Shape Indexing in Digital Media

Joel Kemp

May 20, 2011

Introduction

The shape indexing of images has been a very difficult problem for decades. The importance of this area lies in the recognition and classification of objects amongst large collections of digital media. The Human Vision System (HVS) recognizes objects based on shape analysis. Despite concrete knowledge on how the brain analyzes shapes, there have been many approaches to the shape indexing of digital media problem over the last 30 years.

The most successful (in terms of accuracy) implementation to the shape-indexing problem, proposed by [1], has achieved a recognition rate of 91% about the popular MPEG7 CE Shape dataset. The dataset is composed of 70 groups of binary images with 20 images per group. Within a group, each image contains a deformation of a central object (ex: apple, bat, camel, etc).

The method in [1] utilizes a computationally expensive (cubic-time) feature generation algorithm to index an image by a matrix of inner (geodesic) distances that can then be used for recognition. The benefit to such a representation is its invariance to contour deformations. In addition, the details/articulation points (example: a cat's ears, a dog's tail, etc) of an object are retained in the representation. A cubic ($O(n^3)$) runtime complexity, however, renders this method impractical for near real-time indexing and recognition.

Our efforts in this experiment are not to match the accuracy measure over the entire dataset, but to focus on the subproblem of using clustering to retain articulation points without the expensive inner-distance shape context. Such points are important because they allow for a more detailed analysis of the images. For example, a cat's head without its ears looks a lot like a dog's head without its ears; both of those images now resemble an apple – which is an incorrect match. If such a segmentation scheme could be devised, then the next step would be to explore feature generation/extraction of each cluster. Hence, we would analyze an image by its parts. Recognition would then be achieved by the analysis of shapes across pairs of images in the dataset.

The experiments conducted as part of this final project attempt to enhance (fix) an existing system [5] created in collaboration with Professor Jie Wei (CCNY). The system provides a fast feature generation algorithm that looks at both global and local features of clusters generated from our experiments.

Empirical studies with various segmentations schemes have been explored in this project to determine a scheme that retains articulations and is closest to the ground truth (human vision system based clustering). Several clustering algorithms were explored: K-means, Gaussian Mixture Models, Morphological decomposition with maximal disks [4, 3], Morphological decomposition with K-means reconstruction, and Watershed segmentation [2].

Clustering Experiment

The goal of the clustering experiment was to try different segmentation algorithms, to observe which solutions would yield "correct" clusters. Of course, this is a very hard problem, as the judgement of a usable/correct set of clusters depends on the human vision system. To aid the algorithms, we provided a vector of positive integers corresponding to the ideal number of clusters to look for during segmentation. This reduces the set of erroneous results.

The criteria for our ideal segmentation algorithm includes several points. Such an algorithm should be fast, reliable/consistent across many trails, retain articulation points, and produce a realistic representation of an image's internal components. Unfortunately, it is not guaranteed that such the results of an ideal segmentation method will yield state-of-the-art recognition results when used in the system as a whole; however, the theoretical payoff is quite promising.

K-means

The K-means algorithm is popular for its ease of implementation, linear runtime complexity, and its applications to many different types of data. For this experiment, the K-means algorithm was applied across all of the images in the dataset. This segmentation scheme has several of the desirable qualities for our purposes: fast, retains articulation points, and produces moderately useable clusters. The main drawback to K-means is that there are many random components within the algorithm. This randomness produces inconsistent results across many trails.

For each image, we supplied the ideal number of clusters K to the algorithm; however, K-means had no information regarding where the clusters might be. In turn, the centroid (cluster center) locations are randomly generated and modified at each iteration of the algorithm until the centroids no longer move. The usefulness of the resulting clusters heavily depends on a good starting location for the centroids.

Several runs of the K-means algorithm with a user-supplied K yield different results for each run. This is quite unreliable and leads to the conclusion that K-means (alone) is not suitable for generating an image's internal representation.

Gaussian Mixture Models

Gaussian Mixture Models (GMM) allow us to model an image as a collection of K gaussians. The algorithm, at it's heart, is K-means with an internal probabilistic analysis to govern the changing membership of data points in clusters. The GMM segmentation was applied to the entire dataset and observed for its resulting cluster utility.

The general assumption for GMM is that the data being modeled actually resembles a collection of gaussians. Due to the relatively rounded nature of the shapes in the dataset, GMM seemed applicable. However, the algorithm was also subject to the inconsistencies of K-means; several runs of GMM on the same image yielded different clusters each time. Differing clusters results in differing feature vectors – which, when thinking of the system as a whole (incorporating feature generation and similarity computation) could result in the system changing its decision of similarity between two images. Again, this is an undesirable quality.

