Locating and segmenting 3D deformable objects by using clusters of contour fragments

Manuel J. Marín-Jiménez[1], Nicolás Pérez de la Blanca[1], and J.I. Gómez Espínola[2]

[1] Dpt. Computer Science and Artificial Intelligence, University of Granada
ETSI Informática y Telecomunicación, Granada, 18071, Spain
[2] Dpt. Computer Science, University of Jaén
Campus Lagunillas, Jaén, 23071, Spain
mjmarin@decsai.ugr.es, nicolas@ugr.es, nacho@ujaen.es

Abstract. This paper presents a new approach to the problem of simultaneous location and segmentation of object in images. The main emphasis is done on the information provided by the contour fragments present in the image. Clusters of contour fragments are created in order to represent the labels defining the different parts of the object. An unordered probabilistic graph is used to model the objects, where a greedy approach using dynamic programming is used to fit the graph model to the labels.

1 Introduction

In this paper, we approach the problem of simultaneous detection, location and segmentation of 3D objects from contour images. The contour, as feature, has been recurrently used along the literature to solve location and segmentation problems [11],[5], but using it for simultaneously solving both problems we think that represents a new contribution. The contours extracted from an image by the state-of-the-art algorithms, represent a very noisy and deformed estimation of the object boundaries present in it. For this reason, the more relevant advances so far have been given on images of planar objects where an affine deformation on the contours can be assumed. For images of 3D deformable objects, the problem is much more difficult and the current approaches combine textural and contour information for simultaneous location and segmentation [10][13]. In this paper we support the idea that the contour by itself is a feature that allows to extract enough information to locate and to segment a deformable 3D object present in an image. This is the main novelty of this paper.

The elastic graph based algorithms is a well-known approach to detect and to localize 3D objects in general images, [12][10][6][16]. However, most of these approaches need to know some prior information (landmark node) in order to fit the model, Fergus et al. [7] with a star-graph model, or, Crandall and Huttenlocher [4] with the *k-fans* model. Our approach does not need of any prior information since all the required information is extracted from the contour shapes and

their relative positions in the image. This point represent the main technical contribution of this paper. Song *et al.* [14], very recently, have proposed a new approach to detect 3D objects from motion using a probabilistic graph model. This approach has the main advantage on other graph based approaches of being non-parametric, that means that all relevant information about the graph structure is learnt from the sample data, although the graph is restricted to be a decomposable triangulated graph. Here we assume that this class of graph is rich enough to encode the spatial information provided by the contours present in an image. So, we have adapted this approach to carry out the location and the partial segmentation of the object.

In our method, the learning process follows a semi-supervised approach, where we assume as known the bounding-box of the object. We start estimating the contours inside the bounding-box and we create a set of clusters from contour fragments of different lengths where each cluster is associated to a label. From the clusters, we generate maps of labels over the image for each different fragment length. These labelled maps are the input to the graph learning algorithm. Geometrical and statistical information on the relative position of the contour present on each cluster has been used to bound the search space and to improve the efficiency of the fitting process. On the fitted graph model, the smallest region in the image containing the graph nodes gives the best object location, and the contours defining each node give a partial object segmentation.

Outline of the paper In section 2 we deal with the problem of part learning, where we introduce the representation and distance function of contours fragments, along with the clustering approach. Section 3 presents the structural model used for relating the parts model. In section 4 the experimental results are shown. And finally, the paper concludes with the summary and conclusions.

2 Parts learning

The first step in our approach is the automatic learning of parts that represent our target object category. This problem is divided in the following subproblems: *(i)* the contour representation and distance measure; *(ii)* clustering of similar fragments; and, *(iii)* relations between clusters of fragments of different lengths.

2.1 Contour representation and distance measure

We consider a contour as a ordered sequence of n coordinates in the image reference system since we are not interested in full rotation invariance $c = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), ...(x_{n-1}, y_{n-1}), (x_n, y_n)\}$ In order to impose shift invariance we fix the middle point $(\lfloor x_{n/2}, y_{n/2} \rfloor)$ of the contour as the $(0, 0)$ of a local reference system. Different measures depending on the representation has been tested for the problem of contour fragment matching [15]. We use the Euclidean distance in a modified way (named SED) since the widely accepted Chamfer matching [2] did not result satisfactory for computing distance in clustering contour fragments.

