Putting Real-World Objects into Virtual World: Fast Automatic Creation of Animatable 3D models with a Consumer Depth Camera

Hwasup Lim, Seong-Oh Lee, Jong-Ho Lee, Min-Hyuk Sung, Young-Woon Cha, Hyoung-Gon Kim, and Sang Chul Ahn
Imaging Media Research Center
Korea Institute of Science and Technology, Seoul, Korea
{*hslim,solee,purmod,smh0816,ywcha,hgk,asc*}*@imrc.kist.re.kr*

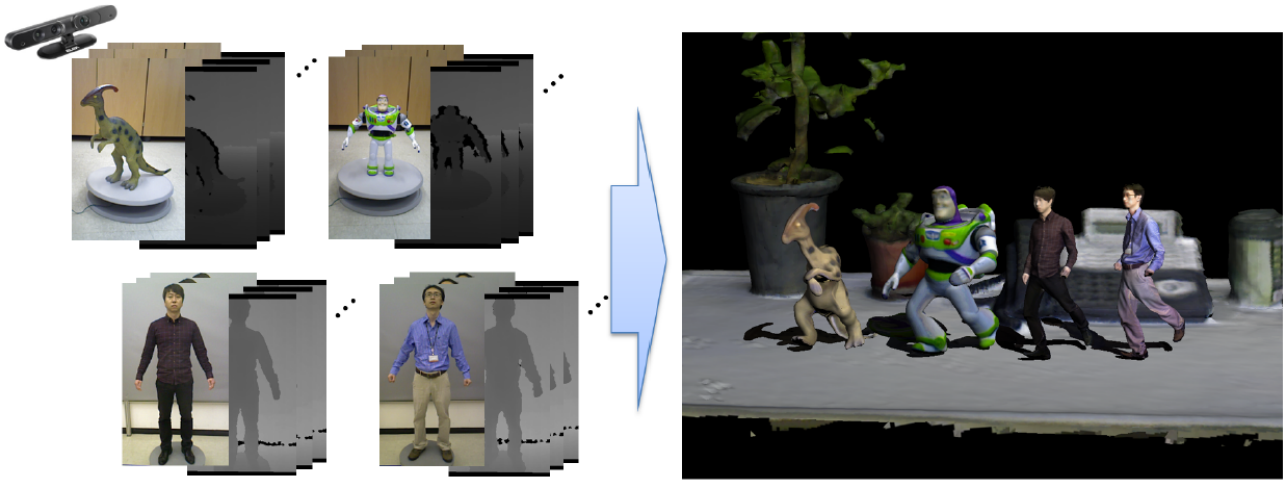

Figure 1. Animation of real-world objects in the virtual world. The 3D models were generated by our prototype system and rescaled for rendering purpose. The actual heights of two toys are around 35 cm and those of two persons are around 175 cm.

***Abstract*—Consumer depth cameras such as Kinect are gaining huge popularity due to their potential to real-time 3D surface reconstruction. The creation of tractable 3D object models from the reconstructed surface model, however, requires further processings for segmentation, mesh/texture generation, and skeletal rigging. Providing solutions for all these issues, this paper formulate a comprehensive framework for generating animatable 3D models of real-world objects using only a single depth camera, with no needs of any generic models and specific viewers. The proposed system enables easy creation of 3D models even for non-experts within a few minutes. As demonstrated in the experimental results, our system generates visually realistic 3D models and their plausible animations.**

***Keywords*-Kinect, 3D modeling, animation, surface reconstruction, texture generation**

I. Introduction

Consumer depth cameras such as MS Kinect and ASUS Xtion are receiving great interest due to their potentials in real-time pose recognition [1], dense surface reconstruction [2], and their applications to augmented reality [3] to list a few.

One of the most promising achievements of above is live surface reconstruction such as KinectFusion [2] whose reconstruction quality is comparable to that of high-end 3D scanners. The KinectFusion addresses the problem of recovering the surface geometry of scanned objects and scenes in real-time. However, the creation of animatable 3D object models with a pair of triangular mesh and texture image is another challenging problem important to augmented and virtual reality applications. In this paper we focus on a practical and comprehensive framework to construct such 3D models of real-world objects by integrating most suitable state-of-the-art algorithms upon the KinectFusion.

Our framework consists of two main flows: triangular mesh generation from a sequence of point clouds and texture image generation from a sequence of color images with their estimated camera poses. Skeletal rigging is performed optionally for skinned animation with motion data. A similar framework has been proposed for texture reconstruction from a set of range data and multi-view images in [4] where image-based registration is more focused for seamless texture image generation.

