Depth-Color Image Registration for 3D Surface Texture Construction using Kinect Camera System

Semih Dinç[1], Madhav Sigdel[1], İmren Dinç[1], Madhu S. Sigdel[1], Farbod Fahimi[2], Ramazan S. Aygün[1]
DataMedia Research Lab, Computer Science Department[1]
Mechanical and Aerospace Engineering Department[2]
University of Alabama in Huntsville
Huntsville, Alabama 35899
Email: {sd0016, ms0023, id0002, mss0025, ff0002, aygunr} @uah.edu

***Abstract*—Today some camera systems provide various opportunities to the scientists in computer vision since they capture color and depth images of a scene simultaneously. This paper presents a new 3D model construction and surface texture mapping technique for real object images captured by Microsoft (MS) Kinect camera system. Our ultimate goal is to construct textured 3D model of the real objects. To achieve this goal, we perform depth-color image registration based on "Scale Invariant Feature Transform" (SIFT) and "Speeded-Up Robust Features" (SURF) features, Hough transform, and least squares optimization. After the registration, using the depth image, non-textured 3D model is created and finally color image is mapped on top of the 3D model of the object based on Delaunay triangulation. In the experiment section, three objects having different sizes and shapes are examined. Their textured 3D models are constructed without significant problems.**

I. Introduction

In the last two decades, digital image processing technology has improved significantly. Constructing synthetic 3D models of real or artificial objects is one of the important applications used in many areas such as medical imaging, target detection/tracking, architecture, engineering, virtual reality, etc. Typically, 3D models of artificial objects are designed in computer environments. Points in 3D space are connected with lines which later form the "polygonal mesh". Surfaces of the mesh are mapped by simple color, texture, or image patches.

Recent improvements enable to construct 3D models of almost any real object. Some specialized camera systems provide users some specific libraries or APIs in order to sense depth information of the objects in the field of vision. The depth data can be stored as a set of points in 3D space which is also called as "point cloud". Then 3D views of the scene are constructed by generating the polygonal mesh using the point cloud. However, most of the tools today, can only generate non-textured 3D models. After construction, surfaces are covered by simple color or some artificial texture, but not by the real texture of the object. Although there are various studies [1] that map artificial textures to the 3D models properly, there is only a few studies that mainly focus on mapping real object texture [2]. Actually, it is a challenging problem to overlap 3D model and the texture properly because mapping 2D image on top of a 3D model requires attention. Some studies uses the idea of "unfolding" [3]. According to this approach, 3D model is unfolded into 2D space and then overlapping two 2D images becomes easier.

An alternative idea is "correspondence based mapping". In this approach, the goal is to extract the correspondence between 3D model and 2D image. Using this relation, regions of the 3D model and patches of the image can be matched properly. This paper follows this idea to accomplish texture mapping operation. To achieve this goal, we benefit from the capabilities of MS Kinect Camera. Kinect is a very well-known and widely used [4] [5] camera system that can capture depth and color images of the scene simultaneously. In order to have accurate correspondence, "depth-color image registration" is required for further steps.

Image registration is a useful solution that has various application areas such as medical imaging [6] [7], geographic information systems (GIS) [8], and mosaic generation [9]. In simple terms, registration is to align multiple images based on the common features of the images [10]. Distinctive points in the image are better candidates to these features. Therefore, we use two well-known feature extraction methods, SIFT and SURF, to extract keypoints from depth and color images and we match features in both images. However, depth-color registration is not a typical way of image registration since depth and color images are in different color spaces (or domains) and the pixel intensities hold different meanings. Thus, we first convert the images into a binary format to be able to compare both image types.

After the registration, we generate the non-textured 3D model of the object using a stable and robust technique called "Delunay Triangulation" [11]. Then we map the color patches to triangular surfaces of the 3D model. And finally, we obtain a fully textured 3D model of the object. Please note that, in this study, we focus on constructing a textured 3D model of the object from a single pair of color and depth images, not the complete 3D model of the object.

Following sections of the paper are organized as follows. In Section 2, a brief overview of the Kinect Camera System is provided. The properties of its sensors and their capabilities are described. In Section 3, the underlying reasons of employing the image registration stage are mentioned. Then, two well-known feature extraction algorithms, SIFT

and SURF, are described briefly. In Section 4, our image registration technique is presented which is based on SIFT and SURF features matching, Hough transform, and least squares optimization. Finally in Section 5, our texture mapping approach is introduced and then, our experiments are presented for three objects having different sizes and shapes. In the conclusion section, we summarize our work and describe our future work related to this study.

