Inducing Semantic Segmentation from an Example

Yaar Schnitman[1], Yaron Caspi[1], Daniel Cohen-Or[1], and Dani Lischinski[2]

[1] Tel Aviv University, Israel
{caspiy, dcor}@tau.ac.il
[2] The Hebrew University of Jerusalem, Israel
danix@cs.huji.ac.il

Abstract. Segmenting an image into semantically meaningful parts is a fundamental and challenging task in computer vision. Automatic methods are able to segment an image into coherent regions, but such regions generally do not correspond to complete meaningful parts. In this paper, we show that even a single training example can greatly facilitate the induction of a semantically meaningful segmentation on novel images within the same domain: images depicting the same, or similar, objects in a similar setting.

Our approach constructs a non-parametric representation of the example segmentation by selecting patch-based representatives. This allows us to represent complex semantic regions containing a large variety of colors and textures. Given an input image, we first partition it into small homogeneous fragments, and the possible labelings of each fragment are assessed using a robust voting procedure. Graph-cuts optimization is then used to label each fragment in a globally optimal manner.

1 Introduction

Image segmentation, the process of identifying homogeneous regions in an image, is a fundamental task in a large number of applications in image and video processing. A particularly challenging instance of image segmentation is the problem of automatically identifying semantically meaningful regions in an image. This problem is often referred to as *image labeling*, since its goal is to associate each pixel in the image with a label denoting a semantically meaningful part.

While the objective of grouping pixels according to color, texture, and other cues has been dealt with in many ways, the challenge of aggregating pixels into segments representing meaningful parts is much harder. This is due to the fact that such parts are often too complex to be characterized using low-level image features, such as color or texture. Furthermore, the semantic interpretation of an image is highly subjective, depending on both the application, and the user. For example, while some applications are concerned with separating a person from the background, others might require the partitioning of a person's body into its various parts, as demonstrated in Figure 1.

In this paper, we present a novel labeling method, which computes a semantically meaningful partitioning of an input image, as induced from one (or more) correctly segmented training image. Both the input and the training image(s)

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 373–384, 2006.

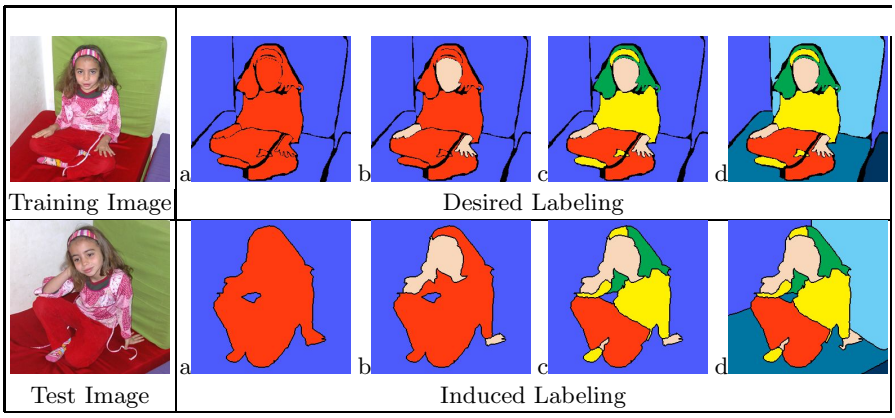

Fig. 1. Inducing different semantically meaningful segmentations. This figure illustrates how four different labelings are induced. A single training labeling is provided each time, all with respect to the train image in left upper corner. Labeling (a) is a binary partitioning between foreground and background. Labeling (b) also distinguishes between skin and clothes. Labeling (c) decomposes the figure into hair and clothes, while labeling (d) breaks up the background into several parts. Note that in general, the various parts cannot be characterized by common image space attributes, and they cannot be inferred without an explicit description or an example.

are assumed to be from the same domain: having similar illumination, resolution and scale characteristics, and depicting similar scenes. The meaningful parts in the training image are recognized in the input image, and the correct assignment of pixels into labels is induced. Such a mechanism is required in various applications, like removal, replacement, or recoloring of a certain object in a series of images. For example, one might want to change the color of a garment worn by a model in all the photographs taken during a particular session.

