International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 18, No. 4 (2004) 583–606
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

ON THE USE OF A VISUAL CORTICAL SUB-BAND MODEL FOR INTERACTIVE HEURISTIC EDGE DETECTION

DOUGLAS A. LYON
Computer Engineering Department, Fairfield University,
Fairfield, CT, 06430, USA
lyon@docjava.com

We present a novel interactive edge detection algorithm that combines A* search with low-level adaptive image processing. The algorithm models the semantically driven interpretation that we hypothesize to occur between the mind and visual cortex in the human brain. The basic idea is that oriented Gabor sub-bands are used to model grating cells in the mammalian visual system. These sub-bands are used during the search for a path to a marker in an image. A domain expert uses image markers to select edges of interest.

We demonstrate the system in several image domains. Examples are shown in the areas of photo-interpretation, medical imaging, path planning and general edge finding. The A* search finds a suboptimal result, but executes in a time that is typically 10 to 1,000 times faster than the dynamic programming approach currently used for this type of edge detection.

Keywords: Edge detection; A* search; heuristic search; vision; Gabor functions.

1. Introduction

There are many edge detectors that are formulated based on the change in pixel intensity power. These edge detectors can locate the so-called *strong* edges with ease. However, it is often the case that the strong edge is the wrong edge. Figure 1 shows an image that was subjected to a Gabor filter and a threshold. It is clear, in Fig. 1, that the shadow cast by the edges of the window provides a good edge for the Gabor edge detector. It is also clear that this is not a good edge from the architect's point of view. The architect is interested in the edges of the structure, and not the edge represented by a shadow. How do we represent this knowledge to an edge detector without making a domain specific edge detector (i.e. one that only works on buildings?). The experts seem to know a good edge when they see it. We present an algorithm that enables domain experts to describe, to the computer, where the edge of interest is. Then we use A* to locate the interesting edges.

A contribution described in this paper is to apply the techniques of Mortensen[23] to Martelli's algorithm in order to obtain a speed-up. The Mortensen and Barett algorithm runs in over a minute. The algorithm presented in this paper can run

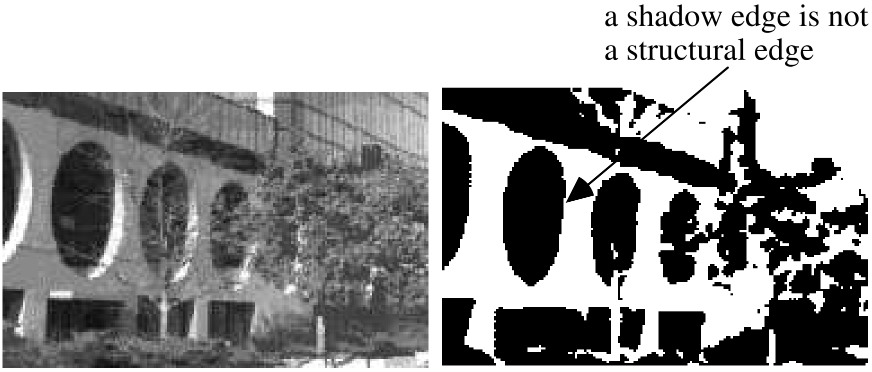

Fig. 1. Strong edges are often wrong edges.

in a few seconds or in as little as a small fraction of a second. Mortensen and Barett's dynamic programming approach has the run-time of Dijkstra's algorithm for finding the shortest path from a start node to all other nodes. Our experiments show execution times for the algorithm to be between 10 and 100 times faster than the Mortensen and Barett algorithm.

A further contribution of this paper is to use steerable, phase modulation, multiscale Gabor filters to further inform the heuristic function used for search. A comparison is made between our edge detection algorithm and three others.

1.1. *Problem statement*

We are given an image and an expert in that image domain. The algorithm tunes a bank of filters to enhance the edges in the image. The expert identifies the pixels on an edge of interest by laying down markers on the edge. We wish to find a good edge that connects those markers. We are subjected to the constraint that the algorithm run in *interactive* time.

The edge represents a series of pixels (some of which may be disconnected), but we want the search to be able to jump gaps during the linking stage. Once tuning of the filter is complete, the search is able to run quickly.

In a color picture containing inhomogeneous (i.e. textured) objects, an edge is the boundary between two regions whose differences are not always easy to quantify (i.e. it is not just a different color). For example, if we look for step edges we will be unlikely to find them. Edges are typically degraded with noise, blurring and suface irregularities. Thus, we assume that the edges of interest are not ideal edges. We also assume that the nature of the differences between different regions is not known *a priori*. If these differences are known then the presented algorithm is unsuitable, as there are provably optimal (and faster) edge detectors for such cases.[6]

In summary, the goal is to find long, continuous edges, that are deemed important by a human operator.

1.2. *Motivation*

We are motivated to solve this class of problems because experts are facile at defining good edges. Thus, it is logical to involve them in the use of the edge detector. We also find that special training of humans is needed before they are considered experts. For example, photo reconnaissance, medical imaging and inspection give rise to applications where an expert is useful. Representing expert knowledge in an edge detector is not easy. We now have a means to represent knowledge about an image domain, edge detector performance, and the location of edges considered to be important to a domain expert. In fact, the quality of the edges produced by the algorithm is considered high only if the edges found agree with the edges the domain expert considered important.

Long, continuous edges, important to a human operator, can segment an image for a special task (i.e. for an anatomical study or surgical planning).[10] In fact this approach has already been shown to be useful in medical image segmentation.[39]

We have seen excellent results from the dynamic programming approach to finding edges, but we find that the speed of the search is an impediment to interactivity. In fact, Mortensen and Barett showed that their computations can often take more than a minute on a complex search.[23] We are interested in looking for an algorithm that can run at a more interactive speed (i.e. less than a few seconds).

1.3. *Approach*

Our approach to solving this problem is to tune a filter-bank so that it is able to resonate on edges. Once this is tuned, we direct the expert to mark those pixels that lie on an interesting edge. We use these markers, and the filter-bank, to help in formulating a heuristic cost function that enables a graph search for a low-cost path between markers.

The sub-bands and markers better inform the heuristic function. The sub-bands in the filters are low-level edge detectors that the heuristic function interacts with, dynamically selecting from the sub-bands in a manner that we think may be consistent with what the visual mind does using low-level geniculate and cortical cells to build a saliency map.

The modeling of orientation cells in the brain is not new.[24] Neibur, Itti and Koch described the filtering of images to produce feature maps for color, intensity and orientation at different scales. Linear combinations of these scales are used as input into the *salancy map*. High-level processing uses a winner take-all network built out of neurons that have later inhibition.

The model is developed in accordance with the known anatomy and physiology of the visual system of the macaque monkey. It comprises two interacting stages: a fast and parallel preattentive extraction of visual features across eight spatial scales using the Laplacian of Gaussian filter. A Winner-Take-All approach is then used to select the scale for each point that offers the highest energy.

The Winner-Take-All approach is inspired by the current thinking in saliency models for selecting the most conspicuous image location, and an inhibition-of-return mechanism to generate attention shifts. It has long been known that the primate visual system selects visual information by saccadic eye movement and mechanisms of visual attention. Exactly how the primate visual system decides on where to look is still an open question.[28]

Normally, a saliency map integrates low-level inputs and codes for the conspicuity of various parts of the visual field. The winner-take all array selects for the correct scale, but only locally. The salancy map describes where objects are, but not what they are.[24] Eye movement is overt evidence of visual attention. Studies show that fixation locations are affected by complex scenes. Thus, the role of the salancy map is to guide attention based on visual stimulus.[28] The center surround pattern activity is evidenced, physically, in the brain by Local Field Potential (LFP) measured directly in synapses by Usher, Stemmler and Niebut.[41]

Thus, the salancy map is a trainable system that does both task dependent and task independent processing. This has given rise to a feature integration theory that makes use of Gabor pyramids that use a scale range from 0..8 (sigma), an approach based on the work of Itti.[12]

Gabor filters have been shown to have optimal localization properties in the spatial and frequency domain. They also approximate visual cortical cells in mammals. A 2D Gabor filter is a sinusoidal plane of given frequency and orientation modulated by a Gaussian envelope.[13]

We have experimented with a 45° orientation filter and found the angle changes to be too coarse. We then show results with a 10° increment between sub-bands in a Gabor filter and find the results improved. Since the human fovea has been shown to be sensitive to better than 1/6 of a degree in the change of the angle of a line,[44] we compute the orientation by measuring the arctangent of the ratio of the energy between the vertical and horizontal directions.

