

A split-and-merge framework for 2D shape summarization

Demetrios Gerogiannis Christophoros Nikou Aristidis Likas
Department of Computer Science, University of Ioannina,
PO Box 1186, 45110 Ioannina, Greece
{dgerogia,cnikou,arly}@cs.uoi.gr

Abstract—An algorithm for the representation and summarization of a 2D shape is presented. The shape points are modeled by ellipses with very high eccentricity in order to summarize the contour of the shape by the major axes of these elongated ellipses. To this end, at first, a single ellipse is fitted to the shape which is then iteratively split to a large number of highly eccentric ellipses to cover the shape points. Then, a merge process follows in order to combine neighboring ellipses with collinear major axes to reduce the complexity of the model. Experimental results showed that the proposed algorithm provides a shape summary which not only overcomes the representation of a shape by a Gaussian mixture model but also is largely more accurate with respect to the progressive probabilistic Hough transform for shape representation. It must be noted that, for our method, a shape is a unordered set of points describing the contour of a region.

I. INTRODUCTION

Shape is an important attribute for describing objects and plays an important role in many computer vision applications, like image retrieval, segmentation and registration. A detailed review of shape description techniques may be found in [21]. In the majority of these methods, before any further processing, a preprocessing step is required to extract a structural definition of the shape from an unordered collection of points. A common approach is to represent the shape as a collection of curves, whose simpler representative is the straight line, turning the structural definition to a model fitting problem.

The commonly used algorithm of Moore [18] was a first solution to shape following. However, this algorithm is appropriate only for traversing curves, without intersections and produces models with high complexity. Other approaches to the problem are the incremental line fitting [2] which is sensitive to noise and, most importantly, needs sequential ordering of the points and probabilistic methods [13] based on the EM algorithm, generally necessitating the prior determination of the number of model components.

The Hough transform (HT) is widely used and many variants have been proposed to improve its efficiency [10]. Among these variants lies the Randomized Hough Transform (RHT) [17] which randomly selects a number of pixels from the input image and maps them into one point in the parameter space which showed to be less complex as far as time and storage are concerned. In [12], the probabilistic HT was proposed whose basic idea is to apply a random sampling of the edge points to reduce computational complexity and execution time.

The method revealed successful and further improvements were introduced in [15]. A similar concept was proposed in [8], where an orientation-based strategy was adopted to filter out inappropriate edge pixels, before performing the standard HT line detection which improves the randomized detection process. Also, the idea of fuzziness is integrated in the main algorithm in [5] to model the uncertainty imposed in the shape contour due to noise. Thus, a point can contribute to more than one bin in the standard HT process. A general comparison between probabilistic and non-probabilistic HT variants can be found in [11].

The Robust HT is introduced in [1] where both the length and the end points of the lines may be computed. Moreover, the algorithm in [6] provides a method for adopting a shape dependent voting scheme for the calculation of the histogram bins. Finally, a novel HT based on the eliminating particle swarm optimization (EPSO) is proposed in [7], to improve the execution time of the algorithm. The problem parameters are considered to be the particle positions, and the EPSO algorithm searches the optimum solution by eliminating the "weakest" particles, to speed up the process.

In this paper, we propose a method to describe the contour of a 2D shape represented by a set of points without ordering. The result of the method is a sequence of straight line segments modeling the shape, which correspond to the major axis of highly eccentric ellipses fitted on the shape contour. The basic idea relies on a split-and-merge process, where the algorithm is initialized by one ellipse, representing the mean and the covariance of the initial point set, which is then split, through an iterative scheme, until a number of small and elongated ellipses occurs. Then, a merge process takes place to combine the resulting ellipses in order to reduce the complexity of the representation. At the end, the algorithm provides a relatively small number of elongated ellipses fitted on the shape. The proposed method is general and is not limited to shapes represented by a closed curve but also models shapes containing inner structures (joints). Numerical experiments on common shape databases are presented to underpin the performance of the proposed method. The number of ellipses is shape dependent and it is determined by two parameters controlling both the split and the merge processes.

