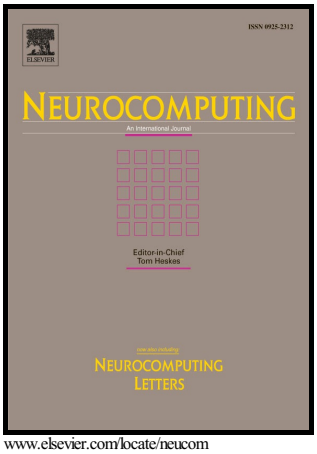


www.elsevier.com/locate/neucom

Dense 3D Reconstruction Combining Depth and RGB Information

Hailong Pan, Tao Guan, Yawei Luo, Liya Duan, Yuan Tian, Liu Yi, Yizhu Zhao, Junqing Yu

PII: S0925-2312(15)01580-5
DOI: http://dx.doi.org/10.1016/j.neucom.2015.10.104
Reference: NEUCOM16282

To appear in: *Neurocomputing*

Received date: 13 July 2015
Revised date: 10 October 2015
Accepted date: 28 October 2015

Cite this article as: Hailong Pan, Tao Guan, Yawei Luo, Liya Duan, Yuan Tian, Liu Yi, Yizhu Zhao and Junqing Yu, Dense 3D Reconstruction Combining Depth and RGB Information, *Neurocomputing* http://dx.doi.org/10.1016/j.neucom.2015.10.104

Dense 3D Reconstruction Combining Depth and RGB Information

Hailong Pan[a], Tao Guan[a,*], Yawei Luo[a], Liya Duan[c], Yuan Tian[b],
Liu Yi[a], Yizhu Zhao[a], Junqing Yu[a],

[a]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

[b]School of Educational Information Technology, Central China Normal University, Wuhan 430079, China

[c]Institute of Oceanographic Instrumentation, Shandong Academy of Sciences, Qingdao, 266001, China

[*]Corresponding author: Tao Guan, E-mail address: qd_gt@126.com, tel number: +8602787556154

Abstract

Dense 3D reconstruction has important applications in many fields. The existing depth information based methods are typically constrained in their effective camera-object distance which should be from 0.4 meters to 4 meters. We present a novel method that can achieve a more accurate dense 3D reconstruction with an RGB-D camera when the distance between the camera and object is less than 0.4 meters, which enlarges the application range. Our approach combines a depth information based 3D model method with a RGB information based method to refine the reconstruction results when the camera fails to acquire the correct depth information. Rich RGB information captured from a color camera along with feature detection and triangulation methods are used to obtain accurate camera poses and 3D points when the camera is close to the object. Compared with the reconstruction results obtained from depth information only, quantitative experimental results show that our method is more effective, particularly when the camera is close to the object in the scene.

Keywords: 3D reconstruction; depth information; RGB information

1. Introduction

Dense 3D reconstruction has many applications in object recognition [4], [20], [34], [42] and [45], object retrieval [23], scene understanding [16], [29] and [40], object tracking [13], autonomous navigation [10-12], human-computer interaction [24], telepresence, telesurgery, reverse engineering, virtual maintenance and visualization [1]. According to the different inputs to the system, prevalent techniques used to reconstruct different kinds of environments are divided into 3D reconstruction methods using color cameras only and methods using RGB-D cameras (color and depth). RGB-D cameras, which are sensing systems equipped with an RGB camera, an infrared projector and an infrared sensor, can gather RGB information and the depth map simultaneously. Particularly with its affordable price, RGB-D cameras are becoming more popular and a better choice for generating 3D information. Moreover, the reconstruction results from the depth information only will not be influenced by the lighting conditions, even in the complete darkness. Therefore, some existing 3D reconstruction methods build the 3D models by using RGB-D cameras. Unfortunately, the RGB-D camera has its depth range limitation from 0.4 meters in near mode to 4 meters in standard mode [7]. If the distance between the RGB-D camera and the

object is less than 0.4 meters, it will fail to get the depth information and lead to poor reconstruction results, as well as significantly narrow the application range.

In this paper, we establish an overall framework of dense 3D reconstruction with an RGB-D camera and take advantage of RGB information to refine the reconstruction results to overcome the above mentioned problem. Rich RGB information can be captured from a color camera, meanwhile, the feature detection and matching are more precise when the camera is closer to the object. Our implementation procedure includes two stages. In the first stage, we build the 3D models using depth information only. First, a depth map is created and converted to a 3D point cloud. Then, the iterative closest point algorithm is implemented to align the new point cloud to the model. Finally, the surface is reconstructed by using truncated signed distance functions. However, the reconstruction result is not satisfactory due to the many "holes" from a failure to acquire the depth information when objects in the scene are close to the camera. In the second stage, RGB information is used to refine the reconstruction results. We first detect and match features and then triangulate the matches. Second, bundle adjustment is implemented to obtain accurate camera poses and 3D points. Finally, the 3D model is aligned to the 3D model built in the first stage. We demonstrate several reconstruction results to prove the efficiency of this method. The accuracy of the reconstruction results is satisfactory. Additionally, the results demonstrate that the proposed approach is able to overcome the problem described above and has no limits to its effective minimum distance.

The remainder of this paper is organized as follows. Section 2 describes related work and our contributions. An overview of the proposed method is provided in Section 3. Section 4 presents the 3D reconstruction method using depth information, and Section 5 describes the approach used to refine the reconstructed 3D models using RGB information. Section 6 demonstrates how to align the new model to the existing model. Experimental results are presented in Section 7, and concluding remarks are provided in the last section.