We also allow the domain expert to tune the performance of the heuristic function. The basic trade-off is one of time versus depth of the search. Typically an increased depth is needed when there is increased noise in the image. This takes the algorithm more time in which to compute a good edge. We hypothesize that this may be what the brain does when the mind tries to follow an edge of interest that spans various grating cells in the visual cortex. Eye tracking studies show that saccadic search is performed and this is evidence that there is an interaction between higher brain functions and low-level processing.[38]

The heuristic search selects a minimal-cost path using a merit-ordering function to decide which unexamined node to examine next. We obtain the merit-ordering function from the domain expert who optimizes the function for a class of images, as described in Ref. 16.

The approach of taking an A* search is typically faster than the approach of dynamic programming, and our experiments bear this out.

In summary, we assume that the user is a domain expert. We represent knowledge using the experts' ability to locate a good edge and assist with filter-bank tuning, if necessary.

In the following section we shall present related work. Section 3 describes the implementation of the algorithm, the heuristic formulation, the grating cells, bar cells and their modeling, using the Gabor function. Section 4 describes results obtained using the new algorithm. Section 5 discloses a comparison with other edge detection techniques.

2. Related Work

Alberto Martelli first disclosed the heuristic search of a state-space to find an edge in 1972.[18,19] Mortenson and Barett used dynamic programming and human-entered markers to guide the search in 1995.[23]

Rianto, Kodo and Kim used a Canny filter and a Hough transform in order to find long straight lines. Their application is in the area of detecting roads (but only straight roads). They make use of eight-direction filters, parallel-edge extraction and their knowledge about roads in order to make their edge detector very domain specific. In comparison, our edge detection algorithm can be tuned to work in several domains.[34]

Weldon and Higgins have done work in the area of oriented Gabor filter design for the segmentation of textures. Their work is designed to model the low-level image processing that occurs in the visual cortex and we make use of their computational model to help formulate the cost function.[43] In fact, the local and oriented nature of the V1 simple cell receptive field has been known since the early 1960s.[11] The resolution of the human fovea has been sensitive to a change of 1/6 of a degree in the angle of a line.[44] Miller and Zucker have done an excellent survey of the low-level image processing that occurs in the brain, but the measured connection to heuristic search remains elusive, in the literature.[22]

The use of a steerable-scalable kernel for edge detection is not new.[30] Field suggested neurons with line and edge selectivity form a representation of a scene. Barlow indicated that an unsupervised learning algorithm is used in the brain to extract visual features.[2,8] Some have combined steerable filters with geometric properties of the image structure when searching for an edge.[9,30] Such edge detectors can yield excellent results, at the cost of requiring *ad hoc* implementations for various edge junctions.

Mehrotra, Manuduri and Ranganathan have done work in the area of optimizing the orientation of the Gabor filter for a specific edge orientation.[20] We use their preprocessing ideas to create sub-bands and speed cost function computation.

Olshausen and Field indicated that Gabor-like filters are used for the line and edge selectivity.[27] Our use of the Gabor-like filters as input to a supervised A* search is new. Our results show that computationally strong, but irrelevant edges, can be ignored by such a search.

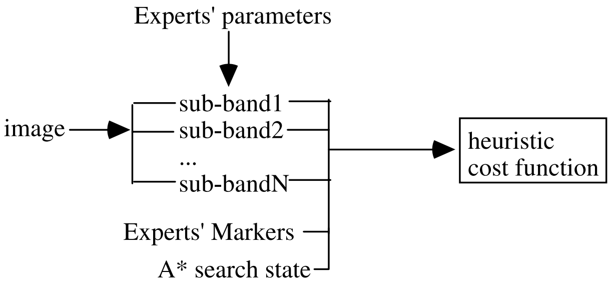

Fig. 2. A multichannel scheme for computing a cost function.

3. Implementation

The A* algorithm searches a graph representation of the image using a heuristic function that returns a scalar value for each position in the graph. The following sections describe the heuristic formulation, grating cells in the visual cortex of the brain and how the heuristic automatically uses the Gabor sub-bands.

Figure 2 depicts the multichannel sub-band scheme for the computation of the heuristic cost function. Sub-bands, the experts' markers and the state of the A* search, are all used to help compute the cost function. Search parameters include the termination criteria, ply of the search, and greediness (i.e. distance to a marker). Using the Laplacian of Gaussian formula, we are able to provide an estimation for sigma for each pixel. The Winner-Takes-All strategy gives rise to the last image (called *Max*), depicted in Fig. 3. These sub-bands are computed using 19 × 19 convolution matrices.

Laplacian of the Gaussian (LoG) is given by

$$-\nabla^2 \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} . \tag{1}$$

Figure 4 shows a plot of (1) in the range of $-4\ldots 4$ for x and y and with $\sigma = 1$. This is often called the Mexican hat (or LoG) kernel. The maximum of (1) occurs at $x = 0$, $y = 0$ and is given by

$$-\left[\nabla^2 \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}}\right]_{\max} = \frac{1}{\pi\sigma^4} . \tag{2}$$

Thus, by precomputing the scale, we can use a locally greedy way to obtain a filter with a highest energy output for each pixel. Section 3.3 shows how to make use of the scale in the design of a Gabor filter bank.

3.1. *Heuristic formulation*

Heuristic formulation of a search can factor in the ply (i.e. depth of the search), the greed (i.e. distance to the goal), and the termination criteria. Termination criteria can include limits to consumed resources (such as CPU time, or memory) as well as the reaching of a goal state.

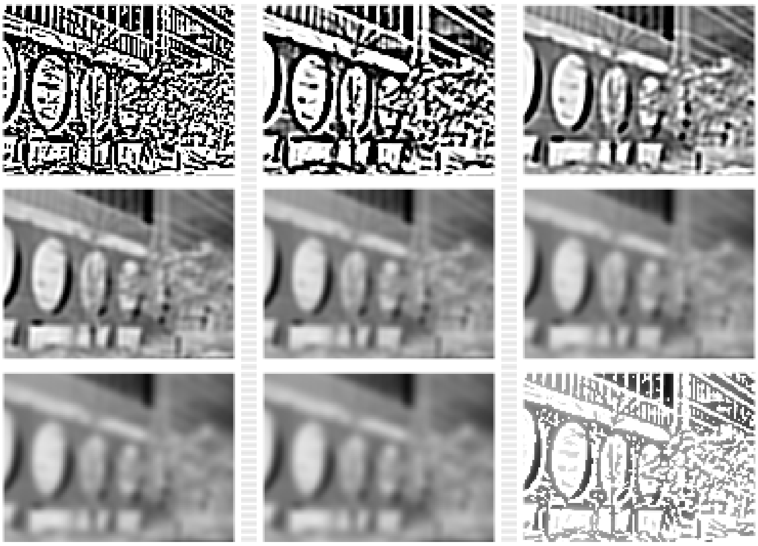

Fig. 3. LoG filters at eight scales. The Max image is shown last.