Watershed Segmentation

Watershed Segmentation is a popular segmentation algorithm for grayscale images, though the algorithm still holds in the presence of binary images. The algorithm computes a gradient image indicating the direction of change about the pixels (i.e., the distance growth between foreground and background pixels). Using the gradient image, watershed segmentation simulates filling each regional minimum with water and creating barriers where the water from different regions coincide. Ultimately, this process results in a closed, skeletal-like segmentation of an image.

When applying the watershed algorithm about the entire dataset, it can be observed that each of the images results in 4 decompositions/clusters. Each of the clusters do contain rather unique shapes and the results are consistent across many trails; however, the results fail to satisfy the "realistic" criterion.

In order for the unrealistic and wildly-shaped watershed clusters to be usable for feature generation, features would have to be chosen that heavily emphasize a contour-based analysis. The unrealistic nature of the clusters results from the introduction of more (artificial) articulation points (sharper edges, elongated segments, and heavily curved contours).

Morphology with Maximal Disks

Morphological segmentation contains very desirable properties. Not only is the algorithm theoretically fast, but the clustering scheme intuitively results in rounded shape representations of an object's internal components. Compared to Watershed segmentation, the rounded clusters of morphological segmentation do not introduce radical, artificial articulation points. Unfortunately, as promising as morphology seems, there is little code available to perform segmentation. Hence, a custom segmentation scheme was built for the experiment.

Morphological segmentation is primarily based on 4 operators: Erosion, Dilation, Opening, and Closing. The most useful operator is opening for its ability to decompose an image into a rounded cluster. Intuitively, we take a small image/kernel (of a circular shape for our purposes) and return all of the area covered by this disk. Finding the appropriate radius for the disk is also a hard problem, and the system automates this by finding the maximal disk per iteration. The result is a rounded segment of the object. The found cluster/decomposition is then subtracted from the image, and the process is repeated until we obtain the empty image. This is known as morphological decomposition. There are many problems with this method.

Due to the digital/discrete nature of the images, image subtraction does not result in a clean removal of pixels. Subtraction results in artifacts/noise that heavily damages the clustering algorithm. The noise is regarded/interpreted as a separate connected component; this equates to the incorrect segmentation of not only one object, but perhaps multiple objects due to varying sizes of noisy clusters. To circumvent the damaging effects, we attempt to remove clusters that are smaller than a certain threshold (area percentage relative to the original object).

For example, all components (found in any iteration of the segmentation scheme) whose area (normalized number of pixels) is less than 10% of the normalized area of the original object should be removed. Of course, a uniform thresholding is subject to inconsistencies in what should be deemed as noise in an image. The solution to the uniform thresholding problem involves defining a threshold vector/space that allows us to use a vector of possible threshold values for removal of noise.

Using the user-defined K number of clusters to look for, the algorithm removes the noisy clusters (where the threshold is a value of the threshold vector) and observes if the resulting number of clusters is close to (preferably equal) the provided K. If not, then we change the threshold value and try again. This process is an exhaustive search that results in slow performance and needs to be rethought.

Morphology with K-means Reconstruction

Using the parameteric/search version of morphological segmentation's (detailed in the previous section) results on the entire dataset, it was observed that the articulation points were lost due to the use of the circular kernel (more formally known as a structural element). Using the rounded clusters, recognition is severely impacted. It becomes rather difficult for the system to differentiate between the head of both a cat and dog and an apple, as discussed in the Introduction section.

The clusters were used in the feature generation and similarity computations by taking each image in the dataset and comparing it to the other 1399 images. Internally, of course, the comparison of two images was actually the comparison of the decompositions/clusters. The resulting accuracy measure was a disappointing 23%; we believe that this is mainly due to the loss of articulations in the segmentation process. This result sparked the experimentation with morphology and its possible use with K-means to reconstruct (i.e, regain articulation points) the image.

K-means reconstruction basically performs K-means clustering using two pieces of data: the ideal number of clusters K, and the centroid locations defined by the centroids of the morphological decompositions. With these two pieces of information, K-means now knows how many clusters to start with, and where to place the initial clusters. Convergence of the algorithm occurs and we end up with consistently good clusters.

