The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

3D Human Modeling using Virtual Multi-View Stereopsis and Object-Camera Motion Estimation

D. Lam*, R. Z. Hong** and G. N. DeSouza***

Abstract—**This paper presents a method for multi-view 3D modeling of human bodies using virtual stereopsis. The algorithm expands and improves the method used in [5], but unlike that method, our approach does not require multiple calibrated cameras and/or carefully-positioned turn tables. Instead, an algorithm using SIFT feature extraction is employed and an accurate motion estimation is performed to calculate the position of virtual cameras around the object. That is, by employing a single pair of cameras mounted on a same tripod, our algorithm computes the relative pose between camera and object and creates virtual cameras from the consecutive images in the video sequence. Besides not requiring any special setup, another advantage of our method is in the simplicity to obtain denser models if necessary: by only increasing the number of sampled images during the object-camera motion. As the quantitative results presented here demonstrate, our method compares to the PMVS method, while it makes it much simpler and cost-effective to implement.**

I. Introduction

Object modeling has a wide range of applications in several areas such as: robotics [1], [18], virtual reality [9], [14], and even medicine and health care [16], [7]. In the latter case, creating 3D models of the human body can pose an even greater challenge due to the lack of texture in the human skin. This problem forces the use of unconventional methods, that is, methods that are not based only on intensity correlation. Hence, many approaches involving constrained global optimizations of photometric discrepancy functions have appeared in the literature recently. Some of these works were surveyed, evaluated, and can be found in [17]. In [8], for example, the authors reported a successful reconstruction of the human body using mult-view.However, the method, which was based on the detection of boundary motion extracted from the level set, still required specialized parallel PC hardware and software for efficient use of computation resources (e.g. load balancing). Other methods, such as [20], proposed a deformable model for 3D reconstruction of the human face. In this work, while the use of a reference face did away with the intensity correlation problem by aligning and warping the reference model with the observed face, their approach could only capture geometric properties of the face. So, additional post-processing for texture mapping was required in order to achieve a realistic representation of the face. Other more traditional approaches require range scanners, structured-light scanners, or any other sort of active sensor [10], [19], [13]. Those methods produced high quality models through the use of a controlled light source, such as a laser. However, the trade off was usually the high cost and the long time required for image acquisition.

Today, possibly one of the most successful approaches using only cameras is the patch-based multi-view system, or PMVS [5]. This method utilizes multiple images and their corresponding calibration matrices to reconstruct accurate 3D models of most objects, including human body and outdoor scenes. The approach starts with a sparse set of corresponding features, which are then expanded to the nearest projected pixels, forming 3D rectangular patches. The algorithm iterates between matching, expansion and filtering while it optimizes the reconstruction under photo consistency and global visibility constraints. The results were favorably contrasted to other 3D reconstruction methods in [17]. However, the method requires the use of well calibrated cameras and turn table positions.

In this paper, we propose a new method for 3D modeling that uses multiple virtual views from a single stereo pair. Our approach, while it is multi-view based, does not require a large number of calibrated cameras positioned around the object. Instead, our method only requires a single pair of calibrated cameras and a motion detection algorithm that estimates the position of virtual cameras as the object moves with respect to such cameras. Besides the much lower cost and despite the much simpler setup, the 3D models created using this approach is highly comparable to the original PMVS, while maintaining the same computational efficiency. Also, as

Department of Electrical and Computer Engineering, University of Missouri, 349 Eng. Building West, Columbia, MO, USA ⋆dlam and ⋆⋆ruizhi @mizzou.edu ⋆⋆⋆ DeSouzaG@missouri.edu

the original PMVS, our method works well on various objects, including human faces, as we demonstrate in the results section. Another great advantage of our method is in the simplicity to obtain denser models if necessary: by only increasing the number of sampled images during that object-camera motion.

The paper is organized as follows. First, we introduce the proposed framework. Next, the method for estimation of the camera-object motion is explained in Section II-B. Finally, the experimental results and discussion are presented in Section III, followed by the conclusions in Section IV.