II. DESCRIPTION OF THE METHOD

Let $X = \{\mathbf{x}_i | i = 1, \dots, N\}$ be the set of points describing the shape and $E = \{\epsilon_i | i = 1, \dots, K\}$ be the set of lines describing the shape segments, where ϵ_i is the line describing the i -th segment. The whole process may be viewed under the concept of a rate-distortion strategy.

We define the Distortion Δ induced by the representation of line segments:

$$\Delta(X, E) = \sum_{i=1}^N \sum_{j=1}^K \delta_{ij} d(\mathbf{x}_i, \epsilon_j), \quad (1)$$

where K is the number of line segments the model uses to describe the contour of the shape, $\mathbf{x}_i \in \mathbb{R}^2$, $i = 1, \dots, N$ are the points of the shape, $d(\mathbf{x}_i, \epsilon_j)$ is the perpendicular distance of point \mathbf{x}_i to line ϵ_j , δ_{ij} is an indicator function whose value is 1 if point \mathbf{x}_i corresponds to line segment ϵ_j , i.e. is closer to line ϵ_j than any other line $\epsilon_k | k=1 \dots N, k \neq j$, otherwise it is 0.

In order to prevent overfitting, models having a large number of line segments should be penalized. Therefore, an ideal shape representation would be the one with simultaneously low value of Δ and low complexity.

The computation of the ellipses, representing the shape segments, is performed in two steps: an iterative *split process*, where the shape contour is approached by a number of line segments represented by the major axes of the corresponding ellipses, and an iterative *merge process*, where small line segments are merged to reduce the model complexity. The split process tries to minimize the distortion of the shape while the merge process increases the compression rate, i.e. the number of line segments compared to the total number of points in the set. In what follows the two steps are presented in detail.

A. Split Process

The ultimate goal of this step is to cover the shape with line segments representing the long axes of elongated ellipses and therefore, each point of the shape should be assigned to an eccentric ellipse. A split criterion is defined, based on the Gestalt theory [14]. It models the linearity and the connectivity the human brain uses when modeling contours.

In order to split a shape segment X , it should be either non linear or non connected, or both. Linearity describes how close to a straight line points of X are, while connectivity measures how close these points are. In our method linearity is described by the minimum eigenvalue of the covariance matrix of the points of X . The connectivity is the maximum gap between two successive points. In order to determine it, X is projected onto the two axis of the ellipse corresponding to the covariance matrix of X . Let X_1, X_2 be the projections of X on the major and the minor axis respectively. Then each $X_{i|i=\{1,2\}}$ is sorted and the connectivity $W_{i|i=\{1,2\}}$ is computed:

$$W_{i|i=\{1,2\}} = \max_{j=1 \dots N-1} |\mathbf{x}_i^j - \mathbf{x}_i^{j+1}|. \quad (2)$$

where N is the number of points in X and \mathbf{x}_i^j is the j -th point of the sorted $X_{i|i=\{1,2\}}$. The larger the connectivity,

the better (more explicit) the distinction between the sets is. The split direction followed is the one providing the largest $W_{i|i=\{1,2\}}$.

Eventually, the adopted strategy that minimizes Δ and privileging elongated ellipses obeys to the following rule: split every ellipse whose minimum eigenvalue is greater than a threshold T_1 (linearity) and the maximum gap, within the current segment is greater than a threshold T_2 (connectivity). The process is initialized with one ellipse, corresponding to the covariance of the initial points set centered at the mean value of the shape.

At time step $t + 1$, a given ellipse, characterized by the eigenvalues λ_1^t and λ_2^t of its covariance matrix Σ^t (with $\lambda_1^t \geq \lambda_2^t$), with center μ^t , is split to two new ellipses with centers the two antipodal points on the major axis:

$$\begin{aligned} \mu_1^{t+1} &= \mu^t + \sqrt{\lambda^t} e^t, \\ \mu_2^{t+1} &= \mu^t - \sqrt{\lambda^t} e^t, \end{aligned} \quad (3)$$

where e^t, λ^t are the eigenvector, eigenvalue respectively corresponding to the split direction along which split is performed (figure 1).

