

Date of publication xxxx 00, 0000, date of current version August 5, 2020.

Digital Object Identifier xx.xxxx/ACCESS.xxxx.xxx

Robust Alignment of Multi-Exposed Images with Saturated Regions

JUN JIANG^{1,2}, ZHENGGUO LI³, SHOULIE XIE³, SHIQIAN WU⁴ and LIANGCAI ZENG^{1,2}

¹Key Laboratory of Metallurgical Equipment and Control Technology (Wuhan University of Science and Technology), Ministry of Education, Wuhan, 430081, China (e-mail: jiangjun85@wust.edu.cn, zengliangcai@wust.edu.cn)

²Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering (Wuhan University of Science and Technology), Wuhan, 430081, China (e-mail: jiangjun85@wust.edu.cn, zengliangcai@wust.edu.cn)

³Institute for Infocomm Research, Singapore, 138632, (e-mail: ezhli@i2r.a-star.edu.sg, slxie@i2r.a-star.edu.sg)

⁴Institute of Robotics and Intelligent Systems, School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China (e-mail: shiqian.wu@wust.edu.cn)

Corresponding author: Liangcai Zeng (e-mail: zengliangcai@wust.edu.cn).

This work was supported by the National Nature Science Foundation of China under Projects 61775172 and 61371190.

ABSTRACT It is challenging to align multi-exposed images due to large illumination variations, especially in presence of saturated regions. In this paper, a novel image alignment algorithm is proposed to cope with the multi-exposed images with saturated regions. Specifically, the multi-exposed images are first normalized by using intensity mapping functions (IMFs) in consideration of saturated pixels. Then, the normalized images are coded by using the local binary pattern (LBP). Finally, the coded images are aligned by formulating an optimization problem by using a differentiable Hamming distance. Experimental results show that the proposed algorithm outperforms state-of-the-art alignment methods for multi-exposed images in terms of alignment accuracy and robustness to exposure values.

INDEX TERMS Multi-exposed images, Image alignment, Hamming distance, Local binary pattern, Normalization

I. INTRODUCTION

IMAGE alignment is an important process in many image processing applications [1], visual odometry and SLAM [2], and panorama imaging [3]. A robust image alignment algorithm should cope with illumination variations and large motion variations while giving a sub-pixel accurate alignment. Existing image alignment approaches can be classified into two categories [1]: intensity-based methods and feature-based methods. The intensity-based methods find motion parameters directly from image intensities on brightness constancy assumption. In contrast, the feature-based methods, working with image features (such as SIFT-based features), are usually used to deal with illumination-varying images.

Since the brightness constancy assumption is not true for differently exposed images [4], the feature-based methods are widely applied to align differently exposed images. Ordering features are usually used to represent invariant property of illumination-variation images. Median threshold bitmap (MTB) [5] is a global descriptor for the representation of illumination-variation images. But the alignment is not accurate due to errors from unequal division on severely saturated regions and ordering problems for many pixels with

the same intensity [6]. Local descriptors, such as census transform (CT) [7], local binary pattern (LBP) [8], BRIEF [9], ORB [10] and so on, have been widely used to describe illumination-robust features. These descriptors depend solely on comparison between a specific pixel and its neighborhood (or a pair of pixels in its neighborhood), and are therefore invariant under intensity changes.

Several learning-based binary descriptors have been proposed to project a local patch to a binary descriptor. Semantic hashing [11] trains a multi-layer neural network to learn compact binary codes. Spectral hashing (SpeH) [12] generates efficient binary codes by spectral graph partitioning. BinBoost descriptor [13] applies boosting to learn a set of binary hash functions that achieve an accurate binary descriptor. The work in [14] presented a binary online learned descriptor by applying LDA criterion. DeepBit [15] generates a compact binary descriptor by designing a CNN in an unsupervised manner. The work in [16] proposed a compact binary face descriptor (CBFD) to learn evenly-distributive and energy-saving local binary codes. Duan et al. [17] presented a context-aware local binary feature learning (CA-LBFL) approach using the contextual information of adjacent

bits. A supervised structured binary code (SUBIC) with a one-hot block structure was proposed in [18]. Those methods utilize the rigid sign function or a hand-crafted threshold for binarization, which is easily affected by exposure change and leads to unequal binarization [19]. Normally, the learning-based matching pipeline consists of three sub-components [20]: keypoints detection, orientation estimation and features extraction. Each subcomponent is trained individually. However, the overall performance may not be optimal when integrating these separately optimized subcomponents into a matching pipeline [21]. Therefore, end-to-end trainable matching networks [22]–[24] have been proposed to optimize the matching performance. But, they require hand-crafted detectors or other functions to initialize the training process [23], [24]. More importantly, the learning-based methods are very data-dependent and require a large amount of data to train the network repeatedly, which is complex, time-consuming and not suitable for using on mobile devices. In contrast, the traditional binary descriptors is a better and simpler choice to represent the differently exposed images.

For binary descriptors [9], [10], [14], Hamming distance was adopted to measure the similarity of two pixels, and can be computed with a bitwise XOR operation followed by a bit count. However, the Hamming distance is non-differentiable and unsuitable for optimization. Wu et al. [6] used the decimal representation to represent the LBP coded images and matched the difference between two coded images under the squared Euclidean norm. However, the decimal representation does not capture the closeness of two bit-strings and it is sensitive to the rotation. It is thus desired to have a differentiable distance which is equivalent to the Hamming distance for aligning of differently exposed images. Such a distance function was recently introduced in [26]. Besides the issues on the brightness constancy assumption and the distance, one more challenging issue for aligning multi-exposed images is possibly saturated regions in the images [26]. Due to the saturated regions, one feature detected in one image may not occur on another; and a specific intensity in one image may map to multiple intensities in the other image, and vice versa. It is thus desired to develop a new image alignment algorithm for multi-exposed images with saturated regions.

In this paper, a novel alignment algorithm is proposed for multi-exposed images with saturated regions by using the differentiable “Hamming” distance function to binary descriptors in [26]. Specifically, a novel intensity mapping function (IMF) based method is first proposed to normalize the multi-exposed images so that the saturated regions are same when they are aligned. Then the normalized images are coded by the LBP. The difference between two coded images is measured by the differentiable “Hamming” distance in [26] which is equivalent to the Hamming distance on top of the bitwise XOR. Same as [6], the alignment of two differently exposed images is formulated as a quadratic optimization which can be easily solved. Since the differentiable “Hamming” distance captures the closeness of two bit-strings

better than the decimal representation in [6], the proposed algorithm is more robust than the algorithm in [6]. Experimental results demonstrate that the proposed algorithm outperforms state-of-the-art alignment algorithms for multi-exposed images. Two major contributions of this paper are summarized as follows: 1) A new normalization method for differently exposed images is proposed which is effective to reduce the effect of saturation to image alignment; and 2) A simple image alignment algorithm for differently exposed images is presented which is robust to the exposure values (EV).

The rest of this paper is organized as follows. Multi-exposed images are normalized in section II and aligned in section III. Experimental results are provided in section IV to validate the proposed algorithm. Finally, conclusions are provided in section V.

II. NORMALIZATION OF MULTI-EXPOSED IMAGES

In this section, a novel normalization method is first introduced for multi-exposed images. Unlike the method in [28], the exposure times of the images are not required by the proposed method.

Consider two images Z_1 and Z_2 of an identical scene with two different exposures Δt_1 and Δt_2 . $Z_1(p)(Z_2(p))$, $Z_1(p')(Z_2(p'))$ and $E(p)(E(p'))$ are two intensities and radiances respectively at pixel positions p and p' . There are two possible issues for two differently exposed images. One is that their intensity values are different for two co-located pixels. Fortunately, the following relationship is generally held due to the monotonic property:

$$E(p) > E(p') \Rightarrow Z_1(p) > Z_1(p'), \text{ and } Z_2(p) > Z_2(p'). \quad (1)$$

Clearly, exposures change the intensities, but keep the relative order of intensities if they are not saturated. Thus, the LBP code could be adopted to represent two differently exposed images [6].

