3D Editing System for Captured Real Scenes

Inwoo Ha, Yong Beom Lee and James D.K. Kim
Samsung Advanced Institute of Technology, Youngin, South Korea
E-mail: {iw.ha, leey, jamesdk.kim}@samsung.com Tel: +82-31-280-6681

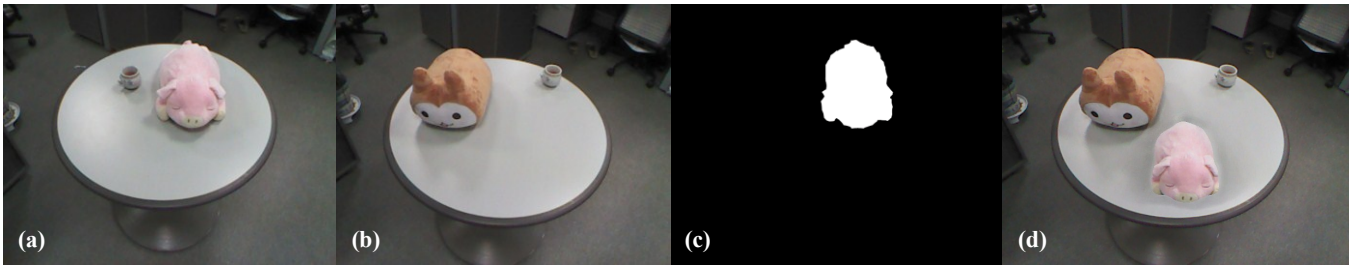

Fig. 1 Our 3D editing system: Each image shows (a) source image, (b) target image, (c) object selection mask and (d) editing result with our proposed algorithm.

***Abstract*—This paper presents a complete 3D editing system for real scenes, captured from conventional color-depth camera. From captured source and target images, desired source objects and target locations are selected. The source objects are copied and pasted to the desired location of the target image in color layer not considering their mutual illumination between the source objects and target image. To seamlessly composite the source objects to the target image based on their mutual illuminations, 3D surface meshes of source and target real scenes are reconstructed interactively, and differential rendering framework based on the instant radiosity is applied. The final result is a seamlessly mixed image considered with correct occlusions and mutual illumination.**

I. INTRODUCTION

To edit real scenes is a promising research topic in computer graphics and vision communities while providing many useful applications such as movies, games, and various augmented or mixed reality applications. Meanwhile due to the recent advancement of depth camera, interactive capturing of coarse depth values of the real scene becomes possible, and color & depth image contents captured by users are increasing. Therefore the demand for an easy and photorealistic interactive 3D content editing tool is growing.

The simplest way to edit 3D content comprising color and depth images is to directly apply well known 2D image editing techniques. Those techniques, however, have fundamental limitation because occlusion and mutual illumination effects, which are perceptually significant, have difficulty in being simulated in 2D image space. Basically all editing works of 3D have to be processed in 3D model space.

Although recent advancement of depth sensor and signal processing technique makes interactive capturing of real scene simple-and-easy as Kinect is prevailing, there are several technical problems, such as limitations of the sensible depth range and accessibility, and improper depth sensing for some reflectance properties. Moreover, disocclusion of color and depth pixel registration and perspective unprojection process produces unexpected large holes. Therefore, the raw depth map captured from the depth camera cannot be applied immediately to applications. This paper presents a novel method for filling holes and filtering noises in depth images. The produced coarse holes are interpolated and filled with suggested dual-weight pull-push algorithm. The reconstructed 3D geometry is filtered by bilateral filtering.

Furthermore, to seamlessly mix source and target real scenes, their mutual illumination should be considered. Differential rendering [1] assumes that captured real scene may not be accurate, where geometry of the real scene is used only to calculate the color difference changed by mutual illumination with inserted synthetic objects. In addition, recent observations in computer graphics show that human visual perception may not sensitively recognize shading difference in diffuse surface with geometrical approximation [8]. Our approach exploits approximated geometry of the real scene modeling as well as its visibility. Sampling method of imperfect shadow map is adjusted to provide more robust composition results under mutual illumination between source and target real scenes. As a result, a complete pipeline for editing real scenes is feasible using only a single frame color-depth image captured from a conventional depth camera. An overview of the system is shown at Fig. 2.

