A Quick Way For Relational Matching: Morphology

Robert M. Haralick, Tapas Kanungo and John Strupp
Intelligent Systems Laboratory
Electrical Engineering Department, FT-10
University of Washington
Seattle WA 98195
U.S.A.

Abstract

This paper describes an approach to do relational matching morphologically with a computational complexity proportional to the number of pixels. We give a description of the theory for binary images and illustrate this theory with an example from handwritten numeral recognition.

1 Introduction

There are a variety of approaches to matching models, approaches as simple as template matching to those as complex as relational matching. The template matching approaches are simple to implement but are rather unforgiving for moderate perturbations, missing parts, or extraneous parts. The relational matching approaches are more robust in this regard but have a higher computational complexity. In this paper we describe an approach which is intermediate between template matching and relational matching. It has a computational complexity proportional to the the computational complexity of template matching and yet it can tolerate moderate perturbations. We illustrate the approach in the two-dimensional domain of handwritten character recognition.

2 Definitions and Notation

In this section we define all the necessary terms and give the notations used in this paper. The definitions of the basic morphological operations based on the tutorial by Haralick *et al.* [HSZ87] have been restated below for easy reference.

Dilation is the morphological transformation which combines two sets using vector addition of set elements. If A and B are sets in $\mathbf{Z}^2$, the dilation of A by B is the set of all possible vector sums of pairs of elements, one coming from A and one coming from B.

Definition 2.1 The *dilation* of A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{c \in \mathbf{Z}^2 \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\}.$$

Erosion is the morphological dual of dilation. If A and B are sets in $\mathbf{Z} \times \mathbf{Z}$, then the erosion of A by B is the set of all elements of x for which $x + b \in A$ for every $b \in B$.

Definition 2.2 The *erosion* of A by B is denoted by $A \ominus B$ and is defined as

$$A \ominus B = \{x \in \mathbf{Z}^2 \mid x + b \in A \text{ for every } b \in B\}.$$

Opening an image with a disk structuring element smooths the contour, breaks narrow isthmuses, and eliminates small islands and sharp peaks or capes.

Definition 2.3 The *opening* of a set B by a structuring element K is denoted by $B \circ K$ and is defined as

$$B \circ K = (B \ominus K) \oplus K.$$

As in the binary case, there are two primitive greyscale mathematical morphology operations, dilation, denoted by "$\oplus$" and erosion, denoted by "$\ominus$."

Definition 2.4 The grayscale dilation of f by k is defined by

$$(f \oplus k)(x) = max\{f(x - z) + k(z) | ((z \in K), (x - z) \in F)\}$$

The result is the maximum value of the sum between the ima and the structuring element value within the local neighbor defined by the spatial extent of the structuring element. In case, where all the values of the structuring element are zero, (height) the equation is reduced to:

$$(f \oplus k)(x) = max\{f(x - z)|((z \in K), (x - z) \in F)\}$$

In a zero height greyscale dilation, the result is just the m value of the image within the local neighborhood, as defined spatial extent of the structuring element. Thus, a zero height g dilation is an operation which tends to blur local bright regio the remainder of this paper, zero height structuring elements assumed for greyscale dilations.

Definition 2.5 The grayscale erosion of f by k is defined by

$$(f \ominus k)(x) = min\{f(x + z) - k(z)|(z \in K)\}$$

The result is the minimum value of the difference between the value, and the structuring element value within the local nei hood, as defined by the spatial extent of the structuring elemen special case, where all the values of the structuring element ar (i.e. zero height) the equation is reduced to:

$$(f \ominus k)(x) = min\{f(x + z)|(z \in K)\}$$

In a zero height greyscale erosion, the result is just the mir value of the image within the local neighborhood, as defined l spatial extent of the structuring element. Thus, a zero height gre erosion is an operation which tends to blur local dark regions the remainder of this paper, zero height structuring elements v assumed for greyscale erosions.

