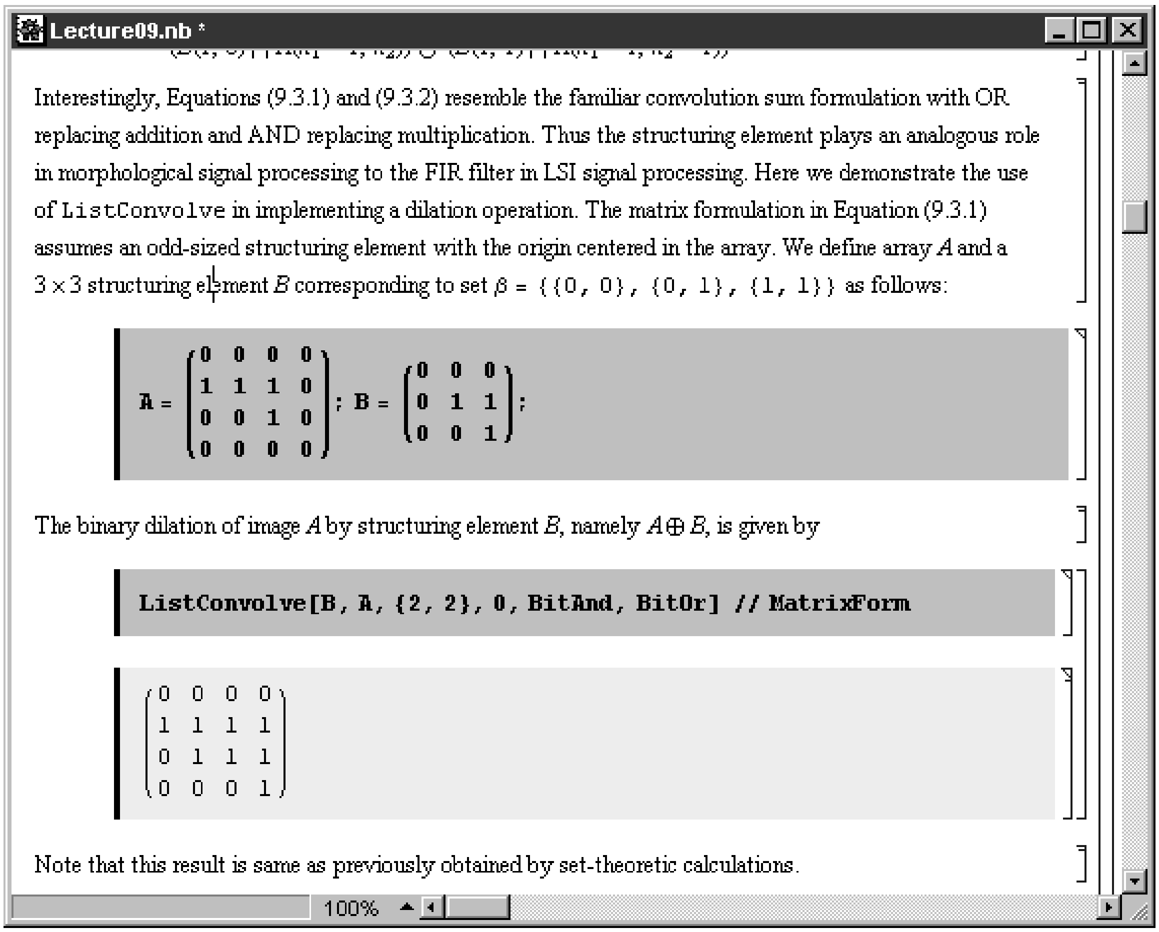
$$C(n_1, n_2) = \bigcup_{i=-M}^{M} \bigcup_{j=-M}^{M} B(i, j) \cap A(n_1 - i, n_2 - j) \qquad (1)$$

with $0 \leq n_1, n_2 \leq N-1$, and where $\bigcup$ and $\bigcap$ are the logical OR (union) and logical AND (intersection) operators, respectively. For the special case of a 3×3 structuring element ($M = 1$) we can write Equation (9.3.1) explicitly as

$$\begin{aligned} C(n_1, n_2) = {} & (B(-1, -1) \cap A(n_1+1, n_2+1)) \cup \\ & (B(-1, 0) \cap A(n_1+1, n_2)) \cup (B(-1, 1) \cap A(n_1+1, n_2-1)) \\ & \cup (B(0, -1) \cap A(n_1, n_2+1)) \cup (B(0, 0) \cap A(n_1, n_2)) \cup \end{aligned}$$

100%

The operation shown in Equation (1) in the previous screenshot can be accomplished in *Mathematica* with a single call to the basic convolution function **`ListConvolve`**. With default arguments **`ListConvolve`** returns the steady-state values of a linear (or circular) convolution of two finite-length sequences (*i.e.*, lists). However its implementation is general enough to admit additional arguments that may be used to replace the default + ("PLUS") and × ("TIMES") operators[6]. This is shown in the following excerpt from the notebook on image morphology, where *Mathematica* input and output cells are marked by medium-gray and light-gray backgrounds, respectively, to easily distinguish them from the surrounding text cells.


