Multi-View Object Extraction

Monica srilakshmi
M.Tech(DECS)-II
Sri Padmavati Mahila Visvavidyalayam,
Tirupati, India

Leela santhoshi
Electronics and communication engineering
Sri Padmavati Mahila Visvavidyalayam
Tirupati,Indi

Abstract: To extract a multi-view foreground object which is bounded by the convex volume of interest defined by the overlapping space of camera viewing frustums. Identifying the target object across different images that share a same geometric representation in space and texture model from background for multi-view binary co-segmentation and detecting the color ambiguous regions along the object boundary for matting refinement. Our matting region detection is based on kullback-leibler divergence of local color distributions of different pixel bands. Our results are high quality alpha mattes consistent across all different viewpoints.

Key words: Co-segmentation, natural image matting

I.LITERATURE REVIEW

The goal of multi-view object extraction is to simultaneously segment the foreground object from multiple images, each captured at different viewpoints of the target object.

In recent years, configurations including multiple cameras have gained interest and many researchworks have targeted reconstruction, tracking, motion analysis and actionrecognition. For all these applications, the identification and extraction of foreground objectsis a key element.

In monocular scenarios, the object to segment is not clearly defined and themethods mostly rely on a user interaction so these methods are generally limited to well controlled environments in multi-camera setups of monocular segmentation.Many works since followed this first attempt tosolve the problem, with the common objective of expressing segmentation constraints from the multi-view geometry. Zeng proposed a method that propagates color consistency between viewpoints by iteratively carving the visual hull with respect to color consistency in each image ,however it only approximates spatial coherence which should be enforced over all viewpoints simultaneously. 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 constrains across viewpoints. Previously for segmenting the foreground object we utilized the technique of co-segmentation algorithm but due to user interactions such as bounding boxes and foreground scribbles for each image at every iteration these systems mostly require user corrections. However, when some of them share a common 3D space among input images. These do not consider any geometric relations. Extending this work, to enforce geometric constrains on the multi view-images, Rother et al proposed an idea based on strong assumption of color similarity that is the presence of same object in different images for segmenting a task, but this category intrinsically restrict to appearance-based cues. In the next work

Yezzi et al used a level set method for evolving 2D contours consistent with the 3D space which is under strict assumption that a scene is composed of several homogeneous backgrounds and strong irradiance discontinuities. He proposed an algorithm to reconstruct solid shape and radiance from a number of calibrated views of scene with smooth shape and radiance or homogeneous fine texture but it is having a disadvantage that it is easily resistant to noise and not work in the presence of strong texture and boundaries. After yezzi et al, Snow et al proposed an approach based on geometric constraint which is applied to their probabilistic functions so that static background models could be successfully merged in to 3D representation but the assumption of clean background does not hold in real world. Due to the errors in previous approaches campbell et al assumed that the target object is fixed at the center of visual hull and they collected definite foreground samples near the camera centers in images but the model is not able to manage when the object is loosely bounded in images. Recently for better segmentation results they come up with the technique of MRF optimization and also uses graph-cut based multi-view co-segmentation algorithms. In MRF optimization technique number of iterations are performed to check whether the current binary labels are also correct from other view points. In contrast to previous approaches at present researchers are utilizing the concept of Structure from motion pipeline instead of using dense geometry acquisition and finds the initial space of interest that is fully visible to all input cameras.

Matting:

Matting is a process of segmenting a foreground object with its fractional boundaries. To specify foreground, background and uncertainty regions there are some approaches dealt sample based, affinity based, single image matting, narrow baseline image matting and wide base line image matting. In single image and narrow-baseline image matting these approaches share a 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 straight forward. For wide-baseline image matting ,Sarimet al partially applied the epipolar line constraint to isolate the shadow regions in pixel spaces and performed matting as post processing.

