Multi-view Object Extraction with Fractional Boundaries

Seong-heum Kim, Yu-Wing Tai, *Senior Member, IEEE*, Jaesik Park, *Student Member, IEEE*, and In So Kweon, *Member, IEEE*

***Abstract*—This paper presents an automatic method to extract a multi-view object in a natural environment. We assume that the target object is bounded by the convex volume of interest defined by the overlapping space of camera viewing frustums. There are two key contributions of our approach. First, we present an automatic method to identify a target object across different images for multi-view binary co-segmentation. The extracted target object shares the same geometric representation in space with a distinctive color and texture model from the background. Second, we present an algorithm to detect color ambiguous regions along the object boundary for matting refinement. Our matting region detection algorithm is based on information theory, which measures the Kullback-Leibler (KL) divergence of local color distribution of different pixel-bands. The local pixel-band with the largest entropy is selected for matte refinement, subject to the multi-view consistent constraint. Our results are high-quality alpha mattes consistent across all different viewpoints. We demonstrate the effectiveness of the proposed method using various examples.**

***Index Terms*—Multiple image co-segmentation, Multi-view object segmentation, Natural image matting**

I. Introduction

Multi-view object extraction aims to simultaneously segment a foreground object from multiple images, each captured at different viewpoints of the target object. This is one of the most important steps in image-based rendering, editing, and many computer vision, graphics, and image processing tasks.

Early approaches to multi-view object extraction often assumed a well-constrained, indoor studio setup with strict illumination and no background clutters [1], [2], [3], [4], [5]. Recent approaches are able to automatically co-segment a multi-view object in natural environments by using either common appearance models in images or geometric constraints across viewpoints. Some reliable solutions [6], [7], [8], [9], [10] utilize three features: bounding-volume prior from camera poses, appearance models under geometric constraints, and iterative Markov Random Field (MRF) optimization. Specifically, it initializes color models from projections of a visual hull by all cameras. In this procedure, segmentations of each viewpoint are geometrically related in the space. The appearance models and foreground masks are simultaneously updated until they converge in the MRF optimizations.

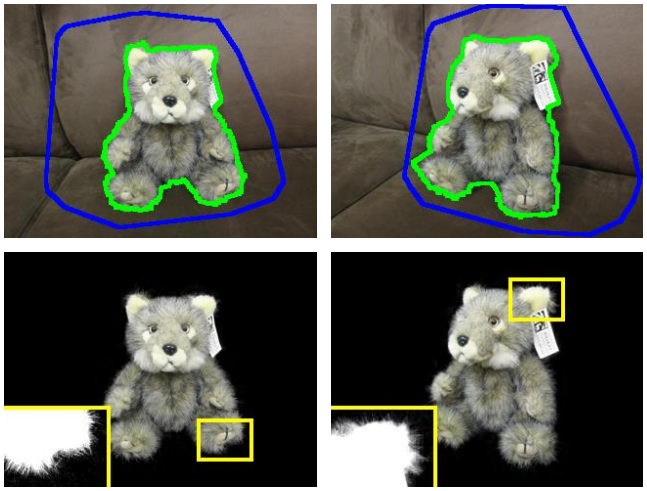


Fig. 1. Multi-view object extraction with fractional boundaries. Top: We assume that the target multi-view object is bounded by the overlapping space of camera viewing frustums (blue lines). Our approach first estimates the binary masks of the target object across the viewpoints (green lines). Bottom: The soft boundaries are obtained after the matting refinement. Our results for two viewpoints on the Teddy are displayed.

However, these approaches only show rough segmentations in relatively low resolution images and do not perform matting to resolve fractional boundary issues. Moreover, there is a problem with automatic initialization when the visual appearance of the target object cannot be simply modelled by color Gaussian mixture models (GMMs).

In this paper, we present a multi-view matte estimation method on top of the previous approaches [7], [8], [10], [11], [12], [13] which not only estimates binary masks, but also soft alpha mattes of a foreground object. Our system utilizes the calibrated camera poses and sparse point clouds acquired from structure-from-motion (SfM) [14], [15], [16]. The initial contour of the foreground region is obtained from the assumption that an object is bounded by the convex hull of camera viewing frustums. Based on the initial contour, we can estimate the appearance models of foreground regions to get a tighter bound of the foreground object. Our appearance model comprises a color model and a texture model. After iterative MRF optimization, we obtain binary segmentations, which are close to the silhouette of the 3D model, and consistent boundaries are obtained from the projected 3D curves. The second step refines these contours so that they are accurately located at object boundaries. The unknown regions that contain

Seong-heum Kim and In So Kweon (Corresponding Author) are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea.
E-mail: shkim2@rcv.kaist.ac.kr, iskweon@kaist.ac.kr

Yu-wing Tai is with SenseTime Group Ltd, Hong Kong. Jaesik Park is with Intel visual computing lab., CA, United States of America.
E-mail: yuwing@gmail.com, jaesik.park@intel.com

Manuscript received XXXX XX, XXXX; revised XXXX XX, XXXX.

color mixing of foreground and background are determined by evaluating the distribution of colors within local pixel-bands. Finally, we solve for the fractional boundaries of the target foreground object by adopting the matting Laplacian [12] subject to the soft multi-view consistent constraint.

This paper extends the work reported in our previous publication [6]. We summarize the major differences as follows. First, we propose a better solution to the initial binary mask segmentation, which considers not only geometric constraints across images, but also color and texture information for more accurate foreground estimation. The efficiency of this step is also improved by the adoption of superpixel representation. Second, instead of simply considering local windows of various sizes, we exploit more rich representation of local windows and adopt a dynamic programming approach to estimate the unknown regions that have the maximum sum of entropies along object boundaries. This improvement allows us to estimate unknown regions more accurately, and hence, improve the quality of estimated alpha mattes. Finally, for challenging cases in which the automatic approach does not perform well, we discuss how to include user interactions to enhance the performance of our algorithm.

We evaluate our algorithm using various real-world images. Based on the results of qualitative and quantitative experiments, we claim three advantages of our algorithm. First, we automatically estimate a tight bound for a foreground object by the convex hull of visible SfM points. Second, the appearance model of the foreground object is more robust than those reported in previous literature when similar texture patterns are observed from multiple viewpoints. Third, our method provides high-quality fractional boundaries using the estimated trimap. Our approach does not require clean background images to separate foreground layers. In addition, our algorithm can efficiently handle high-resolution images with minimum user interventions because the optimization and matting procedure is only performed on the uncertain regions of the trimaps.

II. Related work

Our method deals with a common texture model for multiview object segmentation, geometric foreground representation, and automatic matting for multiple images. The relevant publications are reviewed in this section.

Multiple images co-segmentation. Co-segmenting multiple images refers to the problem which segments two or multiple images using a common appearance model of the target objects. One of the key ideas in co-segmentation is to use common energy terms coming from consistency measurements across different input images, assuming that they share some statistical similarity, such as colors, textures, shapes and other appearance properties [17], [18], [19], [20], [21], [22], [23].

Recent approaches exploit rich representations for the appearance modeling of target objects. They consider mid-level features in their compatible segmentations or information transfers between images [24], [25]. Chai *et al.* [24] showed foreground-background co-segmentation of image sets that is scalable to large datasets. The background appearances are collected and modeled in a high-dimensional feature space, and the unsupervised co-segmentation is driven by these training image sets. Inspired by the method in [24], our texture representation uses superpixels in multiple images and utilizes gradient features to describe superpixels. This approach shows better performance than solely utilizing GMMs for color observations under natural illumination conditions.

To handle more complex illumination, material, and other physical properties, Oquab *et al.* [25] also demonstrated that large-scale image collections can be utilized to train the general appearance of class specific objects. We believe classifying features is a promising direction when the target object is in a natural scene. However, we do not assume that a large number of training examples is available, and we do not include any category learning process.

User interactions, such as bounding boxes and foreground or background scribbles for each image, can be effectively shared to allow for the dissemination of information [29], [30]. At every iteration, these systems require user corrections for a few images, and all color models are updated according to the guidance. It naturally develops interactive image segmentation [26], [27], [28] to the multiple image co-segmentation context. However, they do not consider any geometric correlations among input images even when some of them share a common 3D space. Extending this work, our approach considers the geometry of the target scene, which is powerful clue for improving the co-segmentation performance.

Multi-view geometric constraints. To enforce geometric constraint on the multi-view images, previous studies have assumed that an object is consistently observed with calibrated cameras. Some early works have suggested the following geometric representations. Zeng *et al.* [31] introduced a visual hull constraint on the over-segmented images to determine the foreground segments. Yezzi *et al.* [32] used a level set method of evolving 2D contours consistent with the 3D space, which is under the strict assumption that a scene is composed of several homogeneous backgrounds and strong irradiance discontinuities. Snow *et al.* [33] applied the geometric constraint to their probability functions so that static background models could be successfully merged into the 3D representation. Many background substraction techniques take this probabilistic approach, but the assumption of clean backgrounds does not generally hold in the real-world environment [34]. Here, a clean background indicates an image without the target object, which is intended for easy background subtraction. Our method does not require clean backgrounds nor any strict geometry assumptions regarding the input images, in contrast to these previous studies [33], [34], [35], [11], [36].

Recent approaches have utilized the MRF optimization technique for better segmentation results. Sormann *et al.* [37] proposed a graph-cut-based fast multiview co-segmentation algorithm that uses geometric constraints with the proper initialization. Other studies, such as [38], [39], [11], [40], are jointly optimizing 2D segmentation and its 3D reconstruction under the MRF frameworks. In these approaches, several assumptions, such as object locations, clean backgrounds, or user strokes for the initialization are utilized.

Some approaches for multi-view co-segmentation have accurately estimated the dense geometry of an object, acquired from the SfM pipeline [41], [8]. To gain the reliable geometry, these methods require a relatively large number of input images. Their systems also take computational resources to refine its initial geometry. This is because the quality of 3D reconstruction directly affects the segmentation result. Moreover, the dense reconstruction is often a time-consuming process to obtain 2D segmentations as its side-products.

Other approaches using fewer input images find simple geometric representations, such as epipolar lines on the pixel domain or sparse sample points in 3D space [7], [42], [9], [10]. During iterative optimization, foreground masks of the appearance models are updated and checked whether the current binary labels are also correct from the other viewpoints.

Their systems, without any user-given priors, start with initial masks from camera poses assuming that the object is placed at the intersection of all camera views. For more strict configuration, Campbell *et al.* [38], [39] assumed that the target object is fixed at the center of the visual hull, and they collected definite foreground samples near the camera centers in images. However, in the challenging cases, their appearance models are not able to manage foreground-background ambiguities when the object is loosely bounded in the images.

