

Research Article

Visual Localization by Place Recognition Based on Multifeature (D- λ LBP++HOG)

Yongliang Qiao¹ and Zhao Zhang²

¹Laboratoire Électronique, Informatique et Image, Université de Technologie de Belfort-Montbéliard, 90000 Belfort, France

²College of Electron and Electricity Engineering, Baoji University of Arts and Sciences, Baoji 721016, China

Correspondence should be addressed to Yongliang Qiao; yongliang.qiao@utbm.fr

Received 6 June 2017; Accepted 17 August 2017; Published 22 October 2017

Academic Editor: Stéphane Evoy

Copyright © 2017 Yongliang Qiao and Zhao Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Visual localization is widely used in the autonomous navigation system and Advanced Driver Assistance Systems (ADAS). This paper presents a visual localization method based on multifeature fusion and disparity information using stereo images. We integrate disparity information into complete center-symmetric local binary patterns (CSLBP) to obtain a robust global image description (D-CSLBP). In order to represent the scene in depth, multifeature fusion of D-CSLBP and HOG features provides valuable information and permits decreasing the effect of some typical problems in place recognition such as perceptual aliasing. It improves visual recognition performance by taking advantage of depth, texture, and shape information. In addition, for real-time visual localization, local sensitive hashing method (LSH) was used to compress the high-dimensional multifeature into binary vectors. It can thus speed up the process of image matching. To show its effectiveness, the proposed method is tested and evaluated using real datasets acquired in outdoor environments. Given the obtained results, our approach allows more effective visual localization compared with the state-of-the-art method FAB-MAP.

1. Introduction

One of the prerequisites of navigation issue is to make the vehicle or robot able to reliably determine its position within its environment. With the wide use of cameras, varieties of approaches were proposed to address the challenges of place recognition based visual localization [1, 2].

FAB-MAP (Fast Appearance Based Mapping) method [3] can be considered as the milestone in the field of visual localization. FAB-MAP approach consists of matching the appearance of current scene to the same (similar) past visited place by converting the images into bag-of-words representations built on local features such as SIFT or SURF.

Recently, binary image descriptors that encode patch appearance, using compact binary string with low memory requirements, are widely used in image description and visual recognition [4]. In local feature based place recognition approaches, image representation is defined as collection of local features which contribute to their robustness when faced with local image variations as well as from discriminative

power of their descriptors. Nevertheless, most of these works exhibit a high computation cost or complex feature extraction for image matching [5, 6]. Also, few works pay attention to the depth information for visual place recognition.

Their advantages are that they are invariant to monotonic changes in gray-scale and fast to calculate. One typical binary descriptor is LBP (local binary pattern) [7]. Since it was firstly proposed in 1996, several new variants of binary descriptors have been proposed [8, 9]. They show great invariance to monotonic illumination changes, do not require many parameters to be set, and have a high discriminative power. However, most of them are unfortunately not efficient for background modeling or place describing because of their sensitivity to noise or illumination. In this paper, the most relevant binary descriptors for visual place recognition that will be tested and compared in our approach are LBP, CLBP (complete local binary pattern) [10], CSLBP (center-symmetric local binary patterns) [11], CSLDP (center-symmetric local derivative pattern) [12], and XCSLBP (extended CSLBP) [13]. These different local binary descriptors are noted as λ LBP.

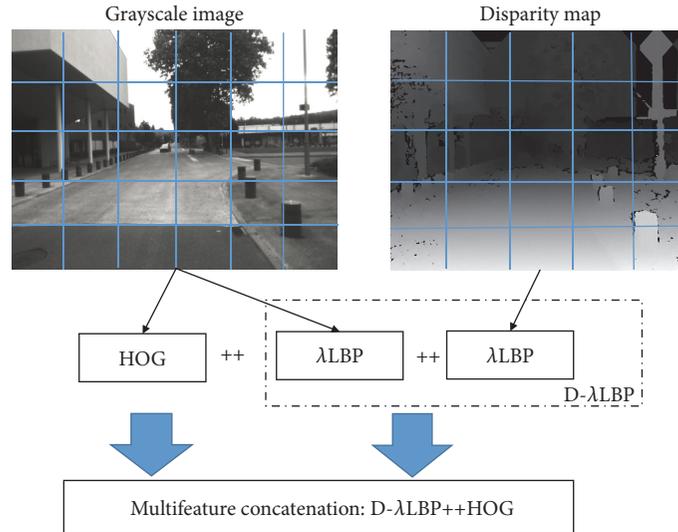


FIGURE 1: Multifeature built from gray-scale image and disparity map. Features are firstly extracted from each image block and then concatenated together. The symbol “++” means concatenation.

Despite local binary features efficiency, histograms of oriented gradients (HOG) features have also been successfully used in various vision tasks such as object classification, image search, and scene classification [14]. Wang et al. [15] combine histograms of oriented gradients (HOG) and local binary pattern (LBP) and propose a novel human detection approach capable of handling partial occlusion. For such applications, HOG is one of the best features to capture edge or local shape information which provides a rough description (shape information) of the scene.

Considering the robust and strong image representation ability of binary descriptors and HOG feature, we expect that their combination would provide more useful information and then should improve place recognition performance. In this paper, stereo images are used for visual place recognition. A novel localization approach is then proposed which uses multifeature fusion by combining HOG and binary features (λ LBP), as shown in Figure 1. HOG features are obtained from gray-scale image while λ LBP features are built from both gray-scale image and disparity map. We note that the features are first extracted from the blocks composing the gray-scale image and the disparity map and then concatenated. We extend the application of λ LBP descriptor to disparity map in order to incorporate disparity information in image representation by simply concatenating the two descriptors (λ LBP from gray-scale image and λ LBP from disparity map). This produces a new descriptor named D- λ LBP. The integration of disparity information in image representation provides depth information which should be helpful for place recognition, especially in complex environment situation. Indeed, image description using features λ LBP and HOG and the depth information will permit reducing perceptual aliasing problems related to visual place recognition. As will be shown in our experiments, features combination permits achieving better recognition performance than single feature. Also the performance of place

recognition is compared with the state-of-the-art FAB-MAP algorithm: the achieved F_1 scores on four tested datasets using our approach are better than those resulting from FAB-MAP method. Furthermore, considering that high-dimensional multifeatures comparison is time-consuming, locality sensitive hashing is applied on multifeatures to speed up the process of features comparison and image matching.

The most important contributions introduced in this paper are the following:

- (i) An innovative method for place recognition based visual localization using multifeature descriptor (D- λ LBP++HOG) extracted from gray-scale image and disparity map. The proposed multifeature descriptor takes advantage of texture, depth, and shape information and hence performs better than single feature (see Section 5.2).
- (ii) The impact image block size for the binary descriptors is studied. Binary descriptor extracted from small block has better discriminative ability in local details of different locations, while considering large block size for image representation may cause loss of some discriminative information (see Section 5.1).
- (iii) A speeding-up of the place recognition method is achieved by approximating the Euclidean distance between features with hamming distance over bit vectors obtained by locality sensitive hashing (see Section 5.3).

The rest of this paper is organized as follows. Firstly, the LBP descriptor and several of its variants as well as HOG feature are introduced in Section 2. Then, in Section 3, the proposed approach is described in detail. Section 4 deals with the presentation of the tested database and the used performance evaluation parameters. The obtained results are presented and discussed in Section 5. Finally, conclusions and future works close this paper (Section 6).

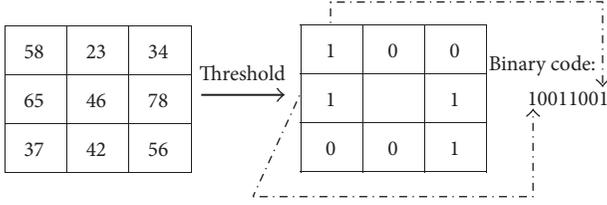


FIGURE 2: Illustration of the basic LBP operator.

2. Overview of Used Image Descriptors

In this part, some of the state-of-the-art image descriptors used and compared in the proposed approach are described.

2.1. LBP (Local Binary Pattern). LBP is a texture descriptor that codifies local primitives (such as curved edges, spots, and flat areas) into a feature histogram. The original LBP operator labels the pixels of an image with decimal numbers, called local binary patterns or LBP codes, which encode the local structure around each pixel [8].

As illustrated in Figure 2, each pixel gray level value is compared with its eight neighbors in a 3×3 region by subtracting the center pixel value. The resulting strictly negative values are encoded with 0 and the others with 1. A binary number is obtained by concatenating all these binary codes, and its corresponding decimal value is used for labeling the central pixel. In Figure 3, examples of neighborhood used for LBP operator are illustrated. The generalized LBP definition uses P sample points evenly distributed on a radius R around a center pixel located at (x_c, y_c) . The position (x_p, y_p) , of the neighboring points, where $p \in \{0, \dots, P-1\}$ is given by

$$(x_p, y_p) = \left(x_c + R \cos\left(\frac{2\pi p}{P}\right), y_c - R \sin\left(\frac{2\pi p}{P}\right) \right). \quad (1)$$

The local binary code for the position (x_c, y_c) can be computed by comparing the gray-scale value g_c of this center pixel located at (x_c, y_c) and the gray-scale values g_p of its neighbor pixels located at (x_p, y_p) where $p \in \{0, \dots, P-1\}$. The value of the LBP code of the center pixel at position (x_c, y_c) is given by

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad (2)$$

where s is the Heaviside function:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The operator $\text{LBP}_{P,R}$ produces 2^P different output values, corresponding to 2^P different binary patterns formed by the P pixels in the neighborhood. Although this method can capture the relations of nearby and adjacent pixels, it leads to a large data dimension.

Ojala et al. [7] further propose an “uniform patterns” to reduce the dimension of LBP feature while keeping its

discrimination power. For this, a uniformity measure of a pattern is used: U (“pattern”) is the number of bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. The U value of an LBP pattern can be computed by

$$U(\text{LBP}_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \quad (4)$$

Uniform LBP patterns refer to the patterns which have limited transitions or discontinuities ($U \leq 2$) in the circular binary representation. For instance, 11111111 (0 transitions) and 01110000 (2 transitions) are both uniform whereas 11001001 (4 transitions) and 01010010 (6 transitions) are not. Thus, for P neighborhood pixels, a uniform $\text{LBP}_{P,R}$ operator produces $P(P-1) + 3$ possible distinct uniform LBP patterns. After the uniform LBP patterns are identified, for an image with size $N \times M$, a histogram is built which can be used as the image feature to represent the image texture:

$$h(l) = \sum_{i=1}^N \sum_{j=1}^M f(\text{LBP}_{P,R}(i, j), l), \quad l \in [0, L], \quad (5)$$

$$f(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise,} \end{cases}$$

where L is the maximal LBP pattern value. The length of the histogram is a $P(P-1) + 3$.

