Multimedia Tools and Applications
https://doi.org/10.1007/s11042-018-6905-5

Object registration in semi-cluttered and partial-occluded scenes for augmented reality

Qing Hong Gao[1] · Tao Ruan Wan[2] · Wen Tang[3] · Long Chen[3]

Received: 6 April 2018 / Revised: 5 November 2018 / Accepted: 14 November 2018 /
Published online: 26 November 2018

Abstract
This paper proposes a stable and accurate object registration pipeline for markerless augmented reality applications. We present two novel algorithms for object recognition and matching to improve the registration accuracy from model to scene transformation via point cloud fusion. Whilst the first algorithm effectively deals with simple scenes with few object occlusions, the second algorithm handles cluttered scenes with partial occlusions for robust real-time object recognition and matching. The computational framework includes a locally supported Gaussian weight function to enable repeatable detection of 3D descriptors. We apply a bilateral filtering and outlier removal to preserve edges of point cloud and remove some interference points in order to increase matching accuracy. Extensive experiments have been carried to compare the proposed algorithms with four most used methods. Results show improved performance of the algorithms in terms of computational speed, camera tracking and object matching errors in semi-cluttered and partial-occluded scenes.

Keywords Augmented reality · 3D object recognition and matching · 3D point clouds · SLAM algorithm

1 Introduction

Augmented Reality (AR) is an emerging field with huge application potentials. Azuma defines that AR is an integration of virtual world and real world with real-time interactions via three-dimensional registrations [3]. By mixing real scenes with virtual information, AR technology enhances human perceptions of the real world and enables novel human-computer interactions. The rapid development in software and hardware technologies in virtual reality and computer vision has made AR technology applicable to a wider range of applications from medicine, military to entertainment [7, 46].

✉ Tao Ruan Wan
t.wan@bradford.ac.uk

[1] School of Electronics and Information, Xi'an Polytechnic University, Xi'an, China

[2] Faculty of Engineering and Informatics, University of Bradford, Bradford, BD7 1DP, UK

[3] Faculty of Science and Technology, Bournemouth University, Poole, BH12 5BB, UK

A crucial process in AR is the registration between a real scene and virtual information or objects. Stable and real-time performance for virtual-real registration remains a challenging issue in markerless AR, because no markers can be used for fast matrix computations during the object recognition and matching. Conventional homography matrix method [13, 36] has low accuracy and is unstable for image-based registrations. In order to improve both registration performance and accuracy, recent approaches are proposed [19, 34], but stability and real-time performance remain an unsolved issue for fast camera movements and challenge scenes such as those containing cluttered objects and occlusions.

In this paper, we present a novel registration pipeline to improve the virtual-real object registration stability, accuracy and real-time performance in markerless AR. Our system is based on the state-of-the-art Simultaneous Localization and Mapping (SLAM) algorithm to achieve a fast and accurate real-time 3D reconstruction of the real scene and use a uniform sampling scheme to calculate feature points of the virtual objects and the scene. Two algorithms are proposed. The first algorithm computes the surface normal and feature points to calculate the Signature of Histograms of Orientations (SHOT) descriptors for the virtual objects and the scene. This process of object matching and recognition is achieved by evaluating the similarity correspondence between the object descriptor and the scene descriptor. Hough voting [51] and Iterative Closest Points(ICP) [6] algorithms are used to calculate an accurate transformation matrix for the virtual-real registration.

While this method works well for simple scenes with few object occlusions, for cluttered scenes with many occlusions, it can lead to matching and object identification errors. Therefore, the second algorithm extends the first one to address cluttered scenes by introducing a Gaussian weight function during the calculation of normal vectors. A locally supported Gaussian weight function enables the repeatability and reliability of detection for point cloud descriptors. The function also takes into account of the distance information of the area so that points closer to the current point in evaluation have a greater impact on the result of the normal vector estimation.We also use a ‘binary’ SHOT (B-SHOT) descriptor to improve the speed for object matching with fewer memory resources, thus more computation power for registration tasks. Random Sample Consensus(RANSAC)is used to remove the incorrect matching.

Some significant advances have been made by the state-of-the-art of object recognition, matching and classification approaches as reported in [2, 9, 16, 33]. Object recognition and matching is a process of detecting the presence of an object in a 3D point cloud with similar characteristics. Iterative Closest Points (ICP) algorithm is also often used for 3D object recognition. An algorithm proposed in [9] is based on a local level curvature estimation for recognizing objects in cluttered point cloud scenes. The method presented in [33] combines plane classification to identify models and method in [18] is based on local surface features for object recognition that automatically models 3D point clouds. More recently, convolutional neural networks (CNNs) and RGB-D data are used to achieve object recognitions [16] and [2]. Although these methods have achieved good results, these methods can still have difficulties to identify objects and carry out stable and accurate object registration in real-time scenes, especially when objects are in partial occlusions. In the paper, we propose a stable and accurate object registration pipeline that targets object recognition in semi-cluttered and partial-occluded real-time scenes for augmented reality.

The main contributions of this paper are summarized as follows:

- A stable and accurate registration framework for real-time object identification, classification and analysis. The computational framework includes a locally supported

Gaussian weight function to enable repeatable detection of 3D descriptors with bilateral filtering and outlier removals.
- A novel robust algorithm for object recognition and matching algorithms is proposed to improve the registration accuracy from model to scene transformation and build complete virtual-real object registrations via point cloud fusion.
- Furthermore, a novel algorithm to handle cluttered scenes with partial occlusions for robust real-time object recognition and matching with increased accuracy. The performance of our algorithms is compared with four state-of-the-art algorithms (Hough, Iterative Closest Point, Generalised Iterative Closest Point and Normal Distributions Transform). The results show good improvements over the state-of-the-art methods.

Our proposed AR registration framework improves registration accuracy with the use of integrated camera poses during the registration of virtual objects. Experiment results show the robustness of the proposed method for object recognition in cluttered scenes with partial occlusions. New experiments are also designed to evaluate the use of our markerless AR for highly interactive applications.

The remainder of this paper is summarized as follows: In the next section, we present the literature review and the related work. The proposed method is introduced in Section 3. Details of the proposed AR framework and new algorithms are discussed in Section 4. Section 5 evaluates the performance of our methods and through extensive experiments, including a comparison of the proposed methods with a number of state-of-the-art methods. Finally, Section 6 concludes the paper and presents further work.

2 Review of previous work

2.1 AR applications

In recent years, a great stride has been made in both software and hardware to improve AR technology and the technology has been used for various applications such as to guide procedural tasks in aircraft engine applications [21], to construct collaborative educational applications that can be used in practice to enhance current teaching methods [24], and to guide medical navigation systems with marker-free image registration for 3D image overlays and stereo tracking [53, 54]. Sensor technologies, such as RGBD sensors, have greatly broadened the horizon of AR technology in achieving many novel applications, offering unique user experiences of using a tabletop with a single depth camera, a stereoscopic projector, and a curved screen [5] or exploring the use of data from a Kinect sensor to perform AR with an emphasis on cultural heritage applications [8], to name a few.

2.2 AR registrations and challenges

Despite the surge of popularity of AR applications thanks to the affordability and the availability of AR technology in recent years, stable, accurate and real-time AR registration remains a challenging issue and is an active research topic. Early methods employ homography matrix for three-dimensional registrations in AR as shown in [13, 36]. Although simple and efficient, this method needs to detect coordinates of four points of a plane for the computation of camera poses (i.e. translation and orientation of the camera) w.r.t. the world coordinate system. Therefore, the fundamental principle of homography matrix method is based on 2D plane registration and the algorithm is prone to the error of misplacement of virtual objects during the registration process, resulting virtual objects being unstable onto the real scene with distracting flashing visual artifacts.