In contrast to previous approaches, our system starts with the SfM pipeline and finds the initial space of interest that is fully visible to all input cameras. Instead of dense geometry acquisition, the SfM pipeline in our approach is only used to determine the convex hull of the visible SfM points. In addition, our approach is based on the simple geometric representation that defines 3D reference points in space and links between superpixels in images. Without requiring the 3D structure refinement stage, our approach samples regular grids of the initial convex hull and only keeps physically meaningful surface samples through the visibility computation at every iteration. Even in challenging cases, our appearance models can seek the texture patterns of an object so that our MRF optimization overcomes possible local minima.

Natural image matting. Matting is the process of segmenting a foreground object with its fractional boundaries. Conventional sampling-based, affinity-based approaches require user inputs in a form of a trimap to specify foreground, background, and uncertainty regions [12], [13], [43], [44], [45], [46]. In this section, we present some examples that are relevant to our trimap generation in terms of camera configurations.

For single-image matting, Rhemanon et al. [13] dealt with fractional boundaries in high-resolution images. This approach can be considered a two-phase method with the first phase of the initial trimap estimation using graph-cut and the second phase of matte refinement. Another approach allows a user to roughly trace the boundary of the target, and the unknown regions are adaptively determined according to neighboring contents [47].

For narrow-baseline image matting, the variances of colors in pixel correspondences at depth planes are measured to automatically identify trimaps and get mattes for an array of cameras [48]. This idea has turned out to be quite suitable, and it has been extended to multiview camera systems and light field images [49], [50]. However, these approaches share the same baseline with almost identical appearance of foreground objects in each input image. Hence, it only works at the front-parallel configuration of cameras, and the generalization is not very straightforward. Hasinoff *et al.* [51] also formulated boundary matting at occlusion boundaries as estimating 3D curves and foreground colors. They considered the boundary curve as piecewise linear, parameterized points in 2D coordinates, and designed an objective function by putting the model into the well-known compositing equation [52].

For wide-baseline image matting, Sarim *et al.* [35] partially applied the epipolar line constraint to isolate shadow regions in pixel spaces and performed matting as post-processing. Recently, Wang *et al.* [36] removed the boundary artifacts of video objects by having static background models.

In contrast to these previous works, we fully associate all camera viewpoints in the absolute 3D coordinate and initialize MRF energies in natural scenes. Our approach considers the geometric relations in input images and minimizes user interactions. We also have added a soft constraint to the matting equation so that the projections of our common geometric representation are implicitly preserved in the final mattes.

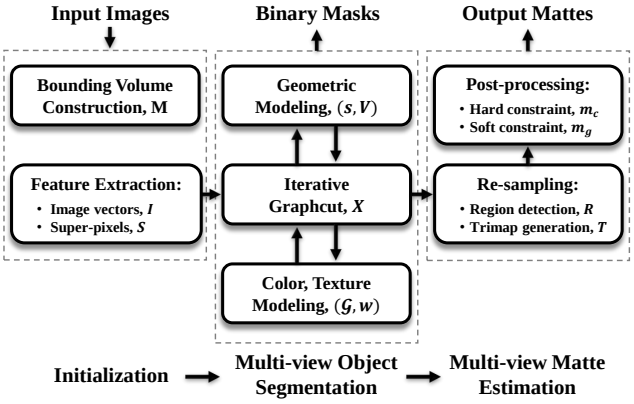


Fig. 2. System Overview. See Sec. III for details.

Symbols	Meanings
$\mathbf{M} \in \{\mathbf{F}, \mathbf{B}\}$	intial projections from a bounding volume
$I_k^n \in \mathcal{I}$	k-th pixel vector on the n-th image
$S_k^n \in \mathcal{S}$	k-th superpixel descriptor on the n-th image
$P_k \in \mathcal{P}$	k-th anchor point in the 3D space
v_k^n, V_k^n	score map for coherent pixels or superpixels
$\mathcal{G}_f$, $\mathcal{G}_b$	foreground, background GMMs
$\mathbf{w}$	discriminative model parameters in SVM
$\mathbf{x}, \mathbf{X} \in \{\mathbf{F}, \mathbf{B}\}$	binary labels for all pixels or superpixels
R_k^n	parameter set for the k-th boundary sample
$T_k^n \in \{F, B, U\}$	trimap segmentation in the n-th viewpoint
m_c, m_g	hard, soft constraints for alpha matting
$\alpha \in [0, 1]$	output mattes for all the input viewpoints

TABLE I
NOTATIONS USED IN THIS PAPER.

III. OVERVIEW

Our system performs an inference procedure to detect the foreground mask at superpixel-level at a low resolution. After that, we estimate the multi-view trimaps and mattes at the original image resolution. This coarse-to-fine approach reduces the overall processing time. Figure 2 shows the pipeline.

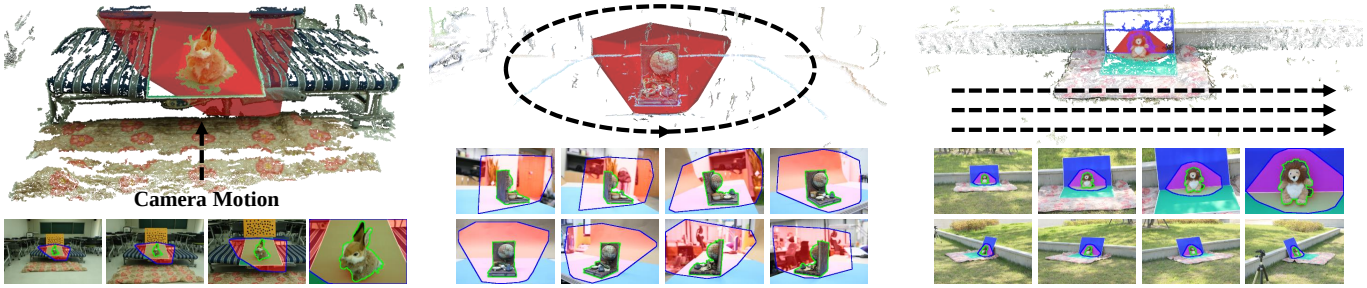


Fig. 3. Bounding volume construction on the Rabbit, Earth, and Lion1 datasets. Left: A camera moves toward the target object in a scene. Middle: A camera moves around the target object. Right: A camera moves around the target object at various distances. The overlapped camera viewing frustums are indicated by the red convex volume overlaid on the reconstructed point clouds of each scene. The blues lines in each input images are the projections of the bounding volume, and the green lines indicate the obtained binary masks of the target objects.

For the automatic initialization of the foreground masks (details in Sec. IV), we assume that the target object is located inside the convex space of reconstructed 3D points that are observable to all the cameras as illustrated in Fig. 3.

Thus, the projections of this convex volume become our initial masks $\mathbf{M}$. Once we get $\mathbf{M}$, we perform superpixel segmentation. Each superpixel consists of color and gradient components. Based on the initial $\mathbf{M}$, we build appearance models, which consist of color GMMs (denoted as $\mathcal{G}_f$, $\mathcal{G}_b$ for the foreground and background model) and a support vector machine (SVM) classifier (the linear weight vector is denoted as $\mathbf{w}$) of foreground/background regions. The color GMMs model color distribution, and the SVM classifier models texture information. To consider geometric information, the regularly distributed 3D points in the convex space are used to connect the correlated superpixels in the foreground masks. We name these 3D samples as *anchor points* $\mathcal{P}$. Each anchor point has occupancy probability $\mathbf{v}$, which indicates the geometric coherency of superpixels. We regard the superpixels and anchor points as nodes, and we build a single graph model. Here, each anchor point becomes an auxiliary node in the graph. The graph labeling problem is iteratively solved by the MRF optimization.

In the second phase (details in Sec. V), we estimate trimaps $\mathbf{T}$ and alpha mattes α of the target object in multi-view images. The key idea is to vary the shape and width of the trimaps according to the degree of color mixing between foreground and background. For instance, if there is a sharp edge between foreground and background, and their colors are clearly distinguished, then the band of the uncertain region should be thin. On the contrary, the band becomes thick if a local region shows the color mixing effect. We use the KL divergences in a local boundary region to measure the degree of foreground-background uncertainty. After trimap estimation, we also compute geometrically consistent masks m_g, which will be used as a soft constraint to the matting equation. Our approach shows the highest quality of soft masks for multi-view images.

IV. Multi-View Object Segmentation

In the first phase, our goal is to estimate binary masks $\mathbf{X} = \{X^1, X^2, \ldots, X^N\}$ of the target object in multiview images, $\mathcal{I} = \{I^1, I^2, \ldots, I^N\}$, where N denotes the number of input images. We use superscripts to represent image indexes and subscripts to represent pixel or superpixel indexes with capital symbols. We denote $\mathcal{S} = \{S^1, S^2, \ldots, S^N\}$ as superpixel sets and $\mathcal{M} = \{M^1, M^2, \ldots, M^N\}$ as the initial masks of the bounding volume. We utilize the iterative graph-cut optimization to achieve our goal.

A. MRF formulation

The binary segmentation in the first phase is formulated as a single energy function in the MRF framework [27], [28]. Our objective function consists of the data term E_d and the neighborhood term E_n as follows. The data term is designed based on the appearance models E_a and the geometric model E_g. Similarly, the MRF allows the regularization terms, such as E_{nc} for similar colors, textures and E_{ng} for geometric linkages across viewpoints.

$$\begin{aligned} E_d &= \rho \cdot E_a + (1-\rho) \cdot E_g \\ E_n &= \lambda_{nc} \cdot E_{nc} + \lambda_{ng} \cdot E_{ng} \end{aligned} \tag{1}$$

The parameter ρ determines which data term is more reliable for the energy assigning of a node:

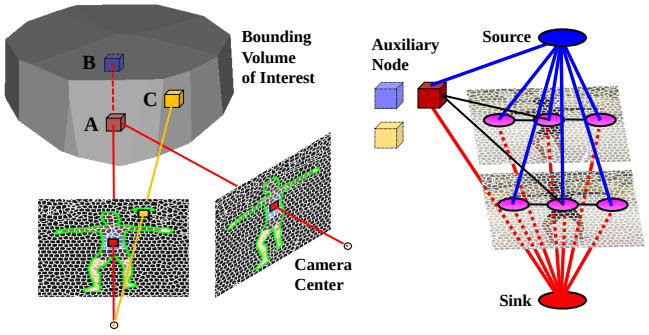
$$\rho_k = \frac{|Pr_c(I_k|\mathcal{G}_f) - Pr_c(I_k|\mathcal{G}_b)|}{|Pr_c(I_k|\mathcal{G}_f) + Pr_c(I_k|\mathcal{G}_b)|}, \tag{2}$$

where $Pr_c(I_k|\mathcal{G}_f)$ is the probability of the pixel I_k having the foreground label $\mathcal{G}_f$. We use subscript c on the probability Pr because it is a color-based probability term. When the colors of a pixel or a superpixel have similar metrics for both the foreground and background models, we give more weight to the geometric consistency term.

The neighborhood terms are weighted by λ_{ng} for geometrically linking nodes across the related viewpoints and λ_{nc} for considering color and texture linkages in each image. Detailed explanation of the terms in Eq. 1 and 2 are given in the next subsections.

