Detecting curvilinear structure in images

Ziv Gigus[†] *and* **Jitendra Malik**[‡]

Computer Science Division
University of California at Berkeley, CA 94720.
e-mail: †ziv@miro.berkeley.edu, ‡malik@robotics.berkeley.edu

February 1991, Revised April 1991

Abstract

Humans have a well developed ability to detect curvilinear structure in noisy images. Good algorithms for performing this process would be very useful in machine vision for image segmentation and object recognition. Previous approaches to this problem such as those due to Parent and Zucker and Sha'shua and Ullman have been based on relaxation. We have developed a simple feedforward and parallel approach to this problem based on the idea of developing filters tuned to local oriented circular arcs. This provides a natural second order generalization of the idea of directional operators popular for edge detection. Curve detection can then be done by methods very similar to those used for edge detection. Experimental results are shown on both synthetic and natural images. We also review data from an experiment investigating human preattentive line segregation and present predictions from our model that agree with this data.

Technical Report UCB/CSD 91/619, Computer Science Division(EECS), University of California at Berkeley.

Contents

1 Introduction **1**

2 Previous Approaches **2**

3 Our Approach **4**
3.1 Oriented Circular Arc Detectors 5

4 Experimental Results **7**

5 Experimental Comparison with Psychological Data **15**

6 Discussion **17**

1 Introduction

Consider the images in Figure 2b. A spiral curve is immediately perceived even though its contour is fragmented and it is embedded in a cloud of randomly oriented and placed edge segments. This is an instance of the phenomenon of curvilinear grouping which has been much studied after its initial discovery by the proponents of the Gestalt school of psychology. Much of this work was before the era of computational modeling – the Gestaltists concentrated instead on constructing clever examples which to this day adorn the pages of undergraduate psychology textbooks. Their explanation of the phenomena was in terms of 'laws' such as proximity, similarity, good continuation etc. Unfortunately, these laws are quite ambiguous and elastic. They lack the precision needed to support computer implementation.

Why should researchers in computational vision be interested in developing models of this phenomena? Is it merely an interesting illusion–after all there isn't really a spiral in the figure–or is this a manifestation of an essential and useful mechanism? We believe that the usefulness of mechanisms for curvilinear grouping comes from the fact that the world consists of a set of piecewise smooth surfaces. As a consequence, any image of the world consists of a set of regions corresponding to the smooth surface patches, bounded by piecewise smooth 1-dimensional curves corresponding to discontinuities in some scene attribute – depth, orientation, reflectance etc. The mechanism of curvilinear grouping helps us to develop explicit representations of these boundary curves. These boundary curves are useful for a number of visual tasks: object recognition, locomotion (by defining the boundaries of free space) and manipulation.

In computer vision, the usual approach to finding curvilinear structure is edge (or line) detection but that is clearly not adequate for the task of detecting the spiral in Figure 2b. In linear filtering based approaches [Can86, PM90], the short segments can be easily detected. To detect the longer range structure, however one needs very elongated oriented filters. While they would be helpful in detecting long straight lines, the curved nature of the spiral would make these straight elongated filters ineffective. Global approaches to edge detection based on the reconstruction of a piecewise smooth surface using line processes [GG84, BZ87] are even more inapplicable.
There are however two families of approaches which are applicable:

1. Edge detection either followed by or accompanied with a relaxation process.

Two recent examples of this approach are Parent and Zucker [PZ89] and Sha'shua and Ullman[SU88].

2. Edge detection followed by recursive grouping processes in which higher level groupings are created based on some measure of their significance. This approach is due to Lowe [Low85].

We review and critique these approaches in Section 2. In Section 3, we develop our approach which is based on using a bank of arc detectors tuned to segments of circular arcs of different orientations and curvatures. These operate on the results of convolving the image with a bank of oriented edge/bar sensitive filters. Experimental results are presented in Section 4. In Section 5 we review data from an experiment investigating human preattentive line segregation and present predictions from our model that agree with this data. In Section 6 we conclude with discussion and directions for future research.

2 Previous Approaches

Parent and Zucker [PZ89] have developed a relaxation labelling approach to curve inference which makes clever use of the notion of curvature consistency. They use a two stage procedure. The first stage – the "measurement" stage – is the convolution of the image with a bank of oriented linear filters which are tuned to oriented line segments. This results in initial tangent estimates at each position for each (quantized) orientation. The second stage – the "interpretation" stage is a relaxation labelling stage. In their approach, a curve is locally characterized by its (quantized) tangent and its curvature class (one of 7, based on empirical reasons). The support for a particular tangent estimate at a particular position is from those tangents at neighboring positions which are members of the same curvature class. The experimental results are impressive, though the run times are very long (upto 12 hours on a DEC VAX 11/780).