The shape points are then reassigned to the new ellipses according to the nearest neighbor rule applied only to those of the initial cluster that was split. By these means, new ellipses occur, which are more elongated as they have greater eccentricity and their minor axes are closer to the shape contour (fig. 2). Moreover, this detailed representation of the point set provides high accuracy modeling of joints, corners and parts of the shape exhibiting high curvature.

In order to describe accurately a shape segment, the covariance of the points defining the convex hull, instead of the whole set, is computed, providing thus a more sensitive to outliers ellipse fit, describing better the spatial distribution of the points and providing a more precise split criterion and direction.

B. Merge Process

The role of the merge process is to reduce the complexity of the model. There are many adjacent ellipses whose major axes have similar orientation and it would be beneficial to merge them and replace them by a more elongated ellipse. Therefore, in this step, ellipses are merged by a similar in spirit rule as in the split process: merge two consecutive ellipse if the resulting ellipse has minimum eigenvalue smaller than a threshold T_1 (linearity) and the marginal width between the two clusters is smaller than a threshold T_2 (connectivity).

Notice that the threshold T_1 may be equal to the threshold used in split process, where the value of parameter T_1 specifies whether an ellipse has low eccentricity and needs to be split. In the merge process, it indicates whether two candidate for merging ellipses would result in an ellipse with high eccentricity. On the other hand, a relaxation of the merge threshold may lead to a rougher model of the shape, smoothing out details like joints. In our experiments, the merge threshold was the same with the split threshold. The same applies for

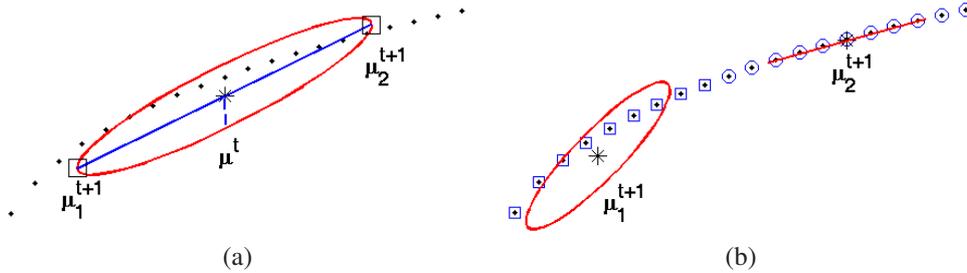


Fig. 1. Split process. (a) At time step $t + 1$, the ellipse with center μ^t is split into two ellipses with centers μ_1^{t+1} and μ_2^{t+1} given by (3). (b) The new centers are marked with a star (*). The reassignment of the points to the new centers is shown. Points of one category, assigned to μ_1^{t+1} , are marked with a circle, while points assigned to μ_2^{t+1} , are marked with a square. The minor axis indicates the border line, thus the major axis is the split direction followed.

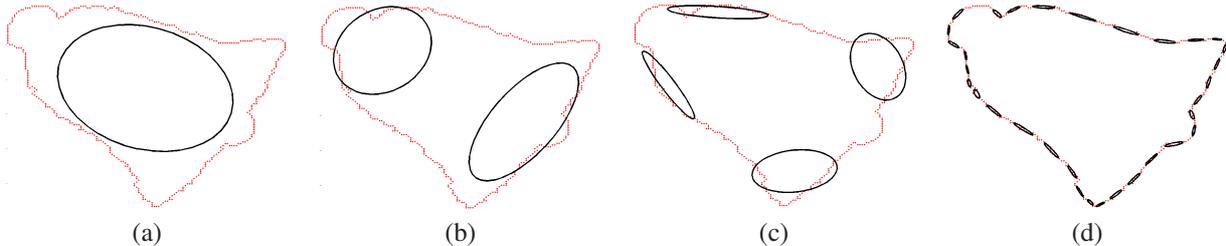


Fig. 2. Steps of the split process. (a) Initialization with the mean and the covariance of the set of points. (b) Split into 2 ellipses. (c) Split into 4 ellipses. (d) Final split.

threshold T_2 that indicates whether two segments are close enough to be considered as one line segment.

