

# Shape Recognition with Generalized Beam Angle Statistics

*Ömer Önder Tola, Nafiz Arıca ve Fatoş Tünay Yarman Vural*

Department of Computer Engineering  
Middle East Technical University, Ankara  
{onder,nafiz,vural}@ceng.metu.edu.tr

## Abstract

In this study, we develop a new shape descriptor and a matching algorithm in order to find a given template shape in an edge detected image without extracting the boundary. The shape descriptor based on Generalized Beam Angle Statistics (GBAS) defines the angles between the lines connecting each boundary point with the rest of the points, as random variable. Then, it assigns a feature vector to each point using the moments of beam angles. The proposed matching algorithm performs shape recognition by matching the feature vectors of boundary points on the template shape and the edge pixels on the image. The matching process also considers the spatial distance of the edge pixels. The experiments performed on MPEG-7 data set show that the template shapes are found successfully on the noisy images.

## 1. Introduction

In many computer vision applications, it is required to recognize objects in a given image. This task is mainly accomplished by the shape information of objects. Since many properties of objects in our world are strongly determined by geometric properties, shape is one of the most important visual attributes of an object. Shape features provide powerful clue to object identity and functionality, opening an important door for object recognition. Humans can recognize objects solely from their shapes, because; shape often carries semantic information [91], [138]. This distinguishes shape from other low level visual features such as color and texture, which in most cases is not sufficient to reveal object identity.

Shape recognition involves the detection and classification of shapes. Related methods can be divided into two classes. In the first class, detection and classification phases are separated from each other. First, the object is detected and then it is classified. Detection phase uses only the information gathered from the image. Shape detection, which involves locating the object of interest in the image, is also considered as an outproduct of the segmentation process, which is one of the most difficult tasks in computer vision. Segmentation is an important stage, because; the extent one can reach in separation of objects affects the result of shape analysis. Segmentation using traditional low-level image

processing techniques is generally based on one of the two main properties of image: discontinuity in between-class-pixels and similarity among within-class-pixels. The techniques using discontinuity, partition an image, based on the abrupt changes in the pixel values. In this category of techniques, edges provide strong visual clues that can help the segmentation. Therefore, the segmentation is generally performed by first detecting the edges and, then, extracting the boundary through linking the edges. Another category of techniques group pixels together into regions of similarity. The principal approaches in this category are thresholding, region growing, region splitting and merging. After the segmentation, each connected part is assumed to be an object. Then, various features of these object regions are used to classify them. However, the result of segmentation techniques using low level image processing, does not always give satisfactory results for various reasons, including the existence of noise, shape complexity, variations within and across objects in the image, etc. Consequently, shape recognition process can not be completed successfully.

In order to improve the quality of segmentation, the techniques based on the second approach, constrain the problem by exploiting the prior knowledge, which partly eliminates the ambiguities in detecting shapes. This task is performed by model based approaches, which mainly use deformable shape templates. These techniques combine the segmentation and recognition processes in order to use the information about the objects that are to be located in the image [34, 20, 23, 1, 18]. These methods locate the template shape within the image. Since the class of the template shape is known a priori, the classification step is eliminated. This way, only the objects of interest are recognized within images.

Once the shape of interest has been acquired in the image, a set of techniques can be applied in order to extract information from the shape, so that it can be analyzed further for classification. This process is called shape description and generally results in a feature vector (descriptor vector). Feature extraction is generally performed in two steps: In the first step,

shape is represented in such a way that the "important" properties of the shape are preserved for a specific application. In the second step, shape description methods are employed to retain maximum amount of information in a more compact form. The goal of description is to uniquely characterize the shape using its shape descriptor.

