Augmented Visual Feature Modelling for Matching in Low-Visibility based on Cycle-Labelling of Superpixel Flow[★]

Shu Zhang[a], Hui Yu[b], Ting Wang[c] and Junyu Dong[a,*]

[a] *Ocean University of China, Qingdao, China*
[b] *University of Portsmouth, Portsmouth, UK*
[c] *Shandong University of Science and Technology, Qingdao, China*

ARTICLE INFO

Keywords:
feature modelling
superpixel
low-visibility
optical flow
illumination variation
outlier removal

ABSTRACT

There are many automation systems that are required to work under poor visual conditions, such as auto-navigation in foggy or underwater environment. The low-visibility poses challenges for traditional feature modelling methods, which commonly contribute as a key component to such autonomous systems. It can thus negatively impact the performance of those computing systems. For example, the matching precision (matching quality) and the successfully identified matches (matching quantity) can both drop dramatically in low-visibility. On the other hand, human vision system can robustly identify visual features correctly despite the variations in lighting conditions. Inspired by human knowledge of perceiving visual features, this paper presents a novel feature modelling solution under poor visual conditions. Based on a color constancy enhanced illumination alignment, a new concept called Superpixel Flow (SPF) is proposed to model the visual features in images. SPF is generated considering the content motions across frame pairs, which make it easier to track across frames compared with classic Superpixels. The matching is achieved by a cycle-labelling strategy using Markov Random Field (MRF) with energy functions composed according to human knowledge of compare visual features. An outlier removal follows to further improve the matching accuracy. Competitive performance is demonstrated in the experiments compared with state-of-the-art approaches.

1. Introduction

For many occasions, a computing system has to be operational in various low-visibility conditions, for example, indoor human face analysis [1] with insufficient illuminations, autonomous aerial vehicle driving [2] in a foggy weather, small object detection [3], or underwater fish tracking [4]. The feature perception is a key component in those computing systems [5]. For most online systems, it is even more crucial to establish feature correspondence between consecutive frames accurately and robustly for proper functionality. However, in low-visibility conditions, the saliencies in the visual images can be highly impaired, where the pixel-based feature methods can fail easily, for example, in underwater environment [6]. Such failures can pose negative impacts on those computational tasks.

In the meanwhile, human can capture and identify effective visual features easily in various visibility conditions. Inspired by such human knowledge of perceiving visual features, this paper presents a reliable solution for feature modelling with poor visual condition. Based on the color constancy in human vision system, a new concept called Superpixel Flow (SPF) is proposed to represent visual features between consecutive video frames. The superpixel is originally used for image segmentation and known as one of the most promising representations of small regions in an image [7]. It assumes that the pixels in a superpixel come from the same projection of a small 3D planar patch. With this knowledge, instead of analysing each single pixel, we propose SPF as a higher-level feature descriptor, which considers a collection of pixels to describe the local salience. Feature correspondence is achieved using Markov Random Field (MRF) [8] as a labelling problem, where human knowledge in comparing two similar visual patches is employed. MRF acts like a 2D version of Markov Chain, which describes a sequence of possible events where each event depends only on the state attained in the previous event. An outlier removal follows in the end of the pipeline to further refine the results.

The contributions of this paper are as follows:

1. **An adaptive illumination alignment (IA)** for low-visibility frame-pair enhancement. Even for consecutive frames in a same environment, the degradation of each frame can vary. An IA will make the surface appearances of those frames to be sufficiently close (aligned) inspired by human vision system. This gives positive assistance for extracting matchable features across frames.
2. **A Superpixel Flow (SPF)** generation algorithm, which adaptively considers the motion state of image content appeared in consecutive image pair. Compared to the traditional superpixel, the proposed SPF can effectively enhance the trackability of contextual features in the frames.
3. **A cycle-labelling based matching strategy** with Markov Random Field. The energy functions are designed inspired by the principle of human visual perception.
4. **An outlier removal** using a Gaussian distribution enhanced randomly down sampling (RDS) strategy. Competitive results are demonstrated compared with several methods.

[★] This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) (EP/N025849/1); National Natural Science Foundation of China (NSFC) (41906177, 41576011); China Postdoctroal Science Fondation (2019M652476); International Science and Technology Cooperation Program of China (ISTCP) (2014DFA10410).

[*] Corresponding author: Junyu Dong.

zhangshu@ouc.edu.cn (S. Zhang); hui.yu@port.ac.uk (H. Yu); wangting@sdust.edu.cn (T. Wang); dongjunyu@ouc.edu.cn (J. Dong)

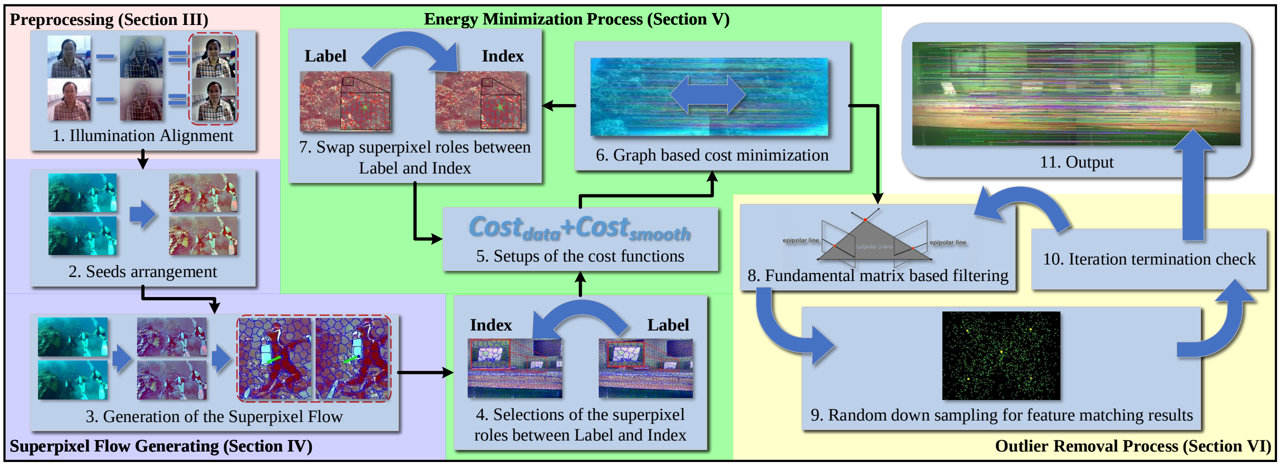

Preprocessing (Section III)
1. Illumination Alignment
2. Seeds arrangement
Energy Minimization Process (Section V)
Label
Index
7. Swap superpixel roles between Label and Index
6. Graph based cost minimization
$Cost_{data}+Cost_{smooth}$
5. Setups of the cost functions
11. Output
epipolar line
epipolar plane
epipolar line
8. Fundamental matrix based filtering
10. Iteration termination check
3. Generation of the Superpixel Flow
Index
Label
4. Selections of the superpixel roles between Label and Index
9. Random down sampling for feature matching results
Superpixel Flow Generating (Section IV)
Outlier Removal Process (Section VI)

Figure 1: Illustration of the work flow proposed in this paper.

The whole pipeline is illustrated in Fig. 1. The rest of the paper is organized as follows: Section 2 discusses the existing researches in the related areas. Section 3 introduces the IA. SPF is presented in Section 4. The MRF-based matching is discussed in Section 5. Outlier removal is described in Section 6. Section 7 demonstrates the experiments. Section 8 concludes the paper.

2. Related Work

Conventional approaches. Feature extraction and matching have been two long studied issues for computing systems in many fields. For example, Narote *et al.*[9] discussed the automatic lane detections by pixel-based features. However, the performance suffers from a lower accuracy due to the low-visibilities caused by fog, rain or night-time. Up to now, there are many popular pixel-based feature methods, such as FAST, SIFT, SURT, ORB, and BRIEF among others, that are still widely applied and demonstrating competitive performance in visual tasks [10]. FAST (Features from Accelerated Segment Test) was proposed in [11], and is still one of the most popular feature extractors till now. BRIEF (Binary Robust Independent Elementary Features) descriptor [12] uses a binary-string based feature description and a hamming-distance based matching. ORB was proposed by Rublee *et al.*[13], and is still widely adopted in many state-of-the-art computational systems. It utilizes the advantages both from FAST and BRIEF to present a feature extraction and matching strategy with the robustness to scales and in-plane rotations. BRISK (Binary Robust Invariant Scalable Keypoints) [14] was another feature method built upon FAST and BRIEF. It was achieved by a different sampling pattern from ORB for more robustness. More recently, Ma *et al.*[15] presented a Feature-guided Gaussian mixture model (FG-GMM) for image matching. It utilized SIFT [16] as the feature detector and FG-GMM as the matcher. Encouraging performance was concluded in their experiments compared with a range of existing methods. However, all above pixel-based feature methods can suffer from low qualities and low quantities in the results when dealing with low-visibility images, which can significantly undermine the performance in their applications. For example, the applications of the image registration through pixel-based feature extraction and matching were discussed in [17, 18]. Their experiments showed that the conventional pixel-based methods such as SIFT [16] and SURF [19] can easily fail when applied with low-visibilities.

