Finding Multiple Object Instances with Occlusion

Ge Guo*+, Tingting Jiang+, Yizhou Wang+, Wen Gao+
+Nat'l Engineering Lab for Video Technology, Key Lab. of Machine Perception (MoE),
Sch'l of EECS, Peking University, Beijing, 100871,China
*Inst. of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
*Graduate University, Chinese Academy of Sciences, Beijing 100039, China
gguo@jdl.ac.cn, {ttjiang, Yizhou.Wang, wgao}@pku.edu.cn

Abstract

In this paper we provide a framework of detection and localization of multiple similar shapes or object instances from an image based on shape matching. There are three challenges about the problem. The first is the basic shape matching problem about how to find the correspondence and transformation between two shapes; second how to match shapes under occlusion; and last how to recognize and locate all the matched shapes in the image. We solve these problems by using both graph partition and shape matching in a global optimization framework. A Hough-like collaborative voting is adopted, which provides a good initialization, data-driven information, and plays an important role in solving the partial matching problem due to occlusion. Experiments demonstrate the efficiency of our method.

1. Introduction

We study the problem of detecting and locating multiple object instances of similar shapes from an image. It is a challenging vision task to find and separate the target objects that matched with a template example, because of the various transformations, deformations and especially occlusion between objects. There are generally three challenging problems. Firstly, due to the similarities in appearance and the overlap between objects in such images (e.g. Fig. 1), it is tough for object detectors based on image patches. Considering the shape similarities, the primary problem is to figure out the matching, transformation and deformation between shapes. Second, when occlusion happens traditional shape matchers always get into trouble. The challenge is how to optimally match the incomplete shape parts to the template. Finally in a cluttered image with noisy distracters, an unknown number of target object instances which might overlap are difficult to detect and segment.

In the literature, classical shape matching methods aim at calculating the correspondence and transformation between two shapes, such as the Shape Context [3] and TPS-RPM [5]. They are able to solve the first problem above. Recently, shape matching and object detection have been more actively studied and combined together to deal with more challenging situations. For example, the proposed Contour Context Selection [11] explores salient shape features and contextual relationships among shape parts to improve shape matching. Ferrari et al. [6] detect objects by learning shape models of object classes. However most of the previous related work mainly focuses on the one-to-one shape matching problem. Although in [6] some detected results of multiple objects are shown, there is little occlusion; it does not provide a globally optimized method and requires training data to learn the class-specific shape models and intra-class variations in advance.

Lin et al. [9] provide a graph matching method to find matched structures, taking advantage of collaborative and competitive interactions. Nevertheless they have not yet considered the problem of finding multiple similar object instances in challenging situations. Besides, another related work on texture segmentation [1] is reported to extract 2.1D texels. However, it runs in an unsupervised way without any information about the target object template. When the object consists of multiple texels, it would be hard to decide whether a segmented element is the whole object or a part of it.

In this paper we present a Bayesian optimization framework utilizing graph partition and shape matching to detect and locate multiple occluded object instances, given a predefined template. Note that the number of matched objects is to be estimated; there exist noisy distracters, as well as occlusion which leads to partial matching between each object and the template. We

solve this problem as follows. (1) Initialize the graph partition by a Hough-like voting, which utilizes the collaborative relationships between object parts in the space of affine transformations; (2) Match each potential target object to the template based on the TPS-RPM method [5], in which the collaborative affine transformations generated as in (1) are used to cope with the partial matching problem; (3) Solve the graph partition problem under the Swendsen-Wang Cuts framework [2], in which the collaborative relations are used as data-driven information. We further accelerate the algorithm by adding another two dynamics of the Markov chain – birth and death.

In the rest of the paper we introduce our method of graph partition based on shape matching in Section 2. The experimental results are demonstrated in Section 3. Finally Section 4 concludes the paper.

2. Approach

Given an input image containing multiple similar object instances and a predefined template as the reference (in this paper it is manually labeled by the user), our purpose is to find all the matched object instances in the image. We formulate this problem by graph partition and labeling, based on shape matching between each partitioned subgraph and the template.