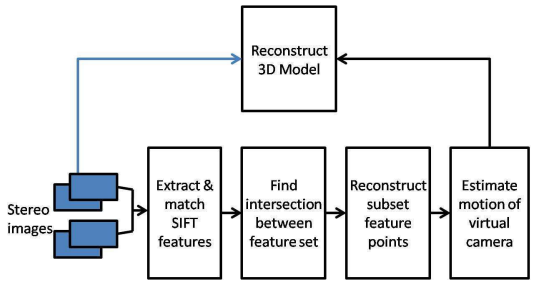


Figure 1: Proposed Framework for Virtual Multi-View 3D Modeling

II. Proposed Framework

Our framework for 3D object modeling consists of six majors steps. Figure 1 depicts such steps, which are: 1) Multiple pairs of stereo images are captured by 2 calibrated cameras while the object moves freely with respect to the cameras; 2) A SIFT-based feature extraction algorithm [3], [11] establishes the correspondence between various points on every stereo pair sampled; 3) The intersection between the sets of points from two consecutive pairs of images is determined. That is, the algorithm finds identical feature points from the left-right image pair obtained at camera-object position i and the subsequent pair obtained at camera-object position $i+1$; 4) The 3D coordinates of every point in the intersection above is calculated; 5) The transformation between camera-object poses are estimated using the 3D coordinates above; and 6) The previous transformations are used to create virtual poses of the camera and fed into a patched-base multi-view software [5] to reconstruct a 3D model of the object.

In the next subsections we explain further each of the steps above.

A. Real vs. Virtual Cameras

As we explained earlier, the input images are captured by a single pair of 640x480 pixel stereo cameras, as shown in Figure 2. Our system relies on two Firewire cameras connected to an embedded device that acquires both images at the exact same time. The cameras are mounted on the same tripod and are calibrated off-line using the CalTech Calibration Toolbox [2].

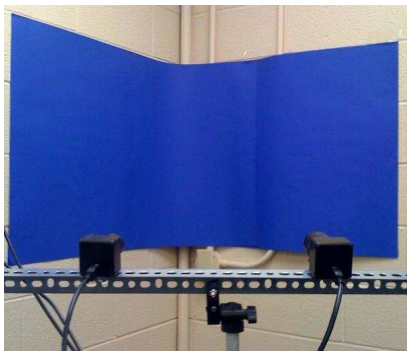


Figure 2: System setup for the Proposed Framework

In the original PMVS method, the reconstruction algorithm also relies on a small number of calibrated cameras: in that case there are three cameras. However, unlike in our system, their approach expands the number of views by employing a carefully-positioned turn table. That is, each camera acquires multiple images of the object, while the turn table is carefully rotated at pre-determined angles. In our method, we achieve an accuracy as good as that of the original PMVS, but we rely only on two cameras and no turn table. Instead, in order to obtain an arbitrary number of multiple views of the object, we resort to virtual cameras.

As illustrated by Figure3, our stereo cameras take images of the object as it moves freely about the camera. This motion of the object is interpreted by the algorithm as if it was the motion of the cameras. Better yet, as if the image sequence acquired by the cameras were taken by different cameras at slightly different poses: that is what we refer to as *virtual cameras*. In that sense, as the object moves in one direction, the algorithm computes the motion as if it was made by the cameras in the opposite direction. In fact, since the cameras are firmly mounted on the same tripod, there is really no different whether it is the camera or object that is actually moving. The problem becomes simply that of finding the pose of the virtual camera, as it is described in detail in Section II-B3.

B. Motion Estimation of Virtual Camera

The most important part of the proposed framework is to estimate the relative motion between camera and object, and from that to calculate the pose of the virtual

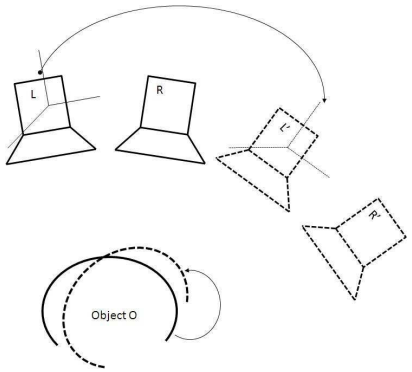


Figure 3: L,R are the real cameras, while L',R' are the estimated virtual cameras due to the motion of the object O (from solid to dotted lines).