Patch-based approaches. Meanwhile, higher-level solutions, such as the one based on image patches, potentially have higher performance in those scenarios since it analyses a larger pixel collection rather than a single or only few pixels for feature description. The image patches can be obtained by either regular or irregular image grid. One of the best known irregular image grids is achieved by superpixel. Recently, superpixel-based feature modelling draws many attentions. However, most of the existing solutions are only dedicated for rectified stereo matching with fine visual conditions. They follow a fixed search path for each feature correspondence, which is the horizontal line in the images. They are also hard to achieve satisfying results for the low-visibility videos. For example, Sunok *et al.*[20] presented a superpixel-based feature augmentation for optimization-based stereo matching. They imposed confidence spatial consistency on the confidence estimation for extracted superpixels. However, their method was only designed for rectified stereo images with fine visibilities, where the search path is only along the scanning line. A patch matching method for consecutive frames based on temporal superpixels was presented in [21]. It utilized a coarse-to-fine strategy for superpixel generation. Despite the encouraging performance, their method was only designed for fine visibility. Moreover, they only considered 200 superpixels in a 240 × 160 image. Their superpixel generation thus can be more computationally expensive in practical applications. Bian *et al.*[22] presented a hybrid scheme that utilized a grid-based feature matching solution. It employed standard ORB as the feature detector and utilized the neighboring matching in a support region to achieve feature correspondence for images with small rotations. They concluded competitive performance

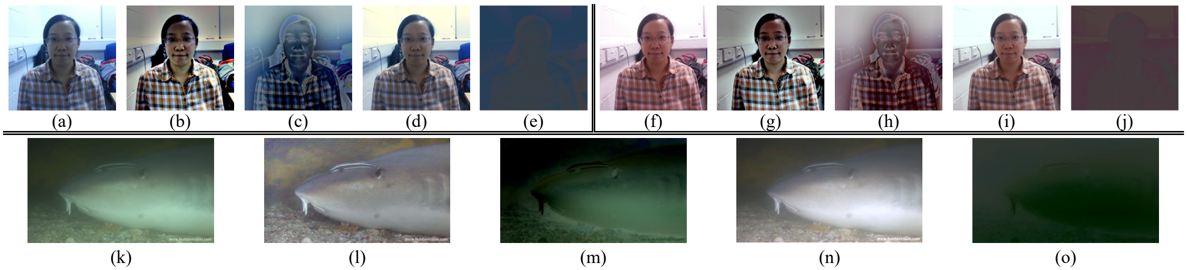


Figure 2: Illumination Alignment (IA). (a) and (f) are of a same scene captured in different lighting conditions, and (k) is from an underwater video. In each block of the figure, IA is applied on the 2^{nd} image. The 3^{rd} image is the difference between the original (1^{st}) and the illumination aligned (2^{nd}) image, which represent the ambient illumination variations between two frames. White balance correction is applied on the 4^{th} image. The 5^{th} image is the difference between the 1^{st} and the 4^{th} image. It is noticed from top two blocks that the illuminations are better aligned using IA (b and g) than using classic methods (d and i).

compared with several existing methods. However, their approach was only designed for images in clear visibilities.

Learning-based approaches. In the past few years, deep learning-based approaches reveal their advantages in many areas. Recently, the solutions from this category also try to step their feet into the feature extraction and matching field. For example, the MatchNet proposed by Han *et al.*[23] tasted the feature matching using convolutional neural networks (CNNs) for the first time in its kind. Their model mapped the patches in the images to feature representations. A metric network then mapped a pair of features to a similarity for matching. Their patch-based matching targeted at stereo matching, and encouraging performance were concluded in their paper. Simo-Serra *et al.*[24] presented a deep convolutional approach for feature description. It utilized a pair of CNNs sharing the same weights for patch-wise feature extraction using a Siamese network structure. Promising results were demonstrated. However, their model only took the fix-sized patches, such as 64 × 64, as the input to extract descriptors. More recently, Gao *et al.*[25] presented a Siamese Attentional Keypoint Network for visual tracking problem and concluded high performance with fine visibility conditions. A new lightweight hourglass network was proposed based on Siamese architecture. It also utilized the dual-attention mechanism, which explored spatial and channel for a better semantic saliency tracking. Based on Long Short-Term Memory (LSTM), Gao *et al.*[26] also presented a hierarchical attentional model for visual tracking. Their approach took full advantage of informative geometries and semantics for the task along with the adaptation of variations in the target appearance. Experiments demonstrated high performance of their method with data in clear vision. Xu *et al.*[10] presented a scene-adaptive descriptor named SAFT for visual SLAM related applications. It used FAST [11] as the feature extractor and a Siamese-based structure as similarity calculator. Their method still designed for images with clear visibility.

Despite the encouraging performance demonstrated, the practicalities of those methods still needs enhancements. The performance highly relies on the preparations of the training data. If the target has less or even no correlation with the data distribution of the training samples, the performance can be impaired. The feature modelling is a fundamental component for many computing systems. The generality of the method should be high enough in order to cover the applications in some of the unexpected situations. This is unfortunately difficult to accomplish with existing deep learning-based approaches. Moreover, for some domains, especially for low visibility situations, the structured training samples of those low-visibility image are much less common, which also present an obstruction to establish a deep learning-based feature matching model for such scenario.

The proposed solution. To address those problems, inspired by human knowledge of perceiving features in limited visual conditions, a SPF-based solution is proposed for feature matching between consecutive frames from low visibility videos. The frame pairs are not necessarily to be the rectified stereo image pairs. The proposed method uses a cycle-labelling strategy for matching with a pair of pre- and post-processing, which provides additional robustness for applications in limited visual conditions.

3. Adaptive Illumination Alignment

The degradations in the visual data can negatively affect the performance of the computational systems. Intuitively, the proposed method uses a pre-processing directly on the input data before the feature representation process. A color constancy inspired Illumination Alignment (IA) is adopted to alleviate the illumination variation across frames, which exists since the ambient illuminations are never evenly distributed. For example, Fig. 2 (a) and (f) are captured in a same scene, yet, the illuminations can vary. The color constancy is achieved automatically in human vision system, which enable us to distinguish true colors in various visual conditions, such as low light, changes in illuminations and many others. This is helpful to perceive constant colors across frames. The first computational model for color constancy is Retinex [27]. A Multi-Scaled Retinex with Color Restoration (MSRCR) was then presented [28] with a color correction step. It can suppress the color desaturation caused by Retinex. Accordingly, the observed image can be decomposed as:

$$\log[I(x, y)] = \log[L(x, y)] + \log[R(x, y)] \tag{1}$$

where $I(x, y)$ denotes the pixel observation at a position of (x, y). L is the luminance. It varies under different illuminations. R refers to the reflectance standing for the true color of a scene, which we need to extract. L can be approximated by a Gaussian convolution on I according to Retinex. Multi-Scaled Retinex (MSR) [28] considers multiple iterations:

$$\begin{cases} R = \Sigma_{i=1}^{N_{scale}} w_i [\log[I] - \log[H^{(\sigma_i)} * I]] \\ H^{(\sigma_i)} = \frac{1}{2\pi\sigma_i^2} e^{\frac{(x - x_{kernel_{center}})^2 + (y - y_{kernel_{center}})^2}{-2\pi\sigma_i^2}} \end{cases} \tag{2}$$