One possible issue for multi-exposed images is the saturation, i.e., part of an image could be over/under-exposed. The saturation is a very challenging issue for alignment [26]. Assume that Δt_1 is larger than Δt_2 . Let α and β be two predefined constants, and their values are selected as 5 and 254, respectively. For simplicity, a pixel $Z(p)$ is regarded as over-exposed if its value is larger than β , and under-exposed if its value is smaller than α . Here, a novel method is introduced to synchronize the under-exposed regions of the images \hat{Z}_1 and \hat{Z}_2 as well as the over-exposed regions of the images \hat{Z}_1 and \hat{Z}_2 by using the IMFs between the images Z_1 and Z_2 .

Let $f_{12}(z)$ and $f_{21}(z)$ be the IMFs from image Z_1 to image Z_2 and vice versa, respectively. By finding the matched points between the cumulative histograms of Z_1 and Z_2 , the IMF $f_{12}(z)$ which maps intensity from Z_1 to Z_2 and the inverse IMF $f_{21}(z)$ can be obtained [25].

It was pointed out in [29] that the IMFs $f_{12}(z)$ and $f_{21}(z)$ have the following properties: 1) The IMF $f_{12}(Z_1(p))$ is

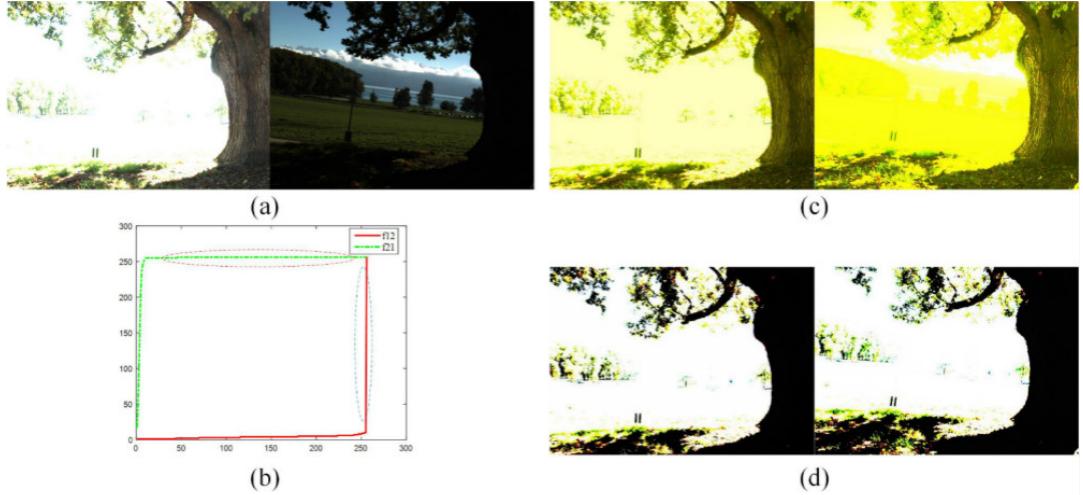


FIGURE 1: Normalization of 1st and 6th images in “BigTree”. (a) Original images. The right image is rotated by 5° , and shifted by 30 pixels and 10 pixels in y-axis and x-axis respectively. (b) IMFs of the two images. (c) Normalized images using a unidirectional mapping function [6]. It maps the image with more information to the one with less information. (d) Normalized images using the proposed bi-directional mapping function. The intensities larger than ζ_1 in the left image are unchanged, the remaining intensities in right image are mapped to left image to ensure the over-exposure regions in both images are consistent. The intensities less than ζ_2 in the right image remain constant, the remaining intensities in the left image are mapped to right image to ensure the under-exposure regions in both images are consistent.

not accurate if $Z_1(p)$ is over-exposed; and 2) The IMF $f_{21}(Z_2(p))$ is not accurate if $Z_2(p)$ is under-exposed. On top of the two observations, two constants are defined as

$$\begin{cases} \zeta_1 = \max_{z_1} \{z_1 | f_{12}(z_1) = \alpha\} \\ \zeta_2 = \min_{z_2} \{z_2 | f_{21}(z_2) = \beta\} \end{cases}, \quad (2)$$

ζ_1 and ζ_2 are further adjusted as

$$\begin{cases} \zeta_1 = \min\{\zeta_1, \beta\} \\ \zeta_2 = \max\{\zeta_2, \alpha\} \end{cases}. \quad (3)$$

The image Z_1 is mapped as

$$\hat{Z}_1(p) = \begin{cases} Z_1(p); & \text{if } Z_1(p) \geq \zeta_1 \\ f_{12}(Z_1(p)); & \text{otherwise} \end{cases}, \quad (4)$$

and the image Z_2 is mapped as

$$\hat{Z}_2(p) = \begin{cases} Z_2(p); & \text{if } Z_2(p) \leq \zeta_2 \\ f_{21}(Z_2(p)); & \text{otherwise} \end{cases}. \quad (5)$$

It can be easily shown that

$$\begin{cases} \hat{Z}_1(p) \in [0, \alpha] \cup [\zeta_1, 255] \\ \hat{Z}_2(p) \in [0, \zeta_2] \cup [\beta, 255] \end{cases}. \quad (6)$$

Clearly, the under-exposed regions of \hat{Z}_1 and \hat{Z}_2 as well as the over-exposed regions of \hat{Z}_1 and \hat{Z}_2 are the same if they are aligned. The translational vector and the rotational angle are estimated by using the mapped images \hat{Z}_1 and \hat{Z}_2 .

As the functions $f_{12}(\cdot)$ and $f_{21}(\cdot)$ are monotonically non-decreasing functions, and

$$\begin{cases} f_{12}(Z_1(p)) \leq Z_1(p) \\ f_{21}(Z_2(p)) \geq Z_2(p) \end{cases}. \quad (7)$$

it can be easily verified that the LBP codes of the mapped images \hat{Z}_1 and \hat{Z}_2 are almost the same as those of the original images Z_1 and Z_2 . Clearly, both the IMF $f_{12}(\cdot)$ and the IMF $f_{21}(\cdot)$ are applied to improve the robustness of LBP codes with respect to possible saturation in the proposed algorithm. The IMFs of images 1 and 6 in “BigTree” sequence is plotted in Fig.1(a). As shown in Fig.1(c), the IMFs result in inconsistency in the two normalized images, such as trees in bright area, and tree trunk in dark area. This is because the region circled by the two ellipses in Fig.1(b) is not reliable due to under/exposure and multi-value mapping. Thus, the parameters α and β are introduced to reduce their effects, and improve the similarity of the two normalized images, as shown in Fig.1(d).

III. ALIGNMENT OF MULTI-EXPOSED IMAGES

In this section, a new alignment algorithm is proposed for multi-exposed images by using a differentiable “Hamming” distance [26].