Our method constructs a *non-parametric* model of the provided training pair by selecting a set of patch-based representatives inside each labeled region in the training image. These representatives are used to quantify the degree of resemblance between small regions in the input image and the labeled regions in the training set. This simple, yet informative representation, which is derived directly from the image, has proved its worth in other applications, such as texture synthesis [1], image analogies [2], recoloring [3], and image and video completion [4, 5]. Here we extend this approach to image labeling.

Image analogies and texture transfer [2, 6] are a general framework by which various types of filters are learned from a single unfiltered and filtered image pair, and induced on novel images. As mentioned above, these methods gain their strength from a simple patch-based sampling scheme. The method we present has a similar flavor and shares their simplicity. However, the former methods cannot induce labeling since the decisions they make are inherently local. In contrast, our method makes use of a global optimization step for finding an optimal pixel labeling.

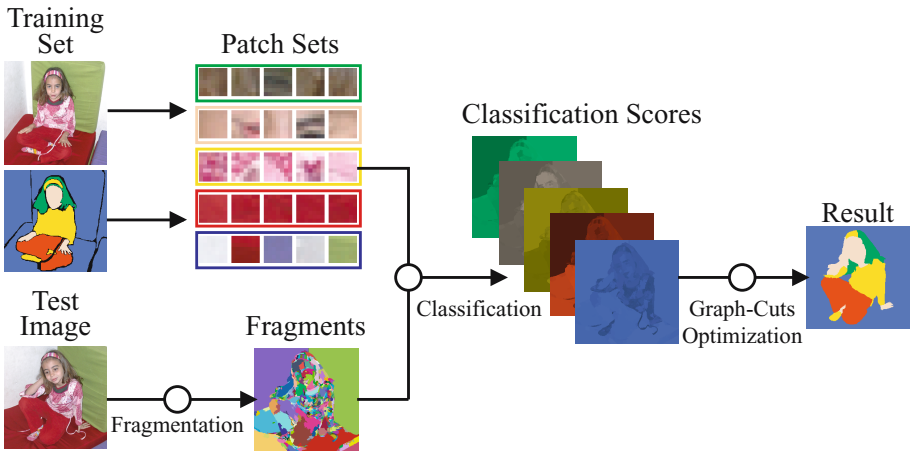

Fig. 2. An overview of our method: The labeled training image is sampled, creating sets of square patches, one set for each label. Given an input image it is first over-segmented into a collection of small homogeneous fragments. Assignment costs are then computed for each fragment-label pair. Finally, a graph-cuts multi-label optimization is used to find the globally optimal labeling of the fragments.

In this work, we assume that small homogeneous regions always belong to the same semantic part of the image. Hence, we over-segment the input image into *fragments*, small arbitrarily-shaped and simply-connected pixel clusters, and compute the labeling at the fragment level. The fragmentation has a profound effect on the final result, as it enforces a locally coherent labeling and facilitates a voting scheme as a means for a robust per fragment label assignment. Furthermore, working at the fragment level reduces the computational complexity and improves performance.

Figure 2 outlines our method. Patch sampling is performed over the labeled training image, defining a set of patches, each representing a labeled region of the image. During labeling induction, the input image is first partitioned into small fragments. Then, assignment cost is computed for every fragment-label pair. Next, these costs together with additional contiguity constraints are incorporated into a graph-cuts multi-label optimization, to yield a global labeling of the fragments. The combination of patch-based sampling, fragmentation, and the graph-cuts optimization results in a segmentation scheme that incorporates both local and global information, allowing effective induction of semantically meaningful labelings from one image to another.

2 Background and Related Work

Segmentation is a well-studied problem. A common approach for segmentation aggregates local cues such as color, texture, edges or various filter responses, by which pixels are clustered into contiguous, homogeneous regions (e.g, [7, 8]. For a survey of segmentation methods, see [9].

While these methods are successful in clustering image pixels into homogeneous regions, they cannot automatically group the resulting clusters into semantically meaningful parts. However, they do provide natural image building blocks, or image fragments, which can facilitate various region based decisions, such as label assignments. We argue that determining whether an entity belongs to a particular semantic part is more easily done at the fragment level, than on a pixel-by-pixel basis.

