Maschine Vision and Applications (2003) 13: 222–233

Machine Vision and Applications

Model-based detection, segmentation, and classification for image analysis using on-line shape learning

Kyoung-Mi Lee[1], **W. Nick Street**[2]

[1] Center for Artificial Vision Research, Korea University, Seoul 136–701, Korea (e-mail: kmlee@image.korea.ac.kr)
[2] Department of Managment Sciences, University of Iowa, Iowa City, IA 52242, USA (e-mail: nick-street@uiowa.edu)

Received: 5 November 2000 / Accepted: 29 June 2001

Abstract. Detection, segmentation, and classification of specific objects are the key building blocks of a computer vision system for image analysis. This paper presents a unified model-based approach to these three tasks. It is based on using unsupervised learning to find a set of templates specific to the objects being outlined by the user. The templates are formed by averaging the shapes that belong to a particular cluster, and are used to guide a probabilistic search through the space of possible objects. The main difference from previously reported methods is the use of on-line learning, ideal for highly repetitive tasks. This results in faster and more accurate object detection, as system performance improves with continued use. Further, the information gained through clustering and user feedback is used to classify the objects for problems in which shape is relevant to the classification. The effectiveness of the resulting system is demonstrated in two applications: a medical diagnosis task using cytological images, and a vehicle recognition task.

Key words: Detection – Segmentation – Incremental clustering – Classification – Unsupervised learning

1 Introduction

Three central tasks in image analysis are the detection, segmentation, and classification of objects from given images. Detection of the location of objects of interest and segmentation of their borders are the first steps of many image analysis tasks, especially for quantitative analysis of objects. For instance, the detection and segmentation of different structures in medical images, such as cells and organs, play important roles in diagnosis and prognosis. Unfortunately, even with the aid of image analysis software, traditional manual analysis is tedious and time consuming, especially in cases where a large number of objects must be specified. Thus the development of highly efficient and robust techniques to automatically and quickly detect objects and segment their exact borders is an important goal.

Correspondence to: K.-M. Lee

Detection and segmentation of an object does not mean its identification. Many applications of image analysis require an object to be classified. For example, Dubuisson et al. (1996) introduced an algorithm to extract the contours of five types of moving vehicles. They found and segmented the vehicle contours using deformable templates, and classified the vehicle of interest by the class of the matched template. In this and other studies, segmentation is used to get shape information. We extend this idea to use classification results to improve detection and segmentation. A current trend in automatic object segmentation and classification is the use of model-based methods to describe expected shapes (Duta et al. 1999). In this paper, we propose a model-based method for combining the three tasks of automatic object detection, segmentation, and classification.

These three problems have been addressed with machine learning methods as a means for improving knowledge used in the imaging process, and thus for producing more robust software (Beymer and Poggio 1996; Draper 1997; Maloof et al. 1998). The main drawbacks of applied machine learning for these tasks are the need for a large training set and the difficulty of learning new patterns after the initial training. Therefore, instead of gathering the entire data set and relearning it every time new data are collected, it is more desirable to learn incrementally, based on examples provided by the user while she is solving the problem. The proposed system uses an on-line learning method ideal for highly repetitive tasks such as morphological analysis of cytological and histological images, in which many cells or cell nuclei must be precisely outlined in many different images, and the shapes of the individual cells vary widely from one sample to the next (Lee and Street 2000a). However, our approach is not limited to medical domains. By using a very general shape model, our system can learn to segment and classify a wide variety of objects. This versatility is demonstrated with a vehicle recognition task.

The remainder of the paper is organized as follows. Section 2 reviews the background algorithms: the generalized Hough transform (GHT) and snake. Section 3 describes our shape model and an uncertainty region. Section 4 describes the proposed system, including the learning of templates. Experimental results in Sect. 5 show that the quality of segmentation produced using the proposed algorithm is comparable to an

exhaustive template search, and the computation time is significantly reduced. Conclusions and future work are presented in Sect. 6.

2 Background

2.1 Platform: Xcyt

The Xcyt program (Mangasarian et al. 1995; Wolberg et al. 1994) is a graphical computer program for diagnosing breast cancer and predicting the course of the disease. It performs analysis of cytological features based on a digital scan of a breast fine-needle aspirate, diagnosis of the image as benign or malignant along with an estimated probability of malignancy, and a prediction of when the cancer is likely to recur for cancerous samples. The program has proven highly effective in clinical practice, correctly diagnosing 97.6% of new cases since 1993 and providing accurate and individualized prognosis without lymph node information (Street 2000; Wolberg et al. 1999). In this paper, we propose a learning algorithm to perform the morphological analysis of the Xcyt program, then extend our approach to more general shapes.

2.2 Object detection: GHT

The proposed system uses the GHT (Ballard 1981) to detect the locations of objects in an image. The GHT is an extension of the Hough transform (Hough 1962), which is a standard template-matching algorithm for detecting complex patterns in images. To perform a GHT, each template is built as a displacement vector ($\vec{r}$) of θ in the look-up table in advance, requiring the user to know exactly what shape will be encountered. Further, when the scale and orientation of an input shape are variant and unknown in advance, brute force is usually employed to enumerate all possible scales (S) and orientations (Θ) of the input shape in the GHT process. This adds two dimensions to the parameter space, thus requiring the use of a 4D accumulator, $\mathrm{H}(x, y, S, \Theta)$. This dramatically increases the execution time and leads to sparsity in the accumulator, making the selection of strong matches more difficult.

There have been several methods proposed to reduce the huge memory storage capacity and computational time required for GHT (Kassim et al. 1999). An iterative approach to GHT (IGHT) was proposed to eliminate the extra storage dimensions by finding the template that best approximates the shape of each object (see Algorithm 1; Lee and Street 1999). Assume that an image includes an object with unknown shape and that there are templates, $\{T_1, T_2, T_3, \ldots, T_j\}$, one of which closely approximates the shape. We generate j accumulators, $\{A_1, A_2, A_3, \ldots, A_j\}$, by performing the GHT with the j templates. When the image includes a feature that closely matches the template, the accumulator contains a peak value. Although every accumulator includes its own peak value, the sizes of the values are different. If T_c is the most closely approximated template, the highest peak point should appear in A_c. In other accumulators, widely distributed and comparatively low values appear. We therefore determine that the image contains an object with the shape at template T_c, centered on the peak point at accumulator A_c. In the IGHT, a global accumulator is designed whose every point has the highest value of all accumulators at that location. After performing GHT with each successive template to fill the local accumulator, it compares each point in the local accumulator with the corresponding point in the global accumulator, and stores the larger value.

```
Initialize a global accumulator G
for s = Min_Size to Max_Size
    for θ = 0 to 360 step r
        Hough transform(s,θ) with a local accumulator L
        for each pixel point (x, y) in L
            if L(x, y) > G(x, y),
                G(x, y) = L(x, y)
Find peak values in G
```

Algorithm 1. Iterative GHT

The major problem of GHTs, as with other template-matching approaches, is the need to predefine templates to represent shapes. If precise segmentation is necessary and the objects' shapes vary, many templates are needed, requiring a significant search time. To overcome this drawback, the proposed system clusters shapes and averages them as a prototype of the cluster (we use the terms "template" and "prototype" interchangeably). In order to reduce the loss of accuracy due to reduction of time and space complexity, we incorporate flexible templates using uncertainty concepts (Lee and Street 2000c).