B. Geometric representation

In our approach, the geometric coherence of the binary segmentations is evaluated for every MRF iteration. In this procedure, a foreground label in one image becomes the true foreground when a warped pixel position (correspondence point using the camera projection matrix) in the other images also belongs to the foreground regions. For this idea, we define *coherency score* v_k^n for pixel k. For superpixels, all the pixels in a superpixel share the same score. The coherency score for



(a) Geometric representation (b) Graph Model

Fig. 4. Geometric representation and its graph model. We keep visible 3D samples as anchor points in space. A is an anchor point linking two superpixels in the graph model. After the visibility check, B is removed. In some cases, 3D samples such as C are only visible to one viewpoint. C is referred to give geometric coherency for the superpixel, but it does not make any geometric linkages across images in the MRF optimization.

the superpixels is denoted as V_k^n. The scores are defined as the sum of binary labels x or X:

$$v_k^n = \frac{1}{N}\sum_{n'=1}^{N} x_{w_k^{n\to n'}}^{n'}, \quad V_k^n = \frac{1}{N}\sum_{n'=1}^{N} X_{W_k^{n\to n'}}^{n'} \tag{3}$$

In Eq. 3, the warping from image n to n' for pixel k is denoted as $w_k^{n\to n'}$. The coherency score is normalized to have a $[0,1]$ range. For the score evaluation, we define *anchor points* to connect superpixels across input images. The anchor points are uniformly sampled 3D points in the bounding volume. We perform a visibility check for the anchor points as illustrated in Fig. 4 to remove occluded samples.

The sampling rate is chosen such that one superpixel observes at least one 3D anchor in the 3D space. Thus, the geometric consistency can be guaranteed by enforcing the same binary label assigned to superpixels which observe the same anchor point. In our experiment, this approximate representation demonstrates good results to impose a penalty for consistent foreground mask estimation.

We model the geometric energy term of background using a sigmoid function:

$$E_g(X_k = B) = \frac{en_b}{1+\exp[-\lambda_v(V_k - V_{th})]} \tag{4}$$

The parameters of this function include a trade-off relationship between the recall and precision rates of predicting segmentations. In practice, we empirically find the parameter values, where the parameter $\lambda_v = 20$ controls the shape of energy functions, and V_{th} is usually set to 0.9 for the tolerance of thin foreground segments with possible calibration errors. Each energy connected to the background is bounded $en_b = 10$ as its maximum value. Similarly, the geometric energy for the foreground, $E_g(X_k = F)$, is defined as $en_b - E_g(X_k = B)$.

To consider the neighborhood terms, we construct inter-view links between superpixel nodes according to the 3D anchors. Our energy term for E_{ng} is defined as follows.

$$E_{ng}(X_p^i, X_q^j) \quad = \textstyle\sum_{i,j,p,q} |X_p^i - X_q^j| \cdot en_b \tag{5}$$

The neighboring term in the constructed graph is solved by considering all superpixels related to one another as sharing the common geometric model.

In comparison to SfM, our system does not refine 3D structures; rather, it concentrates on the 2D segmentation task. Even if the initial reconstruction is roughly given, we are only interested in the tight bounding volume and sparse, accurate anchor positions in it. We do not additionally compute the photo-consistency of these points as seeking a geometrically simple representation. For some missing foregrounds in several images, these connections are also extremely useful for propagating user strokes when they are available.

C. Appearance models

Our appearance models capture colors and texture patterns in superpixels. The appearance energy terms consist of the Gaussian mixture model (GMM) of per-pixel distributions E_c and Fisher kernel (FK) representations of each superpixel E_t:

$$E_a = \; E_c + \lambda_t \cdot E_t. \tag{6}$$

We build color GMMs for each image and classify the foreground, background texture patterns collected from all images. By controlling the weight λ_t for the texture term over the color term, we can control the influence of their cooperative effects.

In modeling the color term, we take the negative log of a probability, which converts an $\arg\max$ MAP problem into an $\arg\min$ energy minimization in the MRF framework. Also, we assign the texture term according to the margins of the respective superpixel descriptor from the hyperplane of the support vector machine (SVM).

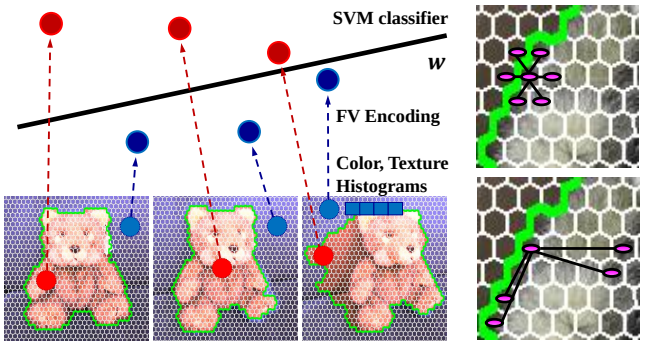
1) Color: In the color consistency measurements, we build a color model using GMMs [53], [54] for both foreground and background color distribution. In our case, we vectorize a color pixel in an image I_k^n as a nine-dimension vector, stacking the lab colors space and the RGB color space for two different Gaussian blurs.

Suppose we have the current binary segmentation of the foreground and the background, we can collect samples to build the GMM color model:

$$\left.\begin{aligned} Pr_c(I_k|\mathcal{G}_f) &= \sum_{c=1}^{c_{max}} w_f^c \cdot \mathcal{N}(I_k|\mu_f^c, \sigma_f^c) \\ Pr_c(I_k|\mathcal{G}_b) &= \sum_{c=1}^{c_{max}} w_b^c \cdot \mathcal{N}(I_k|\mu_b^c, \sigma_b^c) \end{aligned}\right\} \tag{7}$$

where $w_f^c \cdot \mathcal{N}(I_k|\mu_f^c, \sigma_f^c)$ indicates the weighted Gaussian component having mean μ_f^c and variance σ_f^c on the foreground label. We build the GMM color model in a normalized lab color space ranging between 0 and 1. This normalized lab color space gives more weight to the chromatic channels and less weight to the luminance channel to reduce the effects of shadings or shadows. We also apply the Gaussian smoothing to each RGB channel before building the GMMs to remove image noises and subtle image details to avoid over-fitting of the color distribution.

When we measure the distance of a pixel between the foreground and the background, instead of using all Gaussian functions $\mathcal{N}$, we use the 5 nearest Gaussian functions by defining w_f^c, w_b^c as the minimum metric in closer Mahalanobis distances. This is slightly different from the convention considering the number of samples covered by each Gaussian.



(a) Common texture model (b) Superpixel linkages

Fig. 5. Texture appearance model. We encode color and texture components into the superpixel descriptors. At every iteration, they train the linear SVM in the Fisher vector space. In MRF, one superpixel is connected to the nearest nodes, and the non-local linkages are formed by the similarity between these superpixel descriptors.

Consider the example shown in Fig. 14. The proportion of the hand pixel is much smaller than the other foreground pixels. Therefore, the convention GMM will disregard the hand region because it has fewer samples. In our task, we prefer to take all the details of an object because other false alarms can be effectively suppressed by iterative optimization. The color consistency measurements are normalized to satisfy $Pr_c(I_k|\mathcal{G}_f)+Pr_c(I_k|\mathcal{G}_b)=1$ for the proper energy terms, and the average of these energies in a group of pixels is assigned to their superpixel node in MRF.

2) Texture: Our key idea for encoding texture information is building *common texture* that is comprehensive foreground/background texture prior regardless of the viewpoints. To encode texture information, we take the luminance channel l in the lab space, and compute the x, y, xy, and yx directional derivatives of Gaussians at two different sigma scales, and Laplacians of Gaussians at three sigma scales. Then those 11-dimensional vectors in one viewpoint are added to the original 9-dimensional vectors of color components. The new 20-dimensional vectors for each pixel in all input images are clustered to create 64 GMMs, followed by description of all superpixels to Fisher vectors with respect to the global GMMs. In this manner, we get a descriptor S_k for superpixel k. After the normalization of superpixel descriptors as in [55], we train a linear SVM either using all positive and negative vectors and their labels $X_k^{'}=2\cdot(X_k-1/2)$ across viewpoints, or build multiple SVMs by having the samples only in the respective view [56]. This is illustrated as Fig. 5. The scores from the trained $\mathbf{w}$ give the texture-driven energies E_t to each superpixel [24]:

$$\min_{\mathbf{w},\mathbf{b}} \sum_{k=1}^{\mathcal{K}} l(X_k^{'}\cdot \mathbf{w}^T S_k) + \frac{1}{2}||\mathbf{w}||^2 \tag{8}$$

In building an initial graph with neighborhood connections, the superpixel descriptors are compressively represented as $S_k^{'}$ by the standard PCA projections. For greater computational efficiency, mean pooling on color vectors is another good approximate option. The pairwise term is implemented by the Potts model [28] in the superpixel segmentation context, where one node is connected to k-nearest neighbors in the $\mathcal{X}^2$ metric between the superpixel descriptors:

$$E_{nt}(X_p,X_q)=\sum_{p,q}|X_p-X_q|\cdot\exp(-\beta[\chi^2(S_p^{'},S_q^{'})]^2). \tag{9}$$

We normalize this distance function by setting the parameter $\beta=(2\langle[\chi^2(S_p,S_q)]^2\rangle)^{-1}$ as explained in [28]. According to convention, $\langle\cdot\rangle$ denotes the expectation of all the linked superpixels in one image. In the implementation, each superpixel is connected to eight adjacent nodes and is non-locally linked at most with eight similar descriptors in the cost space.

Except for the definition of texture [57], we link superpixel descriptors of one image in a similar way as in [10]. However, we develop the way to have a discriminative appearance model so that the gradient magnitudes do not lose directional information. This texture model is sufficiently representative, especially when we have similar texture parts in foregrounds with blurry backgrounds.

D. Energy optimization

In practice, labeling all pixels at the original image resolution is a time-consuming task. Instead, we take one or two additional coarse resolutions for the rough superpixel segmentations before we start the pixel-level optimization using the initial energy. During iterative refinement of the multi-scale segmentations, we increase the weight of λ_{ng}. This is because we believe that the 3D hypothesis is not reliable in the beginning, but gradually our 3D surface samples become accurate and dense enough to cover pixel-level score maps.

