IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 28, 2019, accepted December 10, 2019, date of publication December 16, 2019,
date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959809

A Hybrid 3D Registration Method of Augmented Reality for Intelligent Manufacturing

XIAN YANG, (Member, IEEE), JINGFAN YANG, HANWU HE, AND HEEN CHEN
School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510075, China

Corresponding author: Heen Chen (chheen@gdut.edu.cn)

This work was supported in part by the National Key Research and Development Program: Multi-Modal Natural Interaction Virtual Reality Open Experimental Teaching Environment under Grant 2018YFB1004902, in part by the Guangzhou Science and Technology Project under Grant 201802020011, in part by the Guangdong Province Science and Technology Program of China under Grant 2017B010110008 and Grant 2016A040403108, in part by the Guangdong Provincial Key Laboratory of Cyber-Physical Systems, and in part by the National and Local Joint Engineering Research Center of Intelligent Manufacturing Cyber-Physical Systems.

ABSTRACT Presently, core 3D registration technologies for augmented reality have problems like low accuracy and poor tracking stability in natural environments pertaining to mass customization and intelligent manufacturing, resulting in error display or poor visual performance. (1) A non-linear scale space was used to alleviate the problem associated with scale invariance, the relevant calculations and construction methods were studied. The adaptive non-maximal suppression method was examined, which reduced redundancy of ORB (Oriented FAST and Rotated BRIEF) feature points. (2) The method of combining nonlinear local descriptors with LK method is studied to improve the low stability problem against Light change by LK method only, and a forward-backward error detection method was studied to evaluate feature point tracking results. (3) The improved ORB and LK methods are used to track the target and achieve data fusion. Then, the fused data is voted and clustered by means of consistent voting, only the maximum number of clusters within the threshold is left as the final tracking result to realize 3D registration. Finally, the paper validates the proposed method through the natural environment dataset of open source. The dimensions of verification include challenging scales, rotation changes, perspective changes, motion blur, occlusion, and out-of-view object.

INDEX TERMS 3D registration, augmented reality, intelligent manufacturing, natural environment.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

With mass production being replaced by mass customization, the manufacturing process is becoming increasingly complicated [1]. Intelligent manufacturing that integrates human-machine interactions is uniquely advantageous as computer technologies are more and more closely integrated in the manufacturing process. However, human operators in intelligent manufacturing tend to entail standardization levels, lower efficiency and higher requirement for proficiency compared with machine operators. Augmented reality (AR) is a technology that integrates virtual information into the real world in order to achieve augmented information in the real world [2]. By seamlessly combining computer-generated virtual information like texts, images and animations with the real world, AR provides users with intuitive and effective visual presentations and interactions. With better suitability to human learning and cognitive processes, such a presentational approach allows users to get feedbacks of real and virtual information simultaneously, so that they can complete tasks in a better and more efficient way and become more adaptive to mass customization and intelligent manufacturing.

Although AR has been broadly used in many areas, most of its current applications only exist in laboratory environments and further development of factors like solution scalability, market acceptance and terminal users is needed. Technologically, AR is still not mature enough. Take the 3D tracking and registration technologies of AR as an example, the observers' view is unrestricted when in the manual operations under AR environment, that is, observers are able to view the objects from multiple angles before proceeding with further operations [3]. Therefore, even if observed objects were not moved, from the observer's perspective, they may seem to be in a natural environment involving multiple combinations

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Xu.

of changing scale, rotation, perspective distortion, occlusion, object out-of-view and motion blur. In addition, the environment background is unrestricted, where a variety of complicated interference objects may be gathered unintentionally. Apart from the natural environments, existing AR technology is also confronted with other problems like low accuracy of 3D registration, low tracking stability and poor serviceability, leading to error display of virtual information and poor visual performance, as exemplified by information jitter, which restrict the application of AR technology in many areas, especially the manufacturing field.

B. RELATIVE RESEARCH

To fulfill the alignment and stitching of virtual information with the real world, camera pose estimation is one of the most basic problems to be addressed. The entire process to solve camera pose, superimpose and render the virtual information is called 3D registration [4]. Depending on conditions required by registration, 3D registration can be classified into methods based on artificial markers, natural features and visual SLAM (Simultaneous Localization and Mapping).

As the most conventional 3D registration method in AR technology, the artificial marker-based method is extremely competitive in terms of computational efficiency, reliability and usability. Kato and Billinghurst [5] are the earliest researchers who fulfilled artificial marker-based 3D registration algorithm. The algorithm was developed into ARToolkit and became the most well-known open source project in AR field. ARToolkit solves camera pose through methods like grayscale thresholding, connecting components, contour extraction, marking edges, corner extraction and template matching. However, ARToolkit is also fraught with problems like low adaptability of grayscale thresholding, high false positive rate in correlation-based matching and reliance of processing time on the size of marker library. Fiala [6] proposed an ARTag marker system that helped alleviate these problems. Wagner *et al.* [7] fulfilled real-time execution of ARToolkitPlus on mobile devices. As artificial markers are not sufficiently compatible to the real environment, the artificial marker-based approach has gradually been shifted to an auxiliary tracking method for AR system in order to improve the overall tracking accuracy.

Given that using artificial marker-based method to register an object is subject to the limitations of markers, the natural feature-based method has gradually moved into the mainstream. The natural feature-based method utilizes natural features of an object, such as point, line, plane and model, as a basis to identify and localize the object and thus fulfill 3D registration. As only part of object features is required, the method can yield relatively higher computational efficiency and better anti-occlusion capability. Earlier natural feature-based approaches, such as those based on region features [8], feature points [9] and model features [10], prevalently have problems like poor real-time performance, low accuracy and high environmental reliance. Wagner *et al.* [7] and Wagner and Schmalstieg [11] combined SIFT with FERNS to fulfill a feature point-based 3D registration method on mobile devices, and later incorporated a template-matching-based algorithm to further improve its computational efficiency [12]. With the advent of feature point-based methods like FAST [13] and BRISK [14], and binary descriptors like BRIEF [15], the natural feature-based method has gained substantial improvement in real-time performance and has been increasingly utilized as a classical natural feature-based 3D object tracking method to address registration problems. Nevertheless, there are a few problems arising from the direct use of natural feature point-based methods: (1) There may be random errors in the poses estimation per frame, which resulting in jitter of virtual information; (2) Under some special circumstances involving with large inclination angles etc., it will lead to problems that cannot be registered.

With an ability to fulfill self-localization and environment mapping, the SLAM method can be used to solve camera pose and achieve 3D registration even in unknown environment. SLAM method is not capable of object recognition, it needs to be combined with other methods in order to perform 3D registration for specific objects. Given large amounts of computation, the SLAM method which is originally meant to build dense and semi-dense maps can hardly run in real time. Therefore, SLAM-based 3D registration methods are mostly those utilizing feature points to build sparse maps. Georg [16] proposed a PTAM method in 2007 to realize an AR environment within a small workspace. PTAM is the first method that splits processing of tracking and mapping in parallel threads and utilizes sparse point cloud and keyframe to update the maps, significantly boosting the performance of the method [17].

This paper presents researches on problems associated with the existing 3D registration methods integrating ORB and LK tracking such as poor accuracy and scale invariance in natural feature point matching, and poor robustness to illumination changes and occlusions in regions being tracked. The researches include three sections: (1) A non-linear scale space was built to improve scale invariance of ORB feature points, and an adaptive non-maximal suppression method was employed to remove redundant feature points, thereby increasing matching accuracy [18]. (2) With respect to LK-based tracking, a non-linear descriptor was used to improve the robustness of LK method to illumination changes and improve the tracking stability. (3) To address the drawback that only one method can be used at a time for tracking with the original ORB-LK method, an improved ORB-LK method was utilized to perform separate object tracking [19], data generated from which were fused via consensus voting and subsequently filtered to generate tracking results. The approach successfully compensated for the problems arising from separate use of ORB or LK method by multiple experiments, and we applied these studies to engineering applications, such as intelligence manufacturing and so on.

II. AN IMPROVED ORB FEATURE POINT MATCHING METHOD

A. ORB FEATURE POINT MATCHING

Rublee *et al.* [20] proposed ORB feature builded on the well-known FAST keypoint detector and the recently-developed BRIEF descriptor. As a highly efficient substitution for SIFT and SURF, the ORB feature point method yields better accuracy and significantly reduces computational complexity. Comprised of improved FAST feature points and BRIEF descriptors, the ORB method incorporates rotation information to FAST feature points after calculating the grayscale centroid, and the result is known as oFAST feature points [21] which are made scale invariant by building Gaussian pyramids. BRIEF (Binary Robust Independent Elementary Features) is an n-dimensional binary descriptor. The ORB method can be used to add orientation information to BRIEF, deriving the rBRIEF descriptor. The rBRIEF descriptor is a binary sequence on which the XOR operation can be performed to efficiently compare each bit of the descriptor. Therefore, ORB feature points embrace considerably high matching efficiency.

In order to improve matching accuracy of feature points, the ratio test method [22] was implemented for screening. The ratio test is defined by (1).

$$r = \frac{d_{\text{best}}}{d_{\text{better}}} \tag{1}$$

For a given keypoint, the ratio r of the matching distance of the most similar feature point d_{best} to the matching distance of the second most similar feature pointd_{better} should be larger than a preset threshold value $h$which is usually set as 0.7 or 0.8 [23], if the matching is to be considered successful. The ratio test is able to screen out those merely with a unique matching relationship while eliminating those with multiple similar matching points, thereby reducing the probability of mismatching.

B. NON-LINEAR SCALE SPACE

Although ORB feature points are rotationally invariant, they do not bear scale invariant properties. While the conventional approach utilizing Gaussian pyramid is viable within a certain range of scale changes, the ORB method fails when scale changes exceed a certain limit. Fig. 1(a) shows Gaussian blur results in different scales. Therefore, a non-linear diffusion method is utilized to build the scale space. In addition to providing a better performance in retaining feature information at edges and other locations, the non-linear diffusion method also transforms image features into spot-like patterns which are capable of retaining more image information even in relatively large-scale changes, as showed in Fig. 1(b).