Soft-Euclidean-Distance (SED) This measure could be understood as a correlation operation, where one contour is fixed and the other is shifted on the first. The curvature of the points lying out of the overlapping region are used to weights this distance. Let C_1 and C_2 be the vectors containing the curvature c_i for each point of the discarded subfragment from the Euclidean distance d_0 of the compared points. The final distance d is given by the following expression:

$$d = d_0 + f(C_1, m) + f(C_2, m) \tag{1}$$

where,

$$f(C,m) = \begin{cases} \lambda \cdot \beta \cdot max(C,m) & \texttt{if } max(C,m) \geq \tau \\ 0 & \texttt{if } max(C,m) < \tau \end{cases} \tag{2}$$

m is the number of discarded points, $max(C, m)$ is a function that returns the maximum value of the m points of C, λ is a value greater than 1 (to tune), and $\beta = 1 + (m/L)$ with L the full length of the contour (without discarding any piece). This implies that penalization only will be added when the discarded points contain a significative curvature, and, in this case, the penalization will depend on the length of the discarded subfragment (regulated by β) and the value of that maximal curvature.

Before measuring the distance between two contours we remove all possible affine deformations. In order to do this we estimate by least-squares the parameters of an affine transformation between both contours

$$\min_{a_{1..4},d_1,d_2} \sum_{i=1}^{n} (x'_i - a_1 x_i + a_2 y_i + d_1)^2 + (y'_i - a_3 x_i + a_4 y_i + d_2)^2 \tag{3}$$

where (x_i, y_i) and (x'_i, y'_i) are the coordinates of the corresponding points, from each contour.

Subcontour matching is a core operation in the location and segmentation stage. For each possible position of the shorter contour in the longer, we compute SED. The position of the smallest distance is returned.

2.2 Clustering contour fragments

The first step in our approach is to learn the most representative contour fragments defining a specific object. To do this we use a semi-supervised approach, that means, that the bounding-box of the object in the set of learning images is known. We compute the contours present in our region of interest using the classical Canny's algorithm [3], but any other algorithm could also be used. The detected contours are extracted and split in overlapping fragments of fixed lengths. These fragments will be clustered, attending their shape, in order to discover the object parts. The least populated clusters will be discarded. Since we don't know the number of representative parts defining the shape of an object, we based our cluster stage in a distance based cluster algorithm. In particular we have used the agglomerative clustering proposed by Fred and Leitão's [9]. This is

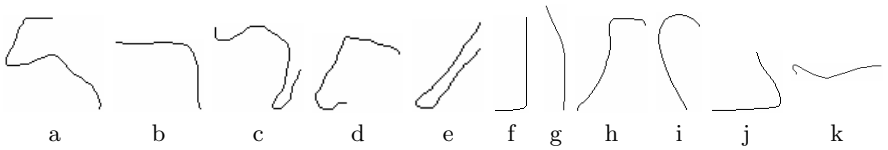

Fig. 1. Contour fragments from different categories: (a,b,c) cow, (d,e) horse, (f,g,h) bottle and (i,j,k) swan.

a hierarchical agglomerative clustering based on dissimilarity between the samples. We tune the parameters α and β described in [9] to generate highly compact clusters, and afterwards we apply the agglomerative clustering proposed in [10] to join some of that clusters (tracking the clusters joint defining equivalencies in matching). For recognition purposes, each cluster is represented by a *representative contour*. This is chosen as the sample contour nearest to the medoid contour of the cluster. In figure 1 some examples of representative contour fragments are shown.