The greyscale closing, which is a dilation followed by an e using the same structuring element, tends to fill valleys (i.e. re local dark areas). The greyscale opening, which is an erosion fol by a dilation using the same structuring element, tends to cl peaks (i.e. remove local bright areas.)

3 Part Detection

In this section we illustrate some example ways in which a part on a binary image can be detected. A model part having a minimum width can be detected by opening the binary image with a horizontal line segment of this minimum width.

$$I_1 = I_0 \circ HorzLine(min\_width)$$

Such an opening might select other kinds of parts as well. Indeed all sections of parts which have a horizontal width greater than the length of the horizontal line segment would be selected. Only if the horizontal width was a distinguishing feature would the opening only select the part. Distinguishing feature in this case would mean that no pixel of any other part is in a horizontal extent of length *min_width*.

A model part having a maximum width can be selected by the complement of the opening intersected with the original image.

$$I_2 = I_0 - (I_0 \circ HorizLine(max\_width))$$

The intersection of these two operations can be used for a part description distinguished by a width range.

$$I_3 = I_1 \cap I_2$$

Similarly, for a part criteria of height, the same operations would be used, but with a vertical line as the structuring element. A class criteria that involves both a width and height component simultaneously, is done by combining both dimensions into a single structuring element. The combination is accomplished by a dilation, specifically dilation of the horizontal line by the vertical line, which results in a rectangle.

$$\begin{aligned} I_3 &= HorzLine(width) \oplus VertLine(height) \\ &= rectangle(width, height) \end{aligned}$$

The combination of two 1D structuring elements of lines into a single 2D structuring element of a rectangle can be extended again to generate a 3D structuring element. For a part criteria that has a given minimum, length width, and height a rectangular parallelepiped can be used.

$$\begin{aligned} I_5 &= Line(x,0,0) \oplus Line(0,y,0) \oplus Line(0,0,z) \\ &= parallelepiped(x,y,z) \end{aligned}$$

Structuring elements are not restricted to those made up of linear segments, for any shape can be used. For a class criteria of minimum diameter at all 2D angles, a disc can be used, while for 3D, a sphere would be used. For objects that have a minimum diameter in 2D and a minimum extent in a third, the dilation of a disc by the extent would be used, namely a cylinder.

For a less restrictive criteria which specifies the part to be of a minimum length in any of one or more directions, a generalized opening can be used. The generalized opening is the union of all the individual openings using the same structuring element, but which have been rotated at different angles. For example, if the part criteria is for a minimum diameter in any 2D multiple of 45 degree direction, the form of the generalized opening would be as follows.

$$I_6 = \bigcup_{\theta \in (0,45,90,135)} I_0 \circ Rotate(Line(min\_diam), \theta)$$

The known spatial relationship between two parts can be used in locating one part based on the location of the other part. The detection algorithm for the second part can be expressed morphologically in three steps. The first step is to dilate the detected first part to account for the distance uncertainty and to account for the size difference in the parts. The second step is to translate the dilated first part so that its center covers the second part. This translation would be the difference in translation of each part's center from the origin of the model coordinate reference frame. The third step is to intersect the translated dilated first part to detect the second part. As an example, consider two disks, R_1 and R_2, of known diameters, r_1 and r_2, $r_1 > r_2$, which are separated by a distance $D \pm d$. Given an image, I_{R2}, of the just isolated first disk and an image, I_B, of both disks, the region of the second disk, I_{R1} can be found as follows:

$$I_{R_1} = I_B \cap (I_{R2} \oplus disk(d + r_1 - r_2))_D$$

4 The Relational Morpology Approach

The idea of relational morphology is as follows. Like relational matching, a model is composed of parts, called primitives, which stand in spatial relation to each other. This spatial relation is translation. Recognizing a model requires identifying the primitives and verifying that they stand in the correct spatial relation. The theory we will give will provide a way of being able to produce morphological recognition algorithms with guaranteed correct classification rates under specified noise conditions.

