A Visual Fingerprint Update Algorithm Based On Crowdsourced Localization And Deep Learning For Smart IoV

Xiliang Yin
Harbin Institute of Technology
Lin Ma (malin@hit.edu.cn)
Harbin Institute of Industry: Harbin Institute of Technology https://orcid.org/0000-0003-1684-9714
Ping Sun
Harbin Institute of Technology

Research

Keywords: Smart IoV, Visual Map, deep learning, visual localization

Posted Date: August 17th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-786878/v1

A Visual Fingerprint Update Algorithm Based On Crowdsourced Localization and Deep Learning for Smart IoV

Xiliang Yin*†, Lin Ma*, and Ping Sun*
* School of electronics and information engineering, Harbin Institute of Technology, Harbin, P. R. China
†Harbin Vocational & Technical College, Harbin, P. R. China
Email: malin@hit.edu.cn

***Abstract*—Recently, the deep learning and vision-based technologies has shown their great significance for the prospective development of smart Internet of Vehicle (IoV). When the smart vehicle enters the indoor parking of a shopping mall, the vision-based localization technology can provide reliable parking service. As known, the vision-based technique relies on a visual map without a change in the position of the reference object. Although, some researchers have proposed a few automatic visual fingerprinting (AVF) methods, which are aiming at reducing the cost of building the visual map database. However, the AVF method still costs too much under such situation, since it is impossible to determine the specific location of the displaced object. In view of the smart IoV and the development of deep learning approach, we propose a crowdsourcing and deep learning based algorithm for solving the problem in this paper. Firstly, we propose a Region-based Fully Convolutional Network (R-FCN) based method with the feedback of crowdsourced images to locate the specific displaced object in the visual map database. Secondly, we propose a method based on quadratic programming (QP) for solving the translation vector of the displaced objects, which finally solves the problem of updating the visual map database. The simulation results show that our method can provide a higher detection sensitivity and correction accuracy as well as the relocation results. It means that our proposed algorithm outperforms the compared one, which is verified by both synthetic and real data simulation.**

***Index Terms*—Smart IoV, Visual Map, deep learning, visual localization**

I. Introduction

A. Background and Significance

The driving assistance system integrating vision and artificial intelligence technology will gradually become a research hotspot in the field of smart Internet of Vehicle (IoV) [43], [44]. This system is more important for smart vehicles running in indoor environment, such as the indoor parking of a shopping mall. The reason is that wireless communication in the indoor environment is more vulnerable to interference, resulting in more packet loss or greater delay. Therefore, vision-based technology is a promising solution for the positioning and navigation of the smart vehicles after entering the indoor environment.

Vision-based localization technology has its unique advantages. The evidence is more obvious in the composite positioning technologies. For instance, WiFi and vision [1], inertia and vision [2], Lidar and vision [3], hybrid localization technology [4]. Visual localization is a key role in their system architecture, respectively. The premise of accurate visual localization is to establish a visual map database on the offline stage in the interested areas. Typically, with the help of customized equipment [5]–[7], the visual fingerprint can be collected very accurately in a given area. However, these devices need to be customized specially, which are also expensive. To solve this problem, Farhang *et al.* proposed an automatic visual fingerprinting method based on the consumer-grade device in [8], which was further improved in our previous work [9]. These methods have well solved the visual fingerprinting problem in different ways. However, some objects in an interested area will be moved randomly, which will lead to localization deviation. Unfortunately, none of the existing visual fingerprinting methods could solve the visual fingerprint updating problem directly, which is caused by displaced objects in the location area.

In contrast, the problem of updating the WiFi radio map is well studied in its corresponding research community. The method of updating a WiFi radio map can be summarized into three categories. One is predicting the Recieved Signal Strength Indication (RSSI) fingerprint by the particular radio propagation model, [10]–[12] can be classified as examples of this type. The other one is leveraging the deep learning framework for generating the renewed radio fingerprint by training the large amount of time-varying RSSI, Signal Noise Ratio (SNR), or Channel State Information (CSI), such as [13]–[15]. These two kinds of methods based on the fingerprint change with some fixed patterns. Unlike the characters of radio fingerprint, the visual fingerprint has no model for learning algorithms. Another method is crowdsourcing. When the user is cooperative, the fingerprint used in his/her localization can be updated from the available radio map. Several methods are proposed in this category for finding the best updated RSSI fingerprint [16]–[20]. Although these methods could not be directly used for updating the visual fingerprint, they provide an inspired idea.

An interesting technology that should mention in this section is the Region-based Fully Convolutional Network (R-FCN), which was proposed by J. F. Dai *et al.* in [21]. R-FCN provides an effective solution for image recognition, which is well proved by [22]–[24]. It should be noted that we also use this framework for semantic segmentation in crowdsourced

query images. By leveraging these regions generated by R-FCN, semantic Speed-Up Robust Feature (SURF) forms. Fortunately, there are many existing frameworks for deep learning implementation, among which *Caffe* is a representative one. A list of works has shown its superior performance [25]–[27]. More details about *Caffe* are referred to [28].

Therefore, the purpose of this paper is to propose a visual fingerprint update algorithm based on crowdsourced visual localization. More specifically, a displaced visual fingerprint detection method and a quadratic programming (QP) based visual fingerprint update method are designed in two consecutive steps. The main contributions of this paper are in three folds:

1). A visual fingerprint update algorithm is proposed in this paper, which is based on crowdsourced visual localization. When the users are cooperative, compared to the location in the visual map database, the algorithm could detect automatically whether the objects in the localization area are displaced or not.

2). A novel method is proposed in this paper for detecting the positional change of the visual fingerprints. A region-based fully convolutional network is used for labeling the SURF descriptor in the query image, which is defined as Semantic SURF. Compared with the traditional Perspective-n-Point (PnP) solver with all 2D-3D correspondences, our method is calculated with semantic 2D-3D correspondences. Our proposed method has a higher detection ratio, which is proved by synthetic and real data simulation results.

3). A Quadratic Programming (QP) based visual fingerprint update method is also proposed. In this way, the fingerprint can update automatically without the real crowdsourced localization results. The accuracy is higher than that of the compared method under different configurations.

The rest of this paper organizes as follows. In Section II, some related works will be discussed. Section III describes the visual fingerprint update problem in visual localization and presents our proposed system model. In Section IV, we propose our semantic SURF based detection method and QP based update method, respectively. Section V provides the simulation results, and the conclusion draws in Section VI.

B. Ralated Works

As far as we know, this paper is the first proposal for solving the visual fingerprint update problem. Therefore, in this section, we introduce the relevant works in our proposed algorithm. Farhang *et al.* proposed an automatic visual fingerprinting (AVF) method for the first time in [8], which improves the efficiency of collecting the fingerprint and reduces the cost of acquisition equipment. Once the object is displaced, which is also shot in the query image by the crowdsourced user, the result of visual localization will deviate. At present, it seems that the only solution is the periodic scanning by AVF method to correct the error in the database due to object displacement. However, this will lead to two problems, one is how to set the optimal acquisition cycle, the other is the consumption of labor and time cost caused by overall rescanning.

Visual localization is more accurate than the other methods, which is mainly due to its 6 degree of freedom mapping equation. The coefficients of these equations are defined by 2D-3D correspondences. [29] proposed an effective and efficient method for finding 2D-3D correspondences. The method expresses the 3D point by 2D image feature descriptor when it is generated by the Structure from Motion (SfM) technique. The mean value of the two matched 2D feature descriptors is saved for representing the particular 3D point. The 2D feature descriptor could be Scale Invariant Feature Transform (SIFT) [30], SURF [31], or any other features. When a 2D image feature and 3D point match, a Fast Library for Approximate Nearest Neighbor (FLANN) [32] search or bag of visual words approach could accomplish this task. In our proposed algorithm, we also use this method to find reliable 2D-3D correspondences. More specifically, SURF and FLANN are chosen in the framework.