It could be argued that the involved thresholds result in total into four parameters, instead of two, that control the performance of the algorithm. However, it should be noticed that the involved thresholds have similar influence both in split and merge steps. Therefore, it is quite logical to adopt equal numerical values for those thresholds in both of these steps. However, different values could be employed by varying the importance of each step in the final result. In other words, one could seek for less elongated ellipses, for example, in the merge step than in the split step. This strategy could improve the time complexity of the algorithm and could be employed in case of pure data, where the correctness of the split is highly guaranteed, in the sense that a lot of small ellipses will be produced, as there are no outliers in the shape. In that case, at a second step, a relaxation of the linearity threshold could produce the same result with fewer iterations as the detected ellipses would describe linear structures.

The overall description of the algorithm is presented in in algorithm 1.

III. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the proposed algorithm we applied it to 8 shapes of the the GatorBait100 database [19] consisting of shapes of different fishes and to 70 shapes of the the MPEG-7 shape database [20]. Since our method models each contour segment with the major axis of an ellipse we compared it to Gaussian Mixture Models (GMM) [3].

The numerical evaluation over the two data sets is presented in table I, where statistics on the value of the distortion Δ in (1) and compression rate (the number of ellipses over the

number of points) are shown. As it may be observed, in all cases, the proposed method outperforms Gaussian mixture modeling. Notice that despite the fact that the GMM provides a slightly better compression rate, the corresponding distortion is quite larger than this provided by our method. This is also confirmed by the representative results shown in figure 3. Notice that the GMM failed to model finely some parts of the shape, especially at corners and joints although it was trained with a large number of components, and during the training process empty clusters were dropped (pruning). It could be argued that by increasing the number of components of the GMM better results could have been obtained. However, this would have increased the model complexity and no structural information about the shape would have been provided (e.g. we could have a GMM with one component per point in the extreme case). In our experiments, the number of clusters used to train the GMM was set to 10% of the number of points of the shape.

TABLE I
STATISTICS FOR THE DISTORTION Δ (1), AND COMPRESSION FOR THE COMPARED METHODS

Distortion Δ (1)						
		MPEG7 [20]			Gatorbait [19]	
method	mean	std	median	mean	std	median
Our Method	0.47	.33	0.42	0.31	0.206	0.27
GMM	1.77	2.70	1.46	11.03	15.66	3.80
Compression Rate (number of ellipses over number of points - %)						
		MPEG7 [20]			Gatorbait [19]	
method	mean	std	median	mean	std	median
Our Method	6.90	4.79	5.39	3.76	0.69	3.59
GMM	10.04	0.05	10.03	2.79	0.75	2.67

Since the proposed algorithm fits line segments to the shape

Algorithm 1 Split-and-merge algorithm for shape 2D description

• SPLIT PROCESS**input:** The set of points $X = \{\mathbf{x}_i | i = 1, \dots, N\}$ representing the shape.**output:** A set of ellipses $\{\mu_j, \Sigma_j\}$.

- Initialize the algorithm by estimating the mean and covariance of the shape.
- While there are ellipses to split
 - * Split every ellipse whose smaller eigenvalue is greater than T_1 and its maximum nearest neighbor distance is greater than T_2 . - Select the direction that provides the greatest marginal width. - The new ellipses have centers determined by (3).
 - * Assign every point of the old ellipse to the new ellipses based on the nearest neighbor rule, applied only to the points of the initial cluster that was split.
 - * Update the centers and the covariance matrices of the new ellipses.