The Parent – Zucker scheme can also be interpreted in an optimization framework. Relaxation labeling tries to achieve 'consistency' which results from the maximization of a functional corresponding to the average local support. An alternative approach is based on formalizing curve detection directly as an optimization problem. This has a long tradition e.g., [Mon71]. A recent representative is the work of Sha'shua and Ullman [SU88] who try to compute a "saliency map" which is a representation of the image emphasizing salient locations. In an image

like Figure 2b, the saliency should be higher for points along the spiral, even at the locations where there is no line segment in the original image. They obtain the saliency map by an iterative algorithm for maximizing a measure which favors long curves that have low total curvature. The computational complexity of the problem is made tractable by their choice of measure which permits a recursive, dynamic programming type approach. The experimental results on a number of examples – similar to Figure 2b – are good. The number of iterations is large (10, 30 or 160 in the examples in [SU88]) though considerably smaller than the length of the curves.

Our primary critique of the Parent–Zucker, the Sha'ashua–Ullman and similar approaches is based on their use of iterative or relaxation labelling algorithms which should be done with extreme caution (if at all) in early vision. When presented as models of human vision, these models must confront the fact that neurons are very slow devices (firing rates usually less than 200-300 Hz) and that curve detection can be done preattentively (times on the order of 100-200 ms) which means that there simply isn't time for a large number of sequential time steps. Even in machine vision applications, where one may not be concerned about matching biological constraints, it is advantageous to have few or no iterations and short run times. One would like to explore fully what can be achieved using simple local, parallel, feedforward mechanisms.

A different approach to the problem of curvilinear grouping is due to Lowe [Low85] who uses a statistical approach to the general problem of perceptual organization, and specifically to the problem of detecting curvilinear structure. The basic idea is to assign a degree of "significance" to each potential grouping of image features – how likely is it to reflect some causal relationship in the scene instead of being merely the consequence of an accidental viewpoint. For curvilinear grouping, he starts with lists of edge pixels and examines groupings at all scales at all locations along the curve with adjacent groupings overlapping by 50%. There is a selection procedure which prunes these groupings on the basis of their significance.

Lowe's demonstration of his approach is primarily for detecting circles and lines. It is not clear (at least to us) how the approach would work, say in the examples in Figures 2 and 4 in this paper. For the general case of arbitrary smooth curve detection, a test of significance of a given curve would have to rely on a statistical model much more sophisticated than that needed for detecting the significance of groupings based on collinearity, co-termination and parallelism. Perhaps the approach can be used in general, but a demonstration is awaited.

Saund [Sau90] presents an algorithm that is a variant of Lowe's approach. The algorithm starts from the list of edge pixels and operates at multiple scales in a fine to coarse fashion. At scale i the algorithm forms the groupings by finding the "best" groups of level $i-1$ arcs that can be fitted with a with an arc of larger curvature such the error of the fit is not above a preset threshold. To avoid the combinatorial explosion that would have resulted from testing all possible groupings at a given stage, the algorithm employs a greedy, best-first strategy. Both Lowe's and Saund's algorithm operate in an iterative manner that prevents them from being fully parallelizable, and makes them implausible biological models of early vision processes.

3 Our Approach

Our approach is motivated by the desire to find a simple feed-forward mechanism for detecting curvilinear structure. Let us first study the simpler case of the detection of linear structure.

Suppose that the image consists of a set of randomly oriented straight lines of constant width. In that case, one could use a linear filtering approach as follows: convolve the image with rotated copies $f \circ R_\theta$ of the vertical *even-symmetric* filter $f(x,y) = G''_{\sigma_1}(x)G_{\sigma_2}(y)$ where R_θ corresponds to rotation by θ. This would give rise to a response image $E(x,y,\theta)$. By choosing larger values of σ_2, one can improve the signal-to-noise ratio of this oriented operator arbitrarily within the practical limits of the size of the image and the lengths of the lines in the image. If the image contains straight edges instead of lines, one could convolve the image with rotated copies of the *odd-symmetric* filter $f(x,y) = G'_{\sigma_1}(x)G_{\sigma_2}(y)$ which gives rise to a response image $O(x,y,\theta)$.

In either of these cases, picking the maximum (over θ) of the response image would enable one to pick the dominant local orientation. Finding the maximum of the response, in the direction perpendicular to this would enable the localization of the edge or line. This approach is quite robust to missing segments of the contour, randomly oriented irrelevant line segments, imaging noise at the pixel level etc. All one needs to do is to use more and more elongated operators to increase the signal-to-noise ratio. The argument for using directional operators has been made many times in the computational vision literature e.g. [Bin81, Can86].