The contribution of our work is that we generalize a fully automatic framework for constructing reusable 3D models in common 3D file format from the raw point clouds and color images. All the processes are performed without user

978-0-7695-4766-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ISUVR.2012.12

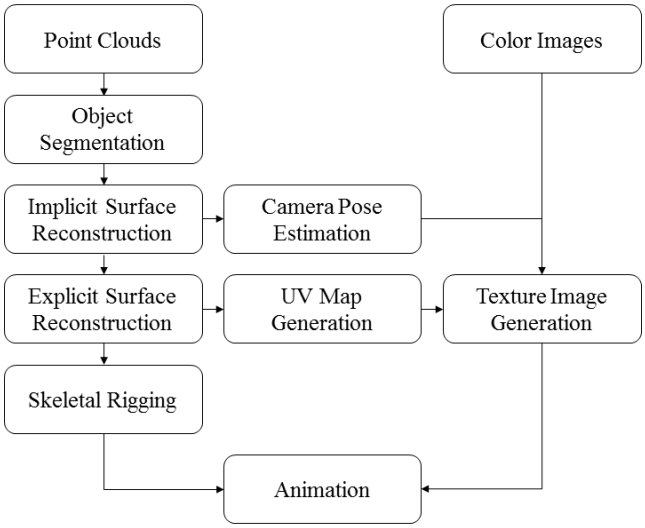

Figure 2. Overall system workflow.

interaction except for scanning the target object with the depth camera so that non-expert users or even children can easily produce high-quality 3D models. To our best knowledge, this systematic framework using Kinect-style depth cameras has not been sufficiently addressed before in the literature. Figure 1 demonstrates the final results of everyday objects and even human bodies using our system.

II. Overall System Workflow

Figure 2 depicts the overall workflow of the proposed system, which is composed of six independent components: target object segmentation from the input point clouds, implicit surface reconstruction by accumulating point clouds that belong to the target object, explicit surface reconstruction by producing a triangular mesh from the implicit surface, uv map generation by parameterization of the triangular mesh, texture image generation by projection of the target regions from the input images on the uv map, and finally skeletal rigging by fitting a human skeleton with skinning weights for animation.

We assume that the input point clouds are acquired by a Kinect-style depth camera and the depth and color cameras are calibrated and registered beforehand. The next section provides more technical details of each component.

III. 3D Model Creation and Animation

A. Object Segmentation

Assuming that the target object is placed on the planar area, we segment an object on a dominant plane around the center of the camera view. Multiple plane candidates are detected using the quick shift algorithm [5] from the normal image where each pixel represents the normal vector of the corresponding point. Similar candidates are grouped together to select the dominant plane by using the mean shift clustering method [6]. The object-centered coordinates are then determined from the center position and normal vector of the dominant plane.

B. Implicit Surface Reconstruction

For the sake of surface reconstruction from raw point clouds from a depth camera we employ the KinectFusion presented in [2][3]. The KinectFusion system takes live depth data from a moving depth camera and creates high-quality 3D dense surface models of objects and scenes in real-time. It consists of the four main stages. First, the live depth map is converted from image coordinates into 3D points and normals in the camera coordinate space. Second, in the tracking phase a rigid 6 degrees-of-freedom (DOF) transformation is computed to closely align the current oriented points with those of the previous frames using a GPU implementation of the iterative closest point (ICP) algorithm. The relative transformations computed at successive frames are incrementally combined to a single transformation that defines the global pose of the Kinect camera. Third, in the mapping phase a volumetric implicit surface representation based on truncated signed distance functions (TSDFs) is used for integrating surface information from point clouds with their normals. Each voxel stores a running average of its distance to the assumed position of a physical surface. Fourth, the intermediate volume is converted to a depth and normal map for model to frame registration in the next iteration.

C. Explicit Surface Reconstruction

In order to reconstruct a water-tight mesh model from the volumetric surface with TSDFs, we apply Poisson surface reconstruction algorithm in [7]. For given sample points with normal vectors computed at each voxel, the 3D surface is reconstructed in the manner of binary indicator function. The reconstruction is formulated as a Poisson equation which describes the mathematical fact that gradient of indicator function fits to the sampled surface normal field. The formulation involves locally basis smoothing functions such as Gaussian filter. Thus, it allows to efficiently remove data noise and approximate surfaces in the region on lacking sample points. Also, the locally supported functions lead to a well-conditioned sparse linear system, which enables to handle scalable 3D data accumulated from the depth camera. The isosurface resulting from the indicator function can be extracted using some conventional approaches such as marching cube method. The water-tight mesh model is essentially required for later skeletal rigging.

