Boston University Computer Science Tech. Report No. 2001-007,June 2001.

Retrieval by Shape Population: An Index Tree Approach

Lifeng Liu
MD OnLine
Lexington, MA 02421

Stan Sclaroff*
Computer Science Dept., Boston U.
Boston, MA 02215

Abstract

Based on our previous work in deformable shape model-based object detection, a new method is proposed that uses index trees for organizing shape features to support content-based retrieval applications. In the proposed strategy, different shape feature sets can be used in index trees constructed for object detection and shape similarity comparison respectively. There is a direct correspondence between the two shape feature sets. As a result, application-specific features can be obtained efficiently for shape-based retrieval after object detection. A novel approach is proposed that allows retrieval of images based on the population distribution of deformed shapes in each image. Experiments testing these new approaches have been conducted using an image database that contains blood cell micrographs. The precision vs. recall performance measure shows that our method is superior to previous methods.

1. Introduction

Image retrieval by content has become an important research area, and has applications to digital libraries and multimedia databases. One useful type of image query is an *object-based query,* where the shape of the object is an important similarity feature. Unfortunately, shape-based retrieval methods require accurate object detection and image segmentation, which are known to be difficult problems in computer vision. In great part, the difficulty is due to shape variation and deformation, illumination variation and shadows, as well as occlusions. Another issue in shape-based indexing is that of efficiency. If the computation time required for indexing (shape detection and segmentation) or retrieval (shape comparison and similarity ranking) is too large, then the method cannot be applied to image databases of any reasonable size.

In this paper, we describe a system for detection, segmentation, and indexing of deformable objects in images. The approach builds on an existing system for region-based segmentation via deformable templates [3]. To make the system practical for image database applications, we propose the use of segmentation *index trees*. An index tree is obtained via hierarchical clustering of a representative set of shape examples, called an instance set. Features that are good for object detection may not be ideal for similarity-based image retrieval; therefore, a second *retrieval index tree* is maintained. The features used in organizing the retrieval tree are chosen to suit application-specific requirements. Given efficient segmentation and retrieval trees, population-based image indexing and retrieval methods are proposed. The methods enable queries based on the distribution of shape statistics for the objects in each image. These methods have been implemented in an image database system, and tested in experiments with a cell micrograph database. Using standard precision vs. recall measures, the system performs markedly better when compared to previous shape histogram methods in the experiments.

*This work was supported in part through ONR grants N00014-96-1-0661 and N00014-01-1-0444, as well as NSF grants IIS-9624168 and EIA-9623865.

2. Related Work

In previous work, researchers have proposed global histogram methods that use shape features, e.g., relational histograms [13] or color correlograms [12]. Jain, et al. [15] introduced a method that combines color and shape features (edge directions). Global shape feature-based methods can be efficient for processing large databases, but they do not allow detailed descriptions of each object's shape if there are multiple objects in the same image.

In order to get shape information for the regions of interest, region-based querying approaches have been proposed [4, 10, 23]. Segmentation methods are employed to get homogeneous regions, and extract region shape features. Some region-based retrieval methods integrate color and texture with geometric information [27, 28]. Unfortunately, retrieval accuracy depends on the segmentation quality and region attributes used in indexing.

Another limitation of region-based methods is that spatial relationships between regions of interest are not considered. Such layout information is required for describing complex objects and their relation to each other in the image. Region-graph based representations can efficiently encode spatial relationships [16, 18, 20, 22, 26]. However, the region-graph approach cannot deal with shape deformation.

To solve the problem of shape deformation, deformable model-based querying methods have been proposed. These methods use either eigen-modes [9, 24], template matching [7, 3, 17, 21], or shape invariants [19]. Unfortunately, these methods assume that object/background segmentation information can be provided in advance, or that a computationally prohibitive shape matching algorithm should be used for detecting objects.