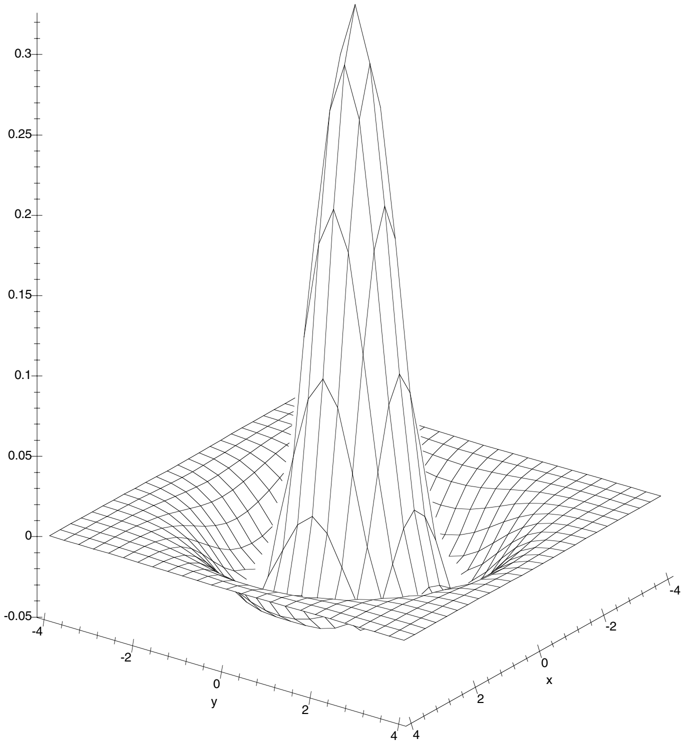

Fig. 4. The Mexican hat kernel.

When searching for a good edge, we use an expert's marker as a start-point or an end-point (i.e. goal state). The computation of our distance to the marker is used to weigh the heuristic function (i.e. make the function more greedy). Greedy heuristic functions make the search faster, but also yield a less optimal result.

The brain makes use of low-level texture detectors that are sensitive to textures of specific orientation. These texture detectors are called *grating* cells and have been found in locations V1 and V2 of the visual cortex in monkeys.[20] The brain brings to the table a massively parallel system, with 100 million brain cells in V1, 4 million of which are classified as grating cells. An additional 1.6% of V2's brain cells are dedicated for grating recognition. Grating cells are not very sensitive to single bars (i.e. a single edge). They are much more sensitive to multiple, parallel edges with a single orientation. Thus, the grating cell model for edge detection is most suitable when an image has textures.[31] The following section describes the use of low-level grating and bar cells to help inform the heuristic function.

3.2. *Grating and bar cells*

Gabor signals are Gaussian modulated sinusoids that are used to create convolution kernels for making Gabor filters. Gabor filters are nonorthogonal but they are complete for the representation of visual information. Empirical studies of two-dimensional receptive field profiles have shown that Gabor filters closely resemble the profile of receptor cells in the mammalian visual system.[14]

Oriented cells are used in the brains' visual cortex to provide a response that is tuned to textures with a specific orientation. These cells are modeled using a Gabor filter that has a specific orientation, center frequency, bandwidth and phase. A series of kernels are computed from these parameters.

The precomputed sub-bands are selected using a Winner-Take-All strategy and assigned to each pixel.

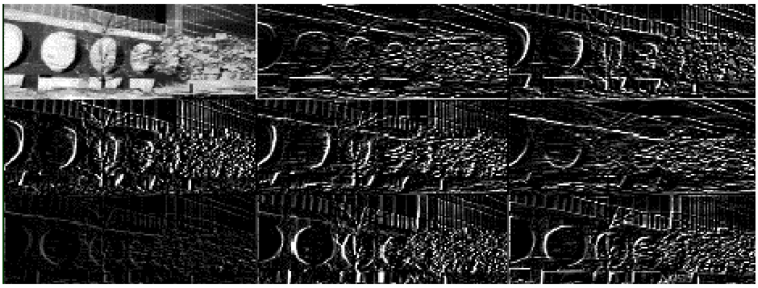

Fig. 5. Robinson 3 × 3 mask at 45° increments.

Using the technique of filtering the input image by a bank of 2D filters tuned to eight compass directions, orientation suffers from low-orientation accuracy. This is due to an overly simplified model of the low-level visual processing in the brain.

Simple cells in the visual cortex have orientation and spatial frequency selectivity for single bars. The research community has focused on the difference between edge detection and local frequency analysis in the cortex. If we use only the bar cells, we can get results that ignore the local frequency analysis that occurs in the cortex.

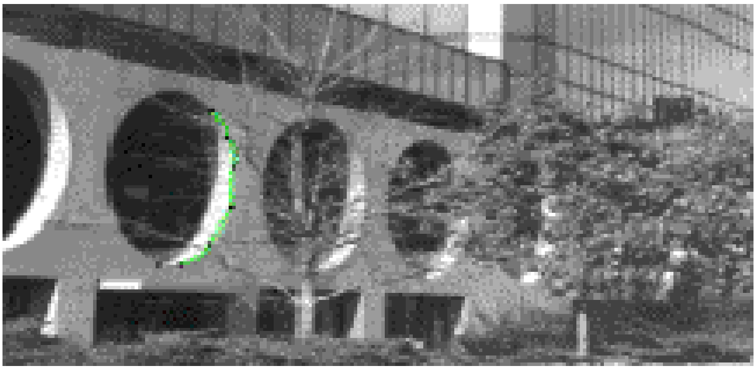

Plate 1. The edges are too straight using Robinson sub-bands.

Plate 1 shows the result of using a low-orientation accuracy filter for the edge following done by the A* search. Also, we would like the ability to use texture information in order to improve our edge detection. As a result, we model the grating cells in the cortex using Gabor functions.

3.3. *Gabor-based heuristics*

The goal of our heuristic formulation is to create a weighting function that is general enough to be domain independent. In order to achieve this generality, we provide a method for adjusting several input parameters. This enables the heuristic to be tuned by the domain expert for a specific class of images.

The input parameters are:

(1) The markers that are placed by the expert into the image.
(2) The sensitivity to the ply of the search.
(3) The greediness of the search.
(4) The sensitivity to the sub-band.

The input parameters are adjusted during a three phase set-up procedure. In phase one, the algorithm adjusts the filter bank to maximize energy output.

In phase 2 of the set-up, markers are placed on the image. The markers lie on the edge of interest. The markers can be used by the heuristic function, both as a goal state and as a means of sub-band selection. Thus, the search interacts with the low-level image processing in order to improve the search. The quality of the search can be measured as the speed of execution and the subjective quality of the edges that it produces.

In phase 3, the expert can adjust search quality by altering the greediness of the search, the sensitivity to the ply and the sub-band values.

Greediness can produce edges that can find a low-cost path, even if there is a noticeable gap in the image.