II. RELATED WORKS

In image processing and computer vision community, a number of papers have dealt with issues involved in editing real scenes in a visually pleasing way [6], [2]. Those ideas commonly assume that a user has chosen the object in advance with a good fit for the target image. In [5], objects

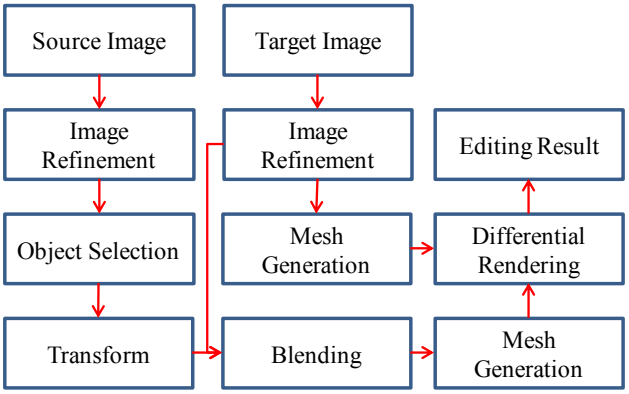

Fig. 2 Our system overview.

are inserted into a target image from a database of pre-segmented and labeled images. The 3D scene structure and lighting are estimated through image analysis. However, because all the approaches are based on 2D information, they can't generally produce 3D effects, such as occlusion and mutual illumination.

It has been mentioned that current depth cameras have limitations of resolution and accuracy to immediate application of 3D scene modeling. To generate 3D mesh from the depth camera image, holes of the image should be filled in through the interpolation algorithms and then noise should be filtered effectively. Interpolation of scattered data points is a classical problem in computer graphics with many applications. In particular, moving least square (MLS) is widely used to generate smooth surfaces. However, since it requires quite a long processing time, this method is not suitable for interactive applications. Kinect Fusion [7] shows a demo to create real scene geometry using the registered multi-frame depth images. Real scene geometry is progressively refined based on successive depth images.

Upon an assumption that real scene information is not complete, the differential rendering framework is [1] introduced. [4] presents the real-time system for seamless composition of synthetic objects with real environment. Although their work simulates mutual illumination between real scene and synthetic objects, the results are still based on pre-modeled real scene. [3] shows seamless composition of synthetic objects into real scene using only color images. However, this method requires user's manual interactions to annotate light and geometry information.

III. REAL SCENE MODELING AND SELECTION

The point clouds achieved from the raw depth image have lots of holes and are noisy. Although the low resolution noisy depth map may be generally suitable for detecting motion, sophisticated steps are required to be applied for 3D geometry reconstruction. Moreover intrinsically, captured depth image has perspectively projected. In order to use the model constructed from the depth map for 3D editing, however, the depth values in the image space need to be unprojected into the model space. In this research, we perform perspective

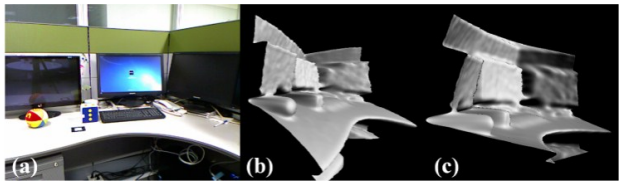

Fig. 3 Perspective unprojection: (a) shows the input color image for the real scene. The distorted geometry at its side view is shown in (b), and the refined geometry by perspective unprojection is shown in (c)

unprojection for the depth values, and fill the holes with our novel dual-weight pull-push algorithm. Finally the reconstructed 3D geometry is refined by bilateral filtering.

The constructed geometry can be applied for 3D editing, which is to select and copy objects from the source image and paste them to the desired target location. For this purpose, a user interface to make selection of objects what a user wants is required. In addition, color-based segmentation is not robust enough because colors between boundary pixels are often similar. Thus, both of color and depth information for the segmentation are considered. Thus, our algorithm detects edges with pixel differences computed with color and hole-filled depth image, which enhances the robustness of segmentation process.

A. *Real Scene Modeling In Perspective-Unprojected Space*

The depth image captured from a depth camera is a perspectively projected image. Therefore, 3D geometry reconstructed without perspective unprojection is distorted in the 3D space as shown in Fig. 3. The perspective unprojection can be easily computed as followings:

$$X(\mathrm{p_i}) = O + OP_i D(p_i) \qquad (1)$$

where $X(p_i)$ is the perspective-unprojected position of i-th pixel, $\mathrm{p_i}$ is the pixel of an input depth image which is projected to the near plane of the camera in the 3D space, O is the camera position in the 3D space, OP_i is the direction vector from the camera to p_i, and $D(p_i)$ is the actual depth value of p_i.