In the pixel segmentation, we use the typical contrast term for eight connected grids and add view-to-view linkages coming from the final 3D surface structure. We define the neighborhood energy for two adjacent nodes x_p and x_q using a color energy E_{nc} and a geometric energy E_{ng}:

$$\left.\begin{aligned} E_{nc}(x_p,x_q)&=\sum_{p,q}|x_p-x_q|\cdot\exp[-\beta(I_p-I_q)^2]\\ E_{ng}(x_p^i,x_q^j)&=\sum_{i,j,p,q}|x_p^i-x_q^j|\cdot en_b, \end{aligned}\right\} \tag{10}$$

where β is the same value used for Eq. 9. In our experiment, most cases required less than three iterations to reach stable labeling results. When we could not solve the MRF with more than eight inputs at a high resolution due to memory limitation, we divided them into eight-viewpoint subsets to be respectively optimized after re-computation of the visibility maps. However, we also found that the 3D structures derived from eight views are good enough to project to the other calibrated views. Hence, we were able to perform per-view graph-cuts using its appearance, geometric models with $\lambda_{ng}=0$. It is still useful to have such inter-view linkages coming from 3D samples [10] or SIFT-based correspondences [11] in inferring correct labels. However, once we had a good 3D model of interest in the space, we observed that additional regularization was no longer required because the simple projection scores v had already achieved geometrically consistent segmentations.

V. Multi-View Matte Estimation

Given the estimated binary labels $\mathbf{x}$ and the scores maps $\mathbf{v}$, our next step is to generate trimaps $\mathbf{T}$ and estimate alpha mattes $\boldsymbol{\alpha}$ of the foreground object.

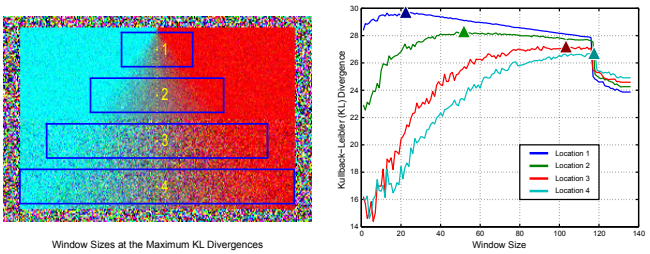


Fig. 6. KL divergences against the sizes of local windows. Left: We obtain the four optimal windows (blue boxes) at different Gaussian noise sigma. Right: The optimal size at the maximal KL divergences are denoted in the right KL curves. We can also note that the synthetic noise is the most severe outside the outer boundary. All window shapes exceeding the boundary get lower values as all KL curves drop steeply around 120.

A. Supporting region detection

We use the KL divergence to measure the amount of local color spread for matte region detection. The use of the KL divergence is inspired by the two-color line model in [12], which indicated that, for any local color distribution, the effects of color mixing can be approximated by a linear combination of two different colors. If the distribution of color samples is concentrated in the middle of the line model, we can judge that the mixing effect of two colors is strong. In contrast, if the color samples are concentrated at the two end points of the line model, we expect a sharp boundary between two regions. A similar approach of using KL divergence to evaluate the reliability of estimated mattes has been used in [6], [58].

The equation for measuring the KL divergence is given by:

$$f_{KL}(r_k, s_k) = \sum_{r'_k \in s_k} [d(I'_{r'_k}, C_0) \cdot \log \frac{d(I'_{r'_k}, C_0)}{d(I'_{r'_k}, C_1)}], \qquad (11)$$

where r_k is the uniformly sampled seed point along the boundaries, C_0 and C_1 are the RGB colors of the two end points on the line, respectively, and $d(\cdot,\cdot)$ is the Euclidean distance operator. In our approach, we measure the KL divergence of the local color distribution using the estimated line model. Instead of a regular grid to measure the divergence, we use the various shapes of local windows to adapt the window according to the boundary shape. The KL divergence is measured with various window sizes, ranging from 7×7 to 81×81 pixels, in three possible shapes (square, thin, thick) having 2, 2, and 9 types of offsets respectively. The windows s_k are illustrated in Fig. 7.

The effect of KL divergence in evaluating color spread is evaluated in Fig. 6. We plot the curve of the normalized KL divergence against the size of a local window. For a region with a sharp boundary, increasing the local window reduces the value of the KL divergence. On the other hand, if a region requires matting, the value of the KL divergence increases with the window size until it reach the optimum region in the sense of the maximum entropy. Therefore, the matting region is obtained by selecting the window shapes, sizes s_k with maximum KL divergence of the local color distribution.

B. Parameter optimization

After we evaluate various KL divergences with various window shapes, we increase the local window size to effectively separate the foreground and background regions. This is based on our observation shown in Fig. 6. Specifically, at the positions r_k^*, the window size increases in 13 different directions. Here, the control point r_k^* is a better localized position of r_k using the maximum response of the Gaussian (DoG) filter difference. For this purpose, we modify the level set implementations as in [59].

(a) Possible shapes and offsets (b) Parameter optimization (DP)

Fig. 7. Parameter optimization. (a) At a fixed size, we have different shapes with shifting offsets of a window depending on the content of its supporting region. (b) We determine window parameters that maximize the cumulative sum of KL divergences along a constrained path. After regularization, the window shapes and sizes along the path change smoothly.

With this procedure, all sampled positions have the KL divergences of the possible window shapes and sizes. The measured divergences are recorded in form of cost volume, and we utilize dynamic programming to maximize the sum of KL divergences along all object boundaries. By indexing the optimum size and shape of a detecting window s_k^* at the best contour point r_k^*, we can easily classify the foreground and background colors as well as the mixed colors. Examples of our various windows shapes and detected windows in a real-world image are illustrated in Fig. 7.

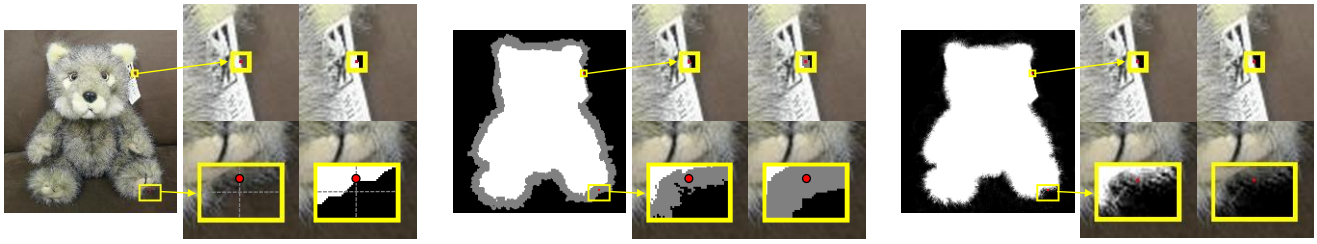
C. Trimap optimization

In the previous section, we described the detection of the optimal local windows of trimaps. We further refine the trimap by MRF optimization. This is done by α-expansion [60] with an objective function similar to that of the first phase. All pixels in the estimated regions are assigned to new labels $T_k \in \{F, B, U\}$ before all the windows are combined with the remaining labels $\mathbf{x}$ to constrain the whole image. Each α-expansion iteration can be identically performed by a series of single graph-cuts, and the reasonable local optimum of the objective function in each window is shown. The MRF equation for trimap optimization is defined as follow:

$$E(x') = E_c(x') + \lambda_g \cdot E_g(x') + \lambda_{nc} \cdot E_{nc}(x', y') \qquad (12)$$

Readers may refer to individual terms in the first-phase graph-cuts (Sec. IV) for the details, but a few aspects are slightly different from the previous formulation. For modeling geometric terms, three values are simply taken for energy functions, with $\lambda_v = 0.9$ for $x'_k \in F$, 0.7 for $x'_k \in U$, and 0.5 for $x'_k \in B$, and they make $E_g(x'_k = F) + E_g(x'_k = U) + E_g(x'_k = B) = en_b$ to give normalized, geometric energies to the graph. Thereby most ambiguities are handled in color models.

The main difference of color models compared to the first phase is that locality of the color samples is given by normalized x, y coordinate information. We also find the pixels using the central colors of a local line model to consider the uncertainty areas, and we build the third GMM to infer the label U. After MRF optimization, the combined trimap can be



(a) Color image (b) Trimap (c) Alpha matte

Fig. 8. Trimap generation and matting refinement. (a) A contour point is selected at the refined edge. The optimal window at the sample point is shown as the yellow box. (b) Trimap segmentation is performed using the color and geometric energies in MRF. The result is further post-processed with morphological filters. (c) Alpha matte and true foreground colors are obtained from the matting equations.

further refined with simple morphology filters to enlarge the unknown regions as shown in Fig. 8. Another way to build GMM for the matting regions is to blend all combinations of foreground and background GMMs as in [61], [13]. In comparison, our method selects blended samples mostly existing on a transition between two strong mean points in a local patch. This selection process captures the better mixed colors at sharp edges and foreground colors in thin structures, while it shows similar performance at blurry edges. In many real-world examples, these procedures are successful for our trimap conversions.

D. Matting refinement

Finally, given the trimap and foreground masks, we solve for the fractional boundaries only within the uncertainty regions. This efficient approach is possible because we have the estimated trimap. Our matting framework is based on standard Laplacian matting with additional constraints [12], [62]:

$$\begin{aligned}\arg\min_{\alpha^*} \alpha^T \mathbf{L}\alpha + \lambda_c(\alpha - m_c)^T \mathbf{W}(\alpha - m_c) \\ + \lambda_g(\alpha - m_g)^T(\alpha - m_g),\end{aligned} \tag{13}$$

where m_c is the trimap label acquired from Eq. 12 as a hard constraint. Before the Laplacian is solved, the hard constraints are post-processed by eroding and dilating the estimated foreground/background regions. Here, $\mathbf{W}$ is a diagonal matrix with its entries 1 if a pixel belongs to mixed pixels and 0 otherwise; $m_g = x^n$ is a soft constraint that utilizes the result of the first phase segmentation, $\mathbf{L}$ is the matting Laplacian matrix, and $\{\lambda_c, \lambda_g\}$ balance the weights between the two constraints. In Eq. 13, m_g guarantees that the estimated alpha matte resembles the sharpness of the first phase segmentation. Therefore, the geometric consistency of the boundaries is implicitly preserved. In Eq. 13, the optimal alpha matte, α, is given by computing the smallest eigenvectors of the composite matting matrix.

To recover the true foreground colors using the matte, we use the method proposed in [12], which exploits a smoothness prior [52]. The smoothness prior is used to smoothly generate foreground/background color layers by minimizing the x, y-directional derivatives of the two layers. In the presence of two simple color distributions, we observe that this assumption is particularly correct along the boundary pixels.

VI. Experimental Results

For the quantitative evaluation as shown in Fig. 9, we tested our algorithm with various objects having soft boundaries.

