# SHAPE RECOGNITION USING RECURSIVE MATHEMATICAL MORPHOLOGY

Mike Schauf, Robert M. Haralick

*Department of Electrical Engineering, University of Washington, Seattle, WA 98195*

This paper introduces a size invariant method to recognize complex two-dimensional shapes using multiple generalized recursive erosion transforms. The method accomplishes the same kind of recognition that templates of each shape at multiple scales would do, but the method takes constant time per pixel regardless of the scale of the shape. We illustrate the technique on shapes having multiple randomly generated parts and randomly scaled and translated into a binary image having other shapes. The paper discusses the shape generation, image generation, image perturbation, recursive transforms, and recognition methodology. Results from the initial feasibility experiments show the methodology is able to detect each model's scale and position.

## 1 Perspective

By a two-dimensional shape we mean a connected area that distinguishes itself from its locally surrounding background. In this note we do not concern ourselves with the nature of the distinction. Here we are only concerned with the shape itself and how the techniques of recursive mathematical morphology can be used to describe and recognize a shape class.

Our setting will be in the domain of binary images. A shape prototype description will be specified by a connected set of binary one pixels and a spatially surrounding set of binary zero pixels. The size of the prototype will be in the high range of the scales of interest. The width of the spatially surrounding set of binary zero pixels is considered to be one of the parameters of the shape prototype itself. The shape class associated with the prototype will be the set of all digital scalings of the prototype. Since we are in the domain of digital images, there will be a smallest scale and it will be the scale for which the thinnest part of the shape is at least two pixels wide.

A shape prototype consists of primitive shape parts translated and rotated with respect to one another and constrained so that the resulting shape prototype is a connected set. The primitives used in the feasibility study reported here consist of circles, rectangles, triangles, sectors, lines, and parallelograms.

## 2 Recursive Erosion Transform

The erosion of an image $A$ by a structuring element $B$ is the set of all translated origins of the structuring element $B$ where $B$ can be wholly placed in $A$. The definition of erosion is defined below

The erosion of $A$ by a structuring element $B$ is denoted by $A \ominus B$ and is defined

$$A \ominus B = \{x \in E^n \mid \textstyle\bigcap_{b \in B} A_{-b}\}$$

As an example take the image and structuring element in Fig. 1

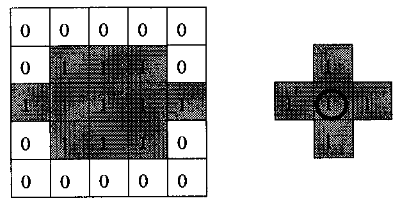

Figure 1: On the left is the binary image and on the right is a cross-shaped structuring element.

The positions where the cross-shaped structuring element can be wholly placed in the image can be seen in Fig. 2 and the resulted erosion is seen in Fig. 3.

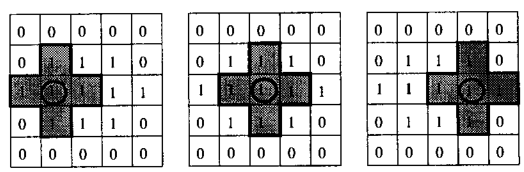

Figure 2: The structuring element can be wholly covered in the three locations shown.

The erosion transform is found by eroding the image and each erosion result by the structuring element $B$ over and over again $(A \ominus B \ominus B \ldots)$ until nothing is left in the result. The number of times a pixel 1 value remains at a coordinate in each erosion result is the value for that coordinate in the erosion transform.

Rather than do the erosions over and over again and count the number of times a pixel appears in the result, recursive morphology simplifies the erosion transform into two passes. First the structuring element is broken into two

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Figure 3: The resulting erosion.

parts, that is before the origin in top-down, left-right scan order, called $y$, and that which is after, called $z$.

The algorithm for finding the erosion transform is listed as follows.