2.3 Segmentation: snakes

To segment the exact boundaries of objects, the proposed system uses an adaptive spline curve-fitting technique known as a snake (Kass et al. 1988). Snakes use an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes have been successful in performing tasks such as edge detection, corner detection, motion tracking, and stereo matching.

In the first application, the model is a closed curve that is attracted to strong edges in the image, and forms an arc in the absence of such edge information. The snakes are initialized using the results of IGHT that searches for ellipses of various sizes. Previous work (Lee and Street 1999; Street 2000) has shown that the Xcyt system isolates cell nuclei very well using the combination of IGHT and snakes. The user can then edit the resulting outline by dragging the boundary points to their desired location. The user may also remove an incorrect boundary, or draw a boundary on an undetected object by manually using the mouse to initialize the snake points. From the perspective of the learning method, this process creates

a collection of positive examples of the shapes that the user would like to find and outline.

3 Modeling shapes

In order to utilize or to extract the shape information of objects in an image, a suitable method for representing shapes is needed. In order to use the shapes as input data of a learning system, they must be positioned consistently on each of the training examples such that a particular point always represents that same part of the shape on each example.

Many different shape models have been proposed. Among the numerous shape models that have been used in learning approaches, the following are well known: moment invariants (Khotanzad and Hong 1990); a landmark-based model (Ansari and Delp 1990); contours using critical vertices (Mitzias and Mertzios 1994); an augmented, weighted model-attributed relational graph (Suganthan et al. 1997), and snakes (Tabb et al. 1999). The proposed system learns templates used in the look-up table of IGHT. Thus we need to find an appropriate shape model for incorporation into the IGHT look-up table. For example, those points should be represented by the relationship with θ.

3.1 Centroid-radii model

In order to model a shape we use a statistical shape model generated from a training set of *N* object descriptions $(\vec{v}^k)$, k=1, . . . , N. An object description, $\vec{v}^k$, is simply a labeled set of *n* points (v_i^k), i=1, . . . , n.

We adopt the centroid-radii model to represent the shape by a set of points (r_{θ_i}) in polar coordinates (Gupta et al. 1990; Chang et al. 1991; Lee and Street 2000b). Assume that OA is an arbitrary radius of the shape. In our algorithm, starting from OA and moving clockwise, we divide the circle into *n* equal arcs to place points around the boundary, with the regular interval being $360/n$ degrees. So the shape can be represented as a vector:

$$\vec{v} = (r_{\theta_1}, r_{\theta_2}, \ldots, r_{\theta_n}),$$

where r_{θ_i}, $1 \le i \le n$, is the ith radius from the centroid to the boundary of the shape and $\theta_i = \left(\frac{360}{n}\right) i$. The reflected shape can be represented as

$$\vec{v}' = (r_{\theta_n}, r_{\theta_{n-1}}, \ldots, r_{\theta_1}).$$

The three representations in Fig. 1 show how this model can facilitate multiresolution representation. This model assumes that *n* is larger than 2. We wish to use only as many points as necessary to adequately model the shape. For example, the breast-cancer cells for the first application use 24 radii and the vehicles for the second application use 36 radii.

In our system, shapes should be represented by the IGHT look-up table which associates a displacement vector $\vec{r}$ with a value of θ. In addition to being an easy representation to understand and implement, this model can be transformed to $\vec{r}$, an element of the IGHT look-up table, by substituting the reference point to *O*. Furthermore, it is invariant to translation through the use of an object-centered coordinate system, and to reflection through the use of the reflected shape. Since each representation can be rotated, scaled, and reflected, objects with the same shape but different sizes, orientations, and reflection can be modeled with one template. However, due to the lack of the internal information, we limit ourselves to topologically simple shapes with no holes, no multiple points at each angle, and no center outside.

3.2 Cluster and uncertainty region

Let a shape $\vec{v} = (v_i)$, i=1, . . . , n. A given set of *N* shapes, $\vec{V} = \{\vec{v}^1, \vec{v}^2, \ldots, \vec{v}^N\}$ will be partitioned into k clusters $\{P^1, P^2, \ldots, P^j\}$ such that shapes within the same cluster have a high degree of similarity, while shapes belonging to different clusters have a high degree of dissimilarity. Each of these clusters is represented by a prototype (P^c), c=1, . . . , j. Suppose N_c is the number of shapes in the cluster $\vec{c}$. Given the set of N_c shapes in the cluster $\vec{c}$, the algorithm finds the centroid of clusters or the prototype P^c by averaging the normalized shapes that belong to the particular cluster:

$$P^c = \bar{\mathbf{v}}^c = \left(\frac{\sum \vec{v}^c}{N_c}\right) = \left(\frac{\sum v_i^c}{N_c}\right), \; i=1, \ldots, n,$$

where $\vec{v}^c$ is a shape included in the cluster $\vec{c}$.

To measure the spread of a set of data around the center of the data in the cluster, we use the standard deviation (σ). A deviation is defined here as any distance from the average of a distribution, and the standard deviation is a measure of the width of a distribution equal to the square root of the average of the squared deviations. The standard deviation of P^c in the cluster **c**, is defined as:

$$\begin{aligned} \sigma(P^c) = \sigma(\bar{\vec{v}}^c) &= \left(\sqrt{\frac{\sum(\vec{v}^c - \bar{\vec{v}}^c)^2}{N_c}}\right) \\ &= \left(\sqrt{\frac{\sum(v_i^c - \bar{v_i}^c)^2}{N_c}}\right), \; i=1, \ldots, n. \end{aligned} \tag{1}$$

Note that here the denomonator in (1) is N_c, whereas some researchers use $N_c - 1$ instead of N_c. The use of $N_c - 1$ comes from a data set where only $N_c - 1$ data points are independent. However, in our shape learning all of our data and shapes are independent, and so we use N_c. For large samples

$$\frac{1}{N_c - 1} \sim \frac{1}{N_c}.$$

Using the standard deviation, the uncertainty region (ur) of the prototype P^c is calculated (Lee and Street 2000c):

$$\begin{aligned} \mathrm{ur}(P^c) &= \mathrm{ur}(\bar{\vec{v}}^c) \\ &= \left(\bar{\vec{v}}^c \pm m\sigma\left(\bar{\vec{v}}^c\right)\right) \\ &= \left(\bar{\vec{v}_i}^c \pm m\sigma\left(\bar{\vec{v}_i}^c\right)\right), \; i=1, \ldots, n, \end{aligned}$$

and the degree of the uncertainty is

$$\left|\mathrm{ur}(\bar{\vec{v}}^c)\right| = 2m\sigma(\bar{\vec{v}}^c)$$

where, for our experiments, we used $m = 2$.

Figure 2a shows the scattering of points in a cluster. It can be seen that some of the points show little deviation over the training set, while others form a more diffuse collection. Figure 2b shows a flexible template containing uncertainty areas, instead of just edges.