Explaining the basis for how relational morphology works with binary images is simple. A model $\mathcal{M}$ consists of an ordered pair of two components identical in form. One component is a model $\mathcal{F}$ for the foreground and the other component is a model $\mathcal{B}$ for the background: $\mathcal{M} = (\mathcal{F}, \mathcal{B})$. The foreground part $\mathcal{F}$ consists of a set $\mathbf{P}$ of parts, each of which contains the origin, and a function s defined on $\mathbf{P}$ giving the translation of each part with respect to the model coordinate system: $\mathcal{F} = (\mathbf{P}, s)$. Likewise the background part $\mathcal{B}$ consists of a set $\mathbf{Q}$ of parts, each containing the origin, and a function t defined on $\mathbf{Q}$ giving the translation of each background part with respect to the model coordinate system: $\mathcal{B} = (\mathbf{Q}, t)$.

When a model appears in an ideal image, the model will be translated into the image. This means that each part will be translated by the same translation. The foreground parts will show up as binary one pixels and the background parts will show up as binary zero pixels. When a model appears in a non-ideal image, the situation is more complicated. Due to clutter, occlusion, and other variations, not all of the foreground or background parts will in fact occur in the image. Some may be missing. Furthermore, when parts appear in an non-ideal image they do not appear in the ideal form or position. Foreground and background parts will appear larger. And the position they occur in is perturbed by a small amount from their ideal positions. Finally, a real image may have many other things of non-interest in the image. This constitutes the clutter. The job of model recognition is to be able to identify each instance of a perturbed model in the image and not hallucinate any of the clutter as being an instance of the model.

Since if a part appears in a non-ideal image, it appears larger, the meaning of a part needs to be explained. A part in the model is not what should be considered to be an idealized model part. It is not in any way the nominal or ideal part. Rather, it is the smallest shape that would appear in the non-ideal image if the part indeed appears in the image. Thus, a model part is like a lower set bound on the nominal or normal part shape.

If a model appears in the ideal image, then each of its parts that appears in the image will be translated the same way into the image. For an non-ideal image, the random perturbation process prevents some parts from showing up in the image, makes those parts that do show up in the image larger, in the sense of set containment, and perturbs the translations of those parts that do show up in the image. Hence, for each number m of foreground parts and each number n of background parts, the random perturbation model has a parameter p_{mn} which is the probability that at least m foreground parts and n backgound parts occur in the image at position x if the model occurs in the image in the image at their positions x plus their translation plus a small translational perturbation.

Therefore, if x is the translation of the model to the image, then if a perturbed foreground part P occurs in the non-ideal image, instead of occuring as the ideal $P_{x+s(P)}$ it occurs as $[P \cup \psi(P)]_{x+s(P)+\delta(P)}$. Here $\psi(P)$ is the subset of points added to the lower set bound P due to the random perturbation process and $\delta(P)$ is the additive perturbation on the translation for the part P. Similarly if a perturbed background part Q occurs in the non-ideal image, instead of occuring as $Q_{x+t(Q)}$ it occurs as $[Q \cup \zeta(Q)]_{x+t(Q)+\eta(Q)}$. In this discussion we will assume that for every foreground part P, $\| \delta(P) \| \leq r$ and for every background part Q, $\| \eta(Q) \| \leq r$.

Let I denote a nonideal image, $\mathbf{P}^* \subset \mathbf{P}$ denote a subset of $\mathbf{P}$ having exactly m parts, $\mathbf{Q}^* \subset \mathbf{Q}$ denote a subset of $\mathbf{Q}$ having exactly n parts. Then we can say that with probability p_{mn}

$$\bigcup_{P \in \mathbf{P}^*} [P \cup \psi(P)]_{x+s(P)+\delta(P)} \subset I$$

and that

$$\bigcup_{Q \in \mathbf{Q}^*} [Q \cup \zeta(Q)]_{t(Q)+\eta(Q)} \subset I^c$$

Now suppose that for every foreground part P there is detection procedure $K(*;P)$ such that P is detected by $K(*;P)$ And that for every background part Q there is a detection procedure $L(*;Q)$ such that Q is by $L(*;Q)$. If the model occurs in the image I at position