II. Overview of the Kinect System

This section provides a brief explanation of the MS Kinect camera system. MS Kinect is a special sensing device by Microsoft that combines different image sensors in order to provide sophisticated visual information to the user. Although the main purpose of this device is entertainment, there are many other application areas. In the academic literature, target detection, target tracking, and object recognition may be considered as possible problems that can benefit from the Kinect System [4]. Microsoft provides an SDK for the Kinect system which includes the libraries of some basic operations such as capturing color and depth image. The SDK includes sophisticated functions such as 3D scene construction and human face/body detection. On the other hand, it has very primitive and unstable version of texture mapping, which is not applicable to our study. Also, the source code is not provided.

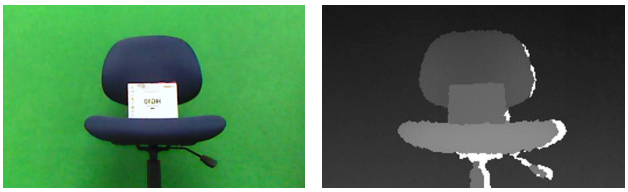
In our experiments, we use the XBOX version of the Kinect shown in Fig. 1. It has a pair of sensors for depth perception and a single camera for conventional RGB (Color) images. A microphone is also included into the system for sound recognition purposes. By using these features Kinect can capture color image, depth image, and sound.

Fig. 1. Kinect System and Sensors

The Kinect sensor we use in our research can capture images with various resolutions. The maximum resolution for the color image is 960×1280 and the depth image is 480×640. Due to computational purposes, we decided to use 480×640 resolution for both color and depth images. A typical color image and a depth image are presented in Fig. 2. In the depth image, the intensity of a pixel corresponds to the distance to the camera. It means if the intensity of a pixel is high then it is close to the camera, otherwise it is far from the camera.

III. SIFT and SURF Feature Extraction for Image Registration

Kinect Camera can capture color and depth images simultaneously. We can normally expect that if the resolutions are the same then the images can be overlapped perfectly without registration. But in practice, this is not the case. As shown in Fig. 3 a raw match of the color and depth images are not aligned properly. There is a significant difference in size and some difference in position. Since the difference takes place in the 2D image space, an affine transformation with 3 parameters is sufficient between two images. This transformation basically includes 3 parameters: scale, X translation and Y translation. Therefore, a registration operation is necessary to be able use the same pixel coordinates of the object in both color and depth images.



(a) (b)

Fig. 2. Typical Example of Color and Depth Images

Fig. 3. Overlapping Original Color and Depth Images

In this paper, we propose a generic image registration method that can be utilized in any image registration problem. Following parts of this section includes brief definitions of two well-known feature extraction methods, SIFT and SURF. Later, in the next section, proposed registration technique is described in detail.

A. SIFT Features

Scale Invariant Feature Transform (SIFT) [12] is a very powerful technique for extraction of distinctive invariant feature points and feature descriptors from an image. SIFT feature descriptors are shown to be invariant to common image deformations such as scale, rotation, affine transformation, changes in viewpoint, and slight illumination changes. Because of these properties, SIFT features are very well suited to use for image matching problems. The algorithm consists of four major stages. Firstly, an image is convolved with a Difference-of-Gaussian (DoG) function across all possible scales. Potential interest points are then identified by searching

stable features across all scales using scale-space extrema in the convolved image. In the next step, candidate keypoints are localized and unstable keypoints are eliminated. In the third stage, a consistent orientation is assigned to each keypoint based on the local image gradient. Finally, descriptor vectors are obtained based on the scale, orientation, and gradient magnitudes in a neighborhood around the keypoints.