Since one may not know in advance whether the linear structure in the image corresponds to lines or edges, one can compute oriented energy as suggested by

Morrone, Owens and their colleagues [MO87, MB88] – create a new response image $V(x, y, \theta) = E^2(x, y, \theta) + O^2(x, y, \theta)$. Maxima in $V(x, y, \theta)$ correspond to oriented lines or edges or some combination as may occur in real images in the presence of mutual illumination and specularities. This approach[PM90] permits the localization of composite edges without systematic error in localization. It may be noted that the computation of $O(x, y, \theta)$, $E(x, y, \theta)$, and $V(x, y, \theta)$ can be done quite efficiently, with bounded error, with only a finite number of convolutions. Details may be found in [FA90, Per90].

Now let us return to the general problem of detecting curvilinear structure. One cannot improve signal-to-noise ratio arbitrarily by having elongated operators because boundary curves in images are, in general, not long straight lines. As the support of the operator is increased, the straight line model soon ceases to be a good approximation of the neighborhood of a curve.

It may be noted that a straight line is merely a first order model of a curve; it represents only tangent information. A second order model of a curve would be an oriented circular arc – that represents both local orientation and curvature. If we could somehow generalize the mechanisms that work for detecting lines to mechanisms for detecting oriented circular arcs, one would have a better chance at improving the signal-to-noise ratio by increasing the support of the operator while remaining in a neighborhood such that the second order model is a good approximation.

We will adopt this approach. When we used first order models of curves, the result of filtering the image was a response image[1] $L(x, y, \theta)$ and the dominant local orientation/s could be picked by finding the θ which maximizes $L(x, y, \theta)$ at a given location in the image. When we use second order models of curves, we will get a response image $C(x, y, \theta, \kappa)$ where κ is the curvature (reciprocal of the radius of circular arc). The dominant local second order model i.e. best fitting oriented circular arc could be picked by finding the (θ, κ) pairs that maximize $C(x, y, \theta, \kappa)$.

3.1 Oriented Circular Arc Detectors

The standard method for detecting parametrized curves in the computational vision literature is the Hough transform – see Ballard and Brown[BB82] (Chapter 4) for a good review. It is an ideal method if one wishes to detect parametrized

[1] the response image $L(x, y, \theta)$ can be either $O(x, y, \theta)$, $E(x, y, \theta)$ or $V(x, y, \theta)$

curves (such as circles) *globally* in the image. However, it is unsuitable for us because it is too global for our purpose. For generic boundary curves, circular arcs are good models locally, but not necessarily globally. An analogy may be made to Fourier Transforms and Gabor filters – Fourier Transforms, which are localized in the frequency domain but are global in the spatial domain, are unsuitable for analysis of images whereas Gabor filters which are localized both in the spatial and in the frequency domain are suitable.

There are several possible designs of local oriented circular arc detectors. We suppose that there is a first stage of filtering for detecting linear structure that results in the computation of $L(x, y, \theta)$. Figure 1 shows a circular arc $(x(s), y(s))$ of radius $\frac{1}{\kappa}$ passing through the point (x_0, y_0). Let $\theta(s)$ be the angle made by the tangent to the curve. Let $G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}}$ be the Gaussian function with σ being the standard deviation.

Here are two expressions for computing $C(x_0, y_0, \theta_0, s_0)$

$$C(x_0, y_0, \theta_0, \kappa) = \int_{-S}^{S} L(x(s), y(s), \theta(s)) G_\sigma(s)\, ds \tag{1}$$

or

$$\begin{aligned} C(x_0, y_0, \theta_0, \kappa) \quad = \quad & \left(\int_{-S}^{0} L(x(s), y(s), \theta(s)) G_\sigma(s)\, ds \right) \times \\ & \left(\int_{0}^{S} L(x(s), y(s), \theta(s)) G_\sigma(s)\, ds \right) \end{aligned} \tag{2}$$

Both these definitions are natural generalizations of the idea of having a blurred template. The Gaussian weighting term makes the process local – pixels further away on the circular arc make smaller contributions. In both the formulas, one is computing the contribution from the circular arc on each side of the point (x_0, y_0). The two one-sided contributions may either be added (Equation 1) or multiplied together (Equation 2). Certainly other definitions are possible.

One weakness of the additive definition is that it leads to the over-extension of circular arcs beyond their endpoints – same as the over-extension of edges beyond junctions when directional operators are used in a straightforward way. The multiplicative definition does not suffer from this weakness – instead it results in a shrinkage at the end-points of arcs. We consider this to be desirable as this favors longer curves.