2.2. CLBP (Complete Local Binary Pattern). LBP feature considers only signs of local differences (i.e., difference of each pixel with its neighbors) whereas CLBP feature [10] considers both magnitude (M) and sign (S) of local differences as well as original center gray level value (C). Consequently, three operators, namely, CLBP_M, CLBP_S, and CLBP_C, are used to code the magnitude, sign, and center gray level.

Given the gray-scale value g_c of the center pixel (x_c, y_c) and its P circularly and evenly spaced neighbors with gray-scale value g_p , $p \in \{0, \dots, P-1\}$, the difference between g_c and g_p can be simply calculated using $d_p = g_p - g_c$. The local difference vector $[d_0, d_1, \dots, d_{P-1}]$ characterizes the image local structure at (x_c, y_c) . Because the central gray level g_c is removed in local difference vector, $[d_0, d_1, \dots, d_{P-1}]$ is robust to illumination changes and is more efficient in pattern matching. d_p can be further decomposed into two components:

$$\begin{aligned} d_p &= s_p * m_p, \\ m_p &= |d_p|, \\ s_p &= \begin{cases} 1, & d_p \geq 0 \\ -1, & d_p < 0, \end{cases} \end{aligned} \quad (6)$$

where s_p is the sign component of d_p and m_p is the magnitude component of d_p .

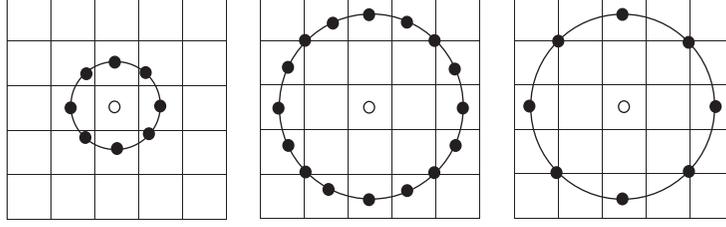


FIGURE 3: Examples of (P, R) neighborhood used to compute LBP: $(8, 1)$, $(16, 2)$, and $(8, 2)$.

CLBP_M is used to code the magnitude information of local differences:

$$\text{CLBP_M}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} t(m_p, c) 2^p, \quad (7)$$

$$t(x, T) = \begin{cases} 1, & x \geq T \\ 0, & x < T, \end{cases}$$

where T is a threshold which is set to the mean value of the m_p values from the whole image.

CLBP_S is the same as the original LBP and is used to code the sign information of local differences:

$$\text{CLBP_S}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} t(s_p, 0) 2^p, \quad (8)$$

$$t(x, T) = \begin{cases} 1, & x \geq T \\ 0, & x < T. \end{cases}$$

CLBP_C is used to code the information of original center gray level value:

$$\text{CLBP_C}_{P,R}(x_c, y_c) = t(g_c, c_l), \quad (9)$$

$$t(x, T) = \begin{cases} 1, & x \geq T \\ 0, & x < T, \end{cases}$$

where the threshold c_l is set to the average gray level of the input image.

The dimension of the histograms corresponding to CLBP_S and CLBP_M is 2^P , while the dimension of CLBP_C is 2. The CLBP_C only uses the center gray level value which can be easily affected by the changing of viewpoints or illumination. Therefore, in our work, only the histograms of CLBP_S and CLBP_M codes are computed and then concatenated together to construct CLBP feature. Thus, the final dimension of CLBP feature is 2^{P+1} .

2.3. CSLBP (Center-Symmetric Local Binary Patterns). CSLBP [11] is another modified version of LBP. CSLBP produces shorter feature set than LBP, but it is also a first-order local pattern in center-symmetric direction and it ignores the central pixel information. CSLBP is closely related to

the gradient operator, because it compares the gray levels of pairs of pixels in centered symmetric directions instead of comparing the central pixel to its neighbors. In this way, CSLBP feature takes advantage of the properties of both LBP and gradient based features.

For an even number P of neighboring pixels distributed on radius R , CSLBP operator produces $2^{P/2}$ possible distinct patterns. The operator is given by

$$\text{CSLBP}_{P,R}(x_c, y_c) = \sum_{i=0}^{(P/2)-1} s(|g_i - g_{i+(P/2)}|) 2^i, \quad (10)$$

$$s(x) = \begin{cases} 1, & x \geq T \\ 0, & \text{otherwise,} \end{cases}$$

where g_i and $g_{i+(P/2)}$ are the gray values of center-symmetric pairs of pixels. T is used to threshold the gray level difference so as to increase the robustness of CSLBP feature on flat image regions. Since the gray levels are normalized in $[0, 1]$, the authors of paper [11] recommend to use small value for T .

It should be noticed that CSLBP is closely related to gradient operator, because like some gradient operators it considers gray level difference between opposite pixels in a neighborhood.

Given an image of size $N \times M$, after the computation of CSLBP patterns, a histogram is built to represent the texture image:

$$h(l) = \sum_{i=1}^N \sum_{j=1}^M f(\text{CSLBP}_{P,R}(i, j), l), \quad (11)$$

$$l = 0, 1, 2, 3, \dots, 2^{P/2} - 1,$$

$$f(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise.} \end{cases}$$

By construction, the length of the histogram resulting from CSLBP feature is $2^{P/2}$.

2.4. CSLDP (Center-Symmetric Local Derivative Pattern). CSLDP operator [12] is a second-order derivative pattern in center-symmetric direction. CSLDP captures more information by encoding the relationship between central pixel and center-symmetric neighbors. Moreover, CSLDP has shorter length than LBP.

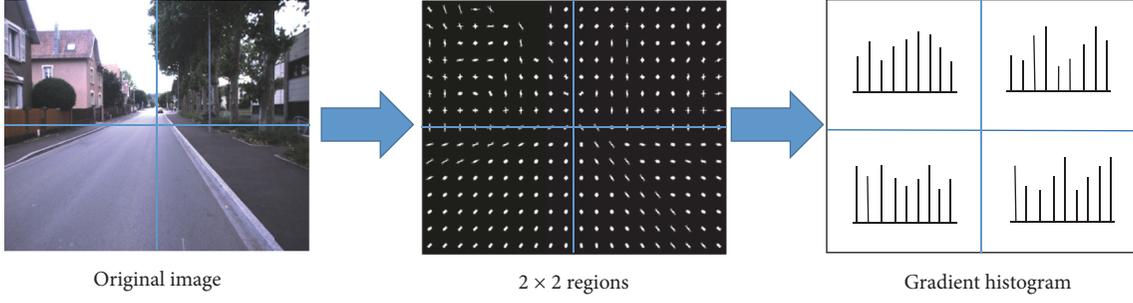


FIGURE 4: Example of HOG feature.

For an even number P of neighboring pixels distributed on radius R , CSLDP operator produces $2^{P/2}$ possible distinct patterns and is defined as

$$\text{CSLDP}_{P,R}(x_c, y_c) = \sum_{i=0}^{P/2-1} t[(g_i - g_c), (g_c - g_{i+(P/2)})] 2^i, \quad (12)$$

where g_i , $g_{i+(P/2)}$ are gray-scale values of neighborhood pixels in center-symmetric direction. g_c corresponds to the gray value of central pixel located at (x_c, y_c) . The threshold function $t(\cdot, \cdot)$ is used to determine the type of local pattern transition and is defined as

$$t(x_1, x_2) = \begin{cases} 1, & x_1 \cdot x_2 \leq 0 \\ 0, & x_1 \cdot x_2 > 0. \end{cases} \quad (13)$$

A CSLDP pattern encodes the second-order center-symmetric derivatives at pixel (x_c, y_c) along 0° , 45° , 90° , and 135° directions. They can be represented as

$$\begin{aligned} \text{CSLDP}_{0^\circ}(x_c, y_c) &= t[(g_0 - g_c), (g_c - g_4)], \\ \text{CSLDP}_{45^\circ}(x_c, y_c) &= t[(g_1 - g_c), (g_c - g_5)], \\ \text{CSLDP}_{90^\circ}(x_c, y_c) &= t[(g_2 - g_c), (g_c - g_6)], \\ \text{CSLDP}_{135^\circ}(x_c, y_c) &= t[(g_3 - g_c), (g_c - g_7)]. \end{aligned} \quad (14)$$

The CSLDP histogram construction method is the same as for CSLBP, and its histogram length is also $2^{P/2}$.

2.5. XCSLBP (Extended CSLBP). The work in [13] proposes a new LBP variant called XCSLBP (eXtended CSLBP), which compares the gray values of pairs of center-symmetric pixels considering the central pixel, without increasing histogram length. This combination makes the resulting descriptor robust to illumination changes and noise. For an even number P of neighboring pixels distributed on radius R , XCSLBP is expressed as

$$\text{XCSLBP}_{P,R}(x_c, y_c) = \sum_{i=0}^{(P/2)-1} s(g_c^2 + g_{i+(P/2)}(g_i - 2g_c)) 2^i, \quad (15)$$

where the threshold function s , which is used to determine the types of local pattern transition, is defined as

$$s(x) = \begin{cases} 1, & (x \geq 0) \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where g_i and $g_{i+(P/2)}$ are the gray values of center-symmetric pixels. XCSLBP operator produces histograms with a length of $2^{P/2}$.

2.6. HOG (Histograms of Oriented Gradients). Besides LBP and its variants, another histogram feature named HOG has also been widely accepted as one of the best features to capture the edge or local shape information. HOG feature is proposed by Dalal and Triggs [16] and widely used to detect objects in computer vision. The essential idea of HOG feature is that the shape or appearance of local object can be described by the distribution of intensity gradients and edge directions [17]. HOG descriptor is a one-dimensional histogram of gradient orientations of intensity in local regions that can represent object shape.

3. Overview of Proposed Approach

In this section, a robust visual localization based on multifeature combination is developed. The general principle is to find the image that best matches the current acquired one, among a set of previously acquired and GPS-tagged training images.

As shown in Figure 4, HOG divides the image into small connected blocks, and, for each block, a histogram of gradient directions for the pixels within the block is computed. The combination of these cell histograms represents the feature vector. At each pixel, the gradient is a 2D vector with a real-valued magnitude and a discretized direction (9 possible directions uniformly distributed in $[0, \pi]$). During the construction of the integral image of HOG, the feature value at each pixel is treated as a 9D vector, and the value at each dimension is the interpolated magnitude value at the corresponding direction. Since HOG takes adjacent pixel gradients information as basis to extract features, it is robust to changes in geometry and is not easily affected by local lighting conditions.