Let the luminance components of the two normalized images \hat{Z}_1 and \hat{Z}_2 be denoted as Y_1 and Y_2 , respectively. In order to reduce the sensitivity of the descriptor to noise, the images Y_1 and Y_2 can be smoothed with a Gaussian filter in a 3×3 neighborhood ($\sigma = 0.5$) [26] or an edge-preserving smoothing filter such as the weighted guided image filter (WGIF) in a 3×3 neighborhood ($\lambda = 1/256$) [27]. For the pixel p , $\chi(j)$ ($j = 1, 2, \dots, 8$) are its eight neighboring pixels. The Census Transform or the LBP code is [7], [8]

$$S_{Y_i}^{(j)}(p) = \begin{cases} 1; & \text{if } Y_i(\chi(j)) \bowtie Y_i(p) \\ 0; & \text{otherwise} \end{cases}, \quad (8)$$

where the operation $\bowtie \in \{>, <, \geq, \leq\}$. Since the binary descriptor requires only eight comparisons, it is commonly stored as a byte according to [6]

$$S_{Y_i}(p) = \sum_{j=1}^8 2^{j-1} S_{Y_i}^{(j)}(p). \quad (9)$$

Unfortunately, the above binary descriptor does not maintain the morphological invariance to intensity changes due to different weights for different neighborhood pixels. For example, consider two pairs of bit-strings $a = \{1, 0, 1, 0, 1, 1, 1, 0\}$ and $b = \{1, 1, 1, 0, 1, 1, 1, 0\}$, $c = \{1, 1, 1, 0, 1, 1, 1, 1\}$ and $d = \{1, 1, 1, 0, 1, 1, 1, 0\}$. Both the pairs differ at a single bit, and they are clearly similar. However, if the equation (9) is used and matched under Euclidean norm, their distance become 64 and 1, respectively. Clearly, the equation (9) does not capture the closeness in the descriptor space and it is sensitive to the rotational. To address this problem, the binary descriptor must be matched using a binary norm, such as the Hamming distance, which counts the number of mismatched bits as follows:

$$\rho_h(S_{Y_i}(p), S_{Y_i}(p')) = \sum_{j=1}^8 (S_{Y_i}^{(j)}(p) \odot S_{Y_i}^{(j)}(p')). \quad (10)$$

where $\rho_h(a, b)$ is Hamming distance denoted as $a \odot b$ and \odot is the XOR operation. It is noted that equation (10) cannot be formulated as an optimization problem due to the non-differentiability of the Hamming distance. As such, an equivalently differentiable measurement is derived in the following.

Proposition 1: The Hamming distance in the equation (10) is equivalent to the following differentiable cost distance:

$$\rho(S_{Y_i}(p), S_{Y_i}(p')) = \sum_{j=1}^8 (S_{Y_i}^{(j)}(p) - S_{Y_i}^{(j)}(p'))^2. \quad (11)$$

Proof Notice that

$$a \odot b = \begin{cases} 0; & \text{if } \{a = 0, b = 0\} \text{ or } \{a = 1, b = 1\} \\ 1; & \text{if } \{a = 1, b = 0\} \text{ or } \{a = 0, b = 1\} \end{cases},$$

$$(a - b)^2 = \begin{cases} 0; & \text{if } \{a = 0, b = 0\} \text{ or } \{a = 1, b = 1\} \\ 1; & \text{if } \{a = 1, b = 0\} \text{ or } \{a = 0, b = 1\} \end{cases}.$$

It can be easily shown that

$$(a - b)^2 = a \odot b \quad (12)$$

holds if both a and b are in the set $\{0, 1\}$. Therefore, the Hamming distance in the equation (10) is equivalent to the differentiable cost distance in the equation (11). \square

The difference of two LBP coded images is computed as

$$J = \sum_p \rho(S_{Y_1}(p), S_{Y_2}(\psi(p))). \quad (13)$$

where $\psi(p)$ is the corresponding pixel in the image \hat{Z}_2 .

Let p be denoted as (x, y) . Same as [6], it is assumed that the motion between the images \hat{Z}_1 and \hat{Z}_2 is the Euclidean

transformation, i.e.,

$$\psi(p) = [x, y]R^T(\theta) + [t_x, t_y], \quad (14)$$

where $R(\theta)$ is a 2D rotational matrix. The values of θ , t_x , and t_y are obtained from the following optimization problem:

$$\arg \min_{\theta, t_x, t_y} J(\theta, t_x, t_y). \quad (15)$$

If the value of θ , t_x and t_y are small, then image S_{Y_2} can be derived as below [6],

$$\begin{aligned} & S_{Y_2}^{(j)}(\psi(p)) \\ &= S_{Y_1}^{(j)}([x, y]R^T(\theta) + [t_x, t_y]) \\ &\approx S_{Y_1}^{(j)}(x - y\theta + t_x, y + x\theta + t_y) \\ &\approx S_{Y_1}^{(j)}(x, y) + \frac{\partial S_{Y_1}^{(j)}}{\partial x}(t_x - y\theta) + \frac{\partial S_{Y_1}^{(j)}}{\partial y}(x\theta + t_y) \\ &= S_{Y_1}^{(j)}(x, y) + \frac{\partial S_{Y_1}^{(j)}}{\partial x}t_x + \frac{\partial S_{Y_1}^{(j)}}{\partial y}t_y + \left(\frac{\partial S_{Y_1}^{(j)}}{\partial y}x - \frac{\partial S_{Y_1}^{(j)}}{\partial x}y\right)\theta. \end{aligned}$$

For simplicity, define a 3×3 symmetric matrix A as

$$\left\{ \begin{array}{l} A_{11} = \sum_{x,y} \sum_{j=1}^8 \left(\frac{\partial S_{Y_1}^{(j)}}{\partial x}\right)^2 \\ A_{12} = A_{21} = \sum_{x,y} \sum_{j=1}^8 \frac{\partial S_{Y_1}^{(j)}}{\partial y} \frac{\partial S_{Y_1}^{(j)}}{\partial x} \\ A_{13} = A_{31} = \sum_{x,y} \sum_{j=1}^8 \left(\frac{\partial S_{Y_1}^{(j)}}{\partial y}x - \frac{\partial S_{Y_1}^{(j)}}{\partial x}y\right) \frac{\partial S_{Y_1}^{(j)}}{\partial x} \\ A_{22} = \sum_{x,y} \sum_{j=1}^8 \left(\frac{\partial S_{Y_1}^{(j)}}{\partial y}\right)^2 \\ A_{23} = A_{32} = \sum_{x,y} \sum_{j=1}^8 \left(\frac{\partial S_{Y_1}^{(j)}}{\partial y}x - \frac{\partial S_{Y_1}^{(j)}}{\partial x}y\right) \frac{\partial S_{Y_1}^{(j)}}{\partial y} \\ A_{33} = \sum_{x,y} \sum_{j=1}^8 \left(\frac{\partial S_{Y_1}^{(j)}}{\partial y}x - \frac{\partial S_{Y_1}^{(j)}}{\partial x}y\right)^2 \end{array} \right., \quad (16)$$

and a vector $b = [b_1, b_2, b_3]^T$ as

$$\left\{ \begin{array}{l} b_1 = \sum_{x,y} \sum_{j=1}^8 \frac{\partial S_{Y_1}^{(j)}}{\partial x} (S_1^{(j)} - S_2^{(j)}) \\ b_2 = \sum_{x,y} \sum_{j=1}^8 \frac{\partial S_{Y_1}^{(j)}}{\partial y} (S_1^{(j)} - S_2^{(j)}) \\ b_3 = \sum_{x,y} \sum_{j=1}^8 \left(\frac{\partial S_{Y_1}^{(j)}}{\partial y}x - \frac{\partial S_{Y_1}^{(j)}}{\partial x}y\right) (S_1^{(j)} - S_2^{(j)}) \end{array} \right.. \quad (17)$$

It is easily shown that the equation $A [\theta \ t_x \ t_y]^T = b$ can be derived from the optimization problem (15). Once the optimal values of θ^* , t_x^* , and t_y^* are obtained, the image Z_2 is aligned as $Z_2([x, y]R^T(-\theta^*) + [-t_x^*, -t_y^*])$.

The optimization is very fast due to linear regression, and the solution can be achieved to sub-pixel accuracy. It is noted that the Taylor series involved in the estimation of (θ, t_x, t_y) is a good approximation only when the motion is very small. To further increase speed and robust to large motion, the optimization is implemented by coarse-to-fine technique which is represented using the Gaussian pyramid [6], shown in Algorithm 1, where the `wrap()` function is an affine transformation and is used to update the position of pixels.

It should be pointed out that the single scale with a few iterations is enough if the initial values of θ , t_x and t_y can be computed by using the inertial measurement unit (IMU)

Algorithm 1 Pseudo code of the proposed method

Input: Two differently exposed images Z_1 (Refernce images) and Z_2 (Slave image).