In [3], a deformable template-based method for automatic object detection, segmentation, and indexing was de-

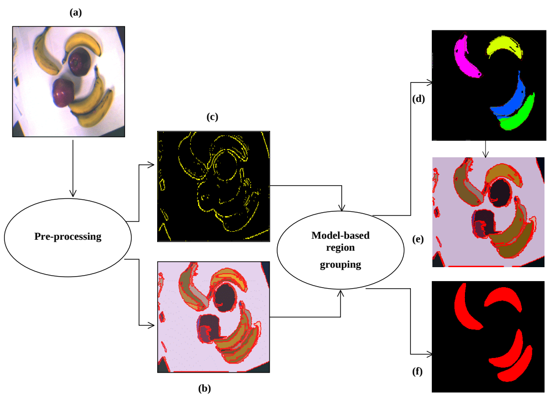

Figure 1: Shape detection and segmentation system diagram. Input color image (a) (image of bananas) undergoes pre-processing, which results in an over-segmentation (b) and an edge map (c). These are inputs to the model-based region grouping stage (using a banana template). The final output includes region groupings for detected objects (four bananas) (d), corresponding region merging (e), and recovered models for the objects (f).

scribed. The method had the benefit that it was automatic and it was robust to occlusions, illumination variation, and shadows. Unfortunately, the computational speed of the method was prohibitive for use in indexing large image databases. The use of an approximation method, the *index tree* yields a more practical system [2] while maintaining accuracy and robustness of the approach. This basic approach will form the foundation for shape category and population-based retrieval presented in this paper.

3. Background and Notation

In [3] we proposed a method that uses a deformable model to guide grouping of image regions. As shown in Fig. 1, system includes a pre-processing (over-segmentation and edge detection) stage, and a model-based region grouping stage. In the region grouping stage, the system tests various combinations of candidate region groupings to obtain an optimal labeling of the image. The shape model is deformed to match each grouping hypothesis $\mathbf{g}_i$ in such a way as to minimize a cost function:

$$E(\mathbf{g}_i) = \alpha E_{color} + (1-\alpha)\left[(1-\beta)E_{area} + \beta E_{deform}\right], \quad (1)$$

where α and β are scalar constants with values in the range $[0, 1]$ that control the relative importance of the three terms: E_{color} is a region color compatibility term for the region grouping, E_{area} is a region/model area overlap term, and E_{deform} is a deformation energy for the shape model.

Further, in order to test the quality of a possible partitioning, a global cost function for partitioning the whole image is defined:

$$\varepsilon = (1-\gamma)\sum_{i=1}^{n} r_i \mathbf{E}(\mathbf{g}_i) + \gamma \mathbf{n}, \quad (2)$$

where γ is a constant factor, $\mathbf{n}$ is the number of the groupings in the current image partitioning, r_i is the ratio of i^{th} group area to the total area of connected regions, and $\mathbf{E}(\mathbf{g}_i)$ is the cost function for the group $\mathbf{g}_i$ (Eq. 1). The highest confidence first (HCF) algorithm [5] is used to find an approximately optimal value for Eq. 2.

4. Index Trees

One problem with the system proposed above is that segmentation can be slow for images of moderate complexity. This is because the shape model fitting procedure must be invoked many times in order to get the cost values of different configurations. Although we utilize methods to speed up the fitting procedure, such as multi-resolution fitting, and caching deformation parameters, most of the CPU time (over 90%) is still used in model fitting. We therefore propose to use an *index tree* [2] method to accelerate the model fitting procedure.

The basic idea is as follows. We first generate many deformed instances of the object class by sampling in the deformation space according to the prior distribution of the deformation parameters. We then compute a shape feature vector for each generated instance. In our implementation, the features employed are the seven normalized central moments. The shape feature vector and the deformation parameters are stored with the instance. Then, in the fitting process, we compute the shape feature vector for a potential region group. By comparing the feature vectors, the shape most similar to the region group is fetched from the set of generated instances (called an *instance set*). Its associated deformation parameters are used as the parameters for the region group, or as a starting point to a refining process.

4.1. Index Tree Structure and Search

To speed up search, we organize the instances in a tree such that the retrieval time can be logarithmic to the number of instances. We use a hierarchical clustering method (minimum variance) to process the shape features of the instances, and get the tree structure [14]. In our experiments, we have used the cophenetic correlation coefficient (CPCC) [14] to validate clustering. Although we uniformly sample in the deformation space, there is indeed a hierarchical structure in the corresponding shape feature space.

By searching for the best match in the index tree, the searching time is reduced but it does not guarantee that the nearest match is always found. We tried to use the mean feature of instances in each non-leaf node to select a branch and go to the next level. However, the covariance and distribution for the instances in each node are not the same; furthermore, their distributions are not Gaussian in general. In order to overcome this problem, we use linear discriminant functions[8] at the non-leaf nodes. Our experiments verified that this method can increase the success rate in finding the nearest neighbors [2].