Recent advances in computer vision, in particular, the Simultaneous Localization and Mapping (SLAM) algorithms for 3D real-world reconstructions have provided new opportunities as well as challenges in generating novel AR technology especially for AR registration methods. Initially, SLAM algorithms are used mainly in robotics for robots navigation in unknown environments [4, 45], but more recently, researchers start to utilize the state of the art SLAM algorithms for virtual information and virtual object registrations in AR. Davison et al. [10, 11] use a monocular camera to achieve fast 3D modelling and camera pose tracking of real scenes and show the potential of SLAM algorithm to be used in many applications other than in the field of robotic vision, such as for locating virtual information through 3D mapping in AR based on the information of 3D point clouds [22]. Reitmayr [37] has demonstrated the use of SLAM with sensor fusion techniques to improve markerless tracking for virtual object registrations. The system consists a magnetic compass and a visual tracker, in which the initial orientation was determined by the compass. If the compass started to drift due to the change of an external magnetic field, it will result in a change in the relative rotation between the two sensors, then only visual tracker is used. Conversely, if the vision tracker fails (such as under a fast motion and blurry images) only the magnetic compass would be used. In [22], an attempt is made to make the use of the 3D map information generated by a SLAM algorithm to improve the registration accuracy. This method builds an initial map from a five-point stereo system and then tracks a camera by using the local bundle adjustment over recent camera poses to achieve more accurate registration.

Object recognition and matching is a process of detecting the presence of an object in a 3D point cloud with similar characteristics. The ICP algorithm is a popular method for 3D object recognition. In [9], a method is proposed based on local level curvature estimation for recognizing objects in cluttered point clouds. In [33], plane classification is utilised to identify models and in [18], local surface features are used for object recognition that automatically model 3D point clouds. More recently, convolutional neural networks (CNNs) and RGB-D data are used to achieve object recognition [2, 16]. Although these methods can achieve good results, it is still difficult to identify objects in real-time scenes, especially for partial occlusions. Object features have been used in human activity recognition [26, 27]. Combined with machine learning methods, object features can be used to create classifiers that improve the performance of activity recognition [25]. Although these algorithms are not aimed at AR, utilizing object features for object recognition in 3D point cloud could be a viable approach to be experimented further.

In this paper, our proposed algorithms target object recognition in real-time scenes focusing on 3D feature extractions and its speed. Hence, we use B-SHOT descriptors for points cloud extraction.

Recent advances use RGB-D sensors to achieve a dense map of the scene, for example, the KinectFusion framework [32] is well known for real-time reconstruction of dense 3D maps of the scene obtained RGBD sensors. ElasticFusion algorithm [55] also achieves fast and accurate real-time 3D scene reconstructions. However, both algorithms heavily rely on GPU accelerations for getting real-time performance, demanding high hardware requirements than normal commodity personal computers. On the other hand, real-time CPU based SLAM software mainly works with sparse point clouds [22, 29] in contrast to the dense point clouds. While sparse point clouds are useful for identifying objects, dense 3D maps of the scene help to increase the AR registration accuracy albeit higher computational costs.

Hence, a trade-off balance needs to be made when adapting SLAM algorithms for the improvement of AR registration accuracy.

Recent work addresses stability issues of AR registration was reported in [15]. The proposed method has been further improved by adding an iterative algorithm to form a computational framework for non-planar object detection and recognition [14], but object recognition and matching in cluttered scenes were not considered in these works. Object recognition and matching in cluttered scenes in AR is challenging. Different approaches are proposed to deal with this issue. A sparse metric model of the real world environment is used to provide interactive pose estimation of a virtual object and a model-based camera tracking method that generated visually pleasing augmentation results [43]. However, this method relies on a large number of natural feature points to be detected for object identification. More recently, a template based learning framework is proposed for 3D object localization and pose estimation in heavily cluttered and occluded scenes [47]. The framework uses synthetic renderings of a 3D model for training to infer latent class distributions. Thought this method can efficiently recognize objects in cluttered scenes, it requires artificial makers to locate the scene. Our current work presents new algorithms in addition to the original proposed. While the first algorithm effectively deals with simple scenes with few occlusions with improved registration accuracy, the second algorithm can handle cluttered scenes with partial object occlusions for robust object recognition and matching. The proposed method includes a locally supported Gaussian weight function to enable the repeatable detection of 3D descriptors, and a bilateral filtering and outlier removal algorithm that preserves the edge of the point cloud to increase the matching accuracy by removing some of interference points. Additional experiments are designed to evaluate the new algorithm.

3 Overview of the method

In this paper, we present a novel approach to improving the accuracy and the efficiency of AR registration between virtual objects and the real scene as well as the numerical stability during the fusion of the virtual-real objects with considerations of object occlusions.

There are a number of core technique steps in a markerless AR registration process. Firstly, a 3D map of the scene is constructed via a SLAM algorithm. Secondly, object recognition in the constructed 3D scene is performed to identify scene objects that needs to be fused by virtual objects. Finally, a matrix that transforms the virtual model to the scene is calculated and the pose of the camera is obtained to transfer the 3D model coordinates into the camera coordinates in order to register the virtual model in the real scene. In [30], although RGB-D images are used to obtain sparse point clouds [30], the depth information provided by the RGB-D images are only used at the initialization step and the conventional ORB was used to reconstruct feature points of the 3D map without fully utilizing the depth information. In contrast, in our proposed AR framework, we reconstruct a dense map in real-time by determining the depth of the keyframe data. We then generate a dense point cloud from the RGB and the depth images of the current frame. Since ORB [39] is a versatile and accurate SLAM solution, we compute are able to compute camera trajectories in real-time and build a sparse 3D reconstruction of the scene of various different environments [40], but by adding a real-time dense point cloud map to the sparse point cloud, we are able to increase the accuracy of virtual object registration.

Our markerless AR registration problem is stated as follows: A point cloud P is a data structure of a collection of multi-dimensional points $p \in \mathbb{R}^n$, and the elements in a 3D point cloud are usually represented as a vector of $X, Y, and\ Z$ of geometric coordinates of an underlying sampled surface. Given the point cloud of a virtual object with surface points $p \in P$, and a target scene point cloud Q with the target surface points $q \in Q$, the task of general/basic AR registration is to find the correspondences between P and Q, and estimating a transformation T that maps all pairs of corresponding points $p_i \in P, q_i \in Q$. The problem of AR registration lies in computing the unknown correspondences and fining the optimal transformation subject to an error metric.

One of the key features of markerless AR applications compared with other 3D data acquisition applications is that AR uses RGB-D images and 3D representations at a high frame rate (e.g. 30 fps) to generate consecutive point clouds that are temporally and spatially close to each other. Therefore, to satisfy the temporal and spatial conditions, AR registration pipeline must process the point cloud information at a comparable speed to the high frame rate during the data acquisition process. The point clouds being close to each other in the AR process make it easy for Iterative Closest Points (ICP) algorithm [6] to reach a good local minimum while finding the transformation matrix for the virtual-real fusion. However, the source data obtained from this type of 3D sensors (such as a Microsoft Kinect device) contains much noise of various forms. Therefore, in our proposed AR pipeline, following steps are designed to increase the accuracy and performance during the object recognition and matching framework:

1) *Prepossessing and Filtering:* We apply an edge-preserving bilateral filter adapted from the field of 2D image processing to remove the outliers of the input point clouds, whilst smoothing neighbouring similar pixels without affecting the edges.
2) *Feature Estimation:* The normal for each point of both the virtual model and the scene points are computed.
3) *Point Cloud Sampling:* A uniform sampling method that is fast and efficient is used to select key points of the input point clouds in order to compute feature descriptors.
4) *Correspondences Estimation:* The key points from 3) are used to calculate the Signature of Histogram of Orientations (SHOT) descriptors for the virtual model and the scene to estimate the correspondences between the two sub-sampling point clouds. False correspondences are rejected by finding a consistent set to reduce the number of outliers.
5) *Object Recognition:* The result of the Hough voting algorithm [51] obtained from the correspondence points is used to identify the objects in the scene.
6) *Transformation Estimation:* Finally a small set of previously found robust correspondences and the transformation between the virtual model and the scene are computed by a point-to-point error metric with the use of ICP to find the optimal transformation through minimizing the error metric.