2. Related Work and our Contributions

The goal of dense 3D reconstruction is to build a complete 3D model of complex scenes. Most existing state-of-the-art 3D reconstruction methods consist of the following two steps: generating a 3D point cloud and then computing a mesh to represent the scene. 3D reconstruction methods using depth information and 3D reconstruction using RGB information are two kinds of popular techniques existing for building 3D models.

2.1 3D reconstruction using RGB-D cameras

RGB-D-camera-based approaches that capture RGB images along with per-pixel depth information can obtain good 3D reconstruction results. Many researches are based on the low priced Kinect-style camera which provides the required data in real time to reconstruct the 3D models. Henry et al. [26] presented a novel 3D modeling method for indoor environments by combining visual features and shape-based alignment. The authors evaluated their overall system by applying it to model two large indoor environments. The experimental results indicated that the system can accurately generate dense 3D maps using RGB-D and is capable of handling situations such as featureless corridors and completely dark rooms. However, the non-real-time RGB-D mapping and its limited global alignment process are two primary disadvantages of the system. Lieberknecht et al. [37] proposed a real-time reconstruction method using parallel tracking and

meshing of unknown environments. The method was tested on different augmented reality scenes and scenarios to prove that it made the augmentations more real by precisely handling occlusions. Izadi et al. [36] presented a real-time 3D reconstruction and interaction system using a moving standard Kinect. Their contributions were threefold. First, their system achieved real-time surface reconstruction using novel extensions to the core GPU (Graphics Processing Unit) pipeline. Second, they made forms of augmented reality more real by correctly managing occlusions. Third, they realized the real-time multi-touch interactions on any indoor scene for the first time. Hu et al. [17] proposed a robust heuristic-switching-based algorithm for simultaneous localization and mapping (SLAM). Their method was proved to be effective in wide area and structurally changing environments. Henry et al. [27] utilized an effective, novel joint optimization algorithm that combined visual features and shape-based alignment. Recently, Yang et al. [48] proposed a bundled-optimization scheme to process the thorough chain from capture to multiview dense depth map generation for the 3D applications. This scheme detected and removed sensor noises through a frequency-counting based non-linear filter.

However, the existing approaches suffer from the same RGB-D camera performance problems. First, the effective distance of RGB-D cameras is limited to a small range from 0.4 meters to 4 meters. If the distance from the RGB-D camera to the object is less than 0.4 meters or more than 4 meters, the reconstruction results will be unsatisfactory. Second, depth estimates are noisy, and the field of view is narrow, with horizontal and vertical view angles limited to less than 60 degrees and 50 degrees respectively. Third, depth images contain numerous "holes" caused by areas does not reflect infrared light.

2.2 3D reconstruction using color cameras

The structure from motion (SfM) technique is a successful reconstruction method with RGB information [14], [21], [35] and [39]. Most SfM systems for unordered images collections are incremental, starting with a few images, repeatedly matching features between two images, adding matched images, triangulating feature matches and performing optimizations to refine camera poses. Such incremental approaches, although quite successful, are unsuitable for large image collections due to the large computation time. The computation time of incremental SfM is commonly known to be $O(n^4)$, where n is the number of images. Crandall et al. [9] proposed a new approach of which the computation time requires $O(n^3)$ in the worst case. Unlike in previous methods, they initialized all cameras at once using a hybrid discrete-continuous optimization on an MRF rather than incrementally building a solution. Therefore, the system could avoid solving sequences of ever larger bundle adjustment problems. This approach was proved to be significantly faster than incremental SfM practically and asymptotically. Wu [6] introduced a novel bundle adjustment strategy that provides a good balance between speed and accuracy. They proposed a preemptive matching method that can save a large amount of feature matching time by identifying good image pairs robustly and efficiently. It was proved that the time complexity of their method is $O(n^2)$ for n input images, and the time complexity of the major steps, including feature matching, bundle adjustment and point filtering, is only $O(n)$. The authors' method maintains high reconstruction accuracy by re-triangulating the failed matches as they appear.

However, the 3D reconstruction methods using color cameras are sensitive to lighting conditions because they depend on RGB. Thus, if the lighting conditions are poor (e.g., in a dark

room), the reconstruction will fail. Additionally, occlusion between objects cannot be managed in dynamically changing scenes, which will significantly narrow the range of applications in which human interaction is required.

2.3 Our contributions

The proposed method distinguishes itself in the following ways:

First, to obtain a satisfactory 3D model of a real scene when the distance between the RGB-D camera and the object is less than the limiting value (0.4 meters), we design a two-step strategy that combines depth information and RGB information. This method takes advantage of both depth information and RGB information and can thus yield better reconstruction results.

Second, to accelerate the processing, we use a point-to-plane error metric instead of a point-to-point metric for aligning a new point cloud to the existing 3D model and use a GPU/CPU mixed implementation to find the best option for each step for reconstructing the 3D model with RGB information.

Third, depth information cannot be obtained when an object in the scene cannot reflect infrared light. In such cases, this method can fill the "holes" created by these non-reflecting objects automatically.