1. In left-right, top-bottom scan order, sweep through the input image. If a pixel value of 0, define the output pixel at that position to have the value 0. If a pixel has a value of 1 than do the following.
   - translate the structuring element $y$ to the current position on the output image
   - find the values of the output image pixels for each of the locations in the translated structuring element $y$
   - select the minimum of the values and add 1
   - give the pixel at the current position on the output image this minimum value
2. Call the resulting image $G$
3. In a right-left, bottom-top scan order, sweep through the image $G$.
   - translate the structuring element $z$ to the current position on the output image
   - find the values of the pixels of $G$ for each of the locations in the translated structuring element and add 1
   - find the value of $G$ at the current position
   - place in the current position of $G$ the minimum value of all the values

Fig. 4 shows the first pass and second pass result of the erosion transform using the image and structuring element in Fig. 1.

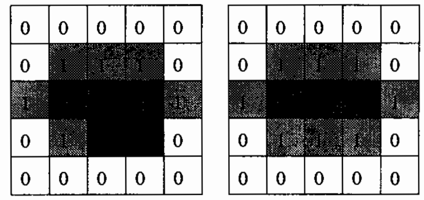

Figure 4: On the left is the first pass of the image in Fig. 1 using the cross structuring element and on the right is the second pass and final erosion transform result.

## 3 Methodology

The scale invariance recognition is based on the following idea: the erosion transform of any scale of a shape prototype will result in a scaled version of the original erosion transform. For example if we were to scale a shape prototype to half size, the erosion transform of the scaled shape will result in a same shaped erosion transform as the original but half the size and the maximum value of the erosion transform will be half that of the original. And proceeding away from the location of the maximum value, the values of the erosion transform will ramp down in the same way as in the original. The key to using the erosion transform for recognition is the fact the position of the scaled original maximum transform value from the center of the model is the scale of the maximum original transform value.

For an example, Fig. 5 is a possible result of an erosion transform of two scaled models. The larger is the outline of the erosion transform of the model and the smaller is the outline of the erosion transform of the model scaled by .5. An X marks the maximum value position of both transforms. As can be seen the global maximum of the original transform is twice that of the scaled transform and the positions also have the same scale.

Using structuring elements that span all directions results in different transforms and can provide information about each of the different shape classes.

One problem with this methodology is the need for a preprocessing noise removal step. This is because any perturbation of the image will result in changes in the erosion transform. The changes could be very large, for example if the perturbation introduces a hole in the middle of a shape. Since the maximum values are mostly in the middle of the erosion transform, noise on the outside of the model will have just a slight effect on the positions and values. But if the maximum is on the edge of the model, it likely will be