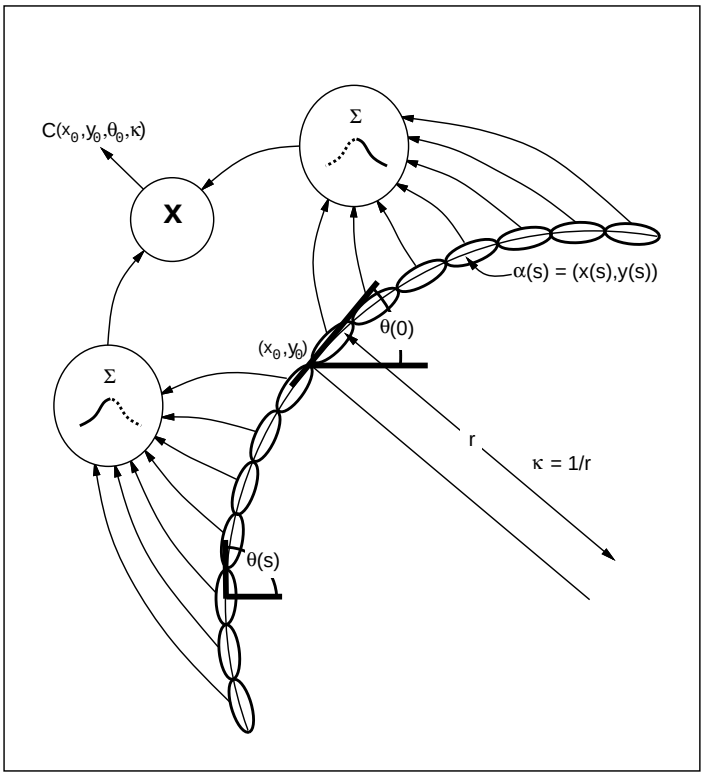


Figure 1: An oriented circular arc detector. The small ellipses represent the linear filters whose output is $L(x(s), y(s), \theta(s))$.

4 Experimental Results

We implemented the scheme described above using definition 2. $C(x, y, \theta, \kappa)$ was computed for quantized values of orientation (every 5 degrees) and curvature (5 or 6 values: $1/4$,$1/8$,$1/16$,$1/32$,0, and $1/64$). There is nothing intrinsic about this particular quantization – in our approach these are just parameters. The integration limit S was the minimum of 2σ and the value that limits the support to 45 degrees of arc on either side of the point (x_0, y_0). The angle limit results in greater range of integration for arcs of lower curvature. This is taken into account by a suitable normalizing factor.

After $C(x, y, \theta, \kappa)$ has been computed, one can declare curves by a similar

approach to that used for edge detection[PM90]. We compute the first 3 local maxima in θ of

$$C_{max}(x, y, \theta) = \max_{\kappa} C(x, y, \theta, \kappa),$$

and then do non-maximum suppression by moving in the direction perpendicular to the tangents corresponding to the maximizing values of θ and marking only those pixels that have a larger value of $C_{max}(x, y, \theta)$ then either of the two directional neighbors that are within $\delta°$ of the maximizing θ value (in these experiments we used $\delta = 30°$). We call the resulting image the curve image, by analogy with the output of edge detectors that is usually referred to as an edge image.

For first three examples, $L(x, y, \theta)$ was set to the combined response image $V(x, y, \theta)$, and in the last two examples L was set to the odd-symmetric response image $O(x, y, \theta)$. The even and odd-symmetric filters used are in the framework described in [PM90], with $\sigma_1 = 1, \sigma_2 = 3$ pixels, where σ_1 and σ_2 govern the "width" and the elongation of the filter, respectively. The distance attenuation parameter σ in Equation 2 used to compute $C(x, y, \theta, \kappa)$ was either 4 or 7 pixels, with 5 or 6 curvature values, respectively.

For each image we computed the global averages μ of the lower 65% response of the sampled angles at each pixel. The figures of curve images shown here are the result of thresholding the curve images at $\gamma\mu$, where γ was chosen manually (values of γ are given in Table 4).

Figure 2b shows the image of a dashed spiral embedded in a cloud of randomly oriented and placed edge segments. Figure 2c, and d are the curve images extracted by our algorithm using σ of 4 and 7, respectively – it may be noted that the spurious structures detected can be traced back to the image.

Figure 3a shows a dotted circle with the dots positionally jittered about the circle. Figure 3b shows the image composed by adding random dot noise. Note that the noise is sufficient to make the detection of the circle difficult even for human observers. The curve image is shown in Figure 3c, and d (σ of 4 and 7). The difference between 3c and 3d does not seem significant.