3. System Overview

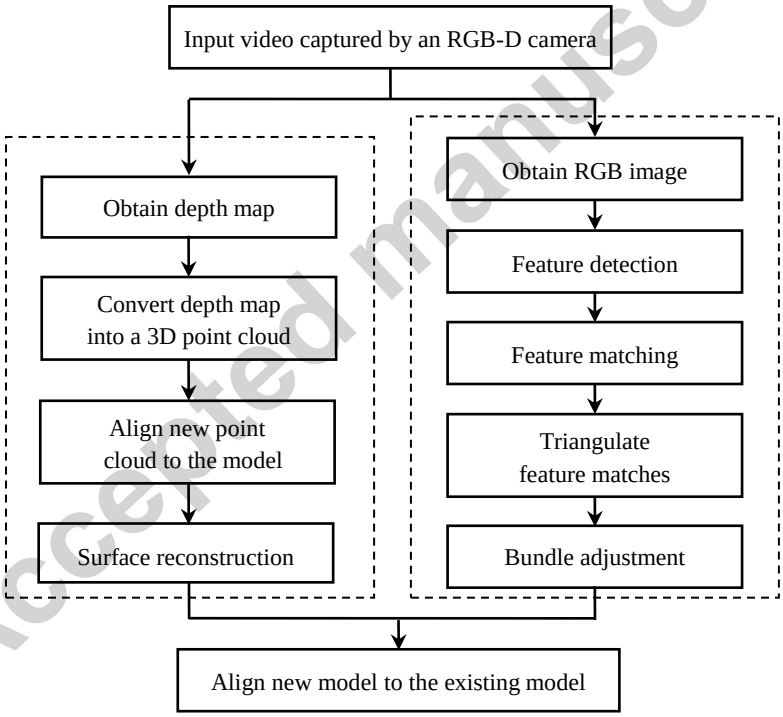


Fig. 1. Overview of the proposed system

As shown in Fig. 1, the system is divided into two stages: reconstruction with depth information and refinement with RGB information. We acquire both the depth and RGB information concurrently from the RGB-D camera. The first step of reconstruction with the depth information is to convert the depth image to a 3D point cloud. We translate the image points in the camera's coordinate space to 3D vertices in the global coordinate system using the calculated camera intrinsic parameters, and use a rotation matrix and a translation vector to represent the corresponding coordinate transformation. Then, we incorporate the new point cloud into the

existing model using the iterative closest point algorithm. Thus, the transformation relationship between the current frame and the previous frame is obtained. Finally, we use truncated signed distance functions to reconstruct a high-quality 3D surface from the point cloud. During the refinement procedure, we first use GPU-accelerated SIFT to detect and match features, and then apply triangulation and bundle adjustment to determine the relationship between two images. Meanwhile, we use filtering and re-triangulation to deal with drifting problems for further improving the reconstruction accuracy. Finally, the 3D model is aligned to the 3D model built in the first stage using the iterative closest point algorithm. This method takes advantage of both depth and RGB information and improves the reconstruction result by aligning the RGB-based 3D model to the existing depth-based 3D model, especially in case that the distance from the camera to the object is too close (less than 0.4 meters) to obtain the depth information or it fails to get the depth information due to the non-reflecting objects in the scene.

4. Alignment of Two Point Clouds Using Depth Information

4.1 Converting the depth image into a 3D point cloud

Unlike a standard color camera, an RGB-D camera provides RGB images along with the corresponding depth information of each pixel. For the following algorithm, the depth image is of each frame k is converted to a 3D point cloud. As shown in Eq. (1), we build the 3D vertex $v_k(p)=(x,y,z)$ of each image point p in the camera's coordinate space by using a calibration matrix:

$$v_k(p) = D_k(p)K^{-1}[p,1] \tag{1}$$

where $D_k(p)$ is the depth value returned by the RGB-D camera and K is the 3×3 upper triangular matrix of the infrared camera intrinsic parameters, which can be calculated by the camera calibration toolbox of OPENCV [18]. The 3D vertex, $v_k(p)$ corresponds to image point p, of which the image coordinates are (i,j).

Then, the corresponding normal vector for each vertex is computed via the cross product of the neighboring re-projected points:

$$n_k(p) = (v(i+1,j)-v(i,j))\times(v(i,j+1)-v(i,j)) \tag{2}$$

where $n_k(p)$ is the normal vector of image point p, which can be normalized to a unit length of $n_k(p)/\|n_k(p)\|$.

We use the rotation matrix $R=[r_x,r_y,r_z]$ and the translation vector $T=[t_x,t_y,t_z]$ to convert the vertex $v_k(p)$ and the corresponding normal $n_k(p)$ at frame k from the camera coordinate system to global coordinates system by using Eq. (3) and (4):

$$V_k(p) = [R_k \mid T_k]v_k(p) \tag{3}$$

$$N_k(p) = [R_k \mid T_k]n_k(p) \tag{4}$$

4.2 Aligning the new cloud to the model

The iterative closest point algorithm (ICP) is a popular and well-studied algorithm for aligning new point clouds to a model [11]. In this paper, the point clouds generated from frame $k-1$ and frame k are respectively called the target point cloud and the source point cloud. After ICP

processing, the correspondence between the source and target can be determined, and the source will be aligned to the target. The primary process includes the following steps:

Step 1: Initialize the source rotation matrix R_k and translation vector T_k using target R_{k-1} and T_{k-1}.

Step 2: Calculate the global coordinates $V^{'}{}_k$ in frame k using Eq. (5):

$$V^{'}{}_k(p) = [R_k \mid T_k] v_k(p) \tag{5}$$

Step 3: For each point in the source point cloud, find the closest point in the target point cloud. The point can be found by computing the total error between the source and target point pairs. The error can be calculated using Eq. (6):

$$error = \sum_{D_k(p)>0} \left\| (V^{'}{}_k(p) - V_{k-1}(p)) \right\|^2 \tag{6}$$

To accelerate convergence, the point-to-plane error metric is used. The total error between each source point and the tangent plane at its corresponding target point is computed using Eq. (7):

$$error^{'} = \sum_{D_k(p)>0} \left\| (V^{'}{}_k(p) - V_{k-1}(p)) \cdot N_{k-1}(p) \right\|^2 \tag{7}$$

Step 4: Update R_k and T_k, and perform Step 2 through 4 iteratively until the total $error$ is minimized. This step can be performed using a linear least-squares optimization technique by assuming an incremental transformation between frames [22].