The limitations of a pixel level decision are also addressed by global methods. By global methods we refer to methods that formulate the problem as a minimization problem over the space of labelings/segmentations. The feasibility of the global approaches is bounded by the exponential complexity of the space of all possible solutions. Therefore, different algorithms restrict the space in order to make the minimization tractable. The restrictions are usually formulated with priors, such as continuity or smoothness. They yield a minimization of an error function comprised of two error terms: the data constraint, and a pairwise constraint. Examples of global methods include normalized cuts [10], belief propagation [11], and graph-cuts [12], which is used in this paper.

Another limitation of previous segmentation methods is the descriptive power of the parametric model that they use to represent a segment, e.g., distribution of colors, textures or some other features. A powerful alternative is to use examples as an implicit representation. Example-based non-parametric modeling avoids the complications of parametric modeling. This approach has been applied successfully in applications ranging from texture synthesis to image completion [1, 2, 6, 13, 3, 4].

An example based representation is also used for detection and segmentation of objects from a specific class [14]. There, the task is to segment an object in an image, based on a large set of pre-segmented images, all from the same family (e.g., horses). In contrast, we are interested in labelings induced by as few as a single example. The image building blocks used in their method are also termed fragments. However, their fragments are rectangular tiles of variable size, while in our work, fragments may have an arbitrary shape determined by the context of the image.

Segmentation is also closely related to the problem of extracting objects from images. Because the task is so challenging, interactive solutions were developed, where the user assists the segmentation process. In particular, graph-cuts optimization has proved to be an effective tool for interactive image segmentation [15, 16]. The optimization is used to find segmentations, which are consistent with color, edges, and the user defined constraints. Graph-cuts have been extended to handle multiple (more than two) segment problems, using the alpha-expansion algorithm [17]. Recent works on video tooning [18] and rotoscoping [19] are related to our work. They also face the problem of producing a consistent segmentation for a sequence of similar images. Their approach takes advantage of frame coherence, computing 3D clusters of pixels in the space-time video volume. The user then outlines the semantic regions using a rotoscoping interface. We are also interested in segmenting similar images, but make no assumptions regarding

coherence among the images, and identify semantic regions automatically based on a small training set.

3 Algorithm

In this section we describe our algorithm for inducing the labeling of the training image onto the test image. Let I_{train} denote the training image and L_{train} the labeling of its pixels by k different labels. Given an input (test) image I_{test} our goal is to compute its corresponding labeling L_{test}. We begin by describing how patches in I_{train} are used to compute labeling costs for pixels in I_{test} (Sec. 3.1). Rather than attempting to label each individual pixel in I_{test} we partition it into small homogeneous fragments (Sec. 3.2) and compute more robust labeling costs for each fragment (Sec. 3.3). Finally, we use graph-cuts optimization to assign a label to each fragment in a globally optimal manner (Sec. 3.4).

3.1 Pixel Labeling Costs

Given I_{train} and L_{train} we create a patch-based classifier by representing each label by a set of square patches, sampled from the corresponding region in I_{train}. We get k such sets $\{S_l\}_{l=1}^{k}$, one for each label. Each set contains a variable number of patches, depending on the number of pixels with that label in I_{train}. All patches are of uniform size $m \times m$, which is chosen beforehand so it is proportional to the scale of details in the image, such as $m = 7$ or $m = 20$. Figure 3 depicts the representation of each segment class by a set of sampled patches. Next, we define $\varphi(p, l)$ to be the cost of assigning label l to a pixel $p \in I_{test}$. Informally, a low cost $\varphi(p, l)$ indicates that there is a high likelihood that p should be labeled with l, and vice versa. We compute $\varphi(p, l)$ by matching P, the $m \times m$ square patch centered at p, with the patches in the set S_l. The cost is proportional to the distance to the nearest neighbor of P within S_l:

$$\varphi(p, l) = \min_{P' \in S_l} \frac{ssd(P, P')}{M},$$

where $ssd(P, P')$ is the sum of squared distances between the patches P and P', both treated as M-length vectors, where $M = m \times m \times 3$ in the case of three RGB color channels.