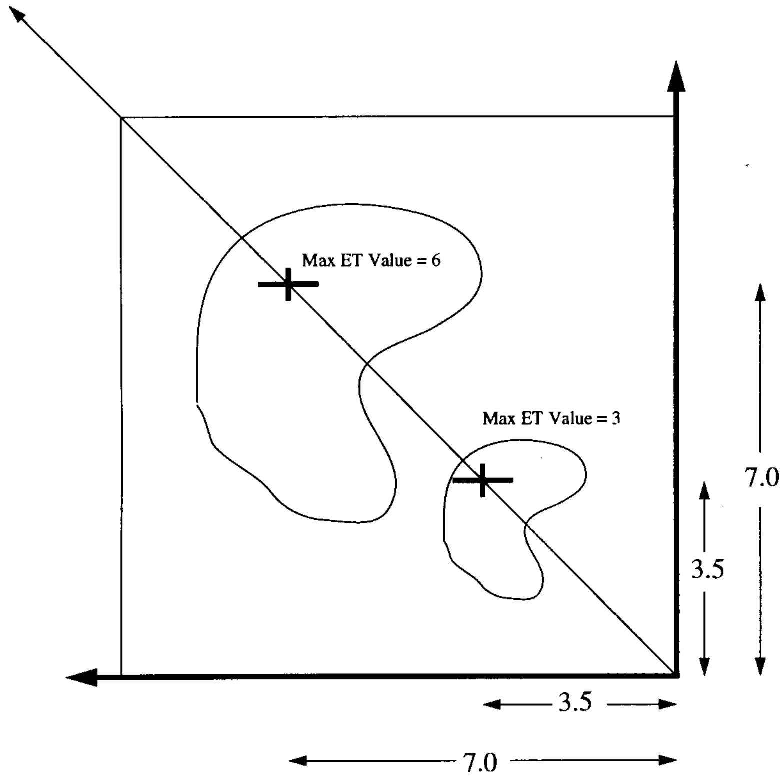


Figure 5: In this figure the shape prototype is the larger of the two objects and its maximum ET value is marked with a cross. The center of the shape prototype is at the origin. The smaller shape is a .5 scaled version of the shape prototype with its center also at the origin. It can be seen that the result from scaling is that the max ET value in the scaled model ET is also scaled .5 and it's coordinates are also scaled by .5.

affected. Fig. 6 and Fig. 7 shows an example of this. In Fig. 7 it can be seen that the maximum erosion transform value is different resulting from just two added pixels.

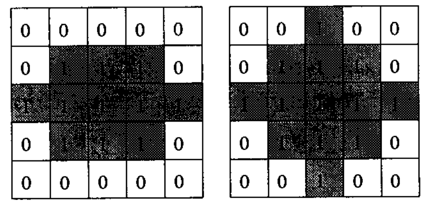

Figure 6: The figure on the left is the image from Fig 1. The central figure is the image from Fig. 1. with a pixel of noise added above and below.

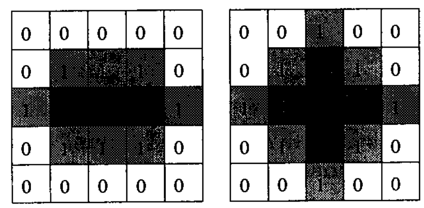

Figure 7: The figure on the left is the erosion of the image from Fig 1 without noise. On the right is the erosion transform of the image with added noise.

To account for these possibilities, the algorithm was formed to include the case of this happening. It was decided that matching scales had to be within one of the scale candidate, and the position of the maximum had to be within five pixel places of the scaled maximum pixel position.

### *3.1 Algorithm*

The algorithm for shape detection has an offline and an online part. The offline part consists of doing the following for each of the shape prototypes:

1. Calculate the recursive erosion transform of the shape prototype using the first structuring element.
2. Mask the first primitive with the model and select the maximum values.
3. Run a connected components on the results of step 2 and select the coordinate that is the center of all the maximum pixel positions.

4. Record in the training data the center position of the masked maximum transform value relative to the center of the model along with the maximum value.

5. Repeat steps 2 through 5 for every primitive in the model.

6. Calculate the recursive erosion transform of the first model using all the other structuring elements and repeat steps 2 through 6.

The offline part produces the training data for the recognition stage. The traning data consists of the $x$ and $y$ coordinates of the central maximum pixel location relative to the center of the shape prototype and the maximum value of the erosion transform.

The online recognition algorithm consists of the following steps.

1. Starting with the image's erosion transform from the first structuring element, find the maximum value and select the first position in the image that has that maximum value in left-right top-down raster scan order.

2. To find a scale candidate, divide the maximum value in the image erosion transform by the maximum value found in the erosition transform of the first shape prototype.

3. Find the model's center candidate by subtracting the scaled coordinates of the maximum transform value in the shape prototype from the position of the maximum erosion transform value in the image.

4. Now with a possible center and scale of the model, check to see if indeed the scaled shape prototype exists by going to each scaled position where the masked erosion transform value was at a maximum for each primitive and calculate the scale. If the scale for each position is the same as the candidate scale, then with respect to the structuring element, we can hypothesize that a scaled version of the shape prototype exits centered at the location.

5. Keeping the center candidate and scale candidate, repeat step 5 using the image erosion transforms for the rest of the structuring elements and the training data of the first shape prototype with each respective structuring element.

6. If there are a sufficient number of times for which we cannot reject the hypothesis that the shape matches we assert the hypothesis of a shape

match and record its scale and position. Then we determine the bounding box for the scaled shape prototype and use it to mask out all those pixels in the bounding box.

