ScienceDirect

Gray Level Image

Consider a gray level image as a function I Ω → □+ where Ω□ □2 is the image domain, typically a rectangle.

From: Handbook of Image and Video Processing (Second Edition), 2005

Related terms:

Image Processing, Discrete Fourier Transform, Histograms, Binary Image

View all Topics

Appendix 4: Color images

Mark S. Nixon, Alberto S. Aguado, in Feature Extraction & Image Processing for Computer Vision (Third Edition), 2012

13.1 Color images

Gray level images use a single value per pixel that is called intensity or brightness, as in Chapter 2. The intensity represents the amount of light reflected or emitted by an object and is dependent on the object's material properties as well as on the sensitivity of the camera sensors. By using several sensors or filters, pixels can represent multiple values for light at different frequencies or colors. In this appendix, we describe how this multivalue characterization is represented and related to the human perception of color. In general, the processing of color images is an extensive subject of study, so this appendix aimed to introduce the fundamental ideas used to describe color in computer vision.

The representation of color is based on the relationships between colored light and perception. Light can be understood as an electromagnetic wave and when these waves hit an object, some light frequencies are absorbed while some others are reflected toward our eye and thus creating what we perceive as colors. Similarly, when the reflected light hits a camera's sensor, it obtains a measure of intensity by adding energy on a range of frequencies. In general, *multispectral* images maintain information about the absorption characteristics of particular materials by maintaining the energy measured over several frequencies. This can be achieved by using filters on the top of the sensors, by using prisms to disperse the light or by including several sensors sensitive to particular frequencies on the electromagnetic spectrum. In any case, color images are obtained by selecting different frequencies. Multispectral images which cover frequencies in the visible spectrum are called color images. Other multispectral images covering other part of the spectrum capture the energy with wavelengths that cannot be perceived by the human eye.

Since (color) cameras have several sensors per pixel over a specific frequency range, color images contain information about the luminance intensities over several frequencies. A color model gives meaning to this information by organizing colors in a way that can be related to the colors we perceive. In color image processing, colors are not described by a frequency signature, but they are described and organized according to our perception. The description of how light is perceived by the human eye is based on the *tristimulus theory*.

> Read full chapter

Compression

StéphaneMallat , in A Wavelet Tour of Signal Processing (Third Edition), 2009

Images

A gray-level image typically has 512×512 pixels, each coded with 8 bits. Like audio signals, images include many types of structures that are difficult to model. Currently, the best image-compression algorithms are the JPEG and JPEG-2000 compression standards, which are transform codes in cosine bases and wavelet bases.

The efficiency of these bases comes from their ability to construct precise nonlinear image approximations with few nonzero vectors. With fewer than 1 bit/pixel, visually perfect images are reconstructed. At 0.25 bit/pixel, the image remains of good quality.

> Read full chapter

Morphological Filtering

Petros Maragos, in The Essential Guide to Image Processing, 2009

13.3.3 Contrast Enhancement

Imagine a gray-level image *f* that has resulted from blurring an original image *g* by linearly convolving it with a Gaussian function of variance 2*t*. This Gaussian blurring can be modeled by running the classic heat diffusion differential equation for the time interval [0,*t*] starting from the initial condition *g* at . If we can reverse in time this diffusion process, then we can deblur and sharpen the blurred image. By approximating the spatio-temporal derivatives of the heat equation with differences, we can derive a linear discrete filter that can enhance the contrast of the blurred image *f* by subtracting from *f* a discretized version of its Laplacian . This is a simple linear deblurring scheme, called *unsharp constrast enhancement*. A conceptually similar procedure is the following nonlinear filtering scheme.

Consider a gray-level image *f*[*x*] and a small-size symmetric disk-like structuring element *B* containing the origin. The following discrete nonlinear filter [15] can enhance the local contrast of *f* by sharpening its edges:

(13.33)

At each pixel *x*, the output value of this filter *toggles* between the value of the dilation of *f* by *B* (i.e., the maximum of *f* inside the moving window *B* centered) at *x* and the value of its erosion by *B* (i.e., the minimum of *f* within the same window) according to which is closer to the input value *f*[*x*]. The toggle filter is usually applied not only once but is *iterated*. The more iterations, the more contrast enhancement. Further, the iterations converge to a *limit (fixed point)* [15] reached after a finite number of iterations. Examples are shown in Figs.13.6 and 13.7.