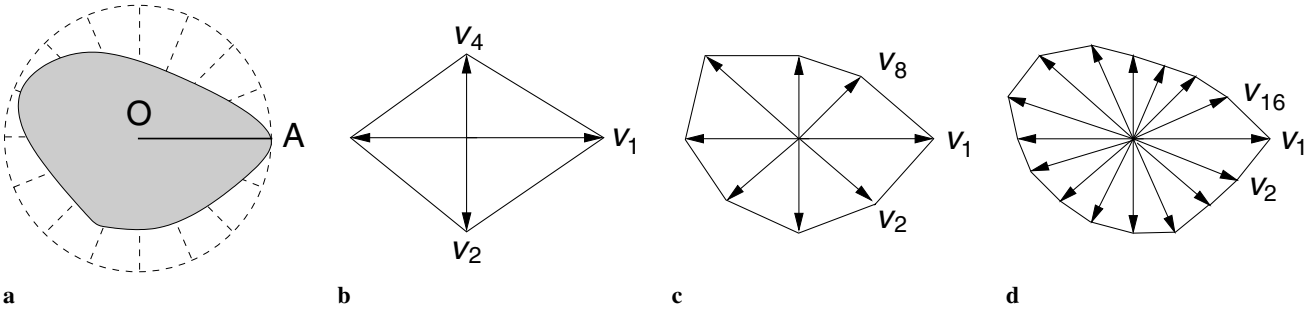


Fig. 1a,b. Shape representation: **a** the original shape; **d** representations of the shape at different resolutions

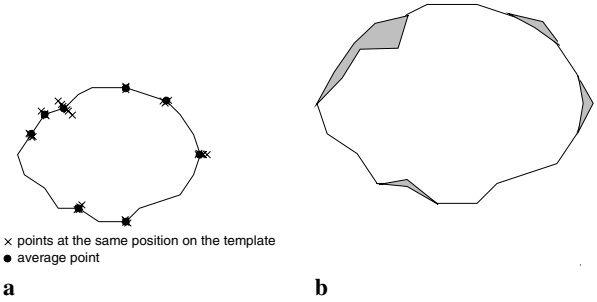


Fig. 2a. The scatter of points from an aligned set of a templates. **b** Flexible template with uncertainty regions

4 Description of the system

Figure 3 shows a flowchart of the proposed system. The system consists of four parts: detecting, segmenting, clustering, and classifying. While these four parts are performed repeatedly, the system learns and gathers knowledge of shapes that the user wants to find.

4.1 Preprocessing images

A preprocessing step is first applied to a given image (see Fig. 4). The preprocessing first smoothes the intensity values (Fig. 4b) across the image to diminish spurious effects that can be present in a digital image as a result of a poor sampling. We use a median-filtering algorithm with a 3×3 median filter. The Sobel edge detector is then used to detect steep grayscale gradients and determine their direction (Fig. 4c). The Sobel detector produces the approximate absolute gradient magnitude at each point in an input grayscale image, and thus introduces weighted incrementing of the accumulator in the GHT. Edge thinning is then performed (Fig. 4d), which locates the local maxima of the gradient magnitudes. This reduces the number of edge components so that further analysis is facilitated. The reference pixel is compared to a neighborhood of adjacent pixels located in vertical, horizontal, and two different diagonal directions. When the value of the reference pixel is greater than or equal that of the neighborhood pixels, the reference pixel is selected as an edge pixel.

4.2 Initializing templates

Initialization of the templates can proceed in one of two ways, depending on whether a priori knowledge about the shapes is available. In the first application, the objects in question are human cell nuclei, which are more or less elliptical in shape. Therefore we can initialize a set of elliptical templates with different aspect ratios. Since the algorithm is invariant to size, orientation, and reflection, only one template with each aspect ratio is created. The initial templates are created automatically but could easily be drawn by the user. In the second application, we assume no knowledge of the shapes, so the initial templates are created by the user outlining the first few objects with a mouse.

4.3 Detecting and segmenting objects

For object detection, we use the IGHT algorithm which incorporates a flexible template with an averaged shape and an uncertainty region to an iterative approach by voting in every position in the uncertainty region (Lee and Street 2000c). Suppose that (f_x, f_y) is a flexible point in an uncertainty region and is a relative x–y coordinate from the point on the template by translating the polar coordinate. When each edge point in the image votes for the positions that could correspond to r_θ of the particular template, the corresponding flexible points vote proportionally to the certainty of the flexible points. We define the certainty of the flexible point to the template $\vec{v}$ as

$$C(f_x, f_y) = \max\left(1 - \frac{D}{\left|\mathrm{ur}(\vec{v})\right|}, 0\right),$$

where $C(f_x, f_y)$ is between 0 and 1, and D is the Euclidean distance between the point on the template and the flexible point in the uncertainty region.

The algorithm proceeds as follows. Given an image and a set of templates, a global accumulator $\mathrm{G}(x, y)$ is initialized. For each scale (S) and orientation (Θ) of each template, a local accumulator $\mathrm{L}(x, y)$ is initialized, and the IGHT algorithm shown in Algorithm 2 is performed for every edge point $(x_\mathrm{e}, y_\mathrm{e})$ in the image.

During the clustering of shapes in the training set, the algorithm defines an uncertainty region on an average shape. Using this information, the proposed algorithm updates first an average point and then the corresponding flexible points.

For each new image, the IGHT is performed using the existing set of templates. The highest values in the global accumulator are then found to define the location of an object and the shape of the template $\vec{v}$ that closely matches the object. When a high peak point is selected, it is usually located at the top of a plateau. Thus the neighbors of the peak point are also

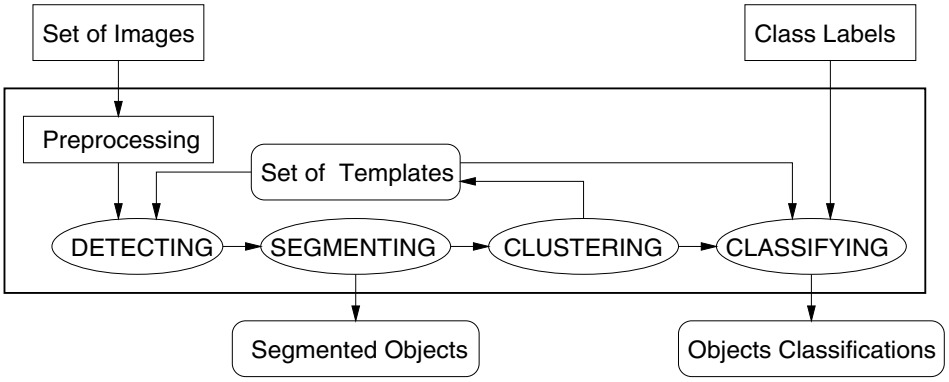
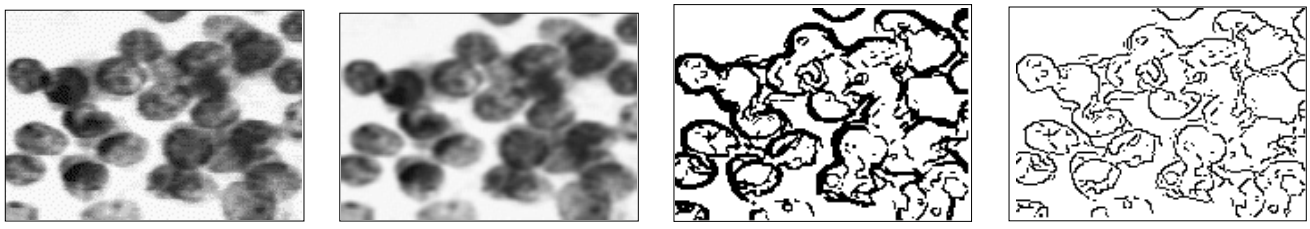