II.INTRODUCTION:

In this paper, we present some reliable solutions based on the previous approaches which utilizes the features of bounding-volume prior from camera poses, appearance models under geometric constraints and iterative Markov random field optimization. So, we present a multi-view matte estimation method on the top of previous approaches which not only estimates the binary masks but also soft alpha mattes of a foreground object.

In contrast to previous paper of the two phase approach for multi-view object extraction, in this we propose a method to estimate the unknown regions more accurately and also improve the quality estimated alpha mattes by providing color and texture information along with the geometric constraints for more accurate foreground estimation and providing a dynamic approach for estimating unknown regions.

Here, we evaluate our algorithm to claim the advantages of automatically estimating a tight bound of foreground object by the convex hull of visible SFM points. In addition it is based on simple geometric representation that defines 3D reference points in space and links between super pixels 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. Secondly our appearance model of the foreground object is more robust because it can seek the texture patterns of an object so that our MRF optimization overcomes possible local minimum. It can also handle high resolutions images with minimum user interventions because the optimization and matting procedure only performed on uncertainty regions of the trimaps.

The outlook for the process is given below: Initially the user defines which is the foreground and background regions assuming the target object is bounded by the convex hull of camera viewing frustums and develop an initial contour for the foreground region. Based on the initial contour, estimate the appearance models of foreground regions which is comprised of color and texture model. We define MRF optimization for getting binary segmentations and refine those contours so that they are accurately located at the object boundaries. To evaluate the color mixing of the foreground and background regions along with the unknown regions we utilize the process of evaluating the distribution of colors within the local pixel bands. Lastly by using the matting laplacian we solve for the fractional boundaries of the target foreground object. The local pixel band with largest entropy is selected for matting refinement ,subject to multi-view consistent constraint.

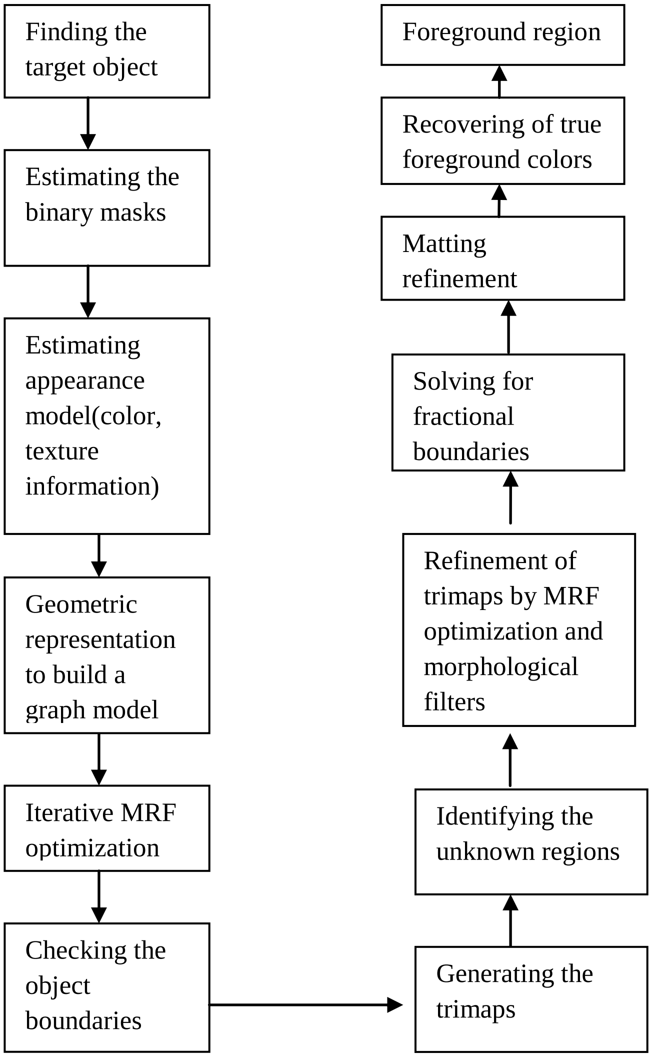

Finding the target object
Estimating the binary masks
Estimating appearance model(color, texture information)
Geometric representation to build a graph model
Iterative MRF optimization
Checking the object boundaries
Generating the trimaps
Identifying the unknown regions
Refinement of trimaps by MRF optimization and morphological filters
Solving for fractional boundaries
Matting refinement
Recovering of true foreground colors
Foreground region

Figure1: Flow of the process

II.PROPOSED METHOD

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. Figure 2 shows the pipeline.

