

Object Localization using Affine Invariant Substructure Constraints.

Ishani Chakraborty, Ahmed Elgammal
Rutgers, The State University of New Jersey
110 Frelinghuysen Road, Piscataway, NJ 08854-8019, U.S.A
{ishanic,elgammal}@cs.rutgers.edu

Abstract

In this paper we propose a novel method for generic object localization. The method is based on modeling the object as a graph at two levels: a local substructural representation and a global object graph. In the first level, an object substructure is a quasi affine-invariant canonical encoding of a set of four straight contour lines of the object. The second level is a connectivity graph of these substructures that defines the object. The candidate substructures in an observed image are selected probabilistically using the model distribution. To extract the object graph from these candidates, we exploit the strong inter-structural affinities within the object. We consider the connected graph of all candidates and find a bi-partition of this graph. Finally, the partition with higher density (and hence with higher affinity) is selected and labeled as the object structure. This method is independent of affine transformations of objects and robust to intra-class variability and partial occlusion.

1. Introduction

Object detection and localization is an important task in many computer vision problems. Objects can range from geometrically constrained structures, like aeroplanes, to highly articulated arrangements like human body. Thus, features and detection methods vary according to the object under consideration. Detection methods can also be classified as shape-based, which exploit structure, and appearance-based, which use texture based features. In this paper we propose a shape-based approach which is well suited for object categories whose instances can have variable texture and color and hence can't be localized solely by appearance-based approaches. The algorithm models structure at two levels: a local *substructure representation* and a global *object graph*.

Shape-based approaches are difficult to realize in

practice. This is because most existing shape features are highly sensitive to transformations and intra-class variability. One way to mitigate this problem is to consider properties that are invariant to transformations [9]. To include intra-class differences, quasi-invariant features [11] have emerged that perform effectively under small deformations. Quasi-invariants are features that when subjected to deformation/intra-class variability change much lesser than the value of deformation itself. Consequently, quasi-invariant shape descriptors can be applied to represent object categories.

In this work, a quasi-affine invariant descriptor is used to encode line-based substructures, which forms the first level of object representation. A substructure is defined as a quadruple of contour lines from the object silhouette. Thus, a substructure essentially describes a *part* of the object. Such a representation is advantageous for several reasons. By modeling separate parts of the object, we allow inter-part flexibility, in the sense that a large global deformation of the entire object is decomposed into small local deformations of the substructures that can be captured by the quasi-invariant description. This summational effect of deformation aids recognition of object instances significantly different from the training set. Another natural consequence of the local model is the insensitivity to partial occlusion.

In the second level of shape modeling, the object shape is represented as a connected graph of the substructures. Each line represents a node and each substructure is represented by a *clique*. So the object itself is a clique with every line associating with every other line to form a substructure. To localize the object in a test image we exploit the property that identified object substructures will strongly connect to one another. These connected substructures can then be identified by using a graph partitioning technique that extracts the higher density partition.

Structural shape has been used before as a robust estimator for object localization [4] [5] [1]. For textureless objects, linear and higher dimensional contours [10]

are preferred to region-based patches. Pairwise relationships between lines have been exploited to identify geometric associations but they perform weakly in presence of noise or require an additional layer of processing [3]. In [7], rectangular objects are recognized using appearance similarity of lines.

Geometric invariants have been used in prior research on structure based recognition but they have mostly focused on specific objects under viewpoint changes. E.g. in [8], affine invariant matching of 3-D objects was done with geometric hashing. In [11], a deformation invariant encoding was proposed for modeling polynomial curves.

Structural affinity has been formulated as grouping algorithm in [2] involving a connected component analysis on an affinity matrix of pairwise relations. However, it is used only as a postprocessing step to reject mismatches in appearance-based correspondences.

A major contribution of this paper is to define generic object shape by integrating a local canonical invariant representation with a global structural graph (Figure 1).

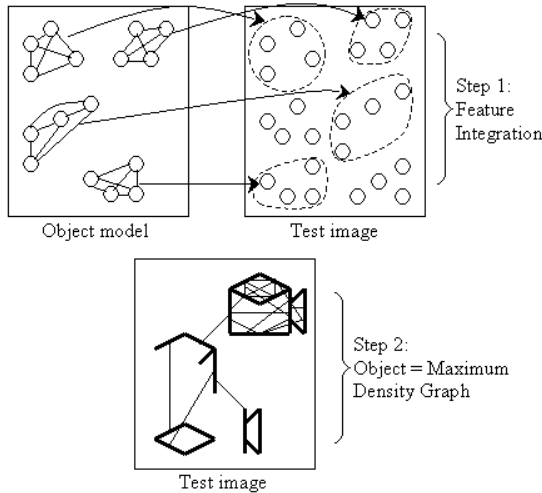


Figure 1: Basic Framework.