Fig. 3. Flowchart of the automatic system



a **b** **c** **d**

Fig. 4a–d. Preprocessing of an image: **a** original; **b** smoothed; **c** edges detected; **d** edges thinned

for each radius r_i, $1 \leq i \leq n$, of the template

Compute a voted position (x_i', y_i')

$(x_i', y_i') = (x_e, y_e) + (Sr_i \cos(\Theta + \theta_i), Sr_i \sin(\Theta + \theta_i))$, where θ_i is $\frac{360}{n} \times i$

Update the value in the local accumulator L by:

$\mathsf{L}(x_i', y_i') = \mathsf{L}(x_i', y_i') + |\mathsf{g}(x_e, y_e)|$, where $\mathsf{g}(x_e, y_e)$ is the gradient value at the edge point

for each flexible point (f_{x_l}, f_{y_l}), $1 \leq l \leq F_i$, where F_i is the number of flexible points at the radius r_i

Compute a voted position, (x_f, y_f), of the flexible point:

$(x_f, y_f) = (x_i', y_i') + (f_{x_l}, f_{y_l})$

Update the position in the local accumulator:

$\mathsf{L}(x_f, y_f) = \mathsf{L}(x_f, y_f) + C(f_{x_l}, f_{y_l}) \times |\mathsf{g}(x_e, y_e)|$

Algorithm 2. Iterative GHT using flexible templates

high values. These high-valued neighbors may be selected as high peak points in the next selection, resulting in the selection of templates for one object. In order to avoid this situation, the appropriate template $\bar{\vec{v}}$ is overlaid onto the image at the appropriate rotation angle, and the area of the global accumulator that is included in the template is eliminated. This process is repeated until a user-defined number of objects has been located, or until the match scores degrade below an adjustable threshold.

Although we are trying to match templates to objects, there is no object that perfectly matches the templates. Therefore, plateaus sometimes occur near the center of a poorly matched object. These plateaus cause difficulties when choosing the highest peak point. When a plateau appears, we choose the center of the plateau as the highest value. To achieve this, peak sharpening is performed as preprocessing for the peak-finding step, which increases the value at the center of the plateaus.

An important aspect of the proposed method is the use of probabilistic searches to guide the order in which template matches are attempted. As the images are processed, the program records the number of times each template has matched an object. This allows us to search first for the object shapes that were most common in previous images. By searching first for the objects that we are most likely to find, we significantly reduce the expected time required, since not all templates will appear in every image. A user might also choose to search only for objects with a particular classification. This is discussed in Sect. 4.5.

A snake is then initialized for each object using the template points, and runs to convergence. After an object has been correctly outlined by a snake, this segmented shape is considered to be best-matched with the template $\bar{\vec{v}}$.

4.4 Clustering shapes

A shape $\vec{v}$ is considered similar to another $\vec{u}$ if and only if $\vec{v}$ and $\vec{u}$ differ by no more than a small value ϵ. Formally, the difference between two shapes can be defined as follows:

$$d_p(\vec{v}, \vec{u}) = \|\vec{v} - \vec{u}\|_p,$$

where $\|\cdot\|_p$ is, for example, the Manhattan (p=1), Euclidean (p=2), or max (p=∞) norm. In our experiments, we use the Manhattan norm. In other words, two shapes are considered similar if

$$d(\vec{v}, \vec{u}) = \sum |v_i - u_i| < \epsilon, \qquad (1)$$

where v_i and u_i are points (or radii, $r\vec{v}_i$ and $r\vec{u}_i$) in the two shapes. To avoid simple orientation differences, the comparison is performed at 16 different rotations, and the final distance is defined to be the minimum of these distances. The similarity threshold, ϵ, is set empirically and can be adjusted by the user.

Many problems in data analysis – especially in signal and image processing – require the unsupervised grouping of data into a set of clusters or regions. The structural relationships between individual data points have to be detected in an unsupervised fashion. Figure 5 shows the clustering process graphically. In our on-line shape learning, we use a modified difference by considering uncertainty regions. The difference between a new shape $\vec{u}$ and a template $\bar{\vec{v}}$ can be calculated as follows:

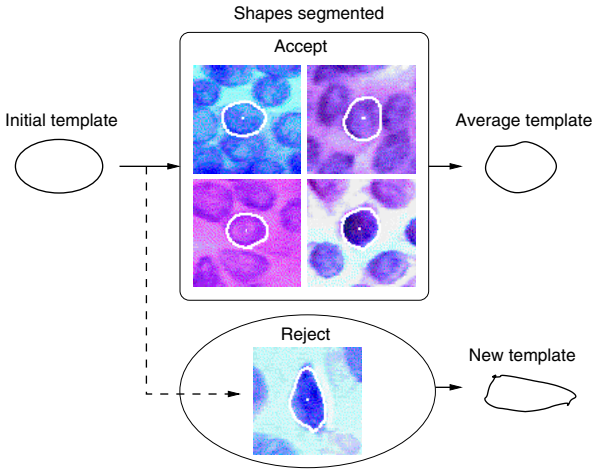


Fig. 5. The clustering of shapes

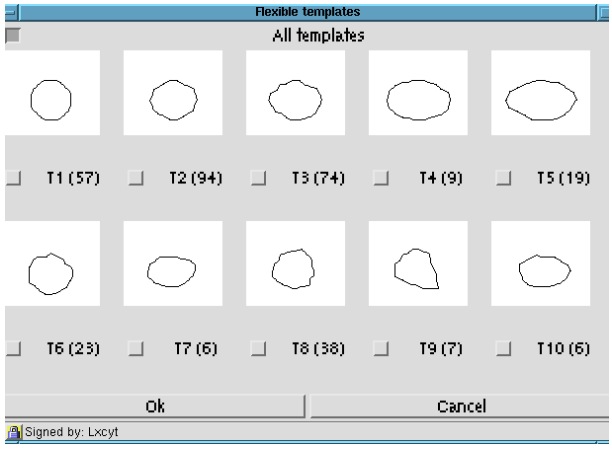


Fig. 6. Learned set of cell templates

$$d(\bar{\vec{v}}, \vec{u}) = \sum s_i \, |\bar{v}_i - u_i|$$
$$s_i = \frac{2\,|\bar{v}_i - u_i|}{|ur(\bar{v}_i)|}.$$

If u_i is in the uncertainty region of $\bar{\vec{v}}$, s_i is between 0 to 1 and the distance is scaled down. If u_i is out of the uncertainty region of $\bar{\vec{v}}$, s_i is larger than 1, increasing the measured distance. If there are several similar templates, the template with the minimum difference is selected.

If the shape $\vec{u}$ is most similar to the template $\bar{\vec{v}}$ and the distance is less than ϵ, $\vec{u}$ is added to the cluster of shapes represented by that template and the template $\bar{\vec{v}}$ is updated using $\vec{u}$, to make a new template $\bar{\vec{v}}'$:

$$\bar{\vec{v}}' = \frac{N\bar{\vec{v}} + \vec{u}}{N+1}.$$

The standard deviation of the new template, $\bar{\mathbf{v}}'$, is

$$\sigma(\bar{\vec{v}}) = \sqrt{\frac{N\vec{w} + (\bar{\vec{v}}' - \vec{u})^2}{N+1}},$$

where $\vec{w} = \sigma^2(\bar{\vec{v}}) + (\bar{\vec{v}}' - \bar{\vec{v}})^2$.

