SCRIBBLE BASED INTERACTIVE 3D RECONSTRUCTION VIA SCENE CO-SEGMENTATION

Adarsh Kowdle[1]*, Yao-Jen Chang*[1]*, Dhruv Batra*[2]*, Tsuhan Chen*[1]

[1]Cornell University, NY, USA. [2]Toyota Technological Institute, Chicago, USA.

ABSTRACT

In this paper, we present a novel interactive 3D reconstruction algorithm which renders a planar reconstruction of the scene. We consider a scenario where the user has taken a few images of a scene from multiple poses. The goal is to obtain a dense and visually pleasing reconstruction of the scene, including non-planar objects. Using simple user interactions in the form of scribbles indicating the surfaces in the scene, we develop an idea of 3D scribbles to propagate scene geometry across multiple views and perform co-segmentation of all the images into the different surfaces and non-planar objects in the scene. We show that this allows us to render a complete and pleasing reconstruction of the scene along with a volumetric rendering of the non-planar objects. We demonstrate the effectiveness of our algorithm on both outdoor and indoor scenes including the ability to handle featureless surfaces.

Index Terms— image based modeling, interactive 3D reconstruction

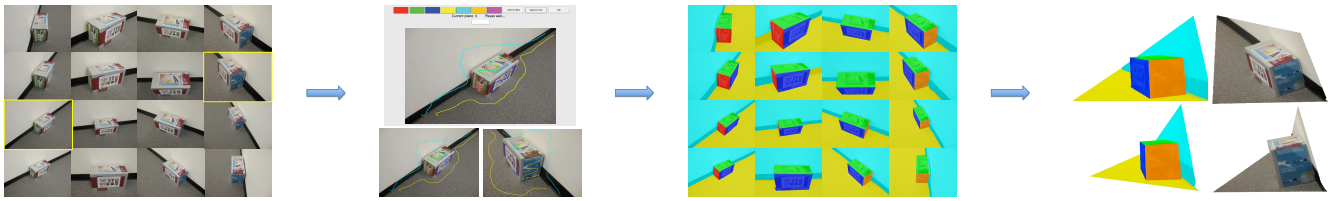
1. INTRODUCTION

We consider a scenario where the user has taken a few images of a scene from varied poses. The goal is to obtain a visually pleasing reconstruction of the scene. Automatic algorithms such as [1, 2, 3], etc have been shown to work well with a large collection of images. When the number of input images is restricted, these automatic algorithms fail to produce a dense plausible reconstruction. There are a number of multiview stereo algorithms which try to obtain a dense depth map for the scene from a set of images [4]. However, multiview stereo algorithms are known to be slow and with a small set of images the reconstruction is usually incomplete, leaving holes on textureless surfaces and specular reflections. In order to improve the reconstruction, some algorithms make planar approximations to the scene [5, 6]. This allows for more visually pleasing reconstructions. However, these algorithms use features such as strong edges and lines which may be absent in textureless surfaces or non-planar objects (like walls, trees, people, etc). This has led to interactive algorithms.

Prior interactive reconstruction algorithms, require involved user-interactions ranging from providing feature correspondence, to marking edges, plane boundaries and detailed line models of the scene [7, 8, 9]. In this paper, we present a novel scribble based interactive 3D reconstruction algorithm where we relax the interactions to mere *scribbles* and render a planar reconstruction of the scene. In a typical scene, non-planar objects occluding the scene can result in holes in the scene reconstruction. The strength of our approach is the ability to use surface-level correspondence across multiple views to create composite texture maps for the scene thereby rendering pleasing planar reconstructions of the scene. Moreover, we use this correspondence to obtain volumetric reconstructions of the occluding object. All of this involves very simple interactions in the form of scribbles.

Scribbles have been used in interactive algorithms in the past. They were first used by Boykov *et al*. for interactive segmentation to indicate foreground and background [10]. Batra *et al*. used scribbles for interactive cosegmentation [11], which was applied to object of interest 3D modeling by Kowdle *et al*. [12]. Srivastava *et al*. improve the 3D model obtained using their Make3D algorithm by using scribbles to enforce coplanarity in their MRF formulation [13]. Sinha *et al*. used scribbles for texture synthesis where scribbles over an occluding object helps *remove* the occluding object in the visualization by using texture from the other views [9]. We use scribbles from the user to indicate the surfaces and non-planar objects in the scene. We believe we are the first to use these scribbles in a multi-class segmentation framework.