In this study, we follow the template based shape recognition paradigm in order to extend the applicability of the boundary based shape descriptor BAS to general images. Accordingly, we suggest a method that performs the detection of a given template shape in images that contain a single object using the shape descriptor, Generalized Beam Angle Statistics, GBAS and a graph based matching algorithm. GBAS, which is obtained with the generalization of BAS, improves BAS so that it can compute the feature vectors of boundary points in the lack of parametric boundary representation. This way, it can be used to compute the features vectors of edge pixels even if it is not possible to extract the shape boundary with the available techniques. Then, the detection problem is reduced to matching the boundary points of the template shape and the edges. This is achieved by a graph based matching algorithm, which transforms the problem to the determination of the optimum path. A polynomial time algorithm based on the dynamic programming approach is employed for the determination of the optimum path.

In this study, a new method is developed for shape recognition in an edge detected image without extracting the shape boundary. The method proposes a new shape descriptor and matching algorithm in order to detect a given template shape. The shape descriptor is formed by generalizing the Beam Angle Statistics (BAS), which is previously developed by the authors of this study. Beam Angle Statistics (BAS) is used to describe shape boundary as a series of 1-D functions independent of rotation, translation and scale [3]. But, like many other shape descriptors found in the literature [4][5], the shape descriptor based on BAS, assumes that the shape boundary pixels are extracted and ordered prior to the representation. This study generalizes the BAS function method in order to describe shapes, independent of the ordering of the boundary pixels. The proposed descriptor is called Generalized Beam Angle Statistics (GBAS). The input of GBAS is the edge detected image that is the output of one of the existing edge detection algorithms. In this edge image, the boundary of the template shape cannot be defined by connected and closed boundary pixels, instead, boundary pixels are found in the set of unordered, missing and excessive edge pixels. Proposed shape descriptor, identifies the required GBAS feature vector of each boundary pixel, performing a computation that is independent from the ordering of the edge pixels of the shape being described.

The proposed matching algorithm is used to search a given template shape in an edge detected image. The algorithm provides matching of the GBAS features extracted

from the boundary pixels of the template shape and the edge pixels of the given image. It also considers the spatial distance of edge pixels by using a dynamic programming approach.

GBAS Shape Descriptor and Matching algorithm is described in sections 2 and 3, respectively. Experiments performed on the proposed method is given in section 4. Finally, the conclusions are discussed in section 5.

## 2. Generalized Beam Angle Statistics

Let  $P = \{p_1, p_2, \dots, p_N\}$ , be a set of edge pixels that are extracted using an edge detection algorithm from an image. The indices of the elements of the set  $P$  are assigned randomly. The Beam Vector,  $V(p_i, p_j)$ , is defined as a vector from an edge pixel  $p_i$  to another edge pixel  $p_j$  (figure-1).

In order to compute the GBAS feature vector of an edge pixel  $p_i$ , we first partition the set of edge pixels  $P$  into two disjoint sets of edge pixels relative to  $p_i$ . For this purpose, the **Mean Beam Vector**  $OK(p_i)$ , of an edge pixel  $p_i$  is found (figure-2).  $OK(p_i)$  is the mean vector of all beam vectors originating at  $p_i$  and directed to all other edge pixels in  $P$ . Thus; in mathematical terms;

$$OK(p_i) = \sum_{j=1}^N V(p_i, p_j), \quad (1)$$

where  $\Sigma$  denotes vector addition operation.

The Mean Beam Vector, partitions the edge pixels set  $P$  into two disjoint sets namely the **forward**,  $I$  and **backward**,  $G$ , edge pixel sets. The forward edge pixel set,  $I = \{i_1, i_2, \dots, i_S\}$  is the union of the edge pixels, whose beam vectors have less than  $\pi$  degrees with the mean beam vector in the counter clockwise direction.

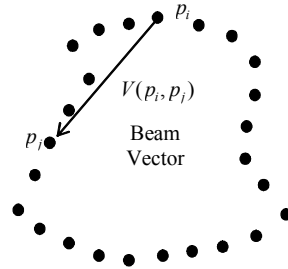


Figure 1: Beam Vector

Similarly, the backward edge pixels set,  $G = \{g_1, g_2, \dots, g_R\}$  is the union of the edge pixels whose beam vectors have more than  $\pi$  degrees with the mean beam vector in the clockwise direction (figure-2).