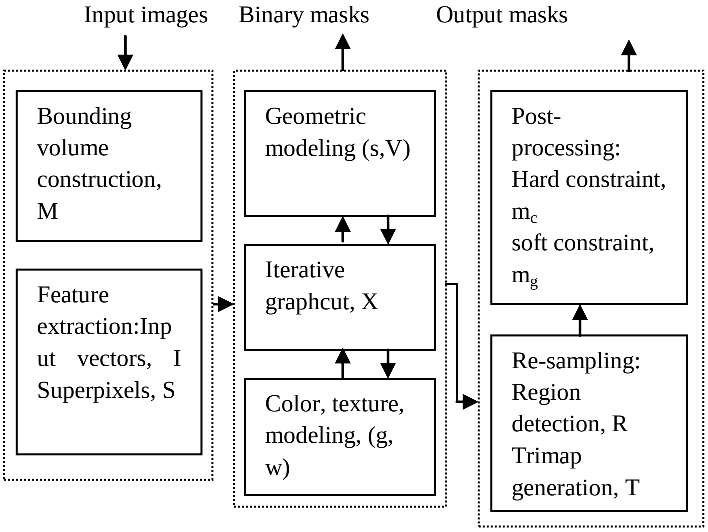

Input images
Binary masks
Output masks
Bounding volume construction, M
Feature extraction:Input vectors, I Superpixels, S
Geometric modeling (s,V)
Iterative graphcut, X
Color, texture, modeling, (g, w)
Post-processing: Hard constraint, m_c soft constraint, m_g
Re-sampling: Region detection, R Trimap generation, T

Figure2. System overview

Assuming that the target object is located inside the convex space of camera viewing frustums for the initialization of the foreground mask. So the first step is to estimate the binary masks X={X^1,X^2,...X^N} of the target object in multi-view images I={I^1,I^2...I^N}, where N is number of input images. Let the initial mask be M={M^1,M^2....M^N}, for the projections of the bounding volume. After gettingan initial mask, we perform the superpixel segmentation in which each superpixel consists of color and gradient components and denote the superpixel sets as S={S^1,S^2...S^N}. We utilize iterative graph-cut optimization to achieve our goal of estimating the binary masks.

Based on the initial mask M, we build MRF formulation utilizing the concept of appearance models and considering the geometric information.

The binary segmentation in the first phase is formulated as single energy function in the MRF frame work. The appearance model consists of color GMM's and a SVM classifier of foreground and background regions. Our objective function consists of the data term $E_{d,}$where the it is defined as the likelihood similarity using the color and the geometric consistency term, which is designed based on the appearance models E_a and a geometric modelE_g.The neighborhood term $E_n$are weighted by$\lambda_{ng}$for geometrically linking nodes across the related viewpoints,λ_{nc} for considering color and texture linkages in each image. In MRF frame work we consider the term E_{nc} for similar colors, textures and E_{ng} for geometric linkages in designing of the neighborhood term.

$$E_d = \rho \cdot E_a + (1-\rho)\cdot E_g$$

$$E_n = \lambda_{nc}\cdot E_{nc} + \lambda_{ng}\cdot E_{ng}$$

The parameter ρ determines which data term is more reliable for the energy assigning of the node. When both foreground and background models of the pixel or superpixel have similar metrics then a more weight is given to the geometric consistency term.

