Multiple View Object Cosegmentation using Appearance and Stereo Cues

Adarsh Kowdle[1], Sudipta N. Sinha[2], and Richard Szeliski[2]

[1]Cornell University, Ithaca, NY, USA
apk64@cornell.edu
[2]Microsoft Research, Redmond, WA, USA
{sudipsin,szeliski}@microsoft.com

Abstract. We present an automatic approach to segment an object in calibrated images acquired from multiple viewpoints. Our system starts with a new piecewise planar layer-based stereo algorithm that estimates a dense depth map that consists of a set of 3D planar surfaces. The algorithm is formulated using an energy minimization framework that combines stereo and appearance cues, where for each surface, an appearance model is learnt using an unsupervised approach. By treating the planar surfaces as structural elements of the scene and reasoning about their visibility in multiple views, we segment the object in each image independently. Finally, these segmentations are refined by probabilistically fusing information across multiple views. We demonstrate that our approach can segment challenging objects with complex shapes and topologies, which may have thin structures and non-Lambertian surfaces. It can also handle scenarios where the object and background color distributions overlap significantly.

Key words: object cosegmentation, multiview segmentation, multiview stereo

1 Introduction

In this paper, we address the task of segmenting a rigid object in multiple images where it has been photographed from multiple viewpoints. This task has a number of applications in image editing [25], image-based 3D modeling [8, 19, 35] and object instance recognition [34]. Single-image foreground extraction [6, 25] is a well-studied problem and has inspired several methods for multi-view segmentation [8, 19, 30], but most of these approaches rely on some degree of user interaction and supervision to obtain accurate results.

Our objective of automatically segmenting a rigid object in multiple images is related to the general image cosegmentation task, where the goal is to extract the region common to an image pair [26] or in multiple images [2, 18, 22]. Although several automatic approaches address the general case, they assume that the common regions appear similar across images whereas the background differ significantly in appearance. A recent approach [32] incorporates a preference for regions resembling objects. However, none of these cosegmentation approaches exploit the rich multi-view constraints satisfied by rigid objects and static scenes.

(a) (b) (c) (d)

Fig. 1: Our approach automatically segments objects in multiple images using the piecewise planar stereo algorithm proposed in the paper and applying multi-view reasoning on the 3D planar segments. (a) One of the 9 images of the COUCH sequence. (b) The segmented object. (c) The 3D plane labeling and (d) its corresponding piecewise planar depth map.

Existing methods for multi-view segmentation from calibrated images that can handle complex objects and scenes are almost all interactive [19,30,35]. The few automatic approaches either rely on the object and background appearances (color distributions) being very different [8, 20] or require the camera to be fixated on the object [8, 9]. Some of these approaches also implicitly estimate a globally consistent 3D reconstruction from the multiple viewpoints. However, reliably extracting 3D models for objects remains a challenging problem since existing methods cannot robustly handle objects with textureless or non-Lambertian surfaces or containing thin structures and holes, which gives rise to complex occlusions. The ambiguity arising from similar object and background colors poses another challenge for unsupervised methods that learn color models in order to distinguish between the object and its background.

In this paper, we present an automatic approach for multi-view object cosegmentation that addresses some of these challenges. Our system takes as input an unordered set of images captured around an object of interest, which is assumed to be fully visible in every image. The cameras are calibrated using a standard structure from motion (Sfm) approach [7, 29]. To automatically cosegment an object, we must address two challenges: (1) we need to reliably infer what constitutes the object and (2) we must accurately recover its contour or silhouettes in all the images.

Our approach has three stages. First, a piecewise planar, layer-based depth map is estimated for each image using a new approach that combines stereo matching with appearance cues based on color models. These depth maps consist of a set of 3D planes and a labeling of each pixel to one of these planes. An appearance model based on Gaussian mixture models (GMM) of color distributions is learnt for each surface in an unsupervised setting and used to recover more accurate depth maps. As shown in Figs 1(c) and (d), depth discontinuities can be accurately captured in this representation even though the depth estimates could be approximate. Next, we infer which of these planar surfaces constitutes the object using both appearance and depth cues as well as by analyzing their visibility in multiple views. Classifying these surfaces into object and background induces an initial segmentation in each image. Finally, in the third stage, these segmentations are probabilistically fused across multiple views to generate the final segmentations.

Contributions. We propose an automatic approach for object co-segmentation that exploits appearance and stereo correspondence cues. It uses a new piecewise planar depth map representation, which removes the need for estimating a globally consistent 3D reconstruction for the segmentation task and is robust to scenarios where stereo match-

ing can be unreliable. In contrast to existing unsupervised approaches that learn global color models for the object and background, our approach learns compact, per-surface appearance models from stereo correspondence cues and we show that this makes it possible to accurately recover depth discontinuities even in the presence of complex occlusions. We also show that the grouping of pixels into planes induced by our piecewise planar representation makes it easier to discern the object from the background when reasoning about them in multiple views.