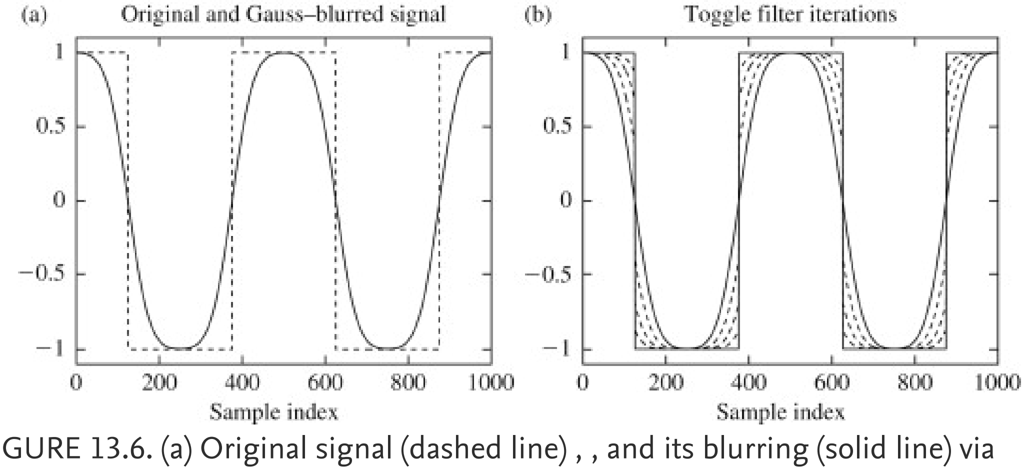

FIGURE 13.6. (a) Original signal (dashed line) , , and its blurring (solid line) via convolution with a truncated sampled Gaussian function of ; (b) Filtered versions (dashed lines) of the blurred signal in (a) produced by iterating the 1D toggle filter (with) until convergence to the limit signal (thick solid line) reached at 66 iterations; the displayed filtered signals correspond to iteration indexes that are multiples of 20.

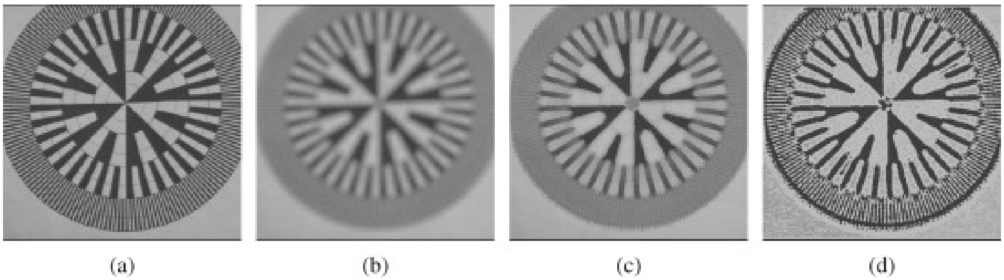

FIGURE 13.7. (a) Original image *f*; (b) Blurred image *g* obtained by an out-of-focus camera digitizing *f*; (c) Output of the 2D toggle filter acting on *g* (*B* was a small symmetric disk-like set); (d) Limit of iterations of the toggle filter on *g* (reached at 150 iterations).

> Read full chapter

Introduction to Basic Image Processing Techniques

In Handbook of Image and Video Processing (Second Edition), 2005

> Read full chapter

Multiatlas segmentation

Bennett A. Landman, ... Andrew J. Asman, in Handbook of Medical Image Computing and Computer Assisted Intervention, 2020

6.3.1 Problem definition

A target gray-level image with *N* voxels is represented as a one-dimensional vector . Given *L* labels, let be the corresponding latent true segmentation to *I*. Since *T* is unknown, the goal is to estimate *T* using *R* atlases (raters) with their intensity values and label decisions . For convenience, we denote true label at voxel *i* () by in the remainder of this chapter.