2.3 Fragments hierarchy

Due to the noise introduced by the current contour detectors, the natural object boundaries appear broken. Hence, we get contour fragments shorter than the fragment models. In order to decide to which cluster to assign each contour fragment, we create a fragment contour hierarchy and propagate the information throughout it. For learning such hierarchy, we work with the set of the longest medoids S_L, and the procedure is as follows: for each predefined length l_i, we split the medoids L_j in S_L into overlapping fragments of length l_i, and put all the fragments of length l_i in a single bag b_i. Them, we compute the SED distance between all the pairs of fragments in bag b_i and perform clustering [9], obtaining k clusters c_i. Since we know where each fragment comes from (L_j), we can set the relations between L_j and the new lower-level clusters c_{ik}. Note that each cluster c_ik can have more than one ascendant medoid L_j. In the practice, this hierarchy is employed during recognition in an implicit way, since the detected shorter fragments are submatched over the longest cluster's representative contours. It is important to remark that the shorter the fragment, the greater the number of possible ascendants and consequently the lower the information provided by this fragment. This is implemented by penalizing the submatching distance with a multiplicative factor proportional to the length of the fragment, respect to length of the model.

3 Including structural information: DTG

Once we have defined an object as a collection of parts located in fixed relative positions, in this section we tackle the estimation of the spatial structure of the object. Our structural model is based on decomposable triangulated graphs (DTG), used satisfactorily in human action recognition by Song *et al.*[14]. This

model allow us to define the spatial structure of a shape as a joint probability distribution on a graph, but with the important property that the conditional independence of model parts can be assumed.

Let $S = S_1, S_2, ..., S_M$ be a set of M labels, and X_{Si}, $1 \leq i \leq M$ is the measurement for S_i. If its joint probability density function can be decomposed as a DTG then,

$$P(X_{S_1}, X_{S_2}, ...X_{S_M}) = P_{B_T C_T} \cdot \prod_{t=1}^{T} P_{A_t|B_t C_t} \tag{4}$$

where $A_i, B_i, C_i \in S$, $A_1, A_2, ..., A_T, B_T, C_T = S$, $(A_1, B_1, C_1), (A_2, B_2, C_2)$,..., (A_T, B_T, C_T) are the triangles in the graph model, and $(A_1, A_2, ..., A_T)$ gives an order for such vertices. Let $\chi = \{\overline{X}^1, \overline{X}^2, ..., \overline{X}^N\}$ be a set of samples from a probability density function, where $\overline{X}^n = \{X^n_{S_1}, ..., X^n_{S_M}\}$, $1 \leq n \leq N$ are labelled data. Let $P(G|\chi)$ be the probability function to maximize, where G is the best DTG for the observed data χ. Assuming all prior $P(G)$ are equal, by Bayes rule, $P(G|\chi) = P(\chi|G)$ then the goal is to find the graph G that maximize $P(\chi|G)$ where

$$logP(\chi|G) = \sum_{n=1}^{N} logP(\overline{X}^n|G) = \sum_{n=1}^{N}(logP(X^n_{B_T}, X^n_{C_T}) + \sum_{t=1}^{T} logP(X^n_{A_t}|X^n_{B_t}, X^n_{C_t})) \tag{5}$$

In order to estimate the optimal G we follow the Dynamic Programming approach given in Song [14]. In our case each triangle of the graph (A_t, B_t, C_t) is characterized by a four dimensional Gaussian distribution X defined on the relative positions of triangle vertices: $X = (B_{t_x} - A_{t_x}, B_{t_y} - A_{t_y}, C_{t_x} - A_{t_x}, C_{t_y} - A_{t_y})$

The most challenge problem in this optimization problem is the size of the configuration space for each image. This size is $O(m^n)$, being m the number of graph nodes and n the number of points. Since each label in the image represents a contour fragment identified by its central point, the statistical and geometrical information about the relative location of the contours can be used to define an heuristic criteria to bound the size of configuration space. We assign a weight vector of each observed contour (label) of being candidate for each node. Only the labels with weight greater than a threshold will be considered as candidates for such node. The *normalized weight* $\hat{p}$ of a label (contour) c belongs to a node (cluster) n_i is defined by:

$$\hat{p}(n_i|c) = \exp(-\gamma * D(c, r_i)) \tag{6}$$

Where γ is a factor which controls the steepness of the function, D is a distance function, and r_i is the representative contour (medoid) of node (cluster) n_i. We adapt the algorithms given in [14] to work with these weight vectors.