• MERGE PROCESS**input:** The ellipses of the split process $\epsilon_j = \{\mu_j, \Sigma_j, j = 1 \dots M\}$ **output:** A reduced number of ellipses.

- While there are ellipses to merge
 - For each ellipse $\epsilon_i, i = 1 \dots M$
 - * For each ellipse $\epsilon_j, j = 1 \dots M, i \neq j$
 - * if merging ϵ_i with ϵ_j provides an ellipse whose minimum eigenvalue is less than T_1 and its connectivity is less than T_2 , perform merging. Set ϵ_i to the new merged result.
-

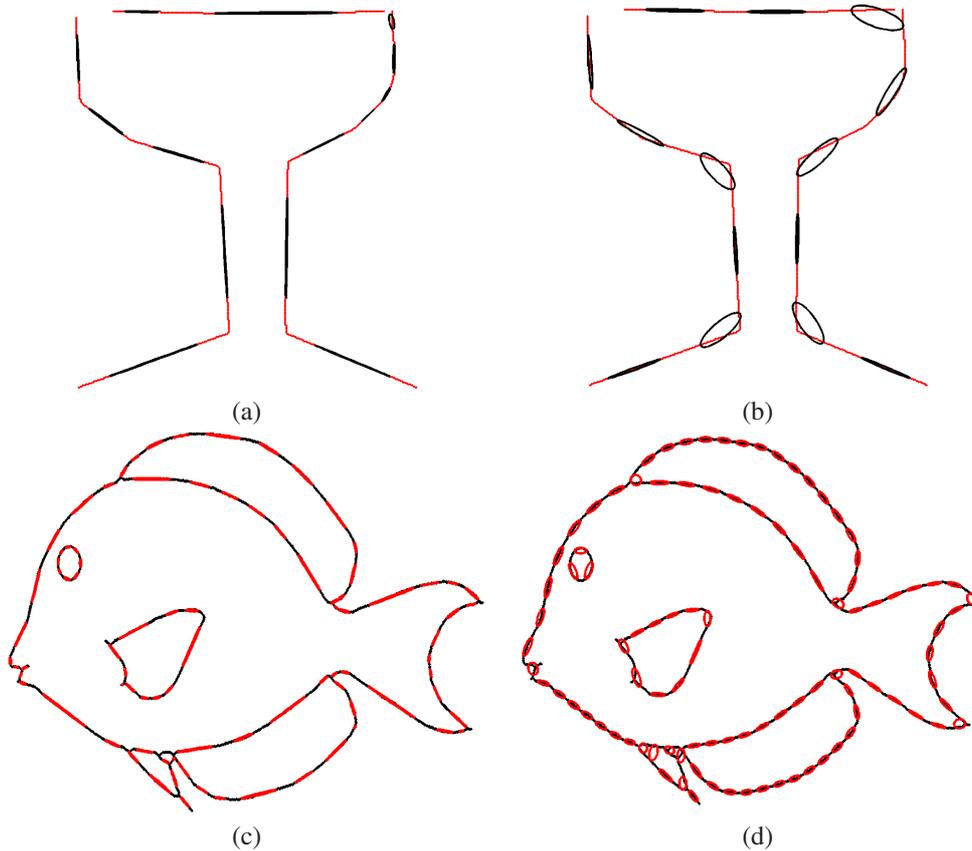


Fig. 3. The summarization of a set of points representing a glass (MPEG-7 dataset [20]) by (a) the proposed method and (b) a GMM. The summarization of a set of points representing a fish (Gatorbait 100 dataset [19]) by (c) the proposed method and (d) a GMM.

we also tested it against the commonly used Hough Transform (HT). However, since the standard HT is appropriate for fitting lines and not line segments, we applied the Progressive Prob-

abilistic Hough Transform (PPHT), as proposed in [16] and implemented in the OpenCV library [4]. The implementation of PPHT imposes three parameters: (i) a threshold, indicating