Efficient PnP (EPnP) is the most widely used solver for PnP problem [33] , which is classical for its O(n) complexity with known camera internal parameters. The algorithm utilizes the linearization and re-linearization method for solving the weight of a linear combination of a matrix eigenvector, which derives from 3D-2D correspondences. With these weights, the camera coordinates of the 3D point can calculate. Then, the rotation matrix and translation vector decompose with the help of Singular Value Decomposition (SVD) for solving the matrix maximum trace problem. Furthermore, a more accurate result will be reached by setting the closed-form solution as the initial input of the Gauss-Newton scheme. Displaced objects in the indoor interested environment may lead to localization error due to the mismatch between the 2D feature image coordinates and 3D point world coordinates. A natural idea is setting a threshold, which acts as a criterion of the 3-dimensional localization results. When the feedback of crowdsourced users are beyond the threshold, it can assume that some reference objects in the scene may move. This method could treat as the traditional strategy and the benchmark of our proposal. Specifically, EPnP with all 2D-3D correspondences with a judging threshold is chosen as our benchmark. Since in the large-scale environment the self-verifiable localization mark is the vertical result, there is only one predefined threshold for judging the reliability of the localization result.

RANdom SAmple Consensus (RANSAC) is well known for filtering outliers in the dataset by calculating the mathematical model parameters of the samples [34]. It is generally applicable to refine the matched pairs during the offline stage due to its time-consuming characteristics in the computer vision field. For the algorithm proposed in this paper, the calculation cost of RANSAC is not so restrictive. Therefore, it can be used for filtering the mismatch, so that a corrected localization result will be generated, which could also locate the displaced semantic object.

In recent years, the Deep Learning (DL) based method has been embedded into the indoor localization framework. One role of DL in localization is to generate new fingerprints compared to traditional ones. [35] proposed a deep learning feature for localization, which was trained by multisensor fingerprints. Ma *et al.* constructed hierarchical convolutional features for visual tracking in [36]. However, the improvement

is limited in terms of localization accuracy, considering the cost of the training. The other application is to map the image information and its location directly through the DL network. This method entails an obvious cost, which needs a substantial set of training data. Although it shows some practical value in the latest research, such as [37], [38], it still has no advantage from the point of theoretical view. Another one is leveraging DL for semantic segmentation, which could provide a pixel-wise classification. Undoubtedly, it is embedded into the DL based visual localization framework to enhance the performance. A representative one based on DL is proposed in [39], which is called VLocNET++. Unfortunately, it can not be used to solve the problem mentioned in this paper.

DL could use as a fundamental tool for filtering features, which could provide more accurate coefficients for solving the PnP equations. We define this as Semantic SURF, and it is a basic tool for challenging the visual fingerprint update problem. Specifically, in this paper, we propose a visual fingerprint update algorithm under the crowdsourced framework, which contains a semantic SURF based visual fingerprint, displacement detection method with the help of R-FCN and a visual fingerprint update method. In this way, the fingerprints can update automatically and reliably.

C. System Model

As shown in **Fig. 1**, once the layout of the located region is known, the visual fingerprint will be collected from the visual sensors mounted on a smart vehicle, which is labeled in green. When it is fingerprinting over the extinguisher, its

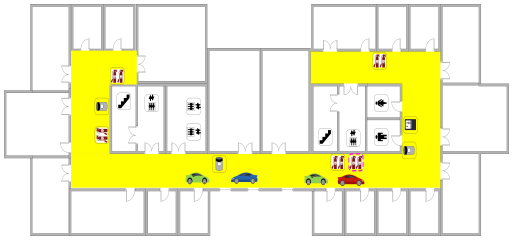

Fig. 1 Visual fingerprint displacement detection initiated by crowdsourced localization.

position will record as the position framed by the grey box in the database. Thereafter, the extinguisher moves to the position of the red dotted box. The visual localization result will deviate when the vehicle labeled with red tends to locate itself by the visual sensor and the surrounding visual fingerprint, due to the displaced extinguisher in the query image. Meanwhile, the vehicle labeled in blue will obtain an accuracy result from its visual sensor since its reference fingerprint is fixed, which may be the unmoved ashbin. The vehicles labeled both red and blue can be regarded as volunteers, which are located by their visual sensor at different times after generating the visual fingerprint database. Whenever the vehicle has completed the visual localization or navigation in the scene, as long as it is pleased to send the data back to the server cooperatively, our proposed algorithm will work automatically without any human aids. Noted that in our proposed algorithm, there is no need to give the ground truth location of the query image or any other additional information.

When the query image from the volunteer sends back to the server, the visual fingerprint update algorithm is initiated. A block diagram of our proposed algorithm shown in **Fig. 2**. The

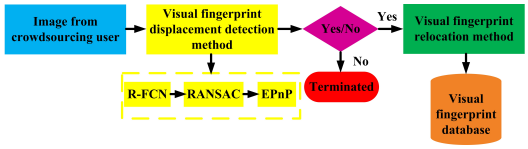

Fig. 2 The overflow of the crowdsourcing visual fingerprint updating algorithm.

location of the query image will recalculate on the server to validate the availability of the feedback from the crowdsourced user. As shown in **Fig. 2**, the visual fingerprint displacement detection method will be applied in the algorithm. When the results obtained by this method are beyond the predetermined threshold, the feedback from crowdsourced users is invalid, so that the overall process will terminate. On the contrary, when the method locks the displaced fingerprint, the subsequent fingerprint relocation method will initiate. Thereafter, the new location of the displaced visual fingerprint will be calculated, and the database refreshes. Our proposed algorithm will automatically detect and update the displaced visual fingerprint to substitute for the periodic manual fingerprint scanning.

Although the overall algorithm typically operates on the server, the calculation should complete as quickly as possible in consideration of the amount of crowdsourcing data. As shown in **Fig. 2**, the visual fingerprint displacement detection method mainly consists of R-FCN, RANSAC, and EPnP. The online computation complexity of R-FCN is O(n), as well as EPnP, while RANSAC is mainly affected by the number of iterations and the value of maximum tolerable error.

A brief illustration of the role of the region-based fully convolutional network (R-FCN) in this paper is provided in **Fig. 3**. The crowdsourced image is an input of R-FCN, whose output is the corresponding semantic segmentation and the label with its score. In the next step, semantic SURF will extract, respectively, which shows in **Fig. 4**. It could be

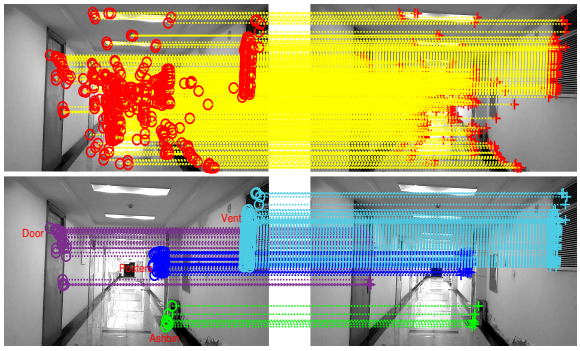

Fig. 3 Comparison of matching results by common SURF vs semantic SURF.

further filtered by RANSAC with its corresponding one in the reference image. According to its matching results with the point cloud simultaneously, several bundles of semantic PnP equations will be established. Then, different localization results calculate. Otherwise, the semantic point cloud can also be generated by leveraging the semantic SURF. Besides

