Shape Prior Segmentation of Multiple Objects with Graph Cuts

Nhat Vu and B.S. Manjunath

Department of Electrical and Computer Engineering

University of California, Santa Barbara, CA 93106-9560

{nhat,manj}@ece.ucsb.edu

Abstract

We present a new shape prior segmentation method using graph cuts capable of segmenting multiple objects. The shape prior energy is based on a shape distance popular with level set approaches. We also present a multiphase graph cut framework to simultaneously segment multiple, possibly overlapping objects. The multiphase formulation differs from multiway cuts in that the former can account for object overlaps by allowing a pixel to have multiple labels. We then extend the shape prior energy to encompass multiple shape priors. Unlike variational methods, a major advantage of our approach is that the segmentation energy is minimized directly without having to compute its gradient, which can be a cumbersome task and often relies on approximations. Experiments demonstrate that our algorithm can cope with image noise and clutter, as well as partial occlusions and affine transformations of the shape.

1. Introduction

Segmentation methods based solely on image information [6, 8, 15, 22] often perform poorly in the presence of noise, background clutter, and object occlusions. The addition of shape prior information has shown to significantly improve segmentation results and is popular among variational approaches [7, 9, 20, 24, 25, 28]. Recently, there has been an increased interest in graph based segmentation algorithms [1, 2, 5], and subsequently the addition of prior shape information into their formulations. However, many continuous shape distances or dissimilarity measures can be difficult, if not impossible, to formulate as discrete energies for graph optimization. This is especially true for graph cut methods.

The graph methods of Felzenszwalb [11] and Schoenemann and Cremers [26] can segment objects under elastic deformations without needing any initialization and guarantee globally optimal solutions. In [11], nonserial dynamic programming is used to find the optimal matching between a deformable template represented by triangulated polygons

and the image pixels. In [26], the segmentation is found by computing the minimal ratio cycle in a product graph of the image and a shape template parameterized by arc length. Both of these methods can be slow in practice, with runtimes of up to several minutes on typical CPUs. Moreover, the triangulated polygon representations and arc length parameterizations limit the topological flexibility of the template shapes and may not easily extend to the 3D case.

There are several algorithms that employ graph cuts for shape prior segmentation. Freedman and Zhang [13] use the shape's distance transform φ to define the edge weight between neighboring pixels p and q as $\varphi((p+q)/2)$. Kolmogorov and Boykov [17] assign neighborhood edge weights to favor cuts that maximize the flux of the distance map gradient. Both of these methods largely rely on user markings to estimate the template pose. Kumar *et al*. [19] also utilize the shape's (signed) distance map, but estimate the pose using shape and appearance models constructed during training. Most closely related to our work, Malcolm *et al*. [21] impose the shape prior model on the terminal edges and perform graph cuts iteratively starting with an initial contour. Given a set of training shapes, their method constructs a statistical shape space using kernel principle component analysis (kPCA). At each iteration, the pre-image of the previous labeling in this shape space is used as the prior probability map, and the negative log of this pre-image is assigned to the terminal weights. While these methods produce promising results, their shape energies are not based on shape metrics, *e.g.* they are unsymmetrical. Furthermore, these methods do not handle affine transformations of the shapes and cannot segment multiple objects simultaneously.

In this work, we present a new shape prior segmentation method using combinatorial graph cuts. First, we define the shape prior energy using a discrete version of the shape distance proposed by Chan and Zhu [7] for the level sets framework, and incorporate this energy into the graph via terminal edge weights. Unlike those of previous graph based approaches, this shape distance is both symmetrical and obeys the triangle inequality. Second, to simultane-

ously segment multiple objects, we propose a multiphase graph cut approach to handle object overlap, where a pixel can have multiple object memberships (labels). This is fundamentally different from the multiway cut solutions, such as can be found using the α-expansion algorithm [5], where each pixel is assigned only one label. We then extend our shape prior energy to incorporate multiple shape priors. To make the algorithm invariant to affine transformations of the shape, we use the theory of moment invariance of binary shapes [23] for alignment, allowing direct computation of the transformation parameters without using gradient descent estimation for each parameter. A major advantage of our algorithm is that the segmentation energy is minimized directly with graph cuts, unlike variational methods which require the energy gradient for minimization. Computation of the gradient for many energy functionals can be difficult because these energies are often non-differentiable and require approximations [9].

In section 2, we provide the necessary background on segmentation using graph cuts. Section 3 describes the shape prior model, and section 4 provides detail on using this energy in the multiphase graph cut framework for the segmentation of multiple objects. Section 5 extends the shape prior model to incorporate multiple prior shapes. Several results of our algorithm are shown in section 6, followed by a brief discussion in section 7.