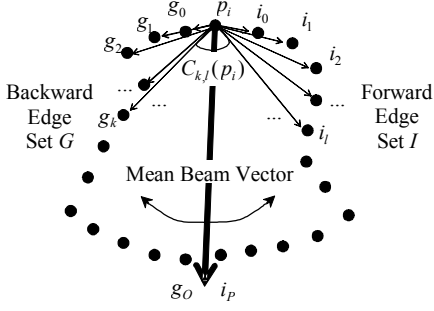


Figure 2: Forward and Backward Edge Pixels, Mean Beam Vector and Beam Angle.

Next, we define the **beam angles matrix**,  $K(p_i)$ , of the edge pixel  $p_i$ .  $K(p_i)$ , denotes all beam angles formed by the forward and backward beam vectors that are created by the sets of forward  $I$  and backward  $G$  edge pixels, respectively. In mathematical terms, beam angles matrix is defined as;

$$K(p_i) = [C_{k,l}(p_i)], \quad k=1,2,\dots,S \text{ and } l=1,2,\dots,R \quad (2)$$

In the above equation,  $C_{k,l}(p_i)$ , denotes the angle that is formed by the forward  $V(p_i, i_k)$  and backward  $V(p_i, g_l)$  beam vectors that originate at  $p_i$  and are directed to  $i_k$  forward and  $g_l$  backward edge pixels, respectively. This angle is named as **generalized beam angle**. Since  $K(p_i)$  incorporates all beam angles formed by the beam vectors that are created by the forward and backward edge pixel sets  $I$  and  $G$  respectively, it has  $S \times R$  elements. Generalized beam angle statistics are computed using all of these beam angles determined by  $K(p_i)$  and GBAS feature vector is extracted independent of the ordering of the edge pixels.

In this study, for each edge pixel  $p_i$ , the beam angle  $C_{k,l}(p_i)$  is taken as a *random variable* with the probability density function  $P_i(C_{k,l}(p_i))$  and considered as an outcome of the stochastic process which generates the shape being described at different scales. Probability density function  $P_i$  is assumed to be uniform. As a result,  $m^{\text{th}}$  moment of the random variable  $C_{k,l}(p_i)$  is defined as follows:

$$\Xi[C^m(p_i)] = \sum_{k=1}^S \sum_{l=1}^R C_{k,l}^m P_i(C_{k,l}(p_i)) \quad m=0,1,2,\dots \quad (3)$$

In the above formula  $\Xi$  indicates the expected value operator.

The moments describe the statistical behavior of the beam angles, that form the beam angles matrix  $K(p_i)$ , of the edge pixel  $p_i$ . Each edge pixel is then represented by the following **GBAS feature vector**;

$$\Gamma(p_i) = [\Gamma^1(p_i), \Gamma^2(p_i), \dots, \Gamma^d(p_i)] \quad (4)$$

whose components are the moments of **the generalized beam angles**;

$$\Gamma^m(p_i) = \Xi[C^m(p_i)] \quad m=1,2,\dots,d \quad (5)$$

### 3. Matching Algorithm

The matching algorithm is developed to detect a template shape given as a set of ordered boundary pixels;

$$S = \{s_1(x_1, y_1), s_2(x_2, y_2), \dots, s_L(x_L, y_L)\},$$

in a set of unordered edge pixels;

$$P = \{p_1(x_1, y_1), p_2(x_2, y_2), \dots, p_N(x_N, y_N)\}.$$

For this purpose, the algorithm finds a correspondence between two sets by comparing each boundary pixel with all edge pixels. As a result, a **matching set** is constructed;

$$M = \{(s_1, p(s_1)), (s_2, p(s_2)), \dots, (s_L, p(s_L))\},$$

where  $p(s_i)$  denotes the edge pixel that is matched with the boundary pixel  $s_i$ .