The goal of any label fusion framework is to accurately estimate the following probability density function:

(6.1)

where can be directly interpreted as the probability that the true label at voxel *i* is equal to label *s* given the provided contextual information. Using a standard Bayesian expansion, Eq. (6.1) can be rewritten by

(6.2)

where represents the prior distribution governing the underlying segmentation and represents distribution governing the relationships between the observed atlas information and the latent target segmentation. Lastly, one of the most common assumptions in the label fusion literature [99] is that the observed atlas labels and the observed atlas intensities are conditionally independent resulting in

(6.3)

While this might seem like it neglects the complex relationships between labels and intensity, the common assumption is that the information gained by direct incorporation of the target/atlas intensity relationships accurately approximates these complex relationships through the assumed conditional independence. With that said, further investigation into estimating the target label probabilities, using joint models of atlas performance is an active area of continuing research [119,120].

Using this general framework, there are two primary fields-of-thought within the label fusion community. First, voting label fusion attempts to find optimal weights in order to determine which atlases are optimally representative in terms of some local/semilocal/global metric. Nevertheless, these techniques are primarily ad hoc and lack a consistent theoretical underpinning. In stark contrast, statistical fusion techniques attempt to model atlas performance using a statistically driven rater performance model. Significantly more detail on these two approaches is provided in the following sections.

> Read full chapter

Geometric Active Contours for Image Segmentation

Vicent Caselles, ... Guillermo Sapiro, in Handbook of Image and Video Processing (Second Edition), 2005

2 Mathematic Notations and Problem Formulation

Consider a gray level image as a function $I\ \Omega \to \square_+$ where $\Omega\square\ \square_2$ is the image domain, typically a rectangle. Although we present the theory for scalar planar images, the model can be easily extended to vector valued images and higher dimensions [13, 64]. The image gradient vector field is given by $\square I(x,y)\ \square\ \{I_x,I_y\}$, were we used subscripts to denote partial derivatives, e.g., $I_x\ \square\ \square I(x,y)/\square x$. We search for a contour, $C\colon [0,L] \to \square_2$, given in a parametric form $C(s) = \{x(s),y(s)\}$, where s is an arc-length parameter, and whose normal is defined by . This contour somehow interacts with the given image, for example, a curve whose normal aligns with the gradient vector field, where the alignment of the two vectors can be measured by their inner product that we denote by . We also use subscripts to denote full derivatives, such that the curve tangent is given by $C_s = \{x_s,y_s\} = \{dx(s)/ds,dy(s)/ds\}$. In some cases we will also use p to indicate an arbitrary (nongeometric) parametrization of the planar curve. In which case, the tangent is $C_s = C_p/|C_p|$, and the normal can be written as

where . We have the well-known relationship between the arc-length s and a general arbitrary parameter p, given by

Define, as usual, □ to be the curvature of the curve C, and the curvature vector [25]. Let us recall that a Jordan curve is a plane curve that is topologically equivalent to (a homeomorphic image of) the unit circle (i.e., it is simple and closed). If C is a Jordan curve, we also define Ωc to be the domain inside the curve C, see Fig. 1. We also assume without loss of generality that the curves are counterclockwise oriented, so that is the inner unit normal vector.

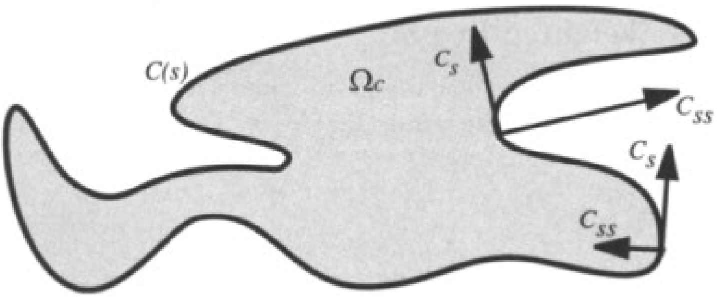

FIGURE 1. A closed curve C, with C_s the unit tangent, the curvature vector, and Ω*c* the region inside the curve.