2.1. Problem formulation

We construct a graph $G = (V, E)$ based on the shape features obtained from the input image. Specifically the shape features are edge curvelets based on the Berkeley edge detector [10]. Each graph node in V represents a curvelet. E denotes the edges between neighboring graph nodes. Similarly the template is represented by a graph T, by dividing the template contour into several small curvelets as basic shape units for matching.

There exist K potential target object instances in the graph G (K is unknown). Each object is represented as a subgraph $G_i (i = 1, 2, ..., K)$. Let G_0 be the set of all the unmatched noise curvelets. So the partition for the graph G is to be estimated. Besides, in order to match each potential object G_i to the template T, we need to estimate the transformation A_i (here we use the affine transformation for computational conveniences), deformation ω_i and the correspondence M_i of the nodes between G_i and T. The goal is to maximize a posteriori,

$$W^* = \arg\max_W p(W|G, T) \tag{1}$$

where $W = (K, G_0, \{G_i, A_i, \omega_i, M_i\}_{i=1}^K)$.

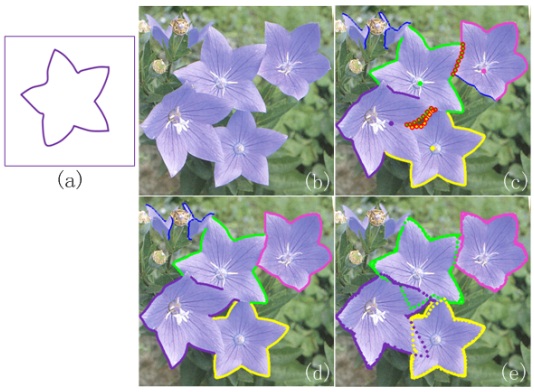

Figure 1. Multiple object recognition and location. (a)The template; (b)The input image; (c)The initial coloring by voting (the curvelets marked by red circles are the ambiguous ones relating to more than one instance); (d)The partition result (with the thin blue ones being noises); (e)The located and completed objects, where the dashed lines are the completed shapes.

2.2. Initialization by collaborative voting

Voting-based methods are popularly used in object detection [8][4][7]. Here we develop a collaborative Hough-like voting in the space of affine transformations. Due to the pose variances among objects and the symmetries of the shapes, each curvelet in the image might be matched to different parts of the template, which generates multiple corresponding affine transformations. This always results in spurious voting. Considering that the parts of the same object should be of consistent affine transformations to the template, our voting scheme is designed under this collaborative constraint. Specifically, we implement the voting in a 3D space spanned by the 2D translation (x- and y-axis) and rotation. This space is uniformly divided into small cells. Each matching between an object curvelet and a part of the template induces a vote in this space.

Each vote is weighted by $w = L \cdot e^{-\frac{\epsilon^2}{b^2}}$ to encourage those of large curvelet length L, and small matching error ϵ (ϵ is the total Euclidean distances between the transformed object curvelet and the corresponding part on the template). b is a controlling parameter set as $b = \bar{L}/3$ ($\bar{L}$ is the average length of the curvelets in the image). Three hierarchical levels of curvelet groups are adopted for voting – the original curvelets, the pairwise and triple-wise curvelets, in order to make the voting more robust to noises. Those over-stretched and mismatched votes with large errors are removed in advance.

The top K' ranked voting results are taken as the initialized locations and transformations of the candidate objects. The related curvelets voting for the same candidate object are colored with the same label. Note

that there may be ambiguous curvelets that are related to more than one object. For such a curvelet, we select the most strongly collaborative object with it, and assign the label of the object to the curvelet. Then we have an initial coloring of the graph.