Output: Motion parameters between image pairs Z_1 and Z_2

- 1: $\hat{Z}_1, \hat{Z}_2 = \text{Normalization}(Z_1, Z_2)$
- 2: Estimate t_x^0, t_y^0 and θ^0 by histogram-based matching
- 3: $\hat{Z}_1 = \text{Wrap}(\hat{Z}_1, t_x^0, t_y^0, \theta^0)$
- 4: Build N-tier pyramid Gaussian pyramid of \hat{Z}_1, \hat{Z}_2
- 5: **for** $\text{pyrlevel} = N:1$ **do**
- 6: Code $\hat{Z}_1^{\text{pyrlevel}}, \hat{Z}_2^{\text{pyrlevel}}$ by equation (8)
- 7: Estimate $t_x^{\text{pyrlevel}}, t_y^{\text{pyrlevel}}$ and θ^{pyrlevel} by “Hamming Distance”
- 8: Update $t_x^{\text{pyrlevel}-1} = 2 * t_x^{\text{pyrlevel}}, t_y^{\text{pyrlevel}-1} = 2 * t_y^{\text{pyrlevel}}, \theta^{\text{pyrlevel}-1} = \theta^{\text{pyrlevel}}$
- 9: Wrap $\hat{Z}_1^{\text{pyrlevel}-1}$ with $t_x^{\text{pyrlevel}}, t_y^{\text{pyrlevel}}, \theta^{\text{pyrlevel}}$
- 10: **return** t_x^*, t_y^* and θ^*

integration [31] or other methods, such as histogram-based matching [6]. Besides, the LBP code is also a simple way to reduce the effects of haze [32].

IV. EXPERIMENTAL RESULTS

In this section, the proposed method is evaluated with a variety of synthesized images from benchmark datasets, public datasets (Cai’s dataset) [33] and real images. All datasets are released in open source [34]. Readers can browse the full-size images in the digital version and zoom in to better observe the differences. The results of the proposed method are compared with: 1) existing non-parametric ordering features using MTB [5], CT [7], LBP [8] and BRIEF [9] methods, while employing the alignment algorithm shown in Section III; 2) the intensity-based method IMF; 3) the feature-based method SIFT [35] and the hybrid method IMF+SIFT; 4) IMF+BRIEF and IMF+BRIEF+“Hamming” distance (“HD”); 5) IMF+LBP [6]; and 6) Learning-based matching methods SuperPoint [22] and LF-Net [23].

A. TEST ON SYNTHETIC IMAGES

9 well geometrically aligned sequences in the benchmark datasets, and 37 randomly selected and well geometrically aligned sequences in [33] are used to evaluate different alignment algorithms. These multi-exposure image sequences are captured by different cameras, which cover a broad range of scenes, subjects and lighting conditions. Each sequence has 3 to 18 differently exposed images and is arranged from the longest exposure to the shortest exposure. Some sample sequences are presented in Fig. 2.

To evaluate the alignment performance, the first image in each sequence is selected as the reference, and other images are rotated by 5° , and shifted 30 pixels and 10 pixels in y -

axis and x -axis, respectively, to produce unaligned samples for latter experiments. The alignment performances are verified in terms of average errors between estimated motion parameters and the true values ($5^\circ, 30, 10$), and the results are shown in Table 1 and 2. The average error of the proposed method is the smallest, which implies the proposed method is suitable for various scene sets alignment and robust to intensity variations.

A typical sequence named “Big Tree” (1EV interval) in benchmark datasets, shown in Fig. 2(a), is selected to test each algorithm’s robustness to exposure changes. Table 3 summarizes the motion errors in aligning for the “BigTree” sequence. The SIFT, BRIEF, SuperPoint and LF-Net do not provide motion parameters but they provide the matching point pairs of two images. Thus, the performances of the four methods are assessed by the number of consistent points and mismatching ones. Table 3 reveals that: 1) Among the order feature-based methods, the error of the MTB method is the smallest, which illustrates that the MTB is robust to the varying exposures. But, as seen in Table 4-5, the performance of MTB is not good in the other two sequences, such as “Snowman” shown in Fig.2(b) because the reference image is seriously over-exposed. 2) The performances of the IMF, CT, LBP, SIFT and BRIEF methods are not good, but the IMF+SIFT and IMF+LBP are robust to intensity variations. This shows that IMF improves the similarity of the differently exposed images. 3) BRIEF and IMF+BRIEF do not offer good results due to large intensity variations, while IMF+BRIEF+“HD” and the proposed method demonstrate high robustness to intensity variations. This indicates that the proposed “Hamming” distance improves the alignment accuracy of binary operators, especially for the rotational angle θ . The proposed normalization method reduces the effects of saturation and noise in the dark regions, and improves the similarity of two differently exposed images. 4) It can be observed from Table 3 and Table 4 that the performance of the learning based methods on “Snowman” sequence is worse than their performance on “BigTree” sequence because the reference image on “Snowman” sequence is completely washed out. The SuperPoint is more robust to intensity variations than the LF-Net, i.e., it can find 6 pairs of consistent points in the third image, but 0 pair from the fourth image due to large exposure difference between them, as shown in Fig. 3(a). By contrast, the method of LF-net can find more feature points, but can not match these feature points correctly, as shown in Fig. 3(b). 5) In comparison with the above methods, the proposed method is robust to large EV intervals (7EV in “Bigtree”), especially for over-exposed image sequences, in which the reference image is completely washed out. The LBP based alignment algorithm does not work as indicated in Table 4. This was also pointed out in [26]. The performance of SuperPoint on sequences “Pillar” and “Inscriptionn” is better than its performance on “Snowman” and “BigTree” sequences because the reference images in sequences “Pillar” and “Inscriptionn” contain rich features as shown in Fig. 4. Nevertheless, the proposed method is also superior to other