In this chapter we deal with two types of integral measures that are related via the Green theorem. The first is defined along the curve by the general form

where *g* is a function of the geometry of the curve *C*(*s*) and the underlying image *I*. Under general assumptions, this functional measures the weighted length of *C* (recall that *s* is the arc-length parametrization), where the weight is given by *g*(·). Formally, we search for the optimal planar curve C, such that

that is our desired geodesic. In general, we start from a specific curve and deform it to locate an extremum contour. Thus the name

geodesic active contour.

The second functional integrates the values of the function *f*(*x*,*y*) defined inside the curve, and is usually referred to as a region-based measure,

where as before, Ω*c* is the region inside the Jordan curve C. Under general assumptions, this is a measure of the weighted area of ΩC, where the weight is given by the function *f*(*x*,*y*).

> Read full chapter

Fractal Images and Image Processing

Jonathan M. Blackledge, in Digital Image Processing, 2005

18.7.2 The Correlation Dimension and Signature

Each pixel in a grey-level image can be regarded as a point in a three dimensional space $X_k = [i, j, g_{ij}]$ where *i* and *j* are the spatial coordinates of a pixel and g_{ij} is the grey level at these coordinates. For each pixel, a cube of size 2□ + 1 is constructed centred on the pixel. The number of points X_ℓ that fall inside this cube is counted for various values of □. The probability, C(□), that at least one point lies within the cube can then be obtained by dividing the number of points by the cube volume:

where N is the number of pixels in the image (if we are considering the whole image or the number of pixels in a moving window - for the purpose of segmentation) and

C(□) conforms to the equation

where D_C is the correlation dimension. Here, a single value of D_C can be computed by the normal bi-logarithmic least squares fit, or we may compute $C(0) = c$ and then compute a value of C(□) for □ = 1, 2, …. In the latter case, we obtain the correlation signature as a plot of D_C against □. For an image, the values of D_C range from two to three. A highly correlated surface gives a correlation dimension close to two whereas a highly uncorrelated surface gives a value close to three.

> Read full chapter

Gradient Vector Flow Deformable Models

Chenyang Xu, … Jerry L. Prince, in Handbook of Medical Image Processing and Analysis (Second Edition), 2009

10.4.2 Results on Gray-level Images

The underlying formulation of GVF is valid for gray-level images as well as binary images. To compute GVF for gray-level images, the edge-map function $f(x,y)$ must first be calculated. Two possibilities are $f_{(1)}(x,y) = |$□$I(x,y)|$ or $f_{(2)}(x,y) = |$□$(G_{□}(x,y) * I(x,y))|$, where the latter is more robust in the presence of noise. Other more complicated noise-removal techniques such as median filtering, morphological filtering, and anisotropic diffusion could also be used to improve the underlying edge map. Given an edge-map function and an approximation to its gradient, GVF is computed in the usual way as in the binary case.

Figure 10.5a shows a gray-level image of the U-shaped object corrupted by additive white Gaussian noise; the signal-to-noise ratio is 6 dB. Figure 10.5b shows an edge-map computed using $f(x,y) = f_{(2)}(x,y)$ with □ = 1.5 pixels, and Figure 10.5c shows the computed GVF field. It is evident that the stronger edge-map gradients are retained while the weaker gradients are smoothed out. Superposed on the original image, Figure 10.5d shows a sequence of GVF deformable contours (plotted in a shade of gray) and the GVF deformable contour result (plotted in white). The result shows an excellent convergence to the boundary, despite the initialization from far away, the image noise, and the boundary concavity.