2. Segmentation with graph cuts

The segmentation problem can be formulated as an energy minimization such that for a set of pixels $\mathcal{P}$ and a set of labels $\mathcal{L}$, the goal is to find a labeling $f : \mathcal{P} \to \mathcal{L}$ that minimizes some energy $E(f)$. Using Markov Random Fields (MRFs) with unary and pairwise cliques to model f [14], the energy is given by

$$E(f) = \sum_{p \in \mathcal{P}} V_p(f_p) + \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} V_{pq}(f_p, f_q), \quad (1)$$

where $\mathcal{N}_p$ is the set of pixels in the neighborhood of p. Here $V_p(f_p)$ is the penalty of assigning label $f_p \in \mathcal{L}$ to p, and $V_{pq}(f_p, f_q)$ is the penalty of labeling the pair p and q with labels $f_p, f_q \in \mathcal{L}$, respectively. In this work, $\mathcal{L} = \{0, 1\}$, and the minimum $E(f)$ can be computed efficiently with graph cuts when V_{pq} is a submodular function, *i.e.* $V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(0,1) + V_{pq}(1,0)$ [18].

2.1. Graph cuts

Before discussing graph cuts, we define the graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed weighted graph composed of a set of nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E}$ with nonnegative weights. The set $\mathcal{V}$ consists of two types of nodes: neighborhood nodes made up of the image pixels $\mathcal{P}$ and two terminal nodes s and t. In the neighborhood system, pixel p

is connected to pixel q if $q \in \mathcal{N}_p$, i.e. if q is a neighbor of p. Neighboring nodes p and q are connected by n-edge e_{pq} with n-weight w_{pq}. Furthermore, p is connected to terminals s and t via t-edges e_{sp} and e_{pt} with corresponding t-weights w_{sp} and w_{pt}, respectively. In our graph, if $e_{pq} \in \mathcal{E}$ then $e_{qp} \in \mathcal{E}$, the n-weights $w_{pq} = w_{qp}$, and all pixels $p \in \mathcal{P}$ are connected to both s and t.

A subset of edges $\mathcal{C} \subset \mathcal{E}$ is called an s/t-cut if the terminal nodes are completely separated in the induced graph $\mathcal{G} = (\mathcal{V}, \mathcal{E} - \mathcal{C})$. That is there are no paths from terminal s to terminal t when all edges in the cut are removed. Hence, the cut partitions the nodes into disjoint subsets $\mathcal{S}$ and $\mathcal{T}$ where $s \in \mathcal{S}$ and $t \in \mathcal{T}$. For simplicity, we will refer to the s/t-cut simply as a cut. The cost $|\mathcal{C}|$ of the cut is the sum of all edge weights in $\mathcal{C}$. For a given graph, the minimum cost cut (mincut) can be found by solving an equivalent maximum flow (maxflow) problem [12].

2.2. Minimizing $E(f)$ with graph cuts

We can assign a binary label to a node depending on whether it is in $\mathcal{S}$ or $\mathcal{T}$. In our notation, a pixel p is assigned label $f_p = 1$ (object) if $p \in \mathcal{S}$ and $f_p = 0$ (background) if $p \in \mathcal{T}$. As a result, each cut produces a labeling f and hence a corresponding energy $E(f)$. The goal is to assign weights to the graph's edges so that the mincut cost $|\mathcal{C}|$ is equal to the minimum energy $E(f)$.

In our framework, the unary penalty $V_p(f_p)$ is the sum of a data penalty $V_D(f_p)$ and a shape prior penalty $V_S(f_p)$. The data term is defined based on the image intensity and can be considered as a log likelihood of the image model, while the shape prior term is independent of image information. These terms will be described in subsequent sections. We use the pairwise penalty (a submodular function) [3]

$$V_{pq}(f_p, f_q) = g(p, q) \cdot |f_p - f_q|, \quad (2)$$

where

$$g(p, q) = \lambda_I \exp\left(-\frac{(I_p - I_q)^2}{2\sigma_I^2}\right) \frac{1}{\text{dist}(p, q)}. \quad (3)$$

Here I_p is the intensity value at pixel p, $\text{dist}(p, q)$ is the Euclidian distance between pixels p and q. The parameter σ_I can be considered an estimate of camera noise, and λ_I weights the importance of the pairwise energy. Accordingly, a penalty $g(p, q)$ is incurred only when neighboring pixels have different labels, and thus V_{pq} encourages region coherence of the labels. For convenience, the second term in Eq. (1) is denoted $E_{pq}(f)$. Finally, the desired graph with cut cost $|\mathcal{C}|$ equalling $E(f)$ is constructed using the following edge weight assignments [2]:

$$w_{pq} = g(p, q), \quad (4a)$$
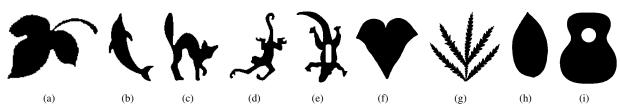$$w_{sp} = V_p(f_p = 0), \quad (4b)$$
$$w_{pt} = V_p(f_p = 1). \quad (4c)$$

3. Shape prior model

In this section, we describe the shape prior model and show how to define the shape penalty $V_S(f_p)$ such that a cut on the graph, with V_{pq} and V_D defined, has an added cost corresponding to the shape prior energy.