x and if part P of the model does indeed show up on the image, then if the image I is processed with $K(*; P)$, it will certainly be the case that

$$P_{x+s(P)+\delta(P)} \subset K(I; P)$$

Similarly if the complement of the image is processed by $L(*; Q)$ and part Q of the model does show up on the image, it will certainly be the case that

$$Q_{x+t(Q)+\zeta(Q)} \subset L(I^c; Q)$$

To determine if a part appears on the image then amounts to doing a detection and compensating for the relative translation of the part and compensating for the random perturbation of the relative translation. Compensating for the relative translation amounts to just translating the detection. Compensating for the random perturbation amounts to doing a dilation. More precisely, if the model is translated onto the image with translation x and if part P of the model is one of the model parts that indeed does show up on the image, then $K(I; P)_{-s(P)} \oplus Disk(r)$ must contain P_x. Therefore, since P contains the origin, it must be the case that

$$x \in K(I; P)_{-s(P)} \oplus Disk(r)$$

The number of foreground parts which appear for a model translated into the image by translation x can then be written as

$$n(x) = \#\{P | x \in K(I; P)_{-s(P)} \oplus Disk(r)\}$$

Likewise the number of background parts which appear for a model translated into the image by translation x can be written as

$$m(x) = \#\{Q | x \in L(I^c; Q)_{-t(Q)} \oplus Disk(r)\}$$

Each of these sums is easily determined by doing a detection on the binary images, translating and dilating the detections, and then adding up the results. Recognition of the model at a translation x can be done by computing $n(x)$ and $m(x)$ and testing whether $m(x) > m_0$ and $n(x) > n_0$. Given that the model is translated into the image at x, the probability that $m(x) > m_0$ and $n(x) > n_0$ is precisely $p_{m_0 n_0}$, guaranteed.

5 Handwritten Number Recognition

When recognition needs to be done among different shapes, pairs of which have common parts, it is possible to make the recognition computation more efficient by not using all the model parts. Rather, the recognition can proceed sequentially examining differences in the presence of the parts permiting the required distinctions to be made. For example, the numerals 1, 4, 7 and 9 have a near vertical stem that differentiates them from 0, 2 , 3 5, 6 and 8. But only 4 and 9 have a "blob" towards the upper left part of the stem. Thus, detection of this part distinguishes 4 and 9 from 1 and 7. Further, the "blob" of 4 usually has a concavity facing north where as the "blob" of 9 does not have such a concavity. Thus, 9 and 4 can be distinguished by using this concavity feature.

In this section, the recognition of handwritten numerals is done sequentially with a morphological decision tree. At each node of the tree a sequence of operations is performed which distinguishes one group of numerals from another. We have just begun the task of developing this decision tree and in this section we describe what we have completed for distinguishing the numerals of 4, 6, 9 from each other. No attempt of character size normalization has been used. This may be necessary to add in at a later time. The work has been mainly concerned with understanding the shape characteristics of the numerals in morphological terms and has not reached the stage where we have begun evaluation. The methodology we are following is to do all our design work on the training set and when the design has been completed, we will then evaluate on a test set which has not been worked with during the design phase. So at this point we are not able to discuss meaningful error rate characteristics except what we have on the training set.

5.1 Morphological Sequences for Extracting Features

In this section we describe the morphological sequences for extracting features that characterize shapes of characters. Some of the primitive features we have looked at are are stems, blobs, vertical bars, and concavities.

5.1.1 Blobs

Blobs are formed in the image when loops or a configuration of pen strokes get connected into an "ink blot" when we close the image with a big structuring element. Thus if the image of the character is I, then the blob can be formed by the operation:

$$I_{blob} = I \bullet K$$

where K is a structuring element big enough to connect the strokes. For example, if we have the image of numeral nine, the loop of the nine can be morphologically closed by a structuring element bigger that the diameter of the loop. The loop will thus get filled and form a blob.