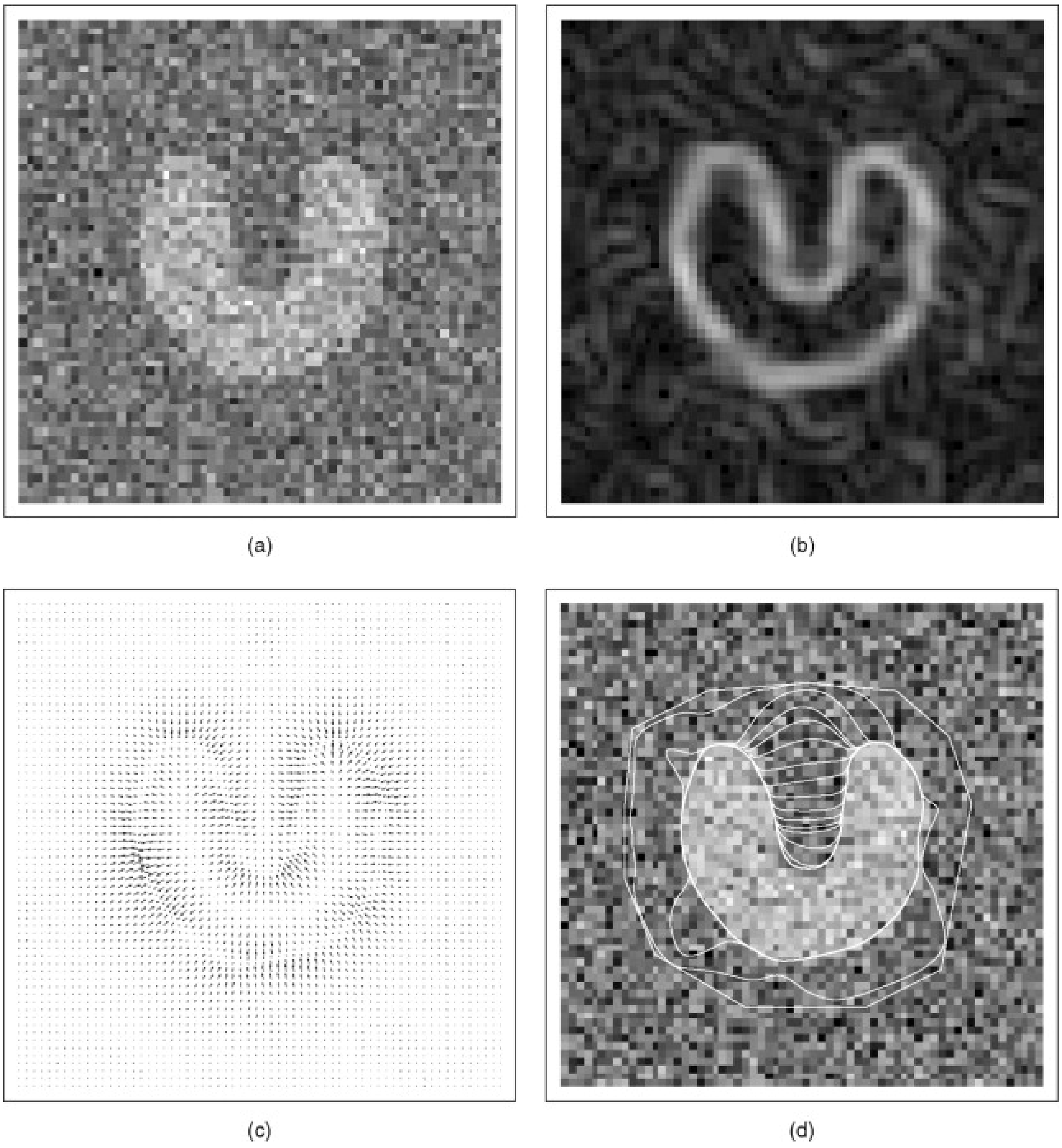

FIGURE 10.5. **(a)** A noisy 64 × 64-pixel image of a U-shaped object; **(b)** The edge map $|□(G_□ * I)|_2$ with □ = 1.5; **(c)** The GVF external force field; and **(d)** Convergence of the GVF deformable contour.