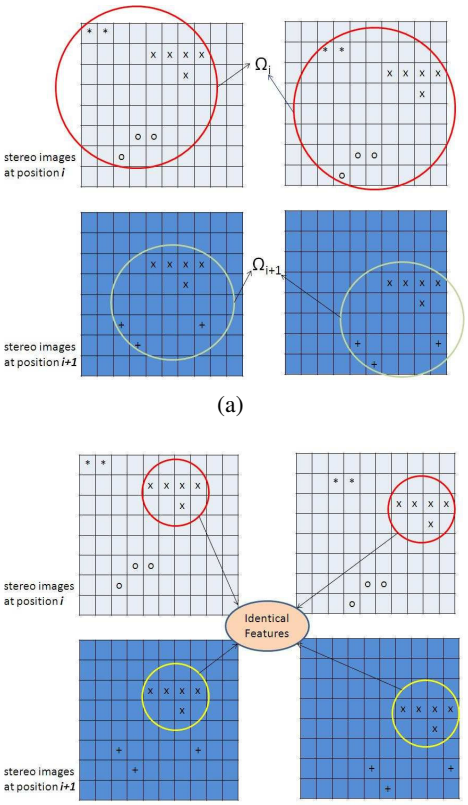


Figure 4: Detecting Identical Features from 2 sets of Matching Features

cameras. This is done by the following three steps of the algorithm.

1) Determining Correspondences in the Image Domain: First, since we are only interested in feature points on the actual object, the background must be subtracted from the foreground using the algorithm presented in [4]. After that, the framework finds matching points between all pairs of stereo images using the SIFT algorithm [3], [11] implemented as a Matlab toolbox[6]. The parameters for this function are adjusted to maximize the number of matching features, but in order to eliminate any possible mismatch by the algorithm, we apply a simple epipolar constraint and a stochastic analysis over the disparity between left and right images to eliminate possible outliers. Next, the framework uses two left images corresponding to two consecutive positions of the camera-object to establish correspondences between these positions. That is, it runs again the SIFT algorithm, but this time using the left image at position *i* and the left image at position *i+1*. The three sets of points – that is, left-right at *i*, left-right at *i+1*, and left-left at *i* and *i+1* – are used to establish the correspondence between feature points in space, as we describe in the next section.

2) Determining Correspondences in Space: Given the matching features for the left and right images, our frameworks also detects matching features between two consecutive positions. As shown in Figure 4a), we use $\Omega_i = \{(\vec{l_i}, \vec{r_i})^j\}_{j=1}^n$ to denote a total number of n matching features at position i, where $\vec{l_i}$ and $\vec{r_i}$ represent the image coordinates of the matching feature on the left and right images. In Figure 4b), the identical features between position i and $i+1$ are defined by $C_{i,i+1} = \Omega_i \wedge \Omega_{i+1}$, where $\wedge$ represents the SIFT matching operation.

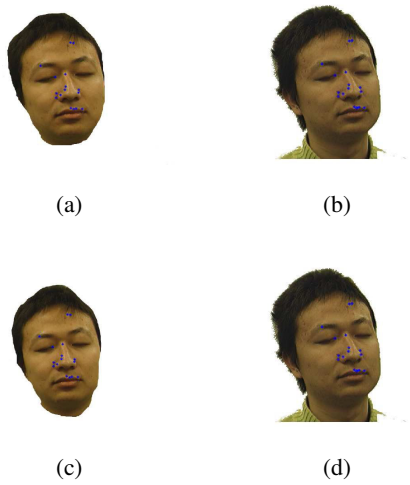
The identical features $C_{i,i+1}$ provide four different sets of image coordinates, as illustrated in Figure 5, which will be used to estimate the motion of the camera. We will explain the details of this step in the next section.

3) Calculating the Pose of Virtual Cameras: The only constraint on the motion of the object is that this be a rigid motion. That is, in this work, we assume that a simple translation and rotation can describe the movement of the feature points from position i and position $i+1$. Given that, and given the sets of 3D points Π_i and Π_{i+1}, the transformation ${}^{L_{i+1}}H_{L_i}$ from position i to position $i+1$ relates these two sets by the following expression:

$$\Pi_{i+1} = {}^{L_{i+1}}H_{L_i} * \Pi_i \tag{1}$$

To determine ${}^{L_{i+1}}H_{L_i}$ we need only 2 pairs of points – each pair provides 3 equations for the total of 6 DOF. However due to noise and camera calibration errors, in practice, ${}^{L_{i+1}}H_{L_i}$ can be better determined through an optimization method using an over specified set of data.

