Recognizing objects by piecing together the Segmentation Puzzle

Timothee Cour  
University of Pennsylvania

Jianbo Shi  
University of Pennsylvania, jshi@cis.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/cis_papers
Part of the Computer Sciences Commons

Recommended Citation

Timothee Cour and Jianbo Shi, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007  
Digital Object Identifier: 10.1109/CVPR.2007.383051
©2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/cis_papers/549
For more information, please contact repository@pobox.upenn.edu.
Recognizing objects by piecing together the Segmentation Puzzle

Abstract
We present an algorithm that recognizes objects of a given category using a small number of hand segmented images as references. Our method first over segments an input image into superpixels, and then finds a shortlist of optimal combinations of superpixels that best fit one of template parts, under affine transformations. Second, we develop a contextual interpretation of the parts, gluing image segments using top-down fiducial points, and checking overall shape similarity. In contrast to previous work, the search for candidate superpixel combinations is not exponential in the number of segments, and in fact leads to a very efficient detection scheme. Both the storage and the detection of templates only require space and time proportional to the length of the template boundary, allowing us to store potentially millions of templates, and to detect a template anywhere in a large image in roughly 0.01 seconds. We apply our algorithm on the Weizmann horse database, and show our method is comparable to the state of the art while offering a simpler and more efficient alternative compared to previous work.

Disciplines
Computer Sciences

Comments
Timothee Cour and Jianbo Shi, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007

Digital Object Identifier: 10.1109/CVPR.2007.383051

©2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Abstract

We present an algorithm that recognizes objects of a given category using a small number of hand segmented images as references. Our method first over segments an input image into superpixels, and then finds a shortlist of optimal combinations of superpixels that best fit one of template parts, under affine transformations. Second, we develop a contextual interpretation of the parts, gluing image segments using top-down fiducial points, and checking overall shape similarity. In contrast to previous work, the search for candidate superpixel combinations is not exponential in the number of segments, and in fact leads to a very efficient detection scheme. Both the storage and the detection of templates only require space and time proportional to the length of the template boundary, allowing us to store potentially millions of templates, and to detect a template anywhere in a large image in roughly 0.01 seconds. We apply our algorithm on the Weizmann horse database, and show our method is comparable to the state of the art while offering a simpler and more efficient alternative compared to previous work.

1. Introduction

We consider the task of recognizing a class of objects in a cluttered environment. By recognizing a class of objects, we mean 1) we can provide a precise and complete object mask (so we can evaluate its shape), and 2) we can provide a labeling of the object parts. We not only seek to classify an object, but also to know its detailed shape and parts. In this work, we will use the horse recognition task of the Weizmann Institute database as our example.

The fundamental difficulty of this mid-level recognition task lies in the fact that the objects (even within a class), can be highly variable in appearance and shape. While generative top-down model approaches can be used in principle, it requires infeasible amount of hypothesis in practice. A blindly generated object hypothesis has few chances of being correct. An alternative approach is to start from the image itself, and construct an object hypothesis using a bottom-up approach, such as image segmentation. However, image segmentation faces its own difficulties with faint or missing contours, which results in either over or under segmentation (with respect to correct object mask).

The main challenge is how to combine bottom-up segmentation and top-down object model to efficiently generate a (small) set of feasible object hypothesis. In this work, we assume there is a fixed over-segmentation of the image, and a set of object models broken down to its parts. Our approach has two main components. First, we developed an extremely fast indexing mechanism for object parts, that effectively searches over the space of all combinations of image segments and affine part transformations for the optimal part alignment. This is crucial since we assume our object has been over segmented into smaller pieces. Second, we designed a contextual interpretation of the parts, gluing image segments using top-down fiducial points, and checking the overall shape similarity. This is important since object shape similarity is a highly non-metric, non-Markov function. Measuring the whole shape can be very different from measuring the sum of its parts in isolation. Without this ability to put together parts correctly, and registering that with a model, we run the risk of not being able to evaluate the hypothesis correctly.