The 2D Gabor filters model the simple cells in the primary visual cortex of primates. The response is obtained by convolving the input image with a kernel whose equation is given by

$$g(x,y) = \exp\left(-\frac{U^2+\gamma^2V^2}{2\sigma^2}\right)\cos(2\pi U/\lambda + \phi) \tag{3}$$

where

$$\begin{aligned}
U &= (x-u)\cos\theta - (y-v)\sin\theta\\
V &= (x-u)\sin\theta + (y-v)\cos\theta\\
(u,v) &\in \Omega\\
(x,y) &\in \Omega\\
\Omega &= \text{visual field domain}\\
\theta &= \text{orientation} \in [O,\pi]\\
\phi &= \text{phase offset} \in (-\pi,\pi]\\
\sigma &= \text{standard deviation (size of receptive field)}\\
\gamma &= \text{spatial aspect ratio} = 0.5,\ \text{by experiment}\\
\lambda &= \text{wavelength of receptive field function.}\\
\lambda &= \sigma/0.56,\ \text{according to Ref. 15,}
\end{aligned}$$

$$g(x,y) = \exp\left(-\frac{U^2+0.25V^2}{2\sigma^2}\right)\cos(0.56 * 2\pi U/\sigma + \phi)\,. \tag{4}$$

Substituting $\theta = \pi/2$ and $\phi = 0$ into (4), we obtain a kernel that detects edges oriented along the x-axis:

$$g_{\theta=\pi/2}(x,y) = \exp\left(-\frac{(v-y)^2+0.25(x-u)^2}{2\sigma^2}\right)\cos(0.56 * 2\pi(v-y)/\sigma)\,. \tag{5}$$

Similarly, we substitute $\theta = 0$ and $\phi = 0$ into (4) to obtain a kernel that detects edges along the y-axis:

$$g_{\theta=0}(x,y) = \exp\left(-\frac{(x-u)^2+0.25(y-v)^2}{2\sigma^2}\right)\cos(0.56 * 2\pi(x-u)/\sigma)\,. \tag{6}$$

Once the measurements of (5) and (6) are computed, we use the results to find the orientation of the edge via:

$$\theta = \arctan\left(\frac{g_{\theta=\pi/2}(x,y)}{g_{\theta=0}(x,y)}\right). \tag{7}$$

Now that we have the scale and orientation for the Gabor filter we compute the phase of the filtered image using:

$$\phi(x,y) = \arctan\left[g_o(x,y)/g_e(x,y)\right] \tag{8}$$

where

$$g_e(x,y) = \exp\left(-\frac{U^2+0.25V^2}{2\sigma^2}\right)\cos(0.56*2\pi U/\sigma)$$
$$g_o(x,y) = \exp\left(-\frac{U^2+0.25V^2}{2\sigma^2}\right)\sin(0.56*2\pi U/\sigma) \tag{9}$$

(Ref. 17). Substituting (9) into (8) yields:

$$\phi(x,y) = \arctan\left[\frac{\exp(-\frac{U^2+0.25V^2}{2\sigma^2})\sin(0.56*2\pi U/\sigma)}{\exp(-\frac{U^2+0.25V^2}{2\sigma^2})\cos(0.56*2\pi U/\sigma)}\right]. \tag{10}$$

Simplifying, yields:

$$\phi(x,y) = 0.56*2\pi U/\sigma \tag{11}$$

which expands to:

$$\phi(x,y) = \frac{0.56*2\pi}{\sigma}[(x-u)\cos\theta-(y-v)\sin\theta] \tag{12}$$

as the expression for phase at any angle and scale. The computation of the Gabor parameters is repeated, for each point on the image and this is computationally intense. As an alternative, we can fix the phase at zero and search precomputed sub-band for maximal output, as we did in the scale space. But this will not allow us the same phase resolution as the direct computation used in (12).

Figure 6 shows 18 Gabor sub-bands at 10° increments and their application to the 64 × 64 camera-man image. These sub-bands are precomputed from 7 × 7 convolution matrices. Computing these sub-bands with the accuracy of the human

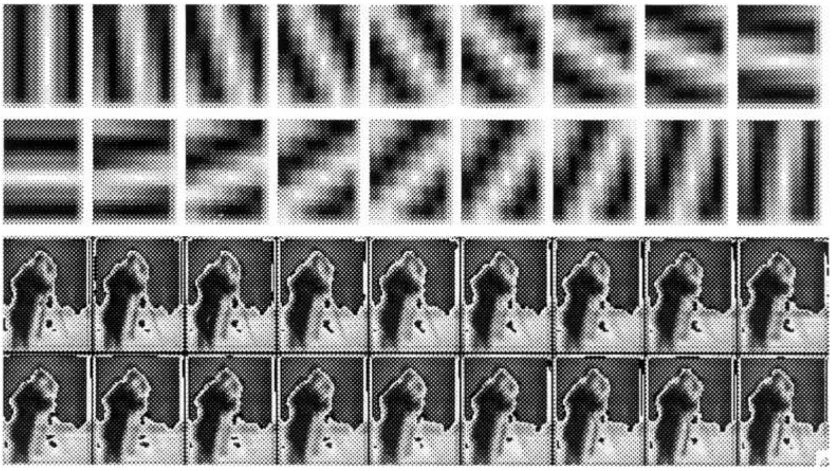

Fig. 6. Gabor sub-bands at 10° increments.

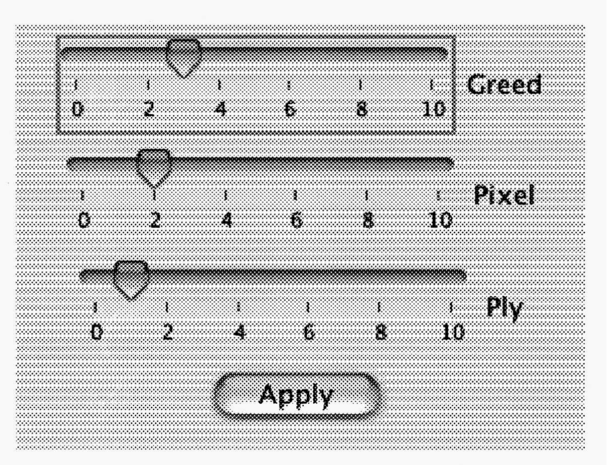

Fig. 7. A* parameter control panel.

visual system (1/6 of a degree) using a range between 0 and 180° would require 1,080 orientation sub-bands for each scale (for 8 scales this is 8,640 sub-bands). Thus, a direct computation of the Gabor parameters saves in the storage of thousands of sub-bands. Only a single filtered output image need to be stored once all the Gabor parameters are known for each pixel.

The cost function is computed by adding the distance of the current point to the next marker (i.e. greed), the depth of the search (i.e. ply) and the pixel value of the Gabor filtered image to favor a min-cost path. The weight for the ply, greed and sub-band contribution are adjusted until a satisfactory balance is achieved between computational performance and quality of result. The adjustment is performed with the GUI shown in Fig. 7.

4. Results

Plate 2 shows a trace that follows a rivers' edge. The edge follows a rough coastline (passing through the experts' markers).

Plate 3 shows an edge selected by a domain expert in an echocardiogram. Areas that are seen as important to a domain expert are often subtle.

Figure 8 shows the echo cardiogram being edge detected with a multiscale Gabor function, without the help of a domain expert. There are many edges shown in Plate 4, none of which corresponds to the experts' edge in Fig. 8.

Figure 9 shows the books after histogram equalization and thresholding.

Figure 10 shows the books after erosion connects the regions too small to provide a passage for the car.

Figure 11 shows the books after the skeletonization step. Using a cooperative target on the car, we are able to locate the car with an overhead vision system. This enables the placement of markers showing a path for the car, as seen in Plate 5.

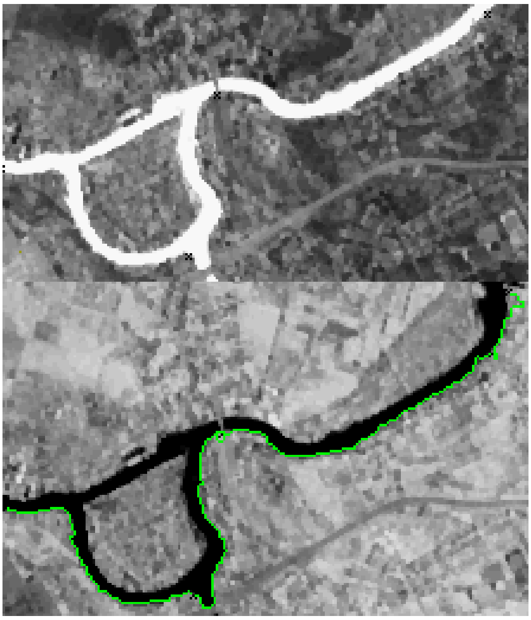

Plate 2. Photointerpretation.