Lecture09.nb *

Interestingly, Equations (9.3.1) and (9.3.2) resemble the familiar convolution sum formulation with OR replacing addition and AND replacing multiplication. Thus the structuring element plays an analogous role in morphological signal processing to the FIR filter in LSI signal processing. Here we demonstrate the use of `ListConvolve` in implementing a dilation operation. The matrix formulation in Equation (9.3.1) assumes an odd-sized structuring element with the origin centered in the array. We define array A and a 3×3 structuring element B corresponding to set β = `{{0, 0}, {0, 1}, {1, 1}}` as follows:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \; B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix};$$

The binary dilation of image A by structuring element B, namely $A \oplus B$, is given by

```
ListConvolve[B, A, {2, 2}, 0, BitAnd, BitOr] // MatrixForm
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that this result is same as previously obtained by set-theoretic calculations.

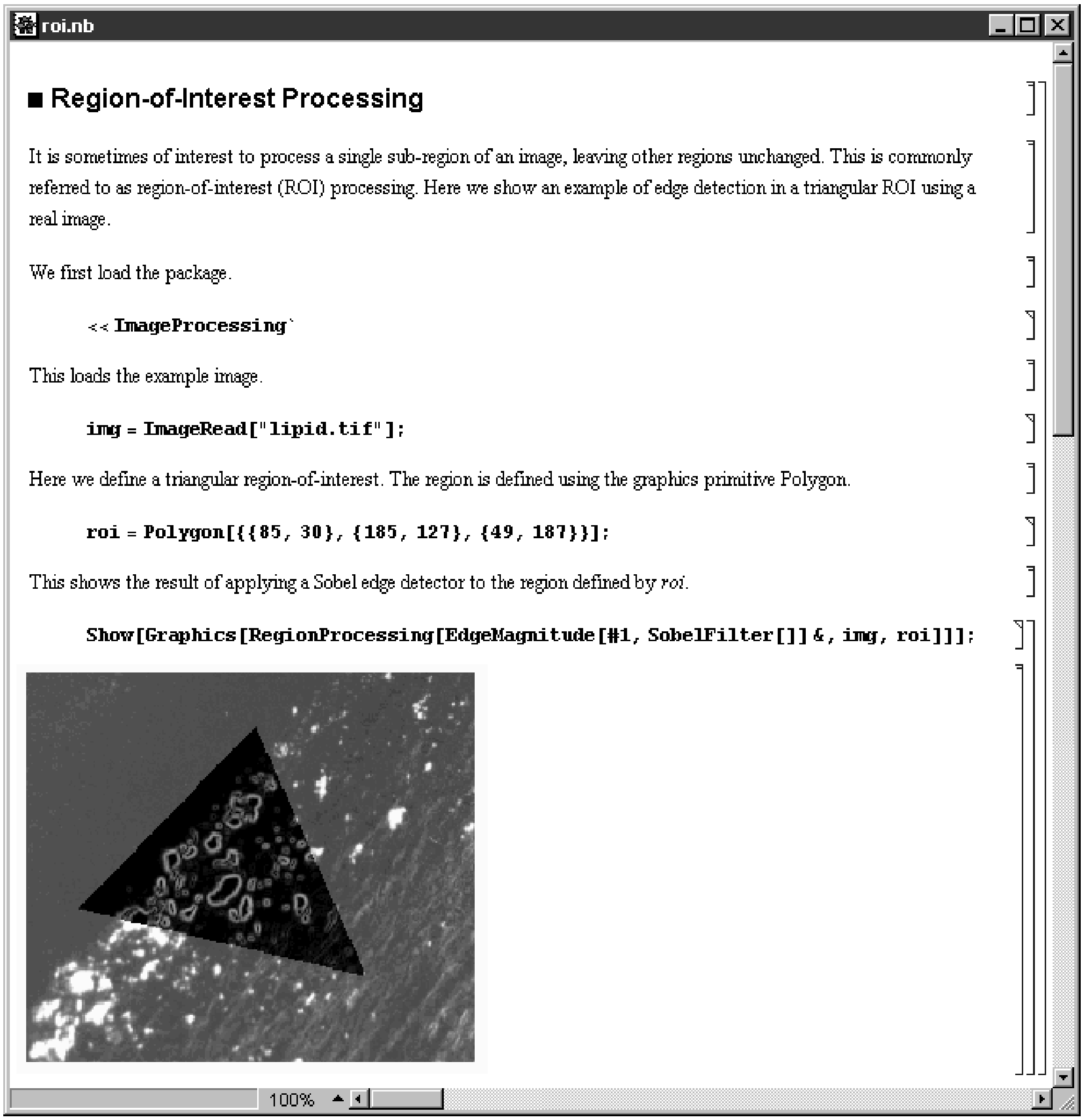
100%

Digital Image Processing Application Package

The development of a digital image processing library begun in 1996. The original plan was to develop a small library of functions to support the image processing courseware modules under development at the same time. The goal was to make it easier for undergraduate students to use *Mathematica* in the classroom. This early version of the software was used for approximately two years and was made largely obsolete by the release of version 4 of *Mathematica*. However, version 4, with its substantial performance improvement opened a much more challenging and interesting opportunity of writing a comprehensive, professional digital image processing extension to

Mathematica. As a result, the Digital Image Processing Application Package[7] was written and in July 2000 was released for commercial distribution.

The Digital Image Processing Application Package, is a so-called add-on package. The software extends *Mathematica*'s functionality in the area of two-dimensional image processing. It is suitable as an image processing research and development tool and a support tool for undergraduate or advanced graduate coursework in image processing. Digital Image Processing was designed for engineers, scientists, medical professionals, and students who need versatile, comprehensive image processing software. The package contains a large assortment of image manipulation, transformation and measurement functions as well as several hundred pages of an interactive image processing tutorial.



Above is a fragment of the User Guide showing an example of edge detection using a Sobel filter in a triangular region-of-interest.

Because the package is completely integrated into *Mathematica,* all its functions can be fully customized and new image processing algorithms easily developed and added. Users can take advantage of this flexibility to free themselves from the constraints found in most of the standard image processing software in use today