1.1 Related Work

Several interactive approaches for multi-view object segmentation have been proposed [19, 30, 35]. They extend appearance based single-image segmentation approaches [6, 25] to multiple views but require user input to train the appearance models for the object and background. Similar approaches have been proposed for interactive video segmentation [1]. In the joint image segmentation formulation [23, 35], given a quasi dense 3D reconstruction, image pixels and 3D points can be jointly segmented into groups that constitute objects or object parts. The *field of view* cue was used in [27, 34] to automatically segment 3D point clouds reconstructed using Sfm but neither of these two approaches address the extraction of a pixel-level segmentation in the images.

A recent line of work has explored surface-based representations in stereo matching [3–5, 13, 15, 28]. While 3D planes are the most common choice [3, 5, 13, 15, 28], representations such as B-splines [4] have also been used. Amongst these approaches, appearance cues were exploited in [5, 15] although the former uses a supervised approach to classify pixels into planar and non-planar regions. Our new piecewise planar approach is closely related to [5] and has some similarities with prior work on bilayer segmentation in binocular video [11]. To the best of our knowledge, however, surface or layer-based representations have not been used in prior work on multi-view object co-segmentation. Many stereo matching approaches use low-level color-based oversegmentation of the image either as hard constraints [31] or soft constraints [4]. However, these low-level color cues cannot be used directly to reason at a higher level about objects, surfaces or their appearance models.

An existing automatic approach for multi-view segmentation [8] assumes that the cameras fixate on object and uses the pixels around the fixation point in each image to learn the object's color model. This as well as another related approach [20], simultaneously compute a visual hull of the object. These methods work best when the object and background color distributions do not overlap, in which case having a global color model for the object and background may be sufficient.

Our work is related to recent work by Campbell et. al. [9], where sparse stereo correspondences are used to simultaneously segment the object in multiple images. However, their approach relies on the fixation condition [8, 20] to bootstrap the color model of the object and requires reliable stereo correspondences for good results. The fixation condition in all these approaches makes them inconvenient when segmenting objects with complex shapes or topologies. In comparison, our approach only requires the object to be visible in all the images. An important distinction in our approach is that appearance (color) models are learnt for a set of surfaces (3D planes) in the scene. These color models are more compact and provide better discriminability in comparison to global

models. Unlike [8,9,20], where a joint pixel-level segmentation is directly computed in multiple images, we first estimate dense surface-based depth maps, then infer which of the underlying surfaces constitutes the object and finally compute a refined pixel-level segmentation. This multi-staged approach provides robustness in ambiguous situations where the object and background color distributions overlap significantly.

2 Overview

We use feature matching and structure from motion [7,29] (Sfm) to recover the camera calibration and then perform asymmetric stereo matching on image triplets using semi-global matching [17] with normalized cross correlation (NCC) as the matching cost[1]. A heuristic was used to select neighboring cameras for each reference view [2]. Fig 2(a) shows an example of a depth map recovered by this approach. A per-pixel confidence estimate that indicates the reliability of the associated depth estimates, is obtained by inspecting the ratio of the matching costs of the best and the second best hypotheses in the cost volume.

In our approach, we first estimate a dense, piecewise planar depth map by combining stereo cues with color-based appearance cues. We describe this in Section 3. We then segment the object independently in each image by inferring which of the planar segments in the estimated depth map belong to the object. Section 4 describes how this is done using a combination of appearance and depth cues along with multi-view reasoning. Section 5 describes how the final segmentations are computed after fusing the initial segmentations across multiple views.

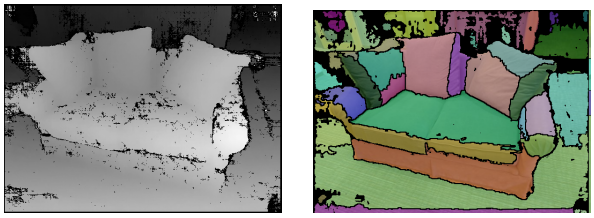
Although our representation is similar to [5], unlike their energy that is optimized via fusion moves [21] computed using QPBO-F, our energy function is amenable to efficient optimization via α-expansion [6] and has fewer tuning parameters. More over the symmetric binocular formulation proposed in [5] is difficult to extend to more than two views. In our approach, we perform stereo matching first and refine the noisy depth map by subsequently incorporating monocular appearance cues. This two-staged approach lets us easily incorporate confidence associated with the depth estimates computed from multiple views into the optimization. This also provides greater robustness when the appearance model parameters are learnt. However, we avoid the additional complexity of a plane+parallax representation [5], as our focus is on recovering precise depth discontinuities which are more important for accurate segmentation.

3 Piecewise planar stereo revisited

Let $p \in P$ denote the pixels in an image and let $D = \{d_p\}$ and $C = \{c_p\}$ where $c_p \in (0, 1)$, denote the corresponding depth and confidence maps respectively. Let $\Pi = \{\pi_i\}$ denote the set of planes hypothesized from the depth map D. In this section, we first describe how these planes are computed. We then describe an energy-based formulation

[1] We use a plane-sweep stereo framework and match the warped neighboring views to the reference image to construct the cost volume.