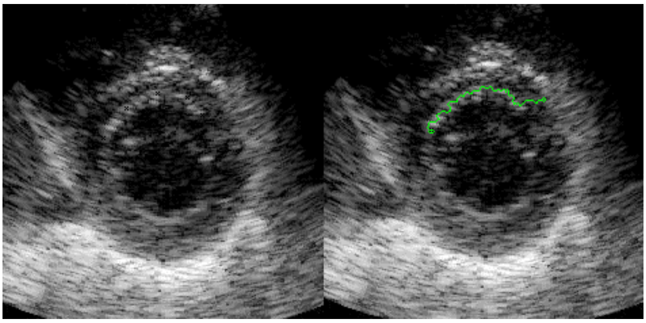

Plate 3. An echocardiogram with experts trace.

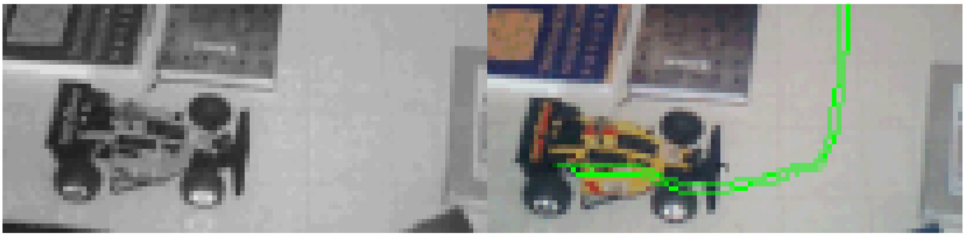

Plate 4. A path planning solution.

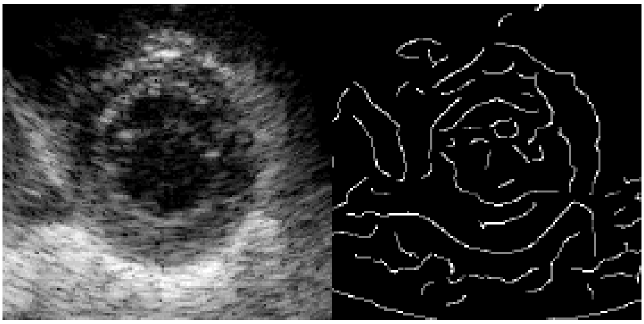

Fig. 8. Multiscale Gabor edge detection.

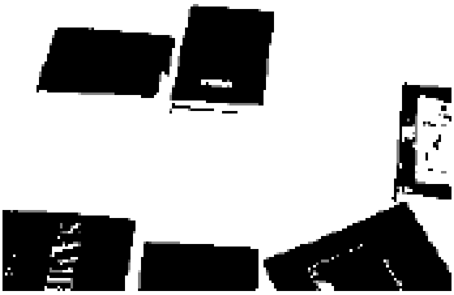

Fig. 9. Histogram equalized and threshold.

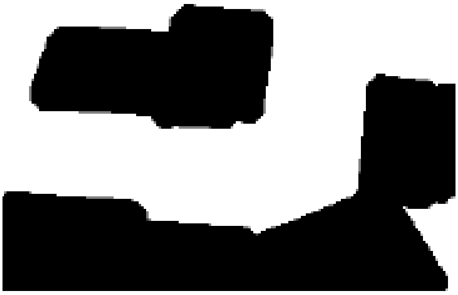

Fig. 10. Erosion.

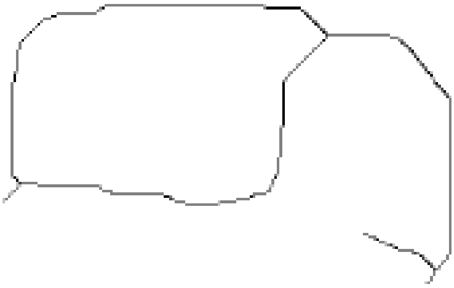

Fig. 11. Skeletonization.

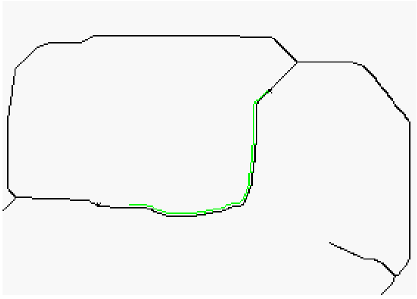

Plate 5. Path computed using A* search.

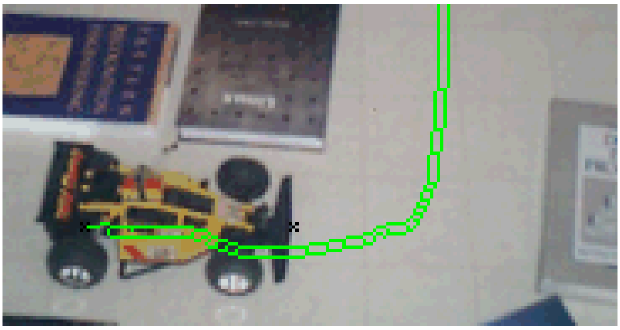

Plate 6. The path superimposed on the car scene.

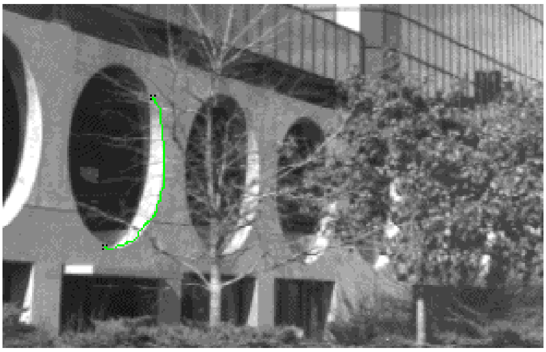

Plate 7. Using Gabor-driven heuristics.

Plate 6 shows the path superimposed on the car scene after low-level image processing and A* search.

Plate 7 shows the result of using the hybrid Robinson–Gabor driven heuristics to derive the edge of interest. The mix ratio between the Robinson and the Gabor costs is 1:1. The question of how to determine the mix ratio is open.

The A* search for Plate 7 ran in 0.07 seconds on a 400 Mhz G4, in Java. In comparison, dynamic programming takes several minutes.[23]

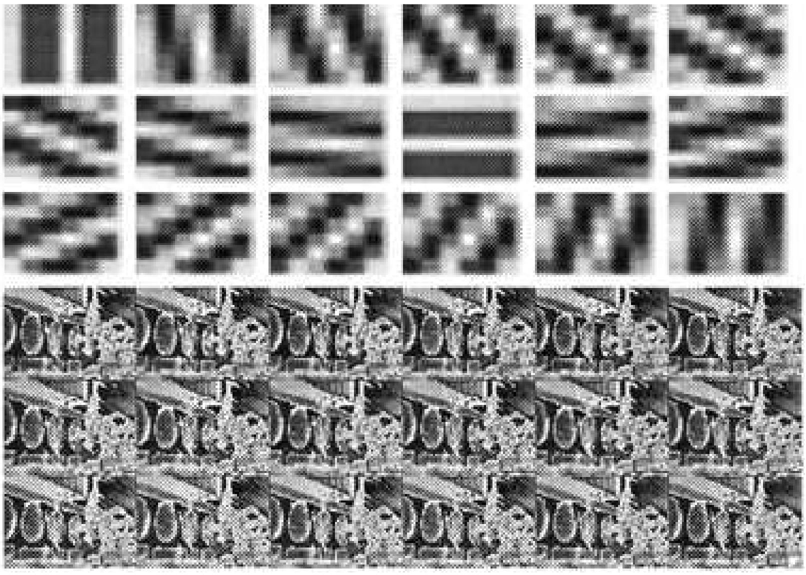

Fig. 12. 18 Gabor sub-bands and the building image.