To avoid running the algorithm running into an endless loop, the process should be terminated when the $error$ is smaller than a user-defined threshold or when the number of iterations has reached a user-defined maximum value.

4.3 Surface reconstruction

The transformation relationship between frames k and $k-1$ can be determined and the new point cloud can be integrated into the model using the rotation matrix R_k and the translation vector T_k. For the future use, a high-quality 3D surface should be reconstructed from the point cloud. The volume of the acquired model is represented by a Truncated Signed Distance Function (TSDF), which has been proved to be efficient particularly for RGB-D data [2]. This function represents the distance to the nearest surface point during the extraction of the surface of the objects. The TSDF combines signed distance functions with weight functions and only truncates the region near the real surface. The TSDF has several significant characteristics. First, it describes the range uncertainty caused by asymmetric error distributions, and makes use of all range data to reduce sensor noise. Second, it is efficient and fast to build a detailed model. Third, it has no restrictions on the object genus. Forth, it has the ability to automatically fill the "holes" during the reconstruction. The implementation steps of this method are as follows:

To begin, the physical space is divided into many cubes. The length d_x, height d_y and depth d_z of each cube are the same (e.g., 3 meters).

Next, each cube is subdivided into a voxel grid that contains a certain amount of voxels per axis (e.g., 512 voxels per axis).

Then, the TSDF value (i.e., the distance to the nearest isosurface) of each voxel is calculated by tracing a ray from the sensor through each voxel near the range surface and intersecting it with the triangle mesh.

The TSDF value features two components: the current truncated signed distance value $F_k(p)$ and a weight $w_k(p)$. The value of $F_k(p)$ indicates the distance from the current voxel to the nearest real surface along the sight line to the sensor. If the voxel is in front of the surface, then $F_k(p) > 0$; if the pixel is behind the surface, then $F_k(p) < 0$; otherwise, if the voxel is near or on the surface, then $F_k(p) = 0$. The value of $w_k(p)$ is proportional to the uncertainty of the surface measurement.

Finally, the 3D surface representing the scene is reconstructed by extracting the surface where $F_k(p)$ changes the sign. We should ignore the voxels when $F_k(p) = 1$ because these represent empty space and are of no use to this model.

To increase the processing speed of the TSDF volumetric integration, data are stored in GPU memory. It is impractical to assign a GPU thread for each voxel because there is a massive number of voxels within a given volume. Thus, we begin with only one GPU thread for the (i, j) positioned on the front slice of the volume.

5. Dense Reconstruction Using RGB Information

5.1 Feature detection and matching

The Harris [3] corner detection method is an efficient and steady feature detection method in which the rotation and illumination change have little effect on the detection result. However, the variation in the distance between the camera and an object will have a large effect on feature tracking accuracy. The SIFT (Scale Invariance Feature Transform) [8] method was proposed by David Lowe in 1999, and it has several advantages. The detected features are scale- and -rotation-invariant, while it's robust to illumination changes, noise and affine transformation. Additionally, the method has a high feature-matching accuracy and correct-matching rate. However, with an increasing number of detected feature points, it becomes time-consuming to run the SIFT method on a CPU. To speed up the feature processing, we implement SIFT on a GPU. However, we use a GPU/CPU mixed implementation to find the best option for each processing step, because not all the computation process runs faster on the GPU. The basic goal is to build Gaussian pyramids and detect DOG (Difference of Gaussians) key points in parallel. The procedure is as follows [43]:

First, we build a Gaussian scale space. Acceleration of the Gaussian scale space construction is achieved by processing Gaussian separable convolutions on the GPU using fragment programs. The intensity image, image gradients and DOG values are all stored in RGBA texture memory.

Second, for each pixel, we detect the local extreme values of the DOG pyramids in parallel on the GPU.

Third, we compress the binary bitmaps indicating key-point locations for each scale in the pyramid into RGBA data and decode them on the CPU.

Fourth, we process the histogram computation on the CPU, because it takes more time to calculate the histogram in the GPU than read the data back from the CPU,

Finally, we perform a GPU/CPU mixed implementation, because SIFT descriptors cannot be efficiently computed on the GPU. In GPU processing, gradient patches are resampled and stored in a tiled texture block. Then, the CPU reads them back and computes the 128-element SIFT descriptors. The above-described partitioning of work between the CPU and GPU minimizes the computation time on data transferring between them.

In a previous study, researchers found that [6] large numbers of image pairs were unmatched using the current feature matching procedure. If good matching of feature pairs can be identified robustly and efficiently, the computation processing will be accelerated significantly. To solve this problem, the preemptive feature matching method was implemented as described by C. Wu in [6]. The basic principle of this method is that fewer features can be detected due to a high Gaussian level, and features detected in the top scale cover a large-scale range, which is sufficient for feature matching and timesaving.

5.2 Feature triangulation and bundle adjustment

According to the basic theory of 3D reconstruction, if the corresponding features between two relative images have been obtained, the fundamental matrix and essential matrix with known intrinsic parameters of the camera can be calculated. The essential matrix can then be decomposed into extrinsic parameters of the camera, which are usually presented as a rotation matrix and translation matrix. Thus, the relative camera poses and the 3D points of matched features in the world coordinate system can be calculated. Because we obtained the corresponding features between images in the previous section, we can easily compute the camera poses and triangulate the matched features.