The algorithm performs the matching by using the GBAS feature vectors and the Euclidean distances (spatial information) of the edge pixels (figure-3). The algorithm constructs a matching graph, which represents all the matching alternatives between the template shape boundary pixels and the edge pixels. In the graph, every match of the elements of the boundary pixels set  $S$  and edge pixels set  $P$  are denoted by a matching node. It, then finds an optimum matching using dynamic programming method.

Formally speaking, given a template shape  $S$  and an edge image  $P$ , let  $Z(E, F)$  be a multistage di-graph (matching graph), where the nodes  $E_{(i,j)} \in E$  represent the matching between  $s_i$  in the boundary pixels set  $S$  and  $p_j$  in the edge pixels set  $P$ . In the matching graph, each row,  $SE_i = \{E_{(i,j)} \mid j = 1, 2, \dots, N\}$  correspond to the set of all matching nodes of the boundary pixel  $s_i$ . The arcs in  $Z$  are of the form  $\langle E_{(i,j)}, E_{(i+1,k)} \rangle$  for all  $1 \leq k \leq N$  and  $k \neq j$ . Note that, in the matching graph, there exists an arc from the node  $E_{(i,j)}$  to the rest of the nodes in  $SE_{i+1}$ . The exception of this rule is the case where the matching nodes  $E_{(i,j)}$  and  $E_{(i+1,k)}$  are matched with the same edge pixel;  $j = k$ . Then, a matching graph can be represented by a multi-stage di-graph, as shown in Figure 4.

In the matching graph, the cost of an arc  $\langle E_{(i,j)}, E_{(i+1,k)} \rangle$  is defined as follows;

$$F_{(i,j)(i+1,k)} = D(i+1, k) + \alpha U(p_j, p_k) \quad (6)$$

where  $D(i, j)$  is the distance of the  $s_i$  and  $p_j$  pixels' GBAS feature vectors;

$$D(i, j) = \sum_{b=1}^d |\Gamma^b(s_i) - \Gamma^b(p_j)| \quad (7)$$

and  $U(p_j, p_k)$  is Euclidean distance of the  $p_j$  and  $p_k$  edge pixels.

$$U(p_j, p_k) = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2} \quad (8)$$

The parameter  $\alpha$  is the normalization factor that is used to normalize the Euclid distance between the edge pixels in order to make the algorithm independent of the image size.

The goal is to find the minimum cost from top row to the bottom row. Therefore, the problem of matching can be represented as finding the shortest path (matching path), which minimizes the cumulative cost.

$$COST = \sum_{1 \leq i < L} F_{(i,j)(i+1,k)} \quad j-1 \leq k \leq j+1 \quad (9)$$

In other words, the matching path is a path that starts from  $SE_1$  of the first boundary pixel  $s_1$ , ending in  $SE_L$  of the last boundary pixel  $s_L$ , forming a matching set such that each boundary pixel is matched with a unique edge pixel in the set of edge pixels,  $P$ .

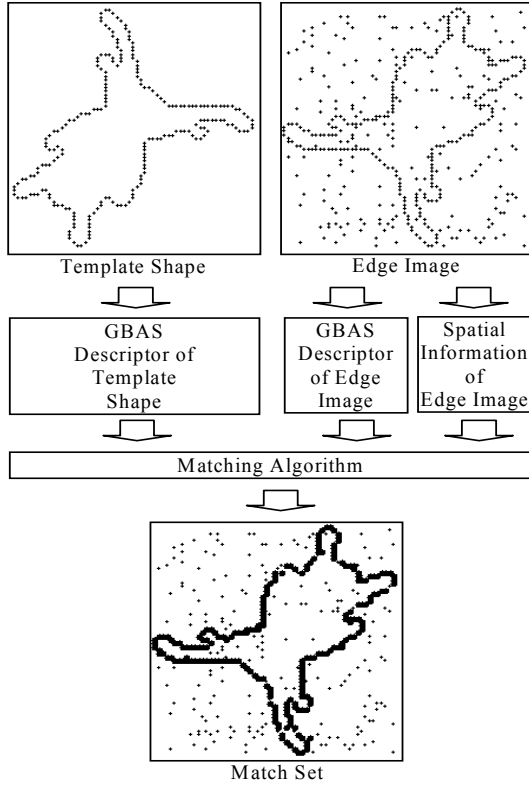


Figure 3: Matching Problem.