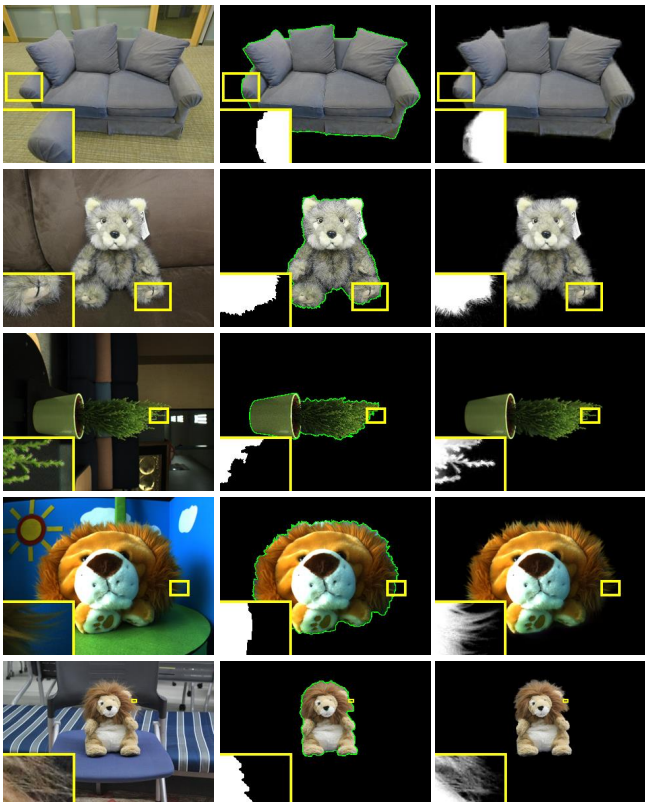


Fig. 9. The dataset used for quantitative comparison with [28], [8], [10]. First column: five of the input images (front views). Second column: our binary segmentation results in the first phase. Third column: our final results after matte refinement.

By comparing our algorithm with previous algorithms, we validated hard segmentations and mattes obtained from the two-phase procedure. Several qualitative results were also obtained to show the ability in handling more challenging examples and to demonstrate the practical uses of our system.

A. Evaluation datasets

We selected Couch and Teddy from a publicly available dataset [8] because they have fractional boundaries. Since few multi-view dataset with fractional boundaries are currently available, we captured images of the target objects. We placed foreground objects, such as Lion2 and Tree on a turntable (SNRT400-Solutionix), and capture images using two cameras (Flea2-PGR 1280x960 res.). With this hardware, the camera poses could be more exactly computed. To validate our algorithm with higher resolution images, we add Lion3, captured by the Canon DSLR Mark3 (5184x3456 res.).

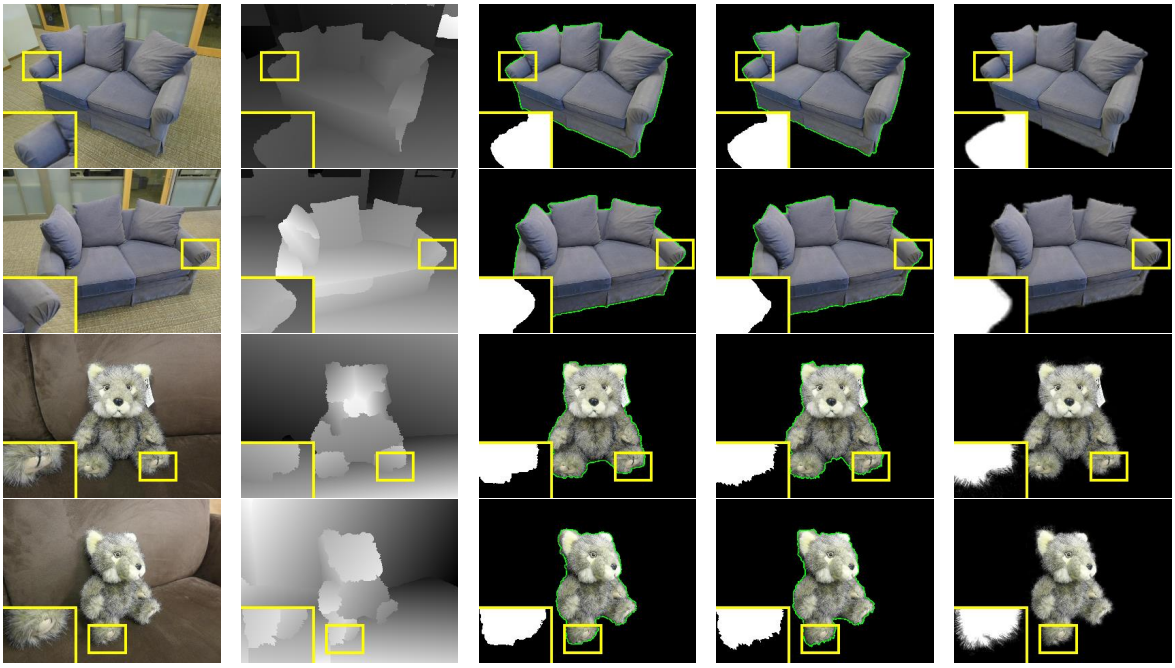


Fig. 10. Comparison with [8] on the Couch and Teddy dataset. First column: Input color images. Second column: Planar reconstruction in [8]. Third column: Segmentation results from [8]. Fourth column: Our results in the first phase. Fifth column: Our final results in the second phase.

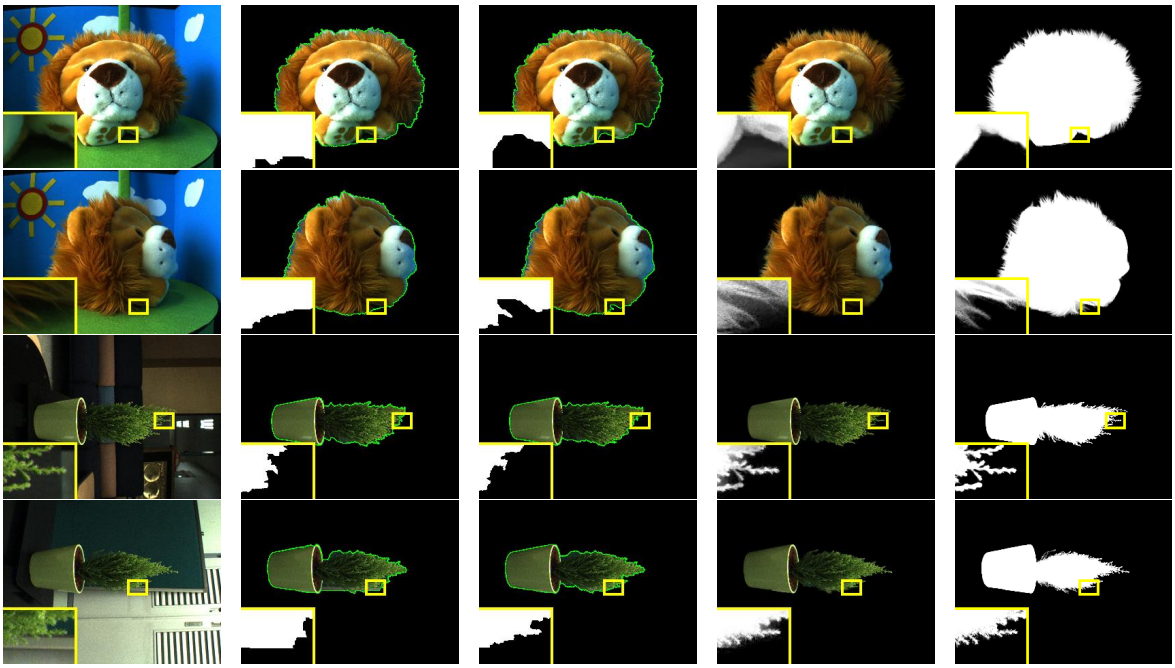


Fig. 11. Comparison with [10] on the Lion2 and Tree dataset. First column: Input images. Second column: Segmentation results from [10]. Third column: Our results in the first phase. Fourth column: Our final results in the second phase. Fifth coulumn: Ground truth (Manual mattes).

For qualitative comparison, we also captured images of large objects, such as a person, by using multiple 20 AW-HE100 Panasonic HDV cameras. In this case, we made use of calibration pattern boards of various sizes to calibrate intrinsic and extrinsic cameras. If the input images were densely captured around an object, we could easily use accessible SfM engines [15] to estimate camera parameters. Our system typically required 8-12 pre-calibrated images to exploit the geometric constraint.

To obtain the ground truth of the fractional boundaries, we manually gave trimap strokes until the final matte was visually flawless. This is because we could not apply simple chroma or difference keying methods for background subtraction from natural scenes. For visually perfect strokes, one input image usually requires 250-500 pin-points. We used Photoshop CS6 with PowerMask plug-in.

B. Comparisons with [8], [10]

For comparison with other approaches, we used the method described in [28] as the baseline algorithm, where only the color model is iteratively updated without any common models. In this quantitative comparison, especially with [8], [10], the proposed method performed the best, as shown in Table II, and we explain this result as follows.

Figure 10 shows two examples comparing our approach with [8]. Our approach differs from [8] in that we do not

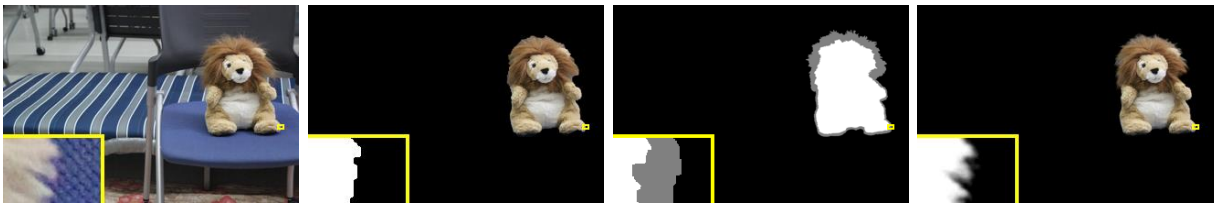


Fig. 12. Handling high-resolution inputs on the Lion3 dataset. First column: One input image (5184x3456). Second column: Rough segmentation at a low resolution (648x432). Third column: Matting region detection and refinement at the original resolution. Fourth column: Our final alpha matte result.

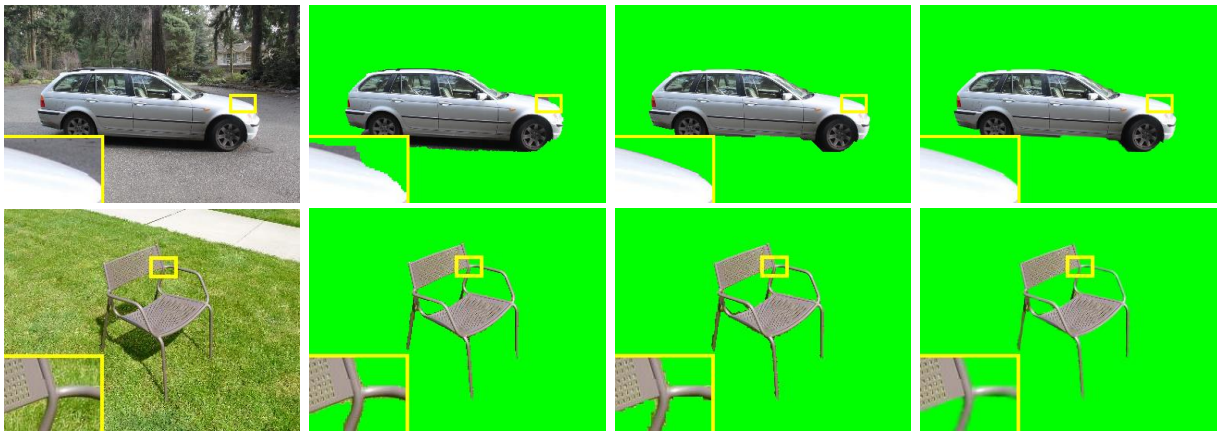


Fig. 13. Qualitative comparision with [8]. First column: Input images. Second column: Segmentation results from [8]. Third column: Our results in the first phase. Fourth column: Our results in the second phase. Note that mixed pixels along the boundaries are resolved after the matting refinement.