Because of the accumulated errors in the calculation procedure, the camera poses and 3D points of the feature points calculated using the above-described method are usually inaccurate. Bundle adjustment is an efficient method for further improving the accuracy. The goal of bundle adjustment is to find the optimized camera poses and 3D points, which minimize the reprojection error. This optimization problem is usually formulated as a non-linear least-squares problem that determines the minimum value of a cost function. The cost function is the total error of the difference between the detected features and the projection of the corresponding 3D point on the image plane. A common method for solving the optimization problem is the Levenberg-Marquardt (LM) method, which is capable of obtaining an optimal result for both the camera pose and 3D points. However, the method is time-consuming, particularly for large-scale bundle adjustments. The Preconditioned Conjugate Gradient (PCG) has been proved to significantly improve the performance of the bundle adjustment. In this paper, we use the GPU version of multicore bundle adjustment [5], which delivers a 10x to 30x boost in speed over existing systems. Additionally, this method can reduce memory usage.

To reduce computation time, bundle adjustment is usually executed after obtaining all camera poses and 3D points. The computational complexity of global adjustment is determined by the number of cameras and the number of points. As the number of cameras grows, the accuracy will become unreliable, and the computation time will increase sharply. To reduce the number of accumulated errors, researchers typically run the global bundle adjustment when adding a new image. However, this process is still time-consuming, which leads us to search for a balance between accuracy and computation time. To reduce the time cost and improve the accuracy simultaneously, we add a single image at each iteration and then run either a full bundle adjustment or a local bundle adjustment according to the real situation. The basic idea is that because it is unnecessary to perform an optimization at each iteration with the camera poses and 3D points rapidly becoming stable, we only run the local bundle adjustment on the latest calculated camera poses and 3D points (e.g., the previous 20), and the full bundle adjustment is

implemented only when the model size increases over a predefined ratio (e.g., 5%).Thus the computation time and accumulated errors are reduced simultaneously.

5.3 Filtering and re-triangulation

There are typically drifting problems during reconstruction due to the following two primary reasons. First, even though bundle adjustment is performed, the accuracy of the computed camera poses is affected by the initial values. Second, the relation between two images is calculated by correct feature matches, therefore, the inaccurate camera poses will lead to a loss of correct feature matches and the failure of some triangulation calculations.

To address this problem, filtering and re-triangulation are implemented in the system. After bundle adjustment, the 3D points with large reprojection errors and small triangulation angles are filtered. Then, the failed feature matches are re-triangulated regularly during reconstruction.

6. Aligning the New Model to the Existing Model

Although the 3D model has been reconstructed using depth information, there are still some problems with it. When the RGB-D camera is too close to the object in the scene or there is an object that does not reflect infrared light, depth information cannot be obtained, and thus, the reconstruction result will not be satisfactory. This situation will result in a failure to create a correct 3D point cloud, and thus, we cannot align the point cloud of the object to the existing 3D model. To solve this problem, we take advantage of RGB information. As a vision-based reconstruction method, dense 3D reconstruction using RGB information will not be affected by the effective minimum distance limitation or objects that do not reflect infrared light. A dense 3D model using RGB information was constructed in the previous section, and we now align these two models to obtain a better reconstruction result.

As mentioned in Section 4.2, the iterative closest point algorithm (ICP) can be used to align the new point cloud to the existing 3D model. In this section, the dense 3D models generated using depth information in Section 4 and RGB information in Section 5 are called the target cloud and the source cloud, respectively. The goal of ICP processing is to determine the correspondence between the target and the source cloud. First, we choose more than four corresponding 3D points in the target and source cloud. Then, we can calculate the rough rotation matrix R and translation vector T between the target and source cloud. Next, we update R and T until the point-to-plane error is minimized. Finally, the correspondence between the target cloud and the source cloud is calculated, and the source cloud is successfully aligned to the target cloud.

7. Results

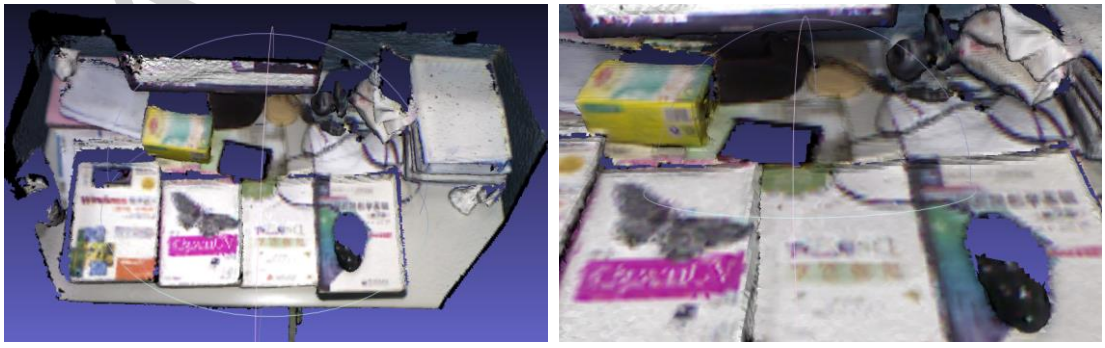
This section presents the experimental results that prove the validity of the proposed method. We ran the proposed algorithm on a desktop PC with an Intel quad core 3.2 GHz processor and a GTX 660 graphics card with 4 GB of RAM. The video sequences were captured using an RGB-D Kinect camera for Windows. In all cases, the input images had a resolution of 640×480 pixels. The intrinsic parameters of the RGB and infrared cameras were determined in advance using the GML calibration toolbox [51].