2. Previous work

The problem of class-specific segmentation has been the focus of a large number of recent papers, including the following [8, 13, 12, 2, 5, 15, 11, 15]. The main challenge arises from integrating bottom-up and top-down cues in a coherent and efficient manner. In [8], this challenge was addressed by a greedy split-and-merge scheme that groups oversegmented patches of an image guided by fitness to a deformable template. In [11], the low-level (soft) segmentation cues are tuned in an iterative manner along with the estimated pose using a pictorial structure. In [10, 9], baseball player images are segmented into regions, from which body parts are detected using a heuristic combinatorial search procedure. For example, the head is searched among all
Several approaches based on Markov Random Fields (MRFs) or Conditional Random Fields (CRFs) were recently proposed, most involving difficult inference problems that must be addressed via sampling or approximate inference based on message passing or graph cuts. In the image parsing framework of Tu et al. [13], Monte-Carlo sampling over possible segmentations is used, with a proposal distribution driven by object-specific detectors. In the OBJ-CUT algorithm [5], a layered pictorial structure is used to define an energy term for a graph-cuts energy minimization algorithm that favors boundaries at image discontinuities. Recently the authors of [6] have proposed an CRF based segmentation, where the smoothness term uses low-level segmentation cues, and the data term is derived from the top-down detection of a few templates, found via normalized cross-correlation. They report good results on the Weizmann horse database; however, their method requires the object to be roughly centered in the image, and seems to suffer when the object is composed of multiple colors. Our approach does not have such restrictions.

Borenstein et al. [2] use a library of image fragments in combination with bottom-up criteria [3] to cover a new image by using regions that are close based on intensity. Dependence on intensity however requires a large fragment library to capture the diversity of appearance across a class of objects. In [1], the authors integrate in a Bayesian framework a bottom-up segmentation prior with a top-down shape detection using stored shape fragment templates in an approximate, non-iterative scheme. Their approach is the most similar to ours, yet our objective is simpler and more efficient to compute, requiring no approximations.

An important component of our approach is an efficient method for shape detection. There is a rich literature on detecting parameterized shapes under different affine transformations. Perhaps the most popular is an elegant technique of Hough [4] and its generalizations. The Hough method transforms points in the image space into a parameter space, where the maxima correspond to shape detections. The Hough transform is relatively insensitive to noise and occlusion but is computationally and memory-wise intensive as the number of shape parameters increases.

For non-parametric shapes, gray-level template detection using normalized cross-correlation (NCC) is a commonly used technique, often as a basic component of a more complicated scheme. It has the advantage of being invariant to the mean and variance of intensity in a given image window, and can be computed relatively efficiently [7]. However, there are fundamental limitations, as we will illustrate later in the paper when comparing it to our approach.

### 3. Our Approach

Given an input image and a set of training templates, we first compute an oversegmentation of the image into \( k \) superpixels. Our goal is to find a combination of superpixels that is as close as possible to one of the training templates, using Hamming distance modulo affine transformations. This provides both a segmentation of the image and an affine registration to the template. Our search space is very large (exponential in \( k \)), but we will see how the objective decomposes, leading to a very efficient segmentation/detection scheme. Our approach is summarized in figure 1.

The core template detection algorithm uses three optimization components: 1) the exponential search over superpixel combinations can be decoupled, leading to a correlation score at every location, between each superpixel and each template. 2) a discrete version of Green’s theorem can be used to compute the correlation score, which is much faster than using naive correlation or even convolution with Fast Fourier Transform (FFT). 3) this is coupled with a hi-
3.1. Templates for shape detection

We hand labeled the first 20 images in the Weizmann database, see figure 2. We used the ground-truth figure ground segmentation as a starting point, and a simple drawing tool to further delineate 12 body parts: torso, head, neck, tail, upper and lower limbs (×4). In our experiments, we only used either the whole object as a template, or 4 of its constituent parts: torso, head + neck, front pair of legs, rear pair of legs, and we dropped the tail, which is not always visible.