In summary, we proposed a new real-time solution for AR registration problems to improve the object recognition and matching process with stable, accuracy and high registration performance. A set of experiments have been devised to evaluate the robustness of the proposed method by conducting two types of analysis: 1)object recognition analysis; 2)registration error analysis. In the object recognition analysis, we compare the standard Hough voting method with the improved method and in the registration error analysis, we compare the standard homography matrix method with the proposed method. Six components of the 3D registration results are analyzed. Finally, we demonstrate AR application examples to highlight the use of our proposed AR framework.

4 The proposed AR framework

The proposed new AR framework consists of two software modules: a SLAM module and a registration module as shown in Fig. 1 for an overview of the system. Tracking in the SLAM module is to find the camera position by processing each image frame and to decide when a new keyframe should be inserted. Firstly, a feature matching process is initialized with the previous frame and the Bundle Adjustment (BA) method [52] is used to optimize the camera poses.

The registration module is called after the 3D map is initialized and successfully created by the SLAM module. The RGB-D data is added to fuse the point cloud by the previously calculated pose to generate a dense 3D map. We have used the three-dimensional model to identify the object and to obtain the transformation matrix. After this process together with the SLAM, the camera position is obtained and the pose is converted into the global coordinate system under the model local view matrix (a matrix transfers the 3D model coordinates into the camera frame). The final step is to register the 3D virtual object to the real world scene to achieve the desired augmented reality effects.

4.1 Tracking

Tracking in our system is achieved via a visual simultaneous mapping and tracking strategy by extracting and matching the Oriented Features From Accelerated Segment Test (FAST) [38] and the Rotated Binary Robust Independent Elementary Features (BRIEF) (ORB) [39]. We compute two models: i) a homography matrix to compute planar scenes; ii) a fundamental matrix to compute non-planar scenes. At each time the two matrices are calculated and scores ($M = H$ for the homography matrix, and $M = F$ for the fundamental

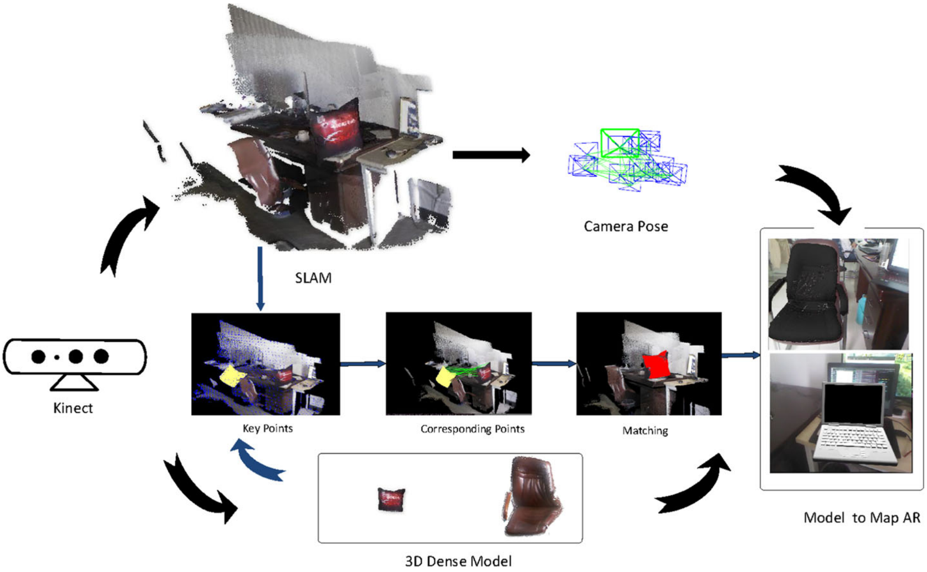

Fig. 1 System overview shows the workflow of our proposed AR framework and the components of the AR system

matrix) are also calculated as shown in (1). The scores are used to determine which of the models is more suitable for the current camera posture.

$$S_M = \sum_i \left(\rho_M\left(d^2_{cr,M}\left(x^i_c, x^i_r\right) + \rho_M\left(d^2_{rc,M}\left(x^i_c, x^i_r\right)\right)\right)\right) \tag{1}$$

$$\rho_M\left(d^2\right) = \begin{cases} \Gamma - d^2\, if & d^2 < T_M \\ 0 \qquad if & d^2 \geq T_M \end{cases}$$

where d_{rc} and d_{cr} is the measure of symmetric transfer errors [1], T_m is the outlier rejection threshold based on the χ^2, Γ is equal to T_m, x_c is the features of the current frame, and x_r is the features of the reference frame. The BA is used to optimize camera poses, which gets a more accurate camera position as shown in the following equation:

$$\{R, r\} = \arg\min_{R,t} \sum_{i \in \chi} \rho\left(\left\|x^i - \pi\left(RX^i + t\right)\right\|^2_\Sigma\right) \tag{2}$$

where $R \in \mathcal{SO}^3$ is the rotation matrix, $t \in \mathbb{R}^3$ is the translation vector, $X^i \in \mathbb{R}^3$ is a three-dimensional point in space, $x^i \in \mathbb{R}^2$ is the key point, and ρ is the Huber cost function. Sigma item is the covariance matrix associated with the key point and π is the projection function.

After obtaining the accurate position estimation of the camera, the three-dimensional map of the point cloud is obtained by triangulating the key frames through the camera poses. Finally, the local BA is used to optimize the map. A detailed description of the process is given in [29].

4.2 Dense mapping

We add a dense 3D map in real-time to the sparse point clouds to increase the accuracy of the registration. In the process of building the dense map, a Kinect sensor is used to extract the RGB-D information so that SLAM poses can be extracted and combined with the sparse point clouds. Central to this method is adding a dense point cloud processing thread when the system is at the initialization stage, which creates a visual window for displaying a dense map.

In order to achieve the required real-time performance (i.e. at least 30 Hz), the map is not captured at every frame of the image, instead only a set of keyframes are captured. When the keyframes of the system are updated, the RGB-D information of the current frame is extracted. Therefore, the point clouds are reconstructed from the key-frame images. The camera pose of the current frame can also be obtained when processing the keyframes. After that, we can transform the point cloud of the corresponding keyframes into the same coordinate system according to the pose of the current keyframe to generate a global point cloud map.

4.3 3D object recognition

In our AR system, the object recognition is not performed during the building mode of the SLAM system, it is only running in the location mode and Hough voting method is used for 3D object recognition to increase the performance and accuracy of ICP algorithm that maps the model to the corresponding transformation matrix. The recognition results are shown in Fig. 2. Figure 2a shows the key points obtained using a uniform sampling method,

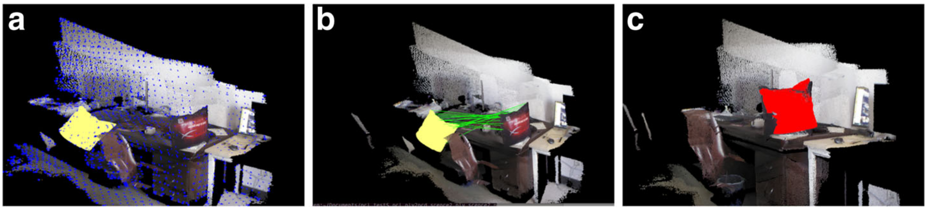

Fig. 2 3D object recognition (the red region is the matched model, the yellow region is the original model and the blue dots are the key points): **a** the key points obtained using a uniform sampling method; **b** descriptors of the model and the scene w.r.t. the matching of the corresponding points; **c** the result of the final match

whereas Fig. 2b shows the descriptors of the model and the scene w.r.t. the matching of the corresponding points and Fig. 2c shows the result of the final match. It can be seen that the algorithm can effectively identify the object in a scene. Details of the specific process are listed in Algorithm 1.