Non-linear diffusion describes the process by which image grayscale value diffuses over time under the control of a certain diffusivity equation. Such method is usually expressed by a partial differential equation, as shown in (2).

$$\frac{\partial u}{\partial t} = div\left(c(x, y, t) \cdot \nabla u\right) \tag{2}$$

(a) Gaussian blur

(b) Nonlinear diffusion

FIGURE 1. ORB method and non-linear diffusion method.

where div denotes divergence, ∇ is gradient operator, c is diffusivity equation which is dependent on the local differential structure of the image, x and y are the x and y coordinates of the image, t is time (i.e. scale variable), and the larger the t, the smoother the image. In order to reduce the loss of edge information, c is defined as Equation (3).

$$c(x, y, t) = g\left(|\nabla u_\sigma(x, y, t)|\right) \tag{3}$$

where ∇u_σ denotes the gradient of the original image u that has undergone Gaussian filtering. g is expressed by (4).

$$g_3 = \begin{cases} 1, & if\ |\nabla u_\sigma|^2 = 0 \\ 1 - \exp\left(-\frac{3.315}{(|\nabla u_\sigma|/k)^8}\right), & if\ |\nabla u_\sigma|^2 > 0 \end{cases} \tag{4}$$

where k is the factor controlling the degree of diffusion. Differences in k value determine which edges are to be retained or smoothed. The value of k can be determined based on specific characteristics and requirements associated with the image of interest, or directly on experience of use. Here, k is set as 0.7 based on related literature [23]. As the analytical solution cannot be directly obtained from (2), the numerical method needs to be used to approximate the true value in order to obtain an approximation of the true value. FED (Fast Explicit Diffusion) is a numerical scheme. Combining semi-implicit and explicit approaches, the FED scheme performs M outer FED cycles and n inner cycles with varying step size t_j for each cycle. t_j is defined as (5).

$$t_j = \frac{t_{\max}}{2\cos^2\left(\pi\,\frac{2j+1}{4n+2}\right)} \tag{5}$$

where $t_{\max}$ is the maximal step size that does not violate the stability condition of FED, that is, for a 2D image, $t_{\max} = 0.25$. The stop time of each outer cycle is expressed by (6).

$$\theta_n = \sum_{j=0}^{n-1} t_j = t_{\max}\frac{n^2 + n}{3} \tag{6}$$

The discretized form of (2) can be explicitly expressed by (7),

$$\frac{u^{i+1} - u^i}{t} = A(u^i)u^i \tag{7}$$

where t is the step size of the n-th cycle, and A is the matrix representation for the information of diffusivity equation. As u denotes the image, $A(u^i)u^i$ can be described by (8),

$$A(u)u = \nabla(g\nabla u) = \frac{\partial}{\partial x}(gu_x) + \frac{\partial}{\partial y}(gu_y) \tag{8}$$

where g is the diffusivity equation, the modulus of the gradient of image u is $\|\nabla u\| = \sqrt{u_x^2 + u_y^2}$. The derivative of image u in x- and y-direction, respectively denoted by u_x and u_y, can be approximated using forward difference method, as shown in (9).

$$\begin{cases} u_x \approx D_x^+ u = u(x+1, y) - u(x, y) \\ u_y \approx D_y^+ u = u(x, y+1) - u(x, y) \end{cases} \tag{9}$$

That is followed by the approximation of the second derivative. Here, u_{xx} is not directly calculated, $\frac{\partial}{\partial x}(gu_x)$ is instead calculated by using forward difference approximation first and backward difference approximation next, as shown in (10).

$$\frac{\partial}{\partial x}(gu_x) \approx D_x^-(gD_x^+ u) \tag{10}$$

The matrix $P = gD_x^+ u$, where the $D_x^+ u$ is the approximation described above, then the backward difference method can be described by (11).

$$D_x^- P = P(x, y) - P(x-1, y) \tag{11}$$

The same method can be used to solve $\frac{\partial}{\partial y}(gu_y)$, thus the solution to $A(u)u$ can be found. The solution to the $i+1$-th layer can be directly obtained from (12).

$$u^{i+1} = \left(I + \Delta t A(u^i)\right) u^i = u^i + \Delta t \left[\frac{\partial}{\partial x}(gu_x) + \frac{\partial}{\partial y}(gu_y)\right] \tag{12}$$

where I is identity matrix, the step size of each of the n inner cycles is t_j, then Equation (12) can be rewritten into the form of (13).

$$u^{i+1,j+1} = \left(I + t_j A(u^i)\right) u^{i+1,j}, \quad j = 0, 1, \ldots, n-1 \tag{13}$$

Unlike the method described in the literature [23], as ORB feature point matching is directly used for feature point extraction, there is no need here to construct a series of groups and layers in order to use approaches similar to the difference of Gaussian in SIFT to calculate local extrema. Instead, we only need to construct a non-linear spatial pyramid with appropriate layers, extract ORB feature points from each layer and then incorporate them into feature point sets in different scale spaces.

C. ADAPTIVE NON-MAXIMAL SUPPRESSION

The adaptive non-maximal suppression method is used to filter the extracted ORB feature points in order to remove those with non-local extrema and obtain those having local extrema. This procedure allows feature points to be more evenly distributed in the image space. The adaptive non-maximal suppression method compares the response value of each feature point, only retaining those having the maximum feature response values within a neighborhood in radius r. Assume the radius $r = 0$, the feature point sequence of suppressed non-extrema is null, and then add feature points conforming to the condition of extrema to the sequence until the number of feature points in the sequence reaches the required quantity. The feature response values of feature points in the sequence are arranged in a descending order, consistent to those of the added ones. To solve the sequence of a feature point, for feature point $\boldsymbol{x}_i = (x_i, y_i)^T$ on image I, the minimum suppression radius r_i of feature point $\boldsymbol{x}_i$ is defined as (14).

$$r_i = \min_j \left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|, \quad where\, f(\boldsymbol{x}_i) < cf(\boldsymbol{x}_j),\ \boldsymbol{x}_j \in I \tag{14}$$

where I is the original image, f is the function of feature response value, $f(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_j)$ are response values of feature points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, c is a steady value smaller than or equal to 1, and it is used to ensure the response value of point $\boldsymbol{x}_j$ in the neighbourhood $f(\boldsymbol{x}_j) > f(\boldsymbol{x}_i)$. c is usually set as $c = 0.9$.

As the minimum suppression radius of the global-maximum feature point is infinitely large, the point, in any case, will not be suppressed. The minimum suppression radiuses of the remaining feature points are calculated via Equation (14). As a result, all feature points and their corresponding minimum suppression radiuses are obtained. Then, to meet the requirement on the number of feature points, k feature points with the largest minimum suppression radius r_i are retained, while others are discarded. A comparison prior (Fig. 2(a)) to and after (Fig. 2(b)) the adaptive non-maximal suppression is shown in Figure 2.

Assume P is the set of feature points, n is the number of feature points in set P, NN is data structure to calculate the minimum suppression radius, $p_1, p_2, \ldots, p_n$ are feature points arranged according to the descending order of response values of feature points in P, $r_1, r_2, \ldots, r_n$ are the minimum suppression radiuses corresponding to $p_1, p_2, \ldots, p_n$, R is the set of feature points and their minimum suppression radiuses. The process of the adaptive non-maximal suppression algorithm can be described by Table 1.

The key that influences the time complexity of this algorithm is the efficiency of calculating the NN data structure with the minimum suppression radius in Step (4). If a direct comparison of feature point p_i with that in NN is conducted to obtain the minimum suppression radius r_i, then the time complexity will be $O(n^2)$. However, such time complexity is excessively high for scenarios involving large numbers of feature points, more efficient algorithms should thus be considered. Chan [24] developed a nearest-neighborhood algorithm based on a randomized search tree data structure, which is capable of supporting nearest neighborhood calculation with a time complexity of $O(\log n)$. The method only requires a preprocessing time with a time complexity of $O(n \log n)$, a spatial complexity of $O(n)$ and a query time of $O(\log n)$.

(a) Prior to the adaptive non-maximal suppression

(b) After the adaptive non-maximal suppression

FIGURE 2. **Comparison of adaptive non-maximal suppression.**

TABLE 1. **Adaptive non-maximal suppression.**

Adaptive Non-maximal Suppression
Pre-calculation:
(1) Arrange P in a descending order based on feature response values to get $p_1, p_2, \ldots, p_n$
(2) Add p_1 to NN
(3) Add p_1 and its corresponding minimum suppression radius (p_1, ∞) to the result R
For each $p_i \in P$:
(4) Calculate point p_i via NN and the minimum suppression radius in NN r_i
(5) Add p_i and its corresponding minimum suppression radius (p_i, r_i) to the result R
(6) Add p_i to NN
Repeat the cycle:
Return:
(7) Return k feature points in R with descending minimum suppression radiuses.

When using the method as NN data structure, only $O(\log n)$ time complexity is required for each calculation of the minimum suppression radius. Therefore, the overall time complexity of the algorithm is reduced from the quadratic $O(n^2)$ to the sub-quadratic $O((\log n)^2)$. In this way, the computational time is significantly reduced.

D. CONTRAST EXPERIMENT

In this experiment, a natural-environment object tracking data set proposed by Liang *et al.* [25] was used. A comparison between the improved ORB method and its original counterpart was conducted. The improved ORB method slightly outperforms the original one along with changing scales (Fig.3(a) and Fig.3(b)). This may be attributable to the fact that the ORB integrated with non-maximal suppression is more difficult in extracting feature points of the tracked object due to scattered feature points, leading to less evident performance of the improvement.

In natural environments that are likely to involve complex backgrounds and numbers of interfering feature points, the improved ORB method yields a greater success rate in tracking than the original method (Fig.3(c) and Fig.3(d)). While enabling the feature point extraction to be more scattered in the image space, adaptive non-maximal suppression also suppresses the influences of other interfering feature points in the environment, thereby increasing the stability of feature points

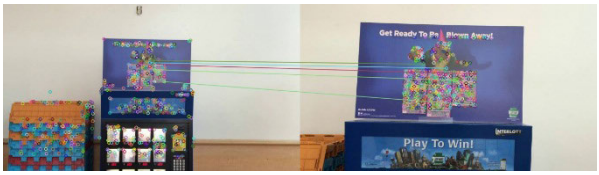
In other scenarios combined with occlusion (Fig.3(e) and Fig.3(f)), object out-of-view (Fig.3(g) and Fig.3(h)), slight blur (Fig.3(i) and Fig.3(j)) and perspective distortion (Fig.3(k) and Fig.3(l)), the overall stability of the improved ORB feature points, as well as their adaptability to environmental changes, is higher than the original ORB method.