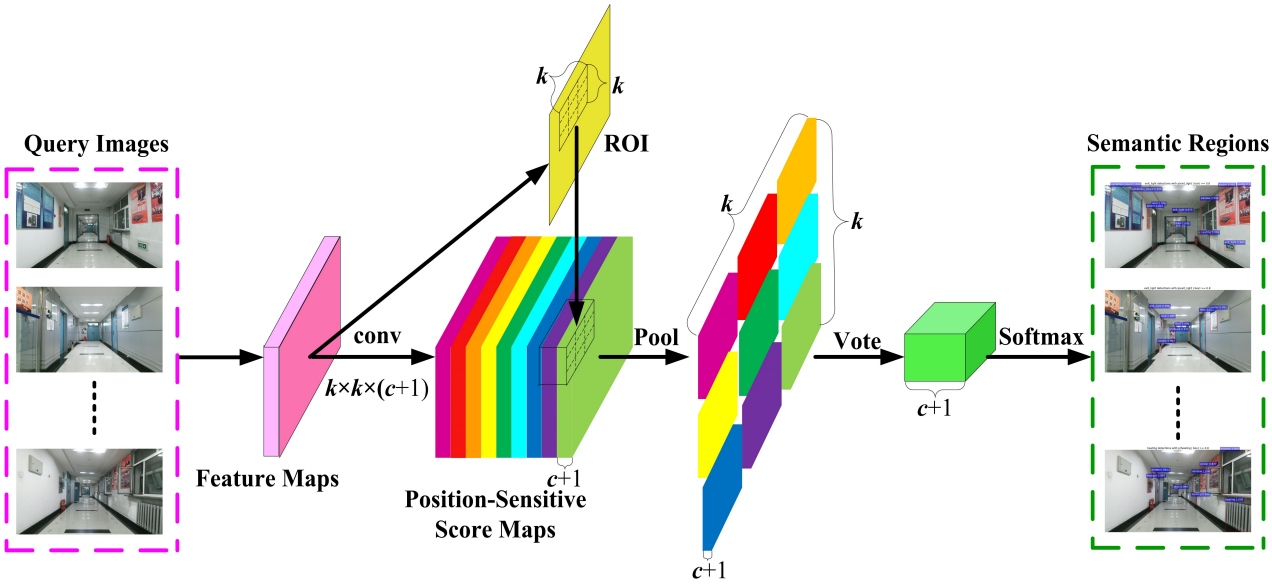

Fig. 4 The architecture of Region-based Fully Convolutional Network(R-FCN) and some crowdsourced images with their labeling results by R-FCN.

matching, the images above could also describe the traditional way of generating point clouds by two frames in a sequential image sequence, while the images below could represent a novel way for creating a semantic point clouds by R-FCN.

When all groups of semantic SURF are beyond the predefined threshold, the crowdsourced data will be determined as invalid by the proposed method, the overall process will terminate as a consequence. When a group of semantic SURF is within the predefined threshold, the displaced object will be locked by the method automatically. The result will be input into the visual fingerprint relocation method, which is aimed at providing a reliable new location of the displaced visual fingerprint locked by the previous method. Then the translation can be solved by the equations, whose coefficients are from the semantic 2D-3D correspondences. The optimal value of the translation is the solution of a constructed quadratic programming problem. Finally, the refreshed location will be recorded into the visual fingerprint database. We will explain the visual fingerprint displacement detection method and show how it defines in Section IV-A. Then, the visual fingerprint relocation method details in Section IV-B.

II. Methods

A. Visual Fingerprint Displacement Detection Method

As stated before, the crowdsourced image sends back to the server. For a Region of Interest (RoI) rectangle of size $w \times h$ in the image, $k \times k$ bins form with each size approximate to $\frac{w \times h}{k^2}$. In the last convolutional layer, the k^2 score map for each category produces, a pooling scheme defined as

$$r_c(i,j|\Theta) = \sum_{(x,y)\in bin(i,j)} z_{i,j,c}(x+x_0, y+y_0|\Theta)/n, \quad (1)$$

where $r_c(i,j)$ is the response in the (i,j)th bin for the cth category, $z_{i,j,c}$ is one score map out of the $k^2(c+1)$ score map, (x_0, y_0) is the top-left corner of an RoI, n is the number of pixels in the bin, Θ is all parameters of the R-FCN. Then, k^2 scores are voted by averaging on the RoI. A $c+1$ dimensional vector generates as $r_c(\Theta) = \sum_{i,j} r_c(i,j|\Theta)$. Finally, the softmax responses computed by $s_c(\Theta) = e^{r_c(\Theta)} / \sum_{ci,i=0}^{c} e^{r_{c_i}(\Theta)}$. Note that the output of R-FCN for each RoI is the particular category c_i and its bounding box $\mathbf{S}_i = (s_x, s_y, s_w, s_h)$. In general, the output from the pretrained R-FCN can express as

$$\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_j, \ldots, \mathbf{S}_m\}, \quad (2)$$

where $\mathbf{S}_j$ is the jth semantic region, m is the total semantic region labeled by R-FCN.

Then, the SURF descriptors extracts, which express as

$$\mathbf{F} = \{F_1, F_2, F_i, \ldots, F_n\}, \quad (3)$$

where n is the number of SURF descriptor, $\mathbf{F}_i$ is a vector. With the help of the descriptor, 2D features and 3D points can be mapped uniquely by [29]. Typically, the 2D image coordinates and the 3D point world coordinates will be the coefficients of our proposed method after semantic segmentation. The pixel coordinates of each SURF descriptor can be used to judge easily which $\mathbf{S}_i$ it belongs to. It describes as

$$F_{s_j} \in \mathbf{S}_j, \quad (4)$$

where $F_{s_j} \subset \mathbf{F}$. Then, RANSAC is used for refining the 2D-2D matches between the crowdsourced image and the reference image. The remaining SURF descriptor with its semantic label can be used for checking the displacement of the object. We define the matched SURF number of each semantic segmentation before RANSAC as $n^{\mathbf{s}_j}_{match}$, and the remaining SURF number after RANSAC as $n^{\mathbf{s}_j}_{filter}$. Typically, when the ratio $\phi_{\mathbf{s}_j} = n^{\mathbf{s}_j}_{filter} / n^{\mathbf{s}_j}_{match}$ is lower than a threshold ϕ_{thr}, it could assume that the corresponding semantic object is displaced comparing to the reference. Furthermore, considering the randomness of RANSAC, m groups of PnP equations will establish, respectively. By EPnP [33] algorithm, the localization results of the crowdsource user can be recalculated, where $\{\mathbf{R}_{filter}, \mathbf{t}_{filter}\}_{RANSAC}$ is the solution

of all the RANSAC filtered 2D-3D correspondences. When the number is too small to solve the accurate solution, the calculation of the reprojection error will substitute for solving the EPnP equations, which is a golden standard for judging the correctness of the solution. It will correct the misjudgment of the first step as far as possible.

Finally, the displaced object can be judged by

$$\begin{cases} \mathbf{S}^{disp}=\{\mathbf{S}_i|\phi_{\mathbf{S}_i}<\phi_{thr}\}\cup\{\mathbf{S}_i|\epsilon_{reproj}^{\mathbf{S}_i}<\epsilon_{reproj}^{thr}\}, n\leq 5 & \text{(5a)} \\ \mathbf{S}^{disp}=\{\mathbf{S}_i|\phi_{\mathbf{S}_i}<\phi_{thr}\}\cup\{\mathbf{S}_i|\epsilon_{diff}^{\mathbf{S}_i}<\epsilon_{diff}^{thr}\}, n>5, & \text{(5b)} \end{cases}$$

where $\mathbf{S}^{disp}$ is the displaced semantic region in the image plane, $\phi_{\mathbf{S}_i}$ is the ratio labeled by the semantic region $\mathbf{S}_i$, ϕ_{thr} is a threshold, $\epsilon_{reproj}^{\mathbf{S}_i}$ is the reprojection error of correspondences labeled by $\mathbf{S}_i$, ϵ_{reproj}^{thr} is the reprojection error threshold. $\epsilon_{diff}^{\mathbf{S}_i}$ is the difference between $\{\mathbf{R}_{filter}, \mathbf{t}_{filter}\}_{\mathbf{S}_i}$ and $\{\mathbf{R}_{filter}, \mathbf{t}_{filter}\}_{RANSAC}$, ϵ_{diff}^{thr} is the difference threshold. When the semantic object is unshifted, the corresponding region is defined as $\mathbf{S}^{unsh}$. Consequently, the displaced object will be ranked according to the semantic region in the image plane. The proposed method is summarized in **Algorithm 1**.