[2] cameras having many 2D features in common with the reference view and which form a suitable baseline with respect to it are preferred



(a) (b)

Fig. 2: (a) A noisy depth maps for one of the COUCH images computed using [17]. (b) A partial labeling of pixels to hypothesized 3D planes is visualized, where the pseudo-colors represent the pixels associated with a particular plane. Black pixels do not have a plane label.

to estimate a dense pixel-to-plane labeling L, where each pixel p is given a label $l_p \in \Pi$. Each plane π_i is parameterized by a 3D plane equation, a 2D extent in the image, and an appearance model, whose parameters we denote by A_i.

3.1 Plane hypotheses generation

We generate multiple plane hypotheses from a semi-dense depth map using a seed and grow approach. In our experience, this approach works better for arbitrary non-planar scenes compared to a sequential RANSAC-based approach [15] that can accurately estimate only dominant scene planes. Unlike [28] our planes also have an appearance model whose parameters are learnt in an unsupervised manner. Our approach is based on K-means clustering and is similar to existing work on mesh simplification [10]. We first compute a set of locally planar patches or *surfels* $\{s_p\}$ from 3D points in the depth map and then robustly cluster them on the basis of coplanarity.

At a pixel p, a surfel s_p is computed from the 3D points corresponding to pixels within a 7×7 patch around p in the depth map D. This step uses a standard least squares plane fitting approach as described in [33]. A surfel is computed when the patch contains enough confident depth estimates (we require at least 10 samples with confidence $c_p > 0.1$). Reliable surfels are retained after checking the quality of the plane fit and pruning unstable and degenerate surfels.

The set of plane hypotheses $\{\pi_i\}$ are then computed by minimizing the total approximation error given by the objective $\sum_p \sum_i f(s_p, \pi_i)$, where the function,

$$f(s_p, \pi) = \left| \frac{1}{d_p} - \frac{1}{d_\pi} \right| / \left(\frac{1}{d_p} \right) = |d_p - d_\pi| / d_\pi \tag{1}$$

measures the error incurred by assigning surfel s_p to plane π. Here, d_p and d_π denote the depth at pixel p corresponding to s_p and the depth of the 3D point, where the ray back projected from p intersects the plane π.

We construct a graph where the nodes represent pixels with valid surfels and edges are present between nodes within w pixels of each other in the image (we use w=5). An iterative seed and grow clustering is now performed on this graph using the approach described in [10]. On convergence, the planes corresponding to the M largest clusters with at least m pixels in each cluster are selected as the final set of plane hypotheses. Here, we set M=50 and m=20. We find the bounding box of the clustered pixels and

expand it by a factor of two to obtain an initial estimate of the 2D image extent of each plane. Note that this clustering induces a noisy and partial labeling of pixels to planes in the image, an example of which is shown in Fig 2(b).

3.2 Formulation

Given the set of hypothesized planes, we formulate the estimation of the piecewise planar depth map as a multi-label MRF optimization problem, which we solve approximately using iterative energy minimization. At each iteration, we obtain an approximation to the Maximum a Posteriori (MAP) labeling of pixels to the set of planes using α-expansion [6], after which the parameters used to compute the appearance terms in the energy function are re-estimated. The pairwise MRF is defined on a graph with the set of pixels P as nodes and all pairs of adjacent pixels on a 4-connected grid denoted by $\mathcal{N}$ as edges. We compute the labeling L that minimizes the following energy.

$$E(L) = \sum_{p \in P} E_p^A(l_p) + \lambda_G \sum_{p \in P} c_p E_p^G(l_p) + \lambda_S \sum_{(p,q) \in \mathcal{N}} E_{pq}(l_p, l_q) \tag{2}$$

The unary terms $E_p^G(l_p)$ and $E_p^A(l_p)$ measure the penalty of assigning pixel p to plane l_p based on a geometric and appearance cost respectively. The pairwise term $E_{pq}(l_p, l_q)$ measures the penalty of assigning pixels p and q to planes l_p and l_q respectively, and λ_S is a regularization parameter.

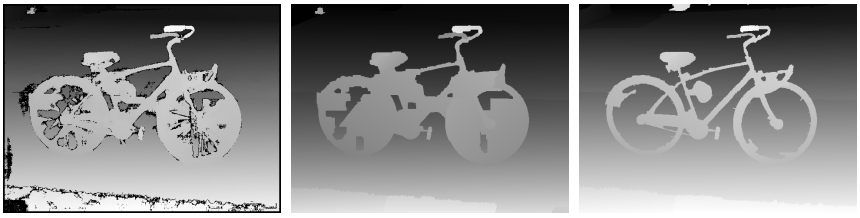
Geometric unary term (E^G) This term measures the geometric cost of assigning a pixel with a given raw depth estimate to a particular plane. It is computed as shown in Equation 1. The effect of this unary term is modulated using a per-pixel weight $c_p \in (0, 1)$, which is the confidence in the depth estimate d_p at pixel p and a global parameter λ_G that scales the geometric term relative to the appearance term.