This experiment demonstrated that our method can obtain good reconstruction results, even though the RGB-D camera is too close to objects in a scene. An RGB image captured from the video sequence is shown in Fig. 2(a), and its corresponding depth image captured by the infrared camera is shown in Fig. 2(b). In the depth image, the depth values range from 0 to 255. If the camera fails to obtain depth information, the depth value will be recorded as zero, and the corresponding areas in the depth map will be black. A large distance between the object and the camera yields a large depth value. In the depth image, there are many black regions, particularly in the areas close to the camera. As shown in Fig. 2(a), the keyboard on the computer desk is near the camera, the distance to which is less than 0.4 meters. As a result, the keyboard regions in the depth image are black, indicating that depth information was not recorded. This situation will lead to inaccuracies in the reconstructed 3D model. As shown in Fig. 3(a), the keyboard area is not

(a) (b)

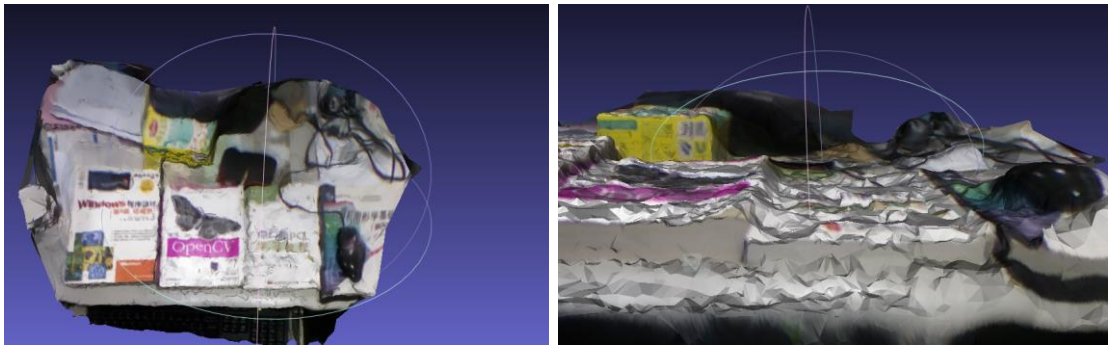
Fig. 2. (a) An RGB image captured from the video sequence. (b) The corresponding depth image captured by the infrared camera.

reconstructed in the 3D model built based on depth information. Additionally, there are many black “holes” in the 3D model. In the detailed image shown in Fig. 3(b), it can be observed that the mobile phone screen is not reconstructed. The mobile phone screen is made of a type of glass material that reflects small amounts of infrared light. Thus, the area is not reconstructed due to its low reflectivity. The reconstruction result obtained using RGB information is shown in Fig. 4(a), which shows that the keyboard region is reconstructed successfully. In addition, the mobile phone screen area is shown to be reconstructed. However, in the detailed reconstruction image shown in Fig. 4(b), the surface of the built 3D model is rough and uneven. The reconstruction results combining depth information and RGB information are shown in Fig. 5(a) and Fig. 5(b). In the final 3D model, the keyboard area is reconstructed, the “holes” are filled and the surface is smooth.



(a) (b)

Fig. 3.Reconstruction result obtained using depth information.



(a) (b)

Fig. 4. Reconstruction result obtained using RGB information.

(a) (b)

Fig. 5. Reconstruction result obtained using the proposed method.

8. Conclusions and Future Work

In this paper, we proposed an efficient 3D reconstruction approach combining a depth information based 3D model method with a RGB information based method to refine the reconstruction results when the camera fails to acquire the correct depth information. Using this method, we can obtain better 3D reconstruction results even though the distance from the RGB-D camera to the object is less than 0.4 meters or there are objects that have low reflectivity. Experimental results show that the proposed method is effective. In the future research, we will try to extend the effective distance beyond 4 meters and integrate more recent studies, including [15], [25], [28], [30], and [38], to further improve the performance of this system. We will also try to implement the proposed method for specific applications [19], [23], [31-33], [41], [44], [46], [47], [49] and [50], such as 3D object retrieval and mobile visual search, to further prove the validity of this method.

Acknowledgments

This research is supported by the Special Fund for Earthquake Research in the Public Interest No. 201508025, the National Natural Science Foundation of China (NSFC) under Grant No.61272202, and the Science and Technology Support Program of Hubei Province under grant 2014BCH270.

References

[1] A. Galvez, A. Iglesias, Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points, Information Sciences, 192 (2012) 174-192, DOI: 10.1016/j.ins.2010.11.007.

[2] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in Proc. 23rd

annual conference on computer graphics and interactive techniques, New York, USA, 1996, pp. 303-312, DOI: 10.1145/237170.237269.

[3] C. Harris, M. Stephens, A combined corner and edge detector, in Proc. 4th Alvey Vision Conference, Manchester, England, 1988, pp. 147-151.

[4] C. Hong, J. Yu, J. You, X. Chen, D. Tao, Multi-view ensemble manifold regularization for 3D object recognition, Information Sciences. DOI: 10.1016/j.ins.2015.03.032.

[5] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, Multicore bundle adjustment, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2011, pp. 3057-3064, DOI: 10.1109/CVPR.2011.5995552.

[6] C. Wu, Towards linear-time incremental structure from motion, in: Proc. International Conference on 3D Vision, Seattle, WA, United States, 2013, pp. 127-134, DOI: 10.1109/3dv.2013.25.

[7] D. Catuhe. Programming with the kinect for windows software development kit, Microsoft Press, United States, 2012.