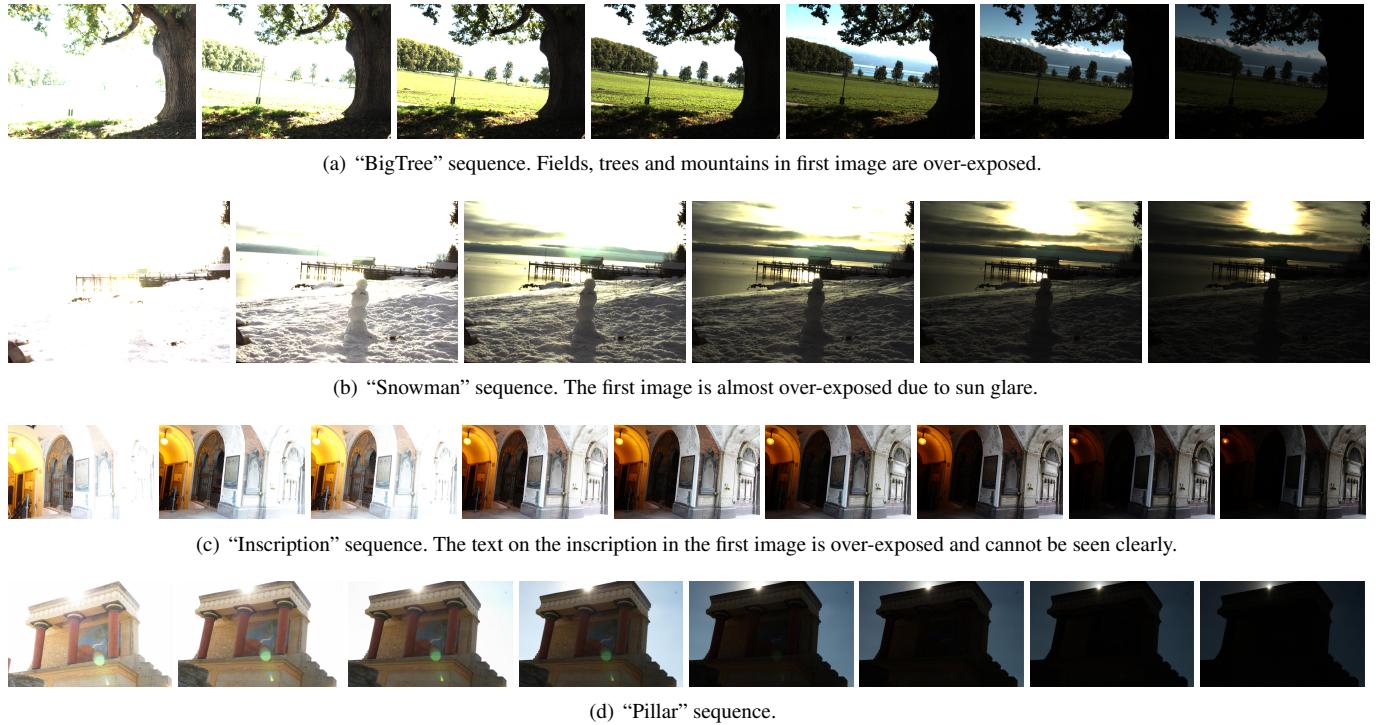


FIGURE 2: Multi-exposed image sequences. Each sequence is arranged from long exposure to short exposure and the first image in each sequence is reference image. The interval between images in (a) and (b) is 1EV. The exposure levels in (c) and (d) are manually set based on the lighting ratio of the scene. The full-size images have been uploaded to the internet, readers can zoom them for better observation [34].

TABLE 1: Overall results performed on benchmark database. $\Delta\theta$, Δt_y and Δt_x , are the average errors between estimated motion parameters and (5° , 30, 10).

Method	$\Delta\theta(^{\circ})$			Δt_y			Δt_x		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
MTB	2.8	23.5	0	17.8	149	0.8	9.6	92	0.4
CT	6	39	0	64	539	0	49	195	0.5
LBP	3	23	0	26	213	0.8	34	282	0
IMF	1.3	6.2	0	4.1	32	0.1	6.1	130	0
IMF+LBP	1.1	13	0	8.5	192	0.9	10	254	0
IMF+BRIEF +“HD”	1.9	14.1	0	14.6	181	0.3	15.8	253	0.3
Ours	0.6	10	0	3.8	34	0.7	1.8	11.1	0.1

TABLE 2: Overall results performed on 37 Cai’s database [33].

Method	$\Delta\theta(^{\circ})$			Δt_y			Δt_x		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
MTB	0.9	23	0	7.2	177	0.1	4.2	151	0
CT	5.2	62	0	86	542	0.5	155	984	0
LBP	0.7	12	0	11	137	0.1	8.3	369	0
IMF	1.6	6	0	7.1	201	0.1	7.3	386	0
IMF+LBP	0.9	24	0	8.8	195	0	8.3	346	0
IMF+BRIEF +“HD”	0.7	14.8	0	11.2	199	0	9.1	352	0
Ours	0.2	3.6	0	6.9	241	0	1.5	9.9	0

methods as shown in Table 5 and Table 6. The proposed image alignment algorithm outperforms the DVO algorithm in [26] and multi-exposed image alignment in [6].

B. TEST ON REAL IMAGES

For further objective evaluation, 35 sequences of real scenes are used here, which are captured using a NIKON D300, and a CANON EOS-1Ds Mark II, they are comprised of various scenarios including small/big camera motion, static/dynamic scenes, little/severe saturation, 1EV/2EV intervals and so on. Each sequence contains 4 to 11 images, two sample sequences of multi-exposure image datasets are presented in Fig. 5. As motion parameters are unknown, alignment results are evaluated in terms of mutual information (MI) [36]. Selecting the brightest image as the reference, the MIs in different exposures are shown in Fig. 6. The performance of alignment methods usually decreases as the EV interval increases. This is because the normalization error increases with the increment of EV interval. On the other hand, the performances of the CT, IMF+LBP and the proposed method are robust to 7EV and 8 EV intervals, among which, the proposed method achieves the largest mutual information, especially when the EV interval between the two images is large. The proposed normalization method reduces the errors of IMF in under/over exposed regions, and improves the similarity of two large-exposure-ratio images. It is also observed from Fig. 6 that the minimal, median and maximal MI of the proposed method is the largest, which implies the proposed method is suitable for various scene sets alignment and robust to intensity variations.

TABLE 3: Performances of different methods on “BigTree” sequence in terms of motion parameters ($\Delta\theta, \Delta t_y, \Delta t_x$).

Method		2	3	4	5	6	7
MTB	$\Delta\theta$	0	0	0	0	0	0
	$\Delta t_y, \Delta t_x$	1.6,1.1	1.6,0.9	1.7,0.8	1.7,0.7	1.7,0.7	1.6,0.7
CT	$\Delta\theta$	0	0	1.8	1.4	3	5
	$\Delta t_y, \Delta t_x$	1,1.1	0.6,1.2	34,19	84,175	21,195	59,192
LBP	$\Delta\theta$	0	0	0.5	1	3.2	11.7
	$\Delta t_y, \Delta t_x$	2,1	2,1	5.5,0.6	19,138	17,33	36,65
IMF	$\Delta\theta$	2.4	1.6	1.9	3	2.6	0
	$\Delta t_y, \Delta t_x$	0.4,3	0.1,2.2	0.2,2.3	2.5,5	1.1,5.1	1.8,1.2
IMF+LBP	$\Delta\theta$	0	0	0	0	0	4.4
	$\Delta t_y, \Delta t_x$	1.7,1.3	1.7,1.3	1.7,1.3	1.6,1.3	1.7,1.3	15,11
IMF+BRIEF+“HD”	$\Delta\theta$	0	0	0	0	0	6
	$\Delta t_y, \Delta t_x$	3,1.2	2,5.5	2.3,0	1,5	2.7,4.3	31,12
Ours	$\Delta\theta$	0	0	0	0	0	0
	$\Delta t_y, \Delta t_x$	1.6,1.4	1.6,1.4	1.6,1.4	1.6,1.5	3.2,0	1.5,1.5
SIFT		✓	✓	✗	✗	✗	✗
IMF+SIFT		✓	✓	✓	✓	✗	✗
BRIEF		✗	✗	✗	✗	✗	✗
IMF+BRIEF		✗	✗	✗	✗	✗	✗
SuperPoint		✓	✓	✓	✗	✗	✗
LF-Net		✗	✗	✗	✗	✗	✗

✓ indicates good alignment, ✗ indicates wrong alignment.

TABLE 4: Performances of different methods on “Snowman” sequence in terms of motion parameters ($\Delta\theta, \Delta t_y, \Delta t_x$).

Method		2	3	4	5	6
MTB	$\Delta\theta$	0.7	8.5	10	10	8.6
	$\Delta t_y, \Delta t_x$	3,1	32,4	38,15	40,17	33,4
CT	$\Delta\theta$	0.2	1.6	7	20	39
	$\Delta t_y, \Delta t_x$	0,1	32,150	2,77	39,38	20,75
LBP	$\Delta\theta$	3	5	7.4	5.4	0.4
	$\Delta t_y, \Delta t_x$	17,34	51,130	0,189	31,145	157,127
IMF	$\Delta\theta$	6	6	5.4	2.4	5.2
	$\Delta t_y, \Delta t_x$	16,3	16,3	30,9	32,10	17,10
IMF+LBP	$\Delta\theta$	0	0	0	0.1	0.1
	$\Delta t_y, \Delta t_x$	1.4,0.8	1.6,0.7	1.4,1	1.4,1	1.5,1
IMF+BRIEF+“HD”	$\Delta\theta$	1.3	0.5	0.1	0.7	0.7
	$\Delta t_y, \Delta t_x$	11,5	1,0	15,16	5,6	11,10
Ours	$\Delta\theta$	0	0	0	0	0
	$\Delta t_y, \Delta t_x$	1.6,1.1	1.7,1.0	1.8,1.0	1.8,1.0	1.8,0.8
SIFT		✓	✓	✗	✗	✗
IMF+SIFT		✓	✓	✓	✓	✓
BRIEF		✗	✗	✗	✗	✗
IMF+BRIEF		✗	✗	✗	✗	✗
SuperPoint		✓	✓	✗	✗	✗
LF-Net		✗	✗	✗	✗	✗

✓ indicates good alignment, ✗ indicates wrong alignment.