III. AN IMPROVED LK TRACKING METHOD

A. NON-LINEAR LOCAL DESCRIPTOR METHOD

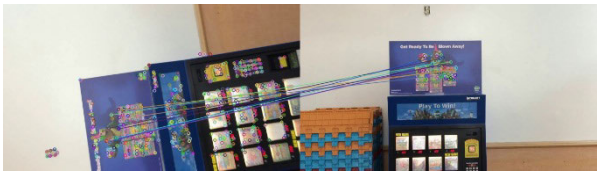
1) WEAKNESS OF LK METHOD

Originally proposed by Lucas and Kanade [26], the Lucas-Kanade (LK) method is widely used to solve image registration in continuous image frames. To solve a problem with the LK method is essentially a problem of finding the optimal solution. Lucas-Kanade method is vulnerable to changes in light and has a relatively narrow range of convergence. To obtain a positive computational speed and tracking accuracy, we propose to first use the local descriptor method, namely, the descriptor field (DF) [27], to give a description of the image, and then use the inverse compositional (IC) method [28] to find the solution, thereby increasing the stability of the IC method. Both LK and IC optimization methods utilize the image grayscale value of a corresponding pixel to find approximation when calculating the sum of squared error. Despite of its speediness and simplicity, direct use of image grayscale values has the following drawbacks: (1) Vulnerability to noise values. Gaussian filtering can be used to reduce noise, but at the same time, it will also smooth image information, leading to a reduction in convergence of the IC algorithm; (2) Vulnerability to light changes. Changes in grayscale value of the same pixel between two continuous frames would also lead to a reduction in convergence of the



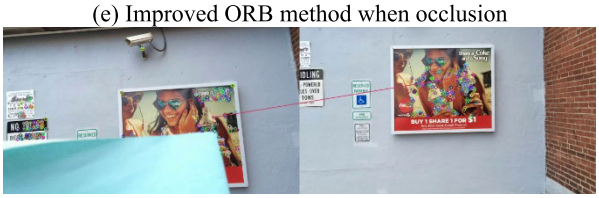
(a) Improved ORB method

(b) Original ORB method


(c) Improved ORB method in natural environments

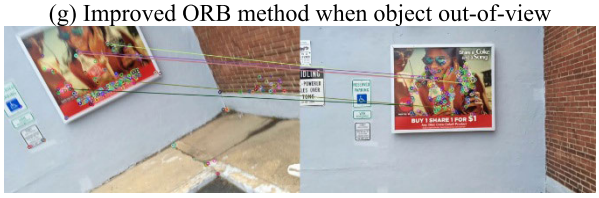

(d) Original ORB method in natural environments

(e) Improved ORB method when occlusion



(f) Original ORB method when occlusion

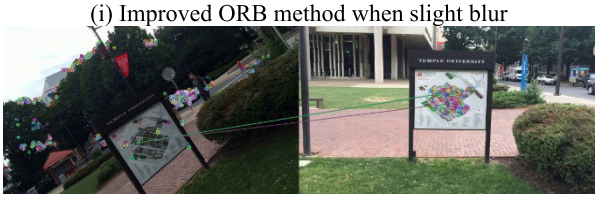
(g) Improved ORB method when object out-of-view

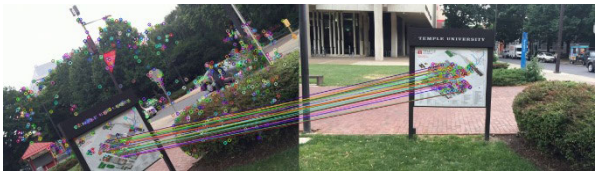


(h) Original ORB method when object out-of-view

FIGURE 3. Comparison between improved ORB and original ORB method.


(i) Improved ORB method when slight blur


(j) Original ORB method when slight blur


(k) Improved ORB method when perspective distortion

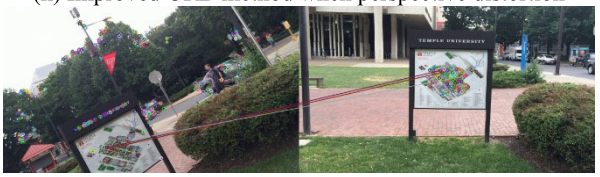

(l) Original ORB method when perspective distortion

FIGURE 3. *(Continued.)* Comparison between improved ORB and original ORB method.

IC algorithm, and a light change also violates the assumption of grayscale value continuity; (3) the IC method can only be used to track relatively small movements.

2) LOCAL DESCRIPTION OF THE DESCRIPTOR FIELD

While possessing a certain level of resistance to image degradation caused by Gaussian filtering, DF is highly resistant to light changes. Local jet [29] was considered to be used for image description. It is widely used as a local image descriptor, as described by (15), where the filter $f_1, f_2, \ldots, f_n$ is generally a dimensional Gaussian derivative kernel, and the original image, after being described, is converted into an image following the n-dimensional Gaussian derivative kernel transformation.

$$D(I, \boldsymbol{x}) = [[f_1 I(\boldsymbol{x})], [f_2 I(\boldsymbol{x})], \ldots, [f_n I(\boldsymbol{x})]]^T \tag{15}$$

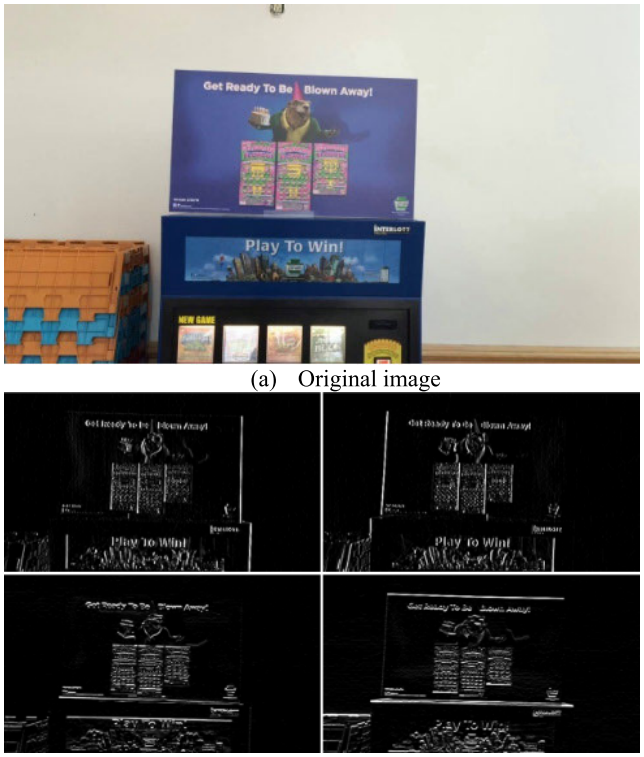
Based on (15), the "+" and "−" operations are introduced to derive the DF, as defined by (16).

$$D(I, \boldsymbol{x}) = \left[[f_1 I(\boldsymbol{x})]^+, [f_1 I(\boldsymbol{x})]^-, \ldots, [f_n I(\boldsymbol{x})]^+, [f_n I(\boldsymbol{x})]^-\right]^T \tag{16}$$

where the "+" and "−" operations are depicted by (17) and (18).

$$[x]^+ = \begin{cases} x, & if\ x \geq 0 \\ 0, & otherwise \end{cases} \tag{17}$$

$$[x]^- = [-x]^+ \tag{18}$$



(a) Original image

(b) the result after first-order DF descriptor processing

FIGURE 4. First derivation of DF descriptor.

Gaussian filter can easily over-smooth image features, leading to a difficulty in convergence with the IC method. On the other hand, while calculating Gaussian derivative kernels, local jet can result in positive and negative values that cancel each other out, causing an inability to retain the original feature information of the image in changing light conditions. In that case, it is still challengeable for the IC method to achieve convergence. DF, however, splits an image processed by Gaussian kernel into an image that only contains positive spatial domain and an image that only contains negative spatial domain, and utilizes the IC method to perform separate calculations on positive and negative spatial domains in a multi-dimensional space. The approach not only retains the original feature information of the image with a certain degree of light changes, but also reduces image degradation following Gaussian filtering. Fig.4(b) shows the result after first-order DF descriptor processing.

B. IMPROVEMENT OF IC OPTIMIZATION METHOD

1) COMBINATION OF IC METHOD AND DF ESCRIPTOR

The way to combine IC method and DF descriptor is shown in Figure 5. First, DF descriptor is separately applied to the image of the current frame and the template image of the previous frame to obtain the multi-dimensional DF descriptor spaces for the input image and the template image, as denoted by $D(I, \boldsymbol{x})$ and $D(T, \boldsymbol{x})$, respectively. Next, *Hessian* matrix and error image are separately computed on each of the same dimension, and the sum of the Hessian matrices and error

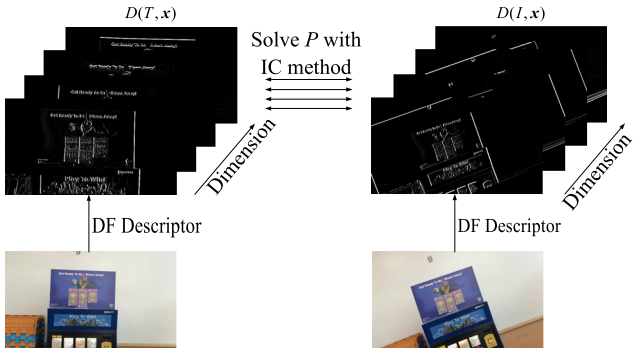


FIGURE 5. Combination of IC method and DF descriptor.

images is calculated. Finally, the solution the transformation relation $\boldsymbol{p}$ can be accessible.

To further expand the tracking range of the IC method, this paper also constructed a Gaussian pyramid to find solutions through coarse-fine iterations from the top. If the tacking threshold value is obtained in advance during the iteration, then the tracking will be considered successful and the computation will be terminated. To strike a balance between efficiency and effectiveness, this paper constructed a three-layer Gaussian pyramid, in which σ is set as 1.6.