Geometric representation:

In our approach the geometric coherence of the binary segmentations us evaluated for every MRF iteration. The geometric coherence can be effectively made by using camera projection matrix for correspondence point in the foreground image become true foreground when warped pixel position in other images also belongs to the foreground regions. In this approach for the geometric coherence we define the term coherence score for a superpixels which is normalized to have a range of [0 1]. The geometric information can be gathered by considering the regularly distributed 3D points in the convex spaceare used to connect superpixels across the input images in the foreground masks for the purpose of score evaluation and named these uniformly distributed 3D points as anchor points P that has the occupancy probability v indicating the coherency of superpixels. Here each anchor point becomes the auxiliary node and superpixels as nodes and build a single graph model. We perform a visibility check for the anchor points to remove the occluded samples. The graph labeling problem is iteratively solved by MRF optimization. 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.The coherence score of the pixel(v_k^n) and superpixel(V_k^n) are defined as the sum of the binary labels x or X :

$$v_k^n = \frac{1}{N}\sum_{n'=1}^{N} x^{n'}_{w_k^{n\to n'}},\ V_k^n = \frac{1}{N}\sum_{n'=1}^{N} X^{n'}_{w_k^{n\to n'}}$$

The geometric energy term of the background and foreground are modeled by using a sigmoid function and is given by

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

λ_v- controls the shape of the energy functions

V_{th} - used for the tolerance of thin foreground segments with possible calibration errors.

The energy connected to the background is bounded e_{nb}=10 as its maximum value

The neighborhood terms can be considered using inter-view links between super pixel notes according to 3D anchors. the neighboring term in the constructed graph is solved by considering all super pixel related to one another as sharing the common geometric model. Our energy term for E_{ng} is given by:

$$\mathrm{E}_{ng}\left(X_p^i, X_q^j\right) = \sum\nolimits_{i,i,p,q} |X_p^i - X_q^j| \cdot en_b$$

Appearance models:

To take color and texture patterns in super pixel in to account we use the appearance model. The energy term in this model consist of Gaussian mixture model and fisher kernel representation .Mixture models are type density model which comprise a number of component functions usually Gaussian. These component functions are combined to provide a multi modal density . They can be employed to model colors of an object in order to perform tasks such as real time color based tracking and segmentation. To perform the task we build color GMM of per-pixel distributions for each image and classify both foreground and background color distribution. To control the influence of texture term and color term we use the parameter λ_t. The appearance energy term is given by :

$$E_a = E_c + \lambda_t \cdot E_t$$

E_t- it is the fisher kernel representation of each super pixel

Color GMM:

From the output of binary segmentation we collect some samples to represent them in color GMM model, using lab color space in the ranging between 0 and 1 which gives more weight to the chromatic channels and less weight to luminance channel to reduce the effects of shadows and also RGB color space. Noise removing and noise reduction of pictures are also are important in classical image problems. To remove image noises and to avoid over-fitting of color distribution we apply Gaussian smoothing to each RGB channel before building the GMMs. 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:

$$\Pr_c\left(\mathrm{I}_k \middle| G_f\right) = \sum_{c=1}^{c_{max}} w_f^c \cdot N(\mathrm{I}_k|\mu_f^c, \sigma_f^c)$$

$$\Pr_c(\mathrm{I}_k|G_b) = \sum_{c=1}^{c_{max}} w_b^c \cdot N(\mathrm{I}_k|\mu_b^c, \sigma_b^c)$$

where$w_f^c \cdot N(\mathrm{I}_k|\mu_f^c, \sigma_f^c)$indicates the weight Gaussian component having mean and variance on the foreground label. previously for each GMM of the current foreground and the background, we use 20 to 30 Gaussian function to model the distribution. when we measure the distance of pixel between the foreground and the background, instead of using all Gaussian functions N, we use the 5 nearest Gaussian functions by defining w_f^c, w_b^c as the minimum metric closer to mahalanobis disance. For suppose if we take the some part of the target object the conventional GMM will disregard some details of the object because if having the fewer samples. In our project we take all details of the object because other false alarms can be effectively suppressed by iterative optimization. The color consistency term are normalized to satisfy $\Pr_c\left(\mathrm{I}_k \middle| G_f\right) + \Pr_c(\mathrm{I}_k|G_b) = 1$for the proper energy terms. The average of these energies in a group of pixels is assigned to their super pixel node in MRF. To convert an arg max MAP problem to an arg min energy minimization we take an negative log of probability in modeling of the color term. So when modeling of the color GMM model the data term is updated iteratively according to binary segmentation results of previous iteration. In the data term, When the color of the pixel has similar distance to both the foreground and background GMM, we give less weight to color consistency term and vice versa.