7. If there are not a sufficient number of times for which we cannot reject the hypothesis that the shape matches, steps 3 through 7 are repeated for the next shape prototype.

8. Finally repeat the entire process on the remainder of the image.

The algorithm is based on the fact that the maximum value in the image erosion transform must be a scale of the maximum value of the erosion transform of one of the shape prototypes. This is so because of the characteristic of the erosion transform and the fact that the scaled shapes on the image do not touch. Knowing this we can divide the maximum value in the image erosion transform by the maximum value in each shape prototype's erosion transform to get candidates for scale. With scale we can now calculate the center of the scaled model since the position of the maximum value is just the scale of the position of the maximum value of the shape prototype. Then we check to see if the shape class does exist there by going out to each scaled maximum masked position and checking if it has the same scale there as the scale just calculated.

## 4 Testing and Results

Testing the feasibility of the method requires the generation of primitives, shape prototypes, and images of perturbed scaled and translated shape prototypes. This is described in the following sections.

### *4.1 Shape Generation*

A shape prototype is composed of a set of primitives. Each primitive is mildly constrained so that its digital image bears a reasonable resemblance to the continuous primitive ideal. The line primitives must be at least five pixel wide and at least five pixels long. The triangles and parallelograms must have vertex angles greater than 30 degrees and each side must be at least 30 pixels long. The circles and sectors must have a radius of at least 30 pixels long and the sector must have an angle greater than 30 degrees. Finally the sides of rectangles must be at least 30 pixels long.

The shape prototype is constrained so that the set of its constituting primitives form a connected set with the overlap constrained to between each pair of connecting primitives constrained to be less than 5%. Overlap between a pair of connecting primitives is measured as the area of the union of the primitives divided by the sum of the areas of the primitives.

*4.2 Image Generation*

Image generation requires scaling the shape prototypes and randomly placing them in the image so that they do not overlap with any other shape already in the image. For this feasibility study, the image size is 512 by 512 and the scales are randomly generated between .33 and 1.

*4.3 Image Perturbation*

1. For each foreground pixel, $f_i$, compute the pixel's distance from the nearest background pixel. Let this distance be $d(f_i)$.

2. For each background pixel, $b_i$, compute the pixel's distance from the nearest foreground pixel. Let this distance be $d(b_i)$.

3. For each foreground pixel, compute the probability of switching to a background pixel,

$$P(0|d(f_i), foreground, c_0, \alpha_0, \alpha) = c_0 + \alpha_0 e^{-\alpha d^2(f_i)}$$

   Generate a random number between 0 and 1 and flip the pixel to a background pixel (0) depending on whether the random number is lower or higher than the pixel-switching probability.

4. For each background pixel, compute the probability of switching to a foreground pixel,

$$P(1|d(b_i), background, c_0, \beta_0, \beta) = c_0 + \beta_0 e^{-\beta d^2(b_i)}$$

   Generate a random number between 0 and 1 and flip the pixel to a foreground pixel (1) depending on whether the random number is lower or higher than the pixel-switching probability.

5. Perform a morphological closing of the resultant image with a disk structuring element of diameter $stElSize$. This introduces a correlation amongst neighboring pixels.

*4.4 Feasibility Test*

It was decided that there would be four different models each 256 by 256. Each model consisted of randomly generated primitives placed at random positions, making sure each was slightly overlapped to another. Than with four different models, four different images were generated randomly placing the four different models in the image at randomly generated scales between .3 and 1. Training data was obtained from the models and images and later the software was run on each of the images.

Fig 8 shows the four different models generated and Figures 9, 10, 11, and 12 show the four generated images.