If the new shape is not similar to any templates, a new template is created with the new shape and the number of clusters, k, is increased. Thus, after training on a collection of

Table 1. Benign/malignant counts for various template scales and shapes

Scale	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6
Shape									
T1	4/0	10/1	19/4	0/0	0/14	0/5	0/0	0/0	0/0
T2	4/0	29/0	19/18	0/0	0/12	0/8	0/0	0/2	0/2
T3	6/1	28/8	5/16	0/0	0/8	0/2	0/0	0/0	0/0
T4	0/3	0/4	0/1	0/0	0/1	0/0	0/0	0/0	0/0
T5	3/8	0/6	0/0	1/1	0/0	0/0	0/0	0/0	0/0

shapes, each template represents a cluster of shapes that are nearby one another in the shape space.

4.5 Classifying objects

The ability to classify objects in an image plays an important role in the projected applications of this system. Consider the problem of isolating cell nuclei from heterogeneous tissue for dissection and genetic analysis. Such tissue may contain both diseased and healthy cells in the same sample. However, molecular analysis may depend on collecting a "clean" sample of all diseased or all healthy cells for comparison purposes. Further, the user should not be expected to wait for the system to locate objects that are not desired for the particular experiment.

Cellular morphometry is often used to diagnose diseases such as cancer. For instance, the Xcyt system was originally designed to diagnose breast tissue as benign or malignant based on derived nuclear features such as area, perimeter, and smoothness (Wolberg et al. 1994). We therefore take the approach that size and shape information already gathered in the clusters may be able to solve classification problems.

The classification method proceeds as follows. As objects are isolated by the user, they may be given a class label. This requires a training phase in which an expert user is available. A count is maintained of the number of objects from each class that are represented in each cluster. These counts are stored based on the scale factor; for instance, Table 1 shows the scales and counts for the first five templates in Fig. 6 for experiment 1. There are two numbers in each entry: the first one is the number of benign cells matching the template, and the second one is the number of malignant cells.

As the training proceeds, the system is able to classify new objects based on the majority class of the matching template. We saved counts by shape, size and the corresponding class. As Table 1 shows, size is a more important factor than shape for cell classification. However, it is very flexible depending on the application. This simple instance-based learning scheme further improves the speed of the detection algorithm in cases where only a particular class of object is desired. For instance, if a user wants only malignant cells, the template search procedure will order the templates based on the probability of malignancy, rather than on the raw count of matched cells. Templates that match mostly benign cells will not be used until later in the search, even if their shapes were more common in previous images.

Figure 7 shows examples of cell segmentation and classification. The fifth cell in the second row is segmented by T2 with size 1.8, and so is classified as a malignant cell according to Table 1. If the matching template has no trained instances, for example T3 with size 1.6, or has the same number of instances for multiple classes, for example T5 with size 1.6, we consider the object to be "undefined". After a few training images in our experiments, the "undefined" class happens only rarely.

Table 2. Intercluster distance between templates

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
T1	0	20.36	34.6	55.76	70.07	41.57	43.57	54.6	49.3	51.5
T2		0	22.6	39.68	54.28	49.59	52.27	60.53	59.61	53.55
T3			0	23.52	35.96	65.26	66.95	70.93	73.87	56.75
T4				0	26.26	85.33	88.57	88.29	93.84	76.9
T5					0	96	101.2	98.85	104.67	90.44
T6						0	38.44	26.79	36.96	37.83
T7							0	35.79	35.03	50.84
T8								0	35.85	37.83
T9									0	52.55

Table 3. Intracluster distance between a template and shapes in its corresponding cluster

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
17.67	20	21.22	16	29	25.98	25.61	24.38	17.12	27.42

5 Experimental results

5.1 Cytological images

The algorithm was tested on cytological images from fine-needle biopsies of breast masses, the same images as those used to train the original Xcyt system.[1] The images are grayscaled with a spatial resolution of 640×480 pixels. These images are classified as benign or malignant on a per-sample basis; no classification is available for individual cell nuclei. Therefore in these results we consider all nuclei in benign images to be benign and all nuclei in malignant images to be malignant. This assumption is reasonable but not entirely accurate, making the classification problem particularly difficult due to classification noise. In particular, images from malignant samples almost certainly contain some benign cells.

Figure 7 shows the example set of templates constructed after training 20 images. To estimate how compact and well separated the clusters are, we calculated two distances (Davies and Bouldin 1979): intercluster distance among templates (Table 2), and intracluster distance between a template and objects in the corresponding cluster (Table 3). Most templates are well clustered, having an intercluster distance from any other templates that is larger than the intracluster distance. Since the intercluster distance (26.26) between T4 and T5 is smaller than the intradistance T5 (29), T5 is clustered so loosely that some objects in T5 are difficult to separate from T4.

[1] Some of these images are available at `ftp://dollar.biz.uiowa.edu/pub/street/images/`. This version of the Xcyt system can be found at `http://dollar.biz.uiowa.edu/~street/xcyt/xcyt.html`.

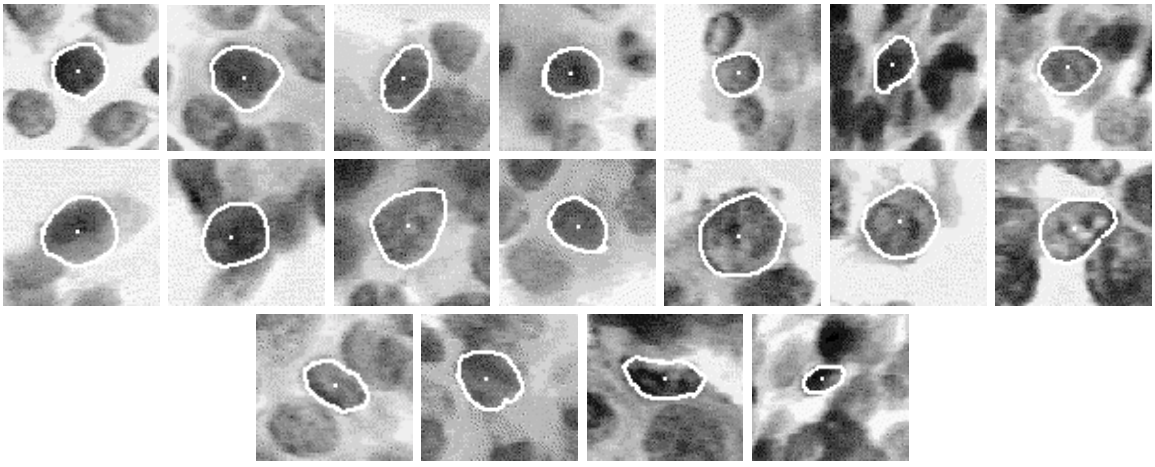


Fig. 7. Classification of the breast-cancer nuclei. The cells in each of the three rows belong to the same class (from *top* to *bottom*): benign, malignant, and undefined