B. SURF Features

Speeded-Up Robust Features (SURF) [13] is an efficient solution for extraction of scale and rotation invariant interest point detectors and descriptors in an image. SURF is said to provide repeatable, distinct and robust features at a much lower computational complexity compared to other approaches. It has been widely used in several applications including image registration and object detection in computer vision. The algorithm makes use of integral images for efficient feature computation. SURF consists of three main stages. Firstly, interest points are determined by calculating the determinant of Hessian matrix that approximates the Gaussian convolution with box filters. In the next step, the dominant orientation for an interest point is determined on the basis of local gradient orientation distribution around its neighborhood. The orientation distribution is estimated with Haar wavelets. This is useful to obtain rotationally invariant features. At the third stage, feature descriptors are constructed by considering square windows around the interest points and concatenating the histograms consisting responses of the Haar wavelets. Once the feature descriptors are available, two features can be matched by comparing the corresponding feature descriptors.

IV. Image Registration

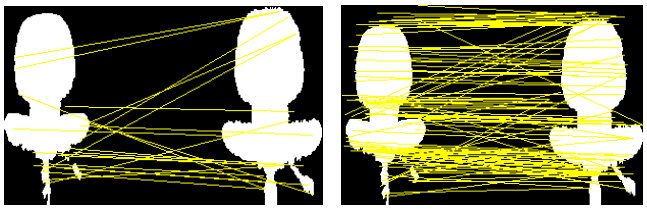
Image registration is one of the key steps of our solution, since we want color and depth images to be properly aligned. In this way, we can find the correspondence between the texture regions and the 3D object model. Our registration technique is based on detecting the corresponding image pixel coordinates of two images. And later, using these feature pairs, least squares method is employed to calculate the 3 affine transformation parameters; scale, X translation and Y translation.

A. Extracting and Matching Feature Points

SIFT and SURF operators are employed to detect feature points in color and depth images. Before applying these operators, some basic preprocessing operations are done, since the original color image has 3 color bands and the depth image has only 1 band. Gray scale conversion is not enough since the pixel intensities of the two images express completely different meanings. For that reason, our problem can not be considered as a typical image registration problem. In a typical example, the goal would be to register multiple images in the same color space or captured by the same camera in different times or places. Characteristics of the same keypoints in the images would be similar and matching them would be possible. However, in this study, color and depth images are not comparable as the original forms. Please see the Fig. 2 for an example.

To solve this problem we convert images into binary form in which both images can be comparable. First, the background is removed from the images using the advantage of the green screen which is very practical environment to separate objects from the background. Then the images are thresholded to obtain binary form of the objects. Finally, SIFT and SURF operators are applied to the binary images.

In our experiments, we have tried two operators separately and compared their accuracy on matching features. After the comparison, we realized that in all cases SURF extracts more reliable features than SIFT for this specific problem. Our experiments show that approximately 75% of the SURF matches are useful while only 50% of the SIFT matches are useful. Based on these results, SURF algorithm is employed to generate and match features for the rest of the study. Fig. 4 shows a sample of matching results of SIFT and SURF operators, respectively.



(a) (b)

Fig. 4. Feature Matching Using (a)SIFT and (b)SURF

As can be seen in Fig. 4, both operators can extract some useful features and correct matches; however, there are also incorrect matches. This may cause a serious problem (irrelevant transformation) since the method we use for parameter calculation is sensitive to the incorrect keypoint matches.

B. Calculating the Affine Transformation Parameters

1) Least Squares Method: In order to calculate 3 unknown transformation parameters, we employ an optimization method based on "least squares" approach. Since there is an affine transform between 2 images, we can simply construct the transformation equation as,

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} s & 1 & t_x \\ 1 & s & t_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

where s is the scale parameter and t_x, t_y are the translation parameters for X and Y dimensions, respectively. x and y are the pixel coordinates in the origin image space. $\tilde{x}$ and $\tilde{y}$ are the pixel coordinates in the destination image space. Since there are 3 unknowns, there must be at least 3 independent equations to calculate these parameters. One pixel match provides 2 equations for X and Y dimensions. Therefore, if we find at least 2 correct matches between two images, we can calculate the transformation parameters. In order to get benefit from the

least squares approach, we rewrite (1) in the form of $W = QT$ as presented in (2),

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{x}_2 \\ \tilde{y}_2 \end{bmatrix} = \begin{bmatrix} x_1 & 1 & 0 \\ y_1 & 0 & 1 \\ x_2 & 1 & 0 \\ y_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ t_x \\ t_y \end{bmatrix} \tag{2}$$

where the left-hand side of the equation represents W and the right-hand side of the equation represents Q and T, respectively. In this equation, our goal is to calculate T, the vector of unknown parameters. Using the least squares method we can solve the unknowns as,

$$T = (Q^T Q)^{-1}(Q^T W) \tag{3}$$

By (3), we can use more than two matching pixels if possible. This is convenient as it may improve the accuracy at the optimization stage.