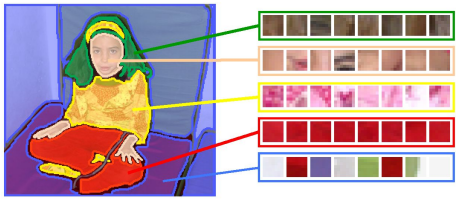

Fig. 3. Patch-based classifier. Each semantic part is represented by a set of square patches, sampled from within the corresponding region in the training image.

Fig. 4. Visualization of fragment labeling costs. Costs range in the interval [0,1] and are colorized according to each label's representative color, as defined in the Figure 3.

3.2 Fragmentation

The search for the nearest-neighboring patches within each set S_l is computationally intensive. In order to reduce the number of such searches, we partition I_{test} into small, color-homogeneous regions, which we refer to as *fragments*. These fragments are arbitrarily-shaped and may contain from a few pixels to thousands of pixels. We exploit the resulting structure to accelerate the algorithm by evaluating the labeling costs only for a small fraction of the pixels within each fragment, and then use voting to arrive at a set of labeling costs for each fragment.

The fragmentation is performed such that fragments are smaller in more detailed areas of I_{test}, and larger in more homogeneous regions. In addition, it is important that fragment boundaries align with edges in the image, since such edges may correspond to the boundary between different semantic regions. Fragments which comply to these criteria may be computed using mean-shift segmentation [8] with sufficiently small kernel bandwidths. Figure 5 demonstrates the result of fragmentation. Notice how small fragments form in highly detailed areas (such as the hair and shirt regions), while large fragments form in homogeneous areas (such as the walls in the background).

In addition to reducing the computational cost, fragmentation actually helps produce better results, for two reasons. First, fragmentation constrains pixels within the same fragment to be assigned to the same label, thereby enforcing a locally coherent labeling. Second, the voting procedure performed on pixels within each fragment produces more robust labeling costs.

Fig. 5. Fragmentation. The input image is fragmented into arbitrarily-shaped homogeneous regions, which we call fragments. Fragment sizes vary according to the amount of detail in various image areas, and their boundaries are aligned with edges in the image. The label assignment of each fragment is computed by choosing representative patches.

3.3 Fragment Labeling Costs

We apply a voting scheme in order to compute the labeling costs of each fragment. For each fragment $f \in I_{test}$ we pick a few representative pixels:

$$Rep(f) = \{p_i \in f\}_{i=1}^{R_f},$$

where R_f is proportional to the number of pixels in f, for example: $R_f = \lfloor\sqrt{|f|}\rfloor$. Figure 5 visualizes fragments along with their representative pixels (and the corresponding patches). The cost of assigning label l to fragment $f \in I_{test}$ is defined as:

$$\varphi(f, l) = \text{median}\left\{\varphi(p, l) | p \in Rep(f)\right\}.$$

Choosing the median value is a robust voting scheme, which is insensitive to outliers. By the end of this process, each fragment is associated with k different costs, one for each label. Figure 4 shows a visualization of the labeling costs that were computed for the example in Figure 1.

As patches and fragment dimensions are frequently similar, it is often the case that the patches centered at the representative pixels contain pixels outside the fragment, affecting the fragment's labeling costs. A simple solution would be to introduce weights into the computation of the distance between patches, but this interferes with the efficient nearest neighbor search that our implementation currently employs. It should be noted however that the effect of these outliers is significantly reduced by the voting scheme.

3.4 Graph-Cuts Optimization

After all pixels in the test image I_{test} have their labeling costs, we need to find L_{test}, the globally optimal labeling. A label assignment that minimizes the total labeling cost and also is devoid of small, disconnected segments. Thus, we also require the labeling to be consistent with the presence (or absence) of edges in I_{test}.

In order to satisfy these requirements, we add an additional pairwise constraint $\psi\left(p, q, L(p), L(q)\right)$ between each pair of neighboring pixels $\langle p, q\rangle$. This constraint enforces label assignments to change only across evident edges in I_{test}. The constraint $\psi(p, q, L(p), L(q))$ is 0 when the labels assigned to p and q are the same ($L(p) = L(q)$) and otherwise, proportional to the evidence of $\langle p, q\rangle$ not being an edge in I_{test}. Specifically,

$$\psi\left(p, q, L(p), L(q)\right) = \begin{cases} 0 & L(p) = L(q) \\ 1 - \nabla(p, q) & \text{otherwise} \end{cases} \tag{1}$$