Algorithm 1 3D object recognition

1: Using the nearest neighbor method to calculate the surface normal of the model and the scene separately. Calculating the surface normal can be done by solving the eigenvectors and eigenvalues of a covariance matrix, which is created by neighboring elements of query points. The normal of each point can be obtained by (3) and (4).

$$C = \frac{1}{k}\sum_{i=1}^{k}\left(P_i - \overline{P}\right)\left(P_i - \overline{P}\right)^T \tag{3}$$

$$C \cdot \overrightarrow{v_j} = \lambda_i \cdot \overrightarrow{v_i}, \; j \in \{0, 1, 2\} \tag{4}$$

where C is the covariance matrix, k (k=10) is the number of neighbor points considered in the neighborhood of P_i, If K value is larger, the speed of computation will be slow and the accuracy becomes lower and vice verse. $\overline{P}$ represents the 3D centroid of the nearest neighbors, λ_i is the j-th eigenvalue of the covariance matrix, and the j-th eigenvector.

2: The Uniform Sampling algorithm is used to calculate the key points of the model and the scene.(The radius of the search is 0.01 for the model and 0.03 for the scene). The algorithm mainly creates a 3D voxel grid and calculates the centroid of each mesh within the grid, using the centroid of each grid to represent the entire point cloud;

3: Using the above-mentioned surface normal and the key points to calculate the Signature of Histograms of Orientations (SHOT) descriptors for models and scenes, the radius of search is 0.02. A detailed description of the approach is given in [49];

4: By calculating the similarity (squared distance) between the model and the scene description points, the corresponding description points can be found;

5: Using the Hough voting to identify the object and calculate the corresponding transformation matrix the radius of search is 0.015, the values of a bin is 0.01 and threshold is 2.0 [51];

6: The transformation matrix in step 5 is further processed by ICP to obtain a more accurate transformation matrix. Times of iteration is 50 and the max distance of correspondence is 2.

4.4 Object recognition and matching in cluttered scenes

The proposed Algorithm 1 performs much better for object recognition and matching in simple scenes where there is few object occlusion. The recognition and matching performance can be low for Algorithm1 to perform in challenging scenes where the object to be recognized and matched is in a cluttered scene with partial occlusions. Hence, we propose Algorithm 2 to address this problem to extend the capability of our AR framework deals with more complex scenes that Algorithm 1 handles poorly.

We add a bilateral filtering process [48] to preserve the edge of the point cloud and an algorithm to remove outliers [41]. Information about edges is important during object identification and matching, and the outlier removal takes out the interference points. The two additional processes greatly increase the matching accuracy as shown in experiment results Fig. 3.

When calculating normal vectors, we weight close points with a locally supported Gaussian weight function [17]. Repeatable detection of a point cloud descriptor refers to a descriptor detected in a rule model that is also detectable in the presence of any pose in a scene with occlusions. If the point cloud descriptors can ensure repeatable detections, the LRF (local reference frame) in the spherical support area must be unique. Meaning, LRF for feature point estimation is unique in any pose and scenes. Therefore, a unique descriptor can be obtained based on the LRF under different scene conditions. The process of establishing LRF is the process of calculating normal vectors. And when calculating the normal vector of the scattered point cloud, we need to consider the distance information of the area that the closer distance should get a greater contribution. The Gaussian weight means that the point closer to the current point will have a relatively greater impact on the result of the normal vector estimation. In the descriptor calculation, we use the B-SHOT descriptor [35], which is a 3D descriptor calculation method based on 'binary' of a SHOT descriptor. Because in real-time augmented reality systems, the choice of B-SHOT descriptor can improve the object matching speed. And the B-SHOT takes up less system memory. In the final calculation of the transformation matrix, we calculate the initial transformation matrix by Random Sample Consensus(RANSAC). The detailed process is given in Algorithm 2.

Algorithm 2 Cluttered scene 3D recognition

```
   The point cloud is processed by a bilateral filter and statistical outlier removal algorithm;
2: Weighting close points with a locally supported Gaussian weight function and calculating the normal vector of the model and the scene by (5);

   $P\left(n\right) = min \sum_{i=1}^{k} \theta\left(\|p_i - q\|\right)\left(n^{\top}\left(p_i - q\right)\right)^2 \ s.t. \ \|n\|_2 = 1$   (5)

      where $n$ is the normal vector of the point $q$, $p_i$ is the neighborhood point of $q$, $\theta\left(\cdot\right)$
   is Gaussian weight$(\theta\left(\delta\right) = \delta^{-r}, r > 0)(r = 0.55)$.
   Same as the Step 2 of Algorithm 1;
4: Calculating the B-SHOT descriptor for the model and the scene and obtaining the corresponding description points by RANSAC method. The value of inline threshold is 0.05;
   Same as the Step 5 of Algorithm 1;
6: Based on the matrix obtained in Step 5, the transformation matrix is firstly calculated by RANSAC.
```

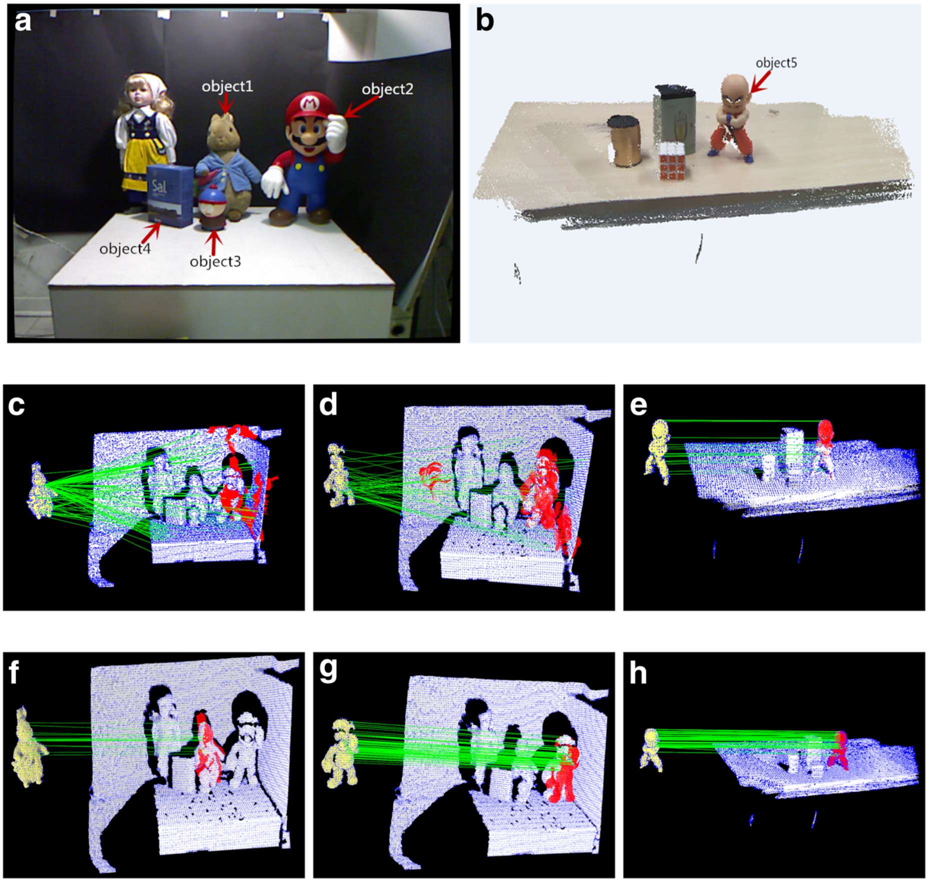

Fig. 3 Cluttered scenes 3D recognition (**a** is a standard data set **b** is our own dataset): The three of middle show result of algorithm 1. The three of bottom show the results of algorithm 2. The algorithm 1 obtains error correspondences and misidentifies in cluttered scenes and algorithm 2 has produced much-improved results. **a** shows colored **c**, **d**, **e** and **e**. **b** shows colored **e** and **f**

To demonstrate the effectiveness of Algorithm 2, we perform a study to compare the recognition results of Algorithm 1 and Algorithm 2, using two test scenes, a text scene for performance evaluation of 3D keypoint detectors from [50] as shown in Fig. 3a and a scene of our own as shown in Fig. 3b. In the test scene of Fig. 3a, objects 1 and 2 are to be tested by Algorithms 1 and 2 for AR object recognition and matching, where the scene contains multiple objects and partial occlusions between objects 1, 2, 3 and 4. This is a standard scene for publicly available for algorithm evaluation in the academic research community. The test scene shown in Fig. 3b is made up by us, in which there are multiple objects close to the test object 5 that has no occlusion by other objects in the scene. The evaluation of object 5 is useful to compare how the performances of the proposed algorithms in a cluttered scene even without occlusion.