2) BILINEAR INTERPOLATION

After undergoing the plane transformation $\boldsymbol{p}$, the coordinates of the image $W(\boldsymbol{x}, \boldsymbol{p})$, in most cases, will fall into non-integer values. A conventional approach is to find a pixel having the integer coordinates that are most close to these non-integer coordinates, and use the grayscale value of the pixel as an approximation for the grayscale value of the non-integer coordinate pixel. In order to improve the accuracy of grayscale value to accelerate iteration by increasing only a fraction of computational burden, this paper selected the bilinear interpolation method to obtain the grayscale values of non-integer coordinate pixels.

As shown in Figure 6, suppose we need to obtain the grayscale value of the non-integer coordinates $q = (x, y)$ in the image coordinate system, and the pixel is located between four integer coordinates, namely, $P_{11} = (x_1, y_1)$, $P_{12} = (x_1, y_2)$, $P_{21} = (x_2, y_1)$ and $P_{22} = (x_2, y_2)$, with $x_1 \le x \le x_2$ and $y_1 \le y \le y_2$. First, apply bilinear interpolation in the x-direction, as shown in (19) and (20), where f is a function for obtaining grayscale values.

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(P_{11}) + \frac{x - x_1}{x_2 - x_1} f(P_{21}) \tag{19}$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(P_{12}) + \frac{x - x_1}{x_2 - x_1} f(P_{22}) \tag{20}$$

Next, interpolate values in the y-direction with the same method, as shown in (21), to obtain final results of the bilinear interpolation.

$$f(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - x_1}{y_2 - y_1} f(x, y_2) \tag{21}$$

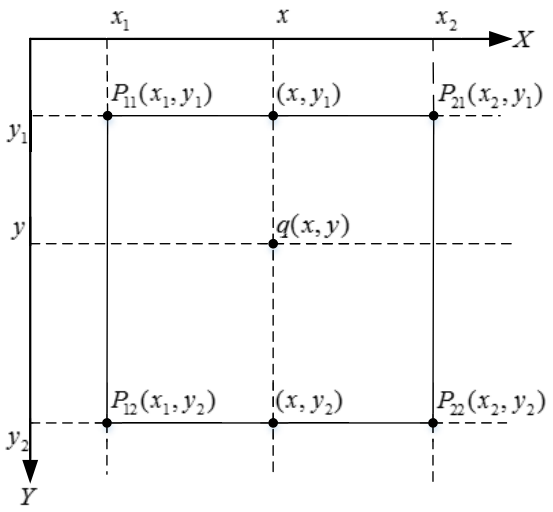


FIGURE 6. Bilinear interpolation.

Despite of the simplicity of bilinear interpolation, it usually requires only 5-10 iterations to converge towards the present error threshold value by combining the IC method with bilinear interpolation in actual experiments involving small movements. Comparatively, it usually requires 10-20 iterations to reach the preset error threshold value when directly calculating the grayscale value of the peripheral pixel.

IV. HYBRID TRACKING AND REGISTRATION METHOD BASED ON CONSENSUS VOTING

A. FEATURE POINT MATCHING, TRACKING AND DATA FUSION

Assume that the feature point obtained from a target object via ORB feature point-based training during the offline stage is $O = \{o_1, o_2, \ldots, o_n\}$, the feature points extracted from the t-th frame and successfully matched with O by using ORB feature point method during the online phase is $P_t = \{x_1^t, x_2^t, \ldots, x_m^t\}$, and the matching relation between P_t and O is $M_t = \{m_1^t, m_2^t, \ldots, m_m^t\}$, where each matching relation is comprised of the feature points of P_t and their corresponding matches in O. As such, use the ORB feature point method to obtain a point pair consisting of 2D and 3D feature points of the target object in each frame.

Given the $t-1$-th frame has P_{t-1} feature points that have been successfully matched, the improved LK method is applied to track the P_{t-1}. After discarding the feature points involving detected tracking failures, the image of t-th frame, along with the tracking result of P_{t-1}, denoted by $T_t = \{x_1^{t-1}, x_2^{t-1}, \ldots, x_i^{t-1}\}$, and the corresponding $t-1$-th frame, along with the matched feature point $M'_{t-1} = \{m_1^{t-1}, m_2^{t-1}, \ldots, m_i^{t-1}\}$, can be obtained, where T_t is a subset of P_{t-1}. Therefore, we can obtain pairs of 2D and 3D points with the tracking method. In that process, P_t and T_t are obtained separately and independently from each other. After obtaining P_t and T_t, data fusion is subsequently performed.

As the same feature point may simultaneously exist in P_t and T_t, redundant feature points may be introduced if we simply merge them into a new set or a new matching relation of feature points. Therefore, during data fusion, we can determine whether there exists a matching relation that involves the same feature point in O through M_t and M'_{t-1}. After that, the feature point set $N_t = \{n_1^t, n_2^t, \ldots, n_k^t\}$ and its matching relation with $OL_t = \{l_1^t, l_2^t, \ldots, l_k^t\}$ can be obtained.

B. CONSENSUS COMPUTATION

Under different camera angles, a target to be tracked may present changes in scale, rotation and projective transformation in its projected image. However, for the feature point O of the target, it is impossible to obtain the position of each point on the image of the current frame when the changing conditions are unknown. Although the position of each individual point can arbitrarily be changed, a consensus has been reached between every two feature points, which remains invariant despite of any change in the target [30]. Therefore, the goal is to find a consensus relation applicable to the majority of the feature points N_t in the current frame, utilize the relation to perform a voting-based estimation of each point, and screen out and eliminate feature points based on voting results. Assume each feature point O only experiences changes in scale and rotatioal angle, for a point n in N_t and a match l in the matching relation L_t, their voting function $h(n, l)$ is defined as (22).

$$h(n, l) = n - s \cdot Ro_l, h(n, l) \to R^2 \tag{22}$$

where l is the index between feature point n and its matched feature point in O, o_l is the position of the relative coordinates for the feature point in O with an index of l. s is the scale parameter for controlling scale invariance; R is the rational matrix for controlling rotational invariance. Voting is performed on all points in N_t, which generates the voting set V for all feature points, as shown in (23).

$$V = \{h(n_i, l_i)\}_{i=1}^{k} \tag{23}$$

1) SCALE CONSENSUS COMPUTATION

For scale changes, assume the subscript indices of a random feature point in N_t are i and j, where $i \neq j$. It can be considered that the proportion of the Euclidean distance between every two pairs of feature points o_{l_i} and o_{l_j} on the target to the Euclidean distance in N_t between feature points n_i and n_j is consistent. Assume $n^{i,j} = n_i - n_j$ and $o^{i,j} = o_{l_i} - o_{l_j}$, then the scale distribution set of all combinations D_s can be described by (24).

$$D_s = \left\{ \frac{\left\| n^{i,j} \right\|}{\left\| o^{i,j} \right\|}, i \neq j \right\} \tag{24}$$

Organize D_s in an ascending order, and obtain the median in order to reduce the influence of outliers, that is, $s = \text{median}(D_s)$.

2) ROTATION CONSENSUS COMPUTATION

A similar approach is utilized to address rotational changes. It can be considered that the difference between the angle of any two pairs of feature points on the target and that of any two pairs of feature points in N_t is a constant value. The rotation matrix R in (22) can be described by (25),

$$R = \begin{pmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{pmatrix} \tag{25}$$

where a is a rotation angle. Let $n_x^{i,j}$ and $n_y^{i,j}$ be the difference values of $n^{i,j}$ in x and y directions, similarly, $o_x^{i,j}$ and $o_y^{i,j}$ be the different values of $o^{i,j}$ in x and y directions. Then between $n^{i,j}$ and $o^{i,j}$, the rotation angle $a_{i,j}$ can be defined by (26).

$$a_{i,j} = \text{atan2}(n_x^{i,j}, n_y^{i,j}) - \text{atan2}(o_x^{i,j}, o_y^{i,j}) \tag{26}$$

The schematic diagram for the calculation of $a_{i,j}$ is shown in Figure 7, where $0^\circ \leq a_{i,j} < 180^\circ$.

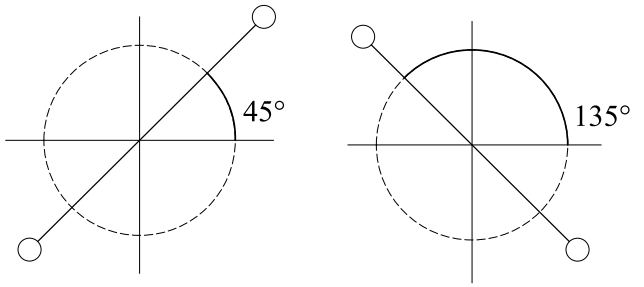


FIGURE 7. Calculation of angle of rotation.

The distribution of rotation angle differences for all combinations, denoted by D_a, is expressed by (27).

$$D_a = \{a_{i,j}, i \neq j\} \tag{27}$$

Similarly, the median of D_a is obtained via $a = \text{median}(D_a)$ to have access to the rotation matrix R.

3) VOTING COMPUTATION

First, scale and rotation consensus parameters, denoted by s and R, are calculated by using the matched feature point N_t, which helps derive the voting function $h(n, l)$. After that, apply h to vote on each feature point in N_t, deriving the voting set V. When calculating D_s and D_a, the scale distance and rotation angle of each feature point pair on the object both remain constant in each round of voting calculation. Therefore, these values can be calculated prior to training of the object's feature points and there is no need to repeat the calculation in each voting round. In V, each voting result is a 2D vector of coordinates, and all similar voting results could form a cluster, as shown in Figure 8. The majority of correct voting results would point to closely adjacent regions and form a cluster, while incorrect results would direct to other incorrect regions and form a variable number of other clusters.