Algorithm 1 The proposed visual fingerprint displacement detection method

Input: $\mathbf{I}$, ϕ_{thr}, ϵ_{reproj}^{thr}, ϵ_{diff}^{thr}
Output: $\mathbf{S}^{disp}$,$\mathbf{S}^{unsh}$,$\mathbf{R}_{filter}$,$\mathbf{t}_{filter}$

```
Put image I into R-FCN in order to get the semantic segmentation S = {S_1, S_i, ..., S_m}
Extract SURF descriptor set F = {F_1, F_i, ..., F_n} in image I
Label each SURF with its corresponding semantics
Run RANSAC for filtering the matched SURF between the query image and the reference image
Map the 2D feature with their corresponding 3D point by method [29], generate the tuple set T_i = {{u,v,x,y,z}_1, {u,v,x,y,z}_j, ..., {u,v,x,y,z}_k}
for i = 1; i < m; i++ do
    if φ^{S_i} < φ_thr then
        Label S_i as S^{disp}
    else
        Label S_i as S^{unsh}
    Calculate the {R, t} of the crowdsourcing image by EPnP [33] with T_i
    if ε_diff^{S_i} < ε_diff^{thr} then
        Revise the judgment of displaced label by previous step
return S^{disp}, S^{unsh}, R_filter, t_filter
```

B. Visual Fingerprint Relocation Method

In the previous subsection, the displaced object has been locked as well as the unshifted objects. The translation vector of the displaced object will calculate for refreshing the visual fingerprint database in this subsection. First of all, a benchmark method will be illustrated. Although it can be derived very simply from the *PnP* equation, we need to give a brief deduction in this subsection to compare with our proposed method. Another reason is that it is the first proposal for solving the visual fingerprint refreshing problem. The semantic 2D-3D correspondences can be clustered into two categories, which describes as two sets $\mathbf{T}^{disp} = \{u_d, v_d, x_d, y_d, z_d\}$ and $\mathbf{T}^{fix} = \{u_f, v_f, x_f, y_f, z_f\}$. $\mathbf{T}^{disp}$ represents the set of displaced 2D-3D correspondences, while $\mathbf{T}^{fix}$ is the set of fixed ones. The rotation matrix and translation vector of the crowdsourcing query image can be calculated by EPnP [33] with the coefficients from $\mathbf{T}^{fix}$, which donates as $\mathbf{R}$ and $\mathbf{t} = [t_1, t_2, t_3]^{\mathbf{T}}$. The relative translation vector of the displaced object defines as c_x, c_y, c_z, respectively. Then, according to the PnP equation, without considering the rotation of the displaced object, we have

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K}(\mathbf{R} \begin{bmatrix} x_i + d_x \\ y_i + d_y \\ z_i + d_z \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}), \tag{6}$$

where u_i, v_i is the ith 2D feature coordinate, λ_i is its depth factor, x_i, y_i, z_i is its corresponding 3D point coordinate, and $\mathbf{K}$ is the camera internal matrix of the crowdsourced user. The internal matrix can be roughly recovered from the crowdsourced image, or sent by the user as a part of the feedback data accurately, $\mathbf{K}$ can further represent as

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

where f is the focal length, u_0, v_0 is the principal point coordinate of the image. Noted that the coefficients of (6) are from the set $\mathbf{T}^{disp}$. For a tuple of $\mathbf{T}^{disp}$, (6) could transform to a simplified form of three linear associated equations with four unknowns. A very natural idea is to do an elimination before solving the equations. When λ_i is eliminated, we have

$$\underbrace{\mathbf{A}^i}_{\mathbf{2\times 3}} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \underbrace{\mathbf{B}^i}_{\mathbf{2\times 1}}, \tag{8}$$

where $\mathbf{A}_{11}^i = fr_{11} + (u_0 - u_i)r_{31}$, $\mathbf{A}_{12}^i = fr_{12} + (u_0 - u_i)r_{32}$, $\mathbf{A}_{13}^i = fr_{13} + (u_0 - u_i)r_{33}$, $\mathbf{A}_{21}^i = fr_{21} + (v_0 - v_i)r_{31}$, $\mathbf{A}_{22}^i = fr_{22} + (v_0 - v_i)r_{32}$, $\mathbf{A}_{23}^i = fr_{23} + (v_0 - v_i)r_{33}$, $\mathbf{B}_1^i = (u_i - u_0)(\mathbf{R}^3\mathbf{X}_i + t_3) - f(\mathbf{R}^1\mathbf{X}_i + t_1)$, $\mathbf{B}_2^i = (v_0 - v_i)(\mathbf{R}^3\mathbf{X}_i + t_3) - f(\mathbf{R}^2\mathbf{X}_i + t_2)$. Note that r_{ij} and $\mathbf{R}^i$ are the element and the ith row vector of rotation matrix $\mathbf{R}$, respectively.

It is clear that for a tuple of coefficients from $\mathbf{T}^{disp}$, two equations can be obtained. Therefore, when there are n tuples in the set $\mathbf{T}^{disp}$, $2\times n$ linear equations will generate. According to linear algebra, a simplified form expresses as

$$\mathbf{Ad} = \mathbf{B}, \tag{9}$$

where $\mathbf{c} = [c_x, c_y, c_z]^{\mathbf{T}}$ is the unknown vector. A direct least-square solution can solve as

$$\mathbf{d} = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}}\mathbf{B}. \tag{10}$$

Finally, we have

$$\mathbf{x}_s^r = \mathbf{x}_s + \mathbf{d}, \tag{11}$$

where $\mathbf{x}_s^r$ is the refreshed location of the displaced visual fingerprint, $\mathbf{x}_s$ is the primitive one in the database. It defines as our benchmark, which could be called the DLS method.

Our proposed method will deduce from the PnP equation, which is

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K}(\mathbf{R} \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix} + \mathbf{t}). \tag{12}$$

Noted that the point coordinates $[x^i, y^i, z^i]^{\mathbf{T}}$ are the projection of the image coordinates $[u_i, v_i]$ in the crowdsourced query

image by the depth λ_i, rotation matrix $\mathbf{R}$, and translation vector $\mathbf{t}$. Be different from equation (6), the goal is to calculate the point coordinates of each semantic feature in the world coordinate system. From equation (12), we have 3 equations and 4 unknowns, which represent infinite solutions. Thus, we formulate a minimization problem as our goal of a solution, which is

$$\min \sum_{j,k,j\neq k}^{3} \left(\lambda_i^j - \lambda_i^k\right)^2, \tag{13}$$

where λ_i^j donates the jth expression of λ_i from equation (12). Typically, (13) is a quadratic programming problem. The optimal solution can obtain by solving the first derivative of (13). The final simplified form is 3 equations of the first degree, which is quite easy for computation. With n tuples of projected point coordinates, the center can obtain as $\mathbf{c}^p$, while the primitive center of the displaced object is $\mathbf{c}^r$. The translation vector of the center coordinate $\mathbf{c}$ represents as

$$\mathbf{c} = \mathbf{c}^p - \mathbf{c}^r. \tag{14}$$

Generally, this value is used as an index to measure the accuracy of the method, since $\mathbf{c}$ is predefined in the simulation. The proposed method summarizes in **Algorithm 2**.

Algorithm 2 The Proposed visual fingerprint relocation method

Input: $\mathbf{R}$, $\mathbf{t}$, $\mathbf{T}^{disp} = \{u_i, v_i\}$
Output: $\mathbf{x}_s^r$

```
Calculate each expression of λ_i from equation (12)
Simplify the first derivative of expression (13)
for i = 1; i < m; i++ do
    Substitute the coefficients into the ternary system of first-order equations and calculate each c_i
Calculate the center c^p
Obtain the translation c
return x_s^r
```

III. Experiments

A. Comparison algorithm

Some researchers have proposed visual localization methods by the leverage of known prerequisites, such as man height [41] or vertical direction [42]. A traditional idea is borrowed from this kind of setting. On the contrary, we can judge whether the localization result is right or wrong by the known prerequisites. However, in the application of pedestrian visual localization, these prerequisites will not be satisfied all time. Fortunately, the localization error still could be utilized for ranking the displaced object. Typically, the localization result is with an error in every direction. Since the user is unaware of his real location in the horizontal or vertical direction, a traditional method can only leverage the distance perception of the human in the vertical direction for judging the accuracy of the localization result. Thus, the conventional trick is to set an error threshold in the vertical direction. Once the result is beyond the threshold, it will determine that some mismatch exists in the 2D-3D correspondences. The threshold method referred to in the comparison is a concrete realization of the traditional one.