5. Comparison with Other Edge Detection Techniques

This section discloses an experimental evaluation of the Canny algorithm, the Marr and Hildreth algorithm and the Mehrotra and Zhang algorithm. The rest of this section summarizes the algorithms and their performance on the *building* test image. We will show that the various edge detectors explored will detect edges, but these are not the edges of interest. Specifically, the edge detectors are attracted to shadows because they represent a step edge transition. Our results show that the supervised (i.e. interactive) edge detector has an unfair advantage over the unsupervised edge detector. Unsupervised edge detectors miss true edges and detect false ones (given that a shadow is not a true edge).

Canny's algorithm has combined the goals of accurate edge detection and spatial localization into a single functional. He points out that we can obtain arbitrarily good localization at the expense of detection. He also shows that we can obtain arbitrarily good detection at the expense of localization. The Canny performance criterion for an edge detector requires good detection, good localization and a minimized number of responses to a single edge. Good detection means a low probability of failing to mark a real edge and a high probability of marking a correct edge. Good localization means that the marked edge should be close to the real edge. The Gaussian PDF is also called the *Gaussian density*, *normal probability density function* or just the *normal density*. In 2D, the Gaussian PDF is given by

$$\text{Gaussian}\ (x, y, x_c, y_c, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{((x-x_c)-(y-y_c)^2)}{2\sigma^2}} \tag{13}$$

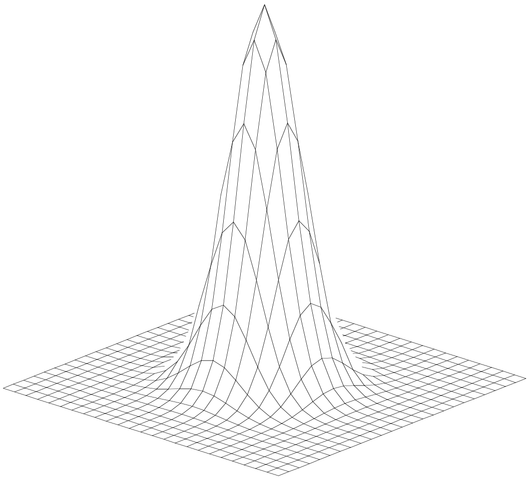

Fig. 13. A Gaussian density.

where σ is the *standard deviation*. The maximum value of (13) occurs at $x = x_c$, $y = y_c$ and is given by:

$$g_{\max} = \frac{1}{2\pi\sigma^2} \, . \tag{14}$$

Figure 13 shows (13) with $x_c = y_c = 0$ and $\sigma = 0$.

Formulation and solution of the optimization criterion in two dimensions is a complex problem.[21] Canny used a criterion of optimality that shows that the Gaussian operator is sub-optimal. But then Canny goes on to use the Gaussian operator because it can be computed with "much less effort".[4] When the criterion of optimality is to arrive at a smoothing filter, with both small variance and limited bandwidth, the optimal solution is the Gaussian distribution. To put it another way, the Gaussian is the only function to minimize the bandwidth-frequency product.

Figure 14 shows the Canny algorithm applied to the building image. We used a 15 × 15 Gaussian preprocessing window with σ ranging from 1 to 8. The Canny algorithm identifies the shadow edge as the strongest edge for most scales. At the lowest of the scales (σ ranging from 1 to 2), the structural edge and the shadow edge are left connected. At higher scales (σ ranging from 3 to 8), the structural edge is seen to be disconnected. In other words, the computationally strongest edge ignores what the domain expert knows about structures. The scale of the Gaussian determines the amount of noise reduction, however, the larger the scale of the Gaussian the less accurate the edge localization.

For our next edge detector comparison we consider the Marr and Hildreth algorithm, based on the Laplacian of the Gaussian filters (LoG). We are motivated to explore the LoG filter because a single convolution can be used to both low-pass filter and edge detect the image. Also, the filter is symmetric, so the only subbands that we need to consider are those that deal with the scale of the filter. This

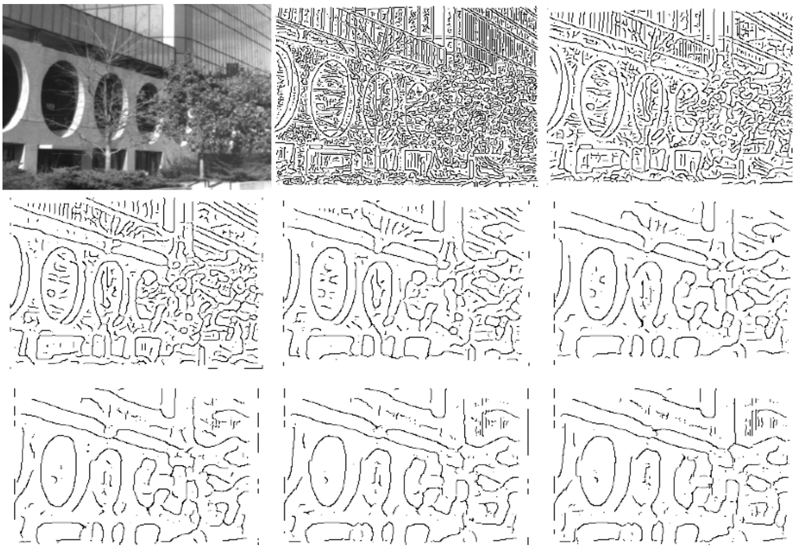

Fig. 14. Canny results.

is sometimes called the Mexican hat function. The proposed algorithm uses the zero crossings of the LoG contours using isotropic derivatives (which reduces the signal to noise ratio of the detector). The symmetric nature of the filter gives no indication of direction (something we use in our algorithm for higher-level processing). The Laplacian of the Gaussian (LoG) is given by (1), (2) and Fig. 15.

We follow the LoG filter with a 3 × 3 Sobel convolution, given by:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \tag{15}$$

Figure 15 shows the LoG filter + Sobel edge detector applied to the building image. We used a 15 × 15 Gaussian preprocessing window with σ ranging from 2 to 9. With σ ranging from 4 to 9, the shadow edge held is much stronger than the structural edge of interest. For σ values at 2 or 3, an examination of the region of interest, as shown in Fig. 16, reveals that the tree limbs in front of the structure make for a strong edge.

While the LoG + Sobel edge detector does give very strong results on the best edges, the criteria for the selection of the best edge is uninfluenced by the human operator. As a result, spurious edges (i.e. tree limbs) are seen as important as structural edges by the LoG + Sobel edge detector.

In 1996, Mehrotra and Zhang presented an optimal approach to the isotropic zero-crossing based edge detection.[21] Like the LoG detector, the rotational invariance provides no information to the high-level detection algorithms, however, it is provably optimal for the detection of 2D step edges.

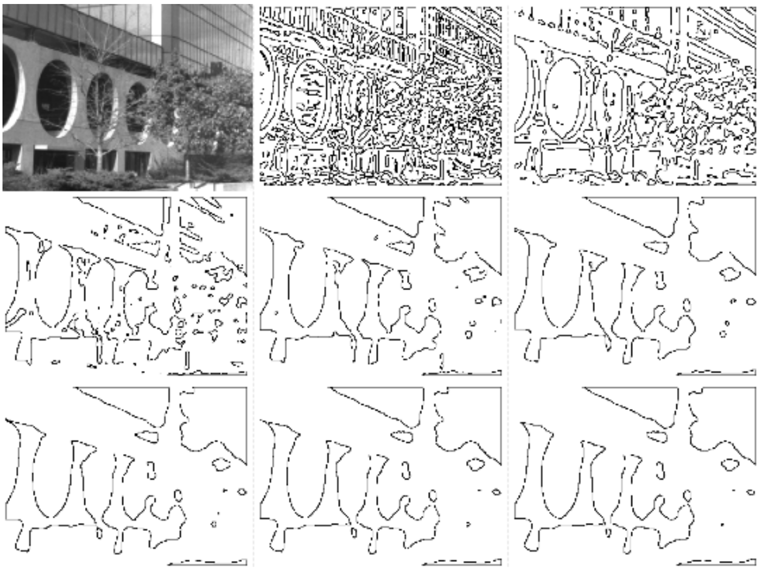

Fig. 15. LoG + Sobel edge detection.