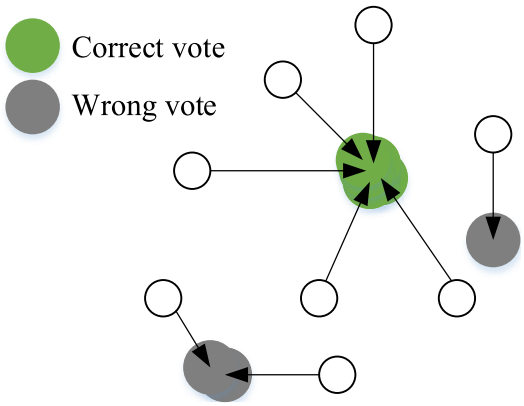


FIGURE 8. Clustering of vote.

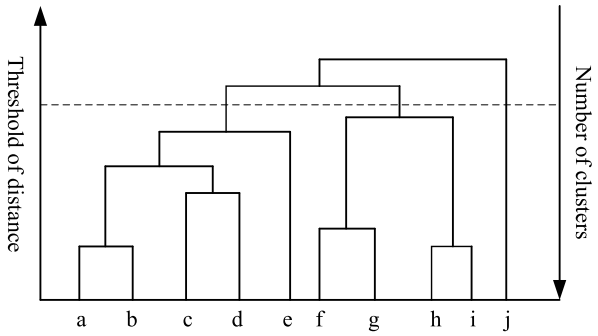


FIGURE 9. Dendrogram.

C. CLUSTERING OF VOTE

Voting results $V = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\}$ are comprised of N voting results, where $\boldsymbol{h}_n = (h_x^n, h_y^n)^T$ is a 2D vector. There are three commonly used methods for calculating the distance between clusters, which are single link, full link, and even link. In this paper, single link method was used because of improving the speed of clustering under limited conditions, according to the single link method, the hierarchical agglomerative clustering (HAC) method is used [31]. The basic idea of the HAC method is to join each individual point from the bottom to the top until the last point, eventually forming up a cluster of dengrogram. Assume that the voting results V are divided by HAC into k clusters $C_1, C_2, \ldots, C_k$, where any one of the cluster C_i cannot be vacant, while the intersection of any two clusters C_i and C_j must be vacant, with $i \in k, j \in k, i \neq j$. The similarity between any two points is measured by Euclidean distance; let the similarity measurement operation w be $w(h_i, h_j) = \|h_i - h_j\|$. First, let each point be a separate cluster, then we can obtain the cluster $C_1, C_2, \ldots, C_N$; and calculate the distance D between cluster, where $i \in N, j \in N$, as shown in (28).

$$D = \{d(C_i, C_j), i \neq j\} \tag{28}$$

Join the clusters with the shortest distances, denoted by C_a, C_b, into one cluster, denoted by $C_{a,b}$, then we can obtain the cluster $C_1, \ldots, C_{a,b}, \ldots, C_N$. Continue to calculate the

distance between the clusters and repeat the above procedure until only one cluster $C_{1,2,\ldots,N}$ is obtained. At this time, a dendrogram structure, as shown in Figure 9, can be obtained.

The root node of the dendrogram is a set of clusters consisting of all points in V, the more layers there are by moving from top to bottom, the greater the number of clusters. The cluster at the i-th layer is an aggregate of clusters at the $i+1$-th layer. Therefore, different numbers of clusters can be obtained when different layers are selected. The depth of each layer denotes the distance between its sub-clusters. The lager the number of layers and clusters, the shorter the distance between clusters; to the contrary, the smaller the number of layers and clusters, the longer the distance between clusters.

V. EXPERIMENT AND ANALYSIS

A. DATASET AND EXPERIMENTAL PLATFORM

Originally proposed by Liang *et al.* [25], a natural-environment object tracking dataset is made up of 30 object tracking test sets, with each containing 7 videos with a resolution of 1080∗720. Each video has 501 frames, the first of which is a template image used for tracking initialization. In this research, the videos were compressed to a resolution of 640∗360 before the tracking experiment. Each video corresponds to 6 different challenging factors for object tracking, as shown in Table 2.

TABLE 2. Tracking challenging factor.

Tracking item	Content
Scale change	Relatively large changes in distance between the camera and the object
Rotation change	Select camera with the same distance between camera and the tracking target
Perspective distortion	Relatively large changes in the camera's perspective
Motion blur	Motion blur caused by rapid movement of camera
Object occlusion	Part of the tracking object is occluded
Object out-of-view	Part of the object is out of view

Among the 30 test sets, 6 groups named Sunoco [0.59000, Snack [0.64000, Map-2[0.67000, Map-3[0.61000, Woman [0.72000 and Pizza [0.519] were selected in this section. The numbers in the "[]" are difficulty coefficients labeled by the author, the higher a number, the more difficult for the test set to be tracked.

The hardware platform of the experiment was comprised of the CPU Intel(R) Core (TM) i5-8300H @2.30GHz, 8MB L3 cache and 8G RAM, which supports 4 cores and 8 threads. The software platform was comprised of the Windows 10 system, Visual Studio 2017 Community, Open CV2.4.13 and Eigen 3.3.4.

B. EXPERIMENTAL PROCEDURES

The overall process of the proposed hybrid tracking and registration algorithm is shown in Figure 10. The algorithm can be divided into three parts: improved ORB method, improved LK tracking method, and consensus voting and data fusion.

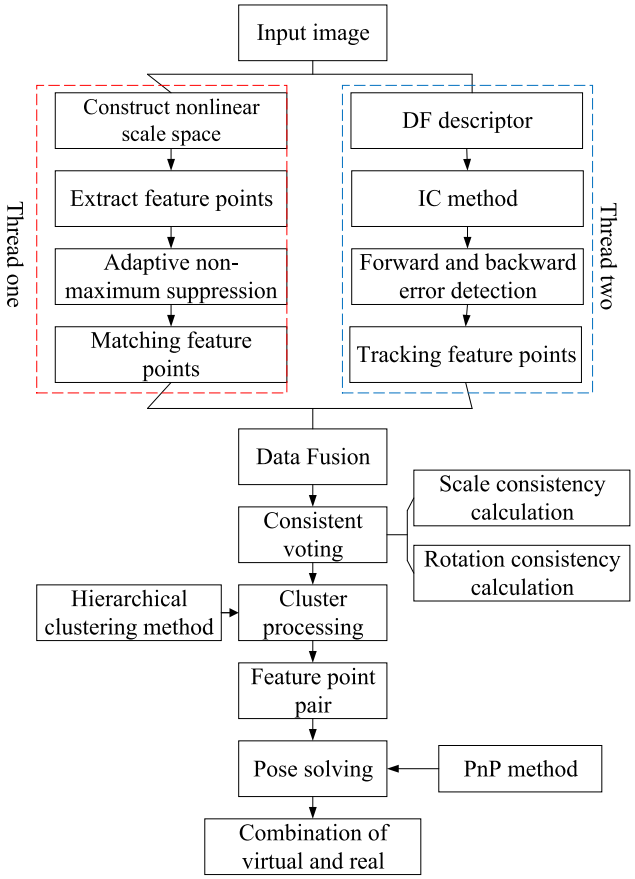


FIGURE 10. Process of hybrid tracking and registration method.

The ORB method and the LK method are calculated in two separate threads in parallel. Any computational thread that has completed its calculation needs to synchronize data and wait for the completion of the other thread before proceeding with the next step.

C. EVALUATION METRIC

Tracking results were evaluated in two aspects: image alignment error and homography matrix error. Image alignment error is mainly used to evaluate the difference between the tracking area of the image and the real area. Homography matrix error is mainly used to evaluate the homography transformation between consecutive frames. Compared with the image alignment error, homography matrix error is more competitive in reflecting the real tracking status of camera pose. Therefore, the lower the homography matrix error, the better the actual display effect of the registration.

Image alignment error is the mean-square error between real and tracking locations of the object's corner point on a plane, as shown in (29).

$$e_{AL} = \sqrt{\frac{1}{4}\sum_{i=1}^{4} \left\| \boldsymbol{x}_i - \boldsymbol{x}_i^* \right\|_2^2} \tag{29}$$

where $\boldsymbol{x}_i = (x_i, y_i)^T$ is the location coordinates of the tracked point, $\boldsymbol{x}_i^* = (x_i^*, y_i^*)^T$ is its real location coordinate. Define a threshold of image alignment error as t_p, set $t_p = 5$, tracking is considered successful when e_{AL} is smaller than t_p.

Homography matrix error is the error between the true value and the tracking value of the homography matrices between template image and current-frame image, as shown in (30).

$$S(T^*, T) = \frac{1}{4}\sum_{i=1}^{4}\left\| c_i - \left(T^* T^{-1}\right) c_i \right\|_2 \tag{30}$$

where T is the tracking value of the current-frame homography matrix, T^* is the true value of its corresponding homography matrix, $\{c_i\}_{i=1}^4 = \{(-1,-1)^T, (1,-1)^T, (-1,1)^T, (1,1)^T\}$; when matrix T is the same asT^*, then $S(T^*, T) = 0$. Define a threshold of homography matrix error as t_s, set $t_s = 10$, tracking is considered successful when $S(T^*, T)$ is smaller than t_s.

The overall evaluation process uses the first frame of each video as the initialization frame and the template image of homography matrices. Relevant approaches pertaining to the ORB and proposed methods are separately used in tracking. For the ORB-LK scheme, the ORB method is utilized for initialization and the LK method is for tracking. If the tracking fails, then initialization will be performed again using the ORB method.

D. EXPERIMENTAL RESULTS AND ANALYSIS

Part of the tracking results for the 6 test sets are presented in the following paper.

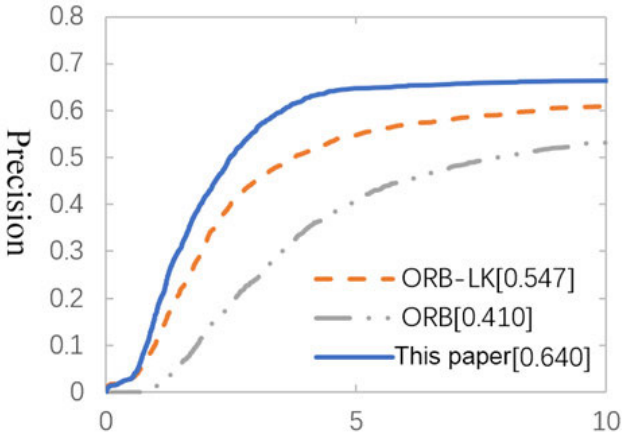
FIGURE 11. Scale change.