where w_i is the weight, and $H^{(\sigma_i)}$ is the Gaussian kernel in the i^{th} iteration standing for the different visual spectrum frequency. The sum of all w_i equals 1.0, which makes the outputs natural-looking. According to our previous research [29], three equal weights will be sufficient enough to produce encouraging performance. MSR can handle more complicated illuminative conditions in a wide range of spectrum. For multi-channel images, MSR is performed for each channel individually. A color correction step is then applied to make the image having a tone close to the original one:

$$\begin{cases} C_i = \beta \log[\alpha \times \frac{I^{(i)}(x,y)}{\Sigma_{j \in \{r,g,b\}} I^{(j)}(x,y)}] \\ R_{MSRCR}^{(i)}(x, y) = C_i R^{(i)}(x, y) \end{cases} , i \in \{r, g, b\} \tag{3}$$

where C_i stands for the channel proportion over three channels of $\{r, g, b\}$, which should remain fixed. α and β are used to maintain this channel proportion constancy. A global adjustment consisting of a *Gain* (G) and an *Offset* (b) is utilized to tackle the color shifting:

$$R_{MSRCR}(x, y) = G[R(x, y) + b] \tag{4}$$

IA aims at making two frames demonstrate the similar tones. This provides helps for feature similarity measurements in the following stages. The goal of IA is not to restore the exactly true tones of the frames. It targets the tone alignment between two frames since the final objective is to find the feature correspondence between frames. This process is similar when human observes an object in two different illumination environments. The parameters are thus adaptively optimized based on this assumption, which makes the two aligned frames demonstrate a similar texture appearance.

According to our previous studies [29], We find that smaller α and β lead to a lower contrast, and smaller G leads to a lower color dynamic range. For demonstrative purpose, we employ 6.0 and 2.0 as initial values for α and β respectively in our experiments. As the pixel values in each channel range from 0 to 255, we find that the choice of 128 as the initialization for auto *Gain/Offset* (*G/b*) can lead to a higher performance, which coincides with the conclusions from previous studies [30, 31, 32]. Additionally, three scales N_{Scale} corresponding to large, medium and low spectrum frequencies are adopted for the demonstrative purpose in the experiments with equal weights w_i for each scale. The σ_i are initialized using 10, 70 and 260 for three scale respectively.

Fig. 2 demonstrates the comparisons of our IA scheme with the widely used white balance correction (WB). It is noticed that the highlights (specularities) are still presented near the fish belly with WB as shown in Fig. 2(n). It can interfere the image texture analysis in many ways. The same area, on the other hand, is better observed with IA applied as shown in Fig. 2(l). Moreover, our method can greatly eliminate illumination difference between two images, as demonstrated in Fig. 2(c) and (h). The illumination difference however still persists with traditional method employed, as in Fig. 2(d) and (i), which can negatively affect the following feature matching. Our method can better align the variations in image appearance as shown in Fig. 2(b) and (g).

4. Superpixel Flow Generation

The classic superpixel generation [7, 33, 34] is similar to an image over-segmentation. A superpixel is considered as a collection of image pixels having continuous depths. The matched 2D features are supposed to be seated in the corresponding matched superpixels. A robust superpixel generation can thus be helpful for feature matching when pixel-based image saliencies are weakened. To this end, we propose a new superpixel algorithm called Superpixel Flow (SPF).

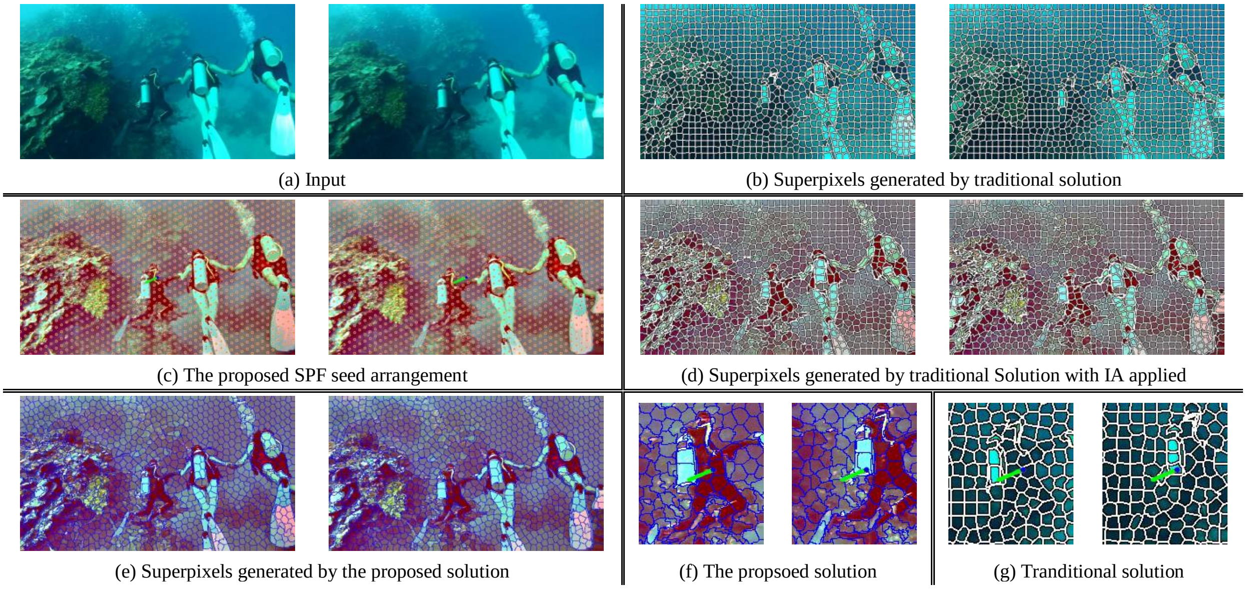
The generation of the SPF is achieved from a set of seeds over the image. The pixels are clustered about these seeds according to two types of the distances, a color distance d_{color} in LAB color space and a spatial distance $d_{spatial}$. They are calculated between each pixel and all its neighboring seeds:

$$\begin{cases} dist_{total} = \sqrt{d_{color}} + \mu \times \sqrt{d_{spatial}} \\ \mu = ((\frac{\Delta x}{\varphi_{compactness}}) \times (\frac{\Delta y}{\varphi_{compactness}}))^{-1} \end{cases} \tag{5}$$

where μ adjusts the proportion between these two distances. μ is controlled by a hyperparameter $\varphi_{compactness}$, which indicates how spatially compact the generated superpixel is. For demonstrative purpose, a $\varphi_{compactness}$ of 15 is adopted in the experiments in this paper. Δx and Δy are the initial seed intervals along x- and y-axis respectively. By finding a smallest $dist_{total}$, the assignment of each image pixel to a certain superpixel block (seed) can be determined.

In practice, the superpixels are generated differently with different seed arrangements. The traditional superpixel generations employ a fixed seed arrangement for every image. It makes the seeds seated at exactly the same coordinates in the image plane for different frames. However, the content moves across frames. Thus, the superpixel generated from those seeds can differ across frames, which leads to two very different sets of superpixels, as shown in Fig. 3(b). This difference adds more difficulties to the matching process.

To tackle this, the proposed method utilizes a new seed arrangement scheme for superpixel generation. It firstly estimates the content motion F_{cm} between two consecutive frames, and then spreads the seeds for those two frames based on this F_{cm}. F_{cm} is calculated by k-means clustering on the top of the optical flow field across two frames. Among many other methods, a dominated optical flow cluster exported by



(a) Input

(b) Superpixels generated by traditional solution

(c) The proposed SPF seed arrangement

(d) Superpixels generated by traditional Solution with IA applied

(e) Superpixels generated by the proposed solution

(f) The propsoed solution

(g) Tranditional solution

Figure 3: The the comparisons between the proposed SPF and the traditional superpixel. (a) is a frame pair from a diving video; (c) demonstrates the SPF seeds arranged by the proposed method with IA applied; (e) shows the generated SPF by the seeds in (c); (b) illustrates the superpixel generated by SLIC; (d) is obtained by SLIC with IA applied; (f) and (g) are four magnified regions (of a diver) in (d) and (b). It can be noticed that, compared to the traditional method, the proposed SPF can better describe a stable feature collection when image content moves. Those stable features demonstrate similar visual appearances across frames, which is helpful for matching. Even with IA applied, the traditional superpixel is still unable to retrive similar visual blocks between two frame when content changes. The green lines in (c), (f) and (g) are the average content motion vector F_{cm}.