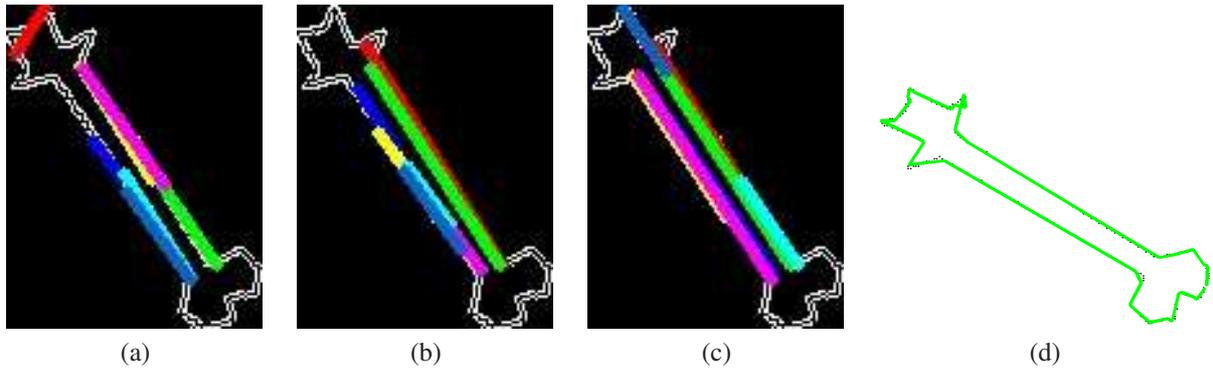


Fig. 4. (a)-(c) Results of the PPHT algorithm [16] to a set of points representing the shape of a bone (MPEG-7 dataset [20]) by varying the minimum number of points in a bin (namely, 5, 15 and 25). Only a small fraction of the lines is drawn for visualization purposes. Notice the overlapping lines. (d) The result of our method. The figure is better seen in color.

the minimum number of points in a bin, in the line parameter space, in order to consider that the line is represented by a sufficient number of points, (ii) the minimum length of a line segment and (iii) the maximum gap between line segments lying on the same axis. In our experiments, we fixed the last two parameters - after a trial and error procedure by keeping those parameters that best fit the examined shapes - and varied the threshold. The obtained results for the PPHT exhibited significant irregularities such as a large number of overlapping lines for the same segment. Also, the corners of the shapes were not correctly captured. Representative experiments are shown in figure 4(a)-(c) while the modeling by our algorithm is illustrated in 4(d).

The PPHT is based on a histogram which correlates the accuracy of the result with the number of bins used. Also, a threshold must be established so as to eliminate lines with small participation in the final result. A small number of bins may lead to an underestimation of the number of segments, while a large number of bins increases the complexity of the model. Moreover, there is no clear borderline between a small and a large number of bins. As far as the threshold is concerned, its value may have similar effects in the final model. A large value may drop some shape segments, while a small value may be responsible for a large number of lines fitted, analogous to a GMM with one component per point. A more important drawback of the PPHT is that many overlapping lines may model the same line segment. Figure 4 presents outputs of PPHT for a given shape, and various parameters values.

In our algorithm there are two tuning parameters, a threshold that determines the elongation of an ellipse (T_1) and a threshold marking the connectivity of a set (T_2). Both are used to decide whether to split (in split process) or merge (in merge process). A small value preserves the shape details, while a larger one provides rougher results. Notice that noise disturbing the contour, may be handled by increasing the values of these thresholds. For our experiments, $T_1 = T_2 = \alpha * \frac{1}{N} \sum_{\mathbf{x} \in X} \{ \frac{1}{8} \sum_{\mathbf{y} \in N_{\mathbf{x}}^8} \{ |\mathbf{x} - \mathbf{y}| \} \}$ where N is the number of points of shape X , $N_{\mathbf{x}}^8$ is the 8-neighborhood of $\mathbf{x} \in X$, i.e. the 8 nearest neighbors to \mathbf{x} and α is a

parameter controlling the accuracy of the result. Large values for α produce better compression rate with greater distortion, and thus not preserving the shape details, while smaller values describe better the shape, but increase the complexity. In our experiments we fixed α to 0.6.