D. UV Map Generation

In most cases of multi-view surface reconstruction where the color images are simultaneously captured, the input color images are intactly used for texture rendering by projecting each triangle on the color image. Such scenario necessitates

specific viewers that load all the input images or selected key-frame images. For the consideration of the use of general 3D application we build a single uv map by partitioning the triangular surface mesh into several charts on the 2D image, based on least squares conformal maps [8]. It minimizes angle deformations and non-uniform scaling over the entire mesh, guaranteeing a unique solution. The boundaries of the charts after uv map generation are stitched again to preserve the water-tight integrity.

E. Texture Image Generation

Once we have the uv coordinates of each triangle of the surface mesh, each input color image can be converted to a texture image by projecting each triangle onto the color image plane with known camera pose and mapping these triangular image patches onto the uv map. For memory and performance issues, key-frames are selected only when significant camera movement is detected during scanning instead of storing all the image frames.

Since each texture image contains only the visible regions from the viewpoint of the camera at the acquisition instant, the texture images need to be combined to generate a single texture image for the model. A simple method of weighted averages of visible fragments is fast and straightforward, but susceptible to the blurring and ghosting artifacts due to the small geometric misalignment. A recent method using image stitching [9] produces high-quality textures. This method, however, requires a time-consuming optimization for re-alignment of each fragment with neighborhood smoothness constraints. Considering the trade-off between time efficiency and image quality, we utilize a median filtering with the visible parts for each pixel on the uv maps. It takes only a few seconds to generate a sharper texture image than that of weighted averaging.

After finishing this step, we now have a 3D model with a triangular mesh and its texture image applicable to general 3D applications. The next section introduces an optional for character animation.

F. Skeletal Rigging

Animating a 3D model requires the rigging process that embeds the skeleton inside the surface and defines the surface deformation according to the input skeletal motion. To make easy accessible to non-expert users we utilize an automatic rigging method introduced in [10]. This method builds a compact graph by packing spheres centered at the medial points. The predefined skeleton, humanoid skeleton here, is then embedded into the graph by minimizing a penalty function empirically learned from a training set. The bone weights for the linear blend skinning (LBS), which determines the vertex position from the weighted linear combination of nearby bone positions while moving, are then computed by finding heat equilibrium over the surface. The heat is transferred from the nearest bone.

IV. Experimental Results

We used a ASUS Xtion Pro Live whose specification is almost as same as MS Kinect and a motorized turntable for stable tracking of the target object. Four objects, toy dinosaur, Toy Story Buzz, and two human bodies, and one desk scene were modeled in this experiment. The voxel sizes were set to 3 mm for the objects and 6 mm for the scene. The texture image size was set to 1280 by 1280. As demonstrated in Figure 3, the generated models are visually realistic and capture fine-scale geometric details, for example, the thumb, physically less than 5 mm, of the Buzz model in the first row and wrinkles of the person models in the third and forth rows. Notice that the stripe pattern of the shirt in the fourth row is clearly visible. A slight surface distortion is, however, observed in the left arm of the person model in the fourth row. Unlike the first two static objects, local body motion while scanning leaded to the mismatch between the previously and newly accumulated surface models.

V. Conclusions and Further Work

We demonstrate that the proposed framework is capable of creating visually pleasing 3D animatable models from real-world objects with a minimal user interaction and very suitable for Kinect-style consumer depth cameras. There are numerous possibilities based upon the proposed framework to increase its texture and surface qualities because our framework generalizes the 3D modeling workflow with substantive components.

In this context our further work includes enhancement of the quality of the texture image using super-resolution algorithms and surface reconstruction of dynamic object with nearly rigid deformation.

Acknowledgment

This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MEST) (NRF-M1AXA003-20110028362)

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. of IEEE CVPR*, 2011, pp. 1297–1304.

[2] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. of IEEE ISMAR*, 2011, pp. 127 –136.

[3] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. of ACM UIST*, 2011, pp. 559–568.

(a)Textured model (b)Surface model (c)Texture image

Figure 3. Generated 3D models by the proposed system.

[4] F. Bernardini, I. M. Martin, and H. Rushmeier, "High-quality texture reconstruction from multiple scans," *IEEE Trans. on VCG*, vol. 7, no. 4, pp. 318–332, 2001.

[5] B. Fulkerson and S. Soatto, "Really quick shift: Image segmentation on a gpu," in *ECCV Workshop*, 2010.

[6] D. Comaniciu, P. Meer, and S. Member, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on PAMI*, vol. 24, pp. 603–619, 2002.

[7] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. of SGP*, 2006, pp. 61–70.

[8] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," in *ACM SIGGRAPH*, 2002, pp. 362–371.

[9] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or, "Seamless montage for texturing models," *Computer Graphics Forum*, vol. 29, no. 2, pp. 479–486, 2010.

[10] I. Baran and J. Popović, "Automatic rigging and animation of 3D characters," in *ACM SIGGRAPH*, 2007, p. 72.