# Fast Monocular Vision-based Railway Localization for Situations with Varying Speeds

*Chu-Tak Li* and *Wan-Chi Siu*, *Life-FIEEE*

Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

*Abstract*— **This paper presents a railway localization algorithm, using a novel tube of frames concept with key frame based rectification. We extract effective feature patches from key frames with an offline feature-shifts approach to real-time train localization. We get the localization results from both actual calculation of frame matching and estimation based on the previous matches, the temporal information about the train motion. We focus on practical situations in which the train travels at inconstant speeds in different journeys and stops at different locations. The experimental results illustrate that our algorithm can achieve 88.8% precision with 100% recall under an acceptable range of deviation from the ground truth, which outperforms SeqSLAM, a benchmark of localization and mapping algorithm. Moreover, our algorithm is robust to illumination and less sensitive to the length of sequences than the benchmark. We also compare with the modern CNN features based approach. We show that blurring and heavy time cost are two limitations of the CNN. Our algorithm only requires 13.4 ms to process a frame on average using a regular desktop, which is 10 times faster than using the CNN approach and also faster than the benchmark, with the best result of 2.1 times faster on the sample dataset.**

***Keywords— Key frame identification, vehicle detection, autonomous driving, scene recognition and tracking***

## I. Introduction

Self-driving car technology has been a hot topic for years and vehicle localization is one of the crucial components among a self-driving car system [1, 2]. Global Positioning System (GPS) is commonly used in many real-world localization systems but satellite signals usually suffer from reflection and masking because of the concrete buildings, dense trees, etc [1]. Different types of sensors such as inertial sensor and wheel encoder [2] are also solutions to the localization problem. However, some of them are expensive and have their respective limitations. For example, some are susceptible to adverse weather or lighting conditions. Under the circumstances, monocular camera based method could play a key role in vehicle localization systems as its cost-effectiveness and richness of information. For monocular vision-based localization systems [3-12], there could be a number of building blocks such as traffic sign recognition [13, 14], traffic light recognition [15, 16], railway or lane detection [17, 18], vehicle detection [19, 20], visual odometry [21-23], and front car distance estimation [24]. Therefore, the complexity of a comprehensive localization system could be very high, and the time cost of each building block algorithm should be as little as possible so as to make the entire system being real-time but not just a single module.

For all detection, recognition, trajectory estimation tasks [3-23], feature extraction is a fundamental step in the algorithms. Handcrafted features and the corresponding matching or evaluation methods are application-oriented for obtaining satisfactory results under certain assumptions. There is always a trade-off between the time cost of the algorithm and the confidence in making final decisions. For example, [3] studied visual localization across seasons. They proposed to use sequence matching based on multi-feature combination. With local and global descriptors to describe sequences of images, their method has outperformed SeqSLAM [4], a benchmark of visual localization systems, on the Nordland dataset [25]. However, the time cost of their method is expensive. Their method requires 122.6 ms to describe an image, and 2.9463 s to process a match on average. In distance or trajectory estimation tasks [21-24], we can also summarize that feature matching and motion modelling are important steps than can be time-costly.

In terms of the time cost, [5] proposed an appearance-based approach to Visual Simultaneous Localization and Mapping (Visual SLAM) using only low-resolution images. [6] combined BRIEF and Gist descriptors into BRIEF-Gist descriptor for scene recognition using Hamming distance. Milford and Wyeth presented a new approach called SeqSLAM [4] which targets at the best match within each local sequence of frames. Their approach outperforms a successful conventional feature-based SLAM algorithm, FAB-MAP [7], and achieves 100% precision with a recall rate around 60%. [8] is an improved version of [4] in which a patch-based verification process was added for refining the place matches. However, varying speeds situations are still challenging to many single camera based SLAM systems.

[9] proposed a key frame approach to reduce the complexity of localization systems. Discriminative frames can act as key frames for high confidence matching to lock the current position of the vehicle and perform undemanding tracking for less discriminative scenes. [10] evaluated the performance of CNN features for scene recognition with AlexNet [26] and found that AlexNet conv3 layer features give the best performance in localization. [11] used HOG [27] and AlexNet conv3 layer features to evaluate their method and claimed that two types of features could achieve similar performance in some situations using their network flow model. [12] proposed to learn the representative features from

different kinds of features. Their results showed that HOG and CNN features occupied the major portion of the learned representative features, from 83.1% to 95.1%.

In this paper, we focus on a monocular vision-based localization system of railway with varying speeds situations. Our major contribution is that we aim to extract key frames and reference frames (to be defined in Section II. *A. 2)*) in a reference sequence, and use only few effective features to represent each key frame. By using key frames and reference frames, the current position of the train can be provided during other traversals at different speeds or even under extreme lighting conditions.

The rest of this paper is organized as follows. Section II describes our proposed method for key frame identification and scene recognition and tracking. Section III gives experimental results and comparisons of different methods. Finally, Section IV provides a conclusion of the paper.