2. Substructure Representation

In this section, we describe the first level of the algorithm, namely, the substructures used to represent the object. We first explain the affine invariant description of object substructures. Then we elaborate on the method employed for finding correspondences between line quadruples in a test image and the model substructures from the training set.

2.1. Affine Invariant Canonical Description

Given a set of straight contour lines of an object, a substructure is defined as a quadruple of lines from the set. We represent the substructures by a canonical affine-invariant descriptor inspired from [12]. The quadruplicity is a necessary condition for the affine invariance that is imposed on the structure.

The input to this method is a set of contour lines that are parameterized into polar coordinates (r, θ) such that each quadruple is denoted by four lines $e_i = (r_i, \theta_i)$, where $i = 1 \dots 4$. A transformation T maps three of the lines e_1, e_2, e_3 to a canonical basis with line equations i.e., $x = 0$, $y = 0$ and $x + y = 1$ respectively. The value of T is then determined and applied to e_4 . The transformed coordinates of the 4th line becomes, -

$$\hat{\theta} = \tan^{-1} \frac{A \csc(\theta_1 - \theta_3) \sin(\theta_1 - \theta) \csc(\theta_1 - \theta_2)}{A \csc(\theta_2 - \theta_3) \sin(\theta_2 - \theta) \csc(\theta_1 - \theta_2)} \quad (1)$$

$$\hat{r} = \frac{1}{\tau} (r_1 \csc(\theta_1 - \theta_2) \sin(\theta_2 - \theta) - r_2 \csc(\theta_1 - \theta_2) \sin(\theta_1 - \theta) + r) \quad (2)$$

where

$$\tau = |A \csc(\theta_1 - \theta_2)|$$

$$\cdot \sqrt{\csc^2(\theta_1 - \theta_3) \sin^2(\theta_1 - \theta) + \csc^2(\theta_2 - \theta_3) \sin^2(\theta_2 - \theta)}$$

and

$$A = r_1 \sin(\theta_2 - \theta_3) + r_2 \sin(\theta_3 - \theta_1) + r_3 \sin(\theta_1 - \theta_2)$$

As per the above formulation, each ordered set of line-quadruples (e_1, e_2, e_3, e_4) where (e_1, e_2, e_3) is the basis, generates a canonical representation $(\hat{r}, \hat{\theta})$ corresponding to e_4 .

Given a training set of images with labeled object contours, the object model M is a distribution in $\langle R, \Theta \rangle$ values arising from each ordered set of edges of each image. This quasi-invariant model inherently captures the deformations due to intra-class variabilities within the training set.

2.2. Substructure Matching

During test phase, the configuration of a line quadruple (e_1, e_2, e_3, e_4) from an observed image is defined by its canonical representation $p = (\hat{r}, \hat{\theta})$ as described above. The object substructure that is most similar to this configuration is found by considering the K-nearest Euclidean neighbors in the model M . The matching score for each label is the ratio between the number of points to average distance to those points. The test quadruple gets classified as the label of the model substructure with maximum the maximum score.

The matching scores $w_{p,i}$ are scaled in the range of $[0, 1]$ to ease future calculations.

2.3. Cyclic Invariance and Label Consistency

One-to-one correspondences determined between test quadruples and object substructures is prone to erroneous or multiple matches. A way to solve this problem is to apply a Consistency Constraint on the decisions. Note that ordered sets of edges include all cyclic permutations of the edge-basis pairs that is, 4 matches are found for the same set of edges at different configurations. If the matches are indeed correct then the labeling decisions should be equivalent in all the cases, i.e., $((e_1|e_2, e_3, e_4) \equiv (e_4|e_1, e_2, e_3) \equiv (e_3|e_4, e_1, e_2) \equiv (e_2|e_3, e_4, e_1))$. Specifically, the labeling of individual lines L_i and the weights associated with match w_i should be same for all the 4 permutations. To realize this consistency constraint in a probabilistic sense, we compute an equivalence score $Q(E)$ for all the permutations as follows:

Let $I(e_i = l_j|k)$ be a binary indicator function to define that line e_i is assigned to label l_j in the k^{th} permutation. We also define an accumulator matrix AM that accumulates the confidence with which each test line is associated with each object label.

$$AM(l_j, e_i) = \sum_{k=1}^4 I(e_i = l_j|k^{th} \text{ permutation}) \cdot w_k \quad (3)$$

where w_k is the matching score as calculated in the above section.