Appearance-based unary term (E^A) This term measures the cost of assigning pixel p to a plane based on the pixel's color. Each plane π_l has an appearance model implemented as a Gaussian Mixture Model (GMM) of colors (in Lab space) with K mixture components. The corresponding GMM parameters are denoted by $\mathbf{A}_l$. The k-th GMM component is a Gaussian distribution with mean μ_k^l and covariance $\mathbf{\Sigma}_k^l$ and has a weight w_k^l. Given the pixel's color $\mathbf{x}$, the appearance cost for assigning it to a plane l is defined as the negative log likelihood, $E_p^G(l) = -log(p(\mathbf{x}|\mathbf{A}_l))$ where,

$$p(\mathbf{x}|\mathbf{A}_l) = p(\mathbf{x}|\{\mu_k^l, \mathbf{\Sigma}_k^l, w_l^k\}) = \sum_{k=1}^{K} w_k \, \mathcal{N}(\mathbf{x}|\mu_k^l, \mathbf{\Sigma}_k^l) \tag{3}$$

and $\mathcal{N}(\mathbf{x}|\mu, \mathbf{\Sigma})$ denotes the probability density function of a Gaussian distribution with mean μ and covariance $\mathbf{\Sigma}$.

Pairwise term (E_{pq}). The pairwise term is modeled using a contrast sensitive Potts model making the energy function regular. We segment the image into small *superpixels* using the GeoS algorithm [12] where the superpixels are no larger than 10×10 pixels but are typically much smaller. Although our MRF is defined on a regular grid, we use this superpixel segmentation as a soft constraint to guide the label boundaries



(a) (b) (c)

Fig. 3: (a) One of the initial depth maps from the BICYCLE sequence computed using [17]. (b) The piecewise planar depth map obtained when only geometric cues are used. This is similar to [28]. (c) The depth map computed by our approach. Using appearance cues, both thin structures and complex occlusion boundaries are accurately recovered.

towards high contrast image edges. We quantize the image by assigning to each pixel p, the mean color $\mathbf{x}_p^s$ in its superpixel and define the pairwise term for pixels p and q as $E_{pq}(l_p, l_q) = \beta_0 + \beta_1 \exp\left(-\gamma|\mathbf{x}_p^s - \mathbf{x}_q^s|^2\right)$ where $l_p \neq l_q$ [3].

Learning the Parameters. In the first iteration, we minimize the energy defined in Equation 2 without the appearance-based unary terms by setting λ_G to 0. In subsequent iterations, λ_G is set to 100 and the appearance term is evaluated using GMM parameters learnt from the labeling obtained from the previous iteration. To learn the GMM parameters for a plane, we consider the pixels assigned to it in the current labeling and consider the distribution of confidence for these pixels. The t-th percentile of this distribution is used to threshold and select confident pixels for training the GMMs. We set t to 50. The GMM parameters are estimated using the EM algorithm after model selection is used to determine the optimal number of mixture components. We allow a maximum of ten components and use the MDL (Rissanen) criterion [24] to select the value of K that maximizes the regularized posterior of all the n_l pixels used to train the GMMs [4].

Unlike interactive segmentation [19,25,30] where training data is user-provided and almost always reliable, the parameter learning step in our algorithm needs to be robust to noisy training samples, which arise due to errors in stereo matching. Usually, pixels with inaccurate depths have low confidence when compared to accurate depth pixels. However there are exceptions – the confidence could be relatively high at specular surfaces, or at occlusion boundaries with textureless occluded surfaces, even though these pixels may have inaccurate depth estimates. Pixels on weakly textured surfaces, on the other hand, have low confidence even though their depth may be accurate [17].

We also use the confidence c_p to weight the geometric term in the energy function 2. This makes our approach robust to errors in stereo matching, which occur in cases just described. As confidence estimates are usually low at textureless regions and depth discontinuities, in both these cases the appearance term dominates the geometric term. This enforces smoothness in homogeneous regions but also allows intricate occlusion boundaries to be recovered. Fig 3 shows an accurate depth map recovered for a challenging object with thin structures and holes.

[3] Here, β_0 = 1.0, β_1 = 50.0, γ = 0.01 and $\mathbf{x} \in (0, 255)$.

[4] MDL criterion: $\left(-log(p(\mathbf{x}|\mathbf{A}_l)) + \frac{dim}{2} log(n_l)\right)$, here $dim = 3$

4 Surface segmentation

Given a labeling L of pixels to planes, we now show how to infer which of the planar segments belong to the object. We compute a binary labeling F_r over regions in L, using labels 0 and 1 for background and object respectively. We first reparameterize the labeling L to obtain a new labeling L' where each connected 2D segment has a unique label. These regions are denoted as $R = \{r_i\}$. We analyze all pairs of segments (r_i, r_j) and link them if their corresponding polygon in 3D are coplanar and satisfy a necessary visibility condition. To do this, we sort the pairs in decreasing order of the degree of coplanarity of their corresponding polygons and test them in a greedy fashion until all pairs within a threshold have been considered.