Another problem with index tree search is that the retrieved result is the nearest neighbor in the shape feature space. However, the distance metric in shape feature space is not the same as the fitting cost (Eq. 1) nor is it monotonic to the fitting cost. A neural network (NN) can be used to map from the difference in the shape feature space to the fitting cost measure. We use a three layer back-propagation network with bias terms and momentum [25]. The NN is only used for mapping in the leaf nodes of the index tree to reduce the on-line computation.

The index tree approach was tested on over one hundred cluttered images of objects taken from a number of different shape classes. It was observed that the CPU time needed for segmentation was decreased by one order of magnitude, while the number of errors in segmentation did not increase appreciably over HCF without index trees.

4.2. Unique Description and Retrieval Trees

To test our approach in an image retrieval application, we implemented a system that uses linear and quadratic polynomials to model stretching, shearing, and bending. These deformations are not independent. As a result, the recovered shape parameters cannot be guaranteed to be unique for the same shape. This problem of non-uniqueness pervades many other deformable shape description methods.

In general, recovering a unique shape description is a challenging problem. Using principal components analysis (PCA) for the recovered parameters can make sure that the coefficients obtained are unique in the coordinate space, but it does not guarantee that similar shapes will have similar parametric descriptions (especially for symmetric objects). For the same shape, there may be multiple parameter vectors that can describe it correctly.

To get around the non-uniqueness problem, we propose a two step approach. First, recover the shape description as described in the previous section. Then, use the recovered shape description to compute a second shape description (e.g., moment invariants[11], eigen-modes coefficients[24], etc. which are more application-specific). Since we use an index tree in segmentation, it is possible to pre-compute direct correspondence between the recovered shape parameters and the new shape feature vector. The model instances generated for the model segmentation index tree can be reorganized to form a second index tree based on the new shape feature vectors. As a result, for a query shape not inside the pre-generated model instance set, we can quickly

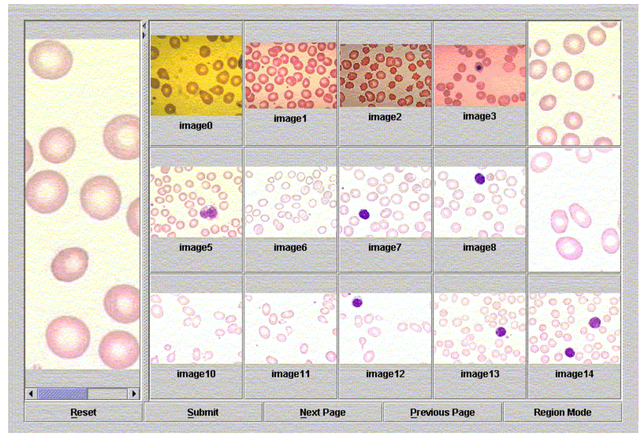

Figure 2: Interface for image retrieval system.

retrieve the instance that is the nearest neighbor (most similar shape) in the application-dependent feature space.

5. Retrieval by Shape Population

After processing the images in the database based on our object detection system, the recovered shape models of detected objects in each image are obtained. Based on this information, it becomes possible to extract the shape statistics in each image and use this for retrieval. Further, a population-based image query method can be implemented to satisfy shape retrieval for images that contain a particular population distribution of shapes.

5.1. Shape Population Statistics

In our model fitting formulation, the fitting cost value (Eq. 1) includes at term that measures the model's deformation from the mean shape. The distribution of fitting cost values can be used as one component in the shape population similarity measurement. The distribution of size variation can also be used in the similarity computation of the shape population. The model fitting cost values are obtained directly from the object detection stage, and the size variation information we used is a relative measurement. It is based on the ratio between each object's scale parameter and the mean scale parameter of the objects in the image.

In the shape population-based retrieval stage, for each image, the shape feature histogram is built according to the recovered shape models of the detected objects. The bins of the histogram correspond to the nodes of the specified level in the retrieval index tree. Depending on the number of bins required (i.e., the recognition resolution decided by the user), the level of the retrieval index tree corresponding to the bins can be decided.