4 Experimental results

The goal of our experiments is threefold. Firstly, to assert that SED is a suitable way for comparing and matching contours of different lengths, providing a valid

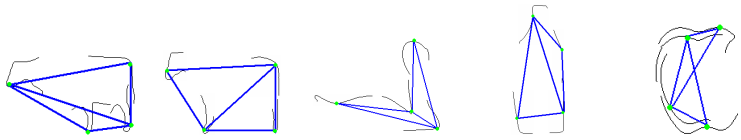

Fig. 2. Model graphs for categories: cow, horse, swan, bottle and applelogo.

distance for computing robust clusters. Secondly, that DTG is able to encode the spatial information of 3D deformable objects. And, finally, that the proposed framework achieves good enough preliminary results in the task of locating and segmenting objects in real images.

For performing the experiments, we use images from databases with well-defined ground-truth (ETHZ-cows [10], Weizmann-horses[3]), and images from databases with annotated bounding-box (ETHZ Shape Classes [4] [8]).

In the learning stage, samples are extracted by using the object bounding-box and resized to a fixed size [0,1]x[0,1]. Then we extract all the contour fragments present in the image in a prefixed range from 40 to 200 pixels in step of 20. Taking into account the number of clusters for each length, we decide which length is the most representative and we use the associated clusters as the estimated nodes. In our examples this length has been: 140 (cow), 100 (horse), 120 (bottle), 120 (swan) and 120 (apple). The associated number of clusters is: 29 (cow), 39 (horse), 29 (bottle), 20 (Swan) and 19 (apple). But for efficiency reason we select a smaller but representative set of clusters. In all our experiments we use four nodes graphs. The graph for each category has been trained with 15 samples. Figure 2 shows the estimated model graphs for the selected categories.

In the location stage, we assume that the object scale is approximately known. So we fix the size of the search window. In order to search the best location we carry out a *Sliding window* technique [1]. That is, for each possible window of size equals to object's bounding-box, we compute the matching graph cost C. This cost is weighted by using the information provided by the initial contour matching. So, the weighted cost C_w is given by: $C_w = C/\left(1 + \sum_i \hat{p}_i\right)$, where $\hat{p}_i$ is the normalized weight (eq.6) of the i-th point in the graph. Due to high cost of fitting the graph, some preliminary heuristics have been tested in order to reduce the computation complexity.

Figure 3 has two parts. The top part shows samples of the databases used in the examples. The bottom part shows some examples of our approach on the different object classes. The number of nodes of the learnt graphs, for each object class, has been fixed equal to the estimated number of representative clusters. We have use ground-truth contour information to estimate the clusters on two of the databases, cows and horses, and the estimated contours inside the bounding-box for the other three classes. According to these preliminary experiments no difference is appreciated neither in the quality of the estimated clusters nor in the quality of the fitted graph.

[3] Horses dataset available at: http://www.msri.org/people/members/eranb
[4] Dataset available at: http://www.vision.ee.ethz.ch/~ferrari

