International Conference on Information and Knowledge Engineering
Las Vegas, Nevada, June 23 - 26, 2003

Hausdorff Metric Based Vector Quantization of Binary Images

Aiyesha Ma, Rishi Mukhopadhyay, Ishwar K. Sethi
Department of Computer Science and Engineering
Oakland University
Rochester, MI

Abstract

In this paper we present a vector quantization method to obtain perceptually meaningful descriptors from binary images for use in pattern recognition tasks. We introduce a distance measure and an averaging method based on the Hausdorff distance metric. Additionally we compare the proposed methods to existing Hard and Soft Centroid methods of vector quantization of binary images.

1. Introduction

The vector quantization (VQ) of images has been an active research area for over twenty years [1-2]. Most of VQ research has been concerned with its application for image compression. Vector quantization design, such as the LBG algorithm [3], involves some clustering mechanism that attempts to minimize the distortion between the cluster members and their representative centroids. Most techniques employ Euclidean or weighted-Euclidean metrics for calculating this distortion to maximize perceptual similarity for faithful reproduction of the vector quantized images.

Euclidean based vector quantization methods are particularly unsuitable, however, for binary images since the representative centroid obtained from using a Euclidean-based averaging method will almost never be a binary image. Two techniques have been proposed in past research to deal with this: the Hard Centroid and Soft Centroid methods [4]. The Hard Centroid method averages all the members of a cluster and then thresholds it to produce the new centroid. The Soft Centroid method keeps the grayscale average of the cluster members and only thresholds the images at the end when the final codebook is outputted. With this method, the cluster centers effectively become probability distributions.

The codebook generated by a vector quantizer can also be viewed as a catalog of features or descriptors and the process of encoding an image with a codebook can be viewed as a feature detection operation. From such a viewpoint, the codebook design criterion of enforcing some suitable measure of pixel-by-pixel similarity seems unsuitable for the creation of descriptors; the descriptors should ideally be tolerant of some variations in location, orientation, size, and shape. In this paper, we explore this issue by developing a vector quantization technique based on the Hausdorff metric of distance between two sets of points [5]. The motivation of our technique comes from the possibility of using the occurrence patterns of codewords as local feature descriptors for pattern analysis and annotation tasks for image category classification [6-7]. The codebook generation in our technique is done by using k-means clustering of binary, line-like images, for example images obtained by the application of an edge detector. The goal of our method is to cluster image blocks containing line segments with similar shapes together.

The organization of the paper is as follows. Section 2 illustrates the need for a distance metric that is able to capture the perceptual shape of binary points in a grid and suggests a measure based on the Hausdorff metric. Section 3 examines the issue of determining an average, i.e. the cluster center representation, for binary images and suggests two different averaging schemes. Section 4 presents a set of results using the proposed VQ scheme and compares them to existing hard and soft centroid methods. It also analyzes the computational complexity of the suggested schemes for averaging. Finally, the concluding remarks and possible extensions of the work are presented in Section 5.

2. Distance Metrics

In vector quantization, the Euclidean or Mean Squared Error distance is a popular distance metric [1,2]. While this works well in most image-compression contexts, in the context of binary images containing line segments, Euclidean distances do not yield descriptors with sufficient discrimination for detecting features. For example, consider the 7-by-7 image blocks in Figure 1 with their perceptual classifications.

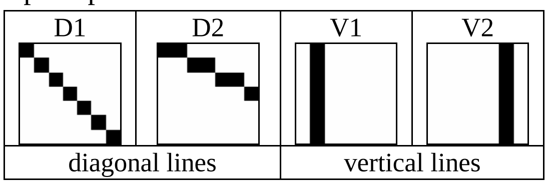

Figure 1

To group these images perceptually, the distance between the two diagonal lines should be less than the distance from either diagonal line to either vertical line. However, the Mean Squared Error distance measure, defined as $D = \sum (A - B)^2$, which in the case of binary images is equivalent to the Hamming distance, $\sum A \otimes B$, does not reflect this. The distance between images D1 and D2 is 12, but the distance between images D1 and V1 or images D1 and V2 is also 12. Additionally, the distance between images V1 and V2 is 14.