k-means clustering process can be used as a stable F_{cm} from noisy data. The seed arrangement takes into account F_{cm} when shifting the seeds:

$$\begin{cases} \Delta x = \frac{\lambda |x_{cm}|}{\sqrt{x_{cm}^2 + y_{cm}^2}}, \Delta y = \frac{\lambda |y_{cm}|}{\sqrt{x_{cm}^2 + y_{cm}^2}} \\ \lambda = \sqrt{height_{img} \times \frac{width_{img}}{N_{sp}}} + 0.5 \end{cases} \tag{6}$$

where Δx and Δy are the same as in (5). x_{cm} and y_{cm} are x and y components of F_{cm}. N_{sp} is the superpixel number in an image. N_{sp} affects the superpixel block size. If N_{sp} is too small, the superpixel will be too large to describe a perceptive region. If N_{sp} is too large, the superpixel will be too small to be useful. For the demonstrative purpose, a N_{sp} of 1200 is utilized as system initialization in the experiments illustrated in this paper. A superpixel seed field can be achieved by (6) as shown in Fig. 3(c left). As for the second frame, the seeds shift along F_{cm}, as shown in Fig. 3(c right). With the seeds arranged over the image planes, the superpixel can be finally generated by clustering each pixel about the seeds according to the distance formulated in (5). Fig. 3(e) demonstrates the superpixels generated from these two seed sets. Compared to the superpixel blocks generated by traditional methods, the proposed SPF patches are much easier to be matched across frames.

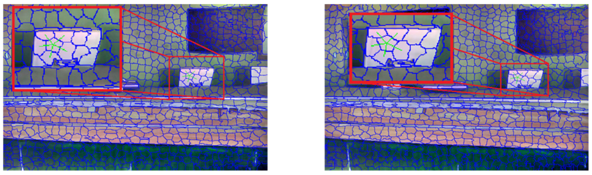
As shown in Fig. 3, since the seed arrangement considers the content movement between two images, the superpixel difference across frames is minimized. It keeps the geometric relationship between the seeds and image pixels remaining as stable as possible when shifting frames. Additionally, the square roots in (5) make the SPF generation more sensitive in short spatial range with low visibility conditions.

5. Multi-Labelling For Matching

The goal of matching is to find two most similar superpixels as a matched feature pair in terms of color, size, and neighbor-relationship. F_{cm} can also contribute to the matching. The matching is achieved by a cycle-labelling strategy. In the first round, the superpixel in the second frame is treated as a Label (feature z_{lbl}), and the superpixel from the first frame is treated as an Index (feature z_{idx}). The matching process assigns Label z_{lbl} to each Index z_{idx}. In the second round, the roles between z_{idx} and z_{lbl} are swapped, which indicates that superpixel in the first frame is treated as a z_{lbl} assigned to the superpixel z_{idx} from the second frame. Only the cycle-matched superpixel pairs in two rounds are selected as the final matching results. MRF is used for labelling process due to its advantage in dealing with neighbor-related problems. When two superpixel are matched, there is a very high chance that their neighbors are also matched.

The energy function used in the labelling process consists of a data cost $Cost_D$ and a smoothness cost $Cost_S$, as shown in (7). $D^{(i)}$ refers to an energy when an index superpixel $z_{idx}^{(i)}$ is assigned with a label superpixel $z_{lbl}^{(i)}$. A better matching leads to a less $D^{(i)}$. $S^{(i,j)}$ indicates the error penalties of label assignment for a pair of neighboring superpixels.

$$\begin{cases} Energy = Cost_D + \delta Cost_S \\ Cost_D = \Sigma_{i=1}^{N_{sp}} D^{(i)}(z_{idx}^{(i)}, z_{lbl}^{(i)}) \\ Cost_S = \Sigma_{i=1}^{N_{sp}} \Sigma_{j \in nbr(i)} S^{(i,j)}(z_{idx}^{(i)}, z_{idx}^{(j)}, z_{lbl}^{(i)}, z_{lbl}^{(j)}) \end{cases} \tag{7}$$



(a) (b)

Figure 4: The demonstration of the similarities in neighboring structure of matched superpixels. For example, green lines in (a) and (b) indicate that for the two successfully matched superpixels between (a) and (b), they both have five neighboring superpixels with similar neighboring superpixel distances.

In human vision system, the cues such as colors, locations, and neighbors contribute more than others for feature extractions. Accordingly, the matched superpixels can share similarities in colors, block sizes, and neighbor structures. The shifting directions of the image contents across frames contribute to the matching as well. For example, in stereo matching, the image contents shift in a direction parallel with the scan lines, which restricts the matching along the horizontal lines. $D^{(i)}(z_{idx}^{(i)}, z_{lbl}^{(i)})$ is thus formed up as follows:

1. The average color of the superpixel block from CIELAB color space are used as a color cost. CIELAB defines colors using a three-dimensional look-up table. Each color is represented by a tuple (l, a, b) for three channels. (l) is for lightness. (a, b) corresponds two axis values of the color table. Since IA is applied in prior, and the lightness variations are intended to be ignored, only a two-value tuple (a, b) is used in color cost with a l_2-normal distance:

$$D_{color}^{(i)} = ||(a,b)_{idx}^{(i)} - (a,b)_{lbl}^{(i)}||^2 \tag{8}$$

2. Size cost is described using the pixel number of a superpixel block, as shown in (9). $N_{idx\_pixel}^{(i)}$ and $N_{lbl\_pixel}^{(i)}$ are the pixel numbers for superpixel z_{idx} and z_{lbl} respectively. $||\cdot||^2$ refers to a l_2-normal distance.

$$D_{size}^{(i)} = ||N_{idx\_pixel}^{(i)} - N_{lbl\_pixel}^{(i)}||^2 \tag{9}$$

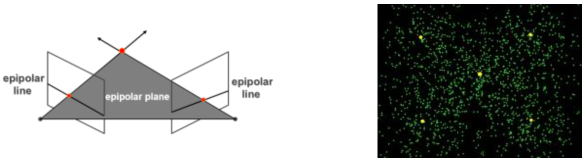
3. The penalties on the spatial information are also considered. There is a high chance that the matched superpixels are positioned along a line parallel with F_{cm}. $D_{spatial}^{(i)}$ is thus constructed based on a cosine angle as shown in (10). dx and dy are the distances between z_{idx} and z_{lbl} along x- and y-axis respectively in image coordinate system. x_{cm} and y_{cm} are two perpendicular components of F_{cm}.

$$D_{spatial}^{(i)} = 1 - \frac{dx \times x_{cm} + dy \times y_{cm}}{\sqrt{dx^2 + dy^2} \times \sqrt{x_{cm}^2 + y_{cm}^2}} \tag{10}$$

4. The neighbor structure is also considered for $D^{(i)}$, as shown in (11), $N_{idx\_nbr}^{(i)}$ and $N_{lbl\_nbr}^{(i)}$ are the numbers of neighboring superpixels of z_{idx} and z_{lbl} respectively. Matched superpixels share similar neighboring structures (Fig. 4).

$$D_{nbr}^{(i)} = ||N_{idx\_nbr}^{(i)} - N_{lbl\_nbr}^{(i)}||^2 \tag{11}$$

The weighted sum of the costs forms up the data cost $D^{(i)}$, as in (12). An additional label is added to the system indicating that no match is found.