B. Synthetic data

To minimize the impact of related parts on our proposed method, firstly the experiment is conducted on the synthetic data for convenience. Moreover, it could assume that the 2D-3D corresponding match and semantic segmentation have 100% accuracy in the synthetic data set, respectively. Thus, the focus will concentrate on the visual fingerprint displacement detection and relocation method. The evaluation will also be more accurate.

For simplicity, we assume that the first camera is set as the origin of the world coordinate system, while the second camera assumes to be translated several units from the first camera along the x-axis. The reason is that the two cameras need to keep a certain distance to keep the overlap between the images. The camera coordinates of 3D points are generated randomly within a box of $[-2, 2] \times [-2, 2] \times [6, 8]$, then the pixel coordinate of the two cameras is projected by a pinhole camera model with an initial point $(960, 540)$ and focal length around 1000. The y-axis angle of the second camera chooses randomly from $-10^\circ - 0^\circ$, while the angle of the first camera varies from $0^\circ - 10^\circ$. In each trial, 500 points generate. Then, the K-means algorithm applies to clustering points. For a cluster, they all belong to the same object as a common assumption. Consequently, any object can be used to simulate a displaced one. The shifted distance is set to $0.5 - 1$ units far from its original position. Suppose that the object displaces too far or a new one places, it will not exist in the crowdsourced image at that particular location. These two kinds of situations are beyond the scope of this paper. The setting of the crowdsourced camera is the same with the second camera. Both the displaced points and fixed ones project on the image plane. With these image coordinates and their corresponding points, the location can be calculated by a PnP algorithm like EPnP [33].

We divided the synthetic data simulation into two parts according to the number of displaced objects. There is only one displaced object in the first part, which is more likely in the indoor environment. There are two displaced objects at the same time in the other part, which is more complex. Both results were achieved under the condition of $\phi_{thr} = 5\%$, $\epsilon_{reproj}^{thr} = 100$, $\epsilon_{diff}^{thr} = 1$, which are the optimal setting in our simulations. We find that when ϕ_{thr} is bigger, the RANSAC threshold method will report false alarm in simulations using synthetic data, especially when the number of 2D-3D correspondences is originally small. Meanwhile, $\epsilon_{reproj}^{thr} = 100$ and $\epsilon_{diff}^{thr} = 1$ are to tolerate the errors introduced by 2D-3D matching procedure for the real dataset. The threshold could be set lower in the simulations with synthetic data, e.g. $\epsilon_{reproj}^{thr} = 20$ in the primitive implementation of EPnP [33].

C. Real Dataset

The real dataset chooses from the image shot at the communication research center in Harbin Institute of Technology, whose floor plan and layout show in **Fig. 1**. The experiment site is mainly the area marked yellow on the plan. The main aisle is about 50m long and 3m wide. To simulate the smart vehicles for convenience, we use wheeled equipment instead,

which mounted the camera on its roof. The training set is a total of 800 images, whose image resolution is 1920×1080. Some samples of the reference image have been shown in **Fig. 3** and **Fig. 4** previously. By leveraging semantic segmentation, we defined 9 kinds of objects in R-FCN. The statistics used in the training step are shown below in **Table I**.

TABLE I The number of different semantic segmentations training in R-FCN

Semantic label	Training number
Door	539
Window	149
Fire extinguisher	30
Ashbin	22
Vent	26
Poster	186
Hydrant	66
Heating	138
Exhibition board	321
Exit sign	63

The learning parameters in the R-FCN have selected default values. The *Caffe* framework is trained on a server with Inter(R) Xeon(R) Gold 5118 CPU @2.3GHz, memory 64GB, NVIDIA SMI 418.56 GPU, and 64 bits Ubuntu OS. More details about the R-FCN configurations for our simulation environments can be referred to in our previous conference paper [40].

IV. Results and Discussion

A. Results from synthetic dataset

Fig. 5-**Fig. 7** show different comparison results under the assumption of one displaced object and two fixed ones from the synthetic dataset. According to the proposed visual fingerprint updating algorithm, the first step is to find the displaced visual fingerprint from a single crowdsourced image. **Fig. 5** shows the detection probability curve w.r.t. the threshold value between the traditional method and our proposed method. As mentioned

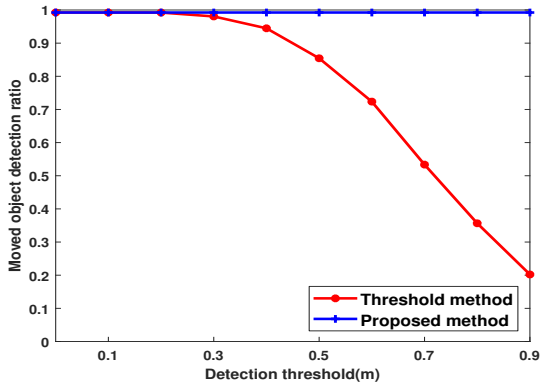

Fig. 5 Detection ratio comparison between our proposed and the threshold method with one displaced object.

before, the traditional one uses all 2D-3D correspondences as the coefficients of the PnP equations. Furthermore, the solution is provided by the PnP solver. In our simulation, EPnP is chosen due to its accuracy and efficiency. For each threshold sampling point, the results are obtained from 5000 repeated random trials. The traditional method is more sensitive to the human-perceivable threshold, which means the performance is disturbed by the threshold value. The threshold value is also contradictory. The bigger is the value, the more sensitive is the crowdsourced user. However, the detection probability decreased dramatically with the increase of the threshold value. The threshold is setted from 0 to 0.9m, the reason is that with the help of an infrared or laser distance sensor equipped with the device, the localization error could be perceived. Moreover, a human will sense at least a 20cm error in the vertical direction.

In **Fig. 6**, the logarithmic localization error of different 2D-3D correspondences shows, respectively. The mean and maximum errors are used for comparison. It concludes that the lo-

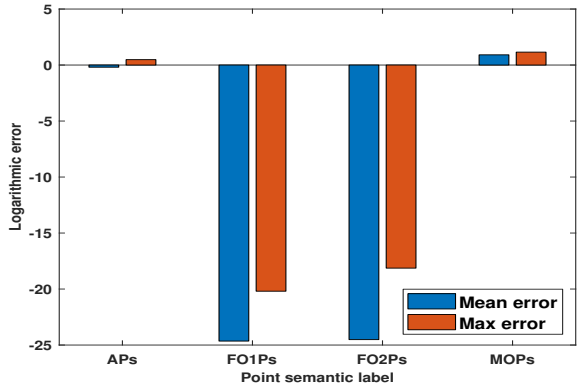

Fig. 6 Localization error CDF under logarithmic level by EPnP with one displaced object.

calization error is small when the 2D-3D correspondences are from the fixed objects, whose point label represents $FOP1s$ and $FOP2s$ in **Fig. 6**. The logarithmic error is positive when the 2D-3D correspondences are from a displaced object, which labels as $MOPs$. The difference is obvious when the 2D-3D correspondences are from fixed objects and displaced ones. The localization result is hard to distinguish whether the positioning results are correct when the correspondences are mixed, which labels as APs. From the results, it is easy to tell which object displaces with its semantic label.

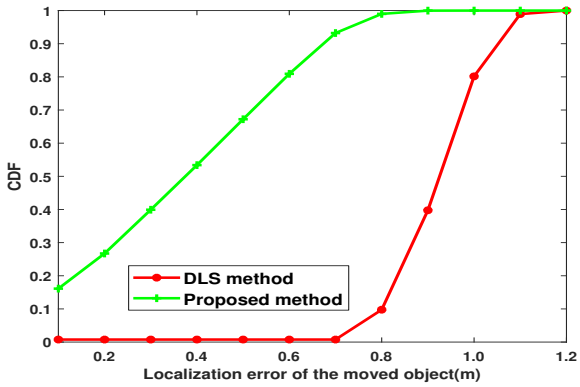

Fig. 7 Localization error comparison by different visual fingerprint relocation method when there is one displaced object.