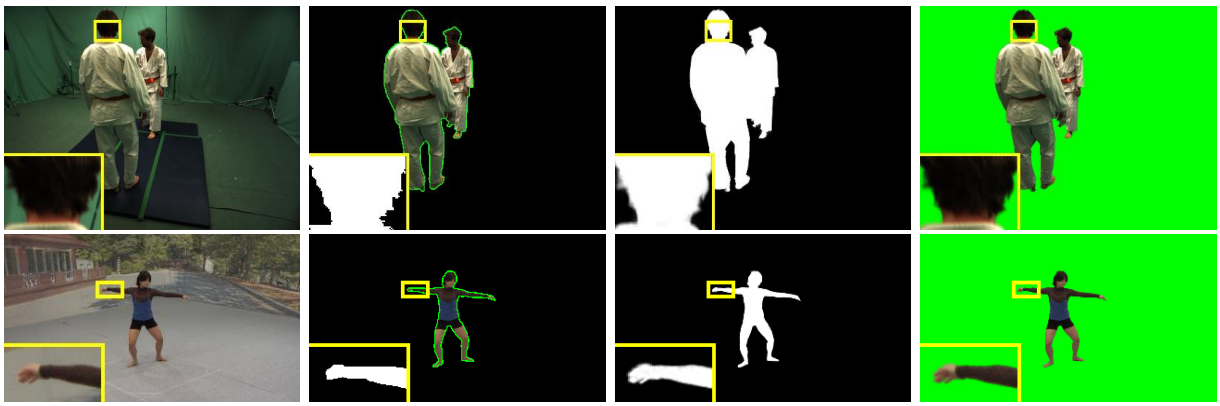


Fig. 14. Multiview mattes for person datasets. First column: Input images. Second column: Our binary segmentations in the first phase. Third column: Our alpha mattes in the second phase. Fourth column: True foreground colors for the extracted foreground mattes.

explicitly rely on the accuracy of 3D reconstruction. Instead, 3D anchor points are involved with the geometric coherency in our approach. When the foreground depth is clearly distinguishable from the background depth, the accurate structure is obviously helpful for [8] as seen in the Couch example of Fig. 10. However, our approach also shows comparable results. In the case of the Teddy example, the foreground depth itself has some ambiguities. While the estimated depths directly affected the segmentaion results in [8], our approach was able to recover the details of Teddy's foot on the ground due to our appearance model.

The effectiveness of our appearance model and comparison with [10] is shown in Fig. 11. Using our own capturing system, we took only eight images around the objects. In this wide-baseline configuration, finding stereo correspondences is challenging. Therefore, we used calibrated camera poses. For some regions in these examples, the background colors are very similar to the foreground colors, such as the shadow of Lion2 and the green plate behind Tree. Hence, purely utilizing color information is not desirable; rather, analyzing the unique patterns of foreground textures provides better results. Compared to [10], which utilizes bag-of-word (BoW) representation, our approach utilize Fisher vector encoding (FV) [55] for the classification scores [24]. The Lion2 and Tree examples shown in Fig. 11 demonstrate that our algorithm outperforms [10] in terms of segmentation quality.

C. Handling high resolution images

When handling high-resolution images, such as the Lion3 example in Fig. 12, we adopt the coarse-to-fine strategy to facilitate the two-stage process. For the computational efficiency of training model parameters, we can take downsampled images with a small amount of accuracy loss. In our second phase, however, there is a resampling procsss along rough boundaries to collect foreground/background pixels at the original image resolution. The final results are locally calculated around the

Methods	Couch	Teddy	Tree	Lion2	Lion3
Grabcut [28]	0.1310	0.1589	0.2910	0.2278	0.1916
ECCV12 [8]	0.0814	0.0944	N/A	N/A	N/A
ICCV13 [10]	0.0829	0.1017	0.1145	0.1062	0.1039
Ours (1st)	0.0846	0.0953	0.0945	0.0987	0.0874
Ours (2nd)	0.0775	0.0782	0.0707	0.0650	0.0711

TABLE II
QUANTITATIVE COMPARISONS WITH STATE-OF-THE-ART APPROACHES. THE ERRORS ARE MEASURED IN RMSE FOR FIVE DIFFERENT DATASETS. NO USER INTERACTION WAS USED TO OBTAIN THESE RESULTS.

Methods	Th-2	Th-6	Th-10	KL div.
CFM [12]	0.0828	0.0745	0.0827	0.0694
KNN [44]	0.0822	0.0754	0.0812	0.0691
LBM [45]	0.0837	0.0770	0.0829	0.0688
WCT [46]	0.0835	0.0781	0.0823	0.0673
Ours	0.0827	0.0722	0.0749	0.0654

TABLE III
QUANTITATIVE COMPARISONS (RMSE) WITH VARIOUS MATTING METHODS UNDER DIFFERENT TRIMAPS. THE ERRORS WERE MEASURED IN RMSE FOR 27 IMAGES.

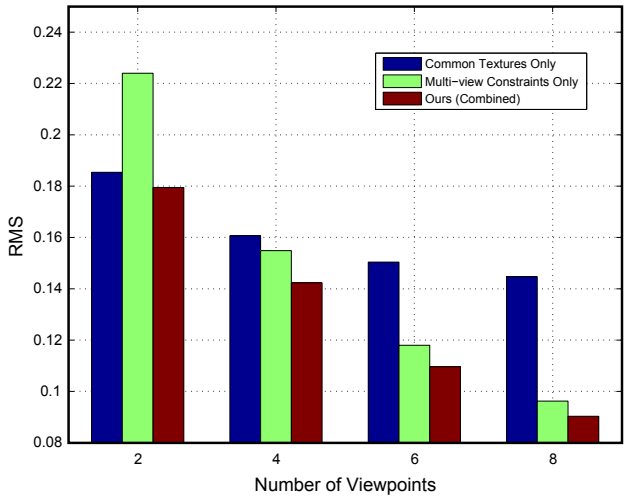


Fig. 15. Self-evaluation of our first phase hard segmentation (Sec. IV) according to the number of images. We only utilized common texture (Sec. IV-C2) or multi-view constraints (Sec. IV-B) for validation. Note that the common textures model is more useful to classify foreground and background patterns when the geometric constraint cannot be meaningfully applied. The geometric constraint becomes a powerful clue when the number of input viewpoints is more than 4-6.

rough segmentation, and a few small errors in the trimap often turn out to be acceptable after matting refinement.

D. Other qualitative results

For more challenging cases, as shown in Fig. 13, our results are comparable to the result of [8]. Our approach attempts to increase the pixel dimension by taking mid-level features in the first phase and removing the mixed color pixels in the second phase. In our experiment, the visual appearance considering the 1st and 2nd order gradient patterns [57] is more discriminative than solely taking color observations.

We also validated our algorithm with large objects, such as a human. Neither any user-assisted initializations nor static background modeling are mandatory for our system, but all the details, such as hair and hands are automatically computed only from eight calibrated color images. In Fig. 14, one example was taken from the INRIA 4D Repository, and the other example was captured by the 20 synchronized cameras. Our results are comparable to those of the state-of-the-art interactive algorithms qualitatively, but we do not require any user markups nor local background models in these examples.

VII. DISCUSSION

In this section, we analyze our approach in terms of three aspects. One is to evaluate the effect of adding mid-level appearance features in the first phase. Another important factor is about the automatic matting refinement in the second phase. The other is user-guided initialization and additional strokes to correct trimaps.

A. Common texture model

The common texture model (Sec. IV-C2) is a comprehensive foreground/background texture prior regardless of viewpoints. In practice, many objects actually exhibit similar texture patterns in both the foreground and background regions. In contrast to [10], which excludes any common appearance models, our experimental results show that having common texture models improves the segmentation accuracy achieved with a wide-baseline capturing setup. To see this effect clearly, we test our first-phase hard segmentation while decreasing the number of input images. The alpha map error was measured in the RMS metric. In this experiment, we used the same bounding volume for the initialization regardless of the number of input images. Fig. 15 shows the evaluation results. Because our common texture model can faithfully distinguish the foreground and background texture, our approach is particularly useful when there are only a few input views. If there are enough images (more than 4 images around the target object), the geometric constraint works as a strong clue in the optimization process.

This study attempted to design a hand-crafted descriptor for a superpixel. By means of Fisher vector encoding, we train a classifier to separate foreground and background samples in the high-dimensional texture feature space. One issue we observed is that the initial foreground/background features of input images are not perfectly correct in the beginning state. Therefore, the influence of the texture energy term over the others in MRF is empirically given. When it comes to the parameters controlling effects of the texture term, we observed that the sensitivity increases if the texture patterns are very weak or similar in both regions. Thses heuristics can be further improved when the classifier is pre-trained in advance.

B. Matting region detection and refinement

The other factor contributing to our results is obviously the matting stage. In the quantitative results for our five multiview datasets, our second phase for matting refinement (Sec. V) reduces the RMS error to 0.015 on average compared to the first phase (Sec. IV).