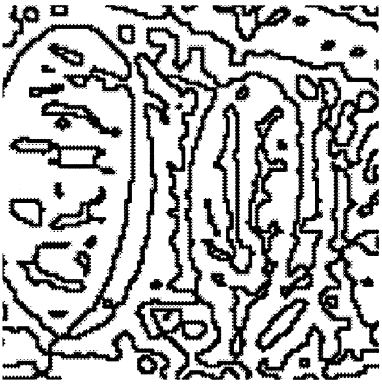

Fig. 16. Structural edges are as strong as tree edges.

The Mehrotra and Zhang detector is interesting because it is optimal, despite its isotropic nature. Anisotropic edge detectors, like Canny's, smear the zero-crossing contours. The detector is given by:

$$\Psi(x,y) = -\frac{100}{9h^2}\left(1 - \left[\frac{x^2+y^2}{h^2}\right]^{3/2} + 7.5\left[\frac{x^2+y^2}{h^2}\right]^{3/2}\ln\sqrt{\frac{x^2+y^2}{h^2}}\right) \tag{16}$$

for $x^2 + y^2 \leq h^2$ and zero otherwise. Figure 17 shows (16) applied with a threshold and the Sobel operator, given in (15). A 15×15 kernel is created from (16) and h is allowed to range from 1 to 8. The results clearly show that the shadow edge

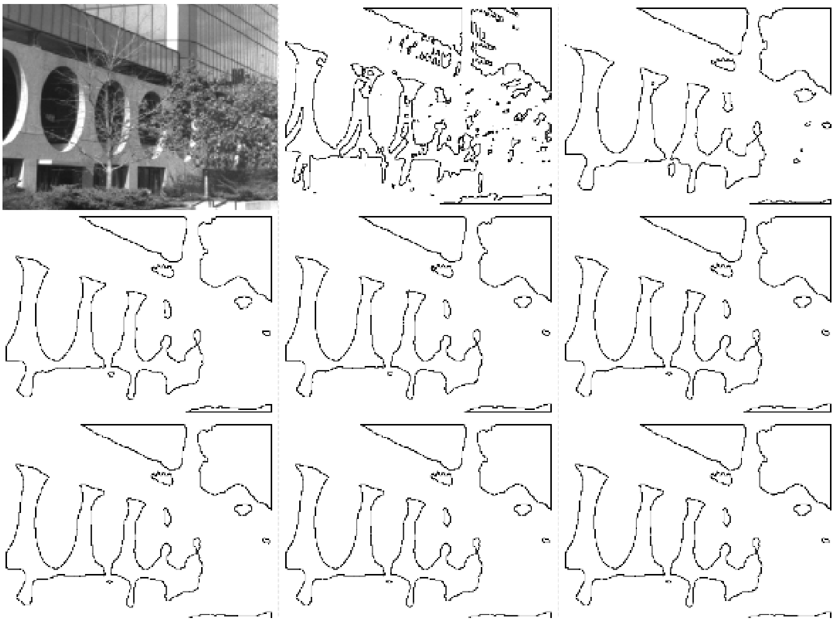

Fig. 17. Optimal 2D edge detector.

is stronger than the structural edge in the building image. Thus, even the optimal edge detector is not able to identify the edge favored by the human operator without some assistance.

Our experiments investigated the use of directional operators only in the Canny algorithm. At increased scales the various algorithms round junctions and reduce noise and the accuracy of the localization of the edge.

In comparison, our algorithm uses a single scale (selected by the operator) and a set of directional sub-bands (selected by the algorithm) to help guide the higher-level search for a best edge. Further, we allow the operator to input points that lie on the important edge and thus create a better informed heuristic function.

6. Summary

We present an attended search technique that uses a cortical model to derive a sub-band driven cost function for heuristic edge detection. The edges produced are good, but at the cost of operator intervention and nondeterministic run-time.

Low-noise, high-contrast images are easier for the algorithm than high-noise, low-contrast images. The more time given to the search, the better the edges. Without tuning, the algorithm performance can be poor.

The users who guide such a search for a good edge are generally good at identifying an important edge. This type of interaction affords an area of flexibility not typically seen in the unattended search techniques. Further, the more the user helps to guide the search, the faster the search can be. This can help keep the search engine from finding local-minima based solutions.

The $\mathbf{A}^*$ search uses a search space that is exponential in path length. If we eliminate the summation of the parent node into the cost, we have a heuristic depth-first search. This is locally minimal in the evaluation function. The heuristic depth-first search is a much faster search than $\mathbf{A}^*$ and is excellent for finding obvious paths in low-noise images.[32]

We have used experimental evaluation of edge detectors to distinguish detectors based on an evaluation criteria. Our informal (and subjective) criteria allows humans to judge what edges are important, based on their experience with the context (i.e. the scene) and the *perceived* edges. This vague citeria is used to establish the failure of unsupervised edge detectors to correctly identify the perceived edges. A more objective evaluation is desirable. Pratt has established a *figure of merit* which combines the factors of nondetection of true edges, detection of false edges and edge delocalization.[33] The unsupervised edge detector will not detect the true edge, and detect shadows (i.e. false edges), thus the unsupervised edge detectors fail on at least the first two of the three objective criteria in the figure of merit.

In the future, the two-dimensional edge detection should be extended to output edges across a sequence of images. This requires a reformulation of the heuristic, and is a topic for future research.

The heuristic cost function also requires further study. In fact, all formulations for heuristics considered, to date, have been *ad hoc*. There is no theory, that we know of, for the design of optimal heuristics. The question of what the mix ratio is between the bar cells and the grating cells in the cortex remains open.

Several implementation issues should also be addressed. The examples shown in this paper do not run in minimal time. The heuristic search looks for an open node with a minimal cost. The search for a minimal-cost node runs in a time that is $O(N)$. One way to speed up the search for a minimal cost node is to use a priority queue. Such a system can delete the minimum node in $O(\log N)$ time. Thus, a reimplementation of the search using a priority queue should provide a large speed-up.[40] This suggestion is similar to the speed-up Ashkar–Modestino algorithm first suggested for this type of problem by Sankar[36] and Ashkar.[1]

Another topic for future research is *vectorization*. Since the search algorithm naturally generates connected components, the vectors that arise from following the good edges represent the linear features in the image. These basic graphic primitives can help in image segmentation.[7] Since the vectors are ordered, they can reduce the pen-up time when they are plotted using a pen plotter. This is important for vector output devices that have a very limited number of vectors that can be drawn before flicker is perceived (i.e. laser displays). Ordering the vectors to minimize the pen-up time is a problem that is known to be NP-complete and is called the *Chinese Postman Problem*.[35]

A more efficient technique for computing the Gabor functions was disclosed in Ref. 42. It would be interesting to try implementing this in order to examine the performance difference.

Full source code (in Java) is available for the code described in this paper at <http://www.docjava.com>.

Acknowledgments

The author thanks Dr. Carl Weiman for numerous useful discussions during the development of this work.