The whole system includes an offline phase and an online phase. In the offline phase, a set of GPS-tagged training image

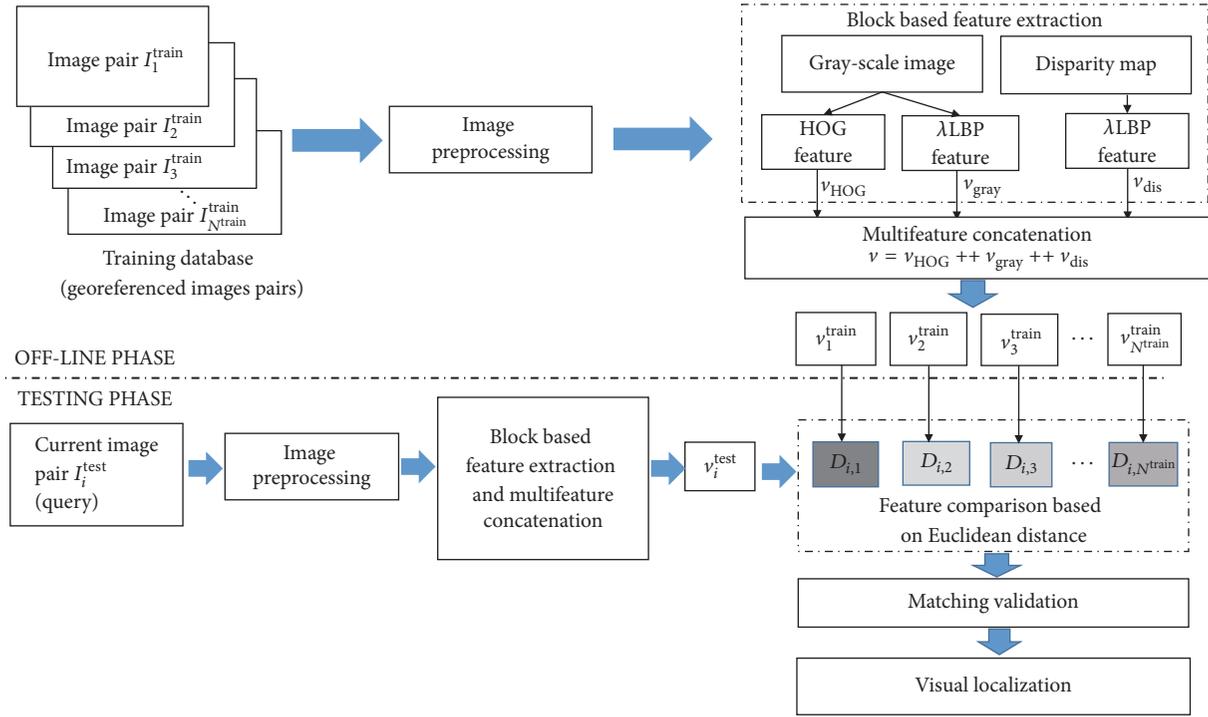


FIGURE 5: The process of the proposed place recognition based visual localization.

pairs (left and right images) $I^{\text{train}} = \{I_j^{\text{train}}\}_{j=1}^{N^{\text{train}}}$ are firstly acquired, where N^{train} is the number of training image pairs. After image preprocessing (see Section 3.1), multifeature set $V^{\text{train}} = \{v_j^{\text{train}}\}_{j=1}^{N^{\text{train}}}$ is extracted from the training database (see Sections 3.2 and 3.3), where v_j^{train} is the multifeature extracted from the training image pair I_j^{train} . In online phase, multifeature v_i^{test} is extracted from current image pair I_i^{test} and then compared with each multifeature of V^{train} based on Euclidean distance. The computed distances are then used to select the best candidate (see Section 3.4); the smaller the distance is, the higher the similarity between the images will be. A distance ratio SS between the two best candidates (i.e., corresponding to the two minimum computed distances) is considered for matching validation (see Section 3.5). If the ratio SS is lower than or equal to a threshold Th , the first best image candidate (with the lower matching distance) is confirmed as positive; otherwise, it is regarded as negative (in this case, no matching result is conserved). When a matching is confirmed as positive, the current position can be obtained from the matched GPS-tagged training image (see Section 3.6).

As illustrated in Figure 5, the overall approach comprises six stages:

- (1) *Image preprocessing*: this step consist of downsampling and contrast-limited adaptive histogram equalization (detailed in Section 3.1)
- (2) *Block based feature extraction*: λ LBP feature is extracted from gray-scale image and disparity map;

HOG feature is extracted from gray-scale image (detailed in Section 3.2)

- (3) *Multifeature concatenation*: the final multifeature D- λ LBP++HOG is obtained by concatenating λ LBP and HOG feature (detailed in Section 3.3)
- (4) *Feature comparison and image matching*: based on the extracted multifeature descriptors, image matching is conducted through multifeature comparison using Euclidean distance (detailed in Section 3.4)
- (5) *Final Matching validation*: according to the distance ratio of the top two best candidates, image matching result is validated (detailed in Section 3.5)
- (6) *Visual localization*: the vehicle current position can be obtained through the matched GPS-tagged training image (detailed in Section 3.6)

3.1. Image Preprocessing. Image preprocessing is composed of two parts: downsampling and contrast-limited adaptive histogram equalization (CLAHE).

Downsampling permits reducing the original image size, which makes feature extraction faster. In fact, it has been already proved in [18] that high resolution images are not more helpful than lower resolution ones. Therefore, downsampling is the first step before feature extraction. As it is well known, illumination has significant influence on outdoor image appearance. Therefore, another applied image preprocessing is contrast-limited adaptive histogram equalization (CLAHE), which permits enhancing the contrast of the

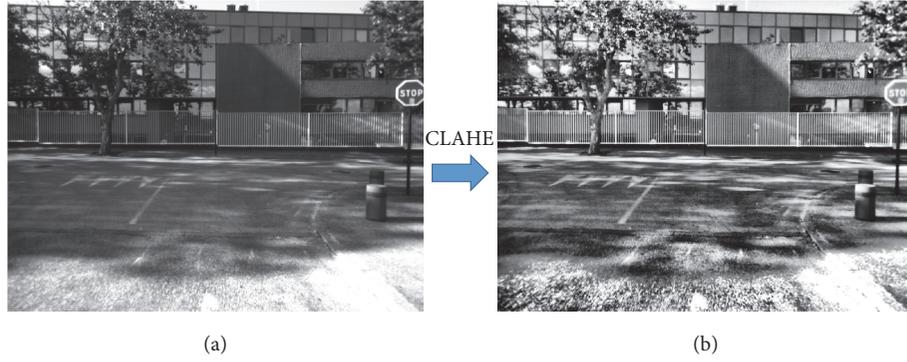


FIGURE 6: Image preprocessing using contrast-limited adaptive histogram equalization (CLAHE). (a) is processed using CLAHE and the preprocessing result is (b).

gray-scale image by transforming the values using contrast-limited adaptive histogram equalization [19]. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain higher contrast. This contrast, especially in homogeneous areas, can be limited to avoid amplifying any noise that might be present in the image. On the same time, it also decreases the shadow influence. An image example after applying contrast-limited adaptive histogram equalization can be seen in Figure 6. It is obvious that CLAHE preprocessing improves the image contrast and makes the image more brightened (especially in some dark parts).

3.2. Block Based Feature Extraction

3.2.1. Concept of Block Based Approach. Traditionally, local descriptors are calculated on full images, which can keep the size of the feature database reasonably low. However, local image areas of interest would be ignored as the full image feature extraction does not contain enough local discriminative information.

With respect to local properties and enhanced image representation ability, image features are extracted from small image blocks (subimage areas) without any segmentation and then these independent feature descriptors are concatenated to obtain final image feature. To illustrate the block based feature extraction process, it is applied on an example in Figure 7. Block based approach (that relies on image blocks) can address spatial properties of images. It can be used for any histogram descriptors.

3.2.2. Block Based Feature Extraction. After image preprocessing, features are extracted, as illustrated in Figure 8. λ LBP feature is extracted from gray-scale image and disparity map independently, while HOG feature v_{HOG} is extracted from gray-scale image. For both λ LBP or HOG, the features are extracted based on image blocks. In order to facilitate the process of block based feature extraction, image blocks in the full image have the same size. The influence of different block sizes will be studied in Section 5.1. Image parts that cannot satisfy a whole block will be ignored.

(i) λ LBP Feature Extraction. λ LBP feature from gray-scale image and disparity map are obtained using the following equations:

$$\begin{aligned} v_{\text{gray}} &= h_1^{\text{gray}} + h_2^{\text{gray}} + \dots + h_m^{\text{gray}} \\ v_{\text{dis}} &= h_1^{\text{dis}} + h_2^{\text{dis}} + \dots + h_n^{\text{dis}}, \end{aligned} \quad (17)$$

where v_{gray} is a vector which stores the λ LBP feature obtained from gray-scale image. v_{dis} stores the λ LBP obtained from disparity map. m and n are the image block numbers of gray-scale image and disparity map, respectively. h_i^{gray} ($i \in [1, 2, \dots, m]$) is the λ LBP histogram of the i th block of the gray-scale image and h_i^{dis} ($i \in [1, 2, \dots, n]$) is the λ LBP histogram of the i th block of the disparity map. In our work, the disparity map is calculated using the SGBM (Semiglobal Block Matching) algorithm [20]. Using this SGBM method, there are some useless parts (“black areas”), for which no depth information is computed, especially on the left and right sides of the disparity map. In these “black areas,” λ LBP operator is not applied; therefore, these useless parts are simply removed. Thus, due to the removing of the “black areas” in the disparity map, m and n are not identical.

By using the block based approach, the features v_{gray} and v_{dis} are extracted from gray-scale image and disparity map, respectively. Then, D- λ LBP feature can be computed by concatenating v_{gray} and v_{dis} :

$$v_{\text{D-}\lambda\text{LBP}} = v_{\text{gray}} + v_{\text{dis}}. \quad (18)$$

(ii) HOG Feature Extraction. HOG feature is also computed for each image block of the gray-scale image. The obtained HOG features from all the image blocks are then concatenated:

$$v_{\text{HOG}} = h_1^{\text{hog}} + h_2^{\text{hog}} + \dots + h_m^{\text{hog}}. \quad (19)$$

Here, h_i^{hog} ($i \in [1, 2, \dots, m]$) is the HOG feature extracted from the i th image block. It should be noted that HOG feature adopts the same image block size as the λ LBP feature extraction from gray-scale image; therefore, the number of image blocks is the same.