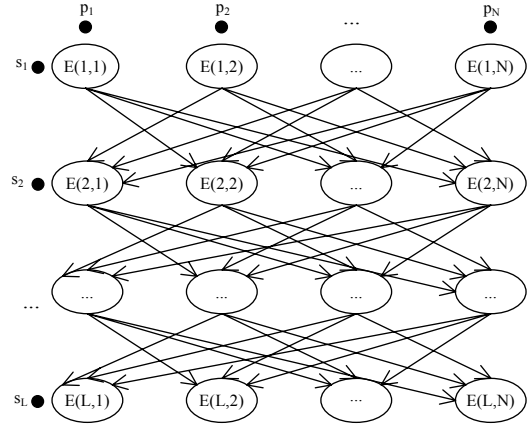


Figure 4: Matching graph, matching nodes and neighborhood arcs.

Let,  $f_i(j)$  be the minimum accumulated cost at node  $E(i,j)$  in row  $i$  and  $Q_i(j)$  be the column of the node in  $SE_{i-1}$  on the shortest path to the node  $E(i,j)$  in  $SE_i$ . We define the path with the minimum accumulated cost as the optimum matching, searched by the following algorithm:

**Matching Algorithm**

**Initialization:**  
for  $1 \leq j \leq N$   
 $f_1(j) = D(1,j)$

**Recursion :**  
for  $2 \leq i \leq L$   
for  $1 \leq j \leq N$   
 $t_{min} = \min_{\{1 \leq t \leq N\}} \{F_{(i-1,t)(i,j)} + f_{(i-1)}(t)\}, \quad p_j \notin Path(E(i-1, t_{min}))$   
 $f_i(j) = F_{(i-1, t_{min})(i,j)} + f_{(i-1)}(t_{min})$   
 $Q_i(j) = E(i-1, t_{min})$

**Termination :**  
 $t_{min} = \min_{\{1 \leq j \leq N\}} \{f_L(j)\}$   
 $Y = Path(E(L, t_{min}))$

In the algorithm,  $Path(E(i,j))$  denotes the subpath that starts from the  $E(i,j)$  matching node of the matching graph that is constructed by following  $Q_i(j)$ . Optimum matching path  $Y$ , is constructed by following  $Q_L(t_{min})$  that is in the field  $SE_L$  of the last boundary pixel  $s_L$  and has the minimum *cost*.

#### 4. Experiments

The proposed generalized beam angle statistics and matching algorithm are tested with the MPEG-7 Core Experiments Shape-1 Part-B data set, which consists of 70 classes of shape images. In the experiments, our aim is to detect template shapes in noisy edge detected images.

An object shape in each class is used a template for the representative of that class of shapes. For this purpose, the

boundary of each template shape is extracted and represented by GBAS feature vector.

In order to construct the noisy edge images, Gaussian noise having zero mean and different values of variance (0.8, 1.4) is added to the shape images in the first step. The Canny edge detection algorithm is, then applied on these noisy images. Additionally, random noise is added to 50 % of the number of edge pixels in the edge detected images. As a result, two types of noisy edge images are obtained for each template shape (see table-1). For each pixel of the edge images, GBAS feature vector is computed using all other edge pixels in the edge image.

In the matching process, template shapes' boundary pixels are matched with the edge pixels of images. The detection of template shape in a given image is achieved by using the matching algorithm proposed in this study. The examples from the experiments are presented in table-1.

## 5. Conclusion

The GBAS shape descriptor with the proposed matching algorithm can successfully extract a shape from an edge detected image, provided that the noise in the image does not exceed a certain value. The amount and character of allowable noise depend on the size and the curvature of the template shape. Experiments are performed on the MPEG-7 Core Experiments Shape-1 Part B data set. Gaussian noise with various variances are added to the images.