The optimization is done by minimizing the sum of the distances:



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

Figure 5: Identical features detected at different positions of camera-object. a) Left image at position *i*. b) Right image at position *i*. c) Left image at position *i+1*. d) Right image at position *i+1*

$$argmin_{R,t} \sum_{j} ||M_j^{i+1} - [R\,|\,t] * M_j^i|| \qquad (2)$$

where M_j^i and M_j^{i+1} are elements of Π_i and Π_{i+1}, respectively, that is, the 3D coordinates of feature points at positions i and $i+1$. And, R and t represent the rotation and translation component of ${}^{L_{i+1}}H_{L_i}$. In order to find R and t, we used the Levenberg–Marquardt optimization algorithm on the objective function (2).

Since we define one of the positions of the left camera to be the world reference frame, W, the transformation from any virtual position of the left camera to this reference frame can be computed as:

$${}^{L_{i+1}}H_W = {}^{L_{i+1}}H_{L_i} * {}^{L_i}H_W$$

Also, since the cameras are mounted on the same tripod, their relative pose never changes and therefore, any virtual position of the right camera can be easily calculate using the following relation:

$${}^{R_i}H_W = {}^{R_i}H_{L_i} * {}^{L_i}H_W = {}^{R}H_L * {}^{L_i}H_W$$

where ${}^{R}H_L$ is the relative pose between left and right cameras which is obtained off line by the camera calibration.

III. Experimental Results and Discussion

In this section, we present both quantitative and qualitative results from our virtual multi-view 3D modeling framework. As for the qualitative part, we used the algorithm applied to human faces and human body. Quantitatively, we also compared the accuracy in the 3D reconstruction by applying the algorithm to a human skull dataset used as a standard dataset in the original PMVS [5]. In that case, we computed the difference between the two approaches, ours and that in [5], and regarded the PMVS as the ground truth.

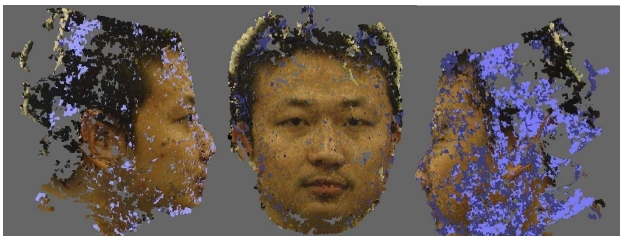


Figure 6: Reconstructed 3D face using 16 images

A. Reconstruction of the human body

In this experiment, the person to be modeled stood in front of the cameras with his right side turned to the cameras. We start the image acquisition at 30fps while the human rotated by 180^0 in front of the cameras. Since the time to complete the 180^0 rotation and the consequent number of images acquired may vary, we subsampled the images by a variable factor that led to a totally of 16 images – 8 images for each camera – or one image roughly every 22.5^0.

We ran the SIFT algorithm to find corresponding feature points, as explained in section II-B1. In average, the SIFT algorithm returned about 150 matching points between left and right images, and a few dozens of other matching points between two consecutive left images. In the end, the algorithm is capable of finding between 15 and 20 points in common for each pair of consecutive positions. Those are the points used to compute the transformation matrix ${}^{L_{i+1}}H_{L_i}$. After running the optimization and obtaining the virtual camera poses for each of the 16 images, the same images and the calculated camera poses were input to the patch-based multi-view program. The outcome of the program for this 3D model of the human face is shown in Figure 6.

B. Increasing the Density of the Model

As we mentioned earlier, one of the major advantages of our method is in how easy it is to increase the density of a 3D model. That is, if an application requires a denser 3D model, all that one needs to do within our framework is to change the sampling factor used in the

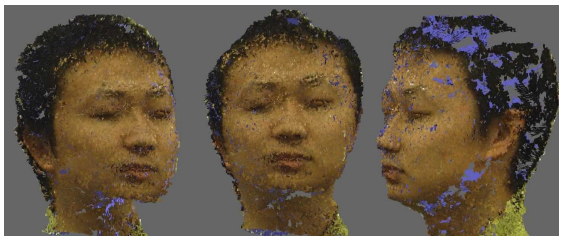


Figure 7: Reconstructed 3D face using 70 images

Figure 8: Reconstructed 3D Human Upper Body

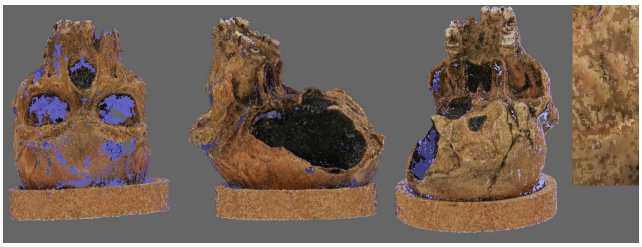
above steps. There is no need to add more calibrated cameras or to calibrate positions of turn tables. As Figure 6 shows, various gaps (blue spots) are present in the 3D model, in special on the head where the low-resolution cameras used and the hair makes it harder to find feature correspondences. To reduce the number of such gaps in the model, we can increase the number of virtual poses of the cameras by simply increasing the number of sampled images after image acquisition.