2.3. Partition based on shape matching

With the above initialization, we adopt the effective sampling algorithm Swendsen-Wang cuts [2] to solve the graph partition problem under the MAP formulation in Eq. (1). According to the Bayesian rule, $p(W|G,T) \propto p(W|T)p(G|W,T)$. The prior model is

$$p(W|T) = p(K)p(G_0)p(G_1...G_K)\prod_{i=1}^{K} p(A_i,\omega_i,M_i|T) \quad (2)$$

where it is expected to find limited number of target objects, $p(K) \propto \exp\{-\lambda_1 K\}$; there exist a number of noises, $p(G_0) \propto \exp\{-\lambda_2|G_0|\}$. The prior on the partition follows the Potts model, in which neighboring nodes should be more likely to have the same label l, $p(G_1...G_K) \propto \exp\{-\sum_{(m,n)\in E}\mathbf{1}(l_m = l_n)\}$. The transformation A_i, deformation ω_i and correspondence M_i are of uniform prior distributions. λ_1 and λ_2 are scaling factors to balance the prior terms (e.g. we set $\lambda_1 = 0.5, \lambda_2 = 0.2$ for Fig. 1).

The likelihood term is measured by the similarity of each potential object to the template. It is computed by their matching cost based on the TPS-RPM method.

$$p(G|W,T) = \prod_{i=1}^{K} p(G_i|A_i,\omega_i,M_i,T) \quad (3)$$

$$p(G_i|A_i,\omega_i,M_i,T) = \exp\{-E_{tps}(G_i,T)\} \quad (4)$$

E_{tps} is the TPS energy, calculated by Eq. (13) in [5].

Due to occlusion as well as the incomplete shapes to be matched during the partition process, there is the partial matching problem, i.e., how to find the corresponding part on the template and match the incomplete shape to it. To solve this problem, we take advantage of the affine transformations induced as in the collaborative voting scheme in Section 2.2. The nodes of G_i are used to vote for a collaborative affine transformation A_i^* to the template. And G_i is transformed according to A_i^*, which generates G_i^* and its corresponding parts on the template T^*. Then the TPS-RPM matching is done based on G_i^* and T^*.

Implementation by the Swendsen-Wang Cuts. To sample a connected component V^*, the probability of turning on an edge is set proportional to the collaborative strength of the nodes (in terms of their affine transformations to the template). As in [2], the Markov chain dynamics – split, merge and regroup are adopted, in which one connected component can be split to generate a new subgraph (i.e. potential object), merged into an existing subgraph or grouped to another subgraph from an old one, respectively. Besides, we add two new dynamics – birth and death to accelerate the algorithm. A new subgraph can be created from the noises and an existing one may degenerate to noises. We use the Metropolis method here. The probability of accepting one transition is $\alpha(W \to W') = \min\left\{1, \frac{p(W'|G,T)}{p(W|G,T)}\right\}$. For each kind of dynamics,

$$\frac{p(W'|G,T)}{p(W|G,T)} = \begin{cases} e^{-\lambda_1+\lambda_2 n}\cdot p(G'_{l'}|T), & \text{birth} \\ e^{\lambda_1-\lambda_2 n}\cdot \frac{1}{p(G_{l^*}|T)}, & \text{death} \\ e^{-\lambda_1}\cdot \frac{p(G'_{l'}|T)p(G'_{l^*}|T)}{p(G_{l^*}|T)}, & \text{split} \\ e^{\lambda_1}\cdot \frac{p(G'_l|T)}{p(G_l|T)p(G_{l^*}|T)}, & \text{merge} \\ \frac{p(G'_l|T)p(G'_{l^*}|T)}{p(G_l|T)p(G_{l^*}|T)}, & \text{regroup} \end{cases}$$

where l^* is the label of the connected component V^* in G, $n = |V^*|$, $l' = K+1$, $l \in \{1,...,K\}$. We use the collaborative transformations as data-driven information. For example, a sampled connected component is proposed to merge or regroup to the most collaborative subgraph; strong collaborative curvelets from noises may birth into a new object, while a subgraph of weekly collaborative nodes may probably die.

When the best partition is sampled, we can recognize and locate the multiple target instances according to their related curvelets based on the labeling results. The occluded parts are completed by transforming the template to the objects, according to the estimated correspondences and transformations (as shown in Fig. 1, 2).

3. Experiments and discussions

Fig. 2 shows some selected results of our experiments. The collaborative voting gives good initializations for coloring, although there are ambiguous curvelets (marked by red circles) and spurious votes. The initialization greatly reduces the computational costs and accelerates the convergence of the algorithm.