Another demonstration of GVF applied to gray-scale imagery is shown in Figure 10.6. Figure 10.6a shows a magnetic resonance image (short-axis section) of the left ventrical of a human heart, and Figure 10.6b shows an edge map computed using $f(x,y) = f_{(2)}(x,y)$ with □ = 2.5. Figure 10.6c shows the computed GVF, and Figure 10.6d shows a sequence of GVF deformable contours (plotted in a shade of gray) and the GVF deformable contour result (plotted in white), both overlaid on the original image. Clearly, many details on the endocardial border are captured by the GVF deformable contour result, including the papillary muscles (the bumps that protrude into the cavity).

> Read full chapter

Basic Binary Image Processing

Alan C. Bovik, in The Essential Guide to Image Processing, 2009

Publisher Summary

Binary images arise in a number of ways. Usually they are created from gray level images for simplified processing or for printing. However, certain types of sensors directly deliver a binary image output. Such devices are usually associated with printed, handwritten, or line drawing images, with the input signal being entered by hand on a pressure sensitive tablet, a resistive pad, or a light pen. Usually a binary image is obtained from a gray level image by some process of information abstraction. The advantage of the *B*-fold reduction in the required image storage space is offset by what can be a significant loss of information in the resulting binary image. However, if the process is accomplished with care, then a simple abstraction of information can be obtained that can enhance subsequent processing, analysis, or interpretation of the image.

> Read full chapter

Adaptive and Neural Methods for Image Segmentation

Joydeep Ghosh, in Handbook of Image and Video Processing (Second Edition), 2005

4 Adaptive Multichannel Modeling for Texture-based Segmentation

Image texture provides useful information for segmentation of scenes, classification of surface materials and computation of shape, and is exploited by sophisticated biologic vision systems for image analysis [11]. In 1980, Marcelja observed that highly-oriented simple cell receptive fields in the cortex can be accurately modeled by 1D Gabor functions, which are Gaussian modulated sine wave functions. The Gabor functions play an important role in functional analysis and in physics, since they are the unique functions that satisfy the uncertainty principle, which is a measure of the function's simultaneous localization in space and in frequency. Daugman [12] successfully extended Marcelja's neuronal model to 2D, also extending Gabor's result by showing that the 2D Gabor functions are the unique minimum-uncertainty 2D functions. The implication of this for texture analysis purposes, and perhaps for neuronal processing of textured images, is that highly accurate measurements of textured image spectra can be made on a highly localized spatial basis. This simultaneous localization is important, since then it is possible to accurately identify sudden spatial transitions between texture types, which is important for segmenting images based on texture, and for detecting gradual variations within a textured region.

Based on these observations, a multiple channel Gabor filter bank has been used to segment textured images [11]. Each filter's response is localized in the frequency (u–v) plane. A large set of these channel filters is used to sample the frequency plane densely to ensure that a filter exists that will respond strongly to any dominant texture frequency component. Segmentation can be performed by assigning each pixel the label of the maximally responsive filter centered at that pixel. The success of this technique is quite impressive, given that no use use is made of any sophisticated pattern classification superimposed on the basic segmentation structure. More details on multichannel image segmentation can be found in Chapter 4.7 of this *Handbook*.

Some smoothing of the filter outputs before doing the max operation provides better results on texture segmentation. Further improvements can be achieved by using a cooperative-competitive feedback network called Smoothing, Adaptive Winner-Take-All Network (SAWTA) [11]. This network consists of n layers of cells, with each layer corresponding to one Gabor filter, as shown in Fig. 4. On the presentation of an image, a feedforward network using local receptive fields enables each cell plane to reach an activation level corresponding to the amplitude envelope of the Gabor filter that it represents, as outlined in the preceding paragraphs. Let $m_i(x,y)$, $1 \le i \le n$, be the activation of the cell in the ith layer with retinotopic coordinates (x,y). Initially, the n cell activations at each point (x,y) are set proportional to the amplitude responses of n Gabor filters.

FIGURE 4. The SAWTA network for segmentation of textured images.