The visibility condition is as follows – by linking two coplanar polygons in 3D, a larger polygon is implicitly generated. If any part of this polygon occludes a planar segment in the original depth map, it violates visibility and the regions are not linked. However, if the polygon is completely occluded by other segments, the regions can be linked. We implement this visibility check using an approach similar to [5], where it was used to infer 3D connectivity of disconnected 2D segments. Linking disconnected coplanar regions lets us reason about them based on the visibility of their common 3D plane in the multiple calibrated images. Finally, we have a new pixel to region labeling L' and a many-to-one mapping from regions in L' to a recomputed set of planes.

We now compute the region labeling F_r using energy minimization on two similar binary pairwise MRFs defined on a graph over the regions $r \in R$ with edges connecting adjacent regions denoted as $\mathcal{N}_R$. We compute labelings F_r^a and F_r^g by minimizing the two energy functions. [5]

$$E(F_r^a) = \sum_{r \in R} E_r^O(f_r) + \sum_{(r,t) \in \mathcal{N}_R} E_{rt}^a(f_r, f_t) \tag{4}$$

$$E(F_r^g) = \sum_{r \in R} E_r^O(f_r) + \sum_{(r,t) \in \mathcal{N}_R} E_{rt}^g(f_r, f_t) \tag{5}$$

The unary term $E_r^O(f_r)$ in both energy functions are identical. However, the pairwise terms $E_{rt}^a(f_r, f_t)$ and $E_{rt}^g(f_r, f_t)$ both of which favor label agreement, are different and based on the compatibility of regions in terms of appearance and depth respectively. These energy functions can be efficiently minimized using an s-t mincut algorithm. In general, the two solutions F_r^a and F_r^g could be different. In the final segmentation F_r, a region is labeled as object if it is labeled object in either of the solutions. Otherwise it is labeled as background.

Multiview objectness unary term (E^O). Using the camera calibration and piecewise planar depth induced by L', we warp image pixels into the other views. If a pixel projects within every other image, we say that it has high *objectness*. Since the object is assumed to be visible in every image, pixels with low objectness are likely to be background. Pixels with high objectness could either be object pixels or lie on background surfaces that are within the field of view of every camera. Thus we measure the

[5] Later in this section, we explain why we choose two energy functions instead of a single unified one.

objectness of a region r as the fraction of its pixels ϕ_r that have high objectness. The term is defined as $E_r^O(0) = (1 + \exp(-(\phi_r - \mu_0)/\sigma_0))^{-1}$ and $E_r^O(1) = 1 - E_r^O(0)$ where μ_0 = 0.9, σ_0 = 0.02. We force a region to take the foreground or background label if $\phi_r > 0.99$ or $\phi_r < .8$ respectively, using hard constraints.

Appearance-based compatibility term (E_{rt}^a). This term measures the dissimilarity of the color histograms of regions r and t. For this, we compute the KL-divergence between their GMMs denoted as $\rho_{rt} = D_{KL}(\mathbf{A}_r||\mathbf{A}_t)$, where $\mathbf{A}_r$ and $\mathbf{A}_t$ are the GMM parameters, using the method proposed in [16]. The term is defined as $E_{rt}^a(f_r, f_t) = \eta_1 \exp(-\mu_1 \min(\rho_{rt}, \rho_{tr}))$ when $f_r \neq f_t$. When two dissimilar regions take different labels, there is a lower penalty than when the two regions are similar.

Depth-based compatibility term (E_{rt}^g). This term measures for two adjacent regions in L', how close their corresponding polygons are in 3D by computing the mean relative depth discontinuity for pixels along the label boundary. Thus, for regions r and t, we find a set B of pairs of adjacent pixels (p_i, q_i) that lie across the corresponding region boundaries in L'. Denoting the depth of pixel p as d_p, we define term as $E_{rt}^g(f_r, f_t) = \eta_2 \exp(-\mu_2 \Delta_{rt})$ where,

$$\Delta_{rt} = \frac{1}{|B|} \sum_{i=1}^{|B|} |d_{p_i} - d_{q_i}| / \min(d_{p_i}, d_{q_i})$$

We set parameters, η_1 = 5.0, μ_1 = 0.1, η_2 = 5.0 and μ_2 = 10.0.

Discussion. An alternative to our approach would have been to include both the pairwise terms in a single energy function. However, this would require setting appropriate weights for the two terms, which is difficult in general. Occasionally, when the object is camouflaged against the background, the appearance cue can be less reliable than the depth cue, whereas when the depths along the region boundaries are approximate, the depth cues can be less reliable. To avoid this interplay between the two pairwise terms in ambiguous cases, we chose to solve two energy functions as described and combine their results later. As the pairwise MRFs are small, there is negligible overhead in doing this. In practice, we often find the two segmentation F_r^a and F_r^g to be identical. In general, our approach has a slight bias towards over-segmenting the object, but this is robustly handled in the final stage of our algorithm, described below.