The fisher kernel in the appearance energy term is a function that measures the similarity of two objects on the basis of sets of measurements for each object and a statistical model. In a classification procedure, the class for a new object (whose real class is unknown) can be estimated by minimizing, across classes, an average of the Fisher kernel distance from the new object to each known member of the given class. It combines the advantages of generative statistical model like Markov field and those of discriminative methods like support vector machines. The fisher kernel can be applied to image representation for classification problems.Currently, the most popular bag-of-visual-words representation suffers from high dimensionality. The Fisher kernel can result in a compact and dense representation, which is more desirable for image classification and retrievalproblems.The Fisher Vector (FV), a special, approximate, and improved case of the general Fisher kernel, is an image representation obtained by pooling local image features. The FV encoding stores the mean and the covariance deviation vectors per component k of the Gaussian-Mixture-Model (GMM) and each element of the local feature descriptors together. In a systematic comparison, FV outperformed all ,Kernel Codebook encoding etc., showing that the encoding of second order information indeed benefits classification performance

Texture:

Our key idea for encoding texture informationis building common texturethat is comprehensive foreground/background texture prior regardless of the viewpoints.To encode texture information, we take the luminance channellin the labspace, and compute the x, y, xy, and yxdirectionalderivatives of Gaussians at two different sigma scales, andLaplacian of Gaussians at three sigma scales. Then those11-dimensional vectors in one viewpoint are added to theoriginal 9-dimensional vectors of color components. The new20-dimensional vectors for each pixel in all input images areclustered to create 64 GMMs, followed by description of allsuperpixels to Fisher vectors with respect to the global GMMs.

In this manner, we get a descriptor $S_k$for superpixel *k*. Afterthe normalization of superpixel descriptors as in, we traina linear SVM either using all positive and negative vectors and their labels $X'_k = 2(X_k - \frac{1}{2})$ across view points, or build SVMs by having the samples only in the respective view. The scores from the trained w give the texture driven- energies E_t to each superpixel:

$$\sum_{k=1}^{k} l(X'_k \cdot w^T S_k) + \frac{1}{2} \parallel w \parallel^2$$

When using appearance-based methods, we usually represent an image of size n·m pixel by a vector in an (n m) dimensional space. In practice, however these (n m) dimensional space are too large to allow robust and fast object recognition. A common way to attempt to resolve this problem is to use dimensionality reduction techniques. While building the initial graph with neighborhood connections the superpixels descriptors are represented by S_k. In the implementations, each superpixel is connected to eight adjacent nodes and is non-locally linked at most with eight similar descriptors in the cost space.Previously we used the standard PCA projections for these reduction technique but due to some limitations we switch to use the techniques of LDA analysis. Both linear discriminant analysis(LDA) and principle component analysis(PCA) are linear transformation techniques that are commonly used for dimensionality reduction. PCA can be described as an unsupervised algorithm, since it ignores class labels and its goal is to find the directions(the so called principle component) that maximize the variance in the data set. In contrast to PCA, LDA is an supervised and computes the direction(linear discriminants) that will represent the axis that maximize the separation between multiple classes. Although it might sound intuitive that LDA is superior to PCA for a multi-class classification task where the class labels are known, this might not always the case. For greater computational efficiency, mean pooling on color vectors is another good approximate option. The pair wise term is implemented by Potts model in superpixel segmentation context, where node is connected to the K nearest neighbors in the X^2 metric between the superpixel descriptors:

$$E_{nt}(X_p, X_q) = \sum_{p,q} |X_p - X_q| \cdot exp(-\beta[\chi^2(S'_p, S'_q)]^2)$$

By using the parameter β we can normalize the distance function. So this texture model is developed in a way that the gradient magnitudes do not lose the directional information and so helpful when we have a similar texture parts in foregrounds with blurry backgrounds.