(a) Epipolar constraint (b) Guassian mixture distribution

Figure 5: (a) The epipolar constraint, which indicate the matched features should be seated on a same epipolar plane; (b) The Gaussian mixture distribution for Randomly-down-sampling. The yellow dots are the distribution expectations.

$$D^{(i)} = \Sigma_{type \in \{color,size,spatial,nbr\}} w_{type} D_{type}^{(i)} \tag{12}$$

Smoothness cost brings penalties to a pair of label assignments for two neighboring superpixels, as shown in (7). For consecutive frame pair, if two neighboring superpixels $z_{lbl}^{(i)}$ and $z_{lbl}^{(j)}$ from one frame are matched with two superpixels $z_{idx}^{(i)}$ and $z_{idx}^{(j)}$ respectively in the other frame, $z_{lbl}^{(i)}$ and $z_{lbl}^{(j)}$ should have a similar distance between $z_{idx}^{(i)}$ and $z_{idx}^{(j)}$. Based on this assumption, $S^{(i,j)}$ provides additional constraints to enhance the matching accuracy. As shown in (13), $dist(\cdot,\cdot)$ gives a spatial distance, and $||\cdot||^2$ is a $l2$-norm.

$$S^{(i,j)} = ||dist(z_{idx}^{(i)}, z_{idx}^{(j)}) - dist(z_{lbl}^{(i)}, z_{lbl}^{(j)})||^2 \tag{13}$$

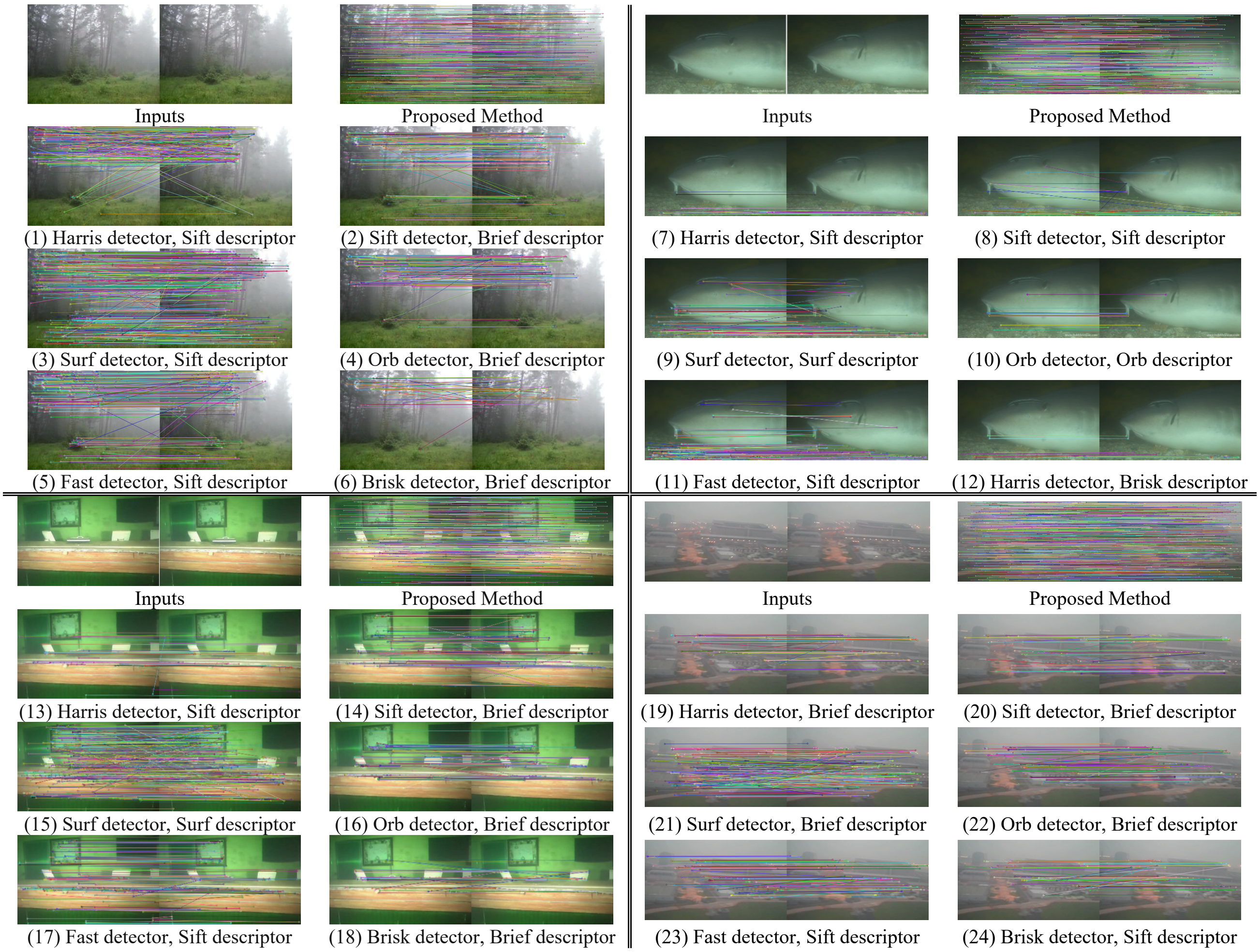
The matching is obtained by minimizing the energy functions. To reduce the processing time, parallel dynamic graph cuts [35] is utilized to perform the energy minimization.

6. Outlier Removal

The proposed outlier removal strategy is based on a RDS (Randomly Down-Sampling) process combined with the uti-

Algorithm 1: RDS-based Outlier Removal

```
Input: The initially matched feature set $M^{(0)}$
Output: The optimal matching result $M_{filtered}^{(k)}$
Estimate fundamental matrix $\mathbf{F}^{(0)}$ based on $M^{(0)}$
Filter $M^{(0)}$ using $\mathbf{F}^{(0)}$ for a refinement $M_{filtered}^{(0)}$
Initialize iteration number $k = 1$
while Outlier number >threshold (10% of all pairs for example) and k <threshold (10 for example) do
    Randomly down-sample $M_{filtered}^{(k-1)}$ for a subset $M_{RDS}^{(k)}$ based on Gaussian mixture distribution
    Estimate a new fundamental matrix $\mathbf{F}^{(k)}$ from set $M_{RDS}^{(k)}$ in the $k^{th}$ iteration. Filter the matched pairs in $M_{RDS}^{(k)}$ by $\mathbf{F}^{(k)}$ for a further refinement $M_{filtered}^{(k)}$ with fewer outliers
    $k$++
```



Inputs

Proposed Method

(1) Harris detector, Sift descriptor

(2) Sift detector, Brief descriptor

(3) Surf detector, Sift descriptor

(4) Orb detector, Brief descriptor

(5) Fast detector, Sift descriptor

(6) Brisk detector, Brief descriptor

Inputs

Proposed Method

(7) Harris detector, Sift descriptor

(8) Sift detector, Sift descriptor

(9) Surf detector, Surf descriptor

(10) Orb detector, Orb descriptor

(11) Fast detector, Sift descriptor

(12) Harris detector, Brisk descriptor

Inputs

Proposed Method

(13) Harris detector, Sift descriptor

(14) Sift detector, Brief descriptor

(15) Surf detector, Surf descriptor

(16) Orb detector, Brief descriptor

(17) Fast detector, Sift descriptor

(18) Brisk detector, Brief descriptor

Inputs

Proposed Method

(19) Harris detector, Brief descriptor

(20) Sift detector, Brief descriptor

(21) Surf detector, Brief descriptor

(22) Orb detector, Brief descriptor

(23) Fast detector, Sift descriptor

(24) Brisk detector, Sift descriptor

Figure 6: Samples of the matching results of the proposed method compared with other most popular methods up to now. Those testing samples are extracted from videos in low visibility [36, 37, 38, 39]. For each block, we visually compare the proposed method with other six standard feature methods composed of different feature detectors and descriptors. They demonstrate that the proposed method can achieve high matching quality and high matching quantity simultaneously while standard methods have difficulties dealing with low-visibilities.

lization of the fundamental matrix **F**. The correctly matched feature pairs should satisfy the model described by this **F**.

The two matched 2D points from those two images are presumably the projections from one 3D point in the space. These two 2D points and their originating 3D point form up a epipolar plane, as shown in Fig. 5(a). The intersections of this plane with two images are the epipolar lines. They are described by **F**. For example, if a 3D point in the space has two projections on two images as $\mathbf{x}'$ and $\mathbf{x}$ respectively, then:

$$\mathbf{x}'^{\mathbf{T}}\mathbf{F}\mathbf{x} = \mathbf{0} \tag{14}$$

This enables **F** as a filter to refine the matching result. In this paper, the superpixel seeds are adaptively adjusted during the superpixel clustering process and are then used as the feature centers. The filtering is an optimization process, which finds the matched features by least errors. However, the performance of this filter is still limited since **F** is initially estimated base on the matched feature candidates that need to be refined. To tackle this problem, a outlier filtering based on an iterated RDS process is proposed as in Alg. 1.

Studies show that the outliers frequently occur around the image borders due to the occlusions or the camera distortion. A Gaussian mixture distribution enhanced RDS helps to collect more pairs near the image center. In our implementation, the mean of the main Gaussian distribution with the standard deviation of 70 is seated at the image center. Other four additional Gaussian distributions seated by the image corners with a larger standard deviation of 80 . They are used to sample more data, as shown in Fig. 5(b).