Figure 4a shows a rather poor quality pointillist image, and Figure 4b shows the edges found in it by the Canny edge detector. Figure 4c, and d shows the curves extracted by our scheme. Note particularly the curves extracted by our scheme on the left side of the silhouette of Maria, also some of the folds in her dress on the right. One may also notice that with σ of 7 longer "gaps" are bridged, but in some cases at the expense of the suppression of shorter curves by longer ones, as is evident in the details around Maria's face.

For the next two examples we ran our algorithm on images used by Lowe [Low88] to demonstrate an algorithm for extracting and segmenting smooth image boundaries. Starting from an edge image produced by the Canny edge detector, his algorithm links the edgels into edge chains that are then smoothed at multiple scales by a shrinkage free method. A combination of low curvature variation and maximum length criteria are used to select the "best" scale for representing the curve segments. The 1D smoothing stage is necessary to be able to estimate the curvature and its derivative at points along the curves.

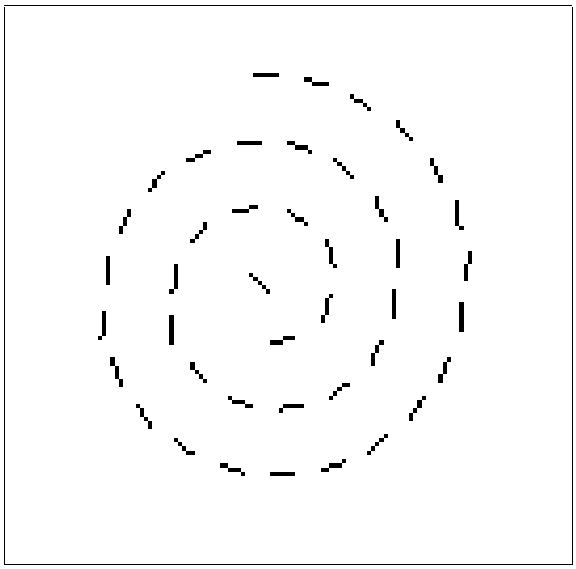
Figure 5a and b show a grainy image of a totem pole and the initial edge image used by Lowe's algorithm. Figure 5c, and d are the curve images produced by Lowe's algorithm and by our scheme, respectively. Note the details around the eye, where due to low contrast, noise, and the inherent inability of the Canny edge detector to declare multiple orientation at a single pixel, a fragmented set of edges is generated. As Lowe's algorithm is dependent on this input, its also outputs a fragmented set of smoothed curves. Similar effects are also evident around the black spot on the totem's cheek, and in the truncation, away from the outline of the nose, of the long curve running underneath the eye. In all case, our algorithm produces a less segmented and more complete set of curves. It may also be noted that as our algorithm generates curvature, as well as orientation estimates it output can be used to segment the curve chains (without the initial smoothing), or provide better input to Lowe's smoothing stage.

Figure 6 is also a comparison with [Low88]. Note, for example, that in contrast to the edge detector output and to Lowe's results, our algorithm produces a complete outline for the cranking disk attached to the drill, and correctly determines the outline of the round box above the staple remover.

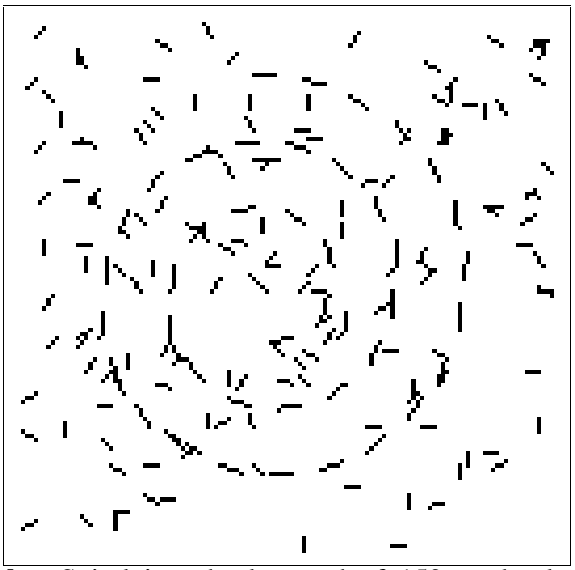
The algorithm was coded in Lisp and C and ran on a SUN 4/300. CPU times and threshold multipliers are quoted in the table below:

Image	Size	$\sigma = 4$	$\sigma = 7$	γ
Spiral	128x128	5min	6.4min	1.5
Circle	128x128	5.16min	6.68min	1.5
Maria	110x110	3.68min	4.95min	1.0
Totem	258x214		45.6min	2.5
Tools	255x255		49min	1.5