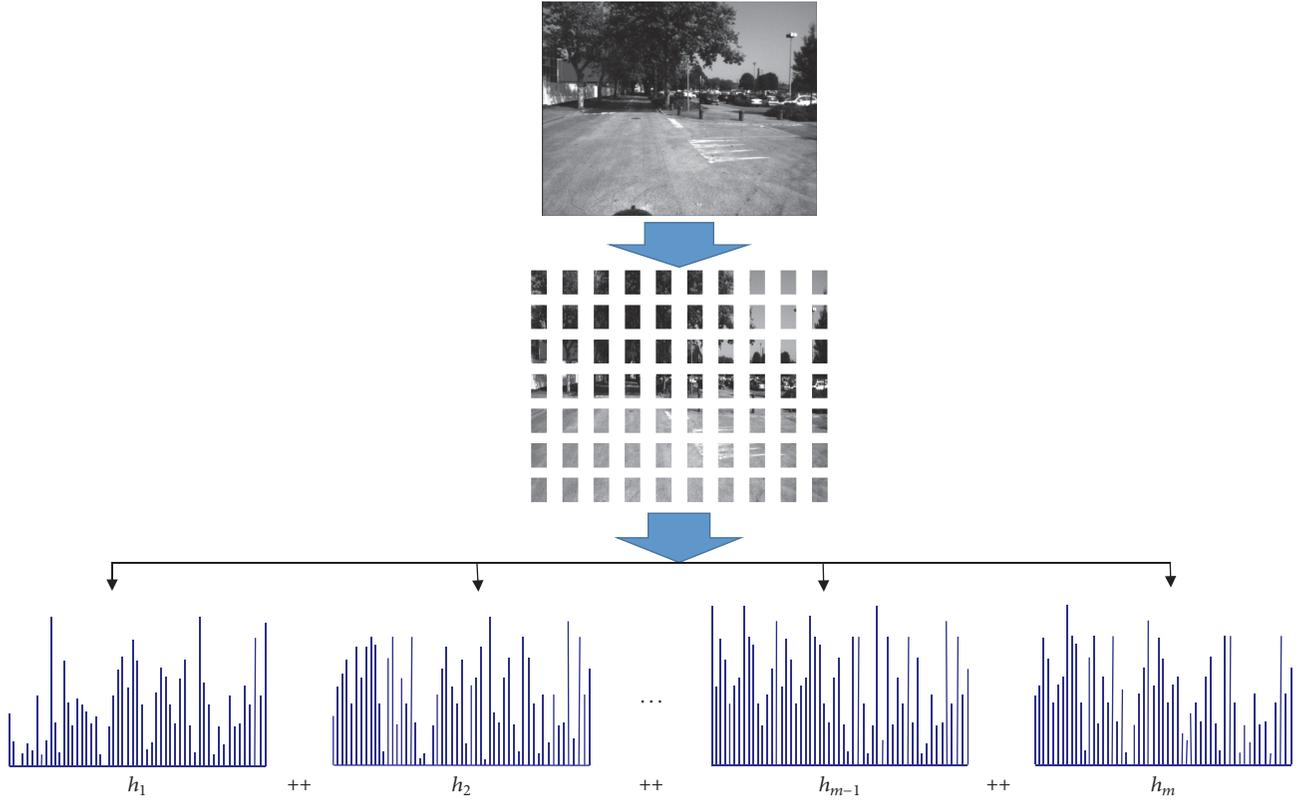


FIGURE 7: An example of block based local binary descriptor extraction. Features are extracted from each image block firstly and then concatenated together. Here, image blocks are nonoverlapped and do not need any image segmentation.

3.3. Multifeature Concatenation. In order to take advantage of the different features, D- λ LBP and HOG are combined together to represent the image. Since the D- λ LBP and HOG are two independent features, we simply consider that they have the same weight in the role of place recognition. The final multifeature can be obtained easily through concatenation using the following equation:

$$v = v_{D-\lambda LBP} + v_{HOG}. \quad (20)$$

Using this method, a multifeature set $V^{\text{train}} = \{v_j^{\text{train}}\}_{j=1}^{N^{\text{train}}}$ of all training image pairs $\{I_j^{\text{train}}\}_{j=1}^{N^{\text{train}}}$ is obtained. For a current testing image pair I_i^{test} , a multifeature v_i^{test} is also obtained. Then the image matching is conducted based on the Euclidean distance comparison between the multifeature v_i^{test} of the current testing image and all training image multifeatures v_j^{train} ($j = [1, 2, \dots, N^{\text{train}}]$) from the training images dataset.

3.4. Feature Comparison and Image Matching. Feature comparison is performed based on the Euclidean distance between features. Each testing image multifeature v_i^{test} is compared with all the training images multifeatures v_j^{train} ($j = [1, 2, \dots, N^{\text{train}}]$) of the training database.

The distance $D_{i,j}$ between the multifeature v_i^{test} ($i = [1, 2, \dots, N^{\text{test}}]$) of the testing image and multifeature vector

v_j^{train} ($j = [1, 2, \dots, N^{\text{train}}]$) of a training image is computed as follows:

$$D_{i,j} = \|v_i^{\text{test}} - v_j^{\text{train}}\|^2, \quad (21)$$

where $\|\cdot\|$ denotes the Euclidean norm.

In fact, small distance means high similarity. Based on Euclidean distance, image matching candidates are searched. After distance computation, for the testing image, the two minimum distances ($D_{i,m1}$ and $D_{i,m2}$) and their corresponding training images (the two best candidates) are conserved.

3.5. Final Matching Validation. For a given current image pair I_i^{test} , the validation of matching candidate from the training database is based on the ratio SS_i , calculated as follows:

$$SS_i = \frac{D_{i,m1}}{D_{i,m2}}, \quad (22)$$

where $D_{i,m1}$ and $D_{i,m2}$ are, respectively, the first and second minimum distances between the current image multifeature v_i^{test} and the multifeatures $\{v_j^{\text{train}}\}_{j=1}^{N^{\text{train}}}$ of all the training images:

$$\begin{aligned} D_{i,m1} &= \min_j \{D_{i,j}\}, \\ D_{i,m2} &= \min_{j(j \neq m1)} \{D_{i,j}\}. \end{aligned} \quad (23)$$

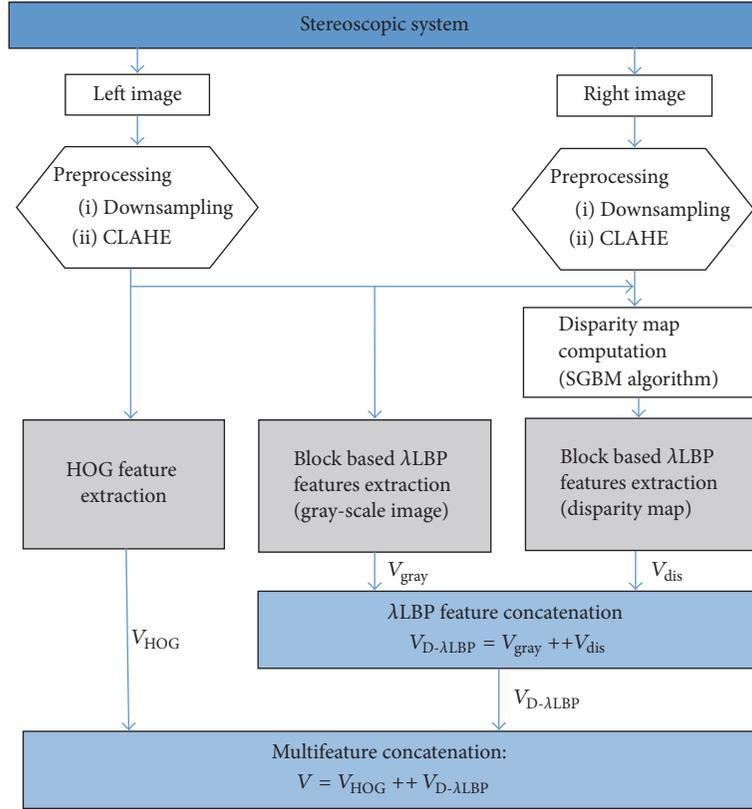


FIGURE 8: Block based feature extraction procedure (applied to each images pair of the training database for the offline phase and to current images pair for the testing phase).

As said before, the lower the distance is, the more similar the images are. The potential matching candidate is the image m_i (the one giving the lower distance with the testing image). However, if the second best matching candidate provides a distance very close to the first one, this means that the matching algorithm provides two confused solutions. In this case, we propose to ignore the matching result and consider that the testing image has no matching image. For that, a threshold Th is applied to the ratio SS_i , which takes its values in the range $[0 \ 1]$.

The last decision is as follows: if SS_i is lower than or equal to the threshold Th , then the pair $(i, m1)$ is considered as positive, and the pair is matched. Otherwise, the pair is considered as negative and the pair is ignored.

3.6. Visual Localization. After image matching result is successfully validated, the vehicle can localize itself through the matched training image position. Since the training images are tagged with the GPS or pose information, the vehicle can get its position information by assimilating its position to the GPS position of the training image matched with the current testing image. This is a topological level localization; that is, the system simply identifies the most likely location. Therefore, this is not a very accurate metric localization, because the training and testing trajectories are not exactly the same.

It should be noted that some places can not be localized at the situation of validation failure (negative matching case).

4. Experimental Setup

4.1. Datasets and Ground-Truth. The proposed method is tested on four different datasets (UTBM-1, UTBM-2, KITTI 05, and KITTI 06).

The taken route for UTBM-1 dataset is shown in Figure 9(a): the experimental vehicle traversed about 4 km in a typical outdoor environment. Three typical areas were traversed: urban city road (area a), lots of factories building (area b), and a nature scene surrounding a lake (area c). The training and testing data were collected at different times, respectively, in 2014/9/11 and 2014/9/5. The training database is composed of 849 images while the testing database is composed of 819 images. The average distance between two successive frames was around 3.5 m. To tag the training images, GPS position of each image is obtained by a RTK-GPS receiver.

The UTBM-2 dataset (Figure 9(b)) consists of a 2.3 km route in Belfort city downtown acquired in 2014/9/5. The first traversal to acquire training dataset was performed in the morning and the second one was conducted in the afternoon to acquire testing dataset. Each travel time across this dataset was approximately 20 minutes. The training database is

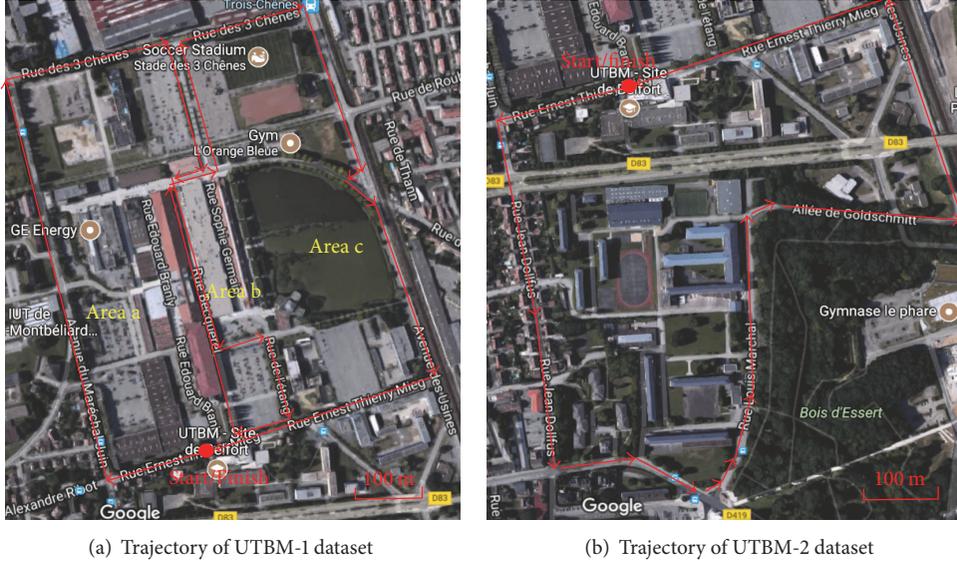


FIGURE 9: Vehicle paths for the UTBM-1 and UTBM-2 datasets. Source: Google Maps.

composed of 540 images while the testing database is composed of 520 images. The GPS information of each image is also collected.