Figure 3c, d and e show performance algorithm 1 on recognition and matching for object 1 (Fig. 3c), object 2 (Fig. 3d) and object 5 (Fig. 3e). As can be seen that Algorithm 1 performed poorly in cluttered scenes especially for partially occluded objects 1 and 2, where in (c), object 1 has not been recognized completely. Although some feature points of object 2 can be identified, there are still many false matching points, and object 5 is only partly identified. In contrast, Fig. 3f, g and h show the performance of Algorithm 2 that has successfully identified object 1 (Fig. 3f), and its performance for objects 2 (Fig. 3g) are greatly improved to find correspondence points and recognize the object. It is almost a complete recognition for object 5 (Fig. 3h) compared to the result produced by Algorithm 1 for the same object. It can be seen that Algorithm 2 can effectively identify the object in cluttered scenes with partial occlusions, and objects without occlusion, its performance is super than algorithm 1. Details of the specific process are listed in Algorithm 2. Figure 4c, d and e show performance Algorithm 2 on recognition and matching for Mario, Duck and Buddha), Algorithm 2 can achieve great results under different scenes.

4.5 AR registration

The virtual object is finally registered in the real world via a series of coordinate system transformations (i.e. from the world coordinate system to the camera coordinate system, to the crop coordinate system, and finally to the screen coordinate system). The transformation sequences can be described by (3) from left to right: the world coordinate system is transformed into the camera coordinate system by a rotation matrix $R_{3\times 3}$ and a translation matrix $T_{3\times 1}$.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & d_x & 0 \\ 0 & f_y & d_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3\times 3} & T_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

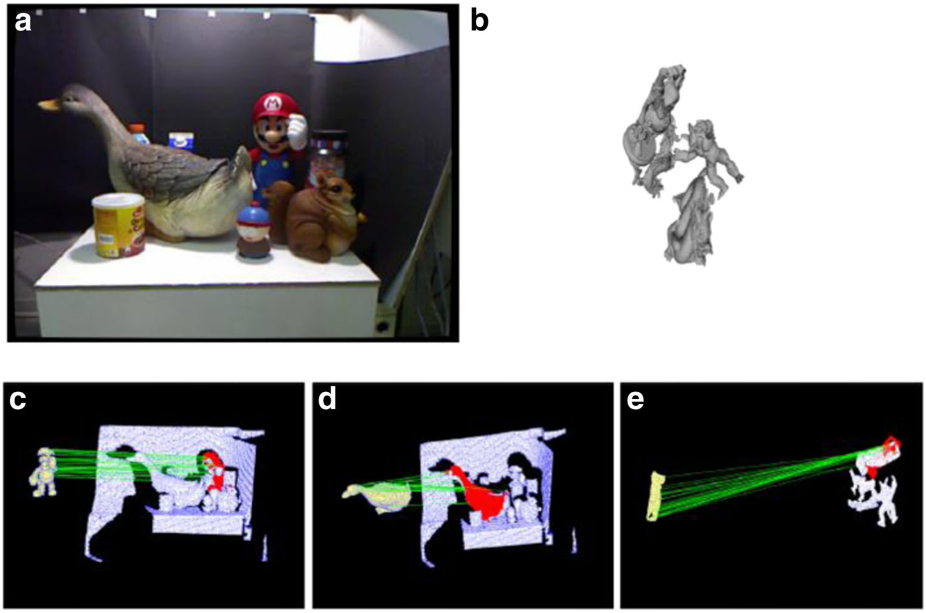

Fig. 4 3D recognition in cluttered scenes by Algorithm 2(**a** and **b** are more complex and clutter datasets with partial occlusions. **c**, **d** and **e** are results achieved by Algorithm 2)

These matrices are constructed by the camera's position and the detected scene information. The camera coordinate system is then transformed into the screen coordinate system (u, v) by the focal length (f_x, f_y) and the principal point (d_x, d_y). These parameters are obtained by the camera calibration. Finally, the virtual object is registered onto the scene of the real world images.

5 Experiment and evaluation

We have designed and conducted a large range of experiments to evaluate the robustness of our proposed AR registration method in terms of accuracy and stability over other four popular methods. Our experiments are run under an Ubuntu 14.04 system, CPU clocked at 2.3GHz, 8GB memory and NVIDIA GeForce GTX 960MB graphics card. The camera resolution is 640 by 480 pixels at 30 Hz.

Figure 5 shows an AR example testing. Figure 5a and b show that the system identifies and registers a virtual table for a real table. Figure 5c shows the system identifies and registers a virtual laptop (the front laptop) for a real laptop (the black laptop). Figure 6a also shows the identification and registration of the virtual laptops. Figure 6b and c show the identification and registration of the 3D model reconstruction of a real char (the black chair in front) from a real chair (red chair behind the virtual black chair).

5.1 Object recognition analysis

To evaluate the accuracy of object recognition, we set the target objects in the scene and match them to virtual models. The corresponding transformation matrix has to be computed first. A fixed degree for the camera rotation with a fixed translation distance for the camera movement is then set. This will be used as the basis of the unit matrix to compute the reference matrix. For example, a reference transformation matrix is composed of a rotation matrix and a translation vector. We can then fix a rotation angle of 45 degrees and the rotation axis as the Z-axis for the rotation vector. we can then transformed it into a rotation matrix by Rodriguez's Formula. The translation vector is then obtained by changing 0.2 meters at each frame time. With the rotation angle fixed and the known camera movement quantity, we are able to evaluate the corresponding error of the translation vector w.r.t different changes of the translation. This error is obtained by comparing the transformation matrix that is obtained from the current values.

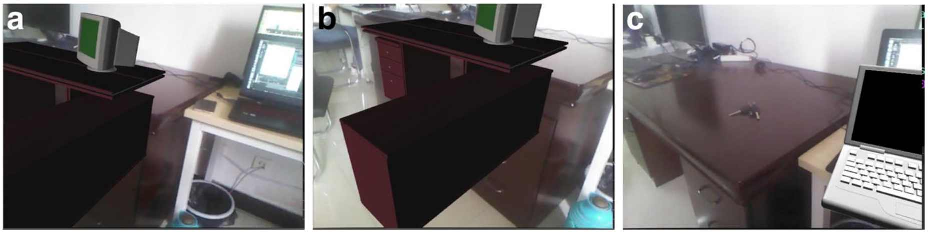

Fig. 5 AR tracking, recognition and registration: **a** and **b** show the system identifies and registers a virtual table for a real table with different view perspectives. **c** shows the identification and registration of a virtual laptop for a real laptop

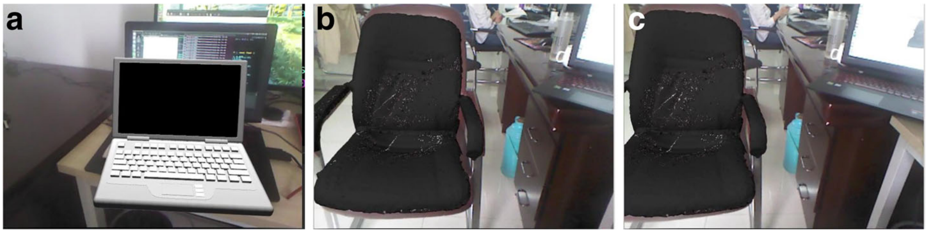

Fig. 6 AR tracking, recognition and registration: **a** shows the identification and registration of the virtual laptop with a real laptop, **b** and **c** show the identification and registration of a 3D model reconstruction from a real chair and registration with a real chair in different view perspectives

The formula parameters used in our algorithms are important, which will affect the performance of the system and are currently set by trial and error in an iteration way to achieve optimized results. In Algorithm 1, the value of k will affect the accuracy about the surface normal and running speed(setting $k = 10$). The radius of search in step 2 and 3 will affect the number of key points and descriptors. we set $r = 0.01$ of model and $r = 0.03$ of scene in key points calculation. The value of radius in calculation descriptors sets 0.02 in model and scene. There are two parameters that are bin and threshold during using a Hough voting algorithm. We set 0.01 and 2.0 respectively detail [49]. The times of iteration are 50 and the max distance of correspondence is 2 in ICP. In Algorithm 2, we set $r = 0.55$ that is an empirical value. It is only greater than zero. The value of the inline threshold is 0.05 in RANSAC method. In addition, the rest is the same as Algorithm 1. To see the details how those parameters are implemented in the calculation, refer to Algorithm 1 and 2 in Section 4.4 in this paper.