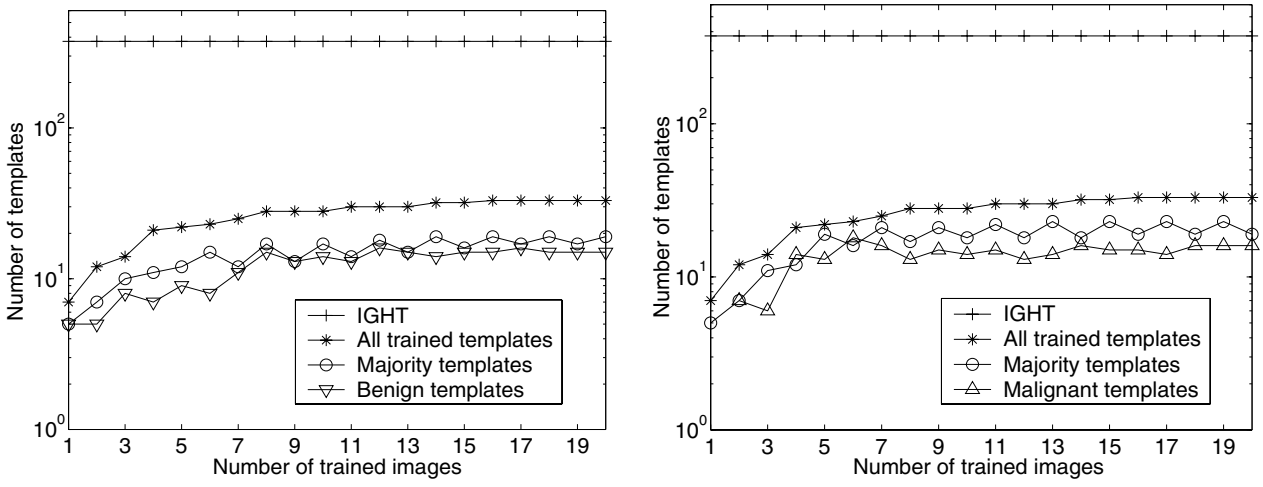
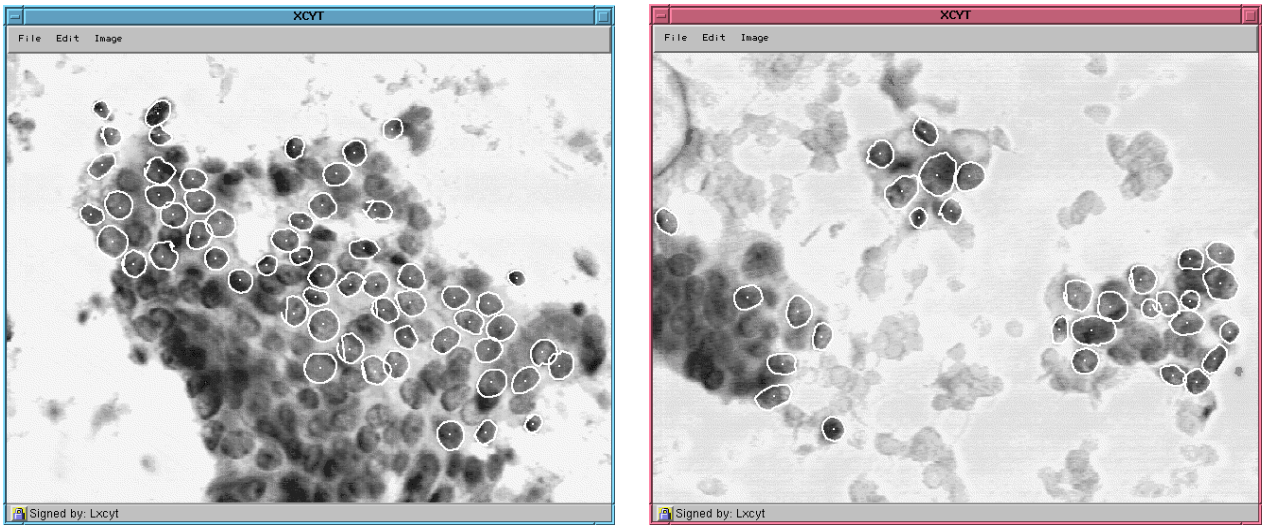


Fig. 8a,b. Number of templates searched: **a** benign images; **b** malignant images

The system was trained on a sequence of training images, alternating between benign and malignant. Figure 8 shows, on a logarithmic scale, the number of templates that had to be searched. The search was carried out in four different ways. The original IGHT algorithm was performed with a constant, predefined set of elliptical templates. These vary significantly in size and shape in an attempt to capture the basic shape of all nuclei that might be encountered. The learning system builds a set of templates as it is used; the curves labeled "all trained templates" show the number (N_c) of these trained shapes in Fig. 6 at all possible scales. After an initial ramping-up period, these numbers grow very slowly at a level approximately an order of magnitude below the original set of templates.

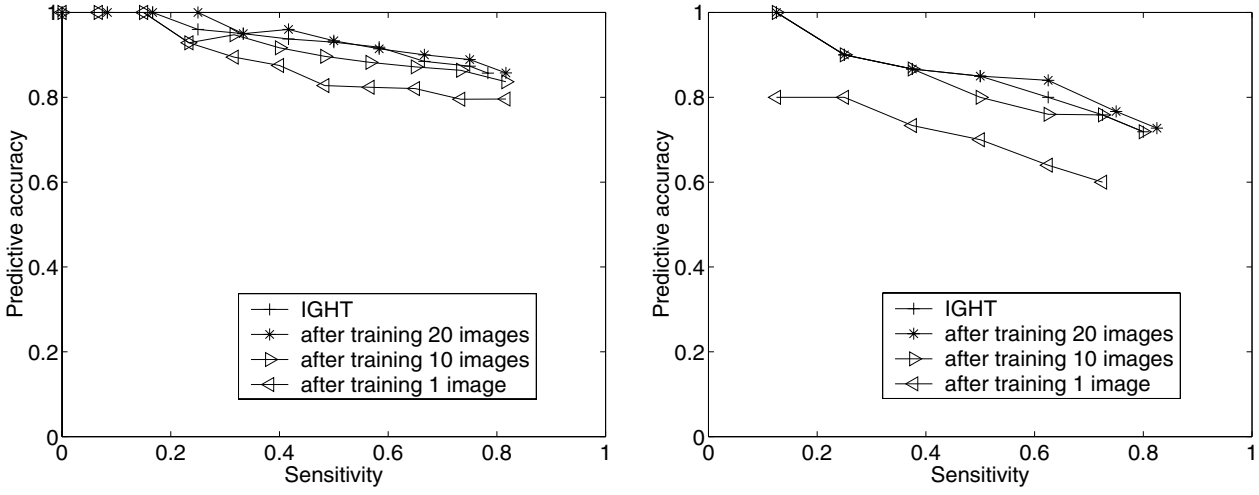
The curves labeled "majority templates" in Fig. 8 show the result of searching the templates in order of their frequency in previous images. After sorting the set of templates depending on the search method, we applied five templates at once. Whenever the result degrades, next five templates are used. The search is stopped when a result equivalent in segmentation accuracy to searching all the learned templates is reached. This results in a 50% reduction in the number of necessary templates. Finally, the templates were ordered based on their classification; e.g., templates with the highest probability of being benign were searched first on the benign test image. This reduces the necessary search time, indicating that the instance-based classification method is able to distinguish reasonably well between nuclei types, even though the ground-truth classifications in this problem were extremely noisy. This bodes well for future applications on heterogeneous tissue in which a trained expert will be available for the training phase. In addition, the user can reduce search time even further, by choosing specific templates or by choosing a preferred size.