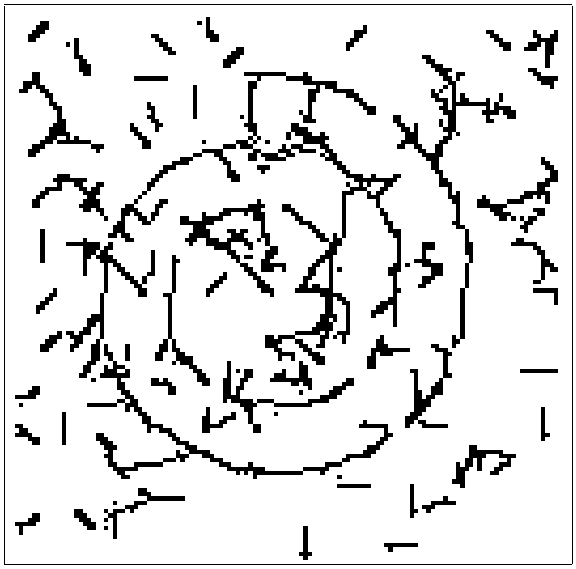
Table 1: Algorithm running times



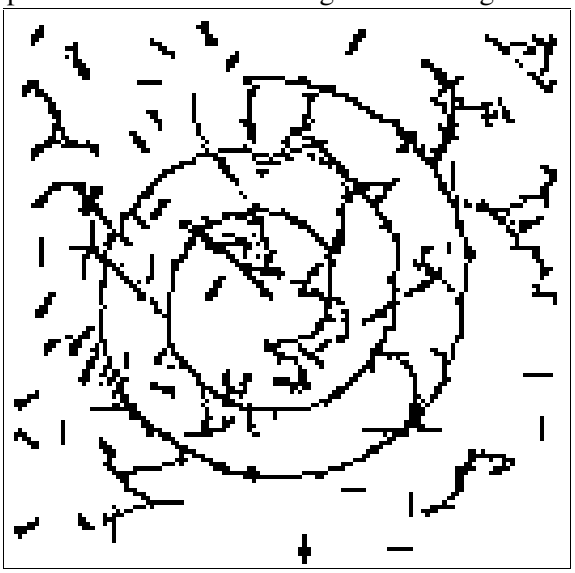
a. A spiral made of 44 segments of length 4.



b. Spiral in a background of 150 randomly placed and oriented line segments of length 3.

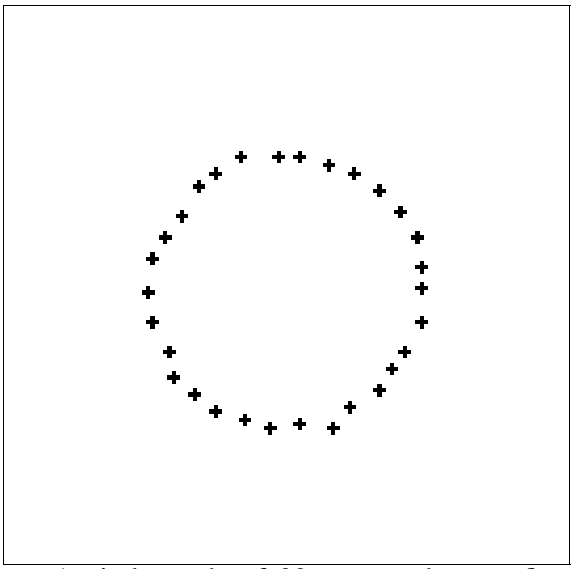


c. Curve image of noisy spiral with $\sigma = 4$.

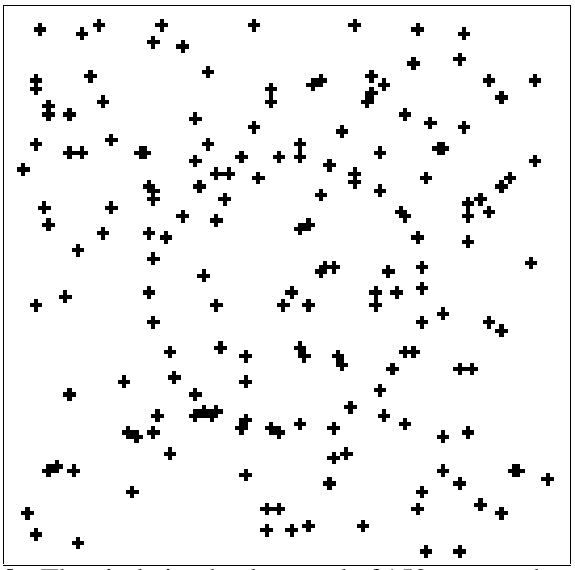


d. Curve image of noisy spiral with $\sigma = 7$.