Once the displaced object is locked, the next step in our proposed algorithm is to relocate the point cloud fingerprint. In the previous step, the ground truth rotation matrix and translation vector of the crowdsourced image can calculate

correspondences from the fixed objects. The process is followed by RANSAC for all correspondences. It makes our method of semantic 2D-3D correspondences more intuitive and efficient. Since the translation vector of the displaced object is predefined, the error could be calculated between the ground truth and the solved one. The predefined translation vector varies by a random value between 0.5 and 0.8 in the x-axis and y-axis. The cumulative probability density curve shows in **Fig. 7**. The error CDF curve of our proposed method is much better than the benchmark. It illustrates that the position of the fingerprint is refreshed by our proposed method more accurately.

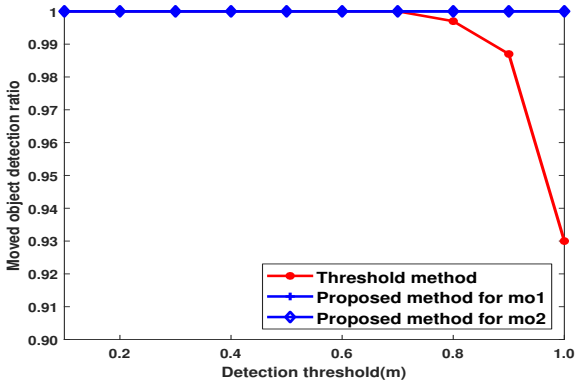

Fig. 8 Detection ratio comparison between our proposed and the threshold method with two displaced objects.

The second part of the synthetic dataset results shows in **Fig. 8**-**Fig. 10**. There are two unmoved objects and two displaced objects, which means more mismatches are in 2D-3D correspondences. The number of all 2D-3D correspondences is 500. Noted that to ensure the normal solution of EPnP, at least 10 points are generated on each object. From **Fig. 8**, the detection probability of both displaced objects is higher than the one of the traditional method. With more mismatched correspondences, the detection ratio of the traditional method promotes indeed from the comparison of **Fig. 5** and **Fig. 8**. However, the trend is the same. Meanwhile, the performance of our proposed method remains unchanged.

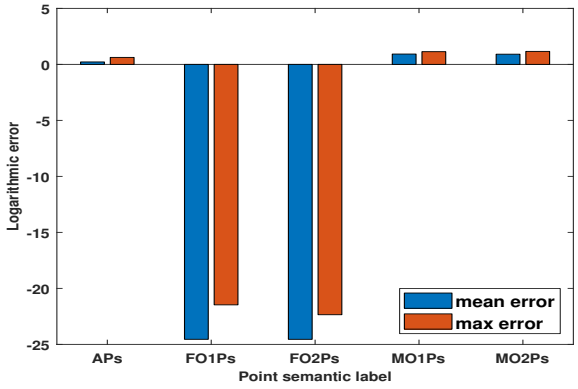

Fig. 9 Localization error under logarithmic level by EPnP with two displaced objects.

Fig. 9 shows the logarithmic error of the localization results from different bundles of 2D-3D correspondences. Comparing with **Fig. 6**, the error between fixed and displaced objects still exists, which also will be enabled to lock the displacement object accurately.

In **Fig. 10**, the CDF curves of relocation error from different displaced object draw. The relocation error of the two displaced objects by our proposed method diverges slightly, whose computation results are both better than that of the traditional one. Advantageously, the performance preserves well when in comparison with **Fig. 7**.

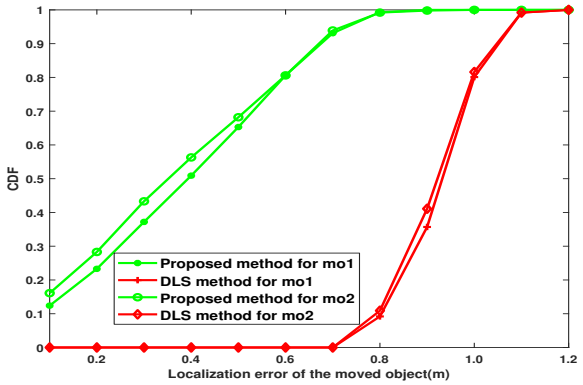

Fig. 10 Localization error CDF comparison by different visual fingerprint relocation method when there are two displaced objects.

B. Results from real dataset

The detection result from real dataset shows in **Fig. 11**. To simplify the process of the simulation, we select a random location in the scene, which contains four semantic objects. One object is moved deliberately to a specific location. It should be noted that the displacement of the object will also happen in normal time, and the chosen location is only for the convenience of measuring the ground truth.

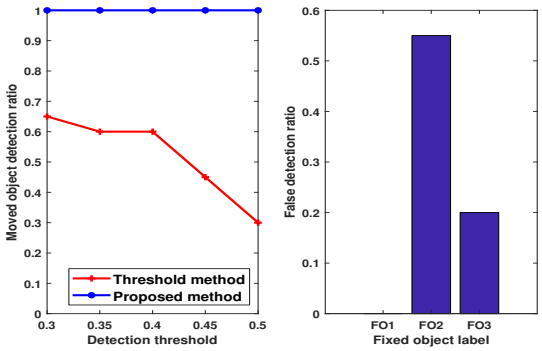

Fig. 11 The detection ratio comparison between our proposed and the threshold method, in addition a false ratio is also shown in this figure when the simulation is running in the real data set.

When the semantic object shifts, 50 test images are collected near the reference location, whose precise locations are unknown. It shows clearly that the successful detection ratio of the displaced object is independent of the human perceivable threshold by our proposed method, while the ratio of the comparing one decreases indeed with the increase of the threshold. However, unlike synthetic data simulation, the fixed objects are misjudged. There are 55% and 30% misjudgments from fixed objects labeled as $FO2$ and $FO3$, respectively. The main reason for this difference is that the number of

effective semantic features on $FO2$ and $FO3$ is smaller in real data simulation than in the synthetic one. The average percentage number of features on $FO2$ is 5.95%, while that on $FO3$ is 17.64%, which are both much smaller than $FO1$. The reprojection errors of $FO2$ and $FO3$ can obtain by using the location results of $FO1$, which can eliminate such misjudgments. Furthermore, it could reduce to 5% and 20%.

Fig. 12 describes the localization result of the displaced object. Our proposed method outperforms the compared method

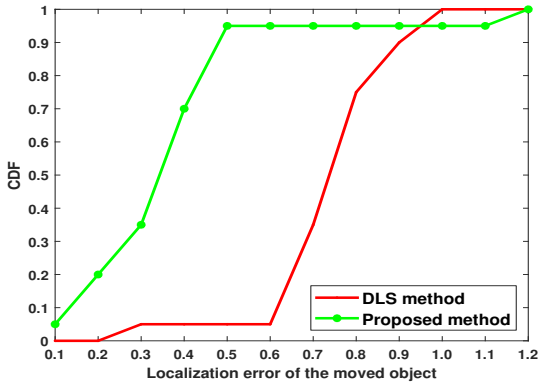

Fig. 12 Localization error CDF comparison of the displaced object when the simulation is running by real data set.

when the error is smaller than 0.94m. Compared with the simulation results of synthetic data, the convergence of the Cumulative Distribution Function (CDF) curve is slower than that of the compared method in real data simulation. In summary, the performance of our proposed method on the real data is worse than that by the synthetic data. The reason is that the feature matching algorithm fails to provide 100% accuracy, which will lead to misjudgments and distortion of equation coefficients. As a result, the solution of our proposed method will have a few deviations.