The popular KITTI benchmark dataset is also used to test our proposal. The KITTI Odometry dataset has 22 sequences containing a total of 44182 stereo images (39.2 km). These sequences include environments with different characteristics and challenging situations such as perceptual aliasing and changes on scene. Among them, the datasets KITTI 05 and KITTI 06 that contain loop closures were selected to evaluate our method. There are 2761 and 1101 images in KITTI 05 and KITTI 06 datasets, respectively.

For UTBM-1 and UTBM-2 datasets, ground-truth was constructed by manually finding pairs of frame correspondences according to the GPS data, while the KITTI dataset ground-truth was built according to the pose information [21].

4.2. Image Preprocessing and Feature Extraction. In our work, for faster feature extraction, the original color images were downsampled into half scale size gray-scale image. That means images in datasets UTBM-1 and UTBM-2 were resized to 640×480 and the images in dataset KITTI 05 and KITTI 06 were resized to 613×235 .

In order to reduce the illumination influence on the outdoor image appearance, contrast-limited adaptive histogram equalization (CLAHE) method was used (see Section 3.1).

Moreover, as a pair of images is acquired at each instant, a disparity map can be computed easily using the SGBM (Semi-global Block Matching) algorithm [20].

After image preprocessing, binary descriptors (LBP, CLBP, CSLBP, CSLDP, and XCSLBP) are extracted with the following parameters: 8 sampling points and 3 pixels radius. HOG descriptor is extracted from the gray-scale images. To capture large-scale spatial information, the cell size of HOG

is 32×32 . The number of cells in each block is specified as a 2-element vector.

An example of extracted image features can be seen in Figure 10. It can be seen that the local binary features pay more attention to texture information. It can also be noted that CSLBP and XCSLBP perform better than LBP. HOG feature depicts object shape information in the image. Therefore, combining LBP and HOG features could bring more information and make place (scene) better described.

4.3. Performance Evaluation. Precision-recall characteristics and F_1 score are widely used to show the effectiveness of image retrieval method. Therefore, our evaluation methodology is based on precision-recall curves and F_1 score. In our experiments, the training image number is larger than or equal to the testing image number; thus each testing image has a ground-truth matching. Therefore, among the positives, there are true positives (correct results among successfully validated images matching candidates) and false positives (wrong results among successfully validated images matching candidates). The sum of the true positives and false positives is the total retrieved images number.

More specifically, precision is the ratio of true positives over the retrieved images number (number of all the successfully validated image matching candidates), and recall is the ratio of true positives over the total testing images:

$$\text{Precision} = \frac{\text{Number of true positives}}{\text{Number of retrieved images}} \times 100\%,$$

$$\text{Recall} = \frac{\text{Number of true positives}}{\text{Number of total testing images}} \times 100\%. \quad (24)$$

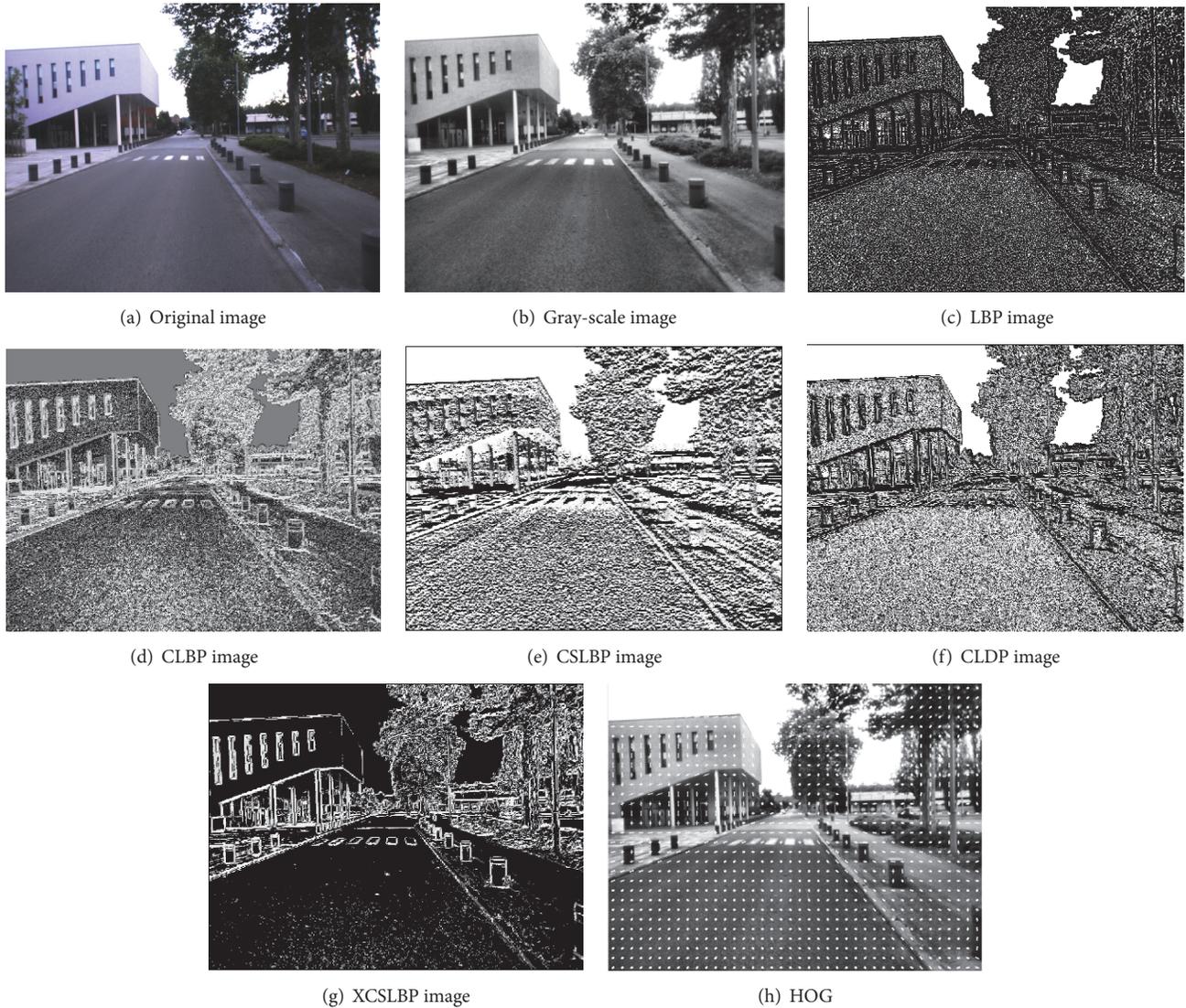


FIGURE 10: Example of gray-scale image and its corresponding local binary images (LBP, CLBP, CSLBP, CSLDP, and XCSLBP) and HOG feature.

The final curve is computed by varying the threshold Th (applied to the ratio SS) in a linear distribution between 0 and 1, with the calculation of the corresponding values of precision and recall. 100 values of threshold Th are considered to obtain well-defined curves. When the threshold is set to 1, the candidates whose ratio is below or equal to 1 are positives. In this case, the number of retrieved images is identical to the number of testing images, while when the threshold is 0, it means that the candidates whose ratio is below or equal to 0 are regarded as positives. In this case, there is no retrieved image.

Precision relates the number of correct matches to the number of false matches, whereas recall relates the number of correct matches to the number of missed matches. A perfect system would return a result where both precision and recall have a value of one. The F_1 score value is a single value that

indicates the overall effectiveness of image retrieval method. Based on the precision and recall, F_1 score is defined as

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (25)$$

5. Experiments and Results

Different aspects of our proposal are evaluated in the following sections. In Section 5.1, the performance of binary features (LBP and its variants) with and without disparity information is studied. In addition, the image block size influence for the binary feature D-CSLBP is investigated in Section 5.1. In Section 5.2, the effect of the multifeature fusion proposed in our approach is analyzed. It is to note that the experiment results obtained in Sections 5.1 and 5.2 are

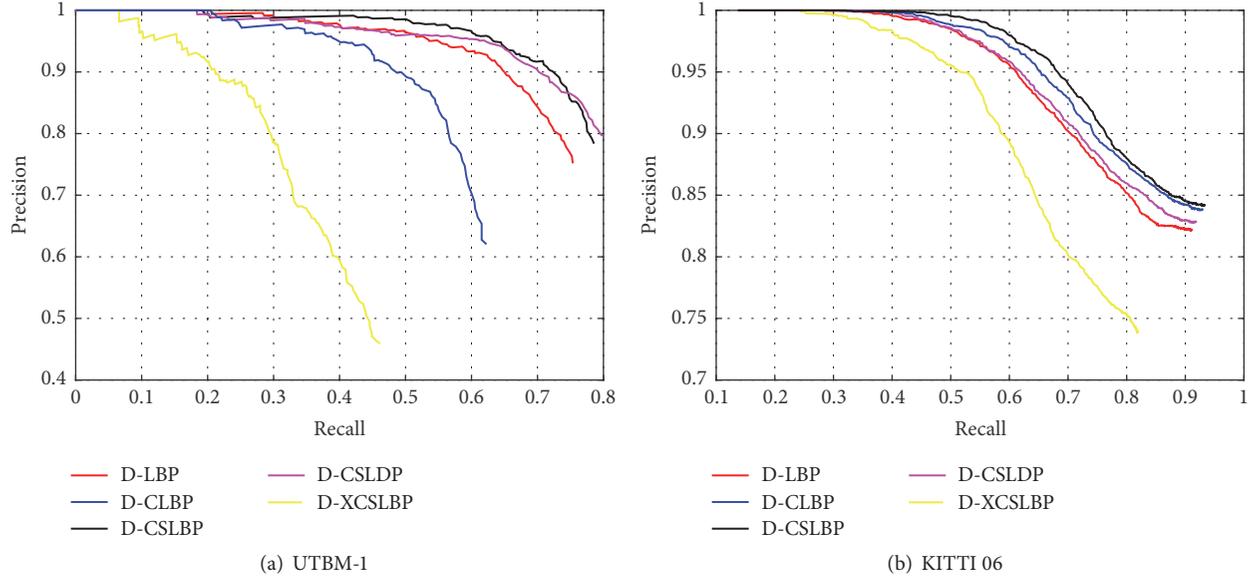


FIGURE 11: Image retrieval performance (precision-recall curve) comparison considering different block based binary features on UTBM-1 and KITTI 06 datasets. Here the image block size is 32.