5 Multi-view segmentation

The final stage in our approach exploits multi-view constraints to refine all the segmentations, which is similar to enforcing *silhouette consistency* in multiple views [9, 20]. The binary labeling $\{F_r\}$ of regions computed in the previous stage, induces a binary segmentations of pixels. We denote this initial labeling as F^i and the final labeling as F and use a subscript j to indicate the j-th image in the sequence of N images, whenever necessary. To compute F, we perform energy minimization on a pairwise binary MRF defined on a 4-connected pixel grid (similar to the first stage of the algorithm described in Section 3). Here we minimize the following energy function,

$$E(F) = \sum_{p \in P} E_p^O(f_p) + \sum_{p \in P} E_p^A(f_p) + \sum_{(p,q) \in \mathcal{N}} E_{pq}(f_p, f_q) \quad (6)$$

The unary terms $E_p^O(f_p)$ and $E_p^A(f_p)$ measure the cost of assigning pixel p to label $f_p \in \{0, 1\}$ and the pairwise term $E_{pq}(l_p, l_q)$ penalizes label disagreement and has the same definition as the pairwise term in Equation 2. We use the s-t mincut algorithm to exactly minimize the energy function.

Multi-view Objectness unary term (E^O). This term measures the *objectness* of a pixel, a term that was defined in Section 4. Unlike earlier, where the whole image was considered to conservatively estimate a pixel's objectness, we now use the available segmentations $\{F^i\}$. For robustness, we compute a confidence-weighted estimate in two passes, where the first pass computes a per-pixel weight reflecting the confidence in the pixel's objectness. For each foreground pixel p in F_k^i, we warp it into every other image using the depth estimate computed in the first stage, and compute a fraction $w_p \in [0, 1]$, indicating how often the warped pixels lies in the foreground in all the images. This is done by using the semi-dense depthmap recovered in Section 3.1, projecting the 3D point onto all the views and computing the fraction of the views in which the point lies within the foreground. For background pixels p in F_k^i, we set $w_p = 0$.

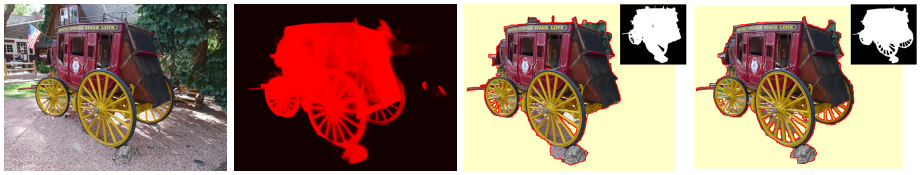
In the second pass, we warp every pixel p in the j-th image into every other image and set its objectness z_p to the mean of the set of numbers $\{w_q\}$, where q denotes the warped pixel in the j-th image and w_q its confidence computed in the first pass. We define $y_p = (1+\exp(-(z_p - \mu_1)/\sigma_1))^{-1}$ where $\mu_1 = 0.25$, $\sigma_1 = 0.1$. and define the unary term for the pixel in terms of y_p for the binary labels as follows: $E_p^O(1) = -\log y_p$, and $E_p^O(0) = -\log(1 - y_p)$.

Appearance unary term (E^A). We threshold the objectness estimates to select pixels that we are confident about being in the set of foreground and background pixels respectively [6]. Appearance models based on GMMs of colors are now trained for the foreground and background set, using model selection to determine the appropriate number of mixture components. The implementation details are similar to the first stage (Section 3.2). The penalty E_p^A is set to the negative log likelihood under the foreground and background appearance models (see Equation 3 for details). Fig 4 shows an example demonstrating the advantages of multi-view reasoning in our method. We draw the reader's attention to the two rear carriage wheels in this example, which get accurately segmented during the final stage.

6 Results

Datasets. We tested our algorithm on eight datasets and perform ground-truth evaluation for six datasets, which are summarized in Table 1. These datasets pose several challenges for existing approaches as the object and background color distributions overlap significantly in many cases. Also in some images, the object contours are very faint due to low image contrast. Illumination changes across images, diffused inter-reflections between object and backgrounds in many cases weakens the discriminability of the appearance cues. The objects in these sequences contain thin structures and have complex

[6] Concretely, we select pixels with $y_p > 0.7$ and $y_p < 0.3$ to serve as the selected foreground and background pixels.



(a) (b) (c) (d)

Fig. 4: (a) One of the images in the CARRIAGE sequence. (b) The corresponding multi-view objectness measure. Foreground segmentations (c) before and (d) after the final multi-view consensus step. Notice how the the initial coarse segmentations are refined and the missing wheel and gaps between the wheels are recovered. In our formulation, the stone next to the rear carriage wheel gets segmented as part of the object.

topologies. The presence of textureless surfaces, strong specularities and reflections in some cases poses additional challenges for an automatic segmentation approach.