7. Experiments

7.1. Experiment methods

To demonstrate the matching performance, the experiments use frame pairs extracted from a range of the unconstrained low-visibility videos. They are publicly accessible on websites. The testing data are all scaled to VGA sizes for the comparisons. Our method is compared with the state-of-the-art methods, which are still the most popular feature

Table 1
Feature Matching Comparisons.

Feature Matching Methods		**Average Accuracy Rate** with [Number of Matched Pairs] in square brackets							
		Experiment (a)		Experiment (b)		Experiment (c)		Experiment (d)	
Proposed Method		**0.98** [565]		**0.99** [461]		**0.98** [475]		**0.98** [719]	
Without IA		0.89 [18]		0.87 [16]		0.81 [23]		0.97 [61]	
Without Outlier Removal		0.71 [867]		0.61 [1511]		0.52 [731]		0.70 [1069]	
Standard Methods		Without IA	With IA	Without IA	With IA	Without IA	With IA	Without IA	With IA
Detectors	Descriptors								
Harris	Sift	0.68 [566]	0.62 [544]	0.95 [120]	0.91 [161]	0.94 [96]	0.96 [49]	0.64 [82]	0.78 [91]
	Surf	0.04 [566]	0.03 [544]	0.78 [120]	0.62 [161]	0.30 [96]	0.27 [49]	0.17 [82]	0.13 [91]
	Brief	0.62 [410]	0.62 [420]	0.00 [3]	0.83 [18]	0.92 [91]	0.95 [39]	0.85 [77]	0.83 [84]
	Orb	0.53 [833]	0.53 [408]	0.00 [3]	0.57 [14]	0.91 [91]	0.89 [38]	0.67 [76]	0.86 [83]
	Brisk	0.48 [507]	0.41 [490]	0.94 [53]	0.84 [76]	0.90 [94]	0.91 [46]	0.63 [79]	0.69 [87]
	Freak	0.08 [449]	0.47 [440]	0.00 [4]	0.41 [24]	0.10 [92]	0.22 [41]	0.20 [79]	0.18 [87]
Sift	Sift	0.66 [312]	0.46 [1302]	0.73 [73]	0.59 [348]	0.78 [196]	0.59 [368]	0.38 [91]	0.57 [227]
	Surf	0.31 [312]	0.04 [1302]	0.52 [73]	0.46 [348]	0.63 [196]	0.43 [368]	0.26 [91]	0.45 [227]
	Brief	0.81 [242]	0.58 [1045]	0.00 [9]	0.60 [151]	0.81 [179]	0.72 [318]	0.79 [88]	0.71 [217]
	Orb	0.00 [0]	0.00 [0]	0.00 [0]	0.00 [0]	0.00 [0]	0.00 [0]	0.00 [0]	0.00 [0]
	Brisk	0.59 [280]	0.30 [1186]	0.44 [29]	0.56 [240]	0.66 [186]	0.54 [339]	0.31 [89]	0.00 [220]
	Freak	0.07 [238]	0.03 [1040]	0.00 [12]	0.12 [177]	0.08 [175]	0.06 [315]	0.24 [87]	0.09 [213]
Surf	Sift	0.68 [927]	0.58 [1842]	0.74 [158]	0.61 [441]	0.73 [737]	0.65 [1291]	0.72 [334]	0.70 [749]
	Surf	0.62 [927]	0.47 [1842]	0.77 [158]	0.65 [441]	0.72 [737]	0.64 [1291]	0.74 [334]	0.68 [749]
	Brief	0.71 [848]	0.56 [1658]	0.71 [105]	0.67 [343]	0.74 [709]	0.67 [1192]	0.76 [329]	0.73 [721]
	Orb	0.23 [825]	0.24 [1626]	0.66 [96]	0.48 [322]	0.57 [704]	0.42 [1176]	0.62 [327]	0.49 [710]
	Brisk	0.59 [756]	0.45 [1519]	0.69 [81]	0.61 [282]	0.73 [641]	0.60 [1052]	0.72 [280]	0.64 [632]
	Freak	0.03 [482]	0.03 [1035]	0.31 [29]	0.11 [129]	0.05 [464]	0.04 [712]	0.07 [215]	0.00 [462]
Orb	Sift	0.74 [481]	0.65 [500]	0.90 [21]	0.62 [292]	0.77 [496]	0.78 [500]	0.78 [350]	0.70 [498]
	Surf	0.82 [481]	0.75 [500]	0.95 [21]	0.76 [292]	0.89 [496]	0.87 [500]	0.88 [350]	0.85 [498]
	Brief	0.86 [481]	0.88 [500]	0.85 [21]	0.72 [292]	0.92 [496]	0.90 [500]	0.92 [350]	0.89 [498]
	Orb	0.63 [481]	0.56 [500]	0.95 [21]	0.59 [292]	0.76 [496]	0.77 [500]	0.78 [350]	0.66 [498]
	Brisk	0.83 [265]	0.75 [318]	0.53 [13]	0.67 [139]	0.85 [424]	0.89 [403]	0.91 [283]	0.80 [407]
	Freak	0.21 [52]	0.21 [96]	0.00 [5]	0.42 [31]	0.08 [212]	0.10 [193]	0.26 [101]	0.17 [144]
Fast	Sift	0.78 [634]	0.71 [2512]	0.82 [369]	0.85 [1056]	0.85 [332]	0.80 [830]	0.85 [123]	0.75 [367]
	Surf	0.06 [634]	0.02 [2512]	0.57 [369]	0.43 [1056]	0.55 [332]	0.20 [830]	0.17 [123]	0.25 [367]
	Brief	0.65 [501]	0.50 [1870]	0.61 [102]	0.71 [486]	0.81 [300]	0.72 [633]	0.78 [114]	0.70 [344]
	Orb	0.55 [479]	0.34 [1799]	0.66 [89]	0.63 [445]	0.82 [296]	0.67 [607]	0.67 [113]	0.59 [342]
	Brisk	0.48 [589]	0.28 [2276]	0.70 [218]	0.67 [772]	0.75 [319]	0.56 [747]	0.63 [118]	0.57 [358]
	Freak	0.06 [530]	0.02 [1983]	0.14 [126]	0.08 [558]	0.04 [307]	0.04 [681]	0.10 [116]	0.09 [350]
Brisk	Sift	0.33 [53]	0.26 [155]	0.00 [1]	0.39 [36]	0.42 [59]	0.53 [93]	0.76 [76]	0.70 [115]
	Surf	0.35 [53]	0.08 [155]	0.00 [1]	0.22 [36]	0.45 [59]	0.26 [93]	0.28 [76]	0.27 [115]
	Brief	0.48 [52]	0.24 [153]	0.00 [1]	0.58 [33]	0.62 [58]	0.53 [91]	0.72 [74]	0.69 [113]
	Orb	0.19 [52]	0.12 [150]	0.00 [1]	0.27 [30]	0.39 [58]	0.35 [91]	0.16 [73]	0.45 [111]
	Brisk	0.43 [53]	0.11 [155]	0.00 [1]	0.36 [36]	0.33 [59]	0.45 [93]	0.56 [76]	0.52 [115]
	Freak	0.37 [27]	0.13 [80]	0.00 [1]	0.37 [19]	0.27 [43]	0.14 [71]	0.19 [67]	0.14 [98]
CenSurE	Sift	0.94 [17]	0.81 [91]	0.00 [0]	0.00 [6]	0.92 [25]	0.75 [87]	0.00 [2]	0.54 [13]
	Surf	0.47 [17]	0.33 [91]	0.00 [0]	0.00 [6]	0.72 [25]	0.52 [87]	0.00 [2]	0.54 [13]
	Brief	0.94 [17]	0.75 [91]	0.00 [0]	0.00 [6]	0.92 [25]	0.54 [87]	0.00 [2]	0.54 [13]
	Orb	0.94 [17]	0.71 [91]	0.00 [0]	0.00 [6]	0.88 [25]	0.64 [87]	0.00 [2]	0.54 [13]
	Brisk	0.47 [17]	0.70 [91]	0.00 [0]	0.00 [6]	0.92 [25]	0.72 [87]	0.00 [2]	0.54 [13]
	Freak	0.94 [17]	0.12 [90]	0.00 [0]	0.00 [6]	0.40 [25]	0.13 [83]	0.00 [2]	0.54 [13]

Note that the numbers of successfully matched pairs by the proposed method are in the unit of superpixel. Each superpixel contains around 250 pixels in average. The number of successfully matched pairs by existing feature approaches are in the unit of pixel. Each frame contains around 230400.

solutions up to now [10]. They are composed of 42 different combinations from 7 standard detectors (HARRIS [41], SIFT [16], SURF [19], ORB [13], FAST [11], BRISK [14], CenSurE [42]) and 6 standard descriptors (SIFT, SURF, ORB, BRIEF, BRISK, FREAK [43]). They are still widely and deeply adopted in various applications, and achieving promising performance in today's industrial tasks. For comparative purpose, the proposed method and the rest existing algorithms are all implemented using C++ on a i7 CPU with a 8GB memory. The computations are based on CPU only.