2) Finding Best Transformation Parameters: As we mention in Section 4.1, some incorrect matches are also produced by SIFT and SURF operators. When there is an incorrect match in (2), incorrect transformation parameters may be obtained. This problem may cause a failure in the entire system. To minimize this effect, we use a solution based on Hough transform [14]. We assume there are more correct matches than incorrect ones between two images. Please note that our experiments show this assumption is true in almost all cases. Suppose there are N matches produced by SIFT or SURF operators. We first take a list of all $\binom{N}{2}$ combinations of the matches. Later, for each iteration, we take 2 matches and calculate the transformation parameters using only those 2 matches. The result is a 3 dimensional vector which can also be represented as a point in 3D space. All other iterations are done in similar manner. Finally we have a set of points in 3D space and each point represents a transformation vector of 3 unknowns. Fig. 5 shows the results of a sample case.

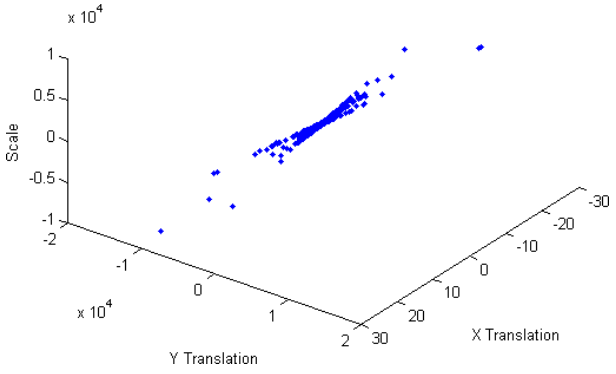


Fig. 5. Hough Transformation Result

We can infer from the Fig. 5 that there is a dense area where most points are located. Center coordinates of this dense area represent the best transformation parameters between two images. That is why, in this point set, we select the median point which is presumably very close to the center coordinates. We use that point as the final transformation parameters.

V. Experiments & Results

After image registration is completed, we obtain the 3D point cloud from the depth image and create triangles using the 3D points to have surfaces of the object. Then, corresponding color image patch is selected and mapped on each triangle. In this way, a complete textured model is obtained when all triangles are mapped with a color patch. For this purpose we employ some additional algorithms to utilize this set of operations. At the first stage, a depth image is converted to a 3D point cloud. Please note that the intensity values of the pixels refer to depth information, which can also be considered as the third dimension. Every pixel corresponds to a 3D point in the point cloud. Please see the subfigures (b) and (e) in Figures 6, 7, and 8.

Noise Removal. In the point cloud, there are some noise pixels that do not hold any useful depth information. We employ DBSCAN algorithm [15] to find out these outlier pixels and remove them from the point cloud. DBSCAN is originally a data clustering algorithm based on the density around a sample. According to the algorithm, samples without any neighbors can be labeled as outliers. This idea is also beneficial for our problem because our goal is to keep the dense area and continuous regions in the point cloud.

Delaunay Triangulation. In the next stage, triangulation operation is applied to the point cloud and hundreds of triangles are created in order to have a complete 3D model of the object. Delaunay triangulation method [11] is selected at this stage. It is a well-known and robust method to create triangles in n-dimensional point cloud. Delaunay triangulation is a very powerful approach but it can still create some incorrect triangles that belong to the background, not the object. In order to eliminate these triangles, we search the center point of each triangle and check whether it is on the object or not. If it is not, we remove that triangle from the 3D mesh. In Figures 6, 7, and 8; (f) and (g) show the triangulated 3D models. In subfigures (f), there are some extra triangles that do not belong to the 3D model of the object. In (g), these incorrect triangles are eliminated.

Texture Mapping. In the final stage, all the remaining triangles in the 3D mesh are matched with a texture patch in the color image. Each patch is mapped on top of the corresponding triangle. Subfigures (h) show the final textured 3D model of the object that is generated from a single image. Since we use patches, our textured models are accurate and clear that there is not a significant distortion on the texture.