3.1. Shape distance

Given two shapes embedded onto the zero level of level set functions ϕ^a and ϕ^b on the image plane $\Omega \subset \mathbb{R}^2$, Zhu and Chan define their distance as [7]

$$d^2(\phi^a, \phi^b) = \int_\Omega \left(H(\phi^a(x)) - H(\phi^b(x)) \right)^2 dx, \quad (5)$$

where $H(\cdot)$ is the Heaviside function. Many level set segmentation methods [9, 24] use this distance as the shape prior energy due to its many attractive properties: it is positive, symmetric, obeys the triangle inequality, and does not depend on the size of the domain Ω.

Since $H(\phi^i)$ effectively binarizes the shape embedding function ϕ^i, for notation simplicity we will replace $H(\phi^i)$ with ψ^i. On the discrete pixel domain, Eq. (5) can be expressed in terms of ψ^a and ψ^b as

$$d^2(\psi^a, \psi^b) = \sum_{p \in \mathcal{P}} (\psi^a_p - \psi^b_p)^2$$
$$= \sum_{p \in \mathcal{P}} \left(\psi^a_p \bar{\psi}^b_p + \bar{\psi}^a_p \psi^b_p \right), \quad (6)$$

where ψ^i_p is the binary value of ψ^i at pixel p and $\bar{\psi}^i_p = 1 - \psi^i_p$. The expansion in Eq. (6) is possible because both ψ^a and ψ^b are binary functions. Note that the binary representation does not restrict the shape to be a single closed contour but allows it to have arbitrary topology (holes and multiple unconnected parts). See Fig. 1 for examples of shape templates used in this work.

3.2. Shape penalty

In order to use the shape distance (6) in the graph cut framework, we must define the shape penalty $V_S(f_p)$ such that, for a given prior shape template ψ^0, a cut with binary labeling f has an added cost equal to $d^2(f, \psi^0)$. Using Eq.

(6), we define the energy of a binary labeling f given a prior template ψ^0 as

$$E_S(f, \psi^0) = d^2(f, \psi^0). \quad (7)$$

Then the shape prior penalty is

$$V_S(f_p) = f_p \bar{\psi}^0_p + \bar{f}_p \psi^0_p. \quad (8)$$

It follows that if p is assigned label $f_p = 0$ (1) but $\psi^0_p = 1$ (0), then the t-edge e_{sp} (e_{pt}) is in the cut and a penalty of 1 is added to the cost. However, when $f_p = \psi^0_p$, no penalty is incurred. Thus, the cut which results in a labeling f that minimizes $d^2(f, \psi^0)$ gives the minimum shape prior energy.

3.3. Affine invariant shape alignment

To make the shape distance in Eq. (6) invariant to geometric transformations, ψ^a and ψ^b must be properly aligned. Since ψ^a and ψ^b are effectively binary images, we use the image normalization work of Pei and Lin [23] to align these shapes. The normalization process transforms a shape to an affine invariant shape space using transformations computed intrinsically from the shape's moments (up to 3rd order). Due to space limitations, we refer the reader to [23] for more details on the normalization process.

For the segmentation, assume that the prior template ψ^0 has been normalized. Given an estimate $\hat{f}$ for the target object (described in §4.2), $\hat{f}$ is normalized by computing the transformation $T(\hat{f})$. Then ψ^0 is aligned to $\hat{f}$ by reversing the normalization procedure on ψ^0 using the transformation computed for $\hat{f}$, i.e. $\hat{\psi}^0 = T^{-1}(\psi^0)$. Finally to make the distance scale invariant, $d^2(\hat{f}, \hat{\psi}^0)$ is divided by $\sqrt{\lambda_1 \lambda_2}$, where λ_1 and λ_2 are the eigenvalues of the covariance matrix of $\hat{f}$. For the remainder of this paper, the notation $d^2(f, \psi^0)$ is assumed to be the invariant shape distance between f and ψ^0.