Instead we look at the Hausdorff distance between sets of points [5]:

$$H(A,B) = \max(\min(\|a-b\| \forall b \in B) \forall a \in A) \cdot \quad (1)$$

Qualitatively, the Hausdorff distance assumes each point in A maps to its nearest neighbor in B. The maximum of these nearest neighbor distances signifies the greatest displacement of any point in A to the point it maps to in B. With our binary images we considered the black pixels to be our set of coordinate points, and then considered both Euclidean and Manhattan metrics of distance between these points in the plane for $\|a-b\|$ in Eqn. (1). The Manhattan distance was selected based on an initial subjective comparison of the two methods and for reasons of efficiency. Calculating the Hausdorff distances between the above images

results in a distance of 5 from image D1 to image V1, and a distance of 4 from image D1 to image D2. However, it also results in a distance of 4 from image V1 to image V2 because of the translation, and a distance of 4 from image V2 to image D2.

Since, in cases of translation, all the nearest neighbor distances are increased by the same amount, we modify the Hausdorff distance to:

$$\bar{H}(A,B) = \max(\min(\|a-b\| \forall b \in B) \forall a \in A) - \min(\min(\|a-b\| \forall b \in B) \forall a \in A) \quad (2)$$

For the above images, this modification gives the distances on the left of Table 1. This modification appears to be an improvement, until we consider the images in Figure 2 containing noise.

Again, perceptually grouping these images would result in image D1 and D2 being together and V1 and V2 being together. However, using Eqn. (2) yields the distances in the middle of Table 1. This illustrates some of the problems that arise with noise in the images. So instead of taking the maximum and subtracting the minimum, we take a percentile [11]:

$$H(A,B) = 80th\%(\min(\|a-b\| \forall b \in B) \forall a \in A) - 20th\%(\min(\|a-b\| \forall b \in b) \forall a \in A). \quad (3)$$

Which yields the distances on the right of Table 1 for the same noisy images (Fig. 2).

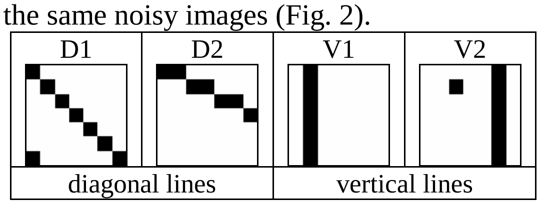

Figure 2

To mitigate the effects of asymmetry inherent in the Hausdorff metric, we sum the distance from A to B with the distance from B to A. Therefore,

$$D(A,B) = H(A,B) + H(B,A) \quad (4)$$

This changes the distances to those on the left for the noiseless data (max minus min) and to those on the right for noisy data (80^{th} percentile minus 20^{th} percentile) in Table 2.

	D1	D2	V1	V2
D1	0	3	5	5
D2	3	0	5	4
V1	5	6	0	0
V2	5	5	0	0

	D1	D2	V1	V2
D1	0	6	5	5
D2	3	0	5	3
V1	3	6	0	3
V2	5	4	3	0

	D1	D2	V1	V2
D1	0	2	3	2
D2	2	0	2	2
V1	1	4	0	2
V2	3	3	0	0

Table 1

This modified Hausdorff measure appears to satisfy a human definition of perceptual similarity between binary images of line drawings and yields a distance measure invariant to translation. Although not impervious to noise, this measure is still moderately robust.

	D1	D2	V1	V2		D1	D2	V1	V2
D1	0	6	10	10	D1	0	4	4	5
D2	6	0	11	9	D2	4	0	6	5
V1	10	11	0	0	V1	4	6	0	2
V2	10	9	0	0	V2	5	5	2	0

Table 2

3. Averaging: Determining a Representative Codeword

Current methods for averaging binary images include soft-centroids and hard centroids [4]. Both of these methods result in probability distributions that need to be thresholded to obtain binary images. Although these methods produce codewords that are an accurate reflection of pixel distribution, they lack the ability to produce codewords that represent the shape a set of pixels form. Consider the 7-by-7 image blocks in Fig. 3.

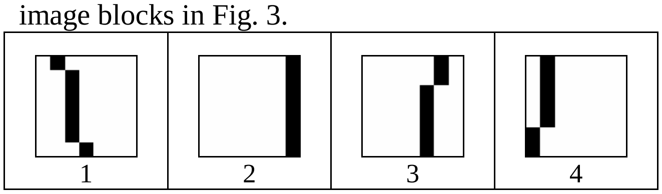

Figure 3

Perceptually these images would probably be called mostly vertical. So the ideal codeword would be the vertical line in Figure 4.

However the Soft Centroid method would result in a probability distribution looking like the image block on the left in Figure 5 and the Hard Centroid method would result in the thresholded binary image on the right. Neither result in Fig. 5 is a meaningful descriptor.