Since it is very hard to find a low-visibility video dataset dedicated for feature matching researches, the experiments are conducted on a series low-visibility videos collected from a range of open sources, such as publicly available videos from youtube, oceanology originations, and many other sources including the dataset of *Tasmania O'Hara 7* [1] provided by the ACFR's (Australian Centre for Field Robotics) marine robotics group [44]. Those videos are much closer to the real scenarios with limited visual conditions. Samples are shown in Fig. 6. As it can be seen, the images suffer from low contrast, loss of color, noisy and blurring. They lead to degradations in the images, which can compromise the traditional feature matching performance.

The ground truth for the experiments are calculated manually. We annotate about 30 pair of 2D correspondences across testing frames manually, and estimate a fundamental matrix accordingly. This fundamental matrix is then used as a ground truth model to check the matching accuracies.

[1] http://marine.acfr.usyd.edu.au/datasets/

Table 2
Comparisons of the Processing Time for Feature Matching.

Feature Matching Methods		**Processing Time** (Sec)			
		Experiment (a)	Experiment (b)	Experiment (c)	Experiment (d)
Proposed Method		0.036	0.040	0.045	0.039
Standard Detectors [a]	Harris	37.026 [Orb] ~ 45.191 [Sift]	28.017 [Freak] ~ 35.381 [Sift]	27.145 [Sift] ~ 28.213 [Brief]	27.679 [Surf] ~ 33.367 [Sift]
	Sift	1.132 [Brisk] ~ 2.468 [Sift]	0.782 [Brisk] ~ 1.529 [Sift]	1.221 [Brisk] ~ 3.039 [Sift]	0.803 [Brisk] ~ 1.589 [Sift]
	Surf	0.491 [Brisk] ~ 13.008 [Sift]	0.258 [Brief] ~ 2.430 [Sift]	0.529 [Orb] ~ 18.571 [Sift]	0.279 [Brisk] ~ 4.993 [Sift]
	Orb	0.062 [Brisk] ~ 5.210 [Sift]	0.025 [Brisk] ~ 0.714 [Sift]	0.096 [Brisk] ~ 5.187 [Sift]	0.050 [Brisk] ~ 4.114 [Sift]
	Fast	0.057 [Orb] ~ 1.647 [Sift]	0.024 [Orb] ~ 1.064 [Sift]	0.798 [Orb] ~ 10.431 [Sift]	0.015 [Brisk] ~ 0.407 [Sift]
	Brisk	0.051 [Brisk] ~ 0.470 [Sift]	0.028 [Brisk] ~ 0.113 [Freak]	0.118 [Brisk] ~ 2.389 [Sift]	0.036 [Brisk] ~ 0.700 [Sift]
	CenSurE	0.084 [Surf] ~ 0.322 [Sift]	—	0.053 [Brisk] ~ 0.485 [Sift]	0.052 [Brisk] ~ 0.188 [Sift]

[a] The name of the descriptor is in the square brackets right after its processing time.
The experiments are conducted by an i7 CPU on a laptop with VGA sized images.

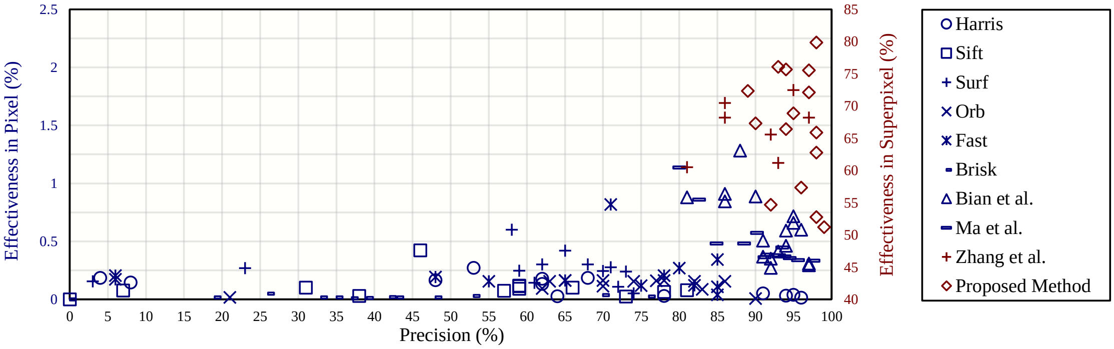


Figure 7: Additional experiment results for the performance comparisons with existing methods (Harris, SIFT, SURF, ORB, FAST, BRISK, Bian *et al.*[22], Ma *et al.*[15], and Zhang *et al.*[40]), in terms of the matching Precision and matching Effectiveness.

7.2. Evaluations and comparisons

Table 1 illustrates the matching accuracies and matched feature numbers by the proposed method and the existing methods. As visually demonstrated in Fig. 6, the accurately matched features by the traditional methods are mainly scattered around the brighter parts of the images, where good visibilities exist. The SPF-based method spreads the matched features all over the image.

In the experiments, the matching accuracies and the matched feature numbers can hardly achieve acceptable performance at the same time by the traditional methods. Moreover, neither of these can be of good results most of the time. For example, The best performance on matched feature number by the traditional methods reaches 927, however, with an accuracy of 68%, as shown in Fig. 6(3). Furthermore, the matched pixel number of this method drops dramatically for other testing data. Its performance lacks robustness. We also test the traditional methods with IA adopted. However, since IA further removes the illuminative variations (pixel saliencies), the performance of the traditional methods can even drop, as shown in Table 1.

The matching accuracies of the SPF-based method are stably occupied by the high performances. Moreover, the matched feature numbers are in the unit of superpixel, which contain around 250 pixels in each under our configuration. Each matched superpixel pair can provide tens or hundreds of matched pixels easily. The results by the existing methods are in the unit of pixel. There are around 900 superpixels in a frame, while more than 230400 pixels in the same image. The matched superpixel number is more efficient.

To fully demonstrate the performance of each components of the proposed method, experiments are also conducted both with and without IA and outlier removal steps. The standard methods highly rely on the clearness of the image saliencies such as the edges or corners. For images in low visibility, those saliencies are heavily weakened due to the image degradations such as blurring, loss of contrast or reduced visual distance. With IA applied, as it can observed in Table 1, the numbers of matched pairs using standard methods significantly increase, which means IA can effectively enhance the feature extraction. However, since IA introduces non-linear color correction on top of the low-visibility images, the feature description calculation that relies on the pixel-level local intensity distribution can be less stable when IA is applied. On the other hand, the proposed method analyses a much larger pixel patch, where illumination aligned color information can positively contribute to the feature matching results.

The computational cost of the proposed method is reasonable based on an Intel I7 quad-core 3.4 GHz mobile CPU on a laptop. As shown in Table 2, the average time consumption of the proposed method is around 40 ms thanks to parallel dynamic processing [35]. The standard methods have a less stable processing time since the feature candidates are quite few in the low-visibility image for some cases. Additionally, the processing time by traditional methods can increase even faster for higher-resolution images.

Fig. 7 presents more performance comparisons in terms of the matching precision (quality) and matching effectiveness (quantity). The matching precision and effectiveness follow the metrics formulated in (15) and (16). The effectiveness is calculated with the unit in Pixel or in Superpixel according to the different implementations of the visual feature methods.

$$\text{Precision} = \frac{\text{Correctly Matched Pairs}}{\text{All Retrived Matching Pairs}} \tag{15}$$

$$\text{Effectiveness} = \frac{\text{Matched Pixel/Superpixel Pairs}}{\text{All Pixel/Superpixel Point in image}} \tag{16}$$

The testing data are provided by ACFR's marine robotics group [44]. It is noticed that most existing methods have difficulties to achieve high matching quality and high matching quantity simultaneously, while our method acquires high performance for both metrics.