Figure 7 shows such a model when 70 images were sampled. By comparing the 3D model obtained in Figure 6 and the model in Figure 7, we can see that the second model is qualitatively better than the first. In Figure 8, we show another qualitative result using our method, this time for the human upper body. In this case, we used 14 images taken from another pair of cameras, using a different settings. For example, for this experiment the baseline between the 2 cameras is approximately 2m.

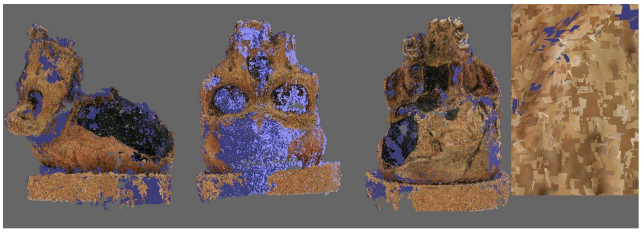
In the next section, we analyze our results in a more quantitative manner.

C. Quantitative Analysis

We performed the first quantitative analysis of our method assuming that the PMVS provided the ground truth. For that, we used the *skull* dataset, which contains 24 calibration matrices, for each of the 24 images taken by 3 cameras and 8 positions of the turntable. In order to test our algorithm, we selected a subset of those images, that is, 16 images among the 24 available – or 2 of the original 3 cameras used by the PMVS. We also selected



(a) skull reconstruction using PMVS



(b) skull reconstruction using proposed approach

Figure 9: Comparison: PMVS and our method

Size of Object (LxWxH in mm)	Error in reconstruction (mm)
$498.8 \times 498.9 \times 612.9$	4.5

Table I: Accuracy of our method with respect to PMVS

the respective camera calibration matrices, but we used only the first pair of camera calibrations and compute all other transformations according to the steps described by our framework. That is, we assumed that no other camera pose or calibration parameters were available and that all necessary data had to be computed from the images and their feature points as already explained. Finally, to generate the so called *ground truth*, we also ran the original PMVS algorithm for the same 16 images. As we can see from Figures 9a and 9b, the two results are quite similar.

Also, in order to measure the difference between the two 3D models, we used the Iterative Closest Point (ICP) [15] to match the 3D cloud of points from our approach to the cloud of points obtained by the original PMVS. As shown in Table I, the average error was only 4.5 (mm), which means that the two 3D models were only 4.5mm different from each other. Compared to the dimensions of object, which were about or greater than 500mm, the error in 3D reconstruction was less than 1% with respect to the original PMVS approach.

Finally, we collected accurate 3D data using a structure laser scanner as presented in [12], [13]. We used two objects, an *angel* and a *bunny*. Figure 10 and Table II summarize the results for those two objects.

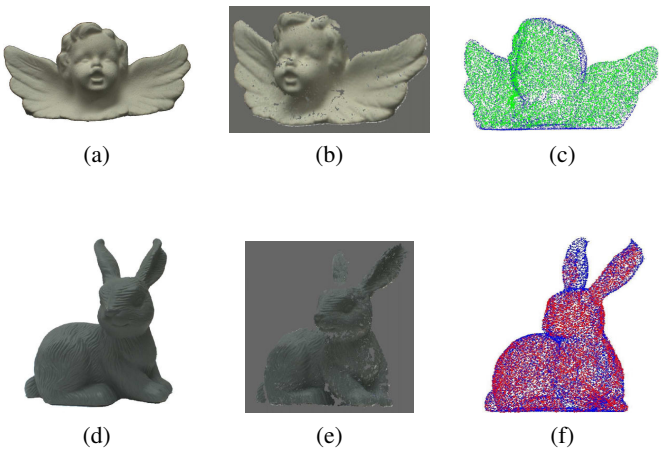


Figure 10: Quantitative Results obtained using two objects: (a) and (d) images of the angel and bunny used for testing; (b) and (e) 3D model created using our method; (c) and (f) the error between ground truth (blue dots) and 3D model obtained by the method (green dots for angel and red dots for bunny)