To implement the SAWTA mechanism, each cell receives constant inhibition from all other cells in the same column, along with excitatory inputs from neighboring cells in the same row or plane. The synaptic strengths of the excitatory connections exhibit a 2D Gaussian profile centered at (x, y). The network is mathematically characterized by shunting cooperative-competitive dynamics [9] that model on-center off-surround interactions among cells which obey membrane equations. Thus, at each point (x,y), the evolution of the cell in the ith layer is governed by:

(10)

where J_+, J_- are the net excitatory and inhibitory inputs, respectively, and are given by:

(11)

Here, R is the neighboring region of support and f is a sigmoidal transfer function. A sigmoidal transfer function is needed to keep the response bounded between 0 and 1 while still maintaining a monotonically increasing response with the argument.

The convergence of a system described by (10) has been shown for the case when the region of support R consists of the single point (x,y). The network is allowed to run for ten iterations before region assignment is performed by selecting the most responsive filter.

Figures 5 and 6 shows comparative experimental result using the SAWTA network for segmentation. The 256 × 256 gray level images are prefiltered using Laplacian-of-Gaussian filters[2] to remove high dc components, low-frequency illumination effects, and to suppress aliasing. Then, only sixteen circularly-symmetric Gabor filters are used to detect narrow-band components as follows: Sets of three filters with center frequencies increasing in geometric progression (ratio = 2:1) are arranged in a daisy-petal configuration along 5 orientations, while the sixteenth filter is centered at the origin. Figure 5 shows the segmentation achieved for a synthetic texture using three different techniques. 5(a) is the original image; 5(b) is the result of the original multichannel segmentation model [13], 5(c) the results of this model with output smoothing, and 5(d) segmentation after 10 iterations of the SAWTA network. The constants A, *B* and C in (10) were taken to be 1, 0, and 10, respectively. The activation function used is $f(x) = \tanh(2x)$. The results are seen to be superior to that obtained by the original multichannel based segmentation scheme.

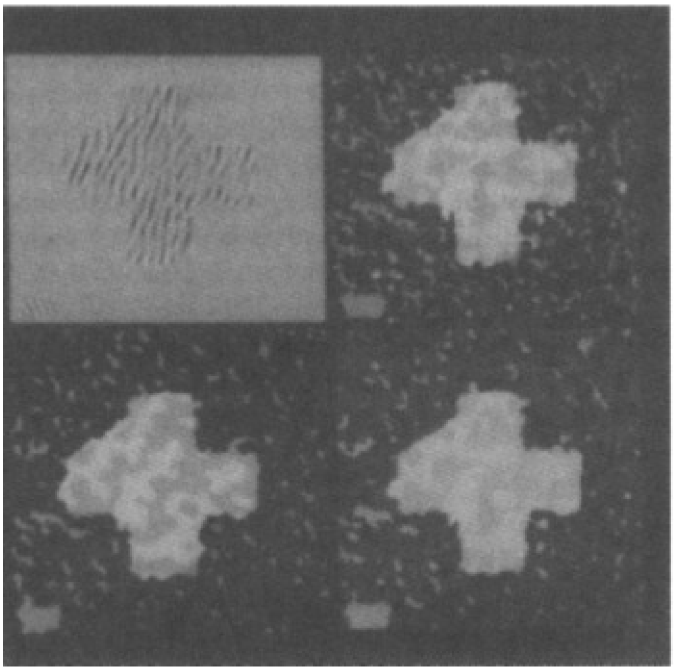

FIGURE 5. Segmentation of a synthetic texture using the SAWTA network (clockwise from top left): (a) Original Image; (b) result of the original multichannel segmentation model [13]; (c) the results of this model with output smoothing; (d) segmentation after 10 iterations of the SAWTA network.