The algorithm for building the histograms is as follows:

1. Select the histogram bins' number n according to the application accuracy requirement and the index tree structure.
2. Get the level of the index tree where there is n nodes at that level, assume it is level k.
3. For each image, initialize its shape histogram to be empty.
4. For each object shape detected in the image:
   (a) Get its corresponding shape model instance, assume it is model instance m.
   (b) From the index tree, get the leaf node which includes model instance m. Assume that this leaf node is node l. This can be done via lookup in a pre-generated table.
   (c) Get the ancestor node l' of leaf node l at level k, via fetching parent nodes of the leaf node and intermediate nodes or by looking in a pre-generated look-up table.
   (d) Increment by 1 the histogram bin which corresponds to node l'.

6. Implementation Details

There are three histograms computed for each image: one is based on the shape retrieval index tree, another is used to represent the distribution of fitting cost values, and the last is for the distribution of size variation. In computing histograms, there are 20 bins for the size component, 20 bins for the fitting cost component, and 100 bins for the shape feature vector component. We tested histogram intersection, chi-square statistic, and Bhattacharyya distance[18] as the histogram similarity metric respectively. In our experiments, we found that the results using the three different metrics are similar.

In our database retrieval system implementation, a graphical user interface is provided for selection of the object of interest in images, as show in Fig. 2. Retrieved images from the database can be shown in order of similarity from the selected object. In addition, query distance can be made based on shape population similarity to the query image. This work includes off-line processing, which makes use of our object detection algorithm to detect objects in images and get the model description for each detected object. Therefore, each image has an associated meta-data file for model description. In the on-line retrieval stage, the user can get a response quickly.

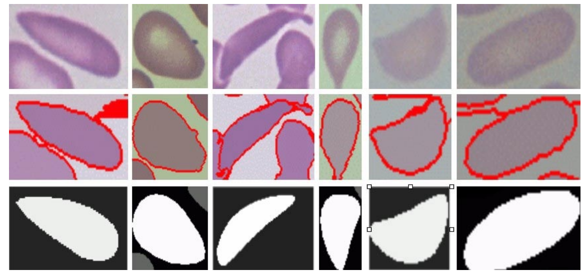

Figure 3: Detected cells and recovered models for different type of red blood cells. The first row shows the cells in the original image, the second row shows the segmentation, and the third row shows the recovered model.

7. Experimental Evaluation

As an example application, we tested the image retrieval strategy using a database of blood cell micrographs. Cell segmentation is an important and challenging task in medical image processing. For example, in hematology, blood cell counting and cell morphology evaluation are indices for certain pathological diagnoses. Images obtained under microscopes or electron microscopes include lots of objects. It is tedious for people to search all the images to get the ones of interest. Our system makes it possible to achieve this automatically.

However, there are some additional problems in cell image segmentation caused by cell attachments, morphological variation, occlusions, the presence of faults, artifacts, etc. Some limitations of the previous methods include: only considering each cell separately, using rigid models, and requiring touching cells to be dissimilar[30]. In addition, some methods require user input for initialization[1, 6], and other methods can only handle non-overlapped cells with smooth boundaries or contours[29].

In Fig. 3, we show examples of cells and their models recovered using our algorithm. As shown, the global deformation description can represent the shape variation precisely in many cases, such as normal cells, ovalocytes, helmet cells, sickle cells, teardrop cells, etc. Example segmentation results for some micrographs of blood cells are shown in Fig. 4. It shows that our automatic method can handle multiple touching objects within the same image and can handle small amounts of overlap as well.

7.1. Population-Based Retrieval

To evaluate population-based retrieval, the experimental setup could be as follows: split each original image into several sub-images, and see whether the sub-images from the same original image can be retrieved. However, since most of the cell images in our database do not have uniform shape populations, the shape population distributions

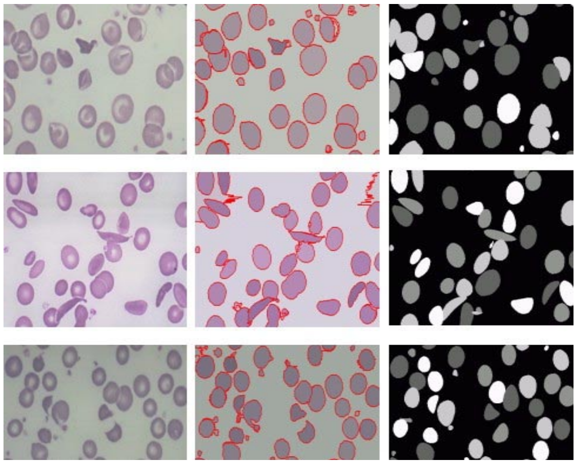

Figure 4: Example segmentation and models for cell micrographs. Each row shows the micrograph, the result after model-based region grouping, followed by the recovered shape models.