In general, alignment by intrinsic normalization does not necessarily result in the minimum distance, especially when $\hat{f}$ contains many spurious or noisy parts. However, the iterative segmentation procedure described in §4.2 allows us to use this alignment scheme even when the initial estimate of $\hat{f}$ is very different from the template. More robust object pose estimation schemes, such as that in [19], can also be utilized but were not necessary in our experiments.
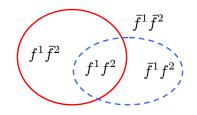
Figure 2. Regions for two labelings f^1 and f^2.

4. Shape prior segmentation

In this section, we present a multiphase graph cut method capable of segmenting multiple, possibly overlapping objects. For each object, the segmentation is posed as a binary labeling problem, with the shape prior penalty imposed independently for that labeling. However the data penalty is dependent on the other labelings. We will denote the jth object labeling by f^j and its value at pixel p by f_p^j. The total segmentation energy can be expressed as the sum of energies over all the labelings, *i.e.*

$$E(\mathbf{f}) = E_D(\mathbf{f}) + \sum_{j=1}^{M} \left(E_{pq}(f^j) + \lambda_s E_S(f^j, \psi^0) \right), \quad (9)$$

where $\mathbf{f} = \{f^1, \ldots, f^M\}$ is the set of M object labelings, and $E_D(\mathbf{f})$ is the sum of the unary data penalties of all labelings. The parameter λ_s controls the strength of the shape penalty. We will show that Eq. (9) can be minimized iteratively by performing M mincuts on a single graph at each iteration, with only t-weight updates. The proposed multiphase graph cuts can be considered a discrete version of the multiphase level set framework of Vese and Chan [29].

4.1. Multiphase graph cuts

As mentioned, we will use M labelings to segment M objects in the image. The labelings can partition the image into a maximum of 2^M regions or phases. The indicator function for region k is denoted as χ^k. Assume that there is a data model to describe each image region, *e.g.* $Pr(I_p|\chi^k)$, and that each model has an associated cost C^k, which could be considered as the log likelihood. Then the data cost for the entire image is the sum of the individual region costs, *i.e.*

$$E_D(\mathbf{f}) = \sum_{p \in \mathcal{P}} \sum_{1 \le k \le 2^M} C_p^k \chi_p^k, \quad (10)$$

where we denote $C_p^k = C^k(p)$ and $\chi_p^k = \chi^k(p)$. We proceed with an intuitive discussion of how to minimize Eq. (10) with graph cuts by describing the procedure for cases where $M = 1$ and $M = 2$. The cases for $M > 2$ can be solved using a similar reasoning, but will not be described in detail due to space limitations.

For a single object ($M = 1$), the image is divided into object ($f_p^1 = 1$) and background ($f_p^1 = 0$) regions, and the model for the two regions will compete to estimate the image. Given the object and background costs C^1 and C^0, respectively, Eq. (10) becomes

$$E_D(f^1) = \sum_{p \in \mathcal{P}} \left(C_p^1 f_p^1 + C_p^0 \bar{f}_p^1 \right). \quad (11)$$

Using the graph described in §2.1, this energy is added via data penalties $V_D(f_p^1 = 0) = C_p^0$ and $V_D(f_p^1 = 1) = C_p^1$.

Now for $M = 2$, consider the two labelings f^1 and f^2 shown in Fig. 2. With the overlap, the image is partitioned into four regions $\bar{f}^1 \bar{f}^2$, $f^1 \bar{f}^2$, $\bar{f}^1 f^2$, and $f^1 f^2$. The data energy is

$$E_D(f^1, f^2) = \sum_{p \in \mathcal{P}} (C_p^{11} f_p^1 f_p^2 + C_p^{10} f_p^1 \bar{f}_p^2$$
$$+ C_p^{01} \bar{f}_p^1 f_p^2 + C_p^{00} \bar{f}_p^1 \bar{f}_p^2), \quad (12)$$

where C^{00}, C^{10}, C^{01}, and C^{11} are the data costs for regions $\bar{f}^1 \bar{f}^2$, $f^1 \bar{f}^2$, $\bar{f}^1 f^2$, and $f^1 f^2$, respectively. To impose this cost into the graph cut framework, first consider the labeling f^1. For the regions of f^1 where $f_p^2 = 0$, *i.e.* $f^1 \bar{f}^2$ and $\bar{f}^1 \bar{f}^2$, the two models with costs C^{10} and C^{00} compete for the best fit. Similarly, the two models with costs C^{01} and C^{11} compete to describe the regions of f^1 where $f_p^2 = 1$, *i.e.* $\bar{f}^1 f^2$ and $f^1 f^2$. Thus, from the perspective of f^1, the labeling problem is similar to the single object case, except that now the object/background costs depend on f^2. A similar reasoning can be used for f^2.