Experiments We have made 3 sets of experiments for different objects that are in various sizes and shapes. In the first experiment, a "chair and box" image is captured. In the image, there is a blue chair and a box placed on it. Chair can be considered as a large size object with medium complexity in shape. Fig. 6 shows the steps of the model generation. In Fig. 6 (d), there are white regions which seem not properly registered. Please note that those regions are problematic regions that cannot be properly captured by Kinect System. Since there are two cameras employed for the depth perception, there may be

some regions that are not in the common field of view of both cameras. Therefore, Kinect cannot capture information from those regions.

Fig. 6 (h) shows the final result of the experiment. Note that 3D model is rotated to show the issues. It is clearly visible that a large portion of the model is correctly generated. However, some parts of the model such as top of the box and the seat of the chair may have problems. The reason is that there are not color image patches for these parts of the model. Instead of leaving blank, those parts are mapped with the texture that is created by interpolation of surrounding patches.

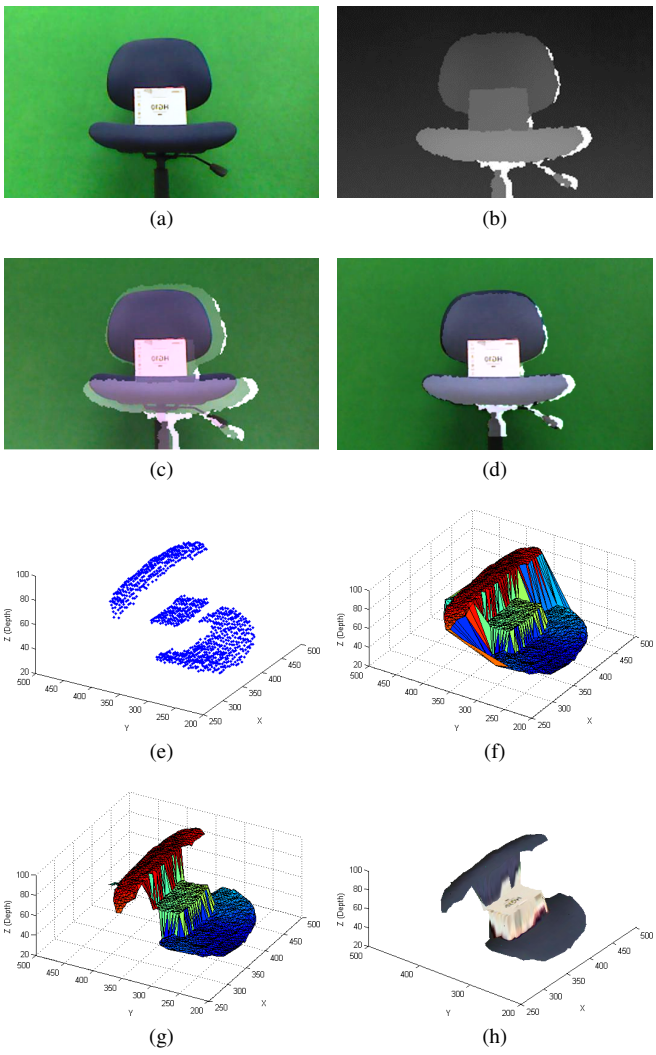


Fig. 6. "Chair and Box" Experiment, (a)color image, (b)depth image, (c)raw overlapped image, (d)overlapped image after registration, (e)point cloud, (f)non-textured 3D model 1, (g)non-textured 3D model 2,(h)textured 3D model

In the second experiment, we capture the images of a laptop computer. It is a medium size object with relatively less complex shape. The important point of this object is that there is a flower figure on the screen. We want keep this flower image without distortion in the textured 3D model. Although there are some minor problems, our system can generate an acceptable model of the laptop. Fig. 7 shows the construction steps of textured 3D model. According to the final image in Fig. 7 (h), the content on the screen is visible and clear.