IV. CONCLUSION

The split and merge algorithm presented in this paper iteratively fits a number of elongated ellipses to the contour of a shape. The shape points are represented by the major axes of the ellipses while the relatively small in size minor axes account for possible perturbations from the linear modeling. Furthermore, a merge process combines neighboring ellipses with collinear major axes in order to reduce their number and consequently the complexity of the model. An important aspect of the algorithm is that it does not necessitate an ordering of the points representing the shape. It handles shapes that present joints and multiple structures.

The method imposes two thresholds, that controls the flexibility of the result. Robust estimation of those parameters, is a matter of ongoing research, along with the implementation of the method in a 3D scale. A straightforward application of the method could be in the implementation of a shape retrieval scheme, where the contour of an object should be detected and matched with that of a model in a reference database [9].

REFERENCES

- [1] M. Atiquzzaman and M. W. Akhtar. A robust Hough transform technique for complete line segment description. *Real Time Imaging*, pages 419–426, 1995. 1
- [2] P. Bhowmick and B. Bhattacharya. Fast polygonal approximation of digital curves using relaxed straightness properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1590–1602, 2007. 1
- [3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 3
- [4] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV*. O'Reilly Media, 2008. 4
- [5] V. Chatzis and I. Pitas. Fuzzy cell Hough transform for curve detection. *Pattern Recognition*, 30(12):2031–2042, 1997. 1
- [6] C. Chau and W. Siu. Adaptive dual-point Hough transform for object recognition. *Computer Vision and Image Understanding*, 96(1):1–16, 2004. 1

- [7] H. D. Cheng, Y. Guo, and Y. Zhang. A novel Hough transform based on eliminating particle swarm optimization and its applications. *Pattern Recognition*, 42(9):1959–1969, 2009. 1
- [8] K. Chung, Z. Lin, S. Huang, Y. Huang, and H. M. Liao. New orientation-based elimination approach for accurate line-detection. *Pattern Recognition Letters*, 31(1):11–19, 2010. 1
- [9] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object Detection by Contour Segment Networks. In *European Conference on Computer Vision*, 2006. 5
- [10] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, 1988. 1
- [11] H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja. Probabilistic and non-probabilistic Hough transforms: overview and comparisons. *Image and Vision Computing*, 13(4):239–252, 1995. 1
- [12] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991. 1
- [13] J. Leite and E. Hancock. Iterative curve organisation with the EM algorithm. *Pattern Recognition Letters*, 18(2):143–155, 1997. 1
- [14] D. Marr. *Vision*. Freeman Publishers, 1982. 2
- [15] J. Matas, C. Galambos, and J. Kittler. Progressive probabilistic Hough transform. In *British Machine Vision Conference*, volume I, pages 256–265, London, UK, 1998. 1
- [16] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic Hough transform. *Computer Vision and Image Understanding*, 78:119–137, April 2001. 4, 5
- [17] R. A. McLaughlin. Randomized Hough transform: Improved ellipse detection with comparison. *Pattern Recognition Letters*, 19:299–305, 1998. 1
- [18] G. A. Moore. Automatic scanning and computer process for the quantitative analysis of micrographs and equivalent subjects. *Pictorial Pattern Recognition*, pages 275–362, 1968. 1
- [19] U. of Florida. GatorBait 100. <http://www.cise.ufl.edu/~anand/GatorBait>, 2008. 3, 4
- [20] Temple University. College of Science and Technology. MPEG-7 dataset. <http://knight.cis.temple.edu/~shape/MPEG7/dataset.html>. 3, 4, 5
- [21] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004. 1