Eq. (12) can be minimized iteratively by performing mincuts on a single graph, with each labeling computed alternatingly by updating the t-weights with data penalty assignments

$$V_D(f_p^1 = 0) = C_p^{01} f_p^2 + C_p^{00} \bar{f}_p^2, \quad (13a)$$
$$V_D(f_p^1 = 1) = C_p^{11} f_p^2 + C_p^{10} \bar{f}_p^2, \quad (13b)$$
$$V_D(f_p^2 = 0) = C_p^{10} f_p^1 + C_p^{00} \bar{f}_p^1, \quad (13c)$$
$$V_D(f_p^2 = 1) = C_p^{11} f_p^1 + C_p^{01} \bar{f}_p^1, \quad (13d)$$

where $V_D(f_p^i)$ is the penalty used to compute f^i.

For $M > 2$, the data penalty for each labeling must account for all 2^M possible regions. These penalties are computed by considering the object/background competition for each region. Note that the graph structure for all labelings is identical, with only the t-weights changing for each iteration. The n-weights remain the same, resulting in less memory storage and faster graph construction. Furthermore, since the mincut algorithm is inherently stable, large moves are possible during each iteration, speeding up convergence. However, like the level sets formulation, the multiphase graph cut framework does not guarantee the globally optimal solution.

4.2. Iterative segmentation

The shape prior energy $E_S(f^j, \psi^0)$ is dependent on the geometric transformations of the template ψ^0. However, unless the target object's pose in the image is known, the shape penalty cannot be defined accurately. To overcome this lack of information, we perform the segmentation in an iterative manner (see Algorithm 1). For the graph $\mathcal{G}$, the n-weights are computed only once and remain the same for all objects. Then given an initial labeling f^j for object j, the template ψ^0 is aligned to f^j. The shape penalties are computed using Eq. (8) and the data penalties are computed as described in §4.1. These unary penalties are summed and assigned to the t-weights, and the mincut solution for this graph produces a new f^j. This process is repeated for all M objects until convergence is reached.

Algorithm 1 Segmentation of M objects given ψ^0.

Compute n-weights of graph $\mathcal{G}$ (same for all objects).
Initialize labelings $\mathbf{f} \leftarrow \{f^1, \ldots, f^M\}$.
while $\mathbf{f}$ not converged **do**
    for $j = 1$ to M **do**
        1. Align ψ^0 to f^j.
        2. t-weights $\leftarrow V_D(f_p^j) + V_S(f_p^j)$
        3. $f^j \leftarrow$ mincut $\mathcal{G}$
    end for
    $\mathbf{f} \leftarrow \{f^1, \ldots, f^M\}$
end while

Ideally the final labeling should be insensitive to initializations since in general, an initialization may not be a good estimate of the final segmentation. To lessen the dependency on initialization, the data penalty should dominate the cost function at the start of segmentation, while the shape penalty should remain small. As the segmentation progresses, the shape penalty then increases and forces the cut to resemble the prior template more closely. The shape penalty can be adaptively controlled by redefining the shape energy for the ith iteration as

$$E_S(f^{j,i}, \psi^0) = \alpha(f^{j,i-1}, \psi^0) \cdot d^2(f^{j,i}, \psi^0), \quad (14)$$

where the weighting function

$$\alpha(f^{j,i-1}, \psi^0) = \exp\left(-\frac{1}{2\sigma_s^2} d^2(f^{j,i-1}, \psi^0)\right) \quad (15)$$

and $f^{j,i}$ is f^j at iteration i. Eq. (15) adaptively weighs the shape energy according to how similar the previous labeling $f^{j,i-1}$ is to the template ψ^0. The parameter σ_s controls the rate at which the shape energy changes.

We would like to point out that the proposed method do not guarantee a globally optimal solution. However, the mincut solution f^j at each iteration is globally optimal for

the graph constructed during *that* iteration. This allows for large stable "moves" in the labeling space between iterations, unlike variational methods where the updates are limited by the time step. These large moves help avoid "shallow" local minima and facilitate faster convergence.