References

1. A. P. Ashkar and J. W. Modestino, The contour extraction problem with biomedical applications, *Comput. Graph. Imag. Process.* **7** (1978) 331–355.
2. H. B. Barlow, What is the computational goal of the neocortex? in *Large-scale Neuronal Theories of the Brain*, ed. C. Koch (MIT Press, Cambridge, MA, 1994).
3. A. Barr and E. A. Feigenbaum, *The Handbook of Artificial Intelligence* (Addison Wesley, NY, 1981).
4. J. F. Canny, *Finding Edges and Lines in Images*, Technical Report No. 720, AI-TR-720, MIT Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139, USA, 1983.
5. W. F. Clocksin and C. S. Mellish, *Programming in Prolog* (Springer-Verlag, NY, 1981).
6. L. S. Davis, A survey of edge detection technique, *Comput. Graph. Imag. Process.* **4** (1975) 349–376.
7. D. Doerman, An introduction to vectorization and segmentation, *Graphics Recognition*; *Algorithms and Systems*, Second International Workshop, GREC 1997, Nancy, France, pp. 1–8.
8. D. J. Field, What is the goal of sensory coding? *Neural Comput.* **6** (1994) 559–601.
9. W. Freeman, Steerable filters and the local analysis of image structure, PhD thesis, MIT Department of Media Arts and Sciences, September 1992. <http://citeseer.nj.nec.com/freeman92steerable.html>.
10. B. Geiger, Three-dimensional modeling of human organs and its application to diagnosis and surgical planning, Technical Report 2105, Institut National de Recherche en Informatique et Automatique, December 1993. <http://citeseer.nj.nec.com/9917.html>.
11. D. H. Hubel and T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, *J. Physiol.* **195** (1968) 215–244.
12. L. Itti, E. Niebur and C. Koch, A model of saliency-based fast visual attention for rapid scene analysis, *IEEE Trans. Patt. Anal. Mach. Intell.* **20**(11) (1998) 1254–1259.
13. A. Jain and S. Bhattacharjee, Address block location on envelopes using Gabor filters, *Patt. Recogn.* **25**(12) (1992). <http://citeseer.nj.nec.com/jain92address.html>.
14. J. P. Jones and L. A. Palmer, An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex, *J. Neurophysiol.* **58** (1987) 1233–1258.
15. P. Kruizinga and N. Petkov, Non-linear operator for oriented texture, *IEEE Trans. Imag. Process.* **8**(10) (1999) 1395–1407. <http://citeseer.nj.nec.com/kruizinga99nonlinear.html>.
16. D. Lyon, *Image Processing in Java* (Prentice Hall, Englewood Cliffs, NJ, 1999).
17. W. Y. Ma and B. S. Manjunath, Edge flow: a framework of boundary detection and image segmentation, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997. <http://citeseer.nj.nec.com/ma97edge.html>.

18. A. Martelli, Edge detection using heuristic search methods, *CGIP* **1**(2) (1972) 169–182.
19. A. Martelli, An application of heuristic search methods to edge and contour detection, *CACM* **19** (1976) 73–83.
20. R. Mehrotra, K. R. Namuduri and N. Ranganathan, Gabor filter-based edge detection, *Patt. Recogn.* **25** (1992) 1479–1494.
21. R. Mehrotra and S. Zhang, A computational approach to zero-crossing-based two-dimensional edge detection, *Graph. Mod. Imag. Process.* **58** (1996) 1–17.
22. D. A. Miller and S. W. Zucker, Computing with self-excitatory cliques: a model and an application to hyperacuity-scale computation in visual cortex, *Neural Comput.* **11** (1999) 21–66.
23. E. Mortensen and W. Barett, Intelligent scissors for image composition, in *Computer Graphics Proc.*, SIGGRAPH 1995, pp. 191–198.
24. E. Niebur and C. Koch, Control of selective visual attention: modeling the \where″ pathway, in *Advances in Neural Information Processing Systems*, eds. D. S. Touretzky, M. C. Mozer and M. E. Hasselmo, Vol. 8 (MIT Press, Cambridge, MA, 1996), pp. 802–808.
25. E. Niebur, L. Itti and C. Koch, Controlling the focus of visual selective attention, in *Models of Neural Networks IV*: *Early Vision Attention*, eds. J. L. Van Hemmen, J. D. Cowan and E. Domany (Springer Verlag, NY, 2001), pp. 247–276.
26. N. J. Nilssen, *Principles of Artificial Intelligence* (Tioga Publishing Company, Palo Alto, CA, 1980).
27. B. A. Olshausen and D. J. Fields, Natural image statistics and efficient coding, *Network*: *Comput. Neural Syst.* **7** (1996) 333.
28. D. Parkhurst and E. Niebur, Scene content selected by active vision, *Spatial Vision* **16**(2) (2003) 125–154.
29. D. Parkhurst, K. Law and E. Niebur, Modelling the role of salience in the allocation of visual selective attention, *Vis. Res.* **42**(1) (2002) 107–123.
30. P. Perona, Steerable-scalable kernels for edge detection and junction analysis, in *Proc. 2nd Eur. Conf. Computer Vision*, ed. G. Sandini (Springer-Verlag, Italy, 1992), pp. 3–18. <http://citeseer.nj.nec.com/perona92steerablescalable.html>.
31. N. Petkov and P. Kruizinga, Computational models of vision neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells, *Biol. Cybern.* **76** (1997) 83–96.
<http://citeseer.nj.nec.com/petkov96computational.html>.
32. D. Poole, A. Mackworth and R. Goebel, *Computational Intelligence*, *a Logical Approach* (Oxford, 1998).
33. W. K. Pratt, *Digital Image Processing* (Wiley-Interscience Publication, 1978).
34. Y. Rianto *et al.*, Detection of roads from satellite images using optimal search, *Int. J. Pattern Recognition and Artificial Intelligence* **14** (2000) 1009–1023.
35. F. Roberts, *Applied Combinatorics* (Prentice Hall, Englewood Cliffs, NJ, 1984).
36. P. V. Sankar and J. Sklansky, A gestalt-guided heuristic boundary follower for X-ray images of lung nodules, *IEEE PAMI* **4** (1982) 326–331.
37. Y. Shirai, *Three-Dimensional Computer Vision* (Springer-Verlag, NY, 1987).
38. F. Smeraldi, A. Makarov and J. Bigun, Saccadic search with Gabor features applied to eye detection, Technical Report 98/256, Swiss Federal Institute of Technology, Computer Science Department, CH-1015 Lausanne, January 1998.
<ftp://lamiftp.epfl.ch/pub/smeraldi/gaboreye.ps.gz>,
<http://citeseer.nj.nec.com/article/smeraldi98saccadic.html>.
39. D. Stalling and H.-C. Hege, Intelligent scissors for medical image segmentation,

in *Proc. 4th Freiburger Workshop Digitale Bildverarbeitung in der Medizin*, eds. B. Arnolds, H. Muller, D. Saupe and T. Tolxdorff, Freiburg, March 1996, pp. 32–36.

40. T. A. Standish, *Data Structures in Java* (Addison Wesley, NY, 1998).
41. M. Usher, M. Stemmler and E. Niebur, The role of lateral connections in visual cortex: dynamics and information processing, *Lateral Interactions in the Cortex*, eds. J. Sirosh, R. Mikkulainen and Y. Choe, The UTSC Neural Networks Research Group, Austin, Texas. Electronic book, ISBN 0-9647060-0-8. <http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook9>.
42. C. Weiman, Efficient discrete Gabor functions for robot vision, SPIE, *Conf. Wavelet Applications*, Vol. 2242, Orlando, FLA, April 1994, pp. 148–160.
43. T. P. Weldon, W. E. Higgins and D. F. Dunn, Gabor filter design for multiple texture segmentation, *Opt. Eng. SPIE* **35** (1996) 2852–2863. <http://citeseer.nj.nec.com/weldon96design.html>.
44. G. Westheimer, The grain of visual space, *Cold Spring Harbor Symp. Quantitative Biology*, Vol. LV, pp. 759–763.

Douglas A. Lyon after receiving his Ph.D. from Rensselaer Polytechnic Institute, worked at AT&T Bell Laboratories. He has also worked for the Jet Propulsion Laboratory at the California Institute of Technology. He is currently the Chairman of the Computer Engineering Department at Fairfield University, a senior member of the IEEE and President of DocJava, Inc., a consulting firm in Connecticut.