In our implementation the matching energy E_{tps} is found to be one of the most important factors that affect the final results. It is easy to fall into local maxima by directly using the TPS-RPM algorithm if the shape is severely occluded; while our strategy provides an effective way to solve the partial matching problem. Meanwhile, because the initial transform provides a good alignment between the potential object instance and the template, we can lower the initial temperature in the TPS-RPM matching for quicker convergence.

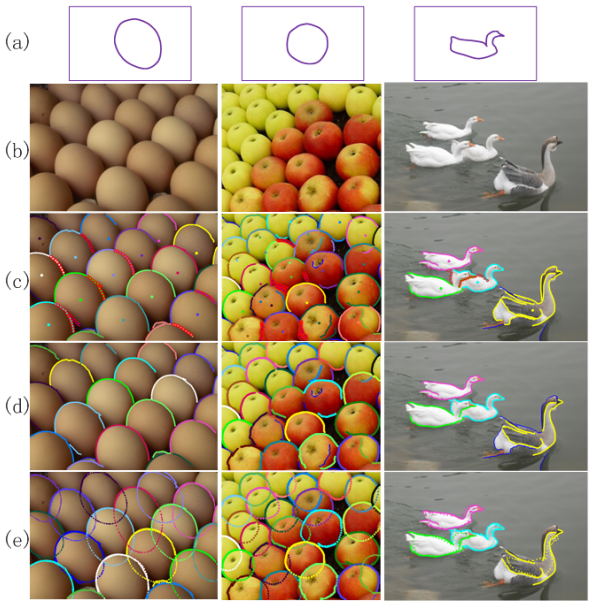

Figure 2. Experimental results. (a)The templates; (b)The input images; (c)The initial coloring by voting; (d)The partition results; (e)The results of located and completed objects.

However, this partial matching strategy tends to relax the penalties on the incomplete matching cases, which may lead to too many spurious matched partial objects. So we add the evaluation of the completeness for each potential object. Also the parameter λ_1 in the prior model can be adjusted to avoid this problem. Nevertheless some noises are labeled to nearby objects by mistake (e.g. in the apple image), for that they help to form more complete shapes very similar to the template. And the underlying occluding relations are not estimated here.

In the goose image it detects a "smaller" one for the right most goose. The reason is some contour curvelets are not detected due to the color similarity between the goose body and the water, which leads the outer contour to be of large matching cost.

Table 1 lists the performance of our method. The ratio of the correct labels is computed based on the number of curvelets with the right coloring. For the Hough voting it excludes the wrong and ambiguous ones. The recognized object rate is computed based on the total area of the correctly localized objects in each image. The missing objects and hallucinated spurious objects are considered as errors.

4. Conclusion

This paper provides a framework to detect, locate multiple object instances that matched a given template.

Table 1. Our average performance

Labels of Hough	Labels of SWC	Recog. objects
78.04%	89.11%	89.82%

A global optimization algorithm based on graph partition and shape matching is introduced. It is effective by the good initialization strategy and data-driven information based on the collaborative voting. One of the limitations is that we currently only use the shape information. In the future we will enrich the object model by adding appearance model, pose estimation to obtain more accurate results.

5. Acknowledgments

We'd like to thank for the support from the 863 Program of China under grant No.2007AA01Z315, the research grant NSFC-60872077, and the Doctoral Fund of MoE of China KEJ200900029. Also thank Wei Wang for the invaluable suggestions in the project.

References

[1] N. Ahuja and S. Todorovic. Extracting texels in 2.1d natural textures. In *ICCV*, 2007.

[2] A. Barbu and S. Zhu. Graph partition by swendsen-wang cuts. *ICCV*, 1:320–327, Oct 2003.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509C522, April 2002.

[4] B.Ommer and J. Malik. Multi-scale object detection by clustering lines. In *ICCV*, 2009.

[5] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 2003.

[6] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *IJCV*, 2009.

[7] L. Gorelick and R. Basri. Shape-based object detection and top-down delineation using image segments. *IJCV*, 2009.

[8] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 2008.

[9] L. Lin, K. Zeng, X. Liu, and S.-C. Zhu. Layered graph matching by composite cluster sampling with collaborative and competitive interactions. In *CVPR*, 2009.

[10] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[11] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, 2008.