TABLE 5: Performances of different methods on “Pillar” sequence in terms of motion parameters ($\Delta\theta, \Delta t_y, \Delta t_x$).

Method		2	3	4	5	6	7	8
MTB	$\Delta\theta$	0.1	0.1	0.1	0.1	0.1	0.3	20.5
	$\Delta t_y, \Delta t_x$	2,1,1.2	1,6,1.5	1,3,1.9	1,4,1.5	0,8,1.4	0,1,0.7	156,5,1.7
CT	$\Delta\theta$	0	0	0	4.7	3.6	3.3	6
	$\Delta t_y, \Delta t_x$	2,0,8	1,6,1,1	1,2,1,5	89,20	470,86	471,115	205,595
LBP	$\Delta\theta$	0	0	0	0.1	0.1	6.3	5.7
	$\Delta t_y, \Delta t_x$	2,0,8	1,6,1,1	1,2,1,5	1,4,1,1	1,6,0,8	47,16,2	46,3,29
IMF	$\Delta\theta$	4,3	4,6	2,2	0,8	0,6	0,4	0.5
	$\Delta t_y, \Delta t_x$	9,7,7,7	12,9	11,6	3,4,2,7	2,5,2,2	1,2,0,8	1.3,0.2
IMF+LBP	$\Delta\theta$	0	0	0	0	0	0.2	3
	$\Delta t_y, \Delta t_x$	2,0,9	1,6,1,2	1,3,1,6	1,4,1,5	1,5,1,4	1,7,0,7	15,6,1
IMF+BRIEF+“HD”	$\Delta\theta$	0.2	0.1	0.6	0	1.2	0.4	0.5
	$\Delta t_y, \Delta t_x$	3,9,12,6	0,7,1	9,8,16,3	5,1,5,5	1,6,3,0	33,6,6,7	12,7,8,9
Ours	$\Delta\theta$	0	0	0	0	0	0.1	0.1
	$\Delta t_y, \Delta t_x$	2,0,9	1,6,1,2	1,2,1,6	1,4,1,4	1,5,1,2	1,2,0,4	1,2,0,2
SIFT		✓	✗	✗	✗	✗	✗	✗
IMF+SIFT		✓	✗	✗	✗	✗	✗	✗
BRIEF		✗	✗	✗	✗	✗	✗	✗
IMF+BRIEF		✓	✗	✗	✗	✗	✗	✗
SuperPoint		✓	✓	✓	✓	✓	✓	✗
LF-Net		✗	✗	✗	✗	✗	✗	✗

✓ indicates good alignment, ✗ indicates wrong alignment.

TABLE 6: Performances of different methods on “Inscription” sequence in terms of motion parameters ($\Delta\theta, \Delta t_y, \Delta t_x$).

Method		2	3	4	5	6	7	8	9
MTB	$\Delta\theta$	0.3	0.1	0	0	0	0.1	0.1	0.1
	$\Delta t_y, \Delta t_x$	3.8,1.2	1.0,1.4	0.8,1.5	0.4,1.8	0.5,1.7	1.1,1.4	1.2,1.5	1.8,1.3
CT	$\Delta\theta$	0	0.1	6	6.8	7.4	3.6	2.6	7
	$\Delta t_y, \Delta t_x$	0.9,1	0.8,1.1	15,233	48,4	82.5,523	51,570	36,675	63,707
LBP	$\Delta\theta$	0	0	0.1	0.1	0.1	0	0	0
	$\Delta t_y, \Delta t_x$	1.3,1.2	1.1,1.3	1.0,1.1	0.6,1.4	0.8,1.2	1.4,0.8	1.4,0.7	1.8,0.5
IMF	$\Delta\theta$	2.2	4	2.1	3.2	3.3	3.3	3.5	0.1
	$\Delta t_y, \Delta t_x$	4.8,6.6	23,7.5	7.4,6.6	11.4,7.9	11.5,8	11.7,9	8.4,9.5	2.2,0.9
IMF+LBP	$\Delta\theta$	0	0	0.1	0.1	0.1	0	0	24
	$\Delta t_y, \Delta t_x$	1.3,1.4	1.1,1.4	1,1.5	0.7,1.8	0.8,1.7	1.4,1.3	1.4,1.3	190,333
IMF+BRIEF+“HD”	$\Delta\theta$	0	0	0	0.2	0.1	0.1	0	14.8
	$\Delta t_y, \Delta t_x$	4.3,17	1.2,2	4.6,20.5	1.5,0.3	0.9,2.3	4.5,14.4	0.6,13.5	27,11
Ours	$\Delta\theta$	0	0	0.1	0.1	0.1	0	0	0
	$\Delta t_y, \Delta t_x$	1.2,1.3	1.1,1.4	1,1.4	0.5,1.6	0.7,1.6	1.3,1.1	1.3,1.2	1.7,1.1
SIFT									
IMF+SIFT									
BRIEF									
IMF+BRIEF									
SuperPoint									
LF-Net									

✓ indicates good alignment, ✗ indicates wrong alignment.