where $\nabla(p, q)$ is the difference (in RGB distance) between pixels p and q, attenuated and scaled to the range $[0, 1]$. Furthermore, we enforce the restriction that pixels within each fragment should be labeled the same, in order to reduce the combinatorial search-space and achieve a satisfactory approximation at reduced computational costs. This is implemented the by specifing our energy term $E(L)$ in terms of fragments instead of pixels:

$$E(L) = \sum_f |f| \cdot \varphi\left(f, L(f)\right) + \alpha \sum_{\langle f_1, f_2\rangle} \psi\left(f_1, f_2, L(f_1), L(f_2)\right).$$

Fig. 6. The contribution of fragmentation and global optimization. The training set consists of four semantically meaningful segments: three plants and the background. Notice that the plants' segments have very similar local characteristics, except in their upper part, which has a unique color. (a) shows that a direct labeling of pixels fails to induce a locally coherent segmentation, due to the close similarity. (b) shows that labeling of fragments produces coherent labeling, but the labeling is over-segmented. (c) shows that a global combinatorial optimization captures semantically meaningful parts, and assigns the correct label.

Here $\langle f_1, f_2 \rangle$ are neighboring fragments in I_{test}. $\varphi\left(f, L(f)\right)$ is the cost defined in Sec. 3.3, weighted by the size of each fragment. The pairwise constraint $\psi()$ is extended to neighboring fragments by summing the constraint over their shared boundary:

$$\psi\left(f_1, f_2, L(f_1), L(f_2)\right) = \sum_{\langle p,q \rangle, p \in f_1, q \in f_2} \psi\left(p, q, L(f_1), L(f_2)\right).$$

Finally, L_{test} is determined by solving: $L_{test} = \min_L E(L)$. We apply the graph-cuts multi-label optimization technique for the fragment-based energy term $E(L)$, using the alpha-expansion method [12].

4 Implementation and Results

Image fragmentation is implemented with the mean-shift algorithm from [20]. Graph-cuts optimization is implemented with the *Maxflow* algorithm from [21], which computes the optimal cut for each alpha-expansion move. In this implementation the trade-off between regions and boundaries, is controlled by a single parameter α. Figure 7 demonstrates the profound effect of this parameter on the results. In all our experiments we used a fixed α value for all the images within

the same series, typically setting α to one or a nearby value. For searching square patches we uses a kd-tree [22]. In most of our results, we use patches of size 7×7. To reduce computation time, we sample only 5% of possible patches within each label in the training pair. Labeling of images of size 256×256 pixels, with three to six labels takes a few seconds on a 1.8 GHz Pentium 4 machine.

We test our method in the following scenario: Within a set of similar images, one image is chosen to be the training image. We manually segmented the image into multiple semantically meaningful parts, and colored each part with a unique color. Ambiguous pixels were marked in black. Trained by this image pair, our algorithm is used to induce the correct labeling on the remaining images. By image similarity we require that all images should depict the same subject (e.g., birds on the grass), have similar illumination conditions and are of similar resolution and scale. In some of the examples, we apply manual histogram equalizations and scaling in order to enforce these requirements.

Depending on the application, there are many ways to segment a particular image into semantically meaningful parts. Figure 1 depicts our experiments of creating different conceivable labelings and their induction on another image within the same domain. Note that certain semantically meaningful labelings, like the one that merges clothes and hair under the same label, cannot be characterized in terms of simple image features, and thus cannot be inferred without an explicit description or an example.

As described above, we use fragmentation to enforce locally-coherent labeling of pixels, and graph-cuts optimization to induce the globally optimal assignment of labels to fragments. Particularly, propagation of information across fragments is crucial in scenarios where different semantic parts share similar sub-parts. We