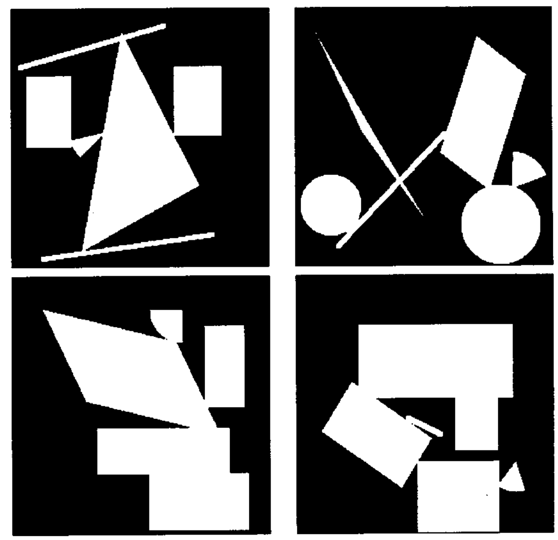

Figure 8: The four randomly generated models for testing

Fig. 13 shows the 14 different structuring elements.

The erosion transform of model one and image one with the eleventh structuring element is shown in Fig. 2.

We ran two experiments in which each there were 14 different noise perturbations of each image. The results of the experiments are summarized in Tables 1-5.

## 5 Further Work

The reported study is only a feasibility study. To refine the method for operational use would require a proper statistical treatment of the estimation and scale and the way in which the matching takes place. We discuss this in the next subsection. Also, it would be reasonable to use the locally surrounding binary 0 pixels of each shape prototype in the same way that the binary 1 pix-