4.3. Data model

We use the piecewise-constant occlusion model of Thiruvenkadam *et al.* [27] to define the data penalty and briefly describe it here. Assume that object j has constant intensity c_j, and the background intensity is c_0. The occlusion model for the image is given by

$$\hat{I} = \sum_{j=1}^{M} c_j f^j + \sum_{k=2}^{M} \sum_{\ell=1}^{\binom{M}{j}} (-1)^{k-1} c_{k,\ell} \chi^{k,\ell} + c_0 \prod_{j=1}^{M} \bar{f}^j, \quad (16)$$

where $\chi^{k,\ell}$ is the ℓth unordered intersection of k labels from $\mathbf{f}$ and $c_{k,\ell}$ takes one value in $\{c_j\}_{j=1}^{M}$. The data cost of using $\hat{I}$ to approximate a given image I is

$$E_D(\mathbf{f}) = \sum_{p \in \mathcal{P}} (I_p - \hat{I}_p)^2. \quad (17)$$

For $M = 1$, Eq. (17) becomes a discrete version of the Chan-Vese energy [8]. For $M = 2$, the image occlusion model is

$$\hat{I} = c_1 f^1 + c_2 f^2 - c_{2,1} f^1 f^2 + c_0 \bar{f}^1 \bar{f}^2. \quad (18)$$

After some rearranging and using the fact that $f^1 = f^1(f^2 + \bar{f}^2)$ and similarly for f^2, Eq. (17) becomes

$$E_D(\mathbf{f}) = \sum_{p \in \mathcal{P}} (I_p^2 + a_p^{11} f_p^1 f_p^2 + a_p^{10} f_p^1 \bar{f}_p^2$$
$$+ a_p^{01} \bar{f}_p^1 f_p^2 + a_p^{00} \bar{f}_p^1 \bar{f}_p^2), \quad (19)$$
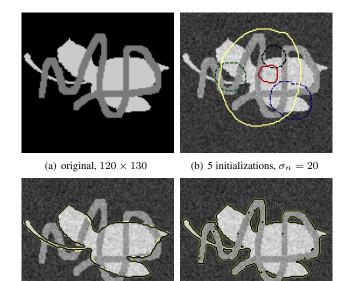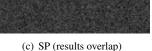
where

$$a_p^{10} = c_1^2 - 2c_1 I_p, \quad (20a)$$
$$a_p^{01} = c_2^2 - 2c_2 I_p, \quad (20b)$$
$$a_p^{11} = (c_1 + c_2 - c_{2,1})^2 - 2(c_1 + c_2 - c_{2,1}) I_p, \quad (20c)$$
$$a_p^{00} = c_0^2 - 2c_0 I_p. \quad (20d)$$

Since the first term in Eq. (19) is independent of f^1 and f^2, it does not factor into the minimization. The data penalties are defined by setting $C^{00} = a^{00}$, $C^{10} = a^{10}$, $C^{01} = a^{01}$, and $C^{11} = a^{11}$ in Eq. (13). Similarly for $M > 2$, Eq. (17) can be factored into a sum of costs for the regions, and the data penalties are assigned accordingly. The intensities $\{c_j\}_{j=1}^{M}$ are estimated by solving a linear system of equations after each iteration, and the occlusion relationship can be easily inferred from these object intensities [27].

(a) original, 120 × 130   (b) 5 initializations, $\sigma_n = 20$

(c) SP (results overlap)   (d) no SP

Figure 3. Five different initializations produced nearly identical results. The longest run took 11 iter, 0.921 sec. The parameters are set as $\lambda_I = \lambda_s = 0.5 \times 255^2, \sigma_s = 3$ and $\sigma_I = 15$.

5. Multiple prior shapes

For many segmentation tasks, the image can contain objects with completely different shapes (see Fig. 6) or an object that exhibits shape variability, such as the side view of a walking person. In such situations, the prior shape energy must make use of a set of prior templates or the multiple instances of a single object. The latter case is normally addressed by formulating the shape energy based on a statistical shape space [9, 10].

In this work, the multi-template shape energy is defined as a weighted sum of the distances between the templates and the labeling f. Given N prior templates $\Psi = \{\psi^1, \ldots, \psi^N\}$, the shape prior energy is

$$E_S(f, \Psi) = \frac{\sum_{n=1}^{N} \alpha(f, \psi^n) \cdot d^2(f, \psi^n)}{\sum_{n=1}^{N} \alpha(f, \psi^n)}, \quad (21)$$

with $\alpha(f, \psi^n)$ given by Eq. (15). The weight $\alpha(f, \psi^n)$ is a measure of the similarity between f and ψ^n, and hence shapes that are "closer" to the labeling f are given higher weights. In fact, Dambreville *et al.* [10] showed that the relationship between the distance, $d_F^2(\psi^a, \psi^b)$, between two shapes in a feature space constructed using kPCA and the distance $d^2(\psi^a, \psi^b)$ is $d_F^2(\psi^a, \psi^b) \propto 1 - \alpha(\psi^a, \psi^b)$. Thus it is reasonable to assume that $\alpha(f, \psi^n)$ is a good measure of similarity between shapes in a feature space, and our experimental results reflect this fact.
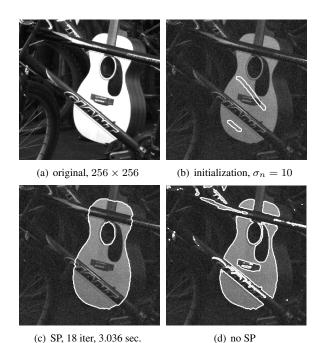