4. Cost Function for Segmentation and Shape Detection

We will define here the cost function used to compare a template to a combination of superpixels.

Definitions and notations We represent a shape $X$ as a subset of the image plane $\mathbb{R}^2$, composed of 2D coordinates $(u, v)$ (or pixels $i$ in the discretized case). If $T : \mathbb{R}^2 \to \mathbb{R}^2$ denotes a transformation of the plane, $T(X) \equiv \{(u, v) : (u, v) \in X\}$.

Shape distance function We define an error measure between two shapes $X, Y$ that has certain invariance properties as we explain below:

$$d(X, Y) = \min_{T \in T} \frac{1}{|T(Y)|} d_H(X, T(Y)), \quad (1)$$

where $|Y|$ is the area of $Y$, $d_H(\cdot, \cdot)$ is the Hamming distance between two sets, and $T$ is a set of transformations. When $T$ is the set of similarity transforms, $d(\cdot, Y)$ is invariant to similarity transforms as is easily checked. In practice however, it is sufficient to restrict ourselves to combinations of all translations and 100 subsampled affine deformations using multiple scales, rotations and aspect ratios, as we do in our experiments.

Detecting a template shape in a segmented image We use the superpixels as shape building blocks and search for the combination(s) that best fits the template. We represent a segmentation of an image as a set of $k$ regions or superpixels $(R_j)_{j=1,k}$. We seek a subset of superpixels $(R_j)_{j \in S}$ that minimizes the shape distance to a given template $Y$:

$$\epsilon(Y) = \min_{S \subseteq \{1...k\}} d(R_S, Y), \quad (2)$$

where $R_S = \cup_{j \in S} R_j$. As this equation implies, we are searching simultaneously for the best combination of superpixels together with the best registration of the template.

4.1. Computational Solution

This is a combinatorial problem, and appears to be intractable in this form. We will show that this problem can in fact be solved efficiently, by factoring out the contribution of each region to the shape distance. Let us represent a set $X$ as an indicator vector $x = I(X) \in \{0, 1\}^n$, where $n$ is the number of pixels in the image: $x_i = 1$ if $i$ is a foreground pixel, $x_i = 0$ otherwise. We also rewrite the segmentation as a $n \times k$ indicator matrix $r$ with columns $r_j = I(R_j)$. Likewise, define $s = I(S) \in \{0, 1\}^k$. We extend all notations from sets to indicators: when $x = I(X), y = I(Y)$, we define $d(x, y) \equiv d(X, Y), T(x) \equiv I(T(X)), \epsilon(y) \equiv \epsilon(Y)$. We will show that $\epsilon(y)$ decomposes as a sum of truncated dot products, and can be computed by truncated convolutions.

Proposition 4.1 ($\epsilon(y)$ is decomposable)

$$\epsilon(y) = 1 + \min_{T \in T} \frac{1}{|T(y)|^2} \sum_{j} \sigma_j(T(y)) \quad (3)$$

where $\| \cdot \|$ is the $L_2$ norm in $\mathbb{R}^n$ and

$$\sigma_j(y) \equiv \min_{s_j \in \{0, 1\}} s_j(|R_j| - 2r_j^T y) = \min(0, |R_j| - 2r_j^T y) \quad (4)$$
The intuitive explanation is that we want to include a superpixel \( R_j \) in the foreground \( (s_j = 1) \) whenever its overlap with the transformed template \( r_j^T T(y) \) is greater than half of the area \( |R_j| \) of that superpixel.

The optimization problem is not only tractable, but it can be computed efficiently by computing **convolutions**, as we explain here: decompose \( T = T_{\text{translation}} \times T_{\text{deform}} \), distinguishing the pure translation part from the deformation part (rotation, scale, sheer). Representing \( r_j, y \) as functions defined over the discretized image plane, we have:

\[
\epsilon(y) = 1 + \min_{T \in T_{\text{deform}}} \frac{1}{||T(y)||^2} \min_{i \in (1..n)} \sum_j \min(0, g_j(T(y)))_i
\]

with \( g_j(y) = |R_j| - 2r_j \cdot \hat{y} \). This gives a total complexity \( O(T_{\text{deform}}|kn \log n|) \) with FFT. Although this would give a reasonable running time given the search space size, we will show next that we can do much better.