Not only for the multiview dataset, but any binary masks may also be improved using our second-phase algorithm. For instance, we tested 27 single images from the alpha matting dataset [63] and assumed that their binary segmentations are given. To detect the region where matting was needed, we took a distance transform with respect to the binary edges. Then, we

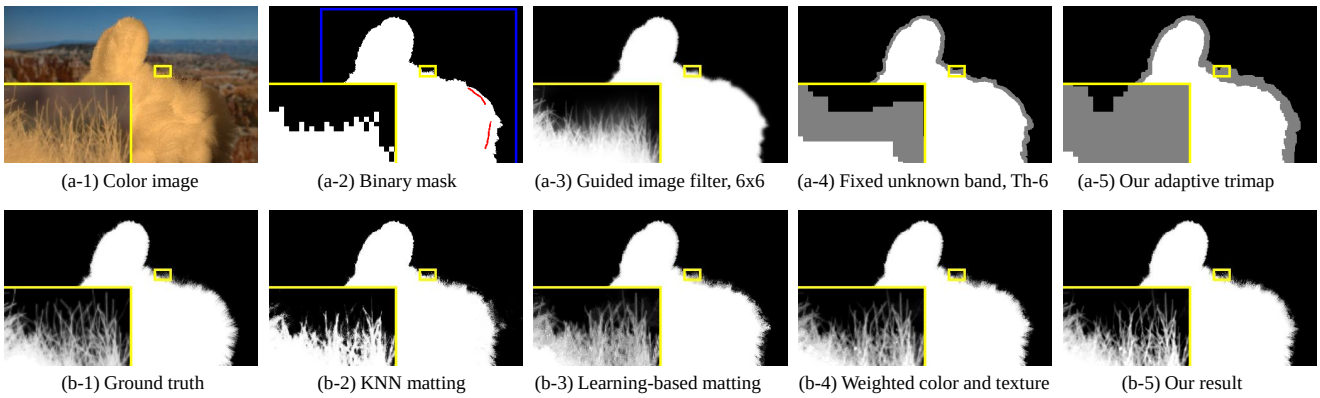


Fig. 16. Evaluating matting given different trimaps. First row: One of 27 single images is shown. Based on a binary mask from its ground truth, [62] was computed by the 6x6 supporting window. The simple trimap was obtained from the distance transform (Th-6) of the binary foreground/background masks. Our adaptive trimap was computed by measuing KL divergences. Second row: The ground truth matte for the input image is shown. The results from [44], [45], [46] were computed based on our adaptive trimap. Our final result is also shown.

applied various thresholds, such as 2,4,..,10 pixel-wide on the distance map to obtain unknown region trimaps with various boundary thicknesses. Given the rough segmentations, we ran our second-phase algorithm for the adaptive matting regions. For computation efficiency, we first downsampled all input images into VGA resolutions, and the estimated trimaps were resized back to the original resolution.

For each trimap, we compared four approaches [12], [44], [45], [46], and ours. The evaluation results of this experiment are shown in Table III. In the quantitative results for the 27 single image datasets, all matting algorithms produced stable results on the 6 pixel-wide unknown regions. Indeed, for such a narrow band, one method in [62] greatly reduced the RMS error in the input segmentation from 0.085 to 0.0714 by using binary guidances with a 6x6 window. An example is shown in Fig. 16. When the initial segmentations are correct, it performs the better than existing matting algorithms, such as [12]. We used these baseline algorithms [12], [62] to test the performance of our algorithm.

One notable aspect is that all the matting algorithms showed the best results when the matting regions were adaptively determined by our approach. If the matting regions are given enough for additional sampling strategies, the WCT [46] sampling approach provides good results in the quantitative comparison. It considers texture compatibility measures to avoid overlapping samples for foreground/background pair sets. As the size of unknown bands increased, we also observed that the assumption of matting Laplacian in [45] worked better than the simple linear model in [12].

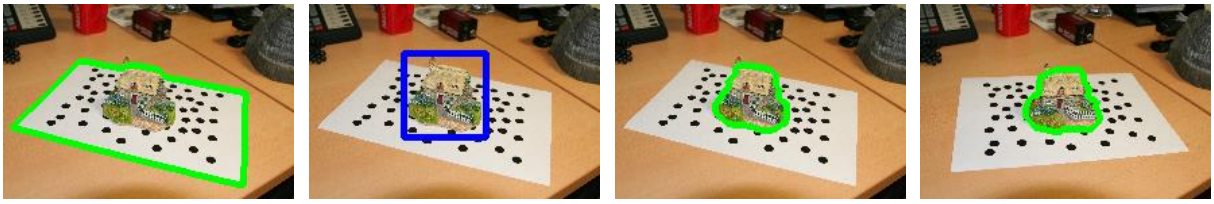
Under the assumption that the foreground/background masks are given, our matting refinement scheme showed the best performance. This is because we first detect the optimal regions where two strong color distributions can be the most easily separated. Not only our algorithm but other matting refinement methods also showed better performance with these contextual KL measurements, resulting in good matting regions. Similar to the method [62], we use a binary guidance as a soft constraint, which effectively suppresses the bleeding effect of output mattes. Our linear model generally holds in simple cases, but we think some cases, especially in the presence of natural illumination and challenging material properties, have fundamental challenges in all processes, such as detection, matting, and estimation of foreground colors.

C. User interaction

Fundamentally, our system defines a foreground as a distinct appearance in the initial bounding volume. We show successful examples in which the iterative optimization overcomes some ambiguities mostly by checking its geometric consistency in the space. Once the foreground definition is loosely initialized, however, the algorithm likely includes some common backgrounds with its wrong 3D hypothesis. To observe this negative effect, we made a 3D cube bounding our initial volume of interest and gradually increased its size. At a size about 15-20% larger than the bounding cube, we confirmed that our system did not initialize well on the target datasets, leading to unsatisfactory results.

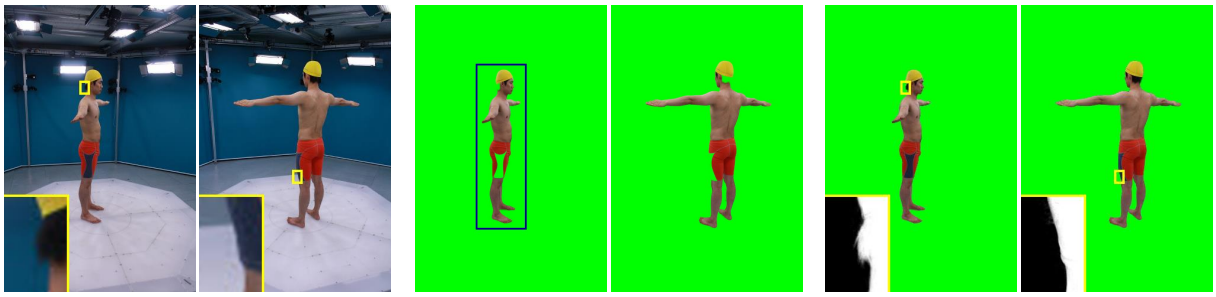
Figure 17 shows a typical failure case of the first phase. Consequently, the second phase does not have the right position of contour points. Another example is also shown in Fig. 18 for this failure case. Some parts of the human body image were completely removed after the iterative optimization in the first phase. This is because the initial appearance models are loosely defined, so we give one box bounding the foreground in one image. Then, the bounding volume of interest is updated according to the modification, eventually leading to the result that all the initial foreground masks are tightly given. Having one user-given box, all the missing parts are recovered with better initialization.

Another example concerns the correction of trimaps with user-given foreground and background strokes. Once we get this type of user-guidance, regardless of what the trimap is, the regions of these stokes become a hard constraint m_c for the corresponding image. We can see the effect of the direct modification to the trimap as in Fig. 19. If many strokes are needed for several images, we can also propagate the accurate information between images. This can be done by assigning big values to the pixel nodes in the first-phase graph model. In MRF optimization, they play the same role in altering foreground and background energies for one viewpoint. However, the difference is that the pixel nodes are geometricallly linked with their common anchor points. Taking



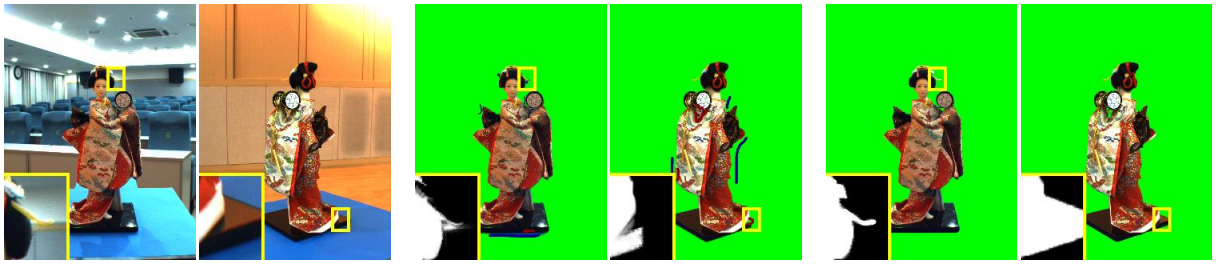
(a) (b) (c) (d)

Fig. 17. User interaction for identifying interested region. (a) The segmented foreground using our approach. (b) A user selects a bounding box in only one view point. (c-d) Our algorithm re-considers the interested bounding volume and refines the results of all the input images.



(a) (b) (c)

Fig. 18. Giving a bounding box for better initialization. (a) The initial bounding volume is loosely defined by the visual hull of sparse cameras. (b) For this reason, some foreground parts are missing. In this case, a user can limit the volume of interest by simply giving one 2D box in only one image. (c) All outputs are updated from the modified volume.



(a) (b) (c)

Fig. 19. User stokes for correcting trimaps. (a) The boundaries are so colorful that user strokes are needed to obtain better trimaps. (b) Note that the 2D strokes in one view are geometrically linked with 3D anchor points and are connected with the visible pixels in the other views as well. (c) All outputs are updated from the modified energy function.

account of these energies in the single graph, we update all binary segmentations $\mathbf{x}$ and check their geometric coherence score $\mathbf{v}$. In practice, we can check whether the additional strokes are needed until the binary segmentation results are visually acceptable. After that, the second phase re-computes new trimaps and gets additional stokes if necessary.

Therefore, definitive foreground and background samples as well as tight bounding boxes can be effectively propagated across all the input viewpoints. Without considering geometric relationships of the information, we may expect that it is a very tedious task to markup intermediate segmentations for each viewpoint over the whole image sequence.

VIII. Conclusion

In this paper, we introduced a framework to extract the soft boundaries of a target object from multi-view images. We utilize coarse 3D reconstruction to define an initial volume bounding the foreground object. Sequentially, we seek geometrically consistent regions having similar appearances across all input images. The Fisher vector encoding adopted in the system allows us to model high-fidelity appearances in images. The consistent regions are cross-validated with one another by referring to their anchor points in space. To detect the optimal matte regions, we optimize the cumulative sum of KL divergences to smoothly take matte regions according to the contexts of object boundaries. Our Laplacian matting equation considers geometrically consistent segmentations in enforcing the multi-view constraint for the final results.

The proposed method was validated using various examples. The results of our method were qualitatively and quantitatively compared with state-of-the-art approaches and the key factors in the algorithm were analyzed in detail. Despite its best performance, a few user interventions might be required for some datasets. We demonstrated that even a few user-given boxes and strokes are effectively shared and propagated across viewpoints for our multi-view segmentation and fractional boundary refinement.

Acknowledgement

This research was supported by the Ministry of Trade, Industry & Energy and the Korea Evaluation Institute of Industrial Technology (KEIT) with the program number of ”10060110”. The first author sincerely appreciates to ETRI (Electronics and Telecommunications Research Institute) for the multiview system of data capturing. We are also grateful to anonymous reviewers for their constructive comments.