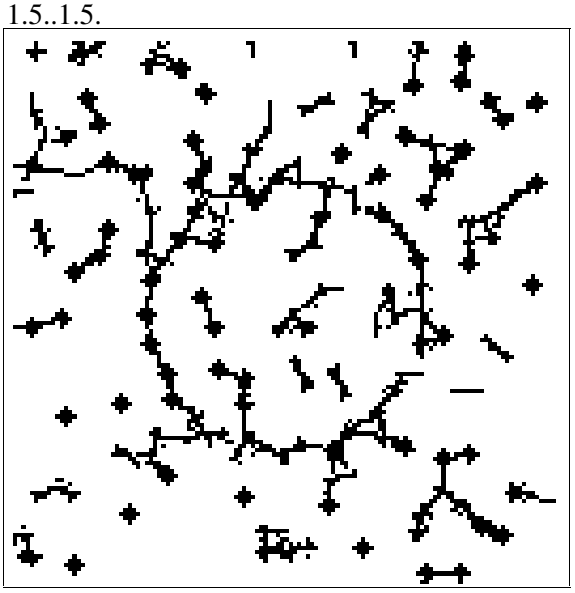
Figure 2



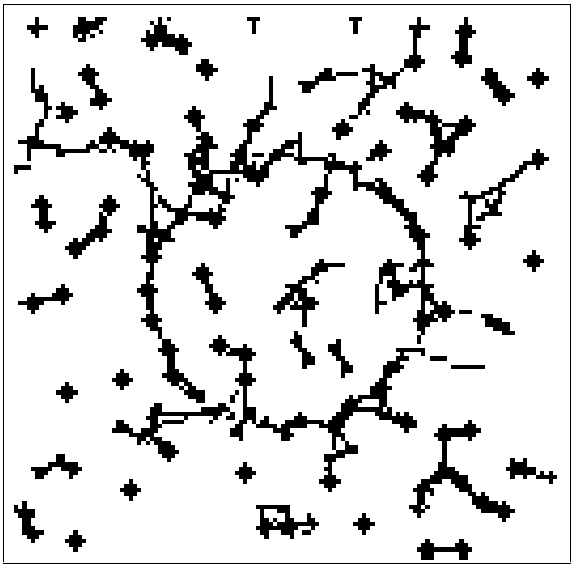
a. A circle made of 30 crosses that are first placed evenly around the circle and then each is jittered uniformly in x and y in the range -1.5..1.5.



b. The circle is a background of 150 crosses that are uniformly distributed on the image.



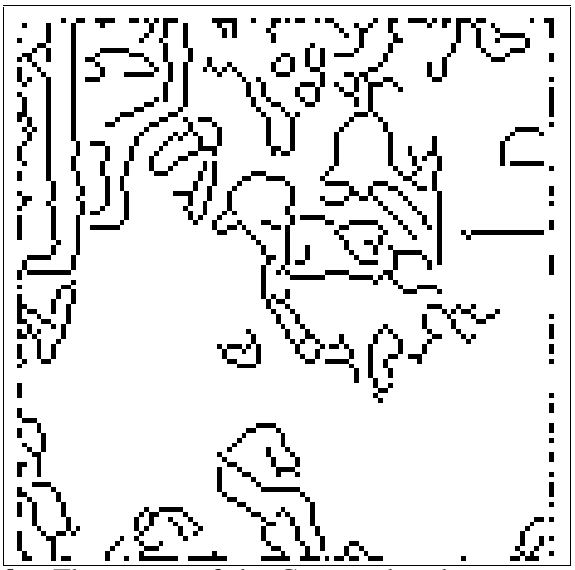
c. Curve image of noisy circle with $\sigma = 4$.



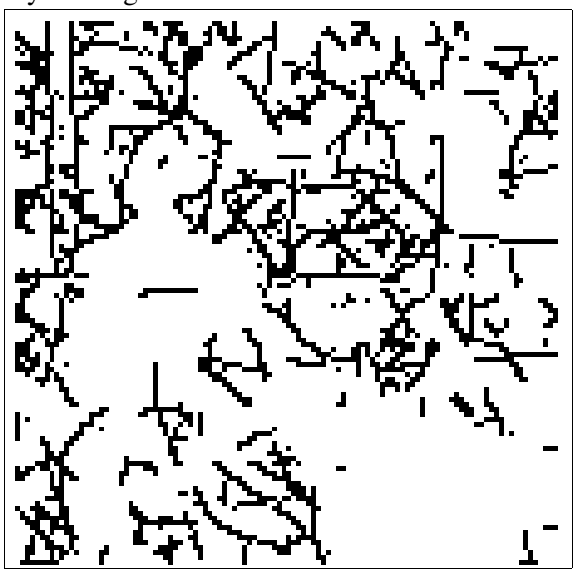
d. Curve image of noisy circle with $\sigma = 7$.