C. Operation efficiency

As known, the computation complexity of EPnP is $O(n)$. It is supposed that the type of semantic segmentation is m. Typically, it regards as $m \leq 10$ in an indoor environment, which is similar to our experiment place. Thus, according to **Algorithm 1**, the computation complexity of our proposed visual fingerprint displacement detection method is $O(m \times n)$. Meanwhile, the computation complexity of the proposed visual fingerprint relocation method is $O(n)$ by **Algorithm 2**, where n is the number of visual fingerprints belonging to the displaced object. The average running time trend w.r.t. the corresponding 2D-3D point number n shows in **Fig. 13**. The running time of each sampling point obtained from the average of 1000 trials. With the increase of the feature number that extracts from the images, the time required by both methods increases slowly. Meanwhile, the number of displaced objects does not significantly affect the performance of the method. The experiment results show that the proposed method is efficient. However, compared with other methods in the proposed algorithm in this paper, the average running time of semantic segmentation by R-FCN is still long, which is 1.63s. It makes the algorithm be only able to run on the server side temporarily.

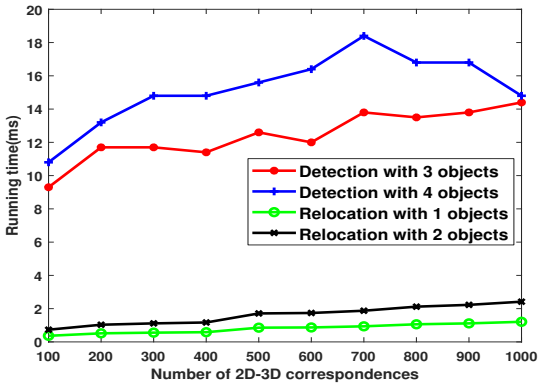

Fig. 13 Running time comparison of our proposed two methods with different displaced object w.r.t. number of 2D-3D correspondences.

D. Comparison with AVF reconstruction

In this subsection, we list the disadvantages and advantages of our proposed visual fingerprint updating algorithm and reconstruction by AVF method, which can be seen from **Table II**. The result of average processing time and relocation error

TABLE II The comparison result by using AVF reconstruction method and our proposed algorithm

Feature	Reconstruction by AVF proposed in [9]	Crowdsource based algorithm
Participant	specialist	crowdsourced user
Periodicity	yes	no
Location perception of the displaced object	no	yes
Average processing time(s)	1221	2.6
Mean relocation error(m)	0.35	0.43

is from 500 trials. In the experiment, we set one moving object, whose translation varies from 50cm to 100cm. Under this condition, it is convenient to calculate the relocation error of the fingerprint. It should be noted that the AVF method is easily affected by the flow of people in the scene. By contrast, our proposed algorithm is more flexible with the aim of crowdsourced localization.

V. Conclusion

In this paper, we propose an algorithm based on crowdsourcing and deep learning for solving the challenging visual fingerprint update problem of smart vehicle, which aims to detect whether the reference object in the crowdsourced image is displaced and provide a refreshed location to facilitate subsequent vehicles. The simulation results are achieved thoroughly from synthetic data with various configurations. Besides, a real indoor dataset is applied to test the performance of our proposed algorithm compared with synthetic data. In summary, our proposed algorithm can promote nearly 100% detection probability, while the average probability by threshold method is 60%. The accuracy of relocated fingerprints by our proposed algorithm is 42% higher than the DLS method. Although the accuracy of our proposed algorithm is 10% lower than the AVF reconstruction method, our proposed algorithm outperforms in other aspects. In future research, the influence of the rotation of

the displacement object will be considered, which will further refine the refreshed fingerprint location.

VI. Abbreviations

IoV: Internet of Vehicle
AVF: Automatic Visual Fingerprinting
R-FCN: Region-based Fully Convolutional Network
QP: Quadratic Programming
RSSI: Recieved Signal Strength Indication
SNR: Signal Noise Ratio
CSI: Channel State Information
SURF: Speed-Up Robust Feature
PnP: Perspective-n-Point
SIFT: Scale Invariant Feature Transform (SIFT)
FLANN: Fast Library for Approximate Nearest Neighbor
SfM: Structure from Motion
EPnP: Efficient PnP
SVD: Singular Value Decomposition
RANSAC: RANdom SAmple Consensus
DL: Deep Learning
RoI: Region of Interest
CDF: Cumulative Distribution Function

References

[1] G. Huang, Z. Z. Hu, J. Wu, H. B, Xiao and F. Zhang, "WiFi and Vision Integrated Fingerprint for Smartphone-Based Self-Localization in Public Indoor Scenes," IEEE Internet Things J., vol. 7, no. 8, pp. 6748-6761, Aug. 2020.

[2] J. Dong, M. Noreikis, Y. Xiao and A. Y.-Jääski, "ViNav: A Vision-Based Indoor Navigation System for Smartphones," IEEE Trans. Mobile Comput., vol. 18, no. 6, pp. 1461-1475, Jun. 2019.

[3] Y. L. Shi, W. M. Zhang, F. X. Li and Q. Huang, "Robust Localization System Fusing Vision and Lidar Under Severe Occlusion," IEEE Access, vol. 8, pp. 62495-62504, Mar. 2020.

[4] X. X. Zuo, P. Geneva, Y. L. Yang, W. L. Ye, Y. Liu and G. Q. Huang, "Visual-Inertial Localization With Prior LiDAR Map Constraints," IEEE. Robot. Autom. Lett., vol. 4, no. 4, pp. 3394-3401, Oct. 2019.

[5] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger and E. Steinbach, "TUMindoor: An extensive image and point cloud dataset for visual indoor localization," in Proc. of the 19th IEEE Int. Conf. Image Processing. (ICIP), Orlando, FL, USA, Oct. 2012, pp. 1773-1776.

[6] H. Xue, L. Ma and X. Z. Tan, "A fast visual map building method using video stream for visual-based indoor localization," in Proc. of the Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), Paphos, Cyprus, Sep. 2016, pp. 650-654.

[7] J. Z. Liang, N. Corso, E. Turner and A. Zakhor, "Image Based localization in indoor environments," in Proc. of the 4th Int. Conf. on Comput. for Geospatial Research and Application, San Jose, CA, USA, Jul. 2013, pp. 70-75.

[8] F. Vedadi and S. Valaee, "Automatic Visual Fingerprinting for Indoor Image-Based Localization Applications," IEEE Trans. Syst., Man Cybern. Syst., vol. 50, no. 1, pp. 305-317, Jan. 2020.

[9] X. L. Yin, L. Ma, X. Z. Tan and D. Y. Qin, "A SOCP-Based Automatic Visual Fingerprinting Method for Indoor Localization," IEEE ACCESS, vol. 7, pp. 72862-72871, Jun. 2019.

[10] G. Caso, L. D. Nardis, F. Lemic, V. Handziski, A. Wolisz and M-G. D. Benedetto, "ViFi: Virtual Fingerprinting WiFi-Based Indoor Positioning via Multi-Wall Multi-Floor Propagation Model," IEEE Trans. Mobile Comput., vol. 19, no. 6, pp. 1478-1491, Jun. 2020.

[11] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson and M. Z. Win, "A Belief Propagation Algorithm for Multipath-Based SLAM", IEEE Trans. Wireless Commun., vol. 18, no. 12, pp. 5613-5629, Dec. 2019.

[12] Y. X. Duan, K.-Y. Lam, V. C. S. Lee, W. D. Nie, H. Li and J. K. Y. Ng, "Multiple Power Path Loss Fingerprinting for Sensor-Based Indoor Localization", IEEE Sensors Lett., vol. 1, no. 4, pp. 1-4, Aug. 2017.

[13] B. Xu, X. R. Zhu, and H. B. Zhu, "An Efficient Indoor Localization Method Based on the Long Short-Term Memory Recurrent Neuron Network," IEEE Access, vol. 7, pp. 123912-123921, Aug. 2019.

[14] T. K.-Akino, P. Wang, M. Pajovic, H. J. Sun, and P. V. Orlik, "Fingerprinting-Based Indoor Localization With Commercial MMWave WiFi: A Deep Learning Approach," IEEE Access, vol. 8, pp. 84879-84892, Apr. 2020.

[15] X. Y. Wang, L. J. Gao, S. W. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," IEEE Trans. Veh. Technol., vol. 66, no. 1, pp. 763-776, Jan. 2017.

[16] C. S. Wu, Z. Yang, and Y. H. Liu, "Smartphones Based Crowdsourcing for Indoor Localization," IEEE Trans. Mobile Comput., vol. 14, no. 2, pp. 444-457, Feb. 2015.