**Proof of proposition 4.1** We can rewrite the shape distance as

\[
d(x,y) = \min_{T \in T} \frac{1}{||T(y)||^2} ||x - T(y)||^2,
\]

Since the \( R_j \) have disjoint support, we have \( I(\cup_j R_j) = \sum_j r_j \), and we simplify the cost function (2):

\[
\epsilon(y) = \min_{s \in \{0,1\}^k} \min_{T \in T} \frac{1}{||T(y)||^2} ||rs - T(y)||^2
\]

\[
= 1 + \min_{T \in T} \frac{1}{||T(y)||^2} \min_{s \in \{0,1\}^k} ||rs||^2 - 2T(y)^T rs
\]

Next, notice \( ||rs||^2 = \sum_j s_j |R_j| \) since the \( r_j \) are orthogonal and \( s_j^2 = s_j \), and also \( T(y)\cdot rs = \sum_j s_j r_j^T T(y) \), which shows the cost function decomposes as promised. \( \square \)

### 4.2. Efficient computation using Green’s theorem

We show here that the particular form of equation (4) leads to an efficient random access computation of \( \sigma_j \). This is particularly important, as it allows one to do a coarse to fine search as we will see in the next section. That would not be possible with the method based on convolution we just described, as it cannot avoid the \( n \log n \) Fast Fourier Transform step. The idea we are using is the following:

\[
r_j^T y = \iint_{R_j} y = \iint_{R_j} \frac{\partial \mathbf{I}_u y}{\partial u} = \int_{\partial R_j} \mathbf{I}_u y,
\]

where \( \mathbf{I}_u y(u,v) \) denotes the signed region boundary or opposite gradient in the first dimension, with signed indicator vector \( I(\partial R_j)(u,v) = r_j(u,v) - r_j(u+1,v) \). Once again we assimilate vectors with functions defined over the discretized image plane. Note, this discrete version of Green’s theorem does not make any assumption on the domain \( R_j \), in particular the regions could have holes and be disconnected. Figure 3 illustrates this equation.

![Image illustrating efficient dot product computation](image)

Each \( \sigma_j(y) \) can therefore be computed in \( O(|\partial R_j|) \), since computing \( r_j^T y \) is the bottleneck. Note, we do not need to compute all the \( \sigma_j(y) \) for each region \( R_j \), but only the ones for which \( R_j \) intersects the template \( y \). This can be done efficiently by computing offline the bounding box of the template and the integral image transform (c.f. [14]) of each region. This determines in constant time whether a given region intersects the rectangle hull of the template. Assuming the regions boundaries are not too circumvoluted (say with a fractal dimension of 1), we can assume there are approximately \( O(Y/n) \) such regions. When \( y \) is binary valued (which is our case here), we can also switch the roles between \( y \) and \( R_j \), giving a \( O(|\partial Y|) \) computation time for each intersecting region. Finally the running time across all regions is \( O(k + Y/n \cdot |\partial Y|) \) per spatial location and per template deformation, or \( O(|\partial Y| \cdot Y \cdot |\partial Y|) \cdot k \) for the total running time. Note, when \( |Y\cdot|\cdot|\partial Y| < n \log n \), this approach is already guaranteed to be more efficient than the approach we described above based on convolution. But we can further improve the running time.
4.3. Sublinear time template detection using a coarse to fine approach

Our final optimization idea is the following: we compute the objective at a coarse image resolution, and then for each template deformation we extract the set of local minima across spatial locations. We interpolate the position of each coarse local minimum on the original image resolution and follow a greedy descent of the cost function starting from each of those seeds. Since we have a fast random access computation of the objective (i.e. not requiring a global computation such as FFT), this is done easily by evaluating $\epsilon(y)$ at 4 nearest neighbors at each time step. Under certain smoothness conditions of the template and the region shapes, this approach guarantees that we will not miss any good local minimum at the fine level. If we take a coarse to fine scale factor approximately equal to $k$, we obtain a sublinear running time. For $k = 16$ regions, that represents scaling the image dimensions by $\frac{1}{4}$. In practice we scale the image by a factor $\frac{1}{3}$.