## II. Proposed Method

### *A. Key Frame Identification*

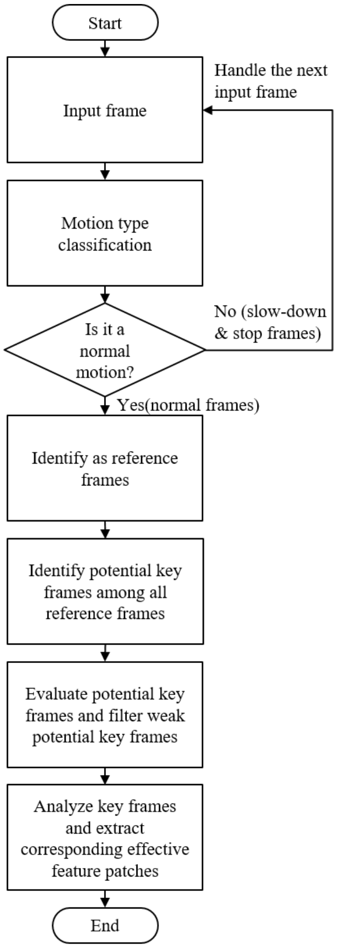

Start
Input frame
Handle the next input frame
Motion type classification
Is it a normal motion?
No (slow-down & stop frames)
Yes(normal frames)
Identify as reference frames
Identify potential key frames among all reference frames
Evaluate potential key frames and filter weak potential key frames
Analyze key frames and extract corresponding effective feature patches
End

Fig. 1. Flowchart of Key Frame Identification

Fig. 1 shows a simplified flowchart for our proposed key frame identification process. Note that the key frame identification is conducted in an offline manner which means that we identify key frames of a route before we perform localization. Selected components in our proposed key frame identification are discussed in detail in the following sub-sections.

#### *1) Motion Type Classification*

The objective of this step is to detect duplicate frames with stopped or dead slow train. Most of the slow-down and stop frames are similar (there are two types of stop frames in the sample dataset). For the 1st type as shown in Fig. 2a, a stop frame offset is required to eliminate the effect of the motion of the vehicle in the front, or otherwise the motion of the frontal vehicle becomes the dominant information in vision-based methods instead of the ego-motion that we want. However, the offset causes a delay in detection the train which moves again in the 2nd type as shown in Fig. 2b. For the 2nd type, the ego-motion is always the dominant information.

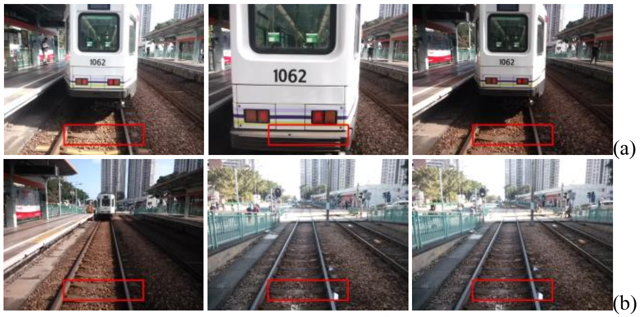


Fig. 2. (a) 1st type: vehicle in the front (b) 2nd type: no vehicle in the front

To perform motion type classification, a simple average pixel difference calculation is applied to the red-bounded region of two consecutive frames, as shown in Fig. 2. The size of the red-bounded region includes 256x64 pixels. We resize it to 64x16 by Lanczos interpolation to reduce the number of pixel comparisons, and perform patch normalization with patch size 8x8 (minus mean and divide by standard deviation) to diminish the influence of changes in illumination. A search window is established for capturing the slow motion of the train (see the yellow-bounded region in Fig. 4b and c) and the operation of average pixel difference calculation is described as:

$$d(\Delta m,\Delta n,I_i,I_{i-1})=\frac{\sum_{n=0}^{H-1}\sum_{m=0}^{W-1}\left|I_i(m+\Delta m,n+\Delta n)-I_{i-1}(m,n)\right|}{W\times H} \tag{1}$$

where $W$ and $H$ are the width and height of the resolution reduced red-bounded region (64x16) respectively. $I_i(m,n)$ is the intensity of the pixel at location $(m,n)$ in the resolution reduced red-bounded region of frame $i$. $\Delta m$ and $\Delta n$ are the shifts in terms of pixels. We look for $\Delta m_{\min}$ and $\Delta n_{\min}$ that minimize the average pixel difference, $d(.)$, between two consecutive frames.

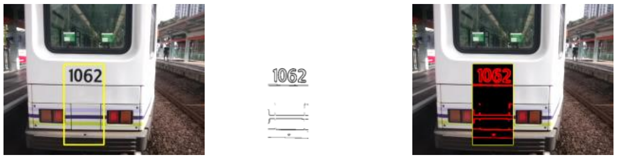
$$(\Delta m_{\min},\Delta n_{\min})=\underset{\Delta m,\Delta n\in[-3,3]}{\arg\min}\ d(\Delta m,\Delta n,I_i,I_{i-1}) \tag{2}$$

where the current search range is ±3 pixels and the minimized $d(.)$ for frame $i$, $d_i$, is the final average pixel difference between frame $i$ and $i$-1.

$$d_i=d(\Delta m_{\min},\Delta n_{\min},I_i,I_{i-1}) \tag{3}$$

If consecutive average pixel differences are smaller than thresholds, $T_{slow\text{-}down}$ and $T_{stop}$, the current frame is classified as slow-down and stop frame respectively. Same outcome could be achieved using wheel odometer or other mechanical sensors. However, in this paper, we aim at a truly single camera based localization system.

We also design a simple operation to classify the type of stop frame after realizing the current frame is a stop frame. We calculate the gradient magnitude of a predefined region, yellow-bounded region, as shown in Fig. 3a. If there is a vehicle in the front, its car number and edges would have large gradient magnitude, as shown in Fig. 3b. We only preserve pixels with large gradient magnitude to form an edge image and compute the corresponding contours and area (see Fig. 3c) using packages provided by OpenCV 2.3.0.



(a) (b) (c)

Fig. 3. (a) Predefined region for stop type classification (b) Pixels with large gradient magnitude (larger than 255) inside the region (c) Detected contours (red lines, area = 6538 pixels)

If the area is larger than a threshold, $T_{car\ in\ front}$, the current stop frame is classified as the 1st type of stop frame and a stop frame offset is applied to compensate for the corresponding effect on stop frame classification.

*2) Reference Frame Identification*

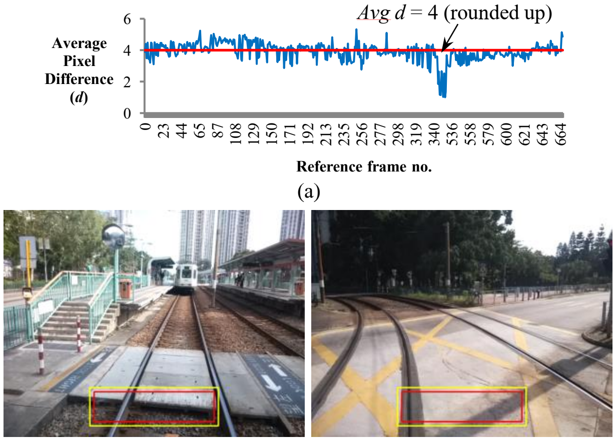
We eliminate all slow-down and stop frames in the reference sequence. The remaining frames are classified as reference frames and key frames are extracted among the reference frames.

*3) Potential Key Frame Identification*

We select potential key frames from the reference frames. We calculate the average of the average pixel differences among all reference frames, *Avg d*, which is rounded up as shown in Eqn. (4).

$$ROUNDUP(Avg\ d = \frac{\sum d_i}{no. of\ reference\ frames}) \tag{4}$$

where $d_i$ is the $d$ of reference frame $i$ from Eqn. (3).



(a)

(b) (c)

Fig. 4. (a) Average pixel difference between each two consecutive frames, *Avg d* = 4 (b) Frame with the largest *d*, Frame 251 (c) Frame with the 2nd largest *d*, Frame 67

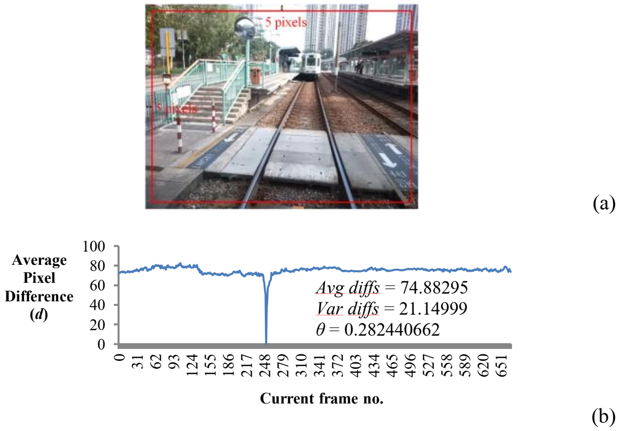
Then, we select reference frames with *d* larger than *Avg d* (see the red line in Fig. 4a) and sort these selected frames according to their values of *d*. Potential key frames are extracted based on a key frame interval, *KFI*, which is in terms of accumulated average pixel difference. Large *d* means that there are larger changes in the red-bounded region (refer to Fig. 2). It usually happens when the train is crossing a train-pedestrian interface or turning, as shown in Fig. 4b and c. These frames have substantially different structures from other frames, hence they are suitable for being key frames.

*4) Potential Key Frame Evaluation*

Let us evaluate the potential key frames and filter weak potential key frames. We compare each potential key frame with all other frames using low-resolution whole frame matching with a search window. The original size of a frame is 640x480 and we preserve 5 pixels from each edge for establishing the search window, as shown in Fig. 5a. The size of the red-bounded region is 630x470. It is downsampled to 64x48 and divided into normalized patches with size of 8x8. We set the search range to ±2 pixels and the operation of frame comparison is similar to Eqn. (1)-(3). The choice of the parameters are decided by extensive experimental work and data observation. For each potential key frame, we compute its average pixel differences with all other frames. Then, we calculate the average and variance of its differences, *Avg diffs* and *Var diffs*.

$$\theta = \frac{Var\ diffs}{Avg\ diffs} \tag{5}$$

Based on Eqn. (5), we compute $\theta$ of each potential key frame. A persuasive key frame has large *Avg diffs* and small *Var diffs*, as shown in Fig. 5b. If $\theta$ of a potential key frame is smaller than a threshold, $T_{good\ key\ frame}$, that potential key frame is identified as a key frame officially and the final key frame list is formed.



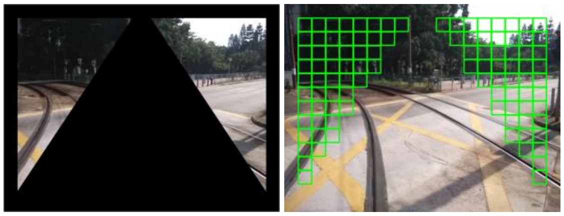
(a)

(b)

Fig. 5. (a) Preserve 5 pixels for establishing a search window (b) Frame 251's average pixel differences with all other frames ($\theta$ = 0.282)

*5) Region of Interest Implementation*

Simple Region of Interest (ROI) is implemented in our proposed method. A triangular mask is applied to each frame so as to conceal part of the railway and the vehicle in the front (see Fig. 6a). The edges are also concealed as features on the edges are highly dependent on the speed of the train. Features on the edges may not be observed in both reference and testing sequences due to the capture intervals of camera. Note that the mask is applied before the HOG feature vector formation [27]. Table I lists the relevant parameters for the formation.



(a) (b)

Fig. 6. (a) Triangular mask for ROI (b) Our ROI - green boxes

TABLE I. Relevant parameters for HOG feature vector formation

| | |
|---|---|
| Input data: | Y component of a frame |
| Frame size: | 640x480 pixels |
| Cell size: | 8x8 pixels |
| Bin size: | 9-bin histogram (unsigned gradient) |
| Block size: | 32x32 pixels (4x4 cells) |
| Feature vector length: | Each block has a 144-length vector |

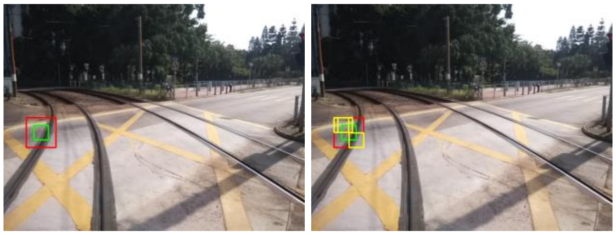
Frame is divided into 300 blocks and there are 100 blocks located at our ROI, as shown in Fig. 6b.

### 6) *Analysis of Key Frame*

Block-to-Block comparison and Cosine distance measure, $C_d$, are employed.

$$C_d = 1 - \cos\omega = 1 - \frac{\mathbf{p}\cdot\mathbf{q}}{\|\mathbf{p}\|\|\mathbf{q}\|} \tag{6}$$

where cos$\omega$ is the Cosine similarity which is defined as the cosine of the angle difference between **p** and **q**.



(a) (b)

Fig. 7. (a) One of the feature blocks (green-bounded) in a frame with the corresponding search window (red-bounded) (b) Some search points within the search window (yellow-bounded)

Each block of a key frame inside ROI is compared with the corresponding block of all other frames. A search window is established for each block so as to compensate for translational shifts in the captured frames and locate the translational shifts invariant features (see Fig. 7). The final Cosine distance between two blocks, $C_{d,\min}$, is the minimum distance within the search window of the studying block.

$$C_{d,\min} = \underset{k\in[0,24]}{\arg\min}\, C_{d,k} \tag{7}$$

where $k$ is the index of the current search point in the search window and $C_{d,k}$ is the Cosine distance of the $k$th search point. The calculation of the Cosine distance of two blocks is also similar to Eqn. (1)-(3). The main difference is that we adopt Cosine distance as a cost function instead of average pixel difference. The range of the search window is ±2 cells and we consider 1 cell shift for each search point. In total, there are 25=(2x2 + 1)$^2$, search points in the search window. For each search point, we regroup different cells as a block and perform L2 contrast normalization [27] again for comparing two blocks so as to consider the possible feature-shifts.

For each key frame, we compute its Cosine distances of each block with all other frames. Then, we calculate the average and variance of its distances, $Avg_{\cos d}$ and $Var_{\cos d}$.

$$\theta_{\cos d} = \frac{Var_{\cos d}}{Avg_{\cos d}} \tag{8}$$

Based on Eqn. (8), we compute $\theta_{\cos d}$ of each block of each key frame inside the ROI. A typical effective feature block has large $Avg_{\cos d}$ and small $Var_{\cos d}$.

$$Eff_{block\ k} = e^{-(\alpha\theta_{\cos d} + \beta(1-Avg_{\cos d}) + \gamma Var_{\cos d})} \tag{9}$$

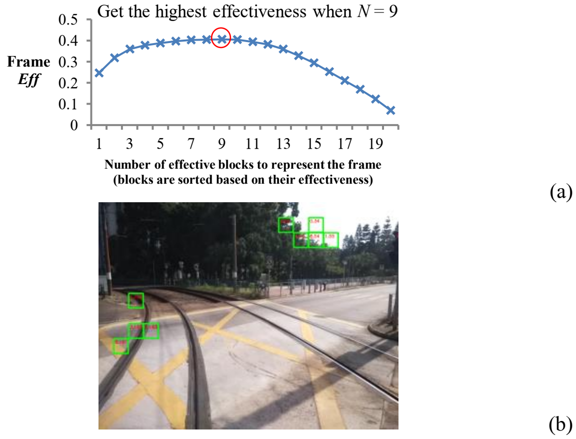
We use Eqn. (9) to calculate the effectiveness of each block of each key frame. $Eff_{block\ k}$ means the effectiveness of the $k$th block in the studying key frame. $\alpha$, $\beta$, and $\gamma$ are the scaling factors to adjust the importance of $\theta_{\cos d}$, $Avg_{\cos d}$, and $Var_{\cos d}$. We sort all blocks of each key frame according to their effectiveness and the effectiveness of each key frame can be computed.

$$Eff_{frame\ i} = \sum^{N} Eff_{block\ k} - N \times penalty \tag{10}$$

where $N$ is the number of effective blocks used to represent the studying key frame and *penalty* is a penalty for use of ineffective feature blocks. We seek for the optimal number of effective feature blocks, $N_0$, which gives the highest effectiveness of the studying key frame.

$$N_0 = \underset{N\in[1,100]}{\arg\max}\, Eff_{frame\ i} \tag{11}$$

As our ROI has at most 100 feature blocks and blocks are sorted based on their effectiveness, we increase $N$ gradually and find $N_0$ to represent the studying key frame effectively, as shown in Fig. 8a.



(a)

(b)

Fig. 8. (a) Relationship between $N$ and $Eff_{frame\ i}$. The highest effectiveness is observed when $N$ = 9 (b) The current key frame and the 9 feature blocks

Fig. 8b shows $N_0$ with 9 effective feature blocks of the current key frame, and we can observe the characteristics of the effective feature blocks. Blocks with special textures or strong alignments such as trees and railway are identified as effective feature blocks. Our approach enables each key frame to be represented by only few but the most effective features which are flexible in size. We do not intentionally extract the artificial features such as traffic signs and railway. We can also observe that some blocks represent the same features. In Fig. 9a, we study the 8 neighboring blocks (red-bounded) of each extracted effective feature block (green-bounded). If the neighboring blocks of an effective feature block are also effective, we combine them to form effective feature patches or otherwise we remove the studying feature block. Fig. 9b shows the final extracted effective feature patches of the current key frame, Frame 67.

The groups of blocks can be regarded as small patterns which are landmarks to be matched, but sometime it is easy to have false positives for small groups which consist of only one or two blocks. By contrast, patches with larger patterns are more expressive to be matched, and they reduce the chance of getting false positives. Each effective feature patch is shifted to form a number of versions. We set the shift range to ±3 cells (3x8=24 pixels) and we shift a patch vertically and/or horizontally by 4 pixels for each version. There are 169=(2x(3x8/4) + 1)$^2$, shifted versions and each version forms a HOG feature vector [27] for the later key frame matching.

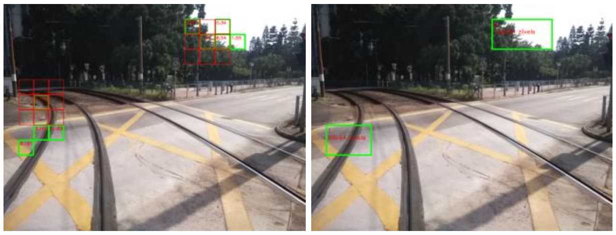

(a) (b)

Fig. 9. (a) Ineffective neighboring blocks, block is removed (left side) Effective neighboring blocks, patch is formed (right-top side) (b) Final effective feature patches (green-bounded) of the current key frame

### B. Scene Recognition and Tracking

A simplified flowchart for our proposed scene recognition and tracking process is shown in Fig. 10. We suppose that key frames and reference frames have already been extracted during the offline learning stage. In online scene recognition and tracking, the first step is to load all the learnt frames into the system. There are two main components namely low-resolution whole frame tracking and key frame matching. Details about these two components are provided as follows.

We assume that the key frames appear chronologically. The idea of key frame matching is that we match some landmarks to lock the current position of the train and then we perform tracking to estimate the position until the next landmark is matched. As a global descriptor performs better in changing conditions and a local descriptor performs is more pose independent, we establish two components and combine them into a real-time localization system.

#### 1) Low-resolution Whole Frame Tracking

For each input frame, we perform Eqn. (1)-(3) to classify its motion type. For slow-down and stop frames, no operation on tracking and matching is performed and the previous result is kept. For normal frame, we perform low-resolution whole frame tracking with tube of frames concept to find the best match to one of the reference frames within a search range and estimate the next match. Tube of frames concept allows us to get the localization result of the current input frame not just based on the instant calculation but also the previous results. This means that we group the current input frame together with the previous frame results and treat them as an input tube. Each input frame is downsampled and patch normalized. We compare the input frame with a set of reference frames using average pixel difference, similar to Eqn. (1), but shifts in frame are not included because of the time cost consideration.

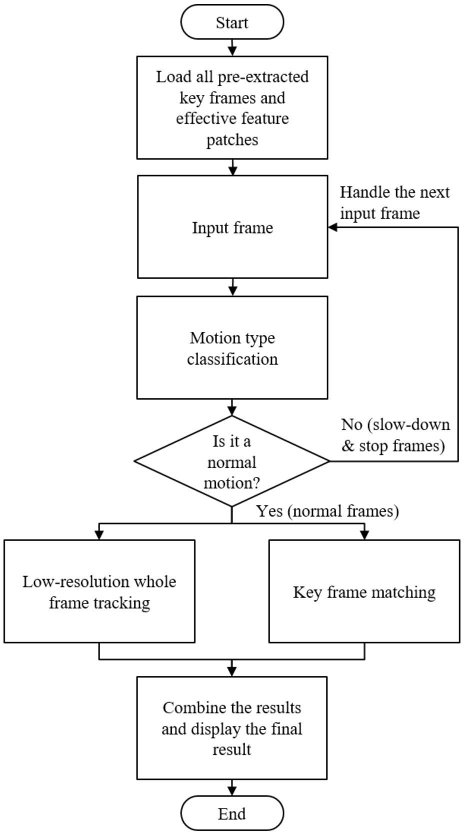

Start
Load all pre-extracted key frames and effective feature patches
Input frame
Handle the next input frame
Motion type classification
Is it a normal motion?
No (slow-down & stop frames)
Yes (normal frames)
Low-resolution whole frame tracking
Key frame matching
Combine the results and display the final result
End

Fig. 10. Flowchart of Scene Recognition and Tracking

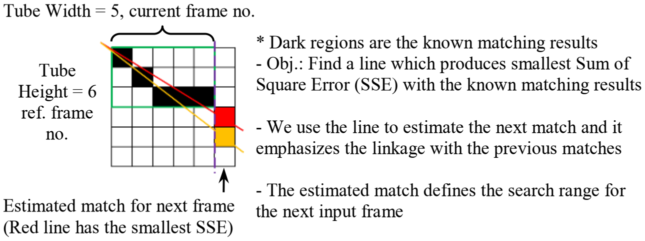

Tube Width = 5, current frame no.
Tube Height = 6 ref. frame no.
* Dark regions are the known matching results
- Obj.: Find a line which produces smallest Sum of Square Error (SSE) with the known matching results
- We use the line to estimate the next match and it emphasizes the linkage with the previous matches
- The estimated match defines the search range for the next input frame
Estimated match for next frame (Red line has the smallest SSE)

Fig. 11. Graphical illustration of tube of frames concept and its mechanism

Fig. 11 shows our idea of tube of frames and its mechanism. We draw a number of lines based on the predefined search range, $\varphi_{search\ range}$ and search step, $\varphi_{search\ step}$. We use the best line to estimate the next match and it emphasizes the linkage with the previous matches, the temporal information. The estimated match defines which set of reference frames is assigned to the next input frame.

#### 2) Key Frame Matching

We separate our key frame matching process into two steps. The first step is low-resolution whole frame matching for quick and rough decision. The second step is HOG patch-based matching for verification of the decision. The first step

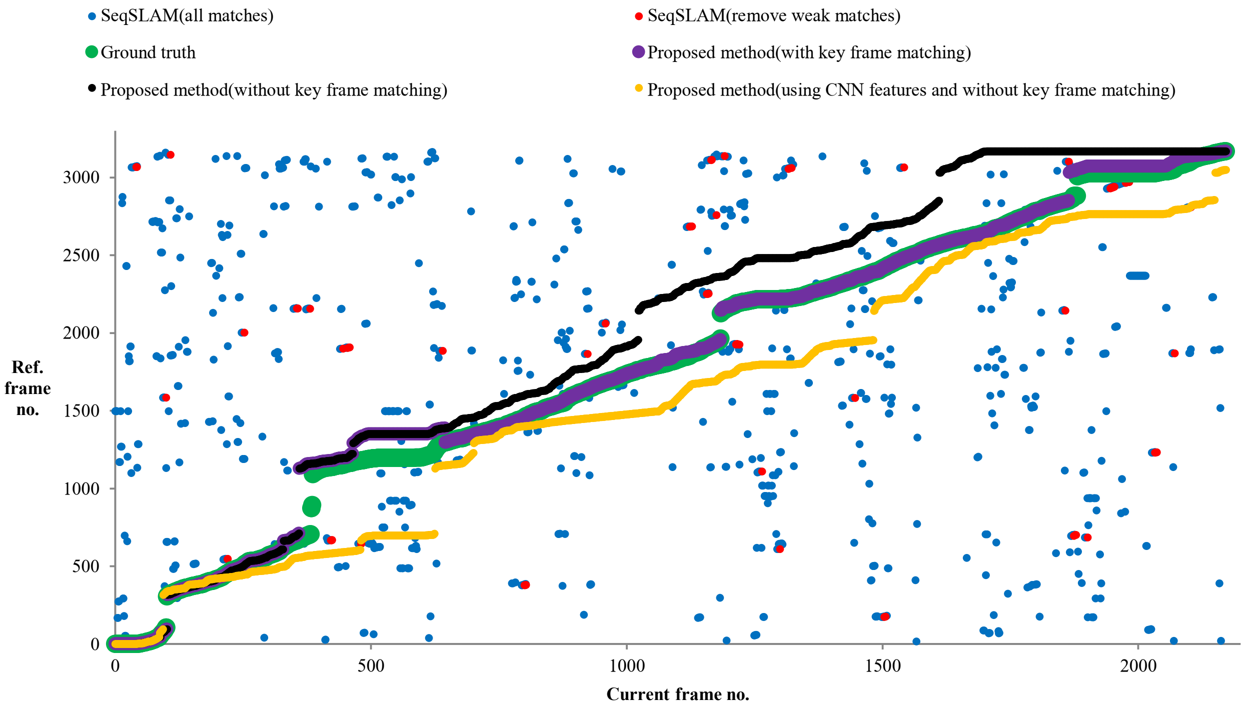


Fig. 12. Match pairs from different methods (ref. seq.: LRT-L1, testing seq.: LRT-L2, nighttime sequence)

TABLE II. Overall precision and recall of different methods

| Testing sequence | Precision(%)(Recall (%)) | | | | Recall(number of frames) | | | |
|---|---|---|---|---|---|---|---|---|
| | Proposed method | SeqSLAM | SeqSLAM (remove weak matches) | Proposed method (using CNN features & without KFM) | Proposed method | SeqSLAM | SeqSLAM (remove weak matches) | Proposed method (using CNN features & without KFM) |
| LRT-L2 | 88.8(100) | 52.7(100) | **90.2**(38.4) | 13.8(100) | **2172** | **2172** | 835 | **2172** |
| LRT-L3 | **67.6**(100) | 23.5(100) | 61.7(14.2) | 36.0(100) | **2534** | **2534** | 360 | **2534** |
| LRT-L4 | **86.5**(100) | 28.9(100) | 60.2(17.3) | 15.2(100) | **2388** | **2388** | 412 | **2388** |

is similar to the aforementioned section and we convert the difference into similarity using Eqn. (12).

$$s=\begin{cases}0 & ,if\ d>upperbound\\ \dfrac{upperbound-d}{upperbound-lower\,bound} & ,otherwise\end{cases} \qquad (12)$$

where $d$ is the average pixel difference between the current input frame and the current key frame. The largest value of $d$ is 255 but we observe that all differences vary between 50 and 90. Therefore, we set the *upper bound* = 90 and *lower bound* = 50 for concise comparison. If $s$ is larger than a threshold, $T_{match}$, the best match is found directly and we reset the tube using the best match to restart the linkage with the previous matches. This means that the best match acts as a new starting point for the tube to slide continuously without suffering from the accumulated deviation of the previous movement of the sliding tube. If $s$ is lower than a threshold, $T_{mismatch}$, no further matching would be performed. For the rest, we perform a HOG patch-based matching for verification. We use Cosine similarity, Eqn. (6), to calculate the confidence level between two patches and two frames, $CL_{patch}$ and $CL_{frame}$.

$$CL_{frame}=\frac{\sum_{p}^{P}((1+(CL_{patch,p}-T_{patch}))\times CL_{patch,p})}{P} \qquad (13)$$

where $P$ is the number of effective feature patches of the current key frame, $T_{patch}$ is a threshold of identifying good match of patches and $CL_{patch,p}$ is the confidence level between patch $p$ of two frames. Each patch is compared with the previously prepared 169 shifted feature vectors. If $CL_{frame}$ is larger than a threshold, $T_{frame}$, the best match is found and the tube would also be reset. We also adopt parallel key frame matching scheme for handling false negatives.

## III. EXPERIMENTAL RESULTS

### *A. Dataset*

A large amount of experiments have been done. We compare our proposed method with OpenSeqSLAM [25] and AlexNet [26] conv3 layer features based approach [10, 11]. We used the default parameters of OpenSeqSLAM except that we modified the reduced frame size from 64x32 to 64x48. OpenSeqSLAM is an implementation of SeqSLAM [4] which makes use of low-resolution whole frames for the localization problem. For CNN approach, we used AlexNet to replace the traditional feature extraction process such as downsampled images and HOG feature vectors. We computed the similarity between frames by comparing the CNN feature vectors with Cosine similarity, Eqn. (6). The dataset is obtained from a public transportation in Hong Kong, Light Rail Transit (LRT) which consists of many practical situations such as varying speeds, blurring and changes in illumination. There are 4 sequences of the same route, 1 at nighttime and 3 in the daytime. We used one of the daytime sequences as a reference sequence for performing key frame identification. On average, each sequence is with the length of 2565 frames. The AlexNet conv3 layer features were extracted under the Caffe framework [28] pre-trained by ImageNet [29]. Note that no code optimization technique and parallel programming has been applied.

### B. Accuracy and Efficiency Comparison

A characteristic of our proposed method is to include the key frame matching (KFM) mechanism which is used to reset the sliding tube for eliminating deviation. From Fig. 12, it is clear that key frame matching (KFM, purple curve) is useful to compensate for deviation to enhance the performance in localization when comparing with our method without KFM (black curve). We evaluated the performance of a current testing sequence by means of frames deviation. We computed the frame number differences between the obtained results and the ground truth (green curve). We found that 50 frames deviation from the ground truth is an acceptable range of error in the sample dataset. From Table II, considering the case with the best performance, our proposed method achieves 88.8% precision with 100% recall when the acceptance of error is 50 frames. By contrast, SeqSLAM achieves 90.2% precision with only 835 recall frames (refer to the obvious discrete distribution of the blue and red dots in Fig. 12). Only 735 frames (835x0.902) report the correct match pairs. If we consider 100% recall, our proposed method has higher precision than SeqSLAM. The main reason for lower precision of SeqSLAM with high recall is that the train runs at different speeds on different journeys and SeqSLAM assumes similar speed situation. For the evaluation of CNN features based approach (orange curve in Fig. 12), we used AlexNet conv3 layer features to replace the low-resolution whole frame descriptor for tracking and no key frame matching (KFM) is used to eliminate deviation. The dimension of AlexNet conv3 layer feature vector is 384x13x13=64896 [10, 26]. The experimental results showed that CNN features also suffer from deviation in temporal informative tasks like localization. This means that techniques for deviation elimination are still needed to boost the performance in localization tasks even modern CNN features were used. Another reason for such poor performance is that blurring is a limitation of the CNN pre-trained on ImageNet [30] and the sample dataset contains blurring problems in the nighttime sequence, as shown in Fig. 13.

The time cost of SeqSLAM is heavily dependent on the total number of frames (23 ms/frame on average on the sample dataset) while our proposed method is less sensitive to the length of sequences. Without KFM for deviation compensation, the proposed method requires only 8.09 ms/frame on average which is nearly invariant to the total number of frames as the tube size is fixed. With KFM, the time cost of our proposed method increases to 13.4 ms/frame on average and it depends on the complexity of key frames. Some key frames have more and larger effective feature patches. Motion type classification and tube of frames concept are effective to simplify the operation per frame. For CNN approach, if only CPU is used, 133.2 ms and 4.9 ms are required to describe an image and process a match on average respectively. It burdens a real-time localization system with the high time cost. On average, our proposed method improves SeqSLAM by a factor of 1.7 in the time cost.

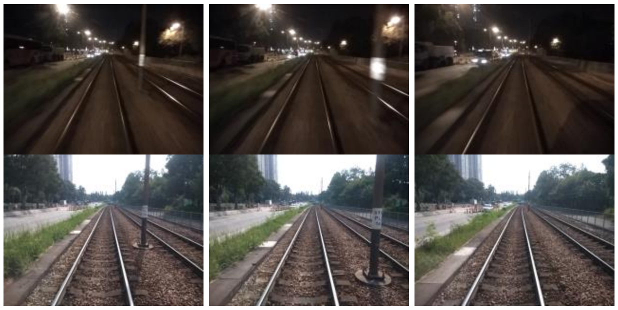


Fig. 13. Match pairs from our proposed method on LRT dataset (top: current testing frames with blurring and changes in illumination, bottom: matched reference frames)

## IV. Conclusion

In this paper, we have proposed a novel method which employs both key and non-key reference frames. Low-resolution whole frame tracking and tube of frames are used to calculate and estimate the best match candidates. Our tube of frames concept emphasizes the linkage with the previous matches to make use of the temporal information. Key frame matching is used to compensate for the deviation from the movement of the sliding tube. The current proposed method can achieve both high precision and recall with nearly length-of-sequence invariant property. Compared with SeqSLAM, our proposed method is more suitable to handle varying speeds situations, especially for stop situations. For future development, segmentation and scene understanding can enhance the quality of key frames by realizing the nature of features. Other feature spaces can be studied to enhance further the adaptability of our proposed method as whole frame matching is sensitive to changes in viewpoints and HOG suffers from severe blurring problems. We will extend the proposed method to non-railway cases and hence there could be more applications such as self-guided tour in museums. For CNN approach, applications to temporal informative tasks such as localization and visual odometry have been studied intensively and there is plenty of room for improvement. We believe that we can utilize the temporal information and CNN features to obtain superior performance in localization tasks in our future studies.

## Acknowledgment

This work was supported by the Hong Kong Polytechnic University (1-ZE1B), and Mr. Li Chu Tak would like to acknowledge the Postgraduate Studentship offered by the same university.

## References

[1] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser, “Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194-220, Sept. 2017.

[2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard, “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the

Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309-1332, Dec. 2016.

[3] Yongliang Qiao, Cindy Cappelle, and Yassine Ruichek, "Visual Localization across Seasons using Sequence Matching based on Multi-Feature combination," *Journal of Sensors*, vol. 17, no. 11, pp. 2442-2463, Oct. 2017.

[4] Michael J. Milford and Gordon F. Wyeth, "SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights," *Proceedings, IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, MN, USA, pp. 1643-1649, May 2012.

[5] Michael J. Milford, Felix Schill, Peter Corke, Robert Mahony, and Gordon Wyeth, "Aerial SLAM with a Single Camera Using Visual Expectation," *Proceedings, IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, pp. 2506-2512, May 2011.

[6] Niko Sünderhauf and Peter Protzel, "BRIEF-Gist - Closing the Loop by Simple Means," *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, pp. 1234-1241, Sep. 2011.

[7] Mark Cummins and Paul Newman, "Highly Scalable Appearance-Only SLAM - FAB-MAP 2.0," *Proceedings, Conference on Robotics: Science and Systems (RSS)*, Seattle, Washington, United States of America, pp. 39-46, 28 Jun. to 1 Jul. 2009.

[8] Michael J. Milford, Walter Scheirer, Eleonora Vig, Arren Glover, Oliver Baumann, Jason Mattingley, and David Cox, "Condition-Invariant, Top-Down Visual Place Recognition," *Proceedings, IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, pp. 5571-5577, May 2014.

[9] Meng Yao, Wan-Chi Siu and Ke-Bin Jia, "Learning-based Scene Recognition with Monocular Camera for Light-Rail System," *Proceedings*, *IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*, Hamilton, New Zealand, pp.230-236, 30 Jan. to 2 Feb. 2018.

[10] Niko Sünderhauf, Ssareh Shirzai, Feras Dayoub, Ben Upcroft, and Michael Milford, "On the Performance of ConvNet Features for Place Recognition," *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, pp. 4297-4304, Sept. 2015.

[11] Tayyab Naseer, Wolfram Burgard and Cyrill Stachniss, "Robust Visual Localization Across Seasons," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289-302, Apr. 2018.

[12] Fei Han, Xue Yang, Yiming Deng, Mark Rentschler, Dejun Yang, and Hao Zhang, "SRAL: Shared Representative Appearance Learning for Long-Term Visual Place Recognition," *IEEE Robotics and Automation Letters (RAL)*, vol. 2, no. 2, pp. 1172-1179, Apr. 2017.

[13] Saleh Aly, Daisuku Deguchi, and Hiroshi Murase, "Blur-invariant Traffic Sign Recognition Using Compact Local Phase Quantization," *Proceedings, IEEE International Conference on Intelligent Transportation Systems (ITSC)*, The Hague, Netherlands, pp. 821-827, Oct. 2013.

[14] Vidyagouri B. Hemadri and Umakant P. Kulkarni, "Road Sign Detection and Recognition in Adverse Case using Pattern Matching," *Proceedings, International Conference & Workshop on Advanced Computing ( ICWAC)*, Mumbai, India, pp. 49-53, Feb. 2013.

[15] Zhenwei Shi, Zhengxia Zou, and Changshui Zhang, "Real-Time Traffic Light Detection With Adaptive Background Suppression Filter," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 690-700, Mar. 2016.

[16] Sanjay Saini, S. Nikhil, Krishna Reddy Konda, Harish S Bharadwaj, and N. Ganeshan, "An Efficient Vision-Based Traffic Light Detection and State Recognition for Autonomous Vehicles," *IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, pp. 606-611, Jun. 2017.

[17] Hao Wu and Wan-Chi Siu, "Real Time Railway Extraction By Angle Alignment Measure," *Proceedings, IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, Canada, pp. 4560-4564, Sept. 2015.

[18] Hunjae Yoo, Ukil Yang, and Kwanghoon Sohn, "Gradient-Enhancing Conversion for Illumination-Robust Lane Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1083-1094, Sept. 2013.

[19] Chup-Chung Wong, Wan-Chi Siu, Paul Jennings, Stuart Barnes, and Bernard Fong, "A Smart Moving Vehicle Detection System Using Motion Vectors and Generic Line Features," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 3, pp. 384-392, Aug. 2015.

[20] Xue-Fei Yang and Wan-Chi Siu, "Vehicle Detection under Tough Conditions using Prioritized Feature