The package adds over 160 functions to *Mathematica*'s core functionality. Over a dozen common image formats can be imported and exported. A multiplicity of image data structures and color models are supported. Images may be filtered using linear or nonlinear methods to achieve any desired effect including common operations such as smoothing, edge detection, and noise reduction. The functions defined in the package include:

- Extensive collection of point and area operators
- Image measurement functions
- Intensity profiles over arbitrary polygonal paths
- Histograms and co-occurrence matrices
- Fast integer-factor interpolation and decimation functions
- Resizing, padding, and merging
- Spatial transformations including rotation and arbitrary-order warping
- Area-of-interest processing over arbitrary polygonal regions
- Dozens of predefined filters and morphologic operators
- FIR filter design algorithms
- Fourier, cosine, Hadamard, and wavelet image transforms

The Digital Image Processing package takes full advantage of the many performance enhancements introduced in *Mathematica* 4 including packed arrays, auto-compilation and many fast list operators. Packed arrays are an efficient internal storage format for lists of machine numbers (integers, real, and complex). The advantages of packed arrays are increased speed of computation and reduction in memory use. This results in fast, memory efficient image processing code. For example a linear filtering operation on a grayscale image of size 512-by-512 using a 3-by-3 FIR filter can be processed in under 0.5 seconds. It takes about the same amount of time to calculate a fast Fourier transform of the image. A grayscale image of size 512-by-512 has a memory footprint of a little over 1 MByte or 4 bytes per pixel plus a few bytes of overhead, reflecting the machine representation of an integer on most architectures in use today. These facts confirm that the Digital Image Processing package and *Mathematica* are a versatile, efficient and fast image processing environment.

Bibliography

1. URL: http://www.wolfram.com, Wolfram Research, Inc.

2. M. Jankowski, New electronic courseware modules for selected upper-level electrical engineering courses, ASEE Annual Conference Proceedings, Session 1526, Seattle, WA, (1998).

3. URL: http://library.wolfram.com/courseware, Courseware Catalog, Wolfram Research, Inc.

4. R. Germundsson, Mathematica Version 4, The Mathematica Journal, Vol. 7, Issue 4, (2000).
5. R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley, (1992).
6. M. Jankowski, Signal and image processing with Mathematica V. 4, The Mathematica Journal, Vol. 8, Issue 1, (2001).
7. URL: http://www.wolfram.com/products/applications/digitalimage/; Digital Image Processing application package.

Mariusz Jankowski

Mariusz Jankowski is an Associate Professor of Electrical Engineering at the University of Southern Maine in Portland, Maine. His primary research interests are in the development of novel image processing algorithms for image enhancement, segmentation, shape description, and recognition with application to biomedical imaging and computer vision. He is also the developer of a new commercial image processing software system. Dr. Jankowski received a B.S. degree in Electrical Engineering from the City College of New York in 1980 and a Ph.D. degree in Electrical Engineering from the City University of New York in 1989.