[17] Y. Zhuang, Z. Syed, Y. Li, and N. E.-Sheimy, "Evaluation of Two WiFi Positioning Systems Based on Autonomous Crowdsourcing of Handheld Devices for Indoor Navigation," IEEE Trans. Mobile Comput., vol. 15, no. 8, pp. 1982-1995, Aug. 2016.

[18] W. L. Zhao, S. Han, R. Q. Hu, W. X. Meng, and Z. Q.-Jia, "Crowdsourcing and Multisource Fusion-Based Fingerprint Sensing in Smartphone Localization," IEEE Sensors J., vol. 18, no. 8, pp. 3236-3247, Apr. 2018.

[19] B. Q. Huang, Z. D. Xu, B. Jia, and G. Q. Mao, "An Online Radio Map Update Scheme for WiFi Fingerprint-Based Localization," IEEE Internet Things J., vol. 6, no. 4, pp. 6909-6918, Aug. 2019.

[20] Y. H. Zhao, W.-C. Wong, T. Y. Feng, and H. K.-Garg, "Calibration-Free Indoor Positioning Using Crowdsourced Data and Multidimensional Scaling," IEEE Trans. Wireless Commun., vol. 19, no. 3, pp. 1770-1785, Mar. 2020.

[21] J. F. Dai, L. Yi, K. M. He and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in Proc. of the Int. Conf. Neural Inf. Process. Syst.(NIPS), Barcelona, Spain, 2016.

[22] T. Wang, Y. T. Yao, Y. Chen, M. Y. Zhang, F. Tao and H. Snoussi,, "Auto-Sorting System Toward Smart Factory Based on Deep Learning for Image Segmentation," IEEE Sensors J., vol. 18, no. 20, pp. 8493-8501, Oct. 2018.

[23] G. Cheng, J. W. Han, P. C. Zhou and D. Xu, "Learning Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection," IEEE Trans. Image Process., vol. 28, no. 1, pp. 265-278, Jan. 2019.

[24] M. Braun, S. Krebs, F. Flohr and D. M. Gavrila, "EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1844-1861, Aug. 2019.

[25] C. Zhang, G. Y. Sun, Z. M. Fang, P. P. Zhou, P. C. Pan, and J. Cong, "Caffeine: Toward Uniformed Representation and Acceleration for Deep Convolutional Neural Networks," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 11, pp. 2072-2085, Nov. 2019.

[26] Y. Wang, W. X. Chen, J. Yang, and T. Li, "Exploiting Parallelism for CNN Application on 3D stacked Processing-In-Memory Architecture," IEEE Trans. Parallel Distrib. Syst., vol. 30, no. 3, pp. 589-600, Mar. 2019.

[27] M. Komar, P. Yakobchuk, V. Golovko, V. Dorosh, and A. Sachenko, "Deep Neural Network for Image Recognition Based on the Caffe Framework," in Proc. of the IEEE. 2nd Int. Conf. Data Stream Mining & Processing. (DSMP), Lviv, Ukraine, Aug. 2018, pp. 102-106.

[28] (Mar. 9, 2020). Caffe. Accessed on Mar. 9, 2020. [online]. Available: http://caffe.berkeleyvision.org

[29] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 9, pp. 1744-1756, Sept. 2017.

[30] D. G. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints," Int. J. Comput. Vis., vol. 60, pp. 90-110, Nov. 2004.

[31] H. Bay, T. Tuytelaars, and L. V. Gool, "Speed-Up Robust Features (SURF)," in Proc. of the European Conf. Comput. Vis. (ECCV), Graz, Austria, May. 2006, pp. 404-417.

[32] M. Muja, and D. G, Lowe, "Scalable Nearest Neighbor Algorithms for High Dimensional Data," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 11, pp. 2227-2240, Nov. 2014.

[33] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP problem," Int. J. Comput. Vis., vol. 81, no. 2, pp. 155-166, 2009.

[34] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,"Commun. ACM, vol. 24, pp. 381-395. 1981.

[35] A. B. -Hernández, G. H. -Peñaloza, D. M. Gutiérrez, and F. Álvarez, "SWiBluX: Multi-Sensor Deep Learning Fingerprint for Precise Real-Time Indoor Tracking," IEEE Sensors J., vol. 19, no. 9, pp. 3473-3486, May. 2019.

[36] C. Ma, J.-B. Huang, X. K. Yang, and M.-H. Yang, "Robust Visual Tracking via Hierarchical Convolutional Features," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 11, pp. 2709-2723, Nov. 2019.

[37] R. H. Li, S. Wang, and D. B. Gu, "DeepSLAM: A Robust Monocular SLAM System with Unsupervised Deep Learning" IEEE Trans. Ind. Electron., vol. 68, no. 4, pp. 3577-3587, Apr. 2021.
[38] G. Costante, M. Mancini, "Uncertainty Estimation for Data-Driven Visual Odometry," IEEE Trans. Robot., vol. 36, no. 6, pp. 1738-1757, Dec. 2020.
[39] N. Radwan, A. Valada, and W. Burgard, "VLocNet++:Deep Multitask Learning for Semantic Visual Localization and Odometry," IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 4407-4414, Sep. 2018.
[40] J. Dai, L. Ma, D. Y. Qin, and X. Z. Tan, "High Accurate and Efficient Image Retrieval Method Using Semantics for Visual Indoor Positioning," in Proc. of the IEEE. 8th Int. Conf. Commun., Signal Process., Syst. (CSPS), Urumqi, China, Jul. 2019, pp. 128-136.
[41] M. B. Qi, R. Zhang, J. G. Jiang, and X. T. Li, "Research of Rotatable Single Camera Object Localization Based On Man Height Model in Visual Surveillance," in Proc. of the 2nd Int. Conf. Bioinformatics and Biomedical Engineering, Shanghai, China, 2008, pp. 1131-1134.
[42] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-Scale Localization for Cameras with Known Vertical Direction," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 7, pp. 1455-1461, Jul. 2017.
[43] J. Wang, C. Jiang, H. Zhu, Y. Ren and L. Hanzo, "Internet of vehicles: Sensing-aided transportation information collection and diffusion", IEEE Trans. Veh. Technol., vol. 67, no. 5, pp. 3813-3825, May 2018.
[44] K. N. Qureshi, S. Din, G. Jeon and F. Piccialli, "Internet of Vehicles: Key Technologies, Network Model, Solutions and Challenges With Future Aspects," IEEE Trans. Intell. Transportation Syst., vol. 22, no. 3, pp. 1777-1786, Mar. 2021.

VII. Declarations

Ethics Approval and Consent to Participate

Not applicable.

Consent for Publication

Not applicable.

Availability of Data and Materials

The datasets used for the evaluation of the algorithm are not sharing publicly. Please contact the corresponding author if neccessary.

Competing Interests

The authors declare that they have on competing interests.

Funding

This paper is supported by National Natural Science Foundation of China (61971162, 61771186, 4181101180).

Author's Contributions

XiLiang Yin developed and implemented the core concepts of the algorithm presented within this manuscript, Lin Ma provideed refinements and proofreadings, Ping Sun provided debugging of deep learning network. All authors read and approved the final manuscript.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions.

Author's Information

XIliang Yin is pursuing his Ph. D. in information and communication engineering, specialises in location-based service, machine learning, and computer vision, and is a lecturer in Harbin Vocational & Technical College, China. Lin Ma has a Ph. D in communication engineering, specialises in location-based service, cognitive radio, and cellular networks, and is currently an Associate Professor with the school of electronics and information engineering, Harbin Institute of Technology, China. Ping Sun is pursuing her M. S., specialises in deep learning, and is a graduate student in the school of electronics and information engineering, Harbin Institute of Technology, China.

Author's Affiliations

Harbin Institute of Technology, School of Electronics and Information Engineering
Xiliang Yin, Lin Ma & Ping Sun
Harbin Vocational & Technical College
Xiliang Yin

Corresponding author

Correspondence to Lin Ma.