Then equivalence score of any set of four lines E is computed as-

$$Q(E) = \sum_{i=1}^4 \max_i (AM(l_j, e_i)) / 4 \quad (4)$$

The above procedure can be simply explained as follows - If a hypothetical line e is assigned the same label l in all 4 permutations with $w = 1$ then the accumulator value becomes $AM(l, e) = 4$ and the equivalence score Q is maximized to indicate a strong one-to-one match between the line and the label. Thus in the implementation, if the value of Q exceeds a certain predetermined threshold (We set (0.75) for our experiments, which is intuitively equivalent to 3 consistent matches out of 4 lines in a substructure) then the test quadruple E is considered a valid match with confidence Q . The valid matches with the associated confidences are enlisted in a set S . This set is a collection of all substructures in the test image that have high resemblance to the object. However, only some of these structures can combine to form a composite object. We explain the method to extract these structures in the next section.

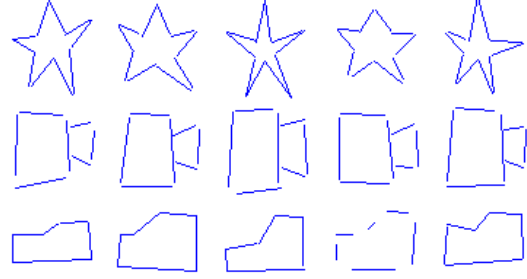


Figure 2: Training images for star, mug and SUV shapes.

3. Combining sub-structures

Substructures may not be unique to an object. For example, a square shape of a mug handle (a substructure in the mug object to be detected) can also occur in background objects like computer screen. Therefore, we need to consider the interaction between substructures to localize the object itself. The intuition behind our approach is that substructures belonging to the object have higher affinity amongst themselves than background clutter. This idea forms the backbone for our object detection. To facilitate separation based on affinities, we view substructure interaction as a graph partitioning problem in which the lines in the test image are the nodes and the substructures form cliques of the graph. This graph is partitioned into two clusters based on the density of connections. The object structure comprises of lines from the cluster with higher connection density. The details of the algorithm are as follows.

3.1. Graph Partitioning and Object Localization

We create the connected graph of substructures as follows. The list S from previous section is transformed into a pairwise weighted graph $G(\hat{V}, \hat{E})$ where the vertices \hat{V} correspond to lines and the weights between two vertices \hat{E} are determined by summing their co-occurrence weights in the list S .

$$W(v_i, v_j) = \sum_k Q(E_k | e_i, e_j \in E_k) \quad (5)$$

Partitioning the above graph produces two clusters in which the strongly connected cluster is separated from the weakly connected one. By our present formulation, the object cluster corresponds to the higher density graph. The density of a graph is measured as weighted cardinality of edges to vertices ratio.

$$d(\hat{G}) = \sum_{i,j \in \hat{G}} W(v_i, v_j) / |\hat{V}| \quad (6)$$



Figure 3: Comparing object and background cluster densities. Top: Images with overlaid test lines. Bottom-left: Density of clusters $d(Object) = 1.76, d(Bkgrnd) = 1.56$. The small difference in values is due to the structural similarities between mug shape and the background (e.g. alphabet U, G etc.). Bottom-right $d(Object) = 2.37, d(Bkgrnd) = 0.06$; the object forms a much stronger cluster than the background.

A multi-resolution graph partitioning method is well suited for density based partition. We applied the metis algorithm [6] that performs local search at several levels in a hierarchical fashion. In general, the object detection method is independent of the choice of partitioning algorithm.

4. Results

We tested our algorithm on three object classes - mug, star and car(SUV) shapes. These objects were chosen because of the distinct structures that can be represented using line segments.

The experiments were specifically designed to understand the behavior of the algorithm and isolate its performance under different conditions. We used the kAS line segment detector [5] for contour line extraction in our experiments. The model data consists of 5 hand drawn shapes (see Figure 2) and 5 versions of each of them, where Gaussian noise with 0 mean and covariance σ^2 is added to the end points of the line segments with σ in the range of 1 to 5.

In the test images, we deal with generic object instances (not in the training data) subjected to affine transformations in cluttered background. The average number of test lines was 30 out of which 65% – 70% of lines were background clutter. We evaluate our model on 10 images from each object class (see Table 1). Density of the object cluster was always higher than the background cluster so test accuracy was computed as accuracy in the chosen cluster = $(true_positives + true_negatives)/(number_of_lines)$. Mean training

accuracy was computed as the number of lines in the object models (25 object models) getting labeled correctly.