1) SCALE CHANGE

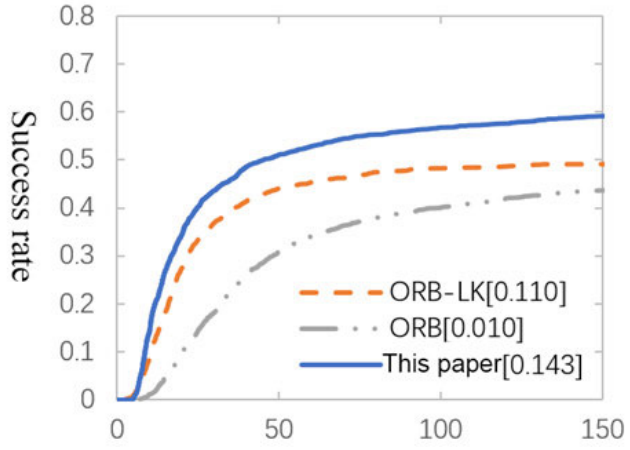
Tracking results pertaining to scale change are shown in Figure 11. Although tracking can be successfully implemented using the Gaussian pyramid-based ORB method in small scale changes, the ORB method rapidly deteriorates and eventually becomes ineffective as scale changes continue to rise.

For ORB-LK method, the LK method plays a dominant role. Therefore, in most cases, the ORB-LK method can successfully track scale changes based on the relation of consecutive frames, without deteriorating along with changes in scale. The LK method may produce random errors that could lead to a failure in tacking. At this circumstance, if the ORB method fails to reinitialize, the overall ORB-LK method cannot be restored.

Compared with the original ORB method, the proposed ORB method constructed upon a non-linear scale space has better performance in terms of scale invariance; when complemented by the LK method, the proposed method can track feature points in scale changes that cannot be tracked by the ORB method only. Therefore, the proposed method outperforms the other two in terms of scale changes. For extremely large-scale changes, as shown in Map-3 test set in Figure 11, all methods fail to maintain successful tracking.



(a) Comparison of image alignment errors in scale changes



(b) Comparison of homography matrix errors in scale changes

FIGURE 12. Error comparison of scale change.

The image alignment errors and homography matrix errors with respect to the three methods in scale changes are shown in Figure 12. The accuracy and success rate of the proposed method are 0.640 and 0.143, respectively, both of which are higher than those of the other two methods. With accuracy

and success rate being the same, the image alignment errors and homography matrix errors of the ORB-LK and the proposed method are both lower than those of the ORB method.

2) ROTATION CHANGE

Tracking results pertaining to rotation change are shown in Fig. 13. The three methods all perform fairly well in scenarios of rotation change. Given the existence of low-quality feature points in the Snack and Woman test sets, the ORB method produces relatively high tracking errors in these test sets.

FIGURE 13. Rotation change.

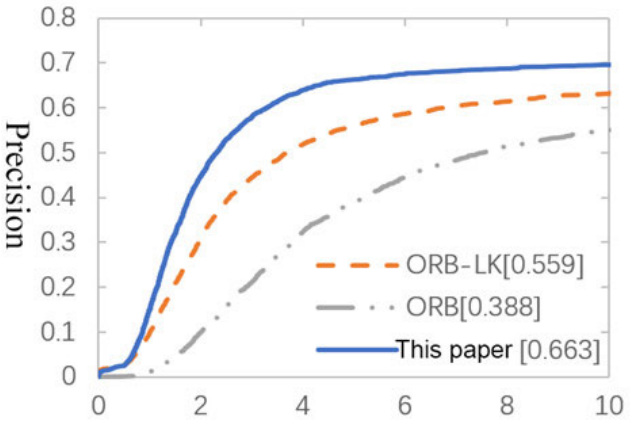
The ORB-LK methods experience failures in scenarios involving slight blur (the Pizza test set in Fig. 13) and random errors, but the proposed method yields more stable performance and thus lower errors in these scenarios. The image alignment errors and homography matrix errors with respect to the three methods in rotation changes are shown in Fig. 14.

3) PERSPECTIVE DISTORTION

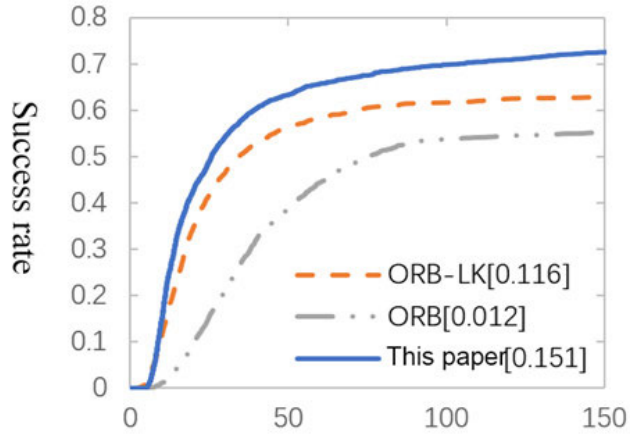
Tracking results pertaining to perspective distortion are shown in Figure 15. The ORB method witnesses tracking failure in scenarios involving relatively large perspective distortions, while the other two methods show a certain probability to succeed. It can be attributed to the fact that tracking methods taking advantage of consecutive frames are more competitive than those directly tracking feature points.

As perspective changes of the camera in the woman test set could cause lens reflection, the LK method will become extremely instable and lose its efficacy in lightning changes, leading to repeated reinitialization.

For the LK tracking part, the proposed method utilizes a DF non-linear descriptor to improve the successful tracking rate of the LK method with changing light conditions. In the meantime, the LK and ORB parts are independent from each other, which means when LK fails, the ORB method can still work for tracking. Therefore, the proposed method will not encounter problems arising from tracking failure associated with the ORB-LK method. As a result, the overall stability of the proposed method can be enhanced.



(a) Comparison of image alignment errors in rotation hanges

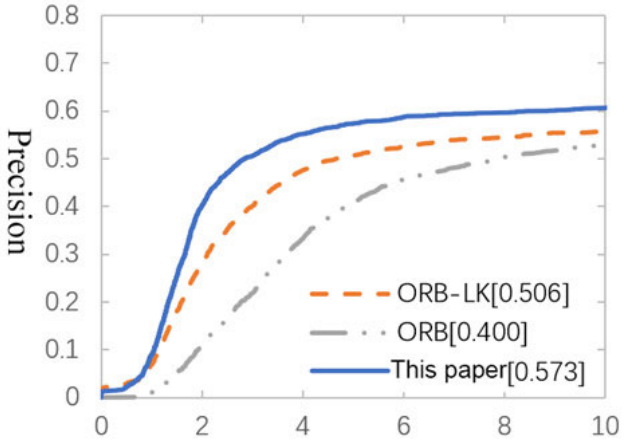


(b) Comparison of homography matrix errors in rotation changes

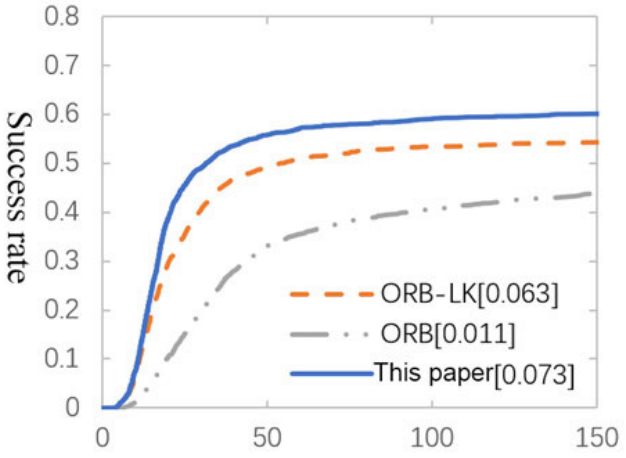
FIGURE 14. Error comparison of rotation change.

FIGURE 15. Perspective distortion.

The image alignment errors and homography matrix errors with respect to the three methods in perspective distortions are compared in Figure 16. The accuracy and success rate of the proposed method are 0.573 and 0.073, receptively, slightly higher than those of the ORB-LK method.



(a) Comparison of image alignment errors in perspective distortion



(b) Comparison of homography matrix error in perspective distortion

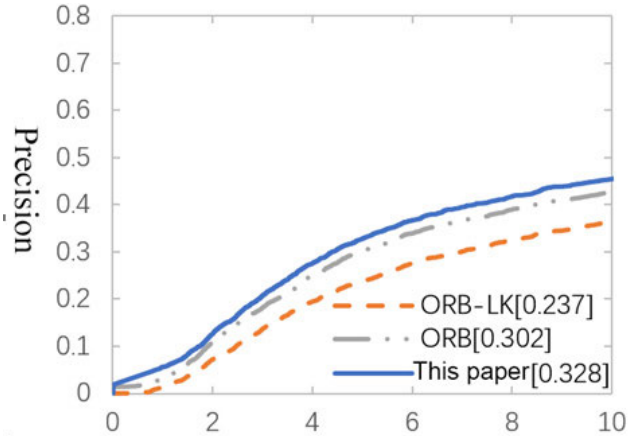
FIGURE 16. Error comparison of perspective distortion.

4) MOTION BLUR

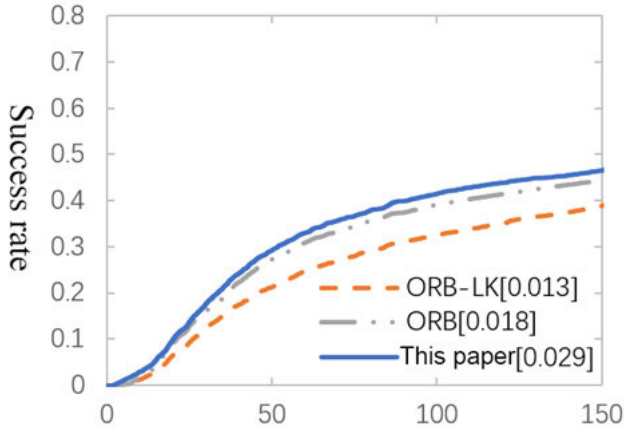
Tracking results pertaining to motion blur are shown in Figure 17. The success rates of all three methods in motion blur test are relatively low, as shown in Map-2 and Map-3 test sets in Figure 17. None of the three methods could perform tracking in conditions with relatively large motion blur. A few successful tracking cases generally occur in test sets with relatively small or no motion blur, for example, the Pizza and Snack test sets shown in Figure 17.