Figure 3

a. "Maria van der Velde at the piano", Theo van Rysselberghe.



b. The output of the Canny edge detector on Maria (standard deviation 1).



c. Curve image of Maria, $\sigma = 4$.

d. Curve image of Maria, $\sigma = 7$.

Figure 4

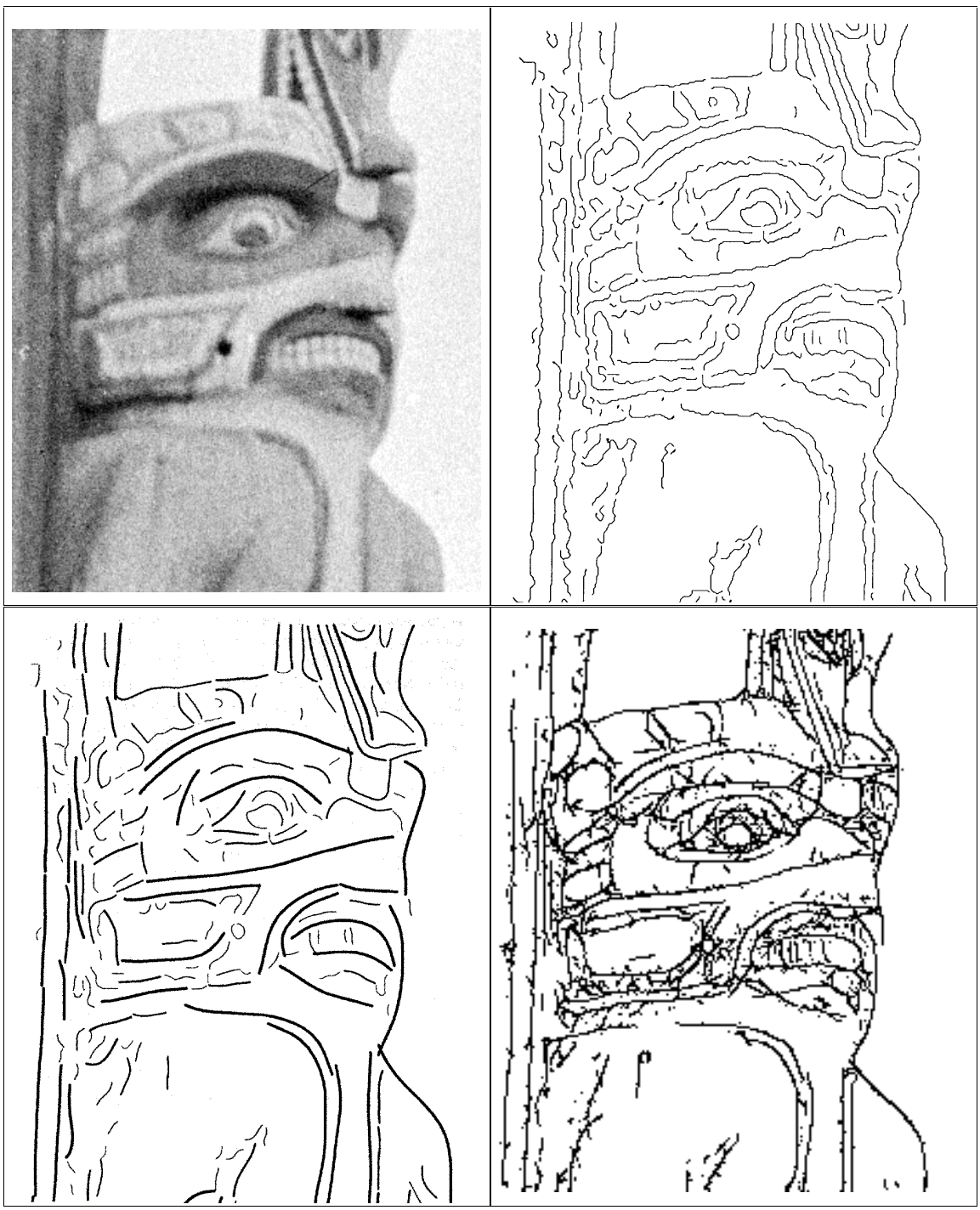


Figure 5: Left to right and top to bottom: **a.** The totem image; **b.** Edges found by the Canny edge detector; **c.** Curves found by Lowe's algorithm; **d.** Curve image produced by our algorithm. b, and c are from [Low88]