Energy minimization:

In practice labeling all pixels at the original image resolution is a time consuming task. So instead of labeling all the pixels we take one or two additional coarse resolutions for the rough superpixel segmentation before we start the pixel-level optimization using the initial energy. The weight of theλ_{ng}can be increased during the iterative refinement of the multi-scale segmentations because to make 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 the eight connected grids and add view-view linkages coming from the final 3D surface structure. The neighborhood term for the two adjacent nodesx_p and x_q using a color energy E_{nc} and geometric energy E_{ng} as:

$$E_{nc}(x_p, x_q) = \sum_{p,q} |x_p - x_q| \cdot exp[-\beta(\mathrm{I}_p - \mathrm{I}_q)^2]$$

$$E_{ng}(x_p^i, x_q^j) = \sum_{i,j,p,q} |x_p^i - x_q^j| \cdot en_b$$

In our experiment,most cases required less than three iterations to reach stablelabeling results. When we could not solve the MRF with morethan eight inputs at a high resolution due to memory limitation,we divided them into eight-viewpoint subsets to be respectivelyoptimized after re-computation of the visibility maps.However, we also found that the 3D structures derived fromeight views are good enough to project to the other calibratedviews. Hence, we were able to perform per-view graph-cutsusing its appearance, geometric models with λng= 0. It isstill useful to have such inter-view linkages coming from

3D sample or SIFT-based correspondence in inferringcorrect labels.

Multi-view matte estimation:

Taking the output of the MRF formulation which have estimated binary labels x and the score maps v,the next step is to generate trimaps T and the estimated alpha matte α of the foreground object.

Supporting region detection:

To measure the amount of local color spread in the foreground region for the target object we use the KL divergence to measure it. The method used for the measurement of color spread for matte region detection is two-color line model applied for any local color distribution. The effects of color mixing can be approximated by a linear combination oftwo different colors. If the color samples are concentrated in the middle of the line model and two end at points of the line model then we can judge mixing effect of two colors is strong in the middle and has a sharp at boundary between two regions at the two end points. KL divergence is measured with various window sizes, ranging from 7× 7 to 81 × 81pixels, in three possible shapes (square, thin, thick) having 2,2 and 9 types of offsets respectively.

By increasing the window size, it reduces the value of KL divergence for a region with a sharp boundary. 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 the selecting the window shape, sizes $s_k$with maximum KL divergence of the local color distribution.

The equation for measuring the KL divergence is given by:

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

$r_k$is uniformly sampled seed point along the boundaries

C_0 andC_1 are the RGB colors of the two end points on the line

$d(\cdot,\cdot)$ - is distance operator

By providing the optimum size and shape of the detecting window s_k^* at the best contour point r_k^*, which is a better localized position of r_k, we increase the local window size to effectively separate the foreground and background regions after the evaluation of KL divergence with various window shapes. Specifically, at the positions r_k^*, the window size increases in 13 different directions and to know the whether the control point is a better position of r_k we check it using the maximum response of Gaussian filter difference.

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 the dynamic programming to maximize the sum of KL divergence along all object boundaries.