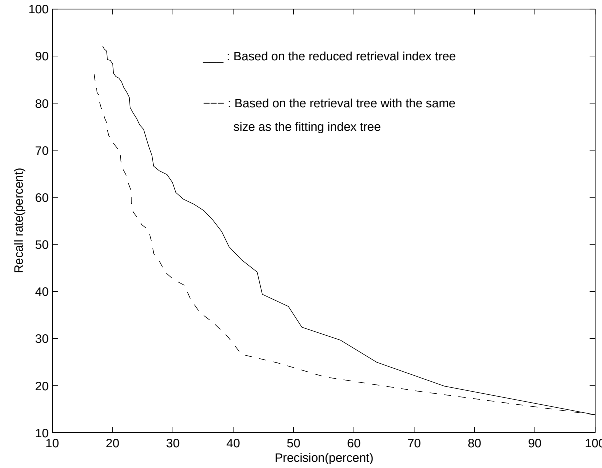

Figure 5: Precision versus recall for population-based retrieval.

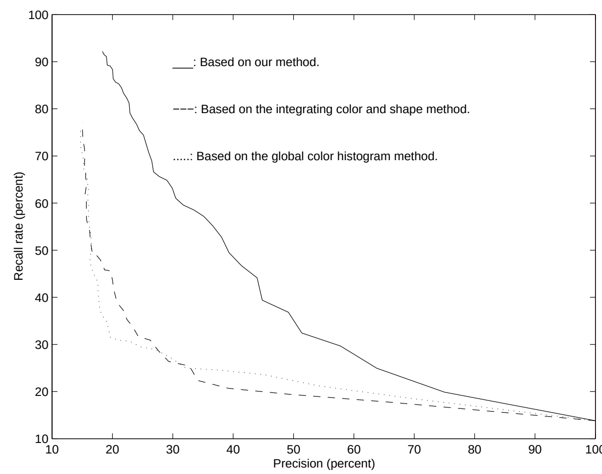

Figure 6: Comparison of precision versus recall rate.

of sub-images from the same original image are not similar enough. Therefore, this kind of validation cannot give an accurate measurement of our system.

Instead, we utilized a strategy based on the classification of images into clusters, i.e., category search. For ground truth, we manually classified images into clusters with similar shape populations. For example, in one cluster, the dominant shapes are ellipses, in another cluster, the dominant cells are normal cells, etc. We built a database that includes about sixty images, and assigned them to eight clusters. Precision and recall were defined based on evaluating how many of the retrieved images came from the same cluster as the query image.

The shape population similarity measure employed in this experiment is the simple sum of three histogram similarity components. If users are only interested in querying objects with similar shapes, then the distance in the application-dependent shape feature space can be used as the similarity metric.

One straightforward option in building the retrieval index tree is to use the same sample set as the model fitting index tree. An alternative is to use only the samples which occur in the image database for retrieval, which we call the reduced retrieval index tree. A graph of precision versus recall for both the retrieval tree and the reduced retrieval are shown in Fig. 5. The upper curve corresponds to using the reduced retrieval index tree, and the lower curve corresponds to using the retrieval tree with the same size as the fitting index tree. As can be seen in the graph, using only the model instances that occurred in the database to build the retrieval index tree improved the retrieval result.

Fig. 6 shows the performance comparison between our method (solid line), a global color histogram method (dotted line), and the color and shape method of Jain, et al. [15] (dashed line). In our implementation Jain's method, the bin number for each color component is 32 (when we changed the bin number to be 64, the result was similar), the bin number for edge direction is 32, and the weights for color and shape are the same. As can be seen in the graph, by including shape information our method out-performs the two other methods.

Table 1 shows the average recall rates with respect to the number of retrieved images. Our method was superior to the global color histogram method and the integration of color and shape method for this experimental data set.