Contrast with gray-level template matching Template detection using normalized cross-correlation (NCC) is a commonly used technique for shape detection. It has the advantage of being invariant to the mean and variance of intensity in a given image window. However, it is very sensitive to clutter and faint edges, and a high contrast spurious edge will likely eclipse a faint figure ground edge. Our approach is much more robust to this phenomenon, so long the oversegmentation is detailed enough, see figure 4.

5. Segmentation Templates for Part Based Models

We extend our approach to deal with articulated segmentation templates. The underlying motivation is that parts are usually harder to detect in isolation than inside the context of other parts. Image region boundaries rarely correspond to the boundaries between different body parts, even when they do correspond to the whole object boundary. Unless we are oversegmenting the image (which increases clutter and the false detection rate), it is likely some parts will be merged together. This affects negatively any part based detection scheme.

We follow a different approach, with a non-additive cost function to detect an articulated object, such as a horse. Instead of summing the detection scores for a torso hypothesis and for a head hypothesis, we directly search for deformable segmentation templates and score those hypothesis. Such a deformable template is composed of a fixed part (for example a template already hypothesized in the image), and a variable part, see figure 5. The variable part is deformed by searching through a range of feasible rotations, scaling, and translations, just as in the previous section for our single segmentation templates, but we require that the junction points between the two parts are within a small radius (30 pixels). The junction points are automatically computed from the hand labeled training images used to compute the templates in figure 2, and then backprojected to the image coordinate system under the current template deformation (rotation, translation, etc). We found this procedure to be quite effective to capture enough articulation.

As before, we search over all possible feasible transfor-
mations of the variable template given the fixed template. This can still be done efficiently and exactly with a slight adaptation of our approach from the previous section. Let \( Y_1 \) be the fixed template and \( Y_2 \) the variable template. The combined template is therefore \( Y_{12} = Y_1 \cup Y_2 \) and, letting \( y_1 = I(Y_1), y_2 = I(Y_2), y_{12} = I(Y_{12}) \), we have:

\[
\iiint_{R_{j1}} y_{12} = \iiint_{R_{j1}} y_1 + \iiint_{R_{j1}} y_2 - \iiint_{R_{j1}} y_1 y_2 \tag{8}
\]

The problematic term is \( \iiint_{R_{j1}} y_1 y_2 \). Fortunately there is a trick we can use: \( \iiint_{R_{j1}} y_1 y_2 = \iiint_{Y_1 \cap R_j} y_2 \). Therefore, given a fixed template \( Y_1 \), we can recompute on an outer loop the signed boundary functions of the new sets \( Y_1 \cap R_j \) for each \( j \), and proceed as in the previous section. The running time is multiplied by a factor \( \leq 4 \), as one can show and as we verified in practice.

6. Reranking the detection hypothesis

What we described so far is an efficient mechanism to propose figure ground detection hypothesis of a whole object or of its constituent parts. The final step is to rerank those hypothesis using more elaborate methods. Since our detection returns a shortlist of local minima of the objective, we only need to evaluate a small number of reasonable hypothesis (typically less than 40), and computation time is less of an issue in that case. The cues we are using are: 1) the original objective function already computed, 2) the edge response at the detected figure ground boundary, 3) color uniformity (standard deviation) inside the figure ground mask. We normalized each cue by its variance across the detections and just added the scores (subtracted for the edge cue) without further optimization. In future work we plan on improving this step, but the focus of this paper was on the generation of figure ground hypothesis using segmentation templates.