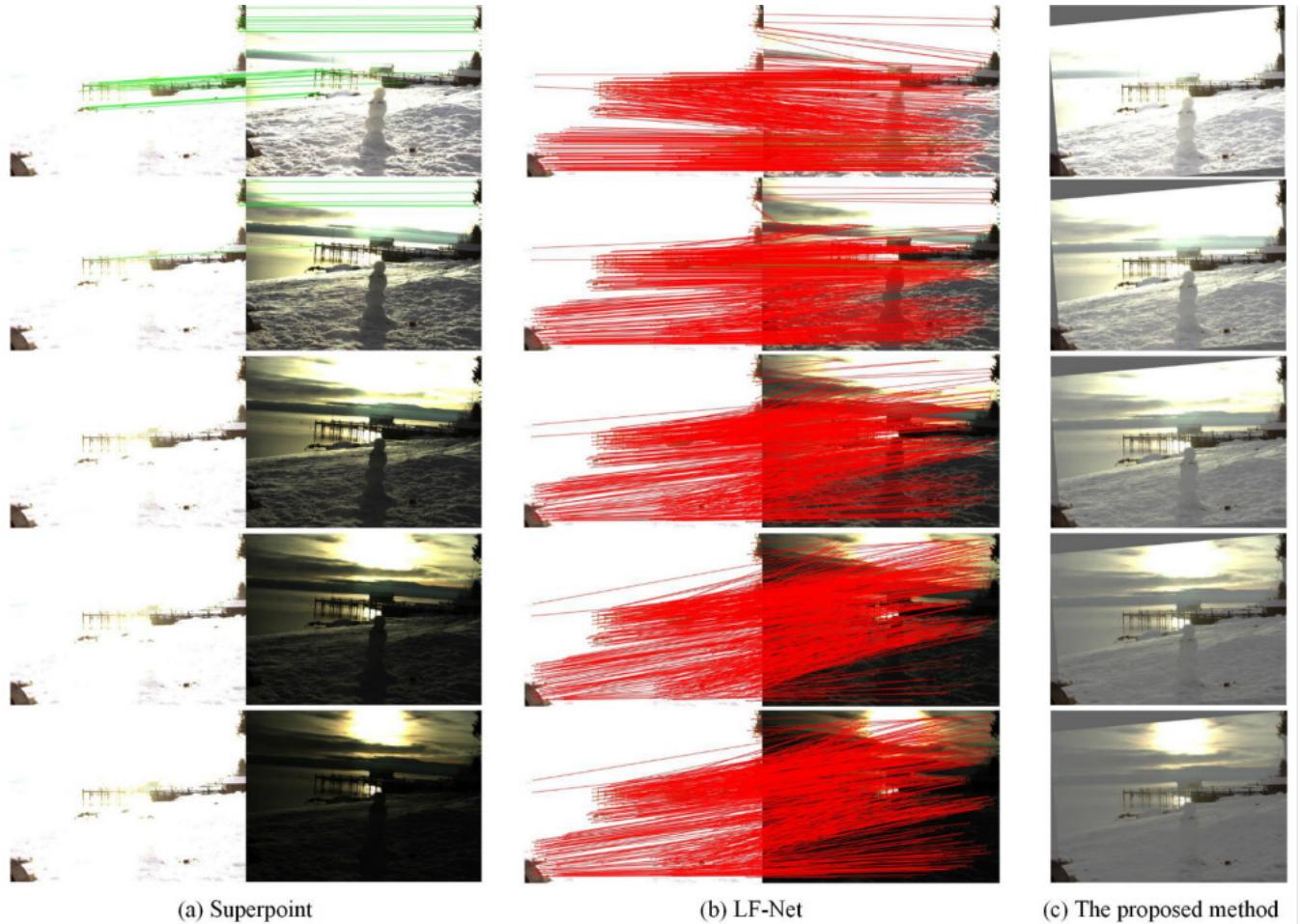


FIGURE 3: Comparison of learning-based methods and the proposed method on “Snowmann” sequence. In this figure, red lines indicate consistent point pairs, and green lines indicate mismatching point pairs.

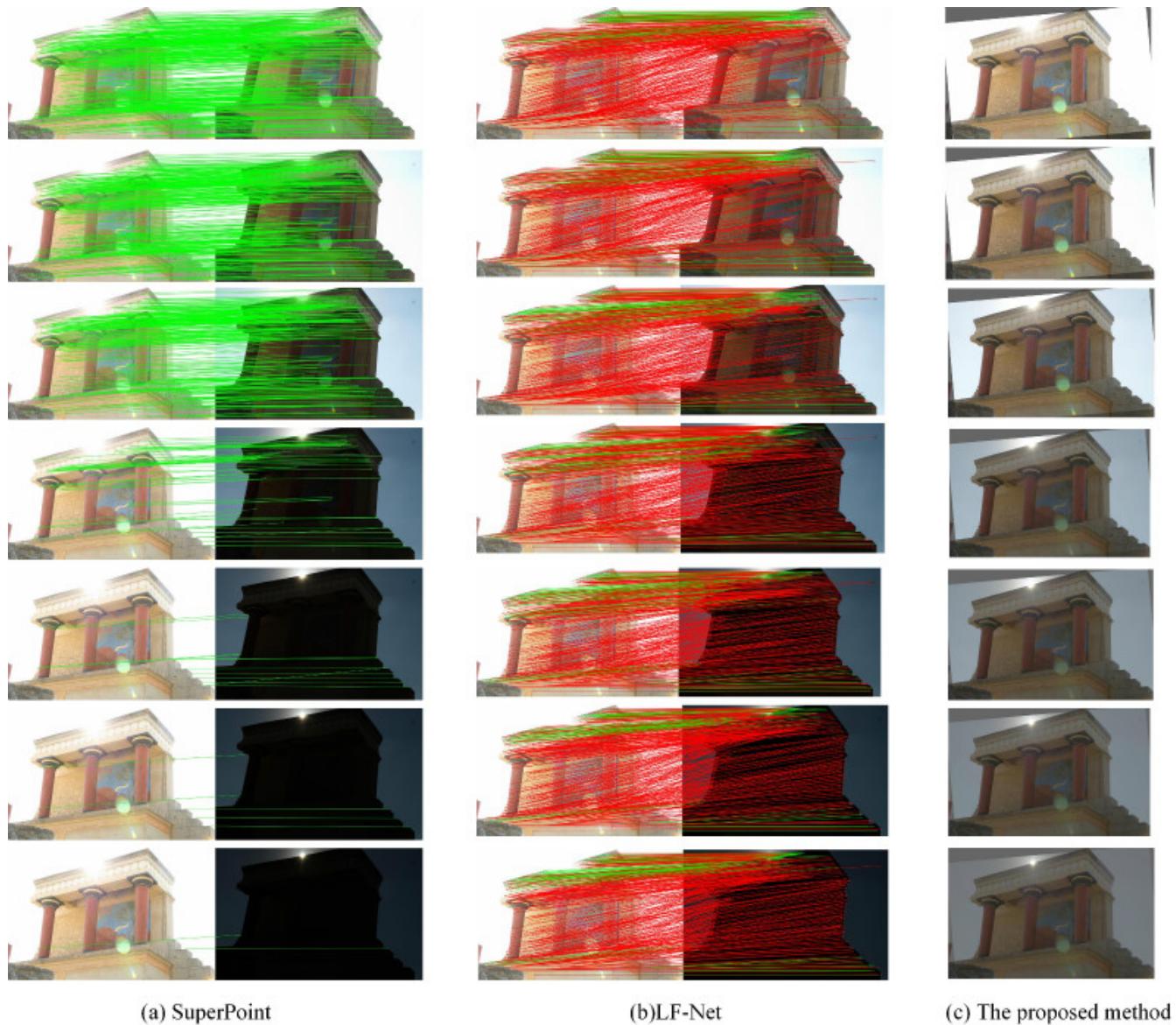


FIGURE 4: Comparison of learning-based methods and the proposed method on ‘‘Pillar’’ sequence. In this figure, red lines indicate consistent point pairs, and green lines indicate mismatching point pairs.

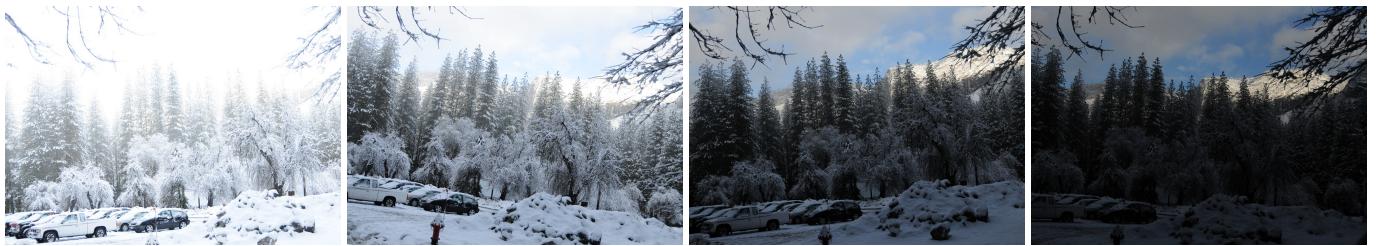
C. TESTS ON EFFICIENCY

In order to demonstrate the efficiency of the proposed method, the operation times of different methods are calculated respectively. The learning-based methods are running in Python code¹, the rest are running in MATLAB code. The experiments are performed on DELL computer with i7-6700 CPU, 8GB RAM, and 64bit OS. However, the performances of MTB, LBP, CT, IMF, SIFT, BRIEF, IMF+BRIEF and the learning-based methods failed to align multi-exposed images in the previous experiment. The descriptor of IMF+BRIEF+“HD” is a 256-vector which needs expensive computation to match two descriptors under “Hamming

distance". Therefore, the operation times of the proposed method are compared with IMF+LBP and IMF+SIFT. The operation times (second) of different algorithms are tabulated in Table 7.

Two sequences are randomly selected to test the efficiency. The results are given in Table 7. The operation time of the algorithm is proportional to image size and the number of features. IMF+SIFT is expensive in computation due to complex feature extraction and description as shown in Table 8. The normalization process of the proposed method is computationally more expensive than that of [6], but its alignment is 2.5 times faster than that of IMF+LBP and IMF+SIFT methods.

¹SuperPoint, <https://github.com/magicleap/SuperPointPretrainedNetwork>; LF-Net, <https://github.com/vcg-uvic/lf-net-release>



(a) "Snow" sequence.