An overview of our algorithm is illustrated in Fig.1. Our algorithm allows the user to pick any image and provide simple scribbles to indicate planar surfaces and non-planar objects in the scene. We use the scribbles to learn an appearance model for each surface and then, formulate the multi-class segmentation task as an energy minimization problem over superpixels, solved via graph-cuts. This scene segmentation along with the sparse 3D point cloud from structure-from-motion (SFM) helps define the geometry of the scene. We introduce an idea of *3D scribbles* which helps propagate this scene geometry to the other images to *co*-segment the images into the various planes and objects in the scene. The scene co-segmentation helps obtain a composite texture map for the scene eliminating holes due to occluding objects, giving a pleasing planar reconstruction of the scene. In addition to this, we use the co-segmentation of non-planar objects in the scene to obtain a visual hull for the occluding object [12], which is rendered as part of the prior planar reconstruction of the scene. We now describe our approach in detail.



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

Fig. 1. Overview of system: (a) Input images (Image selected by user shown in yellow box); (b) User interactions to indicate the surfaces in the scene; (c) Scene co-segmentation of all images by using the idea of 3D scribbles to propagate scene geometry; (d) Some sample novel views of the reconstruction of the scene, with and without texture (Best viewed in color).

2. ALGORITHM

We first run the structure-from-motion algorithm by Snavely *et al*. [1] on the images to recover the camera projection matrices for all the views, a sparse 3D point cloud and the set of the points visible by each camera. We now describe the algorithm starting from the user scribbles, to how we obtain the final 3D reconstruction via scene co-segmentation.

2.1. Scribbles to scene segmentation

We have developed a java based user interface[1] using which the user selects any image in the group and provides scribbles on the image with different colors indicating different surfaces in the scene as shown in Fig.1(b). Given these scribbles, we cast the multiclass labeling problem as an energy minimization problem over a graph of superpixels[2] constructed over the image scribbled on. Specifically, consider an image-scribble pair $D = \{X, S\}$, where the image X is represented as a collection of n sites (superpixels) to be labeled, $X = \{X_1, X_2, \ldots, X_n\}$. The user provides a set of scribbles S on the image with multiple labels (say user defines $\boldsymbol{p}$ surfaces in the scene), which is represented as the partial set of labels for these sites $S = \{S_1, S_2, \ldots, S_n\}$ where, $S_i = \{\phi, 1, 2, \ldots, p\}$. We build a graph, $G = (V, E)$, over the superpixels, with edges between adjacent superpixels.

Using these labeled sites, we learn an appearance model A. We then define an energy function over the image as:

$$E(X : A) = \sum_{i \in V} E_i(X_i : A) + \lambda \sum_{(i,j) \in E} E_{ij}\left(X_i, X_j\right), \qquad (1)$$

where the first term (data term) indicates the cost of assigning a superpixel to one of the labels, while the second term (smoothness term) is used for penalizing label disagreement between neighbors. The colon (:) in the equation indicates that the term is dependent on the learnt appearance model.

Data (Unary) Term. Our appearance model consists of a Gaussian Mixture Model for each of the $\boldsymbol{p}$ surfaces labeled, i.e, $A = \{\text{GMM}_1, \ldots, \text{GMM}_p\}$. Specifically, we use colour features extracted from superpixels [12] on the labeled sites and fit GMMs for the corresponding classes. The data terms for all sites are then defined as the negative log-likelihood of the features given the class model. We set the unary term of the superpixels labeled by the user to $-\infty$ (a large negative value) as hard constraints in the energy minimization.

Smoothness (Pairwise) Term. We use the commonly used Potts model to model the smoothness term,

$$E_{ij}(X_i, X_j) = \mathrm{I}\left(X_i \neq X_j\right) \, \exp(-\beta), \qquad (2)$$

where $\mathrm{I}(\cdot)$ is an indicator function.

Finally, we use graph-cuts (with α-expansion) to compute the MAP labels for all superpixels, using the implementation by Bagon [15] and Boykov *et al*. [16, 17, 18]. The result segments the image into the different surfaces labeled by the user as shown in Fig.2(a); we call this *scene segmentation*. The parameters λ and β were empirically chosen and fixed for all scenes. This was found to work well in practice.

2.2. Scene segmentation to 3D geometry

Using SFM we have a sparse 3D point cloud and the 2D feature correspondence across the images for this point cloud. We therefore know the subset of 3D feature points seen from the current view (scribbled image). This information helps transfer the labels from the 2D scene segmentation to the 3D points, based on which scene segment the 3D points project onto. We now use RANSAC-based plane-fitting on the labeled 3D points to estimate the plane parameters of the labeled planes enforcing that the plane normal points outwards i.e. towards the camera looking at the scene.

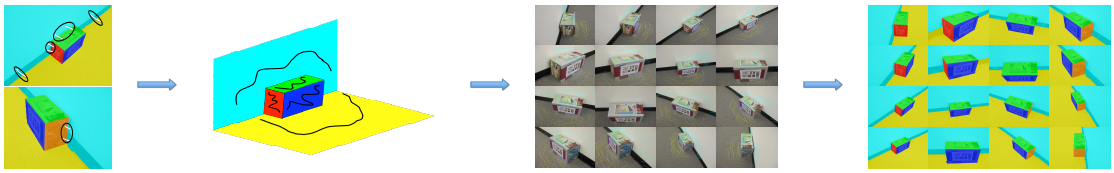
We note here that, there may be featureless surfaces like the wall in the scene, which lacks enough cues to be reconstructed. The algorithm then prompts the user for some simple additional interactions to indicate the edges shared by this surface with the other surfaces in the scene by easily scribbling two lines across the edge shared as shown in black ellipses in Fig.2(a). We obtain an estimate of the plane parameter by enforcing that the boundary points lie on the corresponding connecting planes, thus, resulting in globally optimal plane parameters. However, if the featureless surface shares just one edge with another plane, we make perpendicularity assumptions for that surface to choose the most probable plane amongst the infinite planes which can share that edge. This assumption has been shown to work well [19] and would be the best possible estimate, given the support.

2.3. 3D scribbles and scene co-segmentation

Image co-segmentation has gained a lot of popularity in the community [20, 21, 11]. However, co-segmentation of the multiple surfaces in the scene is not as trivial as the two class image co-segmentation since, it is hard to define features discriminative between geometric surfaces. However, when a

[1] iScribble, *http://chenlab.ece.cornell.edu/projects/iScribble/iScribble.html*

[2] We use mean-shift [14] to break an image to about thousand superpixels.



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

Fig. 2. Scene co-segmentation: (a) Scene segmentation with user interaction indicating connected planes (white scribbles in black ellipses); (b) 3D scribbles inferred from the segmentation; (c) 3D scribbles warped onto the other images to propagate scene geometry (Note: scribbles have been increased to improve visibility; the scribbles used for the results are in Fig.1(b)); (d) Scene co-segmentation (Best in color).

user provides scribbles on an image, they are doing so based on their perception of the geometry of the scene, i.e. they are not just indicating surfaces and objects in *that* image but, are giving us cues about the 3D scene geometry common across all the images. This is the common thread between the images we exploit to perform the co-segmentation.

3D scribbles. Using the estimated plane parameters and the camera projection matrix of the scribbled image, we develop the idea of *3D scribbles*. Let the projection matrix of camera i be defined as $M_i = K_i R_i (I - C_i)$ where, K_i is the intrinsic matrix, R_i is the rotation matrix and C_i is the camera center in the world co-ordinate system. Consider, a 2D scribble point $s_{1,j}$ seen from Cam_1, on a segment which corresponds to the plane l parameterized by $[\hat{n}_l \; d_l]$ where, $\hat{n}_l$ is the plane normal and d_l is the plane constant. The projection of this scribble point on another image seen from Cam_2 ($s_{2,j}$) is given by,

$$s_{2,j} = K_2 R_2 \left(\left(\frac{(-d_l - \hat{n}_l.C_1)}{\hat{n}_l.([K_1 R_1]^{-1} s_{1,j})} [K_1 R_1]^{-1} s_{1,j} + C_1 \right) - C_2 \right)$$

We take care to avoid warping the scribbles onto occluded planes by using the scene geometry and camera pose. For example, we can eliminate many of the warped scribbles by considering only the planes visible from a particular view.

Scene co-segmentation. The resulting scribbles on all the images are as shown in Fig.2(c). Using these scribbles as hard constraints on all the images, we now extend the energy minimization based multi-class labeling described in Sec.2.1 to all the images thereby achieving *co*-segmentation of all the images into the multiple scene classes Fig.2(d).

2.4. Visualization

We develop a back-projection algorithm using the equation above, to evaluate the point of intersection of a ray from the camera center through every pixel on the image plane, and the estimated 3D surface. Using these 3D points, we generate a mesh for the scene with the corresponding image texture and render a texture mapped planar reconstruction of the scene as shown in Fig.1(c), enabling pleasing fly-throughs.

2.5. Rendering non-planar objects

The algorithm thus far renders a planar reconstruction of the scene. In case of non-planar objects, we get an input from the user to indicate these objects, as shown in the blue ellipse in Fig.5(c). This tells the algorithm which surface corresponds to the non-planar object. We then estimate an approximate planar proxy for the object, which helps position the object as part of the rendered scene. Recent automatic approaches [22, 23] can also be used to identify non-planar regions.

(a) (b)

Fig. 3. Non-planar objects: (a) Composite texture map for the scene (top) allows covering up holes due to occlusions (ellipse); (b) Novel views of the reconstruction with a volumetric model of the tree.

Object co-segmentation. At this stage, the algorithm knows which surface indicated by the user corresponds to the non-planar object. We treat the scribbles corresponding to the non-planar object as foreground scribbles and all other scribbles as background scribbles and use ideas from prior work by Kowdle *et al*. [12] to obtain a 3D *visual hull* of the non-planar object via a 2-class co-segmentation, which is rendered using an independent mesh. The scene co-segmentation also allows us to create a composite texture map for the scene covering up holes due to occlusions as shown in Fig.3(a).

Once the algorithm generates the 3D reconstruction, the user can provide more scribbles to indicate new or previously occluded planes, and improve the result, thus closing the loop on our interactive 3D reconstruction algorithm.

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

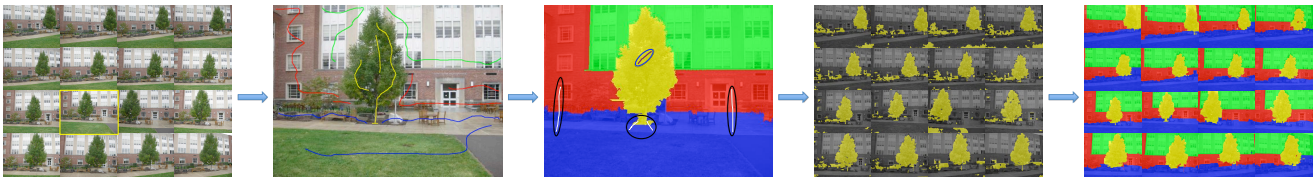
Fig. 4. More results: Novel views of the reconstructed scenes.

3. RESULTS AND DISCUSSIONS

We test our algorithm on a number of scenes (both indoor and outdoor) rendering pleasing, complete reconstructions. Fig.1(d) shows the result on a scene with featureless surfaces. Fig.4 how more results on such planar scenes. The algorithm also render non-planar objects as we show with the tree in the outdoor scene in Fig.3(b) and the person in the indoor scene in Fig.4(d). Please see video summary[3] with fly-through of the 3D reconstructions.

Comparison. We compare our results with other publicly available algorithms to reconstruct a scene. Using SFM [1],

[3] Video Summary: *http://chenlab.ece.cornell.edu/projects/Interactive_3D*



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

Fig. 5. Outdoor scene with occlusion: (a) Input images (Image selected by user shown in yellow box); (b) User interactions; (c) Resulting scene segmentation with the additional interactions to indicate surface connectedness (white scribbles shown in black circles) and non-planar objects (magenta scribble shown in blue scribble); (d) Object co-segmentation (foreground non-planar object in yellow); (e) Scene co-segmentation by using 3D scribbles to propagate scene geometry (Best viewed in color).

on a huge image collection can render dense point clouds however, in this scenario, the point cloud is very sparse. Multi-view stereo algorithms like patch-based multi-view stereo (PMVS)[4] render a denser reconstruction. However, this fails to render a complete reconstruction, leaving holes and rendering inaccurate geometric reconstructions, in the presence of textureless surfaces and specular surfaces. As we show in Fig.6, the results from our interactive reconstruction algorithm is more complete and geometrically accurate.

(a) (b)

Fig. 6. Comparison with patch-based multi-view stereo: The top images show the reconstruction generated by PMVS with the errors shown in black ellipses, while bottom images show our results with corrected reconstructions shown in blue ellipses (Best in color).

To compare our approach with other interactive works, we show our result on the play-house dataset of Sinha *et al*. [9], in Fig.4(c). We note that prior works require tedious user interactions to mark the planes in the scene or provide line models of the scene, while we achieve good results using limited and simple scribbles to indicate the surfaces. Moreover, unlike prior work, the proposed work can reconstruct non-planar objects like the tree, making the reconstruction more complete.

4. CONCLUSIONS

In this paper, we present a novel interactive 3D reconstruction algorithm which uses simple user interactions in the form of scribbles to indicate the surfaces and non-planar objects in the scene. We introduce the idea of 3D scribbles to propagate the scene geometry and co-segment the scene across multiple views. We render a planar reconstruction of the scene and introduce the idea of overlaying a volumetric rendering of the occluding non-planar object as part of the scene thus rendering a more complete reconstruction of the scene.

5. REFERENCES

[1] N. Snavely, S. Seitz, and R Szeliski, "Photo tourism: Exploring photo collections in 3d," in *SIGGRAPH*, 2006, pp. 835–846.

[2] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *PAMI*, 2009.

[4]We use the PMVS implementation by Furukawa *et al*. [2] and available at http://grail.cs.washington.edu/software/pmvs/pmvs-1/index.html

[3] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *IJCV*, vol. V59, no. 3, pp. 207–232, 2004.

[4] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," 2006, vol. 1, pp. 519–528.

[5] S. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *ICCV*, 2009.

[6] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Reconstructing building interiors from images," in *ICCV*, 2009.

[7] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *SIGGRAPH*, 1996, pp. 11–20.

[8] A. Hengel, A. R. Dick, T. ThormŁhlen, B. Ward, and P. H. S. Torr, "Videotrace: rapid interactive scene modelling from video.," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 86, 2007.

[9] S. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3d architectural modeling from unordered photo collections," *SIGGRAPH Asia*, 2008.

[10] Y.Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," *ICCV*, 2001.

[11] D. Batra, A. Kowdle, D. Parikh, J. Luo, and Chen. T, "iCoseg: Interactive co-segmentation with intelligent scribble guidance," in *CVPR*, 2010.

[12] A. Kowdle, D. Batra, W. C. Chen, and Chen. T, "iModel: Interactive co-segmentation for object of interest 3d modeling," in *ECCV - RMLE Workshop*, 2010.

[13] S. Srivastava, A. Saxena, C. Theobalt, S Thrun, and A. Y. Ng, "i23 - rapid interactive 3d reconstruction from a single image," in *Vision, Modelling and Visualization*, 2009.

[14] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *PAMI*, vol. 24, no. 5, pp. 603–619, 2002.

[15] Shai Bagon, "Matlab wrapper for graph cut," December 2006.

[16] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.," *PAMI*, vol. 26, no. 9, pp. 1124–1137, 2004.

[17] Y. Boykov, O. Veksler, and R. Zabih, "Efficient approximate energy minimization via graph cuts," *PAMI*, vol. 20, no. 12, pp. 1222–1239, 2001.

[18] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *PAMI*, vol. 26, no. 2, pp. 147–159, 2004.

[19] D. Hoiem, A. Efros, and M. Hebert, "Automatic photo pop-up," in *ACM SIGGRAPH*, August 2005.

[20] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *CVPR*, 2006.

[21] L. Mukherjee, V. Singh, and C. R. Dyer, "Half-integrality based algorithms for cosegmentation of images," in *CVPR*, 2009.

[22] D. Gallup, J. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *CVPR*, 2010.

[23] F. Lafarge, R. Keriven, M. Brédif, and V. Hiep, "Hybrid multi-view reconstruction by jump-diffusion," in *CVPR*, 2010.