(a) original, 256 × 256   (b) initialization, $\sigma_n = 10$

(c) SP, 18 iter, 3.036 sec.   (d) no SP

Figure 5. Guitar occlude by bicycle frame. The parameters are $\lambda_I = 1.25 \times 255^2$, $\lambda_s = 0.625 \times 255^2$, $\sigma_s = 2.2$ and $\sigma_I = 10$.
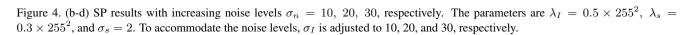
6. Experiments

All experiments were run in MATLAB on a PC with a 2.16 GHz Intel Core Duo processor and 2GB of RAM. We use a MATLAB wrapper to interface with the C++ maxflow code of Boykov and Kolmogorov [4]. The run time can be improved significantly by recycling the graph and the search trees [16], but our current implementation does not make use of such a scheme.

The image size, number of iterations (iter), run time (sec), and parameter settings are indicated directly in the figures. We denote the results using shape prior information by SP, and for several examples, the result without shape prior (no SP) are provided for contrast. Only the gray level intensity is used with $I_p \in [0, 255]$. The shape parameter σ_s, and correspondingly λ_s, is found to depend on the particular shape template, and we are currently investigating ways of determining the optimal σ_s for a given shape. Since the magnitude of the data penalty is in the range of I_p^2, the parameters λ_I and λ_s are shown with a scaling factor of 255^2. In all experiments, either the 4- or 8-connected neighborhood system is used, and convergence is reached when there is less than 1% change in the labeling(s).

Fig. 3 shows an image of a leaf produced by an affine transformation of the template in Fig. 1(a) with added occlusions and Gaussian noise with standard deviation σ_n. To demonstrate the algorithm's robustness to initializations, five different initial contours are used (Fig. 3(b)) resulting in nearly identical segmentations (Fig. 3(c)). Fig. 3(d) provides a comparison when no shape prior is used. Fig. 4

| (a) initialization, 279×330 | (b) 9 iter, 1.554 sec | (c) 13 iter, 2.340 sec | (d) 6 iter, 1.198 sec | (e) no SP, no noise |

Figure 4. (b-d) SP results with increasing noise levels $\sigma_n = 10$, 20, 30, respectively. The parameters are $\lambda_I = 0.5 \times 255^2$, $\lambda_s = 0.3 \times 255^2$, and $\sigma_s = 2$. To accommodate the noise levels, σ_I is adjusted to 10, 20, and 30, respectively.

shows the segmentation of a leaf using the shape prior in Fig. 1(f). Notice that the algorithm can cope with large occlusions and increasing noise levels. Fig. 5 shows the result of segmenting a guitar using the template in Fig. 1(i). Both the guitar's pickguard and bridge can be considered as occlusions and are labeled as background in Fig. 5(d).
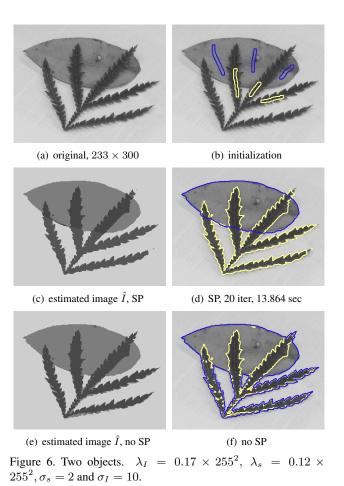
The result for two object segmentation is shown in Fig. 6. The shape prior energy in Eq. (21) is used with templates (g,h). The estimated images $\hat{I}$'s using Eq. (16) for both the SP and no SP cases are almost identical, but using shape prior information encourages the correct segmentation. Fig. 7 shows the results of segmenting three objects. The result in Fig. 7(b) is obtained using three labelings and the prior template (e). The result in Fig. 7(d) also uses three labelings, but three templates (b,c,d) are used for each labeling, with the shape prior energy as in Eq. (21). In all experiments, the objects are affine transformed versions of the templates, and the initializations do not provide good estimates of the shapes' poses nor do they more strongly favor any particular template for the cases with multiple priors.