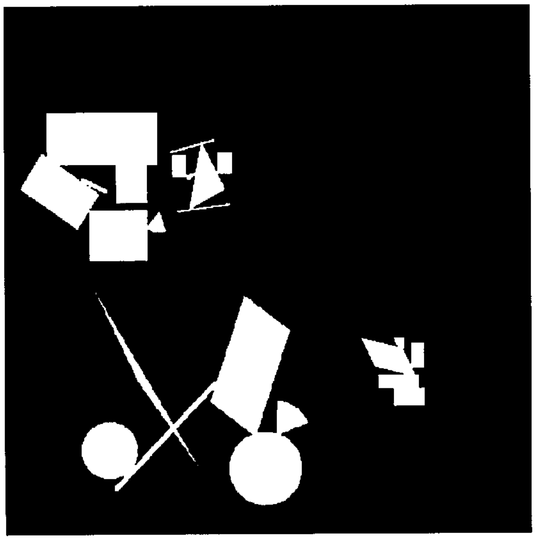

Figure 9: The first generated image

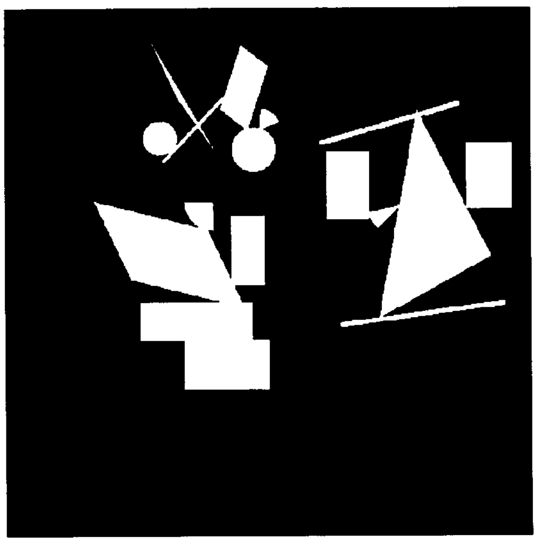

Figure 10: The second generated image

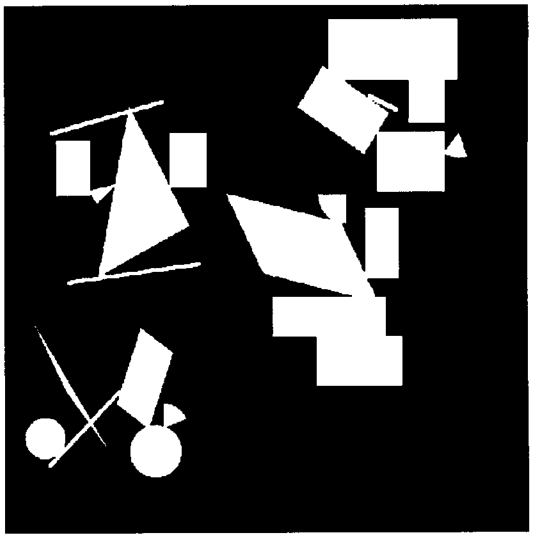

Figure 11: The third generated image

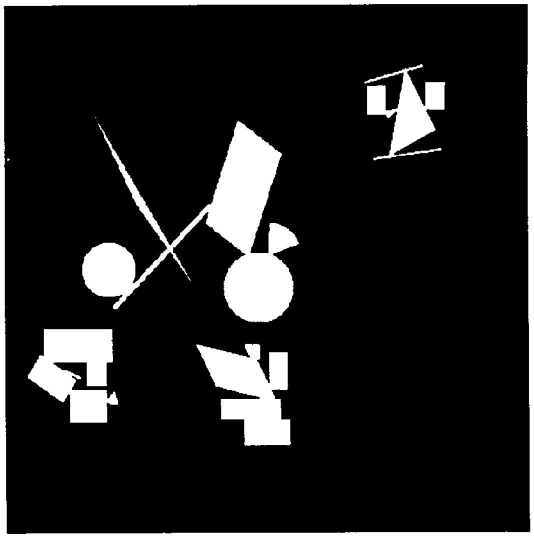

Figure 12: The fourth generated image

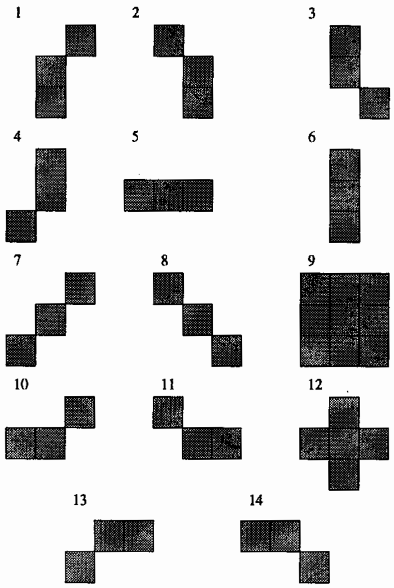


Figure 13: The 14 different structuring elements used for the erosion transforms

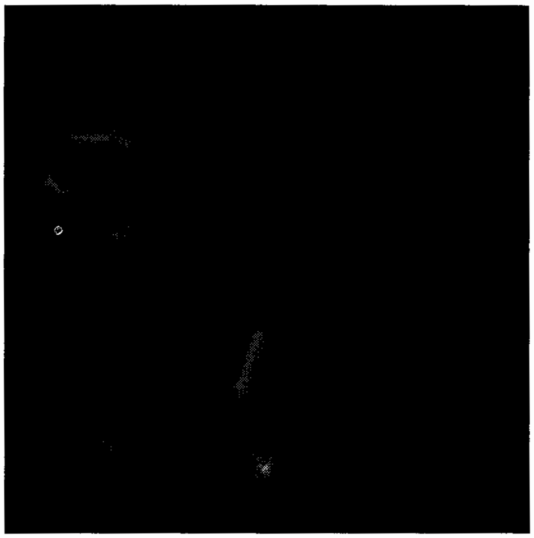

Figure 14: Erosion transform of the first image by the eleventh structuring element

Table 1: Recognition Results from Ideal Images

**Image 1**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (127.36, 160.00) | 0.31 |
| Model 2 | (276.86, 74.71) | 0.90 |
| Model 3 | (321.24, 346.21) | 0.31 |
| Model 4 | (102.93, 15.21) | 0.70 |

**Image 2**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (92.00, 307.46) | 0.92 |
| Model 2 | (32.58, 133.50) | 0.55 |
| Model 3 | (186.85, 86.92) | 0.85 |
| Model 4 | no model | |

**Image 3**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (91.86, 42.93) | 0.77 |
| Model 2 | (309.08, 19.15) | 0.65 |
| Model 3 | (183.00, 216.21) | 0.85 |
| Model 4 | (13.00, 287.43) | 0.45 |