To process further, the virtual model is then multiplied by the reference matrix to get a new model. By using this new model, we use the Hough voting algorithm, algorithm 1, algorithm 2, ICP, Generalized Iterative Closest Point (GICP) [42] and Normal Distributions Transform (NDT) [44], respectively to obtain the transformation matrix of the model transformation to the scene.Here, we use the similarity of the matrix (4) for the rotation matrix and the European distance for the translation matrix respectively.

Experimental results are shown in Fig. 7. Figure 7a shows the rotation angle fixed at 45 degrees, the abscissa indicates the increased distance (x, y, and z components while increasing the same distances). The ordinate represents the errors between the calculated and ground truth values. In Fig. 7b, the translation component is fixed at 0.1 cm, the abscissa represents the increased degrees, and the ordinate represents the similarity measure. the error of the transformation matrix obtained by algorithm 2 indicated as the green is much smaller than that of algorithm 1 indicated as blue and the original Hough voting indicated as red.

$$r = \frac{\sum_m \sum_n \left(A_{mn} - \overline{A}\right)\left(B_{mn} - \overline{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \overline{A}\right)^2\right)\left(\sum_m \sum_n \left(B_{mn} - \overline{B}\right)^2\right)}} \tag{4}$$

where $\overline{A}$ and $\overline{B}$ are the means of matrix elements, mn is m rows and n columns of the matrix, r is correlation coefficient of the matrix (-1 and 1 represent exactly the same matrix, 0 represents the two matrices are completely different).

In order to evaluate the performance of the proposed method over other popular methods, two groups of new experiments (one is normal points cloud data-set, another is added Gaussian Noise) are designed. In the experiments, we also test calculation speed which is

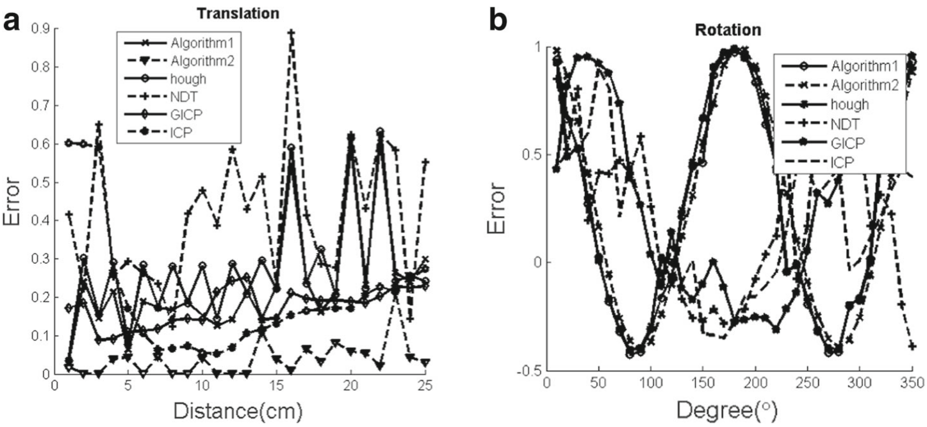

Fig. 7 Recognition Analysis: **a** When the rotation angle is fixed at 45 degrees, the distance errors are shown between the calculated and ground truth values. **b** the translation component is fixed at 0.1 cm, the error of the translation matrix obtained by Algorithm 2 is much smaller than by Algorithm 1 and 4 state-of-the-art methods

important for real-time applications. Final comparison results are shown in Table 1. The Normal in Table 1 shows the mean error of the methods in Fig. 7. Then Gaussian Noise (0,0.01) is added to the point cloud of model scenes and three evaluation metrics are tested and analysis. The results of experiments as shown in the table, it is clear that Algorithm 2 performs best in precision. For running speed, DNT achieves faster speed at the cost of a larger error over other methods. Therefore, Algorithm 2 has achieved better results overall in real-time applications.

5.2 Registration error analysis

The second experiment is to evaluate the registration error. A comparison method is used with fixed camera positions to evaluate the robustness of our proposed method. The 3D registration of the virtual object is carried out by using the proposed method and the standard homography matrix method. Six components of the 3D registration result are analyzed. The differences between the transformation matrix of the current frame and the corresponding component of the transformation matrix of the previous frame are used as the basis for

Table 1 Transformation matrix and speed comparison by three evaluation metrics

Method	Included Gaussian Noise			Normal		
	mean(T)	mean(R)	Speed(s)	mean(T)	mean(R)	Speed(s)
NDT	0.5265	0.6469	**0.1229**	0.4133	0.6545	**0.0384**
GICP	0.1712	0.5388	0.9848	0.1736	0.5683	0.1028
ICP	0.2660	0.7052	0.9836	0.2320	0.5554	0.1642
Hough	0.3918	0.5501	44.6321	0.2717	0.5535	9.9523
Algorithm 1	0.3060	0.5314	45.4459	0.2027	0.7001	10.8922
Algorithm 2	**0.1581**	**0.5142**	0.5887	**0.0367**	**0.5432**	0.0674

Bold number indicate the best performance

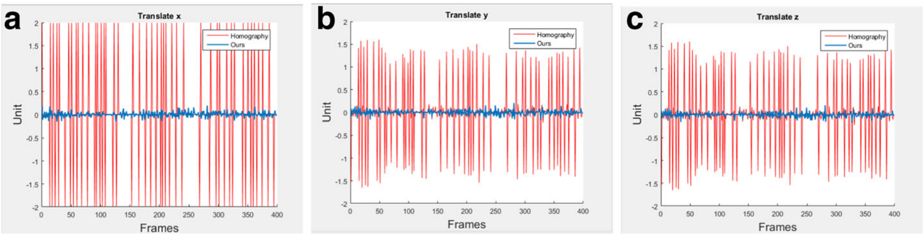

Fig. 8 Registration error of the translate vector: **a** shows the x-axis component error; **b** show the y-axis component error, **c** show the z-axis component error

the comparison. The results are shown in Figs. 7 and 8, where Translate x, Translate y and Translate z are errors of the translation components, respectively, and Rotate x, Rotate y, Rotate z are relative to the x, y, and z-axis of the rotation component errors which are obtained by subtracting the previous frame from the current frame. The result of the rotation component is obtained by dividing the respective components with the dot product of the corresponding coordinate axis, and the translation component is the result obtained by a normalization process.

In Figs. 7 and 8, the red curves are the results of using only the homography matrix, whereas the blue curves are the results of the new registration method described in this paper. As it can be seen, Using a homography matrix method to register the virtual objects has produced large registration errors that are equivalent to the virtual object registration instability. However, the new method tested on each rotation component has been kept the error in a small range below 0.5 degrees. The errors with Translate x, Translate y and Translate z are also small similar to the result of the rotation components.

Through the experimental results, it can be seen that the new method produces stable virtual registration and solves the flickering phenomenon in the virtual reality registration, hence, improves the stability of the AR system.