5.1.2 Stems

Stems are the ascending or descending lines in a character. It is the part of the line that is not also a part of a blob. For example, the numeral nine has a loop and a near vertical line at its right. The part of the line that extends below the loop is the stem. Similarly, the numeral six has a stem that extend above a loop.

Stems can be extracted by subtracting the blob portion from the character. Care must be taken while doing the closing for connecting up the blob. It must be done in a manner so that the concavity between the stem and the blob does not get filled. In the case of the numeral nine we used the following sequence to get the stems:

$$\begin{aligned} I_a &= I \bullet K_{wedge} \\ I_b &= I_a \circ K_{disk} \\ I_c &= I_b \oplus K_{square} \\ I_{stem} &= I_c^C \cap I \, . \end{aligned}$$

Here the structuring element K_{wedge} is a wedge that fits into the cavity between the stem and lower part of the loop but does not fit into the gap in the north east corner of the loop. Thus the closing by the wedge converts loops of size smaller than a wedge to a blob. Therefore, opening with the disk selects pixels which are not stem pixels. The structuring element K_{disk} is a disk with diameter bigger than the stem thickness. Therefore opening with the disk selects pixels which are not stem pixels. The structuring element K_{square} is a square with side equal to the diameter of the disk used in the opening. This dilation compensate for the random perturbation of the loop's relative translation. The origin of the square should be at the bottom one third of the height of the square so that the small tails that sometimes appear in the numeral 6 does not confuse the algorithm and thereby make a 6 get detected as 9. The last operation, the residue operation then selects the stem pixels.

5.1.3 Vertical Bars

Vertical bars are detected by opening the image with one pixel wide lines oriented at near vertical angles and then taking a union of the results. In our experiments we used lines of length 21 pixels and orientations ranging from 50 degrees to 110 degrees, measured counterclockwise from the horizontal axis, in steps of 10 degrees. The difference between vertical bars and stems is that stems are the portions of the vertical bars that extend above or below the blobs.

$$I_{VertBars} = (I \circ K_{\theta_1}) \cup (I \circ K_{\theta_2}) \cup \cdots \cup (I \circ K_{\theta_n})$$

where K_{θ_i} are the line structuring elements at various near vertical orientation.

5.1.4 Concavities

Concavities are regions of the background that are enclosed by the foreground in three of the north, west, south and east directions. The east, west, north and south concavities can be found using the following relations:

$$\begin{aligned}
\text{NORTH-CAVITY} &= (I \oplus N) \cap \overline{(I \oplus S)} \cap (I \oplus E) \cap (I \oplus W) \cap \overline{I} \\
\text{SOUTH-CAVITY} &= \overline{(I \oplus N)} \cap (I \oplus S) \cap (I \oplus E) \cap (I \oplus W) \cap \overline{I} \\
\text{EAST-CAVITY} &= (I \oplus N) \cap (I \oplus S) \cap (I \oplus E) \cap \overline{(I \oplus W)} \cap \overline{I} \\
\text{WEST-CAVITY} &= (I \oplus N) \cap (I \oplus S) \cap \overline{(I \oplus E)} \cap (I \oplus W) \cap \overline{I}\,.
\end{aligned}$$

Here E, W, N, and S are line structuring elements. N is a vertical line with the the origin at the bottom, S is also a vertical line structuring element but with the origin at the top of the line. E and W are horizontal line structuring elements. E has its origin at the left end of the line and W has its origin at the right end of the line. Further details can be found in [MG89].

5.2 Recognition Algorithm

In this subsection we analyze the characteristic features of the numerals belonging to group {4, 6, 9} and how to recognize them. The most salient features of the numerals belonging to this group are their stems, concavities facing the north direction, and "blobs". To distinguish among them, we notice that:

- Numerals 4, 6, and 9 have stems, vertical bars, blobs, and north concavities.
- Numeral 4 has a north concavity or a blob to the west of its vertical bar.
- Numeral 6 has a blob to the southeast of its stem.
- Numeral 9 has a blob or a north concavity to the northwest of its stem.
- Numeral 6 has a north concavity to the east of its stem.
- Numeral 4 has a north concavity to the west of its stem.