**Image 4**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (57.23, 351.54) | 0.40 |
| Model 2 | (108.43, 75.86) | 0.87 |
| Model 3 | (328.00, 185.25) | 0.46 |
| Model 4 | (313.71, 19.79) | 0.45 |

Table 2: Recognition Results from Degraded Images

**Image 1**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (127.00, 160.00) | 0.31 |
| Model 2 | (276.64, 74.50) | 0.90 |
| Model 3 | (321.33, 346.67) | 0.30 |
| Model 4 | (102.67, 15.60) | 0.70 |

**Image 2**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (91.71, 306.71) | 0.93 |
| Model 2 | (32.29, 132.93) | 0.55 |
| Model 3 | (187.00, 86.50) | 0.83 |
| Model 4 | no model | |

**Image 3**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (92.14, 43.86) | 0.76 |
| Model 2 | (308.92, 18.92) | 0.65 |
| Model 3 | (183.14, 216.43) | 0.85 |
| Model 4 | (13.08, 287.46) | 0.45 |

**Image 4**

| Model | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (57.23, 351.62) | 0.40 |
| Model 2 | (108.36, 75.86) | 0.87 |
| Model 3 | (327.83, 185.16) | 0.46 |
| Model 4 | (313.45, 19.79) | 0.45 |

Table 3: Ground Truth Image Data

**Image 1**

| **Model** | **Position (r,c)** | Scale |
|---|---|---|
| Model 1 | (129, 161) | 0.30 |
| Model 2 | (276, 74) | 0.90 |
| Model 3 | (322, 347) | 0.30 |
| Model 4 | (103, 15) | 0.70 |

**Image 2**

| **Model** | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (91, 306) | 0.93 |
| Model 2 | (33, 134) | 0.54 |
| Model 3 | (187,86) | 0.83 |
| Model 4 | no model | |

**Image 3**

| **Model** | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (91, 44) | 0.76 |
| Model 2 | (310, 20) | 0.64 |
| Model 3 | (183, 216) | 0.85 |
| Model 4 | (13, 287) | 0.82 |

**Image 4**

| **Model** | Position (r,c) | Scale |
|---|---|---|
| Model 1 | (58, 352) | 0.39 |
| Model 2 | (107, 75) | 0.87 |
| Model 3 | (329, 186) | 0.45 |
| Model 4 | (314, 21) | 0.44 |

Table 4: Shape Recognition Statistics for the Degraded Images

**Image 1**

| Model | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 14 | 0 | 0 |
| Model 2 | 14 | 0 | 0 |
| Model 3 | 14 | 0 | 0 |
| Model 4 | 14 | 0 | 0 |

**Image 2**

| Model | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 13 | 0 | 1 |
| Model 2 | 12 | 0 | 2 |
| Model 3 | 13 | 0 | 1 |
| Model 4 | no model | | |

**Image 3**

| Model | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 14 | 0 | 0 |
| Model 2 | 13 | 0 | 1 |
| Model 3 | 14 | 0 | 0 |
| Model 4 | 14 | 0 | 0 |

**Image 4**

| Model | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 13 | 0 | 1 |
| Model 2 | 14 | 0 | 0 |
| Model 3 | 12 | 0 | 2 |
| Model 4 | 14 | 0 | 0 |

Table 5: Shape Recognition Statistics for the Degraded Images

**Image 1**

| Models | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 10 | 0 | 4 |
| Model 2 | 14 | 0 | 0 |
| Model 3 | 3 | 0 | 11 |
| Model 4 | 10 | 0 | 4 |

**Image 2**

| Models | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 14 | 0 | 0 |
| Model 2 | 14 | 0 | 0 |
| Model 3 | 14 | 0 | 0 |
| Model 4 | no model | | |

**Image 3**

| Models | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 14 | 0 | 0 |
| Model 2 | 13 | 0 | 1 |
| Model 3 | 14 | 0 | 0 |
| Model 4 | 13 | 0 | 1 |