5.3 Limitations

5.3.1 Dynamic environment mapping

One of the major challenges currently faced with SLAM based 3D sensing systems is dealing with changes in the 3D environment i.e. dynamic environments [23]. For example, in robotic automation, long-term continuous generation of dense environment maps is extremely challenging. In AR, the ability to distinguish static elements from dynamic

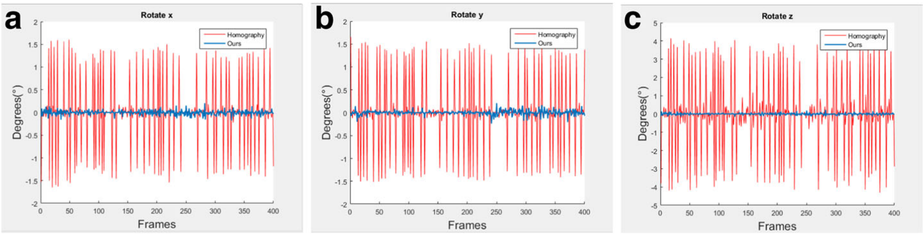

Fig. 9 Registration error of the rotation matrix: **a** shows the x axis component error; **b** show the y-axis component error, **c** show the z-axis component error

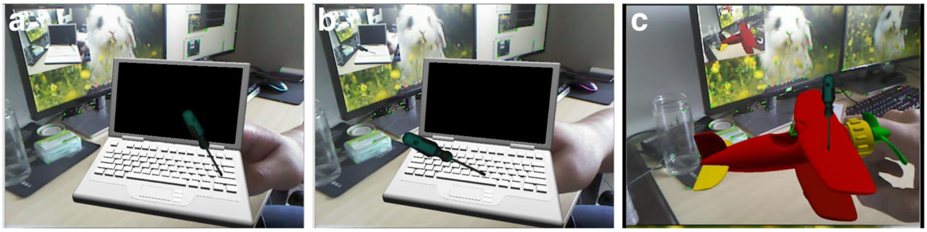

Fig. 10 AR application: **a** shows laptop (static) and screwdriver (moving). **b** shows a different screwdriver's position with the same model. **c** shows the toy plane (static) and screwdriver (moving)

elements in a 3D environment would open up many applications. In many cases, multiple virtual models are required to work together simultaneously. One example is the training in surgical skills, where a fixed body model and a moving scalpel model are required, and in the training of fixing complex electrical equipment, the need for a fixed device model that can follow the handle of a tool model. Such scenarios require dynamic AR registrations, which need solving the problem of consist of dense long-term mapping and 3D reconstruction. Currently, our system does not deal with the dynamic environment, where some of the objects in AR may not be static (i.e. the moving scalpel model in surgical training etc.).

5.3.2 Multiple objects registration

In addition to the AR testing examples, we further designed two examples to show a proposed solution for dealing with dynamic objects by integrating a marker based registration with our markerless system for more complex AR applications (as shown in Fig. 10). We show how our system performs in terms of using two types of AR registrations in terms of stability and performance for multiple static and dynamic objects. Because the static object needs for better stability and does not allow makers in many cases, such as in laparoscopic surgery no markers are allowed in a patient body. We, thus, apply marker based registration to the moving object for real-time detection and matching as a demonstration example. In the two simple AR examples, the static model is registered with the 3D scene reconstructed from the proposed method. The application run under i7-7700k CPU at 15 fps. Figure 9a–c shows two virtual objects which include a static object(laptop and toy plane) and a moving object (screwdriver). The stationary objects are registered with a 3D map. The moving object is registered in real time. As it can be seen that the tracking, recognition and registration have been effectively performed correctly (Fig. 10).

6 Conclusions and future work

This paper presents a stable and high-performance realistic tracking and recognition method in markerless AR based on 3D map information generated by SLAM. The proposed AR framework enables accurate and stable virtual object registration to meet the highly interactive requirements of various AR applications. Our contribution is also the design of experiments for the evaluation of the proposed algorithms. The evaluation method proposed in this paper is genetic, which can be used to test different approaches. The experimental results show that the proposed method can effectively suppress the virtual object jittering, having a higher tracking accuracy with good performance, and the new algorithm 2 effectively handles cluttered scenes.

We integrated two virtual objects(a static object and a moving object)by our method, which can be used in medical training and maintenance training as application examples. The method allows the tracking and the registration of virtual objects to ensure a stable and real-time performance of markerless AR applications. Our proposed method is faster than the standard methods and is able to achieve more accurate registration results compared with the state-of-the-art approaches.

At present, we are using object recognition based on the model recognition algorithm. There are a number of future research directions. We would like to consider multi-model 3D object recognition based on deep learning [20] in our future work. In terms of reconstruction of a dynamic 3D environment and dynamic object registration, the recent advance in Dynamic Fusion [31] has set a benchmark for research in this area. In [28], a method has been proposed for dense semantic segmentation of 3D point clouds, which can be applied to AR semantic recognition. In the dynamic reconstruction, we are prepared to refer to the paper [12] to achieve 3D reconstructions of dynamic objects in the scene in order to facilitate the establishment of dynamic registration based on a mapping.

Acknowledgments This research work is supported by Shanxi Natural Science and Technology Foundation of China, grant number 2016JZ026 and grant number 2016KW-043).

References

1. Hartley R (2000) Multiple view geometry in computer vision, 2nd edn Cambridge University Press
2. Alexandre LA (2016) 3d object recognition using convolutional neural networks with transfer learning between input channels
3. Azuma RT (1997) A survey of augmented reality. Presence: teleoperators and virtual environments 6(4):355–385. https://doi.org/10.1162/pres.1997.6.4.355
4. Bailey T, Durrant-Whyte H (2006) Simultaneous localization and mapping (SLAM): part ii. IEEE Robot Autom Mag 13(3):108–117. https://doi.org/10.1109/MRA.2006.1678144
5. Benko H, Jota R, Wilson A (2012) Miragetable: freehand interaction on a projected augmented reality tabletop. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 199–208. ACM
6. Besl PJ, Mckay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14(3):239–256
7. Bimber O, Raskar R (2005) Spatial augmented reality merging real and virtual worlds, 1st edn. Taylor & Francis Group
8. Bostanci E, Kanwal N, Clark AF (2015) Augmented reality applications for cultural heritage using kinect. Hum Centric Comput Inf Sci 5(1):1–18
9. Czerniawski T, Nahangi M, Haas C, Walbridge S (2016) Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor. Autom Constr 71:346–358
10. Davison AJ, Mayol WW, Murray DW (2003) Real-time localization and mapping with wearable active vision Proceedings of the second IEEE and ACM international symposium.1 Mixed and augmented reality, pp 18–27. https://doi.org/10.1109/ISMAR.2003.1240684
11. Davison AJ, Mayol WW, Murray DW (2003) Real-time visual workspace localisation and mapping for a wearable robot. In: Proceedings of the 2nd IEEE/ACM international symposium on mixed and augmented reality, p 315. IEEE Computer Society