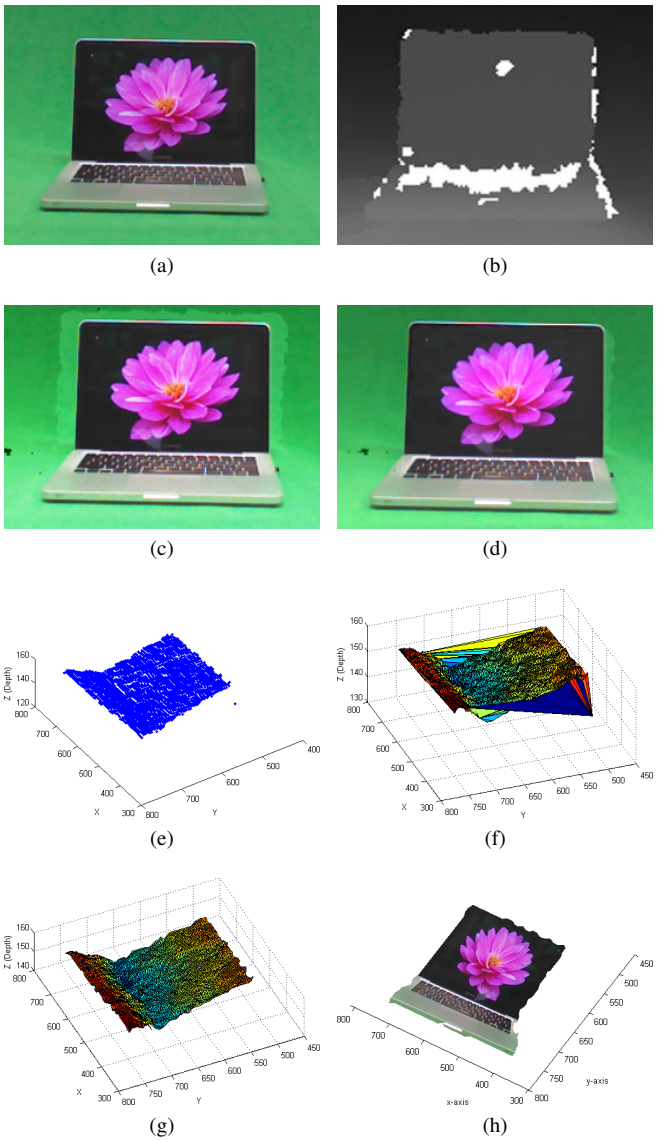


Fig. 7. "Laptop" Experiment, (a)color image, (b)depth image, (c)raw overlapped image, (d)overlapped image after registration, (e)point cloud, (f)non-textured 3D model 1, (g)non-textured 3D model 2,(h)textured 3D model

In the final experiment, we select two small objects, a box and a book that is placed on it. In this experiment we target to preserve the text on object surface. Fig. 8 represents the steps of the textured 3D model construction. As can be seen in Fig. 8 (h), the texture of the box and the book is accurate and the text on the book is readable.

VI. Conclusion & Future Work

We propose an alternative solution to the 3D texture mapping problem using a single view of color and depth images of the Kinect Camera System. In the paper, first, color and depth images are registered using combination of several techniques.