(b) "Table" sequence.

FIGURE 5: Real multi-exposed images. Each sequence is arranged from long exposure to short exposure and the reference and first image in each sequence have over-exposed regions.

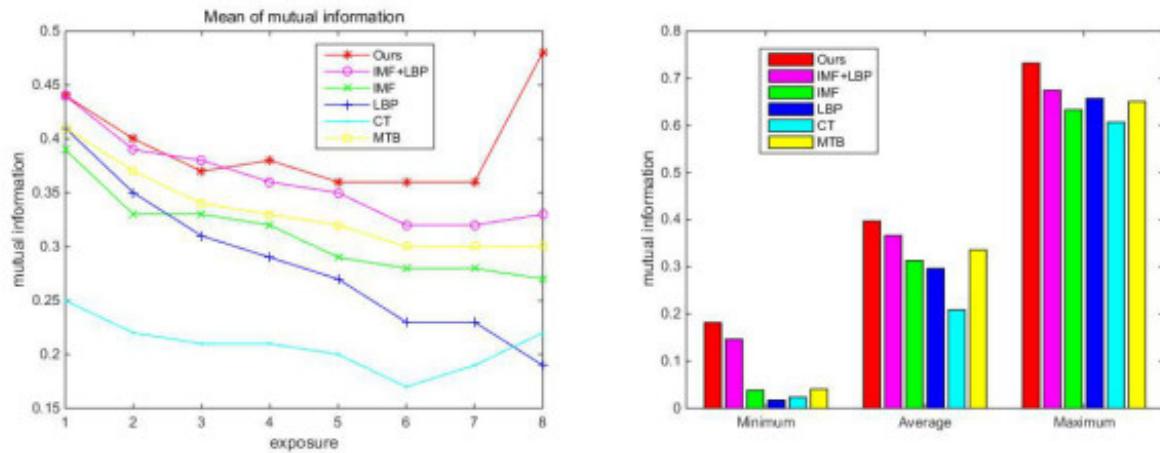


FIGURE 6: Mutual information for 35 real sequences.

TABLE 7: Operation times (second) of different algorithms. All the codes are written in Matlab.

Sequence	No. of images	Image size	IMF+LBP	IMF+SIFT	Ours
Snowman	6	693*505	68.6	183.2	48.4
Snow	4	960*680	57.3	1209	36.6

TABLE 8: Operation times (second) of normalization, feature extraction and description, alignment when aligning the first two "Snowman" images. All the codes are written in Matlab.

Method	Normalization	Feature extraction and description	Alignment
IMF+SIFT	0.7	59.8	6.0
IMF+LBP	0.7	16.3	
Ours	2.7	6.1	

V. CONCLUSION

In this paper, a novel image alignment algorithm has been proposed for multi-exposed images with saturated regions by using a differentiable "Hamming" distance. The images are first normalized by using intensity mapping functions to reduce the effects of saturation. The normalized images are then coded by LBP and aligned by the proposed algorithm.

Experimental results show that the proposed algorithm outperforms state-of-the-art alignment algorithms for differently exposed images.

There are many applications of the proposed algorithm. For example, it can be used in the autonomous SLAM [2] with prior knowledge on the mapping environment, the direct visual odometry [26] with the assistance of IMU integration [31], and panorama imaging for tele-operation of unmanned ground vehicles [3]. All these applications will be studied in our future research.

REFERENCES

- [1] R. Szeliski, "Image alignment and stitching: a tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no.1, pp.1-104, 2007.
- [2] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: dense SLAM without a pose graph," in *Robotics: Science & Systems*, 2015.
- [3] W. Yao and Z. G. Li, "Instant color matching for mobile panorama imaging," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 6-10, Jan. 2015.
- [4] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *proceedings of ACM SIGGRAPH*, ACM, pp.369-378, 1997.

- [5] G. Ward, "Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures," *Journal of graphics tools*, vol.8, no.2, pp.17-30, 2003.
- [6] S. Q. Wu, Z. G. Li, J. H. Zheng, and Z. J. Zhu, "Exposure robust method for aligning differently exposed images," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 885–889, Jul. 2014.
- [7] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision-ECCV*. Springer, pp. 151-158, 1994.
- [8] T. Ojala, M. Pietikainen, and D. Harwood "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol.29, no.1, pp.51-59, 1996.
- [9] M. Calonder, V. Lepetit, M. Ozuval, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.34, no.7, pp.1281-1298, 2001.
- [10] E. Rublee, V. Rabaud, K. Konolige, and G.R. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *International Conference on Computer Vision*, 2012.
- [11] R. Salakhutdinov and G. Hinton, "Semantic hashing," in *International Journal of Approximate Reasoning*, vol.50, no.7, pp. 969-978, 2009.
- [12] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *International Conference on Neural Information Processing Systems*, 2008.
- [13] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary beyond-point descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp.2874-2881, 2013.
- [14] V. Balntas, L. Tang, and K. Mikolajczyk, "BOLD - Binary online learned descriptor for efficient image matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [15] K. Lin, J. Lu, C.S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [16] J. Lu, V. E. Liang, X. Zhou and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2041-2056, Oct. 2015.
- [17] Y. Duan, J. Lu, J. Feng and J. Zhou, "Context-aware local binary feature learning for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1139-1153, 2018.
- [18] H. Jain, J. Zepeda, P. Pálrez, and R. Gribonval, "SUBIC: A supervised, structured binary code for image search," in *IEEE International Conference on Computer Vision (ICCV), Venice*, pp. 833-842, 2017.
- [19] Y. Duan, J. Lu, Z. Wang, J. Feng and J. Zhou, "Learning deep binary descriptor with multi-quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1924-1938, 2019.
- [20] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 596-605, 2018.
- [21] K. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *ECCV*, pp 467-483, 2016.
- [22] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," 2017.
- [23] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-net: Learning local features from images," 2018.
- [24] X. Shen, C. Wang, X. Li, Z. Yu, and Z. He, "RF-Net: An end-to-end image matching network based on receptive field," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [25] M. Grossberg and S. Nayar "Determining the camera response from images: What is knowable?" *IEEE Transactions on pattern analysis and machine intelligence*, vol.25, no.11, pp.1455-1467, Nov. 2003.
- [26] H. Alismail1, M. Kaess, B. Browning, and S. Lucey, "Direct visual odometry in low light using binary descriptors," in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.
- [27] Z. G. Li, J. H. Zheng, Z. J. Zhu, W. Yao, and S. Q. Wu, "Weighted guided image filtering," *IEEE Trans. on Image Processing*, vol. 24, no. 1, pp. 120-129, Jan. 2014.
- [28] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
- [29] Z. G. Li, J. H. Zheng, Z. J. Zhu, and S. Q. Wu, "Selectively detail-enhanced fusion of differently exposed images with moving objects," *IEEE Trans. on Image Processing*, vol. 23, no. 10, pp. 4372–4382, Oct. 2014.
- [30] J. Bouguet "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," Intel, Microprocessor Research Labs, pp. 1-9, Jan. 2000.
- [31] J. Henawy, Z. G. Li, W. Y. Yau, and G. Seet, "Accurate IMU factor using switched linear systems for VIO," *IEEE Transactions On Industrial Electronics*, Accepted, Jun. 2020.
- [32] Z. G. Li and J. H. Zheng, "Single image haze removal Using globally guided image filtering," *IEEE Trans. on Image Processing*, vol. 27, no. 1, pp. 442–450, Jan. 2018.
- [33] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Transactions on Image Processing*, vol.27, no.4, pp.2049-2062. Apr. 2018.
- [34] <https://github.com/gelinlan/Robust-Alignment-of-Multi-Exposed-Images-with-Saturated-Regions>.
- [35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol.60, no.2, pp.91-110, Feb. 2004.
- [36] P. Viola and W. M. Wells "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol.24, no.2, pp.137-154, Feb. 1997.