The LK method would fail in scenarios involving both large displacement and large motion blur. Therefore, the tracking part of the proposed method and the ORB-LK method both perform negatively in this challenging factor. The ORB method and the proposed method would invoke extra consumption in the initialization process following the failed tracking, thus they are more competitive than the ORB-LK method. The image alignment errors and homography matrix errors with respect to the three methods in motion blur are shown in Figure 18. Accuracies and success rates of all three methods are relatively low and none of these methods succeed in tracking in scenarios of relatively large

FIGURE 17. Motion blur.



(a) Comparison of image alignment errors in motion blur



(b) Comparison of homography matrix errors in motion blur

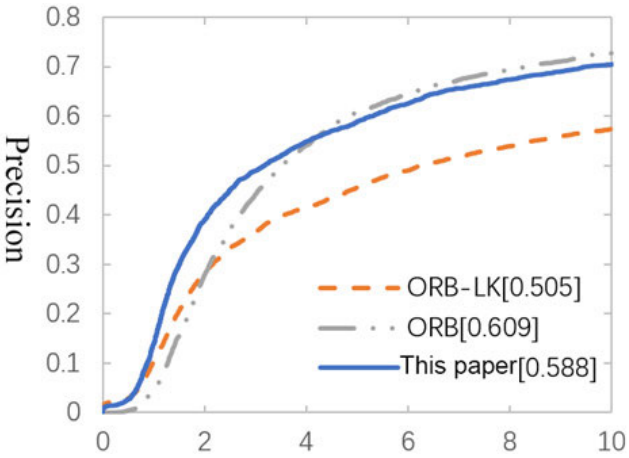
FIGURE 18. Error comparison of motion blur.

displacements. Hence, it is suggested that none of the three methods is resistant to large motion blur.

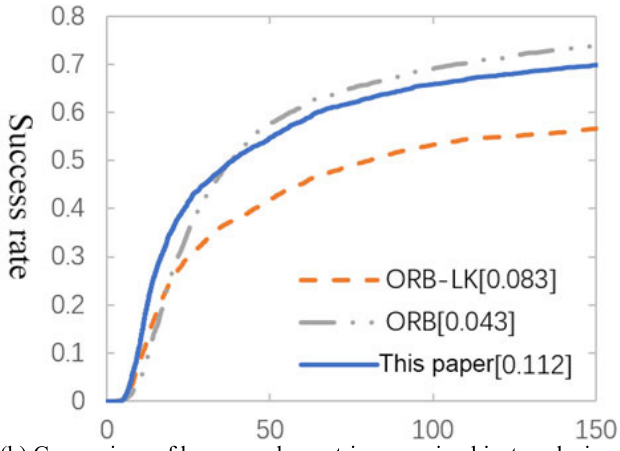
5) OBJECT OCCLUSION

Tracking results pertaining to object occlusion are shown in Figure 19. Due to its property of sparse feature points, the ORB method can perform successful tracking with

FIGURE 19. Object occlusion.



(a) Comparison of image alignment errors in object occlusion



(b) Comparison of homography matrix errors in object occlusion

FIGURE 20. Error comparison of object occlusion.

the presence of only part of the feature points. Therefore, the ORB method is highly competitive in terms of object occlusion.

In the ORB-LK method, each feature point is separately tracked using the LK method, and tracking cannot be restored when a tracked feature point is gradually occluded and then reappears. Therefore, tracking failure may occur when tracked feature points are being continuously occluded until the number is reduced to a certain limit. In the reinitialization process, the relatively small number of available feature points further reduces the stability of LK tracking. Although ORB is employed as its initialization approach, the ORB-LK method loses the ORB method's robustness to occlusion.

The LK tracking part of the proposed method utilizes a different implementation approach from the ORB-LK method: all feature points are treated as a tracking point set, LK tracking is applied on the entire feature point set, and occluded feature points are detected with forward and backward difference methods. Employing the ORB method on regions that have undergone adaptive non-maximal suppression, the proposed method would extract less valid feature points than the original ORB method in scenarios of object occlusion or small object scale. Such weakness is compensated by the improved LK method, making the overall performance of the proposed method as competitive as the ORB method in object occlusion.

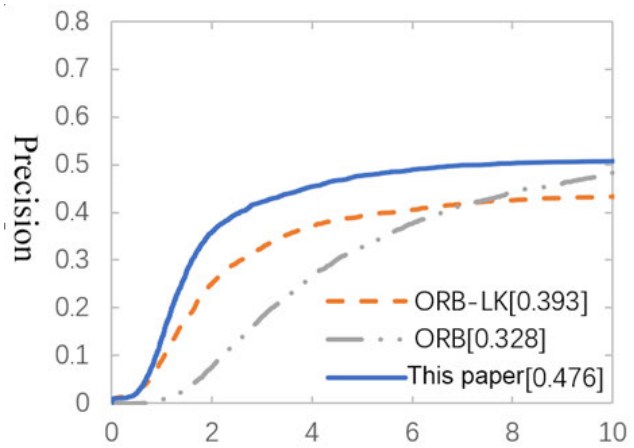
The image alignment errors and homography matrix errors with respect to the three methods in object occlusion are shown in Figure 20. The accuracies of ORB method and the proposed method are 0.609 and 0.588, and their success rates are 0.043 and 0.112, respectively, indicating roughly the same performance.

6) OBJECT OUT-OF-VIEW

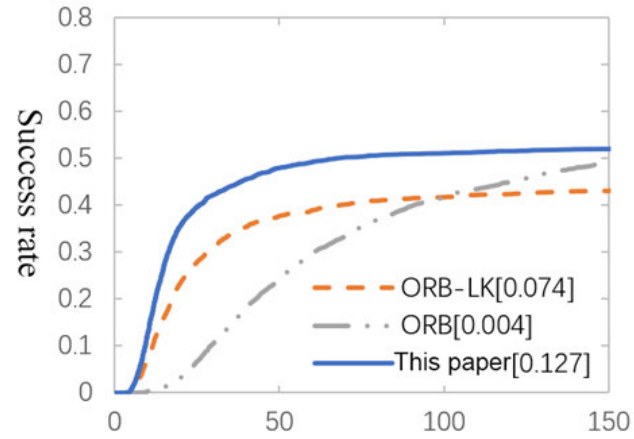
Tracking results pertaining to object out-of-view are shown in Figure 21. Object out-of-view bears certain similarities with object occlusion. The difference lies in that the tracked object can produce some camera-induced distortions on image edges with only a small part of the object completely out of the camera's view instead of large occluded regions.

FIGURE 21. Object out-of-view.

Due to the restrictions of matching accuracy and random errors arising from the detection of each feature point, the ORB method, in most cases, is less competitive than the ORB-LK method and the proposed method in terms of tracking accuracy. Further, due to the particular nature of these test sets, the ORB method presents no opportunity to



(a) Comparison of image alignment errors in object out-of-view



(b) Comparison of homography matrix errors in object out-of-view

FIGURE 22. Error comparison of out-of-view.

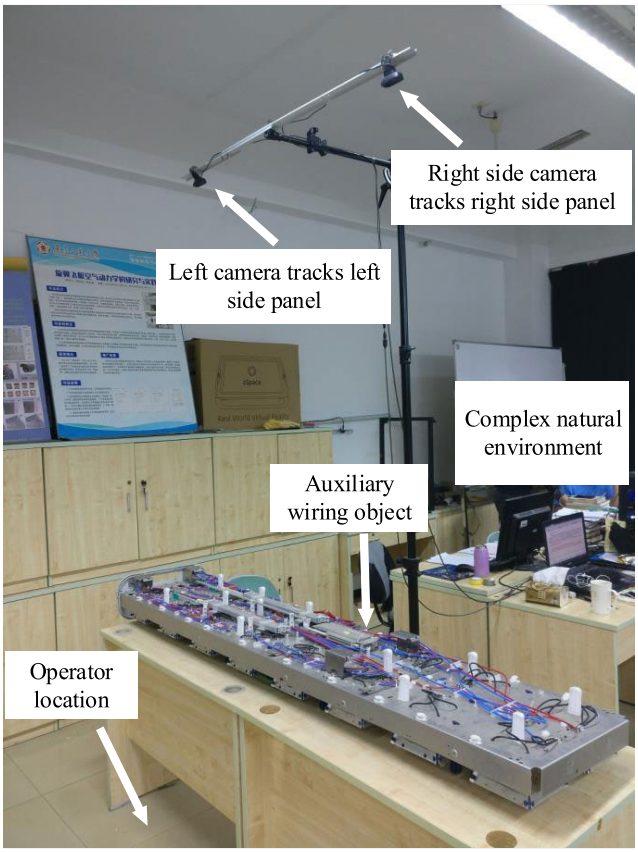


FIGURE 23. Environment of the assembly process.

demonstrate its strength in occlusion resistance. Therefore, the accuracy of the ORB method is lower than that of its counterparts.

Compared with the ORB-LK method, the proposed method has a higher accuracy. It is mainly because the proposed method has a higher tracking stability in different environments provided by the test sets, as well as in senarios involving object out-of-view or object being partially occluded.

The image alignment errors and homography matrix errors with respect to the three methods in object out-of-view are shown in Figure 22. Although the ORB method fell behind the ORB-LK method in terms of accuracy and success rate, it nonetheless outperforms the ORB-LK method when the range of accuracy and success rate are expanded.

VI. CONCLUSION

In this paper, implications of the AR technology for the production process and human operator training are briefly discussed. The current status of researches related to 3D registration is analyzed. In addition, the paper discusses the problems with existing natural feature-based methods in natural environments, such as relatively large perspective distortion, occlusion, complicated background environment and object out-of-view, which include low accuracy in tracking and registration and poor stability and adaptability. Generally, the proposed method has combined the merits of the ORB and LK methods and has reduced the influences of their weaknesses. The image alignment error and homography matrix error of the proposed method are both lower than the ORB and ORB-LK methods in tracking images involving scale change, rotation change, object occlusion and object out-of-view. In multiple tracking environments involving light changes, occlusion and relatively large inclination angles, the proposed method shows better stability.