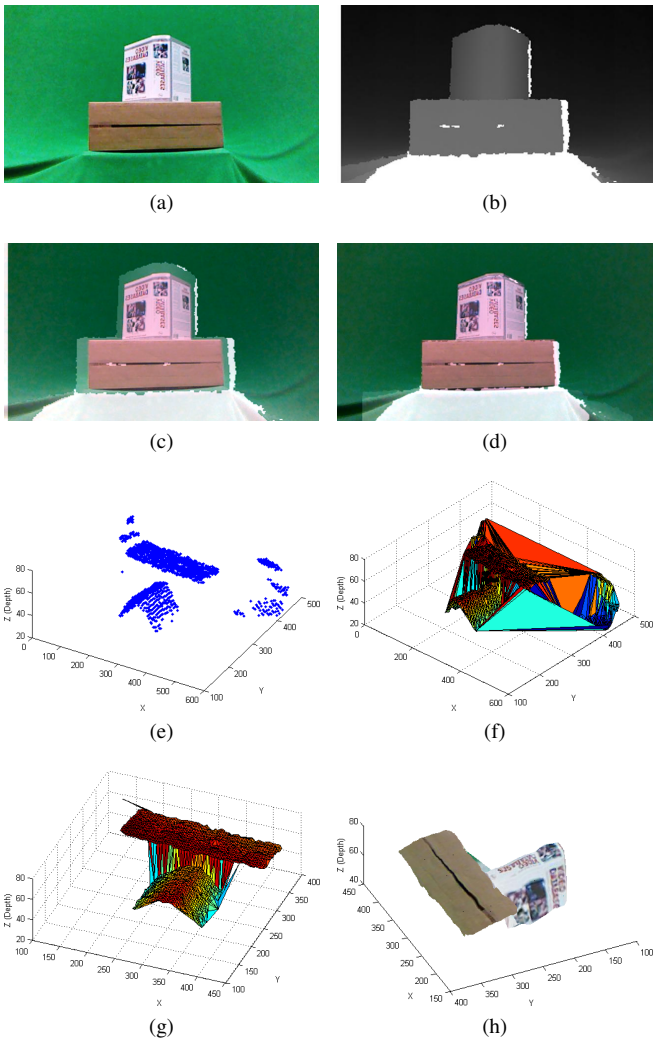


Fig. 8. "Book" Experiment, (a)color image, (b)depth image, (c)raw overlapped image, (d)overlapped image after registration, (e)point cloud, (f)non-textured 3D model 1, (g)non-textured 3D model 2,(h)textured 3D model

SIFT and SURF algorithms are employed to extract matching features of both images and then Hough transform is applied to eliminate incorrect feature matchings. Using the remaining features, the transformation between two image spaces is calculated. After the registration, texture mapping is performed. Non-textured 3D model is generated using Delaunay Triangulation method based on the depth image. Since color and depth images have been properly registered, surface triangles and corresponding color patches are matched to construct textured 3D model. We have performed 3 experiments using the images of different objects with different sizes and shapes. Although there are some minor problems, we obtain promising results.

This study is the first stage of an ongoing research. Our final goal is to develop a robust target detection system that uses complete 3D model of any target as the prior information. Therefore, the 3D model of the target must be constructed with its color texture. In our following studies, we plan to improve our current system by constructing a complete 3D model from multiple views and map the corresponding texture to that model. The results of this paper show that the first step of our goal provides promising results for further steps.

REFERENCES

[1] J. Maillot, H. Yahia, and A. Verroust, "Interactive texture mapping," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '93. New York, NY, USA: ACM, 1993, pp. 27–34.

[2] K. Matsushita and T. Kaneko, "Efficient and handy texture mapping on 3d surfaces." *Comput. Graph. Forum*, vol. 18, no. 3, pp. 349–358, 1999.

[3] E. Zhang, K. Mischaikow, and G. Turk, "Feature-based surface parameterization and texture mapping," *ACM Trans. Graph.*, vol. 24, no. 1, pp. 1–27, Jan. 2005.

[4] 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 *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568.

[5] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, 2011, pp. 15–22.

[6] M. Jenkinson, P. Bannister, M. Brady, and S. Smith, "Improved optimization for the robust and accurate linear registration and motion correction of brain images," *NeuroImage*, vol. 17, no. 2, pp. 825 – 841, 2002.

[7] S. Damas, O. Cordon, and J. Santamaria, "Medical image registration using evolutionary computation: An experimental survey," *Computational Intelligence Magazine, IEEE*, vol. 6, no. 4, pp. 26–42, 2011.

[8] X. Dai and S. Khorram, "A feature-based image registration algorithm using improved chain-code representation combined with invariant moments," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 5, pp. 2351–2362, 1999.

[9] Y. Chen and R. S. Aygun, "Spritecam: virtual camera control using sprite," *Multimedia Tools and Applications*, pp. 1–23, 2013.

[10] M. Hossain, S. W. Teng, and G. Lu, "Achieving high multi-modal registration performance using simplified hough-transform with improved symmetric-sift," in *Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on*, 2012, pp. 1–7.

[11] N. Amenta, M. Bern, and D. Eppstein, "The crust and the beta-skeleton: Combinatorial curve reconstruction," in *Graphical Models and Image Processing*, 1998, pp. 125–135.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[14] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer vision, graphics, and image processing*, vol. 44, no. 1, pp. 87–116, 1988.

[15] M. Daszykowski, B. Walczak, and D. Massart, "Looking for natural patterns in data: Part 1. density-based approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 56, no. 2, pp. 83 – 92, 2001.