	The number of retrieved images is k times of the cluster size			
	k=1	k=2	k=3	k=4
Retrieval index tree with the same size as the fitting index tree	34.29	50.94	65.76	78.71
Reduced retrieval index tree	43.57	61.44	73.69	86.02
Global color histogram	27.83	39.57	50.45	64.48
Method of integrating color and shape	28.57	40.29	51.73	66.96

Table 1: Shape population-based retrieval accuracy on the classified database (recall rate in percent).

8. Conclusion

We proposed a method to use index tree for organizing shape features to image retrieval applications. Different shape parameter sets can be used in indexing (shape detection, segmentation, and description) and shape retrieval (shape comparison and similarity ranking). This overcomes problems with non-uniqueness of the recovered shape deformation parameters, and follows the observation that features which work well in recognition may not work well in similarity comparison. Hierarchical clustering methods were used for feature space partitioning during construction of the index trees.

Shape similarity based on clustering was used for shape population retrieval. A direct mapping is built between the two index trees for efficiency (the fitting index tree and the retrieval index tree). We conducted experiments for object detection and shape-based retrieval for an image database of blood cell micrographs. The precision/recall performance evaluation indicates that our method is superior to the method of Jain [15] in this application, and better suited for object-based retrieval.

References

[1] P.S. Umesh Adiga and et. al. Semi-automatic segmentation of tissue cells from confocal microscope images. *ICPR*, 1996.

[2] Anonymous.

[3] Anonymous.

[4] C. Carson. Region-based image querying. *CBAIVL*, 1997.

[5] P. B. Chou and C. M. Brown. The theory and practice of bayesian image labeling. *IJCV*, 4:185–210, 1990.

[6] G. Cong and B. Parvin. Model based segmentation of nuclei. *CVPR*, 1999.

[7] A. DelBimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *PAMI*, 19(2):121–132, 1997.

[8] R.O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[9] B. Gunsel and A.M. Tekalp. Shape similarity matching for query by example. *PR*, 31(7):931–944, 1998.

[10] N.R. Howe. Percentile blobs for image similarity. In *CBAIVL*, 1998.

[11] M.K. Hu. Visual pattern recognition by moment invariants. *IEEE Trans. on Information Theory*, 8(2):179–187, 1962.

[12] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image indexing using color correlograms. *CVPR*, 1997.

[13] B. Huet and E.R. Hancock. Relational histograms for shape indexing. *ICCV*, 1998.

[14] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[15] A.K. Jain and A. Vailaya. Image retrieval using color and shape. *PR*, 29(8):1233–1244, 1996.

[16] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing. *CVPR*, 1997.

[17] T. Liu and D. Geiger. Approximate tree matching and shape similarity. *ICCV*, 1999.

[18] B. Moghaddam, H. Biermann, and D. Margaritis. Defining image content with multiple regions-of-interest. *CBAIVL*, 1999.

[19] F. Mokhtarian and S. Abbasi. Retrieval of similar shapes under affine transformation. *Proc. Visual 99*, 1999.

[20] B. Ozer, W. Wolf, and A.N. Akansu. A graph based object description for information retrieval in digital image and video libraries. *CBAIVL*, 1999.

[21] P. Pala and S. Santini. Image retrieval by shape and texture. *PR*, 32(3), 1999.

[22] I.K. Park, I.D. Yun, and S.U. Lee. Color image retrieval using hybrid graph representation. *IVC*, 17(7):465–474, 1999.

[23] E.J. Pauwels and G. Frederix. Finding salient regions in images. *CVIU*, 75(1/2):73–85, 1999.

[24] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for Content-Based Manipulation of Image Databases *IJCV*, 18(3), 1996.

[25] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. *Learning Internal Representations by Error Propagation*. Parallel Distributed Proc. Vol. 1, MIT Press, 1986.

[26] J.R. Smith and C.S. Li. Image classification and querying using composite region templates. *CVIU*, 75(1/2):165–174, 1999.

[27] A. Soffer and H. Samet. Using negative shape features for logo similarity matching. *ICPR*, 1998.

[28] S. Tanimoto, S. Levialdi, F. Lecca, and L. Cinque. Retrieval of images using rich region descriptions. *ICPR*, 1998.

[29] H.S. Wu and et. al. Optimal segmentation of cell images. *VISP*, 145(1):50–56, 1998.

[30] P. Zhou and D. Pycock. Robust statistical-models for cell image interpretation. *IVC*, 15(4):307–316, 1997.