Trimap optimization:

In the previous section, we described the detection of the optimal local windows of the trimap. We further refine the trimap by MRF optimization which is done by α-expansion.All pixels in the estimated regions are assigned to a new labels $\mathrm{T}_k \, \epsilon \, \{F, B, U\}$ before all the windows are combined with the remaining labels x to constrain the whole image. Each α-expansion iteration can be identically performed by a series of single graph-cuts similar in first phase with slight differences. The MRF equation for trimap optimization is defined as:

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

For modeling geometric terms , three values are taken for energy functions, with$\lambda_v = 0.9$for $x'_k \epsilon F$, 0.7 for $x'_k \epsilon U$and 0.5 for $x'_k \epsilon B$, 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 first phase is that the locality of color samples is given by normalized x, y coordinate information. To find the pixels of the uncertainty areas we use the central colors of local line model and build the third GMM to infer the label U. The combined trimap after the MRF iteration is further refined with simple morphology filters to enlarge the unknown regions. 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.

Matting refinement:

To get accurate alpha matte and true foreground colors after the estimated trimaps and foreground masks, we perform matting refinement and to solve for fractional boundaries only within the uncertainty regions. Our matting framework is based on standard laplacian matting with additional constraint:

$$arg\ min\ \alpha^T L\alpha + \lambda_c(\alpha - m_c)^T W(\alpha - m_c) \\ + \lambda_g(\alpha - m_g)^T(\alpha - m_g)$$

$m_c$is the trimap label as hard constraint.

W- diagonal matrix

W=1 pixel belongs to mixed pixels

W=0 otherwise

$m_g = x^n$ soft constraint that guarantees the estimated alpha matte resembles the sharpness of first phase segmentation

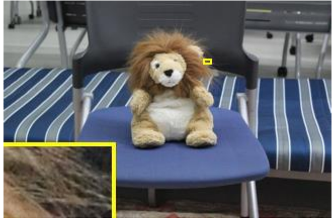
L matting laplacian matrix

$\{\lambda_c , \lambda_g\}$balance the weights between two constraints

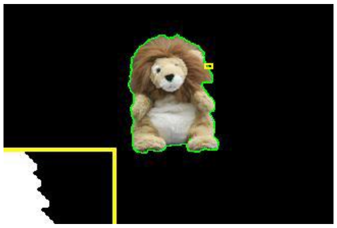
α is given by computing the smallest eigenvectors of the composite matting matrix.

To recover the true foreground colors using the matte, we use the method of closed form solution to natural image matting. We derive the cost function from local smoothness assumptions on foreground and background colors and show that in resulting expression it is possible to analytically eliminate the foreground and background colors to obtain quadratic cost function of alpha. This allows us to find globally optimal alpha matte by solving a sparse linear systems of equations which of simple exploits a smoothness prior that is used to smoothly generate foreground/background color layers by minimizing the x,y- directional derivates of the two layers. In the presence of two simple color distributions we observe that this assumption is particularly correct along the boundary pixels.

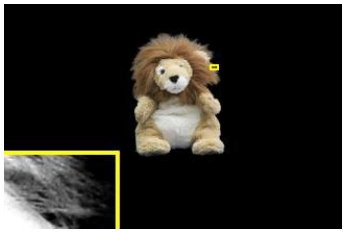
The results are shown in figure 3



input image



Our binary segmentation results in first phase



Our final result after matting refinement

III. CONCLUSION:

In this paper we introduced a frame work to extract the soft boundaries of the 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.

References:

(1). 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.

(2). A. Kowdle, S.N. Sinha and R. Szeliski. Multiple view object co-segmentation using appearance and stereo cues. Proceedings of EuropeanConference on Computer Vision (ECCV), 2012.

(3). S.-H. Kim, Y.-W.Tai, Y. Bok, H. Kim and I.-S.Kweon. Two phase approach for multi-view object extraction. Proceedings of InternationalConference on Image Processing (ICIP), 2011.

(4). C. Rother, V. Kolmogorov and A. Blake. Grabcut: Interactive foregroundextraction using iterated graph cuts. Proceedings of ACM SIGGRAPH,23(3):309–314, 2004

(5). F. Perronnin, J. Sanchez and T. Mensink. Improving the fisher kernel for large-scale image classification.Proceedings of European Conferenceon Computer Vision (ECCV), 2010

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