Another averaging method is to take the clustroid as the codeword. The clustroid is defined as the data point in the set for which the sum of the distances to the other members of the cluster is minimal. Using our modified Hausdorff measure results in the codeword in Fig. 6. Although better than codewords obtained with the hard and soft centroid method, the codeword in Fig. 6 could still be improved.

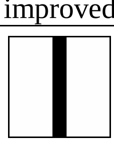

Figure 4

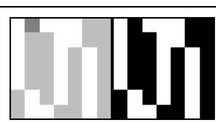

Figure 5

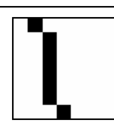

Figure 6

We present a new averaging method based on the Hausdorff mapping concept of the nearest neighbor point. Given a set, $\{C_i\}$, of binary images (where each C_i is a binary image and i ranges from 1 to m, the number of images in the cluster) and the set of points, $\{P_i\}$, in the key-block (where the key-block is a guess at the codeword, and each P_j is a coordinate pair representing one of the black points in the key-block), then the new average is defined as $\{P_j'\}$, where

$$P_j' = round\left(\frac{\sum_{i=1}^{m} NN(P_j, C_i)}{m}\right) \quad (5),$$

and $NN(P,C)$ is a function that returns the coordinate of the nearest neighbor point in image C relative to point P. In clustering applications, the key-block is the current value of the cluster mean.

For the images in Figure 3, if images 1, 2, and 3 are designated as the images in the cluster and image 4 is designated as the key-block then the Hausdorff Based Averaging (HBA) method follows the steps illustrated in Figure 7. In column 2, P_j is depicted in gray. In column 3, the nearest neighbor point in the other images is denoted in gray. In column 4, the resulting change to the key-block ($\{P_j\}$) is depicted.

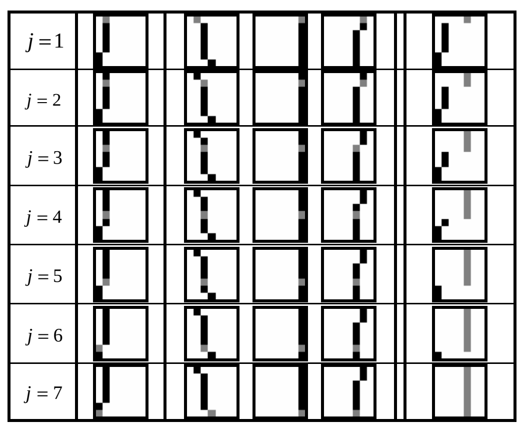

Figure 7

Occasionally, multiple key-block points map to the same location. To keep the key-block from disappearing, points are inserted into the key-block image at the coordinate locations where the images being averaged have the highest frequency of black pixels. Over a few iterations, the inserted points move to more representative places in the key block.

Fig. 8 illustrates the results of averaging the above data set (Fig. 3) with each one of the images tried as a key-block. The leftmost image in Fig. 8(1) portrays the result of the HBA method with image 1 from Fig. 3 tried as a key-block. Fig. 8(2) shows the result when image 2 is the key-block; Fig. 8(3) when image 3 is the key-block; and Fig. 8(4) when image 4 is the key-block.

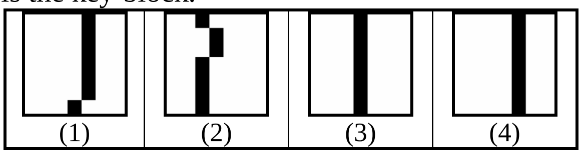

Figure 8

Fig. 8 clearly demonstrates the significant improvement gained with the HBA method over Euclidean methods. Additionally, the HBA method performs at least as well as the clustroid method in all cases.

4. Results

The ideas described above were implemented to perform vector quantization according to the following steps:

Vector Quantization Method

1. Initialize cluster centers to randomly selected blocks from the data set.
2. Assign each block in the data set to its nearest cluster center.
3. Find the new center of each cluster.
4. If any cluster is empty, a randomly selected block from the data set is assigned to the cluster.
5. Repeat steps 2 through 4 for a predefined number of iterations

To test our ideas, we performed edge analysis using a Laplacian of Gaussian operator on six grayscale images to obtain binary line drawings (Fig. 9). Blocks of pixel size 5-by-5, 7-by-7, and 9-by-9 were taken randomly from the binary images; blocks with no points were discarded. The above vector quantization method was applied on these three sets of blocks to obtain codebooks of size 8 and 16. One codeword was automatically designated as empty (all white), so the blocks were clustered into 7 and 15 clusters. A separate codebook was obtained for each of four different averaging methods: Hausdorff Based Averaging (HBA) method, Hausdorff Based Clustroid (HBC) Method, Hard Centroid (HC), and Soft Centroid (SC). For the HBA and HBC methods the modified Hausdorff distance measure was used. The Euclidean distance metric was used with the Hard and Soft Centroid averaging methods.