7. Results

We experimented on the entire Weizmann horse database, splitting the data into 20 training templates (among the first 23 images) and 308 testing images. We generated offline 100 affine transformations for each template: 5 rotations from \(-\pi/10, \pi/10\), 5 scales from 0.5 to 1 (where 1 represents an object that would fit the entire image), and 4 aspect ratios from 0.7 to 1.3. Multiplying this by the number of parts we are detecting (head, torso, front pair of legs, back pair of legs), we get a total of 20 \( \times \) 100 \( \times \) 4 = 8,000 templates. The images were resized to have a maximal side equal to 250 pixels. The initial bottom-up segmentation was computed using the publicly available code for Multiscale Normalized Cuts\(^1\).

Detection using a template for the whole object, see figure 6. To evaluate our method, and since we advertise our methodology as a hypothesis generation scheme, we used our final scoring function to come up with a shortlist of 10 detections. Out of those, we extracted the one with the highest agreement compared to the ground truth. The average pixel consistency (average percentage of correctly classified pixels) across all 308 images was 94.2\%. In comparison, [1] obtained 93\%. They used 60 training images instead of 20, and their method is significantly more complex than ours. However, they didn’t use the oracle strategy. Note, the best performance possible given the base set of segments we used (60 superpixels) is 96.9\%.

Detection using articulated templates Figure 7 shows an example of articulated detections, sorted by the reranking function. Adequate zooming shows the junction points backprojected from the original templates to the image (we displayed junction points for the 12 part decomposition, although we use only 4 coarser parts). We report once again results using the oracle - best out of 10 methodology, for each of the 4 body parts. Figure 8 shows segmentation and detection results for 26 consecutive test images (much more are provided in the supplementary material). We noticed that while the pixel accuracy improves only by about 1\% (20\% error reduction), the visual appearance of the object segmentation is significantly better across the entire test set, for example in terms of segmentation precision for the legs and the head. This points to the fact that the error measure is biased towards the large torso. A better error measure would be pixel accuracy on each body part separately, normalized by their size, which would require hand labeling the entire dataset.

Efficient storage of segmentation templates The templates we use are piecewise constant. This allows for an efficient lossless compressed storage based on Run Length Encoding (RLE), which scales roughly as \( O(\sqrt{n}) \) in our case for a template over \( n \) pixels. In our implementation, 8,000 large templates fit in only 2 MB, which is orders of magnitude smaller than a naive storage implementation. This is especially important when scaling up the system requires storing millions of templates, for example in a multiclass detection framework.

8. Conclusion

We have presented a simple and effective method for combining top-down shape cues with low-level segmentation. By using only shape information for the class-specific component of the approach, the algorithm remains robust to diversity in appearance of the object and the background, as well as situations with low contrast and illumination. The

\(^1\)http://www.seas.upenn.edu/~timotee/

Figure 8. Alignment and segmentation using articulated templates (4 parts). Columns 1/5 and 2/6: image and over-segmentation into 60 superpixels. Columns 3/7: closest mask to the ground truth label among the top 10 hypothesis computed for each part. Columns 4/8: corresponding foreground.
key contribution of our approach is a time- and memory-efficient method for grouping superpixels and matching them to a large library of shape templates. This shape detection scheme could also be useful for other applications, such as fast shape-based image retrieval and classification.

There are several limitations and challenges to the overall segmentation method we outlined. One of the most important problems is appearance of an object in a vastly different pose from the training data or major occlusions. Another possible detractor is excessive clutter and very faint figure-ground edges that would mislead the initial segmentation into superpixels.

The main purpose of this work is to provide an efficient method for generating a short and high-quality list of complete object segmentation hypotheses. The candidates on this list can then be evaluated using a more elaborate and expensive method, possibly depending on learned global figure ground features (color, texture, symmetries, etc.). We plan to address this in future work.

**Acknowledgements** We would like to thank Ben Taskar for his help on the paper.

**References**