7. Conclusion

We presented a new method capable of segmenting multiple objects with possible overlaps. Our framework combines several ideas. First, the shape prior information is incorporated into the graph via the t-weights. Unlike those of many previous graph based approaches, the shape distance is both symmetric and obeys the triangle inequality. Second, we introduced the multiphase graph cuts, whereby the simultaneous segmentation of multiple objects is simplified to a binary labeling problem for each object. Furthermore, we extend the shape prior energy to incorporate multiple shape priors, which is necessary when the object exhibits variability or when several different objects are present in the image. A major advantage of our framework over variational methods is that it explicitly minimizes the segmentation energy and thereby avoids the computation of the energy gradient, which can be difficult and often requires approximations. The results show that the algorithm is insensitive to initializations and noise and is efficient in practice.
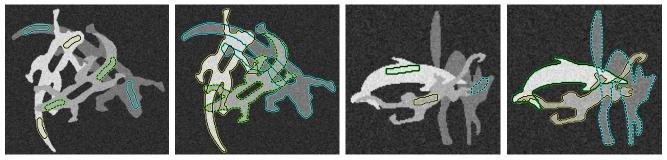

(a) original, 233×300	(b) initialization
(c) estimated image $\hat{I}$, SP	(d) SP, 20 iter, 13.864 sec
(e) estimated image $\hat{I}$, no SP	(f) no SP

Figure 6. Two objects. $\lambda_I = 0.17 \times 255^2$, $\lambda_s = 0.12 \times 255^2$, $\sigma_s = 2$ and $\sigma_I = 10$.

References

[1] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int'l J. Computer Vision*, 70(2):109–131, Nov. 2006. 1

(a) initialization, 210×230.　　(b) result, 18 iter, 7.247 sec.　　(c) initialization, 140×145.　　(d) result, 12 iter, 5.320 sec.

Figure 7. Three objects: (a,b) Single template used with parameters $\lambda_I = \lambda_s = 0.25 \times 255^2$, $\sigma_s = 1.5$ and $\sigma_I = 15$. (c,d) Three templates used with parameters $\lambda_I = 0.35 \times 255^2$, $\lambda_s = 0.23 \times 255^2$, $\sigma_s = 2$ and $\sigma_I = 15$. Noise level $\sigma_n = 15$ for both images.

[2] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & regionsegmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001. 1, 2

[3] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, pages 26–33, Oct. 2003. 2

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, Sept. 2004. 6

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, Nov. 2001. 1, 2

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int'l J. Computer Vision*, 22(1):61–79, Feb. 1997. 1

[7] T. Chan and W. Zhu. Level set based shape prior segmentation. In *CVPR*, pages 1164–1170, June 2005. 1, 3

[8] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, Feb. 2001. 1, 5

[9] D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *Int'l J. Computer Vision*, 69(3):335–351, Sept. 2006. 1, 2, 3, 6

[10] S. Dambreville, Y. Rathi, and A. Tannenbaum. Shape-based approach to robust image segmentation using kernel PCA. In *CVPR*, pages 977–984, June 2006. 6

[11] P. F. Felzenszwalb. Representation and detection of deformable shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2):208–220, Feb. 2005. 1

[12] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962. 2

[13] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, pages 755–762, June 2005. 1

[14] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Stat. Soc. Series B*, 51:271–279, 1989. 2

[15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l J. Computer Vision*, 1(4):321–331, January 1988. 1

[16] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2079–2088, Dec. 2007. 6

[17] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *ICCV*, pages 564–571, Oct. 2005. 1

[18] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, Feb. 2004. 2

[19] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, pages 18–25, June 2005. 1, 3

[20] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *CVPR*, pages 316–323, 2000. 1

[21] J. Malcolm, Y. Rathi, and A. Tannenbaum. Graph cut segmentation with nonlinear shape priors. In *ICIP*, 2007. 1

[22] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(2):158–175, Feb. 1995. 1

[23] S.-C. Pei and C.-N. Lin. Image normalization for pattern recognition. *Image and Vision Computing*, 13(10):711–723, Dec. 1995. 2, 3

[24] T. Riklin-Raviv, N. Kiryati, and N. Sochen. Prior-based segmentation and shape registration in the presence of perspective distortion. *Int'l J. Computer Vision*, 72(3):309–328, May 2007. 1, 3

[25] M. Rousson and N. Paragios. Shape priors for level set representations. In *ECCV*, pages 78–92, 2002. 1

[26] T. Schoenemann and D. Cremers. Globally optimal image segmentation with an elastic shape prior. In *ICCV*, 2007. 1

[27] S. R. Thiruvenkadam, T. F. Chan, and B.-W. Hong. Segmentation under occlusions using selective shape prior. In *Proc. Int'l Conf. Scale Space and Variational Methods in Computer Vision*, 2007. 5

[28] A. Tsai, J. Yezzi, A., W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Trans. Med. Imag.*, 22(2):137–154, Feb. 2003. 1

[29] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int'l J. Computer Vision*, 50(3):271–293, Dec. 2002. 4