For the 5-by-5 blocks, 2956 blocks were randomly chosen from the entire picture set of 15,606 blocks. For the 7-by-7 blocks, 1769 blocks were chosen out of 7776 blocks; and for the 9-by-9 blocks, 1107 out of 4704. In the randomly chosen 5-by-5 blocks, there were 780 distinct blocks; in the 7-by-7 blocks 1188; and in the 9-by-9 blocks 957.

For each data set the distances between the codewords were calculated, and for each cluster the distance from the members in that cluster to their representative codeword were calculated. The ratio between the average inter-cluster distance (codeword to codeword) and the average intra-cluster distance (cluster member to codeword) for each data set was then calculated. This ratio is an indicator of the degree of separation of the clusters from each other. A ratio greater than one indicates that the codewords are more separated from each other than they are from the blocks they represent; the ratio is a quantitative measure of the success of the averaging and distance methods being tested. These ratios are shown in Tables 3 and 4, respectively for 16 and 8 codewords.

	HBA	HBC	HC	SC
5-by-5	1.311	3.223	1.168	0.266
7-by-7	1.296	2.016	1.115	0.114
9-by-9	0.941	2.373	0.952	0.408

Table 3

	HBA	HBC	HC	SC
5-by-5	0.976	5.868	1.834	0.565
7-by-7	0.980	1.298	1.223	0.524
9-by-9	0.766	3.241	1.109	0.575

Table 4

From Tables 2 and 3, the HBC method appears to result in the best separation. Usually the 5-by-5 blocks had the best separation. This is probably due to the fact that the larger blocks tended to pick up multiple line segments that caused more significant inaccuracies.

A visual comparison of methods is provided by the following images (Fig. 10-14). For comparison, the Hard and Soft Centroid methods are also depicted.

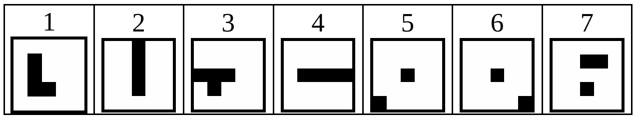

Figure 10: Codewords for 5-by-5 blocks with 8 means, HBA

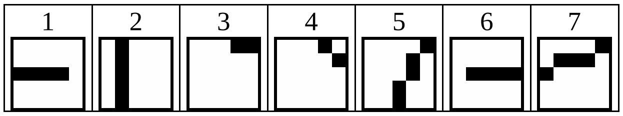

Figure 11: Codewords for 5-by-5 blocks with 8 means, HCA

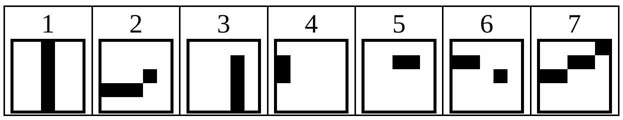

Figure 12: Codewords for 5-by-5 blocks with 8 means, Hard Centroid

In Fig. 12, we see the problem of translation arising. For example, codewords 1, 3, and 4 are all vertical lines. Additionally, codewords 2, 5, and 6 are also similar.

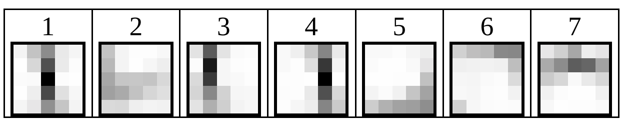

Figure 13: Codewords for 5-by-5 blocks with 8 means, Soft Centroid(Before thresholding)

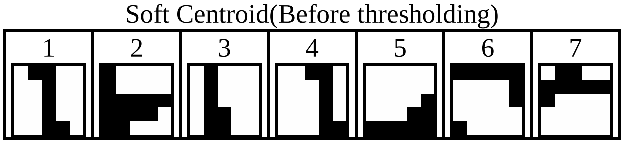

Figure 14: Codewords for 5-by-5 blocks with 8 means, Soft Centroid(After thresholding)

Again, we see the problem with translation (Fig. 14). Codewords 1 and 4 are identical, except for the shift by one pixel.