Hence, the following algorithm can be used to distinguish the numerals 4,6, and 9 from each other:

Algorithm:

1. **Find** stems.
2. **Find** blobs.
3. **Find** vertical bars.
4. **Find** north concavities.
5. **Dilate** the stems in the northwest direction.
6. **If** the dilated stems intersect the blobs,
   (a) **If** the vertical bars dilated west intersect a north concavity, recognize the numeral as 4.
   (b) Else recognize it as 9.
7. **Else:**

(a) **If** the vertical bars dilated west intersect a north concavity, recognize the numeral as 4.

(b) Else recognize it as 6.

5.3 Experimental Protocol

The algorithms were tried out on a data base having hand written numerals zero through nine obtained from C. Y. Suen. There were 200 samples of each numeral. These samples were of isolated numerals and were stored in a runlength encoded form. The characters were converted into binary raster scan formats and stored in matrices of size 65×60. The 200 matrices thus obtained were put in one image with 20 matrices in one row and 10 such rows. Thus the final image was of size 650×1200.

The structuring elements required for the morphological operations in various steps of the algorithms were judiciously chosen in order to accomplish the task specification of that step.

The morphological image processing was done using the GIPSY image processing software package. The machine used was Sun 4 running UNIX operating system.

This data set constituted the training data set and as our work is still in the exploratory stage we have not looked at our tried out anything on a test data set.

5.4 Results and Discussion

In figure 1 we give the sequence of morphological operations performed on for the group I numerals. The processing shown is for the numeral nine. Figure 1(b) is obtained by closing Figure 1(a) with a wedge pointing south west. On opening Figure 1(b) with a disk of diameter 9 we get Figure 1(c). Notice that there is no stem in the image. Now Figure 1(c) is dilated with a square such that it covers the loop of nine completely. The result is shown in Figure 1(d). In Figure 1(e) we get the stems by intersecting Figure 1(a) with the complement of Figure 1(d). To check if the blob is northwest to the stem, we first dilate the stem with a horizontal line 10 pixels in length and 10 pixels above the origin so that it is shifted up and smeared. This is shown in Figure 1(f). This takes care of the tolerance needed because of the variability in the positions of the stem and the loop. The shifted stem in Figure

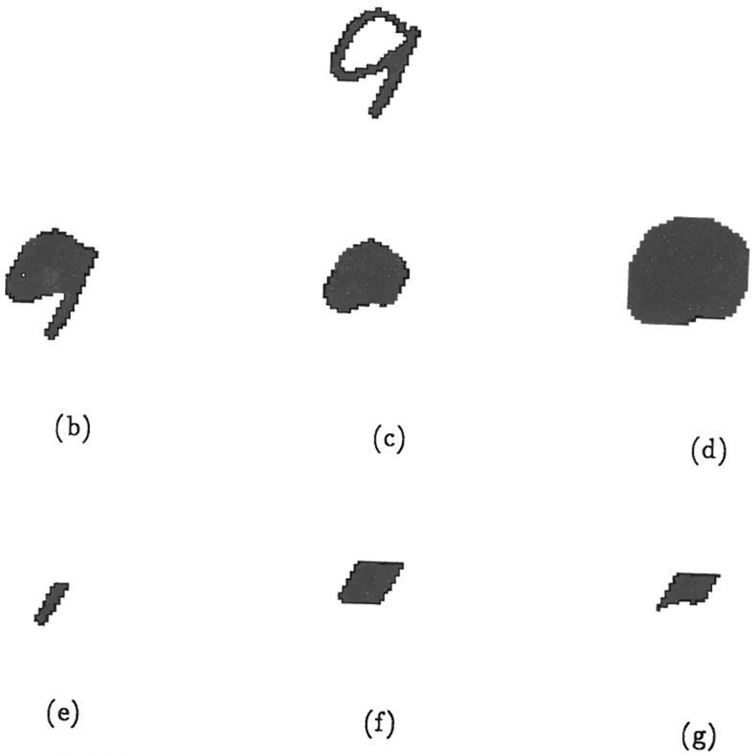


Figure 1: Morphological processing is shown for the numeral 9.