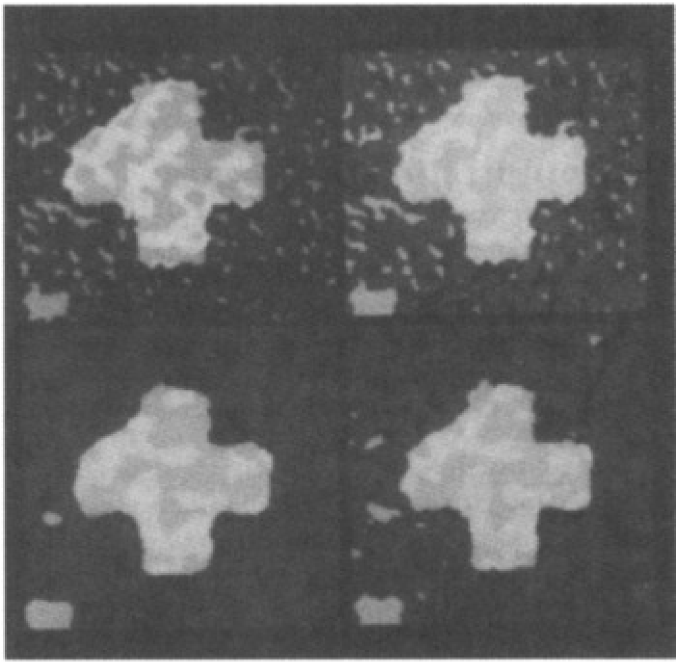

FIGURE 6. Effect of iteration steps and inhibition factor on segmentation: (a) same as Fig. 3(d); (b) segmentation with C = 3; (c) after 100 iterations; (d) after 100 iterations.

Figure 6 shows the effect of varying the number of iteration steps, and the inhibition factor C, on the segmentation obtained. We observe that the SAWTA network achieves a more smooth segmentation in regions where the texture shows small localized variations, while preserving the boundaries between drastically different textures. Usually, 10 iterations suffice to demarcate the segment boundaries, and any changes after that are confined to arbitration among neighboring filters.

The SAWTA network does not require a feature extraction stage as in [14] or computationally expensive masking fields. The incremental and adaptive nature of the SAWTA network enables it to avoid making early decisions about texture boundaries. The dynamics of each cell is affected by the image characteristics in its neighborhood as well by the formation of more global hypotheses. It has been observed that usually four spatial frequencies are dominant at any given time in the human visual system. This suggests the use of a mechanism for post-inhibitory response that suppress cells with activation below a threshold and speeds up the convergence of a SAWTA network. The SAWTA network can be easily extended to allow for multiple “winners”. Then, it can cater to multicomponent textures, since a region that contains two predominant frequencies of comparable amplitude will not be segmented but rather viewed as a whole.

Learning and adaptation is also useful in the multichannel image model for determining the channels (filters) themselves. Indeed, the results of Gabor filtering can be obtained in an iterative fashion, by performing stochastic gradient descent on a suitable cost function. While these filters are useful for a large variety of images, one may wonder whether more customized filters may yield better results for a specific class of images, such as images of barcodes, or MRI scans for the brain. This leads to the concept of “texture discrimination masks”, which may be learned in order to improve performance in the subsequent classification task [15].

First, note that the multichannel framework does not restrict one to using Gabor filters. Other filters reported include Laplacians of Gaussians, wavelets, and general IIR and FIR filters. Each filter can be considered as a localized feature detector, and after performing spatial smoothing if needed, the filter outputs for each pixel can serve as inputs to a multilayered feedforward network such as the multilayered perceptron (MLP) that performs the desired classification task. Thus we effectively have an MLP classifier with an additional hidden layer, i.e., the layer whose inputs are the pixel values in a small image window, and whose afferent weights represent the mask coefficients. While training this network, the filter weights get modified to better perform texture classification. Moreover, by applying node pruning techniques, less important filters can be eliminated. Thus, instead of the usual large set of generic filters, a smaller set of task-specific filters is evolved. Details of this method, along with superior results obtained on page layout segmentation and bar-code localization, can be found in [15]. It is speculated that the efficacy of the learned masks stems from their ability to combine different frequency and directionality responses in the same masks, so that high discrimination information can be captured by a smaller number of filters. On the other hand, if the problem domain changes substantially, a new set of filters needs to be learned for the new set of images.

> Read full chapter