Another visual comparison is provided by the images in Fig. 16, which reproduce the original image of Fig. 15 after vector quantization. Ideally vertical lines should represent vertical lines and diagonal lines should be represented by diagonal lines. Some methods preserve line directions better than others. The HBA method preserves horizontal and vertical lines, but has trouble with the diagonal lines. The HBC method appears to preserve vertical, horizontal, and diagonal lines.

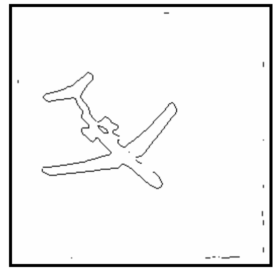

Figure 15: Original image

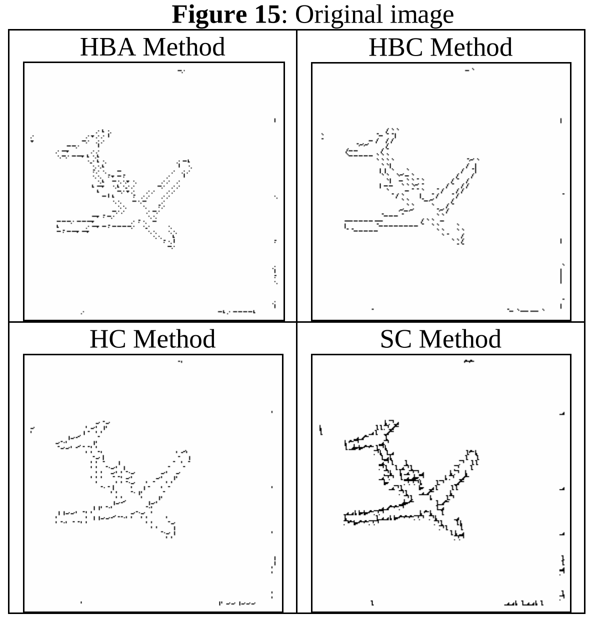

Figure 16: Reproduced images

6. Conclusion

We have presented a binary vector quantization method for obtaining perceptually meaningful descriptors for use in pattern analysis. The use of modified Hausdorff distance in combination with either the Hausdorff Based Averaging (HBA) method or the Hausdorff Based Clusteroid (HBC) method appears better suited than previously established methods for performing vector quantization to perform feature detection.

We are currently exploring several possible variations of the methods presented in this paper to further improve the performance. For example, experimenting with different threshold values for the pixel density or filtering out blocks with no definitive shape may improve the quality of the clusters. In the method described in this paper, blocks with no pixels were ignored. However, a block with one to three pixels in it rarely had a defining shape that was useful to the clustering mechanism. Another possible improvement to the

HBA method is to have an all black key-block instead of using the centroid from the previous iteration as the key-block. This would allow every point in a block to map to some point in the key-block.

Since both the Hausdorff distance metric and the HBA method treat the data as simply a set of points, the low resolution of the lattice may force approximations that are too inaccurate. One potential method to alleviate this problem is to artificially enlarge the blocks, so that each pixel becomes a set of pixels in the enlarged block.

We are also exploring the use of codebook to encode the spatial arrangement of codebook words in images for image retrieval task. Initial progress using these descriptors on line drawings of images seems promising, and these results will be reported soon.

Acknowledgement

The work described in this paper is the result of a NSF funded REU program. NSF award number EIA-0139580.

References

[1] N.M. Nasrabadi and R.A. King, "Image Coding Using Vector Quantization: A Review," *IEEE Trans. Communications*, pp. 957-971, Vol. 36, No. 8, Aug. 1988.

[2] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.

[3] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Communications*, pp. 84-95, Vol. COM-28, No. 1, Jan. 1980.

[4] P. Franti and T. Kaukoranta, "Binary Vector Quantizer Design Using Soft Centroids," *Signal Processing: Image Communication*, 14, 677-681, 1999.

[5] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp 850-863, Vol. 15, No. 9, Sept. 1993.

[6] Q. Iqbal and J.K. Aggarwal, "Applying Perceptual Grouping to Content-Based Image Retrieval: Building Images," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, June 23-26, pp. 42-48, 1999.

[7] L. Zhu, A. Rao, and A. Zhang,"Advanced Feature Extraction for Keyblock-Based Image Retrieval," pp. 179-182, *Proceedings ACM Multimedia Workshop*, 2000.