TABLE 1: F_1 score comparison for different tested binary features on four datasets. Here the block size is set to 32×32 .

Feature	UTBM-1	UTBM-2	KITTI 05	KITTI 06
LBP	0.5171	0.9058	0.7361	0.8261
D-LBP	0.7665	0.9441	0.7663	0.8639
CLBP	0.5111	0.9194	0.7437	0.8279
D-CLBP	0.6672	0.9221	0.7735	0.8813
CSLBP	0.6292	0.9337	0.7569	0.8461
D-CSLBP	0.8043	0.9457	0.7763	0.8850
CSLDP	0.6093	0.9474	0.7536	0.8335
D-CSLDP	0.8062	0.9490	0.7709	0.8709
XCSLBP	0.4796	0.8986	0.7107	0.7814
D-XCSLBP	0.7190	0.8350	0.7401	0.7775

based on Euclidean distance. In Section 5.3, the efficiency of our LSH based visual recognition is checked: the execution time and recognition performance of our complete system are evaluated. Finally, visual localization at 100% recognition level is discussed in Section 5.4.

5.1. Comparison of the Different Binary Features and Image Block Sizes

5.1.1. Performance of Different Binary Features. In this section, we compare binary features performance in two situations: with or without disparity map. Here the features are compared based on the Euclidean distance.

Table 1 gives the F_1 scores of the binary descriptors in two cases (without and with disparity information). It can be seen that LBP, CLBP, CSLBP, and CSLDP with disparity information improve the image retrieval ability as F_1 scores are higher with disparity information than without disparity

information. Among them, D-CSLBP is the best one; it achieves the highest F_1 score.

Figure 11 depicts the precision-recall curves obtained by the different binary features in two typical datasets UTBM-1 and KITTI 06. It can be seen that the performance of D-CSLBP is better than the performance with the features D-LBP, D-CLBP, D-CSLDP, and D-XCSLBP. Also, it can be seen that the maximum recall at 100% precision for D-CSLBP is higher than the one of the other features.

5.1.2. Comparison of Different Image Block Sizes. In this section, the influence of block size of block based D-CSLBP feature is studied.

Small block size permits discriminating local details, while large block size makes the representation more robust. Each image block is a square block in our experiment (block size 32×32 is shorted as 32). The performance of D-CSLBP feature with different block sizes (32, 64, 128, and $32 + 64 + 128$ (multiblock sizes, there different block sizes used together)) in place recognition is evaluated.

According to Figure 12, it can be noted that by increasing the block size from 32 to 64 and 128, the place recognition ability decreases. The computation of D-CSLBP feature with combination of the block sizes 32, 64, and 128 only permits achieving a slightly better performance than the D-CSLBP feature with block size 32.

It is obvious that the binary descriptor D-CSLBP extracted from small block size may benefit from discriminative local details, while feature extraction using larger block size makes it easy for image representation to drop some discriminative information.

However, when the block size is too small, the abundant information can not bring more improvement to the image matching process. At the same time, smaller image block size may lead to computation time increase during feature

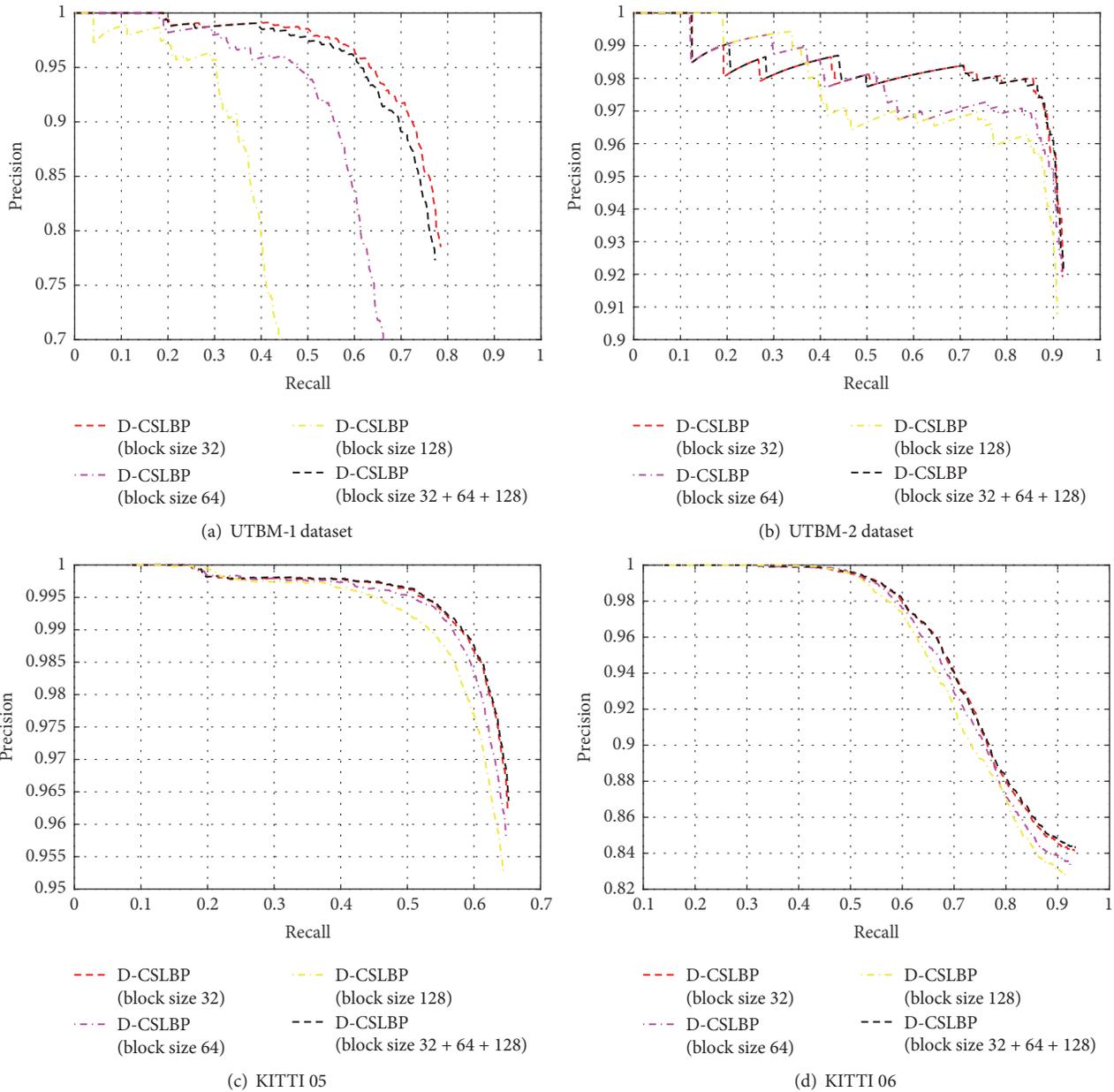


FIGURE 12: Image retrieval performance (precision-recall curve) comparison considering D-CSLBP feature extracted with different image block sizes, on four datasets.

extraction. So, on our following experiments, the image block size for D-CSLBP is set to 32.

5.2. Performance of Multifeature Combination. In this section, we compare the performance of multifeature descriptor (D-CSLBP++HOG) with single independent feature descriptor.

Figure 13 shows the precision-recall curves obtained with the different tested features: D-CSLBP, HOG, and D-CSLBP++HOG. It can be found that the binary feature D-CSLBP combined with HOG permits improving image retrieval performance. Combining D-CSLBP and HOG can

achieve better result than each single feature, which means that the combination is useful for place recognition.

Table 2 compares the F_1 scores of different features with the state-of-the-art FAB-MAP method. It confirms that the multifeature D-CSLBP++HOG achieves better results than single feature. The F_1 score of D-CSLBP++HOG provides the highest value for all the four datasets. Furthermore, the proposed method outperforms the FAB-MAP method.

For a better comprehension of the proposed multifeature, an example of distance matrices for UTBM-1 dataset is presented in Figure 14. Here, for clearly demonstrating the feature performance, the distance matrix D is normalized

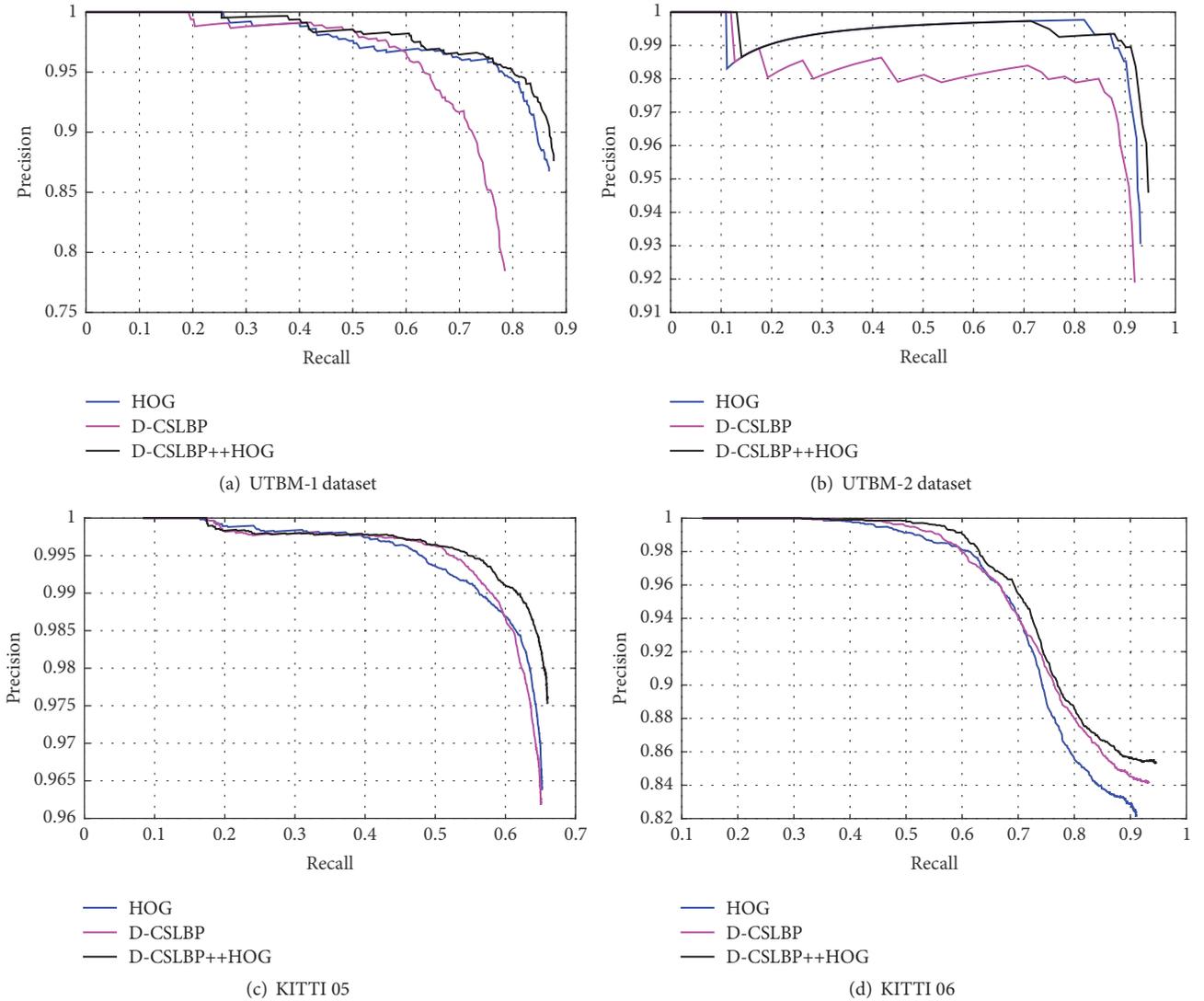


FIGURE 13: Image retrieval performance (precision-recall curve) comparison of HOG, D-CSLBP, and D-CSLBP++HOG based approaches, on four datasets.