**Image 4**

| Models | Cor Det | False Alarm | Mis Det |
|---|---|---|---|
| Model 1 | 13 | 0 | 1 |
| Model 2 | 14 | 0 | 0 |
| Model 3 | 12 | 0 | 2 |
| Model 4 | 11 | 0 | 3 |

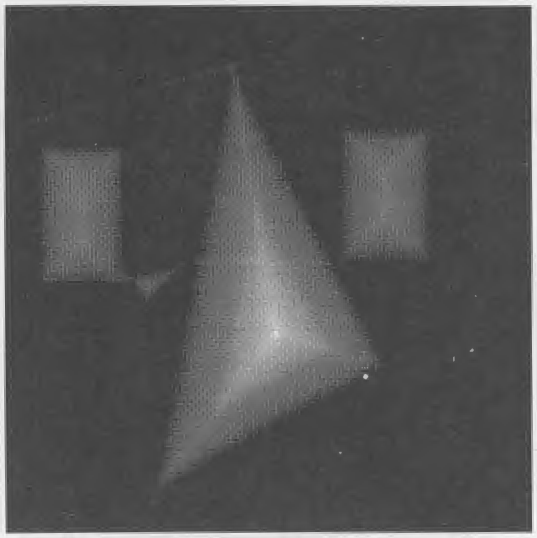

**Figure 15: Erosion transform of the first shape prototype by the eleventh structuring element.**

els are used. Both these improvements would undoubtedly change the results reported from being close to perfect to being very close to perfect.

### *5.1 Scale Estimation*

Let $\mu_1, \ldots, \mu_N$ be a fixed ordering of the $3 \times 1$ mean position-maximum-value vectors and $\Sigma_1, \ldots, \Sigma_N$ be the covariances of a given shape prototype. The first two components of each $\mu$ are the row column coordinates of a maxima of an erosion transform and the third component is the value of the maxima. The row column coordinates are specified with respect to the origin of the shape prototype. The ordering is determined by a fixed rule associated with the shape prototype and depends on the position of the maxima and from which structuring element the maxima comes about.

Let $x_1, \ldots, x_N$ be the same fixed ordering resulting from a noisy observation of a scaling of the shape prototype. Let $s$ be the unknown scale parameter. We model the noisy observations the following way:

$$x_n = s\mu_n + \psi_n$$

where $\psi_n$ has a Normal$(0,\Sigma_n)$ distribution and $\psi_m$ is independent of $\psi_n$ for $m \neq n$.

To estimate the value for $s$, we find that value which maximizes $P(s \mid x_1, \ldots, x_N)$. This is equivalent to obtaining an $s$ that minimizes

$$\sum_{n=1}^{N}(x_n - s\mu)'\Sigma^{-1}(x_n - s\mu) - 2logP(s)$$

We take the prior probability $P(s)$ to be Normal$(\mu, \sigma^2)$.
In this case, we seek the $s$ that minimizes

$$\sum_{n=1}^{N}(x_n - s\mu)'\Sigma^{-1}(x_n - s\mu) + (s - \mu)^2/\sigma^2$$

Differentiating with respect to $s$, collecting terms together and solving for $s$ results in

$$s = \frac{A + \mu/\sigma^2}{B + 1/\sigma^2}$$

where

$$\begin{aligned} A &= \sum_{n=1}^{N} \mu'\Sigma^{-1}x_n \\ B &= \sum_{n=1}^{N} \mu'\Sigma^{-1}\mu \end{aligned}$$

## 8 Conclusion

The feasibility study has four steps consisting of shape prototype generation, image generation, image perturbation, and recognition. The insight of the entire method was the use of the recursively computed erosion transform knowing that the erosion transform of a scaled model has a maximum erosion transform value and position proportional to the maximum erosion transform and position of the original shape prototype. And that proportion is the scale. Testing this methodology resulted in correctly detected position and scales of four different shape prototypes randomly placed in 4 images.


1. Haralick, Robert, (1991) Computer & Robot Vision, Vol. 1, Chapter 5 "Mathematical Morphology." Addison Wesley, New York.
2. Kanungo, Tapas, (1995) DDM Users Manual, University of Washington, Seattle, WA.