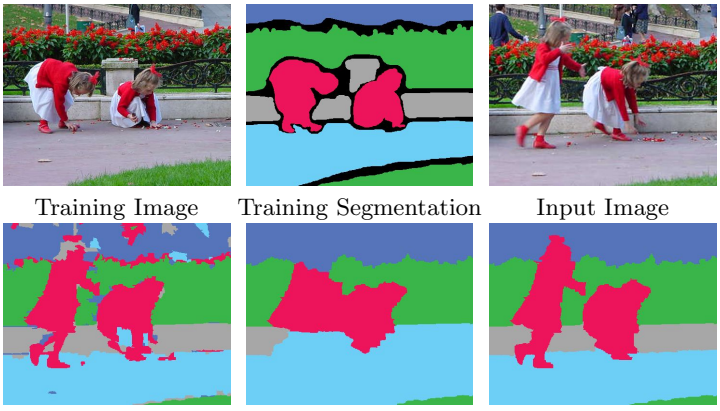

Fig. 7. The tradeoff between fragment labeling costs and the pairwise smoothness constraints is controlled by a single parameter α. A low α value favors boundaries and produces a over-segmentation, while a high α value penalizes boundaries, producing under-segmentation.

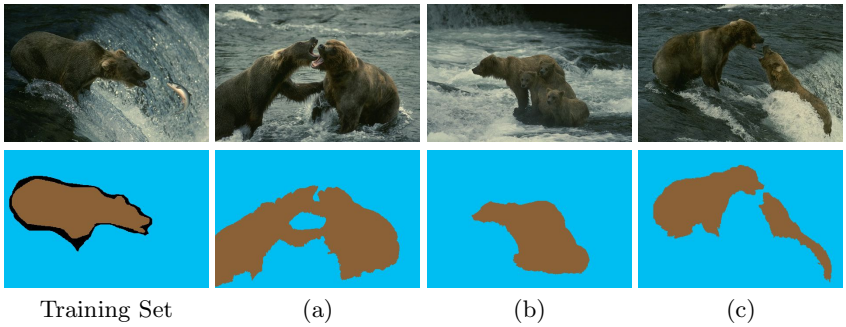

Fig. 8. Arbitrary segment shapes. The segmentation between bear and water is induced on three different images. Notice that the induced segmentation may contain holes (a), and be non-contiguous (b), but our method cannot separate multiple objects belonging to the same label (c).

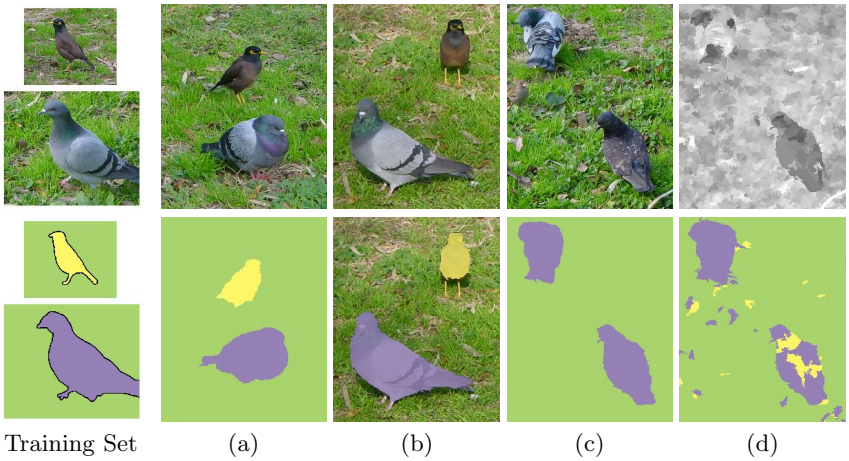

Fig. 9. Object detection and identification. Images of two types of birds are given as a training set, each bird marked by a distinct label. Labeling results (a-c) demonstrate our algorithm's ability to detect the presence of each bird. The gray scale image (d) demonstrates the labeling assignment costs of image (c), disclosing a greater confidence over the labeling of the left bird than the right bird, as the latter differ from the training image. Results without graph cut optimization (d) illustrate its contribution.

demonstrate the effect of the fragmentation and global optimization in Figure 6, by showing the consequences of omitting each of them.

Our method is invariant to the number of instances of each semantic part within the image, and insensitive to the shape of each part. Figure 8 shows the labeling of parts with different topology, in particular, holes (b) and discontinuities (c). On the other hand, our method cannot separate segments which correspond to multiple instances of the same semantic label, as in the bear family image (c).