Evaluation. To evaluate our method, we used Grabcut [25] to segment the objects in every image and treated these results as ground truth. Obtaining the ground truth interactively using Grabcut [25] on our sequences was an extremely tedious process. For instance, accurately segmenting the car, where glass windows reflect the background, the carriage and bicycle wheels where the background is seen through holes in the object, took up to 3 minutes on a single image and multiple user interactions. We evaluated our method using the popular *intersection over union* metric [20], which is computed as the ratio of the size of the set intersection to that of the set union of the computed and ground truth segmentation. The accuracy of our method across all datasets on average was $99.1 \pm 0.8\%$. To evaluate the accuracy more strictly, we mark boundaries in the segmentation and compute the unsigned distance transform with respect to these. This distance map is thresholded to obtain a band of pixels around the boundary over which the accuracy is recomputed. Table 1 shows the accuracy for various thresholds. With a threshold of 10 pixels, the average accuracy of our method was $90.9 \pm 5.6\%$. Fig 5 shows a few examples of segmentations and corresponding depth maps recovered by our method. More detailed results are presented on our website [7].

Comparisons. Fig 6 reports a quantitative comparison of our method with a state of the art unsupervised cosegmentation approach [32]. Our method is consistently more accurate on all four sequences used in the comparison. The assumption in [32], that the background appearance changes across images more quickly than the foreground appearance often causes the background to be misclassified as foreground on many of our examples. Our method is also significantly more flexible than existing automatic multi-view segmentation methods [8, 9] which rely on the fixation condition to learn the object's appearance model. However, this may be difficult or even impossible for objects with complex topologies such as the BICYCLE as the background is consistently seen through the object. In other cases, where the foreground contains multicolored objects such as in the CHAIR1 sequence, a single fixation point is unlikely to provide sufficient representative samples for learning a color model for the complete foreground object. In comparison, our approach makes no assumptions about the shape or appearance of the object of interest.

[7] http://chenlab.ece.cornell.edu/projects/MultiviewObjectCoseg

Fig. 5: [COLUMNS 1 – 5] Results on CHAIR1, CHAIR2, BICYCLE, BIKE and CAR sequences respectively. **[Row 1-3]** A sample image from each dataset is shown along with its piecewise-planar surface labeling and the corresponding depth-map computed by our method. **[Row 4]** The final segmentations recovered by our method. Notice how thin structures and holes in objects with complex topologies are accurately segmented (see columns 1–3). Our method accurately deals with camouflage where the object and background colors are similar (see columns 4 and 5), and is also robust to the presence of strong specularities and reflections (see column 5).

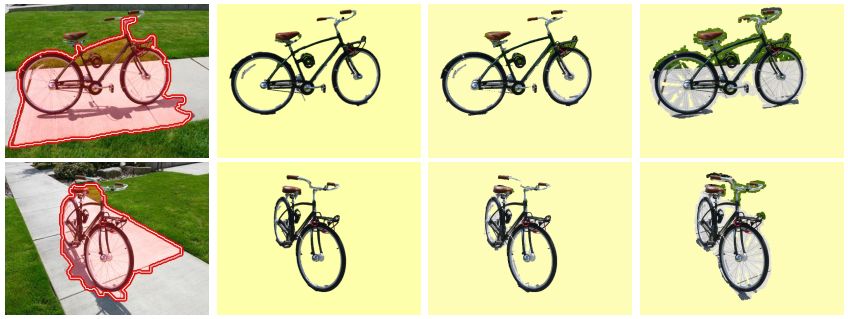
Name	#Images	Acc-2	Acc-5	Acc-10	Acc-Full
COUCH	9	87.5 ± 2.0	93.9 ± 1.2	96.4 ± 0.7	99.6 ± 0.1
TEDDY	15	69.0 ± 4.9	79.5 ± 5.3	86.9 ± 3.8	98.8 ± 0.4
BIKE	34	83.8 ± 3.8	89.8 ± 4.1	92.7 ± 3.9	99.4 ± 0.4
CHAIR1	17	88.0 ± 4.6	91.4 ± 3.9	93.9 ± 3.1	99.2 ± 0.4
CHAIR2	45	90.5 ± 1.7	94.0 ± 0.8	95.8 ± 0.6	99.5 ± 0.1
CAR	45	74.8 ± 3.3	80.8 ± 2.9	84.2 ± 2.9	98.0 ± 0.7

Table 1: The percentage of correctly labeled pixels (mean$\pm$std. dev.) in the segmentations computed by our method is listed for six datasets (see text for details).

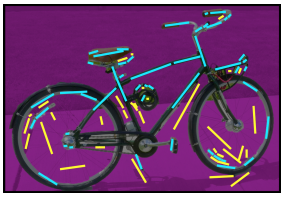
Although our piecewise planar stereo method is closely related to [5], their method cannot ensure that the foreground will be segmented as a single object. This makes it difficult to compare the methods quantitatively. A C++ implementation of our approach runs in under 2 minutes on a PC with an Intel 3GHz Xeon processor and 4GB RAM on a single 640 $\times$ 480 resolution image. Approximately one minute is spent on the piecewise planar depth map estimation step. In comparison, the method proposed in [5] takes 20 mins. on a single Middlebury image pair.