We evaluate our algorithm for robustness to affine transformations, ability to detect multiple objects and with background clutter and occlusion. Figure 3 compares the object-background cluster density difference between two images. We notice that the difference depends on the distinctness of the object compared to the background. 4 shows the result with the original and the affine transformed image. We see that object was detected even after significant affine transformation. Moreover, the internal clutter (the alphabets) were rejected correctly. 5 shows the detection with multiple objects of same category. Our clustering algorithm was able to isolate the two strongly connected object groups belonging to the different instances. This highlights the effect of the second level of processing where the substructures were similar but grouped in different clusters because of substructure affinities within the object. Also in 5, the same test is done with different object categories mug and star. Correct object clusters are detected based on the search criteria. 6 shows object detection with structured background (lines form other objects) and unstructured clutter (with no coherent form). The object is extracted out from both types of background. Finally, 7 illustrates the results with occlusion. We see that the combination of substructure modeling and global graphical scheme makes the algorithm capable of handling real conditions.

5. Conclusions

The experiment outcomes illustrate that the algorithm is able to extract the object from structured and unstructured background. In our experiment, we learn the object model from hand-drawn shapes which illustrates that structure in real images can be learned from few basic hand sketches. The model can also be learnt from object contours from real data. The algorithm can also handle fragmented or redundant lines. The method is robust to affine transformations, intra-class variability and partial occlusion under realistic conditions. Our algorithm can be used for localized selection of bottom appearance-based features and paves the way for integrating top down and bottom up approaches for generic object recognition. However, the most powerful concept arising from this work is that of defining an object model from simple substructures. These substructures can form a *shape alphabet* using which a complete object *shape vocabulary* can be defined and used for generic object representation, search and recognition. Our algorithm provides an novel approach to meet



Figure 4: Affine Invariance. Left: Original image with overlaid test lines. Middle/Right: black lines: object cluster, gray lines: background

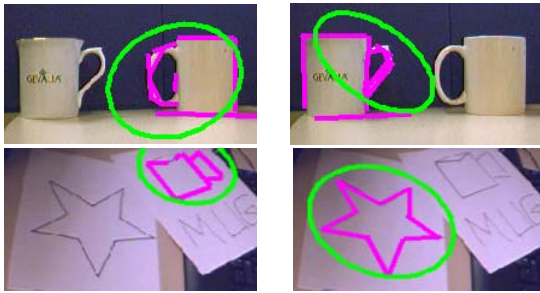


Figure 5: Multi-object detection. Top: Two clusters separate out two instances of same object. Bottom: Detection with mug model (left) and star model (right).



Figure 6: Left: Object cluster, Right: Background cluster. Detection with structured (top) and unstructured (middle, bottom) background clutter.

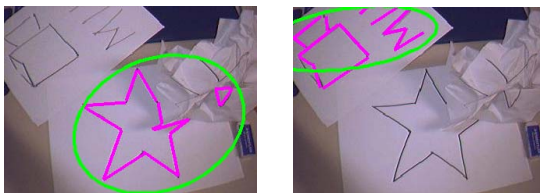


Figure 7: Left: Occluded object cluster, Right: background.

Object class/Accuracy	Car SUV	Star	Mug
Training	0.86	0.89	0.88
Object class/Accuracy	Car SUV	Star	Mug
Test	0.83	0.84	0.86

Table 1: Results.

that end.

References

- [1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522, April 2002.
- [2] G. Carneiro and A. Jepson. Flexible spatial configuration of local image features. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 29(12):2089–2104, December 2007.
- [3] P. David and D. DeMenthon. Object recognition in high clutter images using line features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages II: 1581–1588, 2005.
- [4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II: 264–271, 2003.
- [5] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.
- [6] E. Karypis and V. Kumar. hmetis: A hypergraph partitioning package. Technical report, Department of Computer Science, University of Minnesota, MN, 1998.
- [7] G. Kim, M. Hebert, and S. Park. *Preliminary Development of a Line Feature-Based Object Recognition System for Textureless Indoor Objects*. Elsevier, 2005.
- [8] Y. Lamdan, J. Schwartz, and H. Wolfson. Object recognition by affine invariant matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 335–344, 1988.
- [9] J. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.
- [10] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages II: 575–588, 2006.
- [11] E. Rivlin and I. Weiss. Deformation invariants in object recognition. *Computer Vision and Image Understanding (CVIU)*, 65(1):95–108, January 1997.
- [12] F. Tsai. A probabilistic approach to geometric hashing using line features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1993.