REFERENCES

[1] A. Laurentini. The visual hull concept for sihouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(2):150–162, 1994

[2] K. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal on Computer Vision (IJCV)*, 38(3):199–218, 2000

[3] G. Vogiatzis, P.H.S. Torr and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(12):2241–2246, 2007

[4] A. Kowdle, Y.-J. Chang, A. Gallagher, D. Batra and T. Chen. Putting the user in the loop for image-based modeling. *International Journal on Computer Vision (IJCV)*, 108(1):30–48, 2014

[5] J. Park and S.N. Sinha and Y. Matsushita and Y-.W. Tai and I.S. Kweon Multiview photometric stereo using planar mesh parameterization. *Proceedings of International Conference on Computer Vision (ICCV)*, 2013

[6] S.-H. Kim, Y.-W. Tai, Y. Bok, H. Kim and I.-S. Kweon. Two phase approach for multi-view object extraction. *Proceedings of International Conference on Image Processing (ICIP)*, 2011

[7] W. Lee, W. Woo and E. Boyer. Silhouette segmentation in multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(7):1429–1441, 2011

[8] A. Kowdle, S.N. Sinha and R. Szeliski. Multiple view object cosegmentation using appearance and stereo cues. *Proceedings of European Conference on Computer Vision (ECCV)*, 2012

[9] A. Djelouah and J.-S. Franco and E. Boyer and F.L. Clerc and P. Perez. N-tuple color segmentation for multi-view silhouette extraction. *Proceedings of European Conference on Computer Vision (ECCV)*, 2012

[10] A. Djelouah and J.-S. Franco and E. Boyer and F.L. Clerc and P. Perez. Multi-view object segmentation in space and time. *Proceedings of International Conference on Computer Vision (ICCV)*, 2013

[11] J.-Y. Guillemaut and A. Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International Journal on Computer Vision (IJCV)*, 93(1):73–100, 2011

[12] A. Levin and D. Lischinski and Y. Weiss. A closed form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):228–242, 2008

[13] C. Rhemann, C. Rother, A. Rav-Acha and T. Sharp. High resolution matting via interactive trimap segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008

[14] N. Snavely and S. M. Seitz and R. Szeliski Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006

[15] C. Wu. VisualSFM: A visual structure from motion system. *http://ccwu.me/vsfm/*, 2011

[16] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1376, 2010

[17] C. Rother, T. Minka and A. Blake and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006

[18] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L.V. Gool and X. Tang. Transductive object cutout. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008

[19] D.S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. *Proceedings of International Conference on Computer Vision (ICCV)*, 2009

[20] A. Joulin, F. Bach and J. Ponce. Discriminative clustering for image co-segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010

[21] C. Lee and W-.D. Jang and J-.Y. Sim and C-.S. Kim Multiple random walkers and their application to image cosegmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015

[22] F. Wang and Q. Huang and L.J. Guibas Image co-segmentation via consistent functional maps. *Proceedings of International Conference on Computer Vision (ICCV)*, 2013

[23] T. Ma and L.J. Latecki Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

[24] Y. Chai, V. Lempitsky and A. Zisserman. BiCos: A bi-level co-segmentation method for image classification. *Proceedings of International Conference on Computer Vision (ICCV)*, 2011

[25] M. Oquab, I. Laptev, L. Bottou and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014

[26] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *Proceedings of International Conference on Computer Vision (ICCV)*, 2001

[27] Y. Li, J. Sun, C.-K. Tang and H.-Y. Shum. Lazy snapping. *ACM Transactions on Graphics (TOG)*, 23(3):303–308, 2004

[28] C. Rother, V. Kolmogorov and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *Proceedings of ACM SIGGRAPH*, 23(3):309–314, 2004

[29] D. Batra, A. Kowdle, D. Parikh, T. Chen and J. Luo. iCoseg: Interactive co-segmentation with intelligent scribble guidance. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010

[30] A. Kowdle, D. Batra, W. Chen and T. Chen iModel: Interactive co-segmentation for object of interest 3D modeling. *Trends and Topics in Computer Vision (ECCV Workshops)*, 2012

[31] G. Zeng and L. Quan. Silhouette extraction from multiple images of an unknown background. *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2004

[32] A. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal on Computer Vision (IJCV)*, 53(1):31–43, 2003

[33] D. Snow, P. Viola and R. Zabih. Exact voxel occupancy with graph cuts. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000

[34] J.-S. Franco and E. Boyer. Fusion of multiview silhouette cues using a space occupancy grid. *Proceedings of International Conference on Computer Vision (ICCV)*, 2005

[35] M. Sarim, A. Hilton, J.-Y. Guillemaut, T. Takai and H. Kim. Natural image matting for multiple wide-baseline views. *Proceedings of International Conference on Image Processing (ICIP)*, 2010

[36] T. Wang, J. Collomosse and A. Hilton. Wide baseline multi-view video matting using a hybrid markov random field. *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2014

[37] M. Sormann and C. Zach and K. Karner Graph cut based multiple view segmentation for 3D reconstruction. *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, 1085–1092, 2006

[38] N.D.F. Campbell, G. Vogiatzis, C. Hernandez and R. Cipolla. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing (IVC)*, 28(1):14–25, 2010

[39] N.D.F. Campbell, G. Vogiatzis, C. Hernandez and R. Cipolla. Automatic object segmentation from calibrated images. *Conference for Visual Media Production (CVMP)*, 126–137, 2011

[40] K. Kolev and T. Brox and D. Cremers Fast joint estimation of silhouettes and dense 3D geometry from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(3):493–505, 2012

[41] J. Xiao, J. Wang, P. Tan and L. Quan. Joint affinity propagation for multiple view segmentation. *Proceedings of International Conference on Computer Vision (ICCV)*, 2007

[42] C. Reinbacher and M. Rüther and H. Bischof Fast variational multi-view segmentation through backprojection of spatial constraints. *Image and Vision Computing (IVC)*, 30(11):797–807, 2012

[43] Y.Y. Chuang, B. Curless, D.H. Salesin and R. Szeliski. A bayesian approach to digital matting. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001

[44] Q. Chen and D. Li and C-.K. Tang KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(9):2175–2188, 2013

[45] Y. Zheng and C. Kambhamettu Learning based digital matting. *Proceedings of International Conference on Computer Vision (ICCV)*, 889–896, 2009

[46] E.S. Varnousfaderani and D. Rajan Weighted color and texture sample selection for image matting. *IEEE Transactions Image Processing (TIP)*, 22(11):4260–4270, 2013

[47] J. Wang, M. Agrawala and M. Cohen. Soft scissors: an interactive tool for realtime high quality matting. *ACM Transactions on Graphics (TOG)*, 26(3), 2007

[48] N. Joshi, W. Matusik and S. Avidan. Natural video matting using camera arrays. *ACM Transactions on Graphics (TOG)*, 25(3):779–786, 2006

[49] M-.H. Hyun and S-.Y. Kim and Y-.S. Ho. Multi-view image matting and compositing using trimap sharing for natural 3-D scene generation. *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, 397–400, 2008

[50] D. Cho, S. Kim and Y.-W. Tai. Consistent matting for light field images. *Proceedings of European Conference on Computer Vision (ECCV)*, 2014

[51] S.W. Hasinoff, S.B. Kang and R. Szeliski. Boundary matting for view synthesis. *Computer Vision and Image Understanding (CVIU)*, 103(1):22–32, 2006

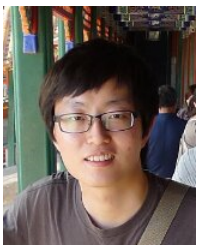
[52] A.R. Smith and J.F. Blinn. Blue screen matting. *Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, 1996

[53] Y.-W. Tai and J. Jia. Local color transfer via probabilistic segmentation by expectation-maximization. *Proceedings of IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), 2005
[54] Y.-W. Tai, J. Jia and C.-K. Tang. Soft color segmentation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(9):1520–1537, 2007
[55] F. Perronnin, J. Sanchez and T. Mensink. Improving the fisher kernel for large-scale image classification. *Proceedings of European Conference on Computer Vision (ECCV)*, 2010
[56] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. *Proceedings of the 18th ACM International Conference on Multimedia*, 2010
[57] J. Shotton, J. Winn, C. Rother and A. Criminisi. TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal on Computer Vision (IJCV)*, 81(1):2–23, 2009
[58] I. Choi, M. Lee and Y.-W. Tai. Video matting using multi-frame nonlocal matting Laplacain. *Proceedings of European Conference on Computer Vision (ECCV)*, 2012
[59] C. Li and C. Xu and C. Gui and M. D. Fox Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions Image Processing (TIP)*, 19(12):3243–3254, 2010
[60] Y. Boykov and O. Veksler and R. Zabih Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001
[61] O. Juan and R. Keriven. Trimap segmentation for fast and user-friendly alpha matting. *Variational, Geometric, and Level Set Methods in Computer Vision (VLSM)*, 2005
[62] K. He, J. Sun and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(6):1397–1409, 2013
[63] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli and P. Rott. A perceptually motivated online benchmark for image matting. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009

Seong-heum Kim received his Bachelor degree in Electrical Engineering from Yonsei University in 2007 and his Master degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2010. From Sept. 2011 to Sept. 2014, he worked for LG Electronics. He is currently pursuing his PhD degree in KAIST. His research interests include segmentation, object pose estimation and computer graphics.

Yu-Wing Tai received the B.Eng. (Hons.) and M.Phil. degrees in computer science from The Hong Kong University of Science and Technology, in 2003 and 2005, respectively, and the Ph.D. degree in computer science from the National University of Singapore, in 2009. From 2007 to 2008, he was a full-time Student Internship with Microsoft Research Asia. He was an Associate Professor with the Korea Advanced Institute of Science and Technology (KAIST), from 2009 to 2015. He received the Microsoft Research Asia Fellowship, in 2007, and the KAIST 40th Anniversary Academic Award for Excellent Professor in 2011. He has served as an Area Chair of ICCV 2011 and ICCV 2015. He is currently a Principal Research Scientist with SenseTime Group Ltd, Hong Kong. His research interests include computer vision and image processing.



Jaesik Park received his Bachelor degree (Summa cum laude) in media communication engineering from Hanyang University in 2009. He received his Master and Ph.D. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2011 and 2015, respectively. He received the Microsoft Research Asia Fellowship in 2011. From April 2012 to Oct. 2012, he worked as a full time internship in Microsoft Research Asia (MSRA). He is currently a research scientist of Intel visual computing lab., CA, United States of America. His research interests include depth map refinement, rigid/non-rigid 3D reconstruction. He is a member of the IEEE.

In So Kweon received the BS and MS degrees in mechanical design and production engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the PhD degree in robotics from the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1990. He worked for the Toshiba R&D Center, Japan, and joined the Department of Automation and Design Engineering, KAIST, Seoul, Korea, in 1992, where he is now a professor with the Department of Electrical Engineering. He is a recipient of the best student paper runner-up award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09). His research interests are in camera and 3D sensor fusion, color modeling and analysis, visual tracking, and visual SLAM. He was the program co-chair for the Asian Conference on Computer Vision (ACCV '07) and was the general chair for the ACCV '12. He is also on the editorial board of the International Journal of Computer Vision. He is a member of the IEEE and the KROS.