Segmentation performance was measured as follows. Two test images (one benign, Fig. 9a, and one malignant, Fig. 9b) were set aside and used for segmenting and classifying automatically with all trained templates after training every image. We used two measures to evaluate the performance of the algorithm: sensitivity and predictive accuracy. Sensitivity is the likelihood that a nucleus will be detected when it has been marked by an expert. Predictive accuracy is defined as the likelihood that a template match is actually associated with a nucleus segmented by an expert. A well-segmented nucleus



a **b**

Fig. 9a,b. Test images (reduced by 50%) segmented using all trained templates after training 20 images: **a** benign; **b** malignant



a **b**

Fig. 10a,b. Segmentation performance: sensitivity and predictive accuracy for experiment 1. **a** benign images. **b** malignant images

is declared when the segmented shape is visually similar to the exact nucleus boundary. Thus, a well-segmented nucleus is assumed to have a shape accuracy of very close to 100%. Figure 10 shows that as the system is trained on more images, its ability to correctly segment the cells in the test images increases. In short, after a brief training phase, the new system has an accuracy that comparable to that of the original system, and it achieves this performance in an order-of-magnitude less time per image.

We tested ten more images – five benign and five malignant – with templates obtained after training 20 images. Since diagnosis and prognosis typically require the segmentation of a few dozen cells, we use the predictive accuracy of the segmentation method on the first 50 outlined nuclei (in the case of the benign images) or at 80% sensitivity (in the case of the malignant images). For 408 more cells in the ten images (250 cells in the five benign images and 158 cells in the five malignant images) – with all trained templates – we achieved 78.2% (319 cells) as an average segmentation correctness: 80.4% in the benign images and 74.7% in the malignant images.

Table 4. Confusion matrix for well-segmented cells on ten test images

True class	Predicted class	
	Benign	Malignant
Benign	175	26
Malignant	28	90

Classification accuracy was also measured on the two test images (Fig. 9) as the training proceeded. Figure 11 shows that as the system is trained on more and more images, its ability to correctly classify the nuclei in the test images increases. After a few images, the classification rate stabilizes at 88% for the benign image and 65% for the malignant image.

Table 4 shows the counts for well-segmented cells using the Table 1 templates obtained from 20 training images. For

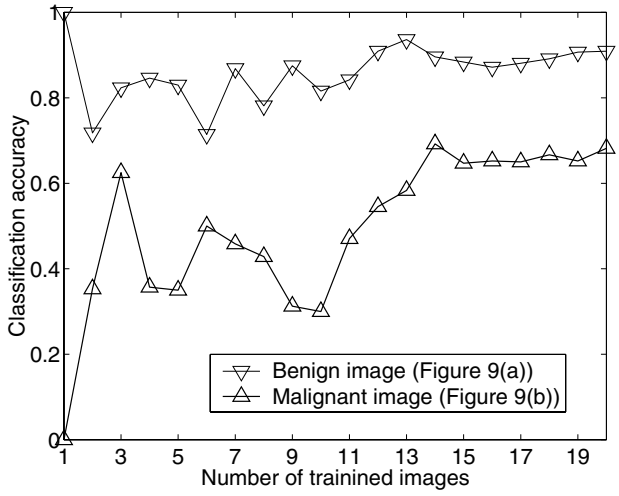


Fig. 11. Classification accuracy for experiment 1

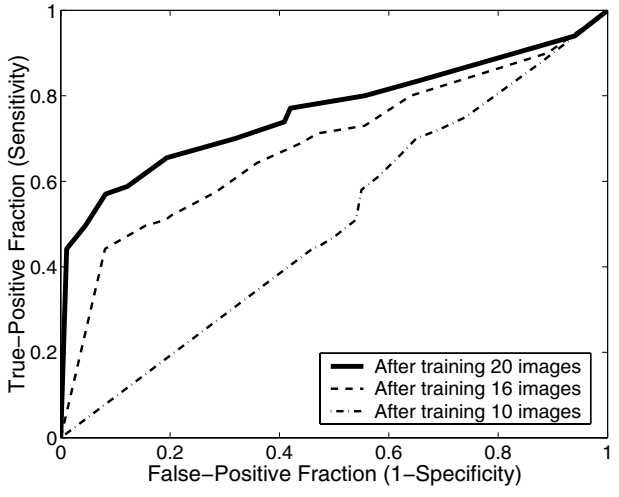


Fig. 12. Classification performance on ten test images: sensitivity and one-specificity for experiment 1

319 cells in the segmentation test of ten additional images, we achieved 83.1% as an average classification correctness: 87.1% in the 201 benign cells and 76.3% in the 118 malignant cells. Another measurement of classification performance is the ROC curve showing sensitivity and specificity. Sensitivity is the probability that a nucleus will be correctly classified as malignant, and specificity is the fraction of those without the disease correctly identified as benign. Figure 12 shows that as the system is trained on more images, its ability to correctly classify cells in the 10 test images increases. The area under the curve (AUC) estimates to ability of correctly classifying malignant cells from benign cells (Bradley 1997). The AUC after training 20 images (0.762) is larger than that after training 10 images (0.501) and that after training 16 images (0.685).

5.2 Vehicle images

The goal of this experiment is to extend the domain of the proposed system. The system for finding vehicles was trained

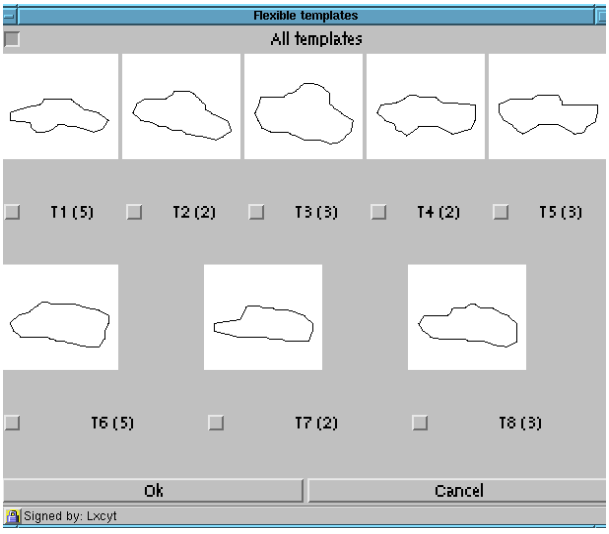


Fig. 13. Learned set of vehicle templates

on 25 color images. These images are classified as automobile, military tank, pickup track, van, and station wagon, and each class has five training images.

Figure 13 shows the set of vehicle templates, starting with no knowledge of the shapes. After training 25 images, there are two templates each for tanks (templates 2 and 3), pickup trucks (templates 4 and 5), and station wagons (templates 7 and 8). Since there were no heterogeneous templates, classification is unambiguous for all detected objects.

Figure 14 shows the segmentation results from 15 test images. The outlines of the vehicles were accurately identified, and 13 of the 15 vehicles were correctly classified. The fourth middle image was mistaken for a station wagon, because the contrast along the front side of the van is sharp and it forms a corner. The fifth bottom image was classified as a van because the template matched the shadow at the bottom of the station wagon instead of the edge along the bottom of the body of the van. The bottom row of Fig. 14 shows our system has reflection invariance.