One more thing, the methods used in this paper are not only the experimental stage, but also applied to AR auxiliary wiring project. Due to confidentiality reasons, this article only releases the pictures of experimental environment (Fig.23) and experimental effect (Fig. 24-27) which are showed as follow.

AR auxiliary wiring is an important research direction of intelligent manufacturing. It has changed the wiring process of communication equipment from the guiding mode of assembly instruction in the past to the guiding mode of AR visualization. Workers can see the virtual assembly process information on physical communication devices by wearing HoloLens glasses. The virtual information of the wire to be

FIGURE 24. Navigation interface of AR auxiliary wiring system.

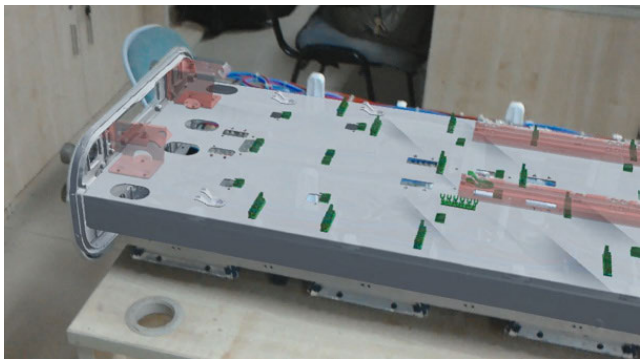


FIGURE 25. Virtual and real fusion effects at different angles.

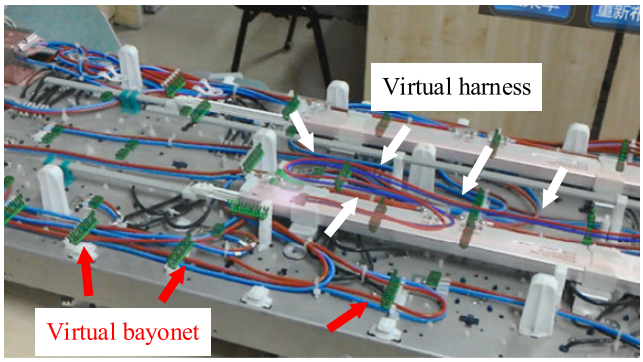


FIGURE 26. Tracking result during the assembly process.

connected and the bayonet of each wire in each assembly step is displayed in the corresponding wiring position. When the worker assembles the line of the current step, the system can also detect the wiring condition. If the wrong wiring situation occurs, the related virtual information prompt will also appear in the corresponding position. As shown in Fig.23, since the harness device is very long, the operator can only see a part of the harness device during the assembly process, and the operator moves around the harness device. Therefore, complex viewing angle changes, occlusion, rotation changes, scale changes, and target departure vision will occur during the assembly process. These conditions are well solved by the methods used in this paper, as shown in Fig.24-27.

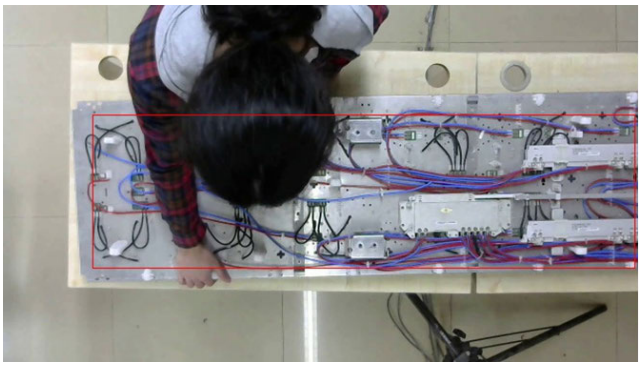


FIGURE 27. Tracking result of wiring harness equipment.

The work of this paper is mainly based on the research of the predecessors in the field of computer vision. We have improved the previous research and integrated innovation. More importantly, we have used these studies in the field of augmented reality for the first time and applied to intelligence manufacturing. However, this research is still subject to drawbacks. Although the proposed hybrid tracking method that combines ORB with LK has improved to some extent the accuracy and reliability of tracking, it has a relatively poor performance in environments involving large computational burdens and motion blur, and the objects that can be tracked are limited to those on a 2D plane. Furthermore, the natural feature-based 3D registration method can only be implemented with unknown object features and feature training. As augmented reality can merely be implemented on particular objects, the range of its application is restricted as well.

REFERENCES

[1] M. Yuan, H. Yu, J. Huang, and A. Ji, "Reconfigurable assembly line balancing for cloud manufacturing," *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2391–2405, 2019, doi: 10.1007/s10845-018-1398-7.

[2] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 12, pp. 2633–2651, Dec. 2016, doi: 10.1109/TVCG.2015.2513408.

[3] P. Krompiec and K. Park, "Enhanced player interaction using motion controllers for first-person shooting games in virtual reality," *IEEE ACCESS*, vol. 7, pp. 124548–124557, 2019, doi: 10.1109/ACCESS.2019.2937937.

[4] Z. K. JS Ong and A. Y. C. Nee, "Development of an AR system achieving in situ machining simulation on a 3-axis CNC machine," *Comput. Animat. Virt. Words*, vol. 21, no. 2, pp. 103–115, Mar./Apr. 2010.

[5] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. 2nd Int. Workshop Augmented Reality (IWAR)*, Oct. 1999, pp. 85–94.

[6] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 590–596.

[7] D. Wagner, G. Reitmayr, M. Alessandro, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proc. 7th IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2008, pp. 125–134.

[8] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Markerless augmented reality with a real-time affine region tracker," in *Proc. Int. Symp. Augmented Reality*, Oct. 2001, pp. 87–96.

[9] G. BleserY Pastarmov and D. Stricker, "Real-time 3D camera tracking for industrial augmented reality applications," in *Proc. 13th Int. Conf. Central Eur. Comput. Graph.*, 2005, pp. 47–54.

[10] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 4, pp. 615–628, Jul./Aug. 2006, doi: 10.1109/TVCG.2006.78.

[11] D. Wagner and D. Schmalstieg, "ARToolkitplus for pose tracking on mobile devices," in *Proc. Comput. Vis. Winter Workshop*, Feb. 2007, pp. 139–146.

[12] D. Wagner, G. Reitmayr, M. Alessandro, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 3, pp. 355–368, May/Jun. 2010.

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[14] R. Jason, P. Alain, M. Schneider, A. Oleksandr, and S. Didier, "6DoF Object Tracking based on 3D Scans for Augmented Reality Remote Live Support," *Computer*, vol. 7, no. 1, pp. 6–13, 2018, doi: 10.3390/computers7010006.

[15] A. Ufkes and M. Fiala, "A markerless augmented reality system for mobile devices," in *Proc. Int. Conf. Comput. Robot Vis.*, May 2013, pp. 226–233.

[16] K. Georg, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th Int. Symp. Mixed Augmented Reality*, 2007, pp. 1–10.

[17] P. Martin, E. Marchand, P. Houlier, and I. Marchal "Mapping and relocalization for mobile augmented reality," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3352–3356.

[18] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good features track better," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1998, pp. 178–183.

[19] M. Calonder, V. Lepetit, S. Christoph, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2010, pp. 778–792.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1–8.

[21] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, 2016.

[22] G. L. David, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/b:visi.0000029664.99615.94.

[23] P. Liang, Y. Wu, H. Lu, L. Wang, C. Liao, and H. Ling, "KAZE Features," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 214–227.

[24] T. M. Chan. (2019). *A Minimalist's Implementation of an Approximate Nearest Neighbor Algorithm in Fixed Dimensions [EB/OL]. Urbana: 2006*. [Online]. Available: http://tmc.web.engr.illinois.edu/pub_ann.html

[25] P. Liang, Y. Wu, H. Lu, L. Wang, C. Liao, and H. Ling, "Planar object tracking in the wild: A benchmark," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 651–658.

[26] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004, doi: 10.1023/b:visi.0000011205.11775.fd.

[27] A. Crivellaro and V. Lepetit, "Robust 3D tracking with descriptor fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3414–3421.

[28] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 1999, p. 1090.

[29] L. Florack, B. Ter Haar Romeny, M. Viergever, and J. Koenderink, "The Gaussian scale-space paradigm and the multiscale local jet," *Int. J. Comput. Vis.*, vol. 18, no. 1, pp. 61–75, 1996, doi: 10.1007/bf00126140.

[30] N. Georg and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 862–869.

[31] D. Müllner, "Fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python," *J. Stat. Softw.*, vol. 53, no. 9, pp. 1–18, 2013.

XIAN YANG (M'19) was born in Guangdong, China, in 1982. He received the Ph.D. degree in engineering from the School of Electromechanical Engineering, Guangdong University of Technology, in 2018. He is currently studying in the fields of human–computer interaction and cognitive psychology.

He has been engaged in human–computer interaction-related research at universities in Guangzhou, China, and also directed several graduate students to conduct research in this area. They also provide technical solutions for enterprises. He and his team have published several articles in the IEEE-related journals and directed graduate students to continue to publish articles in the IEEE Journals.

JINGFAN YANG was born in September 1994. He is currently pursuing the graduate degree with the School of Electromechanical Engineering, Guangdong University of Technology. He is mainly engaged in computer vision research. He has a solid foundation in the development of augmented reality applications. He has participated in the development of many projects.

HANWU HE was born in Hubei, in 1964. He is currently a Professor and a Ph.D. Supervisor with the School of Electromechanical Engineering, Guangdong University of Technology. He has been engaged in the basic theory and key technology research of intelligent manufacturing for many years and has undertaken a lot of projects in virtual reality and intelligent manufacturing. His research group has studied computer vision, stereoscopic display methods, and the theory and methods of constructing virtual reality environments.

HEEN CHEN was born in 1975. He received the Ph.D. degree from the School of Electromechanical Engineering, Guangdong University of Technology, China. He has been involved in a number of AR and VR research projects funded by the Chinese National Natural Science Fund. His current research interests focus on the development and research of SLAM, augmented reality, virtual reality, and smart manufacturing.