7 Conclusions

In this work, we have proposed an unsupervised algorithm to obtain the joint multiview foreground segmentation. We have developed a novel approach that combines multi-view cues and appearance cues in a hierarchical reasoning framework to extract out the



Vicente'11 [32] Grabcut [25] **Ours** PMVS [14]



Name		Vicente'11 [32]	PMVS [14]	Ours
BIKE	Acc-10	68.1 ± 6.7	61.0 ± 3.9	$\mathbf{90.0 \pm 4.9}$
	Acc-Full	88.9 ± 6.3	96.0 ± 1.8	$\mathbf{99.1 \pm 0.7}$
BICYCLE	Acc-10	56.5 ± 3.1	56.2 ± 1.8	$\mathbf{89.8 \pm 3.8}$
	Acc-Full	81.7 ± 8.9	89.1 ± 3.9	$\mathbf{98.0 \pm 0.5}$
CHAIR1	Acc-10	73.3 ± 4.8	72.7 ± 2.1	$\mathbf{93.9 \pm 3.1}$
	Acc-Full	86.9 ± 7.8	96.6 ± 0.4	$\mathbf{99.2 \pm 0.4}$
CAR	Acc-10	74.4 ± 5.3	59.6 ± 4.3	$\mathbf{83.2 \pm 1.1}$
	Acc-Full	91.8 ± 4.3	91.2 ± 5.5	$\mathbf{97.9 \pm 0.6}$

Fig. 6: COMPARISONS: Two out of 61 images from the BICYCLE sequence are shown. **[Column 1]** shows results from Vicente et. al. [32] in the red overlay. **[Column 2]** shows results from Grabcut [25] with exhaustive user input. **[Column 3]** shows our results. **[Column 4]** shows results generated from a 3D reconstuction (PMVS [14]) with manual segmentation. Our segmentations are accurate on thin structures such as the handle-bars and wheel rims and visually comparable to Grabcut [25] on this example. The interactive segmentation took 4 minutes during which the user had to provide about 80 strokes to obtain a perfect result. [BOTTOMLEFT] The foreground and background strokes drawn by the user are shown in blue and yellow respectively. [BOTTOMRIGHT] Quantitative accuracy of our results compared to [14, 32].

foreground object across the multiple views. As we show via quantitative and qualitative results, our algorithm accurately handles a wide variety of objects that pose challenges due to specular surfaces, diverse and overlapping color distributions, complex occlusions, and thin structures.

References

1. X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. In *SIGGRAPH*, 2009.
2. D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. *CVPR*, 2010.
3. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999.
4. M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. *CVPR*, 2010.
5. M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereo - joint stereo matching and object segmentation. *CVPR*, 2011.
6. Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001.

7. M. Brown and D. G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *3DIM*, pages 56–63, 2005.
8. N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28:14–25, 2010.
9. N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic object segmentation from calibrated images. In *CVMP*, 2011.
10. D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23:905–914, 2004.
11. A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR*, pages 53–60, 2006.
12. A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *ECCV*, 2008.
13. Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, 2009.
14. Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.
15. D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR*, 2010.
16. J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *ICCV*, 2003.
17. H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 30(2):328–341, 2008.
18. D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.
19. A. Kowdle, D. Batra, W. Chen, and T. Chen. iModel: Interactive co-segmentation for object of interest 3d modeling. In *ECCV - RMLE Workshop*, 2010.
20. W. Lee, W. Wontack, and E. Boyer. Silhouette segmentation in multiple views. *PAMI*, 2010.
21. V. S. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *PAMI.*, 32(8):1392–1405, 2010.
22. L. Mukherjee, V. Singh, and C. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009.
23. L. Quan, J. Wang, P. Tan, and L. Yuan. Image-based modeling by joint segmentation. *IJCV*, 75:135–150, October 2007.
24. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
25. C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.
26. C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006.
27. I. Simon and S. M. Seitz. Scene segmentation using the wisdom of crowds. In *ECCV*, pages 541–553, 2008.
28. S. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009.
29. N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006.
30. M. Sormann, C. Zach, and K. Karner. Graph cut based multiple view segmentation for 3d reconstruction. *3DPVT*, 0:1085–1092, 2006.
31. H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, pages 532–539, 2001.
32. S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.
33. J. W. Weingarten, G. Gruener, and R. Siegwart. Probabilistic plane fitting in 3d and an application to robotic mapping. In *ICRA*, pages 927–932, 2004.
34. J. Xiao, J. Chen, D.-Y. Yeung, and L. Quan. Structuring visual words in 3d for arbitrary-view object localization. In *ECCV*, 2008.
35. J. Xiao, J. Wang, P. Tan, and L. Quan. Joint affinity propagation for multiple view segmentation. In *ICCV*, 2007.