[8] D. G. Lowe, Object recognition from local scale-invariant features, in Proc. IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157, DOI: 10.1109/ICCV.1999.790410.

[9] D.J. Crandall, A. Owens, N. Snavely, D.P. Huttenlocher, SfM with MRFs: Discrete-Continuous Optimization for Large-Scale Structure from Motion, Ieee T Pattern Anal, 35 (2013) 2841-2853, DOI: 10.1109/Tpami.2012.218.

[10] D. Valiente, A. Gil, L. Fernández, Ó. Reinoso, A modified stochastic gradient descent algorithm for view-based SLAM using omnidirectional images, Information Sciences, 279 (2014) 326-337. DOI: 10.1016/j.ins.2014.03.122.

[11] D. Viejo, J. Garcia-Rodriguez, M. Cazorla, Combining visual features and Growing Neural Gas networks for robotic 3D SLAM, Information Sciences, 276 (2014) 174-185, DOI: 10.1016/j.ins.2014.02.053.

[12] D. Yang, Z. Liu, F. Sun, J. Zhang, H. Liu, S. Wang, Recursive depth parametrization of monocular visual navigation: Observability analysis and performance evaluation, Information Sciences, 287 (2014) 38-49, DOI: 10.1016/j.ins.2014.07.025.

[13] D.Y. Kim, M. Jeon, Data fusion of radar and image measurements for multi-object tracking via Kalman filtering, Information Sciences, 278 (2014) 641-652. DOI: 10.1016/j.ins.2014.03.080.

[14] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Real Time Localization and 3D Reconstruction, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 363-370, DOI: 10.1109/CVPR.2006.236.

[15] F. Wang, Y. Zhen, B. Zhong, R. Ji, Robust infrared target tracking based on particle filter with embedded saliency detection, Information Sciences, 301 (2015) 215-226, DOI: 10.1016/j.ins.2014.12.022.

[16] F. Xue, S. Xu, C. He, M. Wang, R. Hong, Towards efficient support relation extraction from RGBD images, Information Sciences. DOI: 10.1016/j.ins.2014.12.035.

[17] G. Hu, S. Huang, L. Zhao, A. Alempijevic, G. Dissanayake, A robust RGB-D SLAM algorithm, in: Proc. IEEE/RSJ International conference on intelligent robots and systems, Vilamoura, Portugal, 2012, pp. 1714-1719, DOI: 10.1109/IROS.2012.6386103.

[18] G. Tao, L.J. Li, W. Cheng, Registration Using Multiplanar Structures for Augmented Reality Systems, J Comput Inf Sci Eng, 8 (2008), DOI: 10.1115/1.2987402.

[19] G. Yue, W. Meng, J. Rongrong, W. Xindong, D. Qionghai, 3D Object Retrieval with Hausdorff Distance Learning, IEEE Transactions on Industrial Electronics, vol.61, no. 4, pp. 2088-2098, 2014, DOI: 10.1109/TIE.2013.2262760.

[20] J. Li, W. Huang, L. Shao, N. Allinson, Building recognition in urban environments: A survey of state-of-the-art and future challenges, Information Sciences, 277 (2014) 406-420, DOI:10.1016/j.ins.2014.02.112.

[21] J.M. Frahm, P. Georgel, D. Gallup, T. Johnson, Building Rome on a Cloudless Day, in: Proc. European Conference on Computer Vision, Heraklion, Greece, 2010, pp. 368-381, DOI: 10.1007/978-3-642-15561-1_27

[22] K. Low. Linear least-squares optimization for point-to-plane ICP surface registration. Technical report, TR04-004, University of North Carolina, 2004.

[23] K. Lu, Q. Wang, J. Xue, W. Pan, 3D model retrieval and classification by semi-supervised learning with content-based similarity, Information Sciences, 281 (2014) 703-713, DOI: 10.1016/j.ins.2014.03.079.

[24] L. Sun, Z. Liu, M.-T. Sun, Real time gaze estimation with a consumer depth camera, Information Sciences. DOI: 10.1016/j.ins.2015.02.004.

[25] M.D. Robles-Ortega, L. Ortega, F.R. Feito, A new approach to create textured urban models through genetic algorithms, Information Sciences, 243 (2013) 1-19, DOI: 10.1016/j.ins.2013.03.053.

[26] P. Henry, M. Krainin, E. Herbst, X. F. Ren, D. Fox, RGB-D mapping: using depth cameras for dense 3d modeling of indoor environments, in: Proc. International Symposium on Experimental Robotics, Delhi, India, 2010, pp. 54-59.

[27] P. Henry, M. Krainin, E. Herbst, X.F. Ren, D. Fox, RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments, Int J Robot Res, 31 (2012) 647-663, DOI: 10.1177/0278364911434148.

[28] Q. Liu, Z. Zha, Y. Yang, Gradient-domain-based enhancement of multi-view depth video, Information Sciences, 281 (2014) 750-761, DOI: 10.1016/j.ins.2014.04.053.

[29] R. Ala, D.H. Kim, S.Y. Shin, C. Kim, S.-K. Park, A 3D-grasp synthesis algorithm to grasp unknown objects based on graspable boundary and convex segments, Information Sciences, 295 (2015) 91-106, DOI: 10.1016/j.ins.2014.09.062.

[30] R.A. Newcombe, et al., KinectFusion: Real-time dense surface mapping and tracking, in: Proc. 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2011, pp. 127-136, DOI: 10.1109/ISMAR.2011.6092378.