into 0-1 range. The distances of the same or similar images are close to 0 (red color), while, for the larger distances, the corresponding color is close to yellow. As plotted in Figure 14(b), the ground-truth line is red. When perceptual aliasing occurs, some red points (noise) will appear which is outside the ground-truth line. In the distance matrix provided by our method using the D-CSLBP++HOG feature (see Figure 14(c)), it can be seen that the noise which appears around the diagonal (ground-truth line) due to perceptual aliasing is clearly reduced with respect to other feature approaches (CSLBP, D-CSLBP, and HOG). All the previous affirmations are supported by the precision-recall curves depicted in Figure 13(a) and results in Table 2.

We can thus conclude that integrating HOG and disparity information permits improving the image matching results. The reason why the D-CSLBP++HOG achieves better performance than the other features is mainly because the feature combination takes the advantage of texture, shape, and depth

TABLE 2: Comparison of F_1 scores for different features and the state-of-the-art FAB-MAP method, on four datasets.

Dataset	F_1 score			
	D-CSLBP	HOG	D-CSLBP++HOG	FAB-MAP
UTBM-1	0.8043	0.8752	0.8869	0.2356
UTBM-2	0.9299	0.9440	0.9532	0.4813
KITTI 05	0.7763	0.7782	0.7873	0.7417
KITTI 06	0.8850	0.8648	0.8973	0.3519

information, which makes image representation more robust than considering each single feature independently.

5.3. LSH Based Visual Recognition. Since the block based feature dimension is huge in our approach, computing the Euclidean distance between high-dimensional feature vectors is an expensive operation. Therefore, in order to speed up

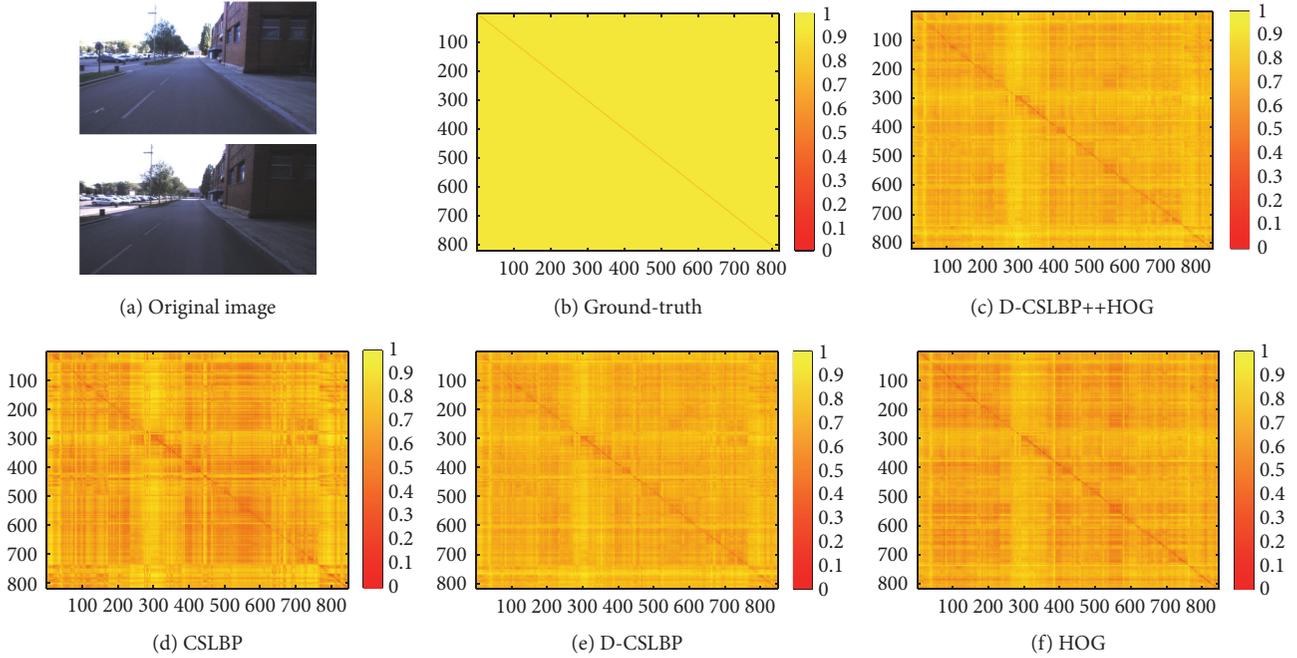


FIGURE 14: Example of distance matrices for UTBM-1 dataset. Here, the distance matrix D is normalized into 0-1 range. The distances of same or similar images are close to 0 (red color), while, for the larger distances, the corresponding color is close to yellow. In (a), two images from the same place are taken at different times (difference of two weeks). From Figure (b) to Figure (f), the distance matrix D is plotted. The distance matrices show that multifeature combination (c) reduces the noise appeared around the diagonal (ground-truth line). Besides, compared with (d), after adding disparity information in (e), perceptual aliasing decreases, as confirmed by the precision-recall curves in Figure 13(a).

image matching significantly, locality sensitive hashing (LSH) method that preserves the Euclidean similarity [22] is used for visual recognition. LSH is arguably the most popular unsupervised hashing method and has been applied to many problems, including information retrieval and computer vision [23]. The paper [23] demonstrates that Euclidean distance between two high-dimensional vectors can be closely approximated by the hamming distance between the respective hashed bit vectors. The more the hash bits that hash method contains, the better the approximation.

The LSH method simply uses a random projection matrix to project the high-dimensional data into a low-dimensional binary (Hamming) space; that is, each data point is mapped to a K -bit vector, called the hash key. Thus approximate nearest neighbors in sublinear time can be found. A key ingredient of locality sensitive hashing is mapping similar features to the same bucket with high probability.

More precisely, for multifeatures V^{test} obtained from testing image and V^{train} obtained from training image, the hashing functions $H(\cdot)$ from LSH family satisfy the following elegant locality preserving property:

$$P\{H(V^{\text{test}}) = H(V^{\text{train}})\} = \text{sim}(V^{\text{test}}, V^{\text{train}}), \quad (26)$$

where the similarity measure sim is directly linked to the Euclidean distance function. Hash keys are constructed by applying K binary-valued hash functions to each image

feature. The K binary-valued LSH functions consists of random projections and thresholds as

$$\begin{aligned} H^{\text{test}}(K) &= \text{sign}(w^T V^{\text{test}} + b), \\ H^{\text{train}}(K) &= \text{sign}(w^T V^{\text{train}} + b), \end{aligned} \quad (27)$$

where w is a K dimensional data-independent random hyperplane, which is usually constructed from a standard Gaussian distribution [24]. b is a random intercept. For a normalized dataset with zero mean, the approximately balanced partition is obtained with $b = 0$.

By applying K binary-valued hash functions to each image feature, high dimension multifeatures V^{test} and V^{train} are converted into a low K dimension bits H^{test} and H^{train} . Since H^{test} and H^{train} are binary bits, they can be more efficiently compared in low dimension space than original feature.

In our experiment, we compare the place recognition performance achieved with hashed multifeature of different binary lengths ($2^8 \dots 2^{12}$ bits) on four datasets in Figure 15. Since the image size is different, multifeature dimension in datasets UTBM-1 and UTBM-2 is 18696 while the multifeature dimension in KITTI 05 and KITTI 06 is 6432. It can be seen that, using 4096 and 2048 bits retains above 86% total place recognition performance.

Table 3 shows the F_1 score obtained from different hash bit lengths applied on the multifeature (D-CSLBP++HOG) of

TABLE 3: F_1 score and matching times comparison of different hash bit lengths for our approach and the state-of-the-art FAB-MAP method.

Method	F_1 score				Average time per matching (all datasets)
	UTBM-1	UTBM-2	KITTI 05	KITTI 06	
256 hash bits	0.6571	0.8989	0.7425	0.8411	0.11×10^{-2} s
512 hash bits	0.8097	0.9409	0.7862	0.8786	0.38×10^{-2} s
1024 hash bits	0.8488	0.9562	0.7794	0.8936	0.88×10^{-2} s
2048 hash bits	0.8771	0.9532	0.7865	0.8973	1.78×10^{-2} s
4096 hash bits	0.8869	0.9537	0.7873	0.9006	3.61×10^{-2} s
Multifeature	0.9258	0.9110	0.9166	0.9203	8.82×10^{-2} s
FAB-MAP	0.2356	0.4813	0.7417	0.3519	2.83×10^{-2} s