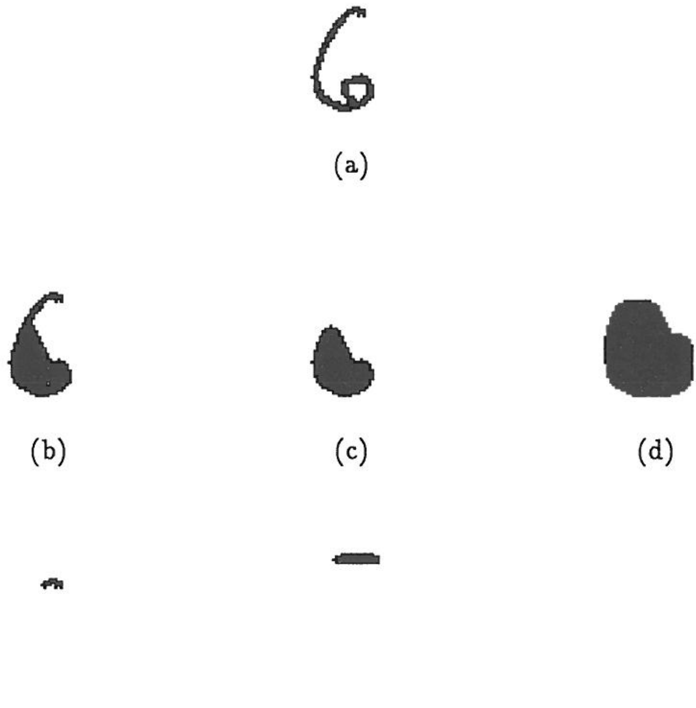
1(f) is then intersected with Figure 1(c). A non-empty intersection says that there is a blob to the northwest direction of the stem. Figure 2 is similar to Figure 1, only it shows the 9 recognition procedure being applied to a numeral 6. Of course, the 6 is not recognized as a 9.

We obtained the following result for the training data set: Two hundred samples of 4, 6, and 9 each were considered.

Numeral 4: 186 were recognized correctly as 4. Four were classified wrongly as 6 and ten others were recognized as 9.

Numeral 6: 189 were recognized correctly as 6, and eleven were recognized as 4.

Numeral 9: 191 were recognized correctly as 9, one was recognized as 6 and eight others were recognized as 4.



(a)

(b) (c) (d)

(e) (f) (g)

Figure 2: The 9 recognition procedure on the numeral 6.

It is important to note that the recognition procedure uses information in in the foreground as well as in the background. For example, for the numeral three we can either try and detect two semi-circular curves in the foreground or three wedges in the background due to the foreground.

6 Conclusion

We have described a general technique for recognition of a model, binary or gray scale, in which the spatial relation between the model parts is a translation. We have illustrated these ideas in the domains of handwritten character recognition. We have also done similar work in the extraction of biologic structure from MRI data with good success. Space has forced us not to say much about this work in this paper. We will report it in the near future. Our future work must concentrate on how to optimally estimate the parameters for all the processing sequences. We believe that this can be done since we have been able to put the entire process in the setting of a well-defined random perturbation model.

References

[Bio90] Biotechnology Computer Resource, Mayo Foundation, Rochester, MN. *Analyze User's Manual*, version 4.0 edition, 1990.

[HSZ87] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI–9(4):532–550, July 1987.

[MG89] B.T. Mitchell and A.M. Gillies. A model-based computer vision system for recognizing handwritten zip codes. *Machine Vision Applications*, 2:231–243, 1989.