6 Conclusions and future work

This paper presents a model-based system to aid the detection, segmentation, and classification of objects being outlined. The system can automatically or manually extract the shape of objects in the given images. The system demonstrates dramatic improvements in the time needed to perform template matching when compared to an exhaustive search, without sacrificing the quality of the result. The system can be initialized on-line, removing the requirement for a predefined set of templates. Finally, our results show that the size and shape features are sufficient for accurate object classification. The system provides useful user interfaces, including selection of a specific prototype for classification and selection of a size factor.

This paper also introduces an incremental learning scheme for large-scale image-analysis tasks. The extracted shapes are clustered by our on-line learning algorithm. The prototype of a cluster is calculated by averaging shapes in the cluster.

Fig. 14. Segmentation and classification of vehicle images

This learning approach was chosen for its simplicity and extensibility. Its primary function in this phase is to guide the template search. However, it is certainly plausible that future applications will require other features, such as chromaticity and second-order size and shape features, to classify the objects sufficiently well. These features can also be collected and stored for each cluster. At that point, a more sophisticated classification method such as an artificial neural network could be added to include these features in the classification.

In summary, we note that a major advantage of our on-line, model-based learning approach is the elimination of a separate training phase. Training is performed incrementally while the system is being used. Our learning approach improves object detection, by gathering a collection of desired objects specific to the application; it improves segmentation by generating higher quality initial outlines with different classes of objects. Further, it offers the user a variety of methods for reducing the search time of the template-matching step, which is a major goal of object detection.

As part of future work, flexible templates can be extended to handle more complex objects. Here we used the centroid-radii model for use with the GHT look-up table by dividing the circle into equal arcs. We can give more flexibility by using a pair (r_i, θ_i) instead of r_{θ_i}. However, as mentioned in Sect. 3.1, the use of the centroid-radii model means that the technique is unable to represent more sophisticated shapes. We will study more general shape models, with the requirement that they can be used as templates in the GHT.

In addition to detection, segmentation, and classification, this shape-learning approach can be applied to various imaging and vision applications, such as:

1. Segmenting objects from image sequences: The estimates obtained at time t can be integrated in a flexible template as a priori knowledge for the segmentation at time $t+1$. The flexible template can be updated on-line from the shapes estimated previously in the image sequence. In addition, the updated flexible template can be exploited for the segmentation of new shapes in the current image sequence.
2. Content-based image retrieval by indexing shapes: In indexing, our learning method can be used to cluster and classify the incoming examples. Learned templates can be used to search for objects in the images with match scores used as the index key. In retrieval, a given input query shape can be compared with the templates to find the best match. After finding the template that is most similar to the shape that a user wants, it can retrieve objects having the same template.

Acknowledgements. This work was partially supported by NIH grant CA64339-04 and NSF grant IIS-99-96044. The authors would like to thank Dr. William Wolberg of University of Wisconsin Department of Surgery for his helpful suggestions and for providing the test images for the medical task.

References

Ansari N, Delp EJ (1990) Partial shape recognition: a landmark based approach. IEEE Trans Pattern Anal Mach Intell 12:470–483

Ballard DH (1981) Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn 13:111–122

Beymer D, Poggio T (1996) Image representations for visual learning. Science 272:1905–1909

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30:1145–1159

Chang CC, Hwang SM, Buehrer DJ (1991) A shape recognition scheme based on relative distances of feature points from the centroid. Pattern Recogn 24:1053–1063

Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 1:224–227

Draper B (1997) Learning control strategies for object recognition. In: Ikeuchi K, Veloso M (eds) Symbolic visual learning. Oxford University Press, Oxford, pp 49–76

Dubuisson M-P, Lakshmanan S, Jain AK (1996) Vehicle segmentation and classification using deformable templates. IEEE Trans Pattern Anal Mach Intell 18:293–308

Duta N, Jain AK, Dubuisson-Jolly M-P (1999) Learning 2D shape models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colo., 23–25 June, pp 8–14

Gupta L, Sayeh MR, Tammana R (1990) A neural network approach to robust shape classification. Pattern Recogn 23:563–568

Hough PVC (1962) Method and means for recognizing complex patterns U.S. Patent 3069654

Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vis 1:321–331

Kassim AA, Tan T, Tan KH (1999) A comparative study of efficient generalised Hough transform techniques. Image Vis Comput 17:737–748

Khotanzad A, Hong YH (1990) Invariant image recognition by Zernike moments. IEEE Trans Pattern Anal Mach Intell 12:489–497

Lee K-M, Street WN (1999) A fast and robust approach for automated segmentation of breast cancer nuclei. In: Proceedings of the IASTED International Conference on Computer Graphics and Imaging, Palm Springs, October, pp 42–47

Lee K-M, Street WN (2000a) Automatic segmentation and classification using on-line shape learning. In: Proceedings of the Fifth IEEE Workshop on the Application of Computer Vision, Palm Springs, Calif., 4–6 December, pp 64–70

Lee K-M, Street WN (2000b) Dynamic learning of shapes for automatic object recognition. In: Proceedings of the 17th ICML Workshop on Machine Learning of Spatial Knowledge, Stanford University, Calif., 29 June–2 July, pp 44–49

Lee K-M, Street WN (2000c) A new approach of generalized Hough transform with flexible templates. In: Proceedings of the International Conference on Artificial Intelligence, Las Vegas, Nev., 26–29 June, pp 1133–1139

Maloof MA, Langley P, Binford T, Nevatia R (1998) Generalizing over aspect and location for rooftop detection. In: Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision, 19–21 October, pp 194–199

Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via linear programming. Oper Res 43:570–576

Mitzias DA, Mertzios BG (1994) Shape recognition with a neural classifier based on a fast polygon approximation technique. Pattern Recogn 27:627–636

Street WN (2000) Xcyt: A system for remote cytological diagnosis and prognosis of breast cancer. In: Jain LC (ed) Artificial intelligence techniques in breast cancer prognosis and diagnosis. World Scientific, Singapore, pp 297–322

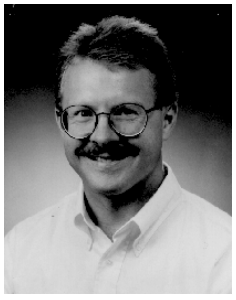
Suganthan PN, Teoh EK, Mital DP (1997) Optimal mapping of graph homomorphism onto a self-organizing Hopfield network. Image Vis Comput 15:679–694

Tabb K, George S, Adams R, Davey N (1999) Human shape recognition from snakes using neural networks. In: Proceedings of the Third International Conference on Computational Intelligence and Multimedia Applications, New Delhi, India, 23–26 September, pp 292–296

Wolberg WH, Street WN, Mangasarian OL (1994) Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. Cancer Lett 77:163–171

Wolberg WH, Street WN, Mangasarian OL (1999) A comparison of computer-based nuclear analysis versus lymph node status for staging breast cancer. Clin Cancer Res 5:3542–3548

Kyoung-Mi Lee received her Ph.D degree in computer sciences from the University of Iowa in 2001. She is currently a research assistant professor in the center for artificial vision research at Korea university. Her research interests include computer vision, image processing, image retrieval and mining, machine learning, neural network, and related applications.



W. Nick Street is an assistant professor in the Department of Management Sciences at the University of Iowa. His research interests include machine learning, data mining, computer vision, evolutionary systems, and bioinformatics. Street has a PhD in computer sciences from the University of Wisconsin-Madison. He is a member of IEEE, ACM SIGKDD, AAAI, and INFORMS.