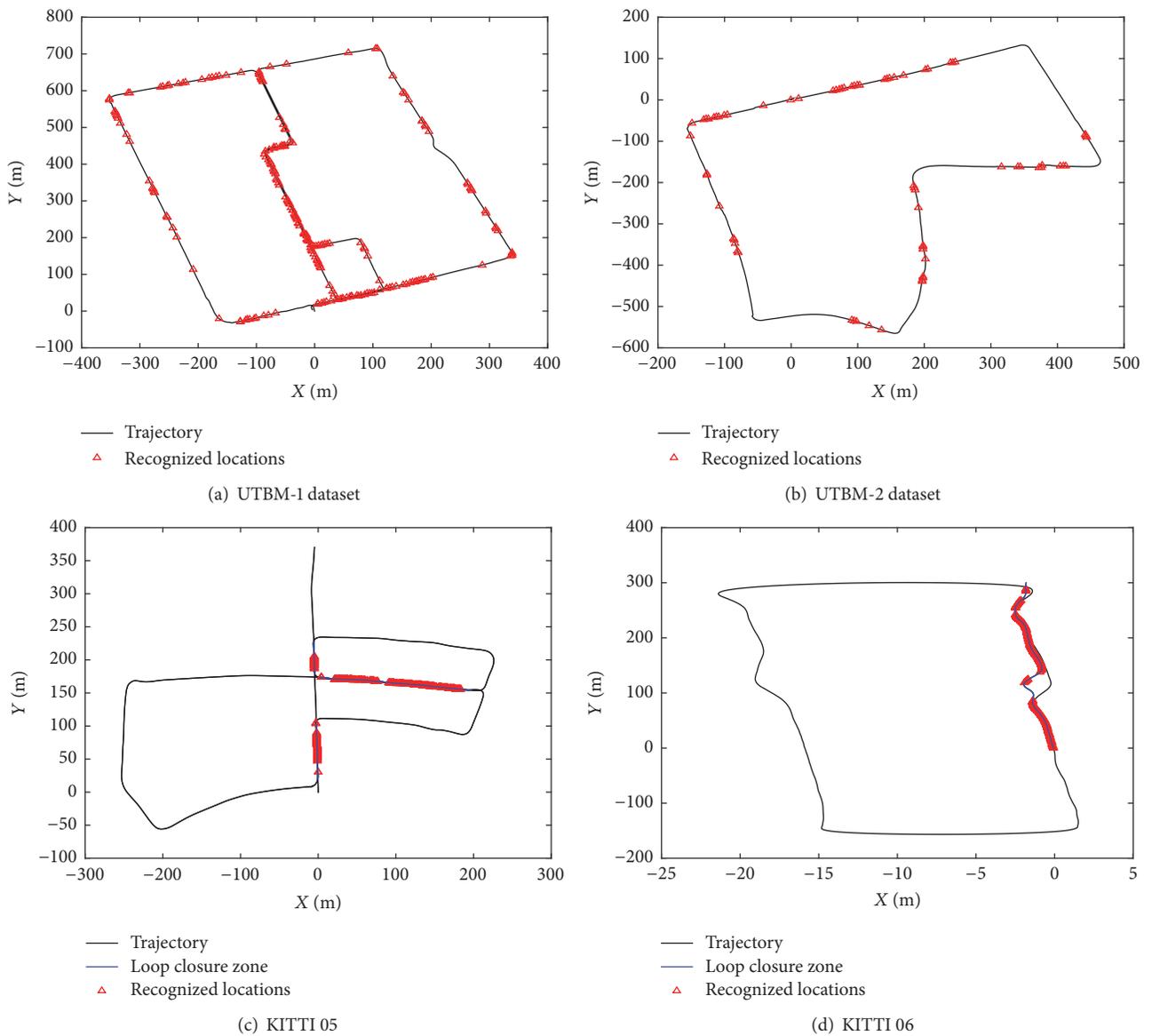


FIGURE 15: Visual localization results obtained by our system on four datasets. The trajectory of the vehicle is depicted with black lines; the loop closure zone is plotted by blue lines. Red points are correctly recognized locations at 100% precision by using our proposed approach. There are no false positives in any case. It is noted that the loop closure zone of datasets UTBM-1 and UTBM-2 is the whole trajectory, while the loop closure zone of KITTI 05 and KITTI 06 is only parts of the trajectory in blue.

TABLE 4: Recall results and average localization error at three precision levels (4096 hash bits).

Dataset	100% precision		99% precision		90% precision	
	Recall (%)	Error (/m)	Recall (%)	Error (/m)	Recall (%)	Error (/m)
UTBM-1	23.81	2.08	89.62	2.34	95.00	3.53
UTBM-2	11.35	2.11	51.89	2.41	89.50	2.49
KITTI 05	17.38	3.50	61.59	13.08	65.92	13.08
KITTI 06	32.39	2.56	59.12	3.18	77.26	4.20

our place recognition method. The average matching time is also presented. Here average matching time does not include the feature extraction time. The experiments were conducted on a laptop machine with intel i7-4700MQ CPU and 32 G RAM.

As Table 3 shows, the average matching time using 4096 bits is almost half of the one using the Euclidean distance over the original full features. Compared with the full multifeature matching, hashing the original multifeature into 4096 bits makes the distance computation and comparison easier and faster. There is no doubt that, for large-scale datasets, the speed-up advantages can be more significant.

5.4. Visual Localization Results. In the previous section, 4096 bits obtained by hashing the original feature shows its good performance in place recognition. Therefore, in this section, we describe visual localization results achieved by 4096 hash bits.

Figure 15 shows the final place recognition results for the different datasets at a precision level of 100%. For the datasets UTBM-1 and UTBM-2, we obtained 23.81% and 11.35% recall at the 100% precision, respectively, while in the KITTI 05 and KITTI 06 datasets, a recall rate of 17.38% and 32.39% is achieved, respectively, at the total correctly level. It should be noted that, at 100% precision level, the obtained place recognition result is totally correct. A correct place recognition means a successful visual localization; therefore, the higher the recognition rate (recall) at 100% precision is, the more robust the visual localization system is.

When adjusting the threshold value T_h , the recognition precision is also changing. At 100% precision level, each recognized place is true positive and its localization error is small (depending on the ground-truth criteria, in our case it is 5 m). For achieving the 100% recognition precision level, threshold value is set to 0.88 and 0.58 for UTBM-1 and UTBM-2 datasets, respectively.

When the threshold is set to 1, which means every image matching result is positive, in this case, the precision level is the lowest and there are many false matching for place recognition, which lead to huge localization error. In general, if small threshold is used, there are few false recognition cases.

In addition, for visual recognition precision level below 100%, meaning that recognized places are not totally correct, some false recognition places appear. For these false recognized places, the localization error can be very large, because the testing image can be wrongly matched to anyone in the training image database. That is also the reason why some locations have huge localization error.

Table 4 gives the average localization error and recall ratio at different precision levels. For all these datasets, at 100% precision, the minimum localization error is 0 while the maximum error is not larger than 5 m. It should be noted that, at 100% precision level, some places can not be recognized and no localization results are obtained at these places. This problem can be easily solved by visual odometry technique or extra sensors (as LiDAR or Inertial Measurement Unit (IMU)).

6. Conclusion and Future Works

In this paper, we presented a visual vehicle localization approach that uses multifeature built from gray-scale image and disparity map. The multifeature concatenates the D-CSLBP and HOG features together to take the advantage of texture, depth, and shape information. Also, block based feature extraction was used to consider the spatial information. Image matching using the proposed multifeature D-CSLBP++HOG based on local sensitive hashing makes the visual recognition more efficient. The results of our experiment demonstrated that this approach provides an available place recognition based visual localization in outdoor environment compared with the state-of-the-art FAB-MAP method.

However, in the long-term visual localization, place recognition is prone to be influenced by appearance or seasonal changing. The future objective of our research is to achieve a robust long-life localization at different times and seasons. Sequence matching will be considered for place recognition in the following research.

Abbreviations

FAB-MAP: Fast Appearance Based Mapping.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

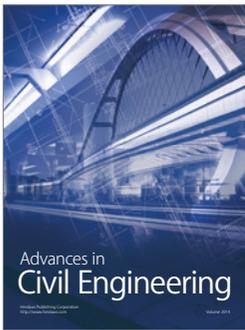
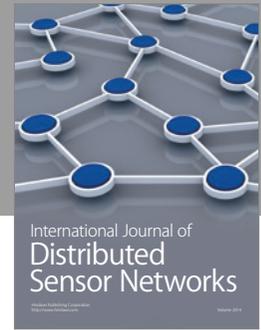
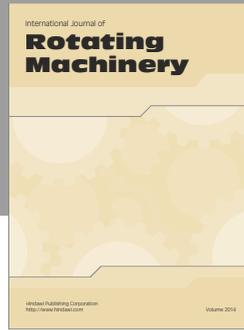
Yongliang Qiao performed experiments, analyzed the data, and wrote the paper; Zhao Zhang participated in paper preparation and revising.

Acknowledgments

Thanks are due to Mr. Yassine Ruichek and Ms. Cindy Cappelle for the writing guidance and help.

References

- [1] S. Lowry, N. Sunderhauf, P. Newman et al., "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [2] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: a survey," *Robotics and Autonomous Systems*, vol. 64, pp. 1–20, 2015.
- [3] M. Cummins and P. Newman, "FAB-MAP: probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [4] Z. Sun, X. Wang, and J. Liu, "Application of Image Retrieval Based on the Improved Local Binary Pattern," in *Proceedings of the 4th International Conference on Computer Engineering and Networks*, pp. 531–538, 2015.
- [5] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1023–1029, 2000.
- [6] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, Article ID 258619, 12 pages, 2015.
- [7] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [8] G. Kylberg and I.-M. Sintorn, "Evaluation of noise robustness for local binary pattern descriptors in texture classification," *EURASIP Journal on Image and Video Processing*, vol. 2013, pp. 1–20, 2013.
- [9] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [10] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.
- [11] M. Heikkilä, M. Pietikinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," in *Computer Vision, Graphics and Image Processing*, pp. 58–69, Springer, 2006.
- [12] G. Xue, L. Song, J. Sun, and M. Wu, "Hybrid center-symmetric local pattern for dynamic background subtraction," in *Proceedings of the 2011 12th IEEE International Conference on Multimedia and Expo, ICME 2011*, July 2011.
- [13] C. Silva, T. Bouwmans, and C. Frélicot, "An eXtended center-symmetric local binary pattern for background modeling and subtraction in videos," in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP*, 2015.
- [14] Y. Yamauchi, C. Matsushima, T. Yamashita, and H. Fujiyoshi, "Relational HOG feature with wild-card for object detection," in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2011*, pp. 1785–1792, November 2011.
- [15] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proceedings of the IEEE 12th International Conference on Computer vision*, pp. 32–39, 2009.
- [16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, 2005.
- [17] H. Ren and Z.-N. Li, "Object detection using edge histogram of oriented gradient," pp. 4057–4061.
- [18] N. Roy, P. Newman, and S. Srinivasa, "Visual route recognition with a handful of bits," in *Robotics: Science and Systems VIII*, p. 504, MIT Press, 2013.
- [19] E. D. Pisano, S. Zong, B. M. Hemminger et al., "Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms," *Journal of Digital Imaging*, vol. 11, no. 4, pp. 193–200, 1998.
- [20] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
- [21] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Bronte, "Fast and effective visual place recognition using binary codes and disparity information," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2014*, pp. 3089–3094, September 2014.
- [22] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pp. 380–388, ACM, 2002.
- [23] D. Ravichandran, P. Pantel, and E. Hovy, "Randomized algorithms and NLP: Using locality sensitive hash function for high speed noun clustering," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, ACL-05*, pp. 622–629, 2005.
- [24] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the 20th Annual Symposium on Computational Geometry*, pp. 253–262, ACM, 2004.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