By using morphological segmentation as a heuristic for finding the starting centroid locations, we guarantee that the application of K-means for reconstruction yields the same results across many trails. This is mainly due to the fact that maximal disks used in morphological segmentation will always be the resulting maximal disks for any run of the algorithm – given that our exhaustive thresholding process is also run consistently across many trails.

Results

Morphological segmentation with K-means reconstruction yields the best results for our search for an "ideal" clustering algorithm. In turn, the clustering results are very close to how a human might cluster the objects (the ground truth). These findings are shown in the figures following this discussion.

When looking at the figures, it must be noted that the results of K-means and morphological segmentation with K-means yield very similar results. Again, it must be stated that multiple runs of K-means will yield different clusterings. Hence, we cannot omit the usage of morphology.

Further tests will be conducted to see if the results of morphological segmentation with k-means clustering will yield promising accuracy measures for the system as a whole.

References

[1] X. Bai. Learning context sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):861 – 874, 2009.

[2] Q. Chen. Watershed segmentation for binary images with different distance transforms. In *The 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications, 2004. HAVE 2004. Proceedings.*, 2004.

[3] R. M. Haralick. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4), 1987.

[4] J. Reinhardt. Comparison between the morphological skeleton and morphological shape decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9), 1996.

[5] J. Wei. Shape indexing and recognition based on regional analysis. *IEEE Transactions on Multimedia*, 9(5):1049 – 1061, 2007.

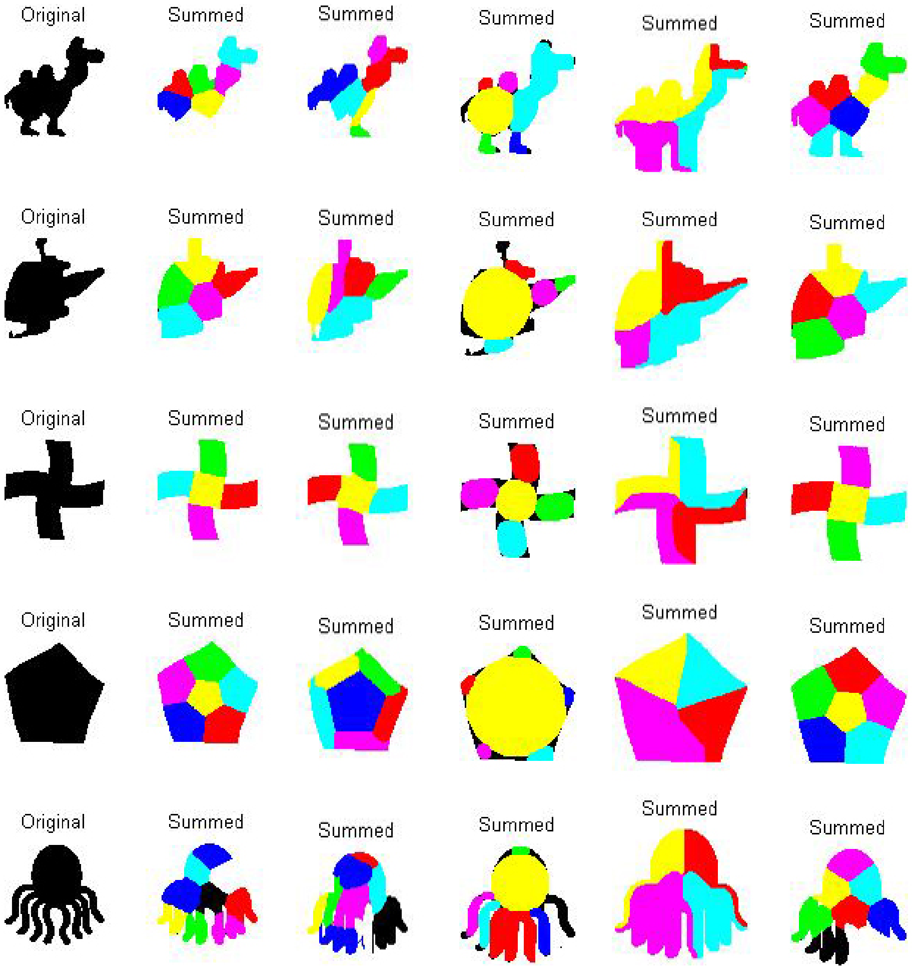

Figure 1: Columns from left to right: Sample images from MPEG7 dataset, K-means, GMM, Morphology w/ Maximal Disks, Watershed, Morphology w/ K-means.