[31] R Ji, H Yao, W Liu, X Sun, Q Tian, Task-dependent visual-codebook compression Image Processing, IEEE Transactions on 21 (4), 2282-2293, 2012, DOI: 10.1109/TIP.2011.2176950.

[32] R. Ji, L.-Y. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, W. Gao, Location Discriminative Vocabulary Coding for Mobile Landmark Search, Int. J. Comput. Vision, 96 (2012) 290-314, DOI: 10.1007/s11263-011-0472-9.

[33] R. Ji, L.-Y. Duan, J. Chen, L. Xie, H. Yao, W.Gao, Learning to Distribute Vocabulary Indexing for Scalable Visual Search. IEEE Transactions on Multimedia, 2012, DOI: 10.1109/TMM.2012.2225035.

[34] R. Liang, W. Shen, X.-X. Li, H. Wang, Bayesian multi-distribution-based discriminative feature extraction for 3D face recognition, Information Sciences. DOI: 10.1016/j.ins.2015.03.063.

[35] S. Agarwal, N. Snavely, I. Simon, S. Seitz, R. Szeliski, Building rome in a day, in: Proc. IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 72-79, DOI: 10.1109/Iccv.2009.5459148.

[36] S. Izadi, et al., KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera, in: Proc. Proceedings of the 24th annual ACM symposium on User interface software and technology, ACM, Santa Barbara, California, USA, 2011, pp. 559-568, DOI: 10.1145/2047196.2047270.

[37] S. Lieberknecht, A. Huber, S. Ilic, S. Benhimane, RGB-D Camera-Based Parallel Tracking and Meshing, in: Proc. IEEE International symposium on mixed and augmented reality, Basel, Switzerland, 2011, pp. 145-155.

[38] T. Weise, T. Wismer, B. Leibe, L. Van Gool, In-hand scanning with online loop closure, in: Proc. 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), 2009, pp. 1630-1637, DOI: 10.1109/ICCVW.2009.5457479.

[39] X. Li, C. Wu, C. Zach, S. Lazebnik, J. Frahm, Modeling and recognition of landmark image collections using iconic scene graphs, in: Proc. European Conference on Computer Vision, Marseille, France, 2008, pp. 427-440,

DOI: 10.1007/978-3-540-88682-2_33.

[40] Y. Chen, D. Pan, Y. Pan, S. Liu, A. Gu, M. Wang, Indoor scene understanding via monocular RGB-D images, Information Sciences, DOI: 10.1016/j.ins.2015.03.023.

[41] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3D Object Retrieval and Recognition with Hypergraph Analysis, IEEE Transactions on Image Processing, vol.21, no.9, pp. 4290 – 4303, 2012, DOI: 10.1109/TIP.2012.2199502.

[42] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, M. Lu, A novel local surface feature for 3D object recognition under clutter and occlusion, Information Sciences, 293 (2015) 196-213, DOI: 10.1016/j.ins.2014.09.015.

[43] Y. Li, W. Liu, X. Li, Q. Huang, X. Li, GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra, Information Sciences, 281 (2014) 559-572, DOI: 10.1016/j.ins.2013.12.022.

[44] Y. Tian, Y. Long, D. Xia, H. Yao, J. Zhang, Handling occlusions in augmented reality based on 3D reconstruction method, Neurocomputing, 156 (2015) 96-104. DOI: 10.1016/j.neucom.2014.12.081.

[45] Y. Xia, L. Zhang, W. Xu, Z. Shan, Y. Liu, Recognizing multi-view objects with occlusions using a deep architecture, Information Sciences. DOI: 10.1016/j.ins.2015.01.038.

[46] Y. Yang; H. Deng; J. Wu; L. Yu, Depth map reconstruction and rectification through coding parameters for mobile 3D video system, Neurocomputing, DOI: 10.1016/j.neucom.2014.04.088.

[47] Y. Yang, Q. Liu, H. Liu, L. Yu, F. Wang, Space Coordinate Synthesis via Energy Minimization for Three-dimensional Video, Signal Processing, DOI: 10.1016/j.sigpro.2014.07.020.

[48] Y. Yang, X. Wang, Q. Liu, M. Xu, L. Yu, A bundled-optimization model of multiview dense depth map synthesis for dynamic scene reconstruction, Information Sciences. DOI: 10.1016/j.ins.2014.11.014.

[49] Y. Yang, X. Wang, T. Guan, J. Shen; Li Yu, A Multi-dimensional Image Preference Prediction Model for User Generated Images in Social Networks, Information Sciences, Vol. 281, 10 October 2014, pp: 601–610.

[50] Z. Yan, L. Yu, Y. Yang, Q. Liu, Beyond the interference problem: hierarchical patterns for multiple-projector structured light system, Applied Optics, Vol. 53, No. 17, June 2014, 3621-3632.

[51] GML Camera Calibration Toolbox downloads resource. Available online: http://research.graphicon.ru/calibration/gml-c-camera-calibration-toolbox-5.html.

Hailong Pan is currently a Ph.D. candidate at School of Computer Science and Technology of Huazhong University of Science and Technology, Wuhan, China. His research interests include augmented reality, 3D reconstruction and image processing.

Tao Guan received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China in 2008. He is currently an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology. His research interests include mobile visual search and mobile augmented reality.

Yawei Luo is currently a Ph.D. candidate at School of Computer Science and Technology of Huazhong University of Science and Technology, Wuhan, China. His research interests include augmented reality, 3D reconstruction and image processing.

Junqing Yu received the Ph.D. degree from Wuhan University, Wuhan, China in 2002. Currently, he is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology. His research interests include digital media processing and retrieval, multicore programming environment. He is a member of the IEEE and ACM.