It is observed that up to variance 1.4 and 50 % of additive noise give satisfactory results in localizing the boundary shape in the edge detected image.

The proposed matching algorithm yields very satisfactory results provided that the normalization factor  $\alpha$  is adjusted carefully. The experiments indicate that the matching algorithm is sensitive to the shape symmetry and size. The dynamic programming algorithm allows half of the symmetric shape twice, leaving the other half untouched. Therefore, the matching algorithm extracts only half of the shape for fully symmetric objects. Also, the matching algorithm is not size invariant, which can only match the exact shape in the edge detected image.

The weaknesses mentioned above can be attacked by putting some controls in the matching algorithm, which is the subject of future studies.

## 6. References

- [1] Anil K. Jain, Yu Zhong, Sridhar Lakshmanan, "Object Matching Using Deformable Templates", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, p.267-278, Vol. 18, 1996.
- [2] Aparna Lakshmi Ratan, W.E.L. Grimson, William M. Wells III. "Object Detection and Localization by Dynamic Template Warping" *CVPR'98, June 1998*, p.634-640, Santa Barbara, CA.
- [3] Nafiz Arica, Fatoş Yarman-Vural. "BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics." *Pattern Recognition Letters*, Vol.24, pp.1627-1639, 2003.
- [4] S.Loncaric., "A survey of shape analysis techniques." *Pattern Recognition*, Vol.31, pp.983-1001, 1998.

- [5] R.C.Veltkamp and M.Hagedoorn., "State of the art in shape matching." *In Principals of Visual Information Retrieval Springer*, pp.87-119, 2001.


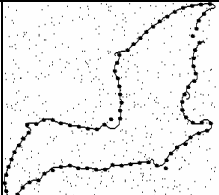
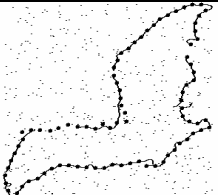

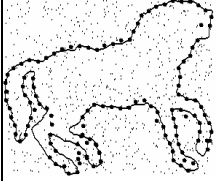
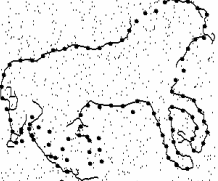

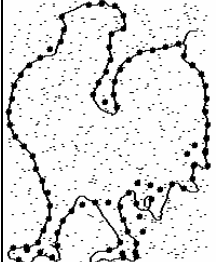
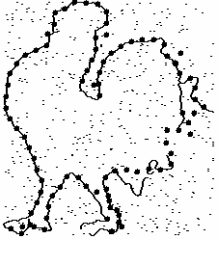

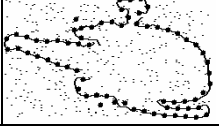
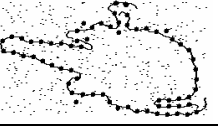

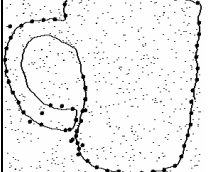
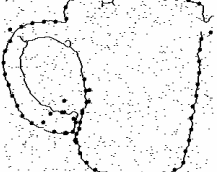

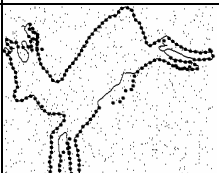
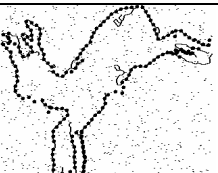
| Template Shape  | Matching on the noisy image (0.8)  | Matching on the noisy image (1.4)   |
|---|--|---|
|    |    |    |
|    |    |    |
|    |   |   |
|  |  |  |
|  |  |  |
|  |  |  |

Table 1: Examples from the experimental results. Leftmost column shows template shapes. Other columns show the results of the experiments performed. In the images, highlighted points indicate the matching points.