Object	Number of Views	Error (mm)	Max error	Percentage <1 mm	Percentage <1.5 mm
Bunny	7	.49	5.9	85.5%	93.4%
Angel	12	1.24	11.4	89%	96%

Table II: Accuracy of our method for real objects

IV. Conclusion

This paper presented a novel approach of multi-view 3D modeling using virtual cameras and object-camera pose estimation. This work is a significant extension to the PMVS method as it eliminates the need for multiple camera calibrations or any other special apparatus. Also, as we demonstrated in the result section, our method can be run for different densities of the 3D model, without any change in the setup of the cameras and/or their calibration. In the future, we intend to integrate the feature matching using SIFT and the relative motion estimation between camera and object into the optimization step in the PMVS algorithm. That change should further improve the performance and computational complexity of the proposed method.

References

[1] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *International Conference on Robotics and Automation*, May 2008, pp. 292–298.

[2] J.-Y. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/.

[3] M. Brown and D. G. Lowe, "Invariant features from interest point groups." in *British Machine Vision Conference*, 2002, pp. 656–665, cardiff, Wales.

[4] Y. Dong and G. N. DeSouza, "Adaptive learning of multi-subspace for foreground detection under illumination changes," *submitted to Journal of Computer Vision and Image Understanding*, 2008.

[5] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," in *Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[6] A. Fusiello and L. Irsara, "Quasi-euclidean uncalibrated epipolar rectification," in *International Conference on Pattern Recognition*, 2008, pp. 1–4.

[7] A. C. Halpern, "The use of whole body photography in a pigmented lesion clinic," in *http://www.ncbi.nlm.nih.gov/pubmed/11134998*, 2000.

[8] Y. Iwashita, R. Kurazume, R. Kurazume, S. Uchida, K. Morooka, T. Hasegawa, *et al.*, "Fast 3d reconstruction of human shape and motion tracking by parallel fast level set method," in *International Conference on Robotics and Automation*, May 2008, pp. 980–986.

[9] W. S. Kim, "Computer vision assisted virtual reality calibration," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 15, pp. 450–464, 1999.

[10] M. Levoy, S. Rusinkiewicz, M. Ginzton, J. Ginsberg, K. Pulli, D. Koller, S. Anderson, J. Shade, B. Curless, L. Pereira, J. Davis, D. Fulk, *et al.*, "The digital michelangelo project: 3d scanning of large statues," in *SIGGRAPH*, 2000.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] J. Park and G. N. DeSouza, *Photorealistic Modeling of Three Dimensional Objects Using Range and Reflectance Data, in Studies in Computational Intelligence, Vol. 7, Machine Learning and Robot Perception.* Springer-Verlag, 2005.

[13] J. Park, G. N. DeSouza, and A. C. Kak, "Dual-beam structured-light scanning for 3-d object modeling," in *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, May-Jun 2001, pp. 65–72, quebec City, Que., Canada.

[14] B. Reitinger, C. Zach, and D. Schmalstieg, "Augmented reality scouting for interactive 3d reconstruction," in *Virtual Reality Conference*, 2007.

[15] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings of the Third Intern. Conf. on 3-D Digital Imaging and Modeling*, vol. 1, May 2001, pp. 145–152, quebec City, Canada.

[16] A. Santhanam, T. Willoughby, I. Kaya, A. Shah, S. Meeks, J. Rolland, and P. Kupelian, "A display framework for visualizing real-time 3d lung tumor radiotherapy," *Journal of Display Technology,*, vol. 4, Dec 2008.

[17] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, *et al.*, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR*, vol. 1, 2006, pp. 519–526.

[18] M. Tomono, "3-d object map building using dense object models with sift-based recognition features," in *Intelligent Robots and Systems*, Oct 2006, pp. 1885–1890.

[19] T. Weise, B. Leibe, and L. V. Gool, "Fast 3d scanning with automatic motion compensation," in *CVPR*, 2007, pp. 1–8.

[20] Y. Zheng, J. Chang, Z. Zheng, Z. Wang, *et al.*, "3d face reconstruction from stereo: A model based approach," in *ICIP*, vol. 3, 2007, pp. 65–68.