The ability to segment an image and detect the semantic meaning of each part is demonstrated in Figure 9. Images of two types of birds are given as a training set, where each bird is marked by a distinct label. The results demonstrate the ability to correctly detect and distinguish between the birds (a). The bottom image in (b) also demonstrates that since fragments respect image edges, the labeled regions have correct boundaries, which agree with the underlying image.

Note that the lower right bird in (c) top constitutes a difficult case, since it is a bit darker than its counterpart in the example image, making it more similar to the second type of bird. This is evident in the gray-scale figure (d) top, which visualizes the optimal cost of the globally optimal labeling, demonstrating the problem of making clear cut decision. This image can be treated as a confidence map, and it discloses a greater confidence over the labeling of the left bird than the right bird.

5 Discussion and Future Work

"The whole is greater than the sum of its parts" [23] is one of the Gestalt principles. In this paper, we identify the parts (fragments) of the whole (meaningful object) by assigning them a common label. In general, labeling meaningful parts is known to be a difficult task. We have shown that inducing a labeling from an example can effectively perform this task for a set of images from the same domain. We can attribute this to the following reasons: (i) The example defines the granularity of the desired output. That is, whether we expect to label a complete human body, or its sub-parts: hands, torso, head, etc. (ii) The example allows the use of a non-parametric model to alleviate the huge space of parts. These have more discriminative properties than parametric models. Figure 6 demonstrates that applying the labeling to fragments rather than pixels provides better results. Note that the shapes of our fragments are data dependent rather than being predefined (e.g., rectangles or ellipses). We believe that the labeling problem should address meaningful building blocks, and that pixels are too small to be informative.

In the future we would like to investigate the applicability of our method to a series of images with some spatial coherence. Such coherence can assist the labeling of fragments across the images by considering their relative spatial position in the image. This can then lead to various tracking methods applicable to video with scenarios which include occlusions and frequent scene cuts.

References

1. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: International Conference on Computer Vision, Corfu, Greece (1999) 1033–1038
2. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. Computer Graphics and Interactive Techniques (2001) 327–340
3. Welsh, T., Ashikmin, M., Mueller, K.: Transferring color to greyscale images. Computer Graphics and Interactive Techniques (2002) 277–280

4. Drori, I., Cohen-Or, D., Yehurun, H.: Fragment-based image completion. ACM Transactions on Graphics, (SIGGRAPH) (2003) 303–312
5. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: IEEE Conference on Computer Vision and Pattern Recognition. (2004) 120–127
6. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. ACM Transactions on Graphics, (SIGGRAPH) (2001) 341–346
7. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. Trans. on Pattern Analysis and Machine Intelligence **13** (1991) 583–598
8. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. Trans. on Pattern Analysis and Machine Intelligence (2002) 603–619
9. Lucchese, L., Mitra, S.K.: Color image segmentation: A state-of-the-art survey. Proc. Indian National Science Academy (INSA-A) **67** (2001) 207–221
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. Trans. on Pattern Analysis and Machine Intelligence **22** (2000) 888–905
11. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium (2003) 239–269
12. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: International Conference on Computer Vision, Vancouver , BC. (2001) 105–112
13. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Comput. Graph. Appl. **22** (2002) 56–65
14. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: European Conference on Computer Vision. Volume 2., Copenhagen, Denmark (2002) 109–124
15. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Transactions on Graphics, (SIGGRAPH) **23** (2004) 303–308
16. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics **23** (2004) 309–314
17. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23** (2001) 1222–1239
18. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Transactions on Graphics, (SIGGRAPH) **23** (2004) 574–583
19. Agarwala, A., Hertzmann, A., Salesin, D., Seitz, S.: Keyframe-based tracking for rotoscoping and animation. ACM Transactions on Graphics, (SIGGRAPH) **23** (2004) 584–591
20. Christoudias, C.M., Georgescu, B.: Edge detection and image segmentation (edison) system. (http://www.caip.rutgers.edu/riul/research/robust.html)
21. Boykov, Y., Kolmogorov, V.: Maxflow software. (http://www.cs.cornell.edu/People/vnk/software.html)
22. Mount, D., Arya, S.: Ann: Library for approximate nearest neighbor searching. (http://www.cs.umd.edu/ mount/ANN/)
23. Wertheimer, M.: Productive Thinking. Collins, NY (1945)