12. Fehr M, Furrer F, Dryanovski I, Sturm J, Gilitschenski I, Siegwart R, Cadena C (2017) Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery. In: 2017 IEEE International Conference on Robotics and automation (ICRA), pp 5237–5244. IEEE
13. Fiala M (2005) Artag, a fiducial marker system using digital techniques. In: Proceedings of the IEEE computer society conf. Computer vision and pattern recognition (CVPR'05), vol 2, pp 590–596 vol 2, https://doi.org/10.1109/CVPR.2005.74
14. Gao QH, Wan TR, Tang W, Chen L (2017) A stable and vaccurate marker-less augmented reality registration method. In: 2017 International conference on CYBERWORLDS 20-22 september 2017
15. Gao QH, Wan TR, Tang W, Chen L, Bing W, Zhang M (2017) An improved augmented reality registration method based on visual slam. E-Learning and Games, LNCS 10345:11–19
16. Garcia-Garcia A, Gomez-Donoso F, Garcia-Rodriguez J, Orts-Escolano S, Cazorla M, Azorin-Lopez J (2016) Pointnet: a 3d convolutional neural network for real-time object class recognition. In: International joint conference on neural networks, pp 1578–1584
17. Gross M, Pfister H (2007) Point-based graphics / Morgan Kaufmann
18. Guo Y, Sohel F, Bennamoun M, Wan J, Lu M (2015) A novel local surface feature for 3d object recognition under clutter and occlusion. Inf Sci 293:196–213
19. Hagbi N, Bergig O, El-Sana J, Billinghurst M (2011) Shape recognition and pose estimation for mobile augmented reality. IEEE Trans Vis Comput Graph 17(10):1369–1379
20. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: The IEEE conference on computer vision and pattern recognition (CVPR)
21. Henderson SJ, Feiner SK (2011) Augmented reality in the psychomotor phase of a procedural task. In: 2011 10th IEEE International Symposium on Mixed and augmented reality (ISMAR), pp 191–200. IEEE
22. Klein G, Murray D (2007) Parallel tracking and mapping for small ar workspaces. In: Proceedings of the 6th IEEE and ACM international symposium mixed and augmented reality, pp 225–234. https://doi.org/10.1109/ISMAR.2007.4538852
23. Krajnik T, Fentanes JP, Cielniak G, Dondrup C, Duckett T (2014) Spectral analysis for long-term robotic mapping. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp 3706–3711. https://doi.org/10.1109/ICRA.2014.6907396
24. Liarokapis F, Anderson EF (2010) Using augmented reality as a medium to assist teaching in higher education
25. Liu L, Cheng L, Liu Y, Jia Y, Rosenblum DS (2016) Recognizing complex activities by a probabilistic interval-based model. In: Thirtieth AAAI conference on artificial intelligence, pp 1266–1272
26. Liu Y, Nie L, Han L, Zhang L, Rosenblum DS (2016) Action2activity: recognizing complex activities from sensor data, pp 1617–1623
27. Liu Y, Nie L, Liu L, Rosenblum DS (2016) From action to activity: sensor-based activity recognition. Neurocomputing 181:108–115
28. McCormac J, Handa A, Davison A, Leutenegger S (2017) Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In: 2017 IEEE International Conference on Robotics and automation (ICRA), pp 4628–4635. IEEE
29. Mur-Artal R, Montiel JMM, Tardós JD (2015) Orb-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 31(5):1147–1163. https://doi.org/10.1109/TRO.2015.2463671
30. Murartal R, Tardos JD (2016) Orb-slam2: an open-source slam system for monocular stereo and rgb-d cameras
31. Newcombe RA, Fox D, Seitz SM (2015) Dynamicfusion: reconstruction and tracking of non-rigid scenes in real-time. In: The IEEE conference on Computer Vision and Pattern Recognition (CVPR)
32. Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, Kohi P, Shotton J, Hodges S, Fitzgibbon A (2012) Kinectfusion: Real-time dense surface mapping and tracking. In: IEEE International symposium on mixed and augmented reality, pp 127–136
33. Pang G, Qiu R, Huang J, You S, Neumann U (2015) Automatic 3d industrial point cloud modeling and recognition. In: Iapr international conference on machine vision applications, pp 22–25
34. Park N, Lee W, Woo W (2011) Barcode-assisted planar object tracking method for mobile augmented reality. In: 2011 International Symposium on Ubiquitous virtual reality (ISUVR), pp 40–43. IEEE
35. Prakhya SM, Liu B, Lin W (2015) B-shot: a binary feature descriptor for fast and efficient keypoint matching on 3d point clouds. In: IEEE/RSJ international conference on intelligence robots and systems (IROS), pp 1929–1934
36. Prince SJD, Xu K, Cheok AD (2002) Augmented reality camera tracking with homographies. IEEE Comput Graph Appl 22(6):39–45. https://doi.org/10.1109/MCG.2002.1046627
37. Reitmayr G, Langlotz T, Wagner D, Mulloni A, Schall G, Schmalstieg D, Pan Q (2010) Simultaneous localization and mapping for augmented reality. In: 2010 International Symposium on Ubiquitous virtual reality (ISUVR), pp 5–8. IEEE

38. Rosten E, Drummond T (2006) Machine learning for high-speed corner detection. Computer Vision–ECCV 2006:430–443
39. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: an efficient alternative to sift or surf. In: Proceedings of the international conference on computer vision, pp 2564–2571, https://doi.org/10.1109/ICCV.2011.6126544
40. Rusu RB, Cousins S (2011) 3d is here: Point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and automation (ICRA), pp 1–4. IEEE
41. Rusu RB, Marton ZC, Blodow N, Dolha M, Beetz M (2008) Towards 3d point cloud based object maps for household environments. Robot Auton Syst 56(11):927–941
42. Segal A, Haehnel D, Thrun S (2009) Generalized-icp. In: Robotics: science and systems, vol 2, p 435
43. Skrypnyk I, Lowe DG (2004) Scene modelling recognition and tracking with invariant image features
44. Stoyanov T, Magnusson M, Almqvist H, Lilienthal AJ (2011) On the accuracy of the 3d normal distributions transform as a tool for spatial representation. In: IEEE International conference on robotics and automation, pp 4080–4085
45. Strasdat H, Montiel J, Davison AJ (2012) Visual SLAM: Why filter? Image Vis Comput 30(2):65–77. https://doi.org/10.1016/j.imavis.2012.02.009
46. Szalavári Z, Gervautz M (1997) The personal interaction panel – a two-handed interface for augmented reality. Comput Graph Forum 16(3):C335–C346. https://doi.org/10.1111/1467-8659.00137
47. Tejani A, Tang D, Kouskouridas R, Kim TK (2014) Latent-class hough forests for 3D object detection and pose estimation. European Conference on Computer Vision (ECCV 20104), pp. 462–477
48. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: Sixth international conference on computer vision, 1998, pp 839–846. IEEE
49. Tombari F, Salti S, Stefano LD (2010) Unique signatures of histograms for local surface description. Lect Notes Comput Sci 6313:356–369
50. Tombari F, Salti S, Stefano LD (2013) Performance evaluation of 3d keypoint detectors. Int J Comput Vis 102(1-3):198–220
51. Tombari F, Stefano LD (2010) Object recognition in 3d scenes with occlusions and clutter by hough voting. In: Fourth pacific-rim symposium on image and video technology, pp 349–355
52. Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW (2000) Bundle adjustment — a modern synthesis. In: Vision algorithms: theory and practice, pp 298–372. Springer nature
53. Wang J, Suenaga H, Hoshi K, Yang L, Kobayashi E, Sakuma I, Liao H (2014) Augmented reality navigation with automatic marker-free image registration using 3-d image overlay for dental surgery. IEEE Trans Biomed Eng 61(4):1295–1304
54. Wen R, Yang L, Chui CK, Lim KB, Chang S (2010) Intraoperative visual guidance and control interface for augmented reality robotic surgery. In: 2010 8th IEEE International Conference on Control and automation (ICCA), pp 947–952. IEEE
55. Whelan T, Salas-Moreno RF, Glocker B, Davison AJ, Leutenegger S (2016) Elasticfusion: real-time dense slam and light source estimation. Int J Robot Res 45(14):1697–1716

Qing Hong Gao is a Master student at College of Electronic and Information, Xian Polytechnic University, China. His research interests are in Virtual Reality and Augmented Reality Technologies, especially SLAM and AR.

Tao Ruan Wan is a senior lecturer in Faculty of Informatics and Engineering at the University of Bradford, UK. His research interests are in computer simulation, virtual reality and augmented reality.

Wen Tang is a professor in Faculty of Science and Technology at the Bournemouth University, UK. Her research interests are interactive computer graphics, virtual reality and augmented reality and related applications.

Long Chen currently is a PhD student in Faculty of Science and Technology at the Bournemouth University, UK. His research interests are virtual reality and augmented reality and related medical applications.