8. Conclusion

This paper presents a solution to model features for frame pairs in low-visibility, where standard methods can fail easily. The frame pairs are not necessarily to be the rectified stereo image pairs. This makes the proposed method more flexible than existing ones. SPF is proposed to describe high-level visual features in images. It can significantly enhance the feature perception in poor visual conditions compared to the existing approaches. With the adoption of IA inspired by human vision system for visual sensing in low-visibility, the illuminative variance can be extensively alleviated. The energy functions composed according to human knowledge of comparing visual features, collaborate with a graphcut-based energy minimization scheme to find the feature correspondence across frames. A cycle-labelling strategy produces higher accuracy and robustness in the results. Since the matched superpixels cannot be identically the same, the proposed MRF-based matching aims at identifying the most similar superpixel pairs, which is proven to be a better solution according to the experiments. An outlier removal component follows to further improve the performance. As illustrated in the experiments, the proposed solution demonstrates a competitive performance compared to a range of most popular approaches that are still widely used today in practice.

References

[1] H. Yu, H. Liu, Regression-based facial expression optimization, IEEE Trans. Hum. Mach. Syst. 44 (3) (2014) 386–394.

[2] Y. Zhao, Z. Zheng, Y. Liu, Survey on computational-intelligence-based UAV path planning, Knowledge-Based Systems 158 (2018) 54–64.

[3] F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, F. Herrera, Object Detection Binary Classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance, Knowledge-Based Systems (2020) 105590.

[4] M. C. Chuang, J. N. Hwang, J. H. Ye, S. C. Huang, K. Williams, Underwater Fish Tracking for Moving Cameras Based on Deformable Multiple Kernels, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (9) (2017) 2467–2477.

[5] P. Hu, D. Peng, J. Guo, L. Zhen, Local feature based multi-view discriminant analysis, Knowledge-Based Systems 149 (2018) 34–46.

[6] M. Jian, Q. Qi, H. Yu, J. Dong, C. Cui, X. Nie, H. Zhang, Y. Yin, K.-M. Lam, The extended marine underwater environment database and baseline evaluations, Applied Soft Computing 80 (2019) 425–437.

[7] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, K. Siddiqi, Turbopixels: Fast superpixels using geometric flows, IEEE Trans. Patt. Anal. Mach. Intell. 31 (12) (2009) 2290–2297.

[8] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Patt. Anal. Mach. Intell. 23 (11) (2001) 1222–1239.

[9] S. P. Narote, P. N. Bhujbal, A. S. Narote, D. M. Dhane, A review of recent advances in lane detection and departure warning system, Pattern Recognition 73 (2018) 216–234.

[10] L. Xu, C. Feng, V. R. Kamat, C. C. Menassa, A scene-adaptive descriptor for visual SLAM-based locating applications in built environments, Automation in Construction 112 (2020) 103067.

[11] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: Computer Vision–ECCV 2006, Springer Berlin Heidelberg, 2006, pp. 430–443.

[12] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: binary robust independent elementary features, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 778–792.

[13] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: Proc. ICCV, Vol. 58, IEEE, 2011, pp. 2564–2571.

[14] S. Leutenegger, M. Chli, R. Y. Siegwart, BRISK: Binary robust invariant scalable keypoints, in: Proc. ICCV, IEEE, 2011, pp. 2548–2555.

[15] J. Ma, X. Jiang, J. Jiang, Y. Gao, Feature-guided Gaussian mixture model for image matching, Pattern Recognition 92 (2019) 231–245.

[16] D. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[17] W. Ma, Z. Wen, Y. Wu, L. Jiao, M. Gong, Y. Zheng, L. Liu, Remote Sensing Image Registration With Modified SIFT and Enhanced Feature Matching, IEEE Geoscience and Remote Sensing Letters 14 (1) (2017) 3–7.

[18] S. Zhao, G. Yu, A new image registration algorithm using SDTR, Neurocomputing 234 (2017) 174–184.

[19] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded up robust features, in: Proc. ECCV, 2006, pp. 404–417.

[20] S. Kim, D. Min, S. Kim, K. Sohn, Feature Augmentation for Learning Confidence Measure in Stereo Matching, IEEE Transactions on Image Processing 26 (12) (2017) 6019–6033.

[21] S.-H. Lee, W.-D. Jang, C.-S. Kim, Temporal Superpixels Based on Proximity-Weighted Patch Matching, in: The IEEE International Conference on Computer Vision (ICCV), 2017.

[22] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, M.-M. Cheng, GMS: Grid-based Motion Statistics for Fast, Ultra-Robust Feature Correspondence, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[23] X. Han, T. Leung, Y. Jia, R. Sukthankar, A. C. Berg, MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[24] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, F. Moreno-Noguer, Discriminative Learning of Deep Convolutional Feature Point Descriptors, in: The IEEE International Conference on Computer Vision (ICCV), 2015.

[25] P. Gao, R. Yuan, F. Wang, L. Xiao, H. Fujita, Y. Zhang, Siamese attentional keypoint network for high performance visual tracking, Knowledge-Based Systems (2019) 105448.

[26] P. Gao, Q. Zhang, F. Wang, L. Xiao, H. Fujita, Y. Zhang, Learning reinforced attentional representation for end-to-end visual tracking, Information Sciences 517 (2020) 52–67.

[27] E. H. Land, The retinex, Am. Sci. 52 (2) (1964) 247–264.

[28] S. Parthasarathy, P. Sankaran, An automated multi scale retinex with color restoration for image enhancement, in: Communications (NCC), 2012 National Conference on, IEEE, 2012, pp. 1–5.

[29] S. Zhang, T. Wang, J. Dong, H. Yu, Underwater image enhancement via extended multi-scale retinex, Neurocomputing 245 (2017) 1–9.

[30] M. C. Hanumantharaju, M. Ravishankar, D. R. Rameshbabu, S. Ramachandran, Color Image Enhancement Using Multiscale Retinex

with Modified Color Restoration Technique, in: Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on, 2011, pp. 93–97.

[31] Y. Wang, H. Wang, C. Yin, M. Dai, Biologically inspired image enhancement based on Retinex, Neurocomputing 177 (2016) 373–384.

[32] K. R. Joshi, R. S. Kamathe, Quantification of retinex in enhancement of weather degraded images, in: Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on, IEEE, 2008, pp. 1229–1233.

[33] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, G. Jones, Superpixel lattices, in: Proc. CVPR, 2008, pp. 1–8.

[34] A. Radhakrishna, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels, Dept. School Comput. Commun. Sci., EPFL, Lausanne, Switzerland, Tech. Rep 149300.

[35] M. Yu, S. Shen, Z. Hu, Dynamic Graph Cuts in Parallel, IEEE Transactions on Image Processing 26 (8) (2017) 3775–3788.

[36] Fog in rila mountain 2 (2019).
URL https://www.youtube.com/watch?v=pX9RyH0IubY

[37] bubblevision (2019).
URL http://www.bubblevision.com/underwater-videos/Myanmar-Burma

[38] Costa concordia: New video of the inside of sunken cruise ship (2019).
URL http://www.telegraph.co.uk/news/worldnews/europe/italy/10943328/Costa-Concordia-New-video-of-the-inside-of-sunken-cruise-ship.html

[39] Delta airlines boeing 777-200lr landing in hong kong in low visibility (2019).
URL https://www.youtube.com/watch?v=djbN7Dv0tTo

[40] S. Zhang, J. Dong, H. Yu, Feature Matching for Underwater Image via Superpixel Tracking, in: Automation and Computing (ICAC), 2017 23rd International Conference on, Huddersfield, UK, 2017, pp. 1–5.

[41] K. G. Derpanis, The harris corner detector, Tech. rep., York University (2004).

[42] M. Agrawal, K. Konolige, M. R. Blas, CenSurE: Center surround extremas for realtime feature detection and matching, in: Proc. ECCV, 2008, pp. 102–115.

[43] A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: Fast retina keypoint, in: Proc. CVPR, 2012, pp. 510–517.

[44] M. Bryson, M. Johnson-Roberson, O. Pizarro, S. B. Williams, Colour-consistent structure-from-motion models using underwater imagery, Robotics: Science and Systems VIII (2013) 33.