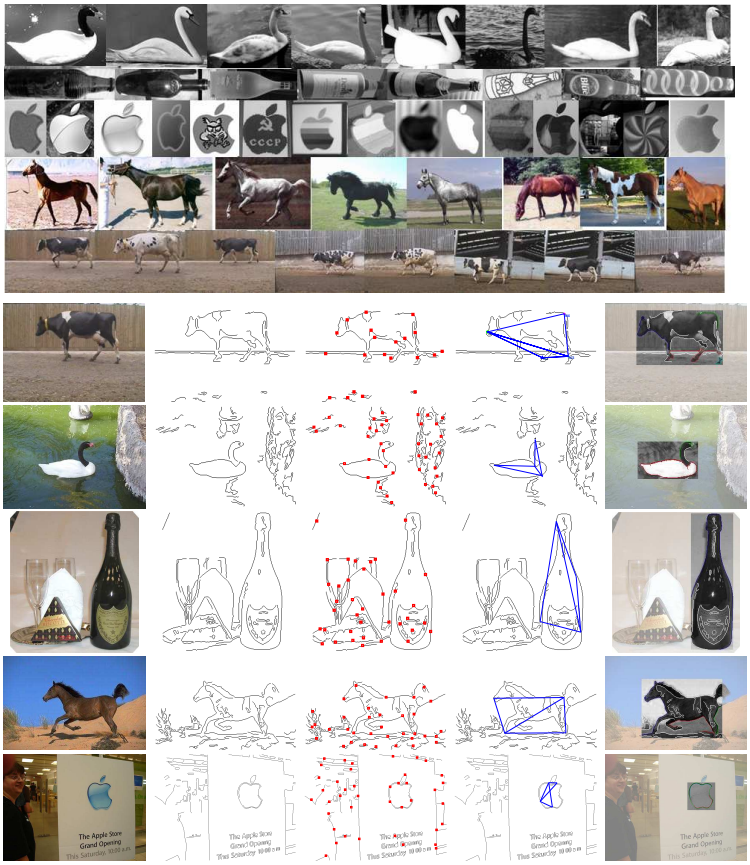

Fig. 3. The top image (5 rows) shows learning samples of the databases. The bottom image shows examples of different stages of our approach. By columns, the images show: 1) original image, 2) detected contours using the canny algorithm, 3) points representing the location of the estimated label, 4) fitted graph, and 5) estimated bounding-box with the contours associated to the fitted nodes, respectively. (Best viewed in color.)

Although some of the parameters have been partially fixed by hand, from our experiments we can say that the proposed approach has worked very well on the set of images we have used, but much more experimentation is needed. Global affine transformation and local large deformation on the object shape are very well absorbed by the clusters of contours, but strong changes in the point of view remains an important challenge for future research. In the graph fitting process we only have taken into account the relative location of the graph nodes in the optimization function, but shape information could also be incorporated.

5 Summary and conclusions

A new algorithm for simultaneous object location and segmentation from estimated contours has been proposed. This algorithm generalizes the current ap-

proach in two different ways. Firstly, it only uses the information provided by the contours present in the image, secondly no landmark or other specific reference is needed in order to locate the object in the image. The central point of the approach is twofold. In a first stage a contour clustering is carried out in order to extract all relevant geometrical information about persistent or relevant shapes present in the contour fragments. In a second stage a non-parametric approach is used to encode the statistical information provided by the relative positions of the clusters obtained from the learning samples. Based on this statistical information we also suggest a mechanism to bound the configuration space.

The approach is semi-supervised, and therefore it only need to know the bounding-box of the object in the learning stage, but we think that after improving the clustering stage this constraint could be relaxed. Further experimental work is needed in order to assess the final performance of the approach.

Acknowledgments This work was supported by the Spanish Ministry of Education and Science (grant FPU AP2003-2405) and project TIN2005-01665.

References

1. Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11):1475–1490, Nov. 2004.
2. Gunilla Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE PAMI*, 10(6):849–865, Nov 1988.
3. John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov 1986.
4. D.J. Crandall and D.P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*, pages 16–29, 2006.
5. D. Cremers, T. Kohlberger, and C. Schnorr. Nonlinear shape statistics in Mumford-Shah based segmentation. In *ECCV*, volume 1, pages 93–108, 2002.
6. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, Feb 2003.
7. R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, pages 380–387, 2005.
8. V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *Proc ECCV*, May 2006.
9. Ana L.N. Fred and José M.N. Leitao. A new cluster isolation criterion based on dissimilarity increments. *IEEE Trans. on PAMI*, 25(8):1–15, Aug. 2003.
10. Bastian Leibe. *Interleaved Object Categorization and Segmentation.* PhD thesis, ETH Zurich, October 2004.
11. F. Mokhtarian and M. Bober. *Curvature Scale Space Representation: Theory, Applications & MPEG-7 Standardisation.* Springer, 2003.
12. A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proc of ECCV*, volume 2, pages 575–588, 2006.
13. J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.
14. Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 25(7):1–14, July 2003.

15. Remco C. Veltkamp. Shape matching: Similarity measures and algorithms. pages 188–199, 2001.
16. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, 2000.