B. *Hole Filling With Dual-Weight Pull-Push Scheme*

The pull-push algorithm is based on the hierarchical interpolation using image pyramids. In the pull phase, image pyramids of both depth image and color image are generated while averaging colors and depth values of fine level hierarchically. In the push phase, holes of the depth image are filled while recomputing all levels of image pyramids from coarse pixels p_j^{r+1} to fine pixels p_i^r. To interpolate the values of holes, the bilateral upsampling is exploited [8]. We compute the 3D position of the interpolated depth pixel with the perspective unprojection. Because the 2D distance used as the bilinear weight in the usual bilateral upsampling is not correct in the 3D space, thus we use 3D Euclidean distance with its neighbors, $W_d(p_i^r, p_j^{r+1})$, for the weights.

$$W_d(p_i^r, p_j^{r+1}) = \frac{1}{(\epsilon + \|X(p_i^r) - X(p_j^{r+1})\|)} \quad (2)$$

Since color image has no holes, color similarity weights can guide the hole-filling. The color similarity weight $W_c(p_i^r, p_j^{r+1})$ is defined using the color difference between fine and coarse level pixels.

$$W_c(p_i^r, p_j^{r+1}) = \frac{1}{(\epsilon + \|C(p_i^r) - C(p_j^{r+1})\|)} \quad (3)$$

where $C(p_i^r)$ is the RGB color vector of the pixel p_i in the level r, and ϵ is a tiny constant.

Then, each hole $D(p_i^r)$ in the level r is interpolated with coarse level pixels $D(p_j^{r+1})$ in the level r+1.

$$D(p_i^r) = \frac{\sum_j W_d(p_i^r, p_j^{r+1}) W_c(p_i^r, p_j^{r+1}) D(p_j^{r+1})}{\sum_j W_d(p_i^r, p_j^{r+1}) W_c(p_i^r, p_j^{r+1})} \quad (4)$$

After filling holes using the dual-weight pull-push algorithm, 3D point clouds $X(p_i)$ and it's triangular mesh are constructed. The results of our hole-filling are shown in Fig. 4.

The constructed triangular mesh is noisy. We apply bilateral filtering to the hole-filled depth image. We use both of color and depth similarity for weight. With this process, we can refine noisy depth values while preserving important features such as salient edges.

C. Selection

All the edges in the source image are detected from combined weights of color and depth derivatives. The combined weight of a pixel is a weighted sum of color and depth derivatives. When the event of mouse click occurs, the initial pointer moves to the nearest detected edge. Then line segment begins, where initial position information is stored. While mouse movement is tracked, the pointer moves to the detected edges continuously. The positions are stored continuously at some pixel intervals. If the mouse button is clicked again, the pointer moves to the nearest detected edge again thus connecting a line segment between the previous stored position to the pointer position. When the pointer arrives at the initial point, the detected area is enclosed by creating a line segment from the initial point to the final point. Through this interactive approach, we can select the desired region, which is stored as source object mask α (1 where source objects is present and 0 otherwise).

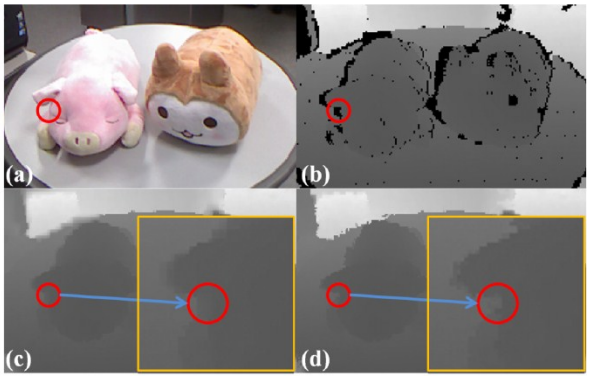

Fig. 4 Results of the hole filling: the holes in the depth image (b) is filled with the pull-push algorithm with the bilinear weights (c) and our algorithm (d).

A user first selects a region to be duplicated. Since the depth value scales of source and target image are normally different, the source depth image is scaled to the suggested initial value. The suggested initial value is scaled so as the bounding box of source object doesn't penetrate the target surface. Then a user is able to adjust the scale value to the desired value by monitoring the result interactively.

Finally the source object is moved to the desired target location, and blended with the target image in both color and depth layers. Since the boundary of the source object usually is not continuous with the target image, Gaussian filtering to the source object mask is applied. In the composition process, the source object mask is used as blending weights between source and target images.

IV. COMPOSITION

Seamless 3D copy and paste editing requires to handle occlusion and mutual illumination between source objects and target image.

Occlusion is handled with depth value comparisons, which determine if source object is in front of target image for each pixel or not. If a depth pixel of source object has deeper value than that of target image, it means that the pixel of source object should be occluded by that of target image. Then we select target image color value instead of source image color value in that pixel.

To render mutual illumination between source objects and target image, imperfect shadow map is adjusted for adaptive differential rendering to provide real time global illumination.

A. Adaptive Differential Rendering for Mutual Illumination

The reconstructed real scene geometries are then used to paste source objects into the target real scene considering mutual illumination. We exploited the differential rendering scheme [1] to update only the color difference produced by augmented source objects. Instant radiosity with imperfect shadow map (ISM) [8] is exploited for mutual illumination rendering. When using the environment map as the light source, virtual point lights (VPLs) are assigned into the

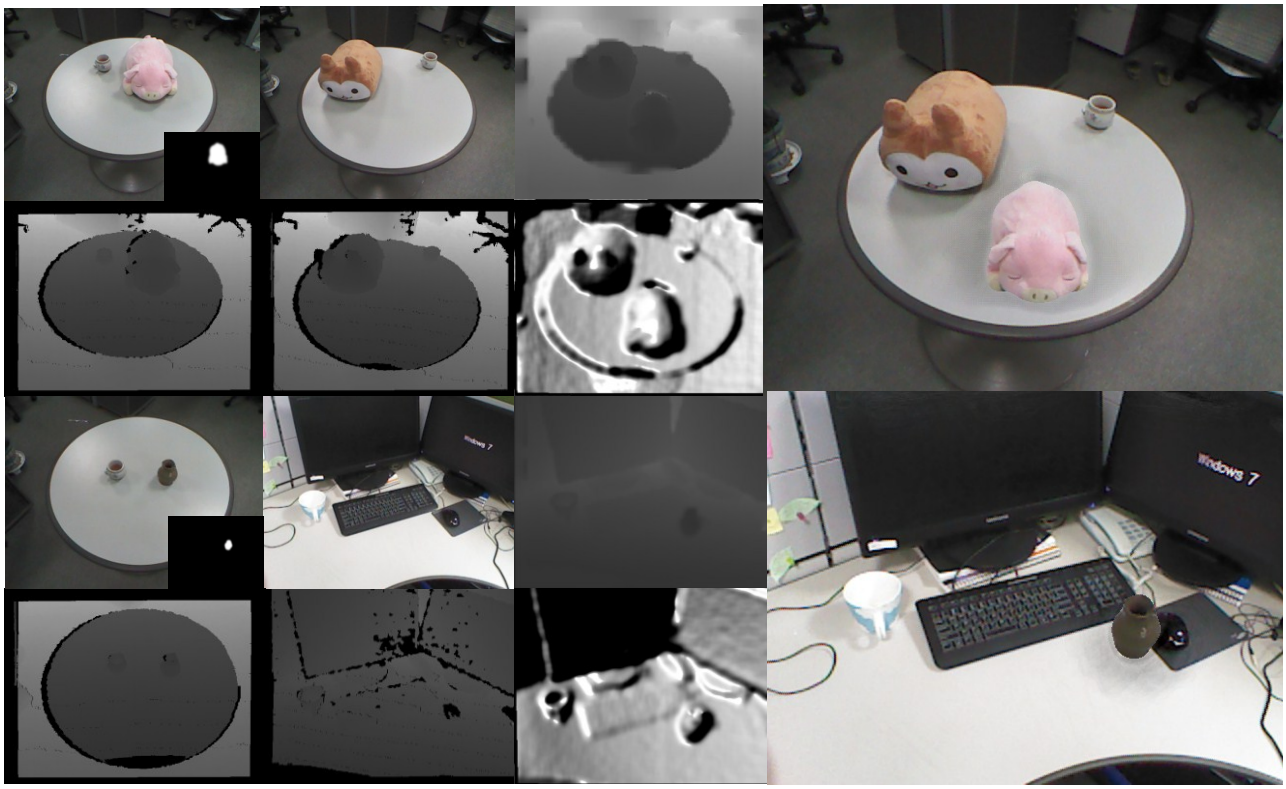

Fig. 5 3D editing results: The left two columns show the captured source and target images with color and depth, where the source object mask is shown in the corner of the source color image. The middle column presents refined depth image and generated mesh for the blended scene. The right column shows the editing results.

environment map as the primary light sources. We approximate the environment map with VPLs using importance sampling and use ISMs to compute shadows.

Two images need to be rendered under real light environment; the image I_{s+t} contains both source and target objects, and the image I_t contains only target objects. Then the composition I_m of the synthetic objects on the input color image P is computed by

$$I_m = \alpha I_{r+s} + (1- \alpha)(P + I_{r+s} - I_r) \quad (5)$$

The depth sensing camera such as Kinect has wide depth range due to its initial setting for full body motion capture. Hence, the real scene geometry created from the current depth sensing camera is generally large. The ISM uses shadow maps based on the approximated point samples of the scene. The resolutions of the shadow map and the number of point samples are highly limited for interactive rendering. When a relatively small source object is inserted into the large real scene, the details of the source object are condensed resulting in blurry shadow. Thus, we adjust the method based on the intuition that the region near the inserted source object is visually important for composition. Importance sampling is performed for each triangle p_i with inverse distance weight I_{pi} from each source object center o_j to be placed.

$$I_{p_t} = \sum_j \frac{1}{d(p_i,o_j)} \quad (6)$$

where $d(x,y)$ is a function to represent the Euclidean distance between x and y.

V. RESULTS

Our method is tested using a conventional depth camera working on a PC; 3.2GHz i7 CPU and NVIDIA Geforce GTX 590.We tested our method using Microsoft Kinect to capture color and depth images. To align color and depth pixels, the factory calibration information is used. To capture a lighting environment, fish-eye lens camera is used. 3D geometry of the real scene is constructed from the depth images. The constructed geometry is then applied to the 3D editing of real scenes as shown in the Fig. 5. For rendering, we used 256 VPLs and 8000 points for each VPL. Since our pipeline is carefully designed for interactive applications, the editing process can be performed at interactive frame rates. Around 30 FPS is required from capturing to final editing of Fig. 5.

Fig 5. shows the results of suggested 3D editing system. Because the depth images have lots of holes and noises, we can't apply these images in a raw of mesh generation and rendering. Therefore, the suggested novel image refinement algorithm is applied to the input source and target images first. The selection mask is generated by user-assisted segmentation along with color and depth discontinuity information. The modeled mesh is not complete to directly render. With

differential rendering framework, however, source objects could be seamlessly inserted to the target images.

VI. CONCLUSIONS

This paper presents a complete pipeline of editing real scenes from desired source object selection to composition. 3D geometry reconstructed from the current depth camera has limitations to provide necessary details for high quality geometry. However, our results provide enough details for 3D copy and paste editing of real scenes with our novel mesh refinement algorithm and adaptive differential rendering. Our basic intuition is that to compensate inaccurate depth input with color input for mesh refinement and segmentation. We assume that all the materials of real scene are Lambertian, and its light environment is approximated with the captured environment map by fish eye lens camera. In the future, we will investigate sophisticated methods for estimating reflectance model and lighting environment to enhance the overall quality. A further interesting research direction is to transform source objects with more degrees of freedom, which couldn't be handled in the current system. The overall performances can be improved using further optimization and GPU acceleration.

REFERENCES

[1] P. Debevec, “Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography.”, In Proc. *ACM SIGGRAPH* '98, pages 189-198, 1998.

[2] J. Jia, J. Sun, C.-K. Tang, H.-Y. Shum, “Dragand-drop pasting.”, ACM Trans. on Graph. 25, 3 (July), 631–637, 2006

[3] K. Karsch, “Rendering synthetic objects into legacy photographs.”, In Proc. *ACM SIGGRAPH Asia* '11, pages 157:1-157:12, 2011.

[4] M. Knecht, “Differential instant radiosity for mixed reality.”, In Mixed and Augmented Reality (ISMAR), 2010 9th *IEEE International Symposium* on, pages 99-107, oct. 2010.

[5] J. F. Lalonde, and et al., “Photo clip art.”, ACM Trans. on Graph. 26, 3 (July), 2007.

[6] P. Perez, M. Gangnet, A. Blake, “Poisson image editing”, ACM Trans. on Graph. 22, 3 (July), 313–318, 2003.

[7] A. Richard, “Kinectfusion: Real-time dense surface mapping and tracking.”, *In Mixed and Augmented Reality (ISMAR)*, 2011 10th IEEE International Symposium on, pages 127-136, oct. 2011.

[8] T. Ritschel, “Imperfect shadow maps for efficient computation of indirect illumination.”, *ACM Trans. Graph.* (SIGGRAPH Asia '08), 27:129:1-129:8, December 2008.

[9] P. Sloan, “Image-based proxy accumulation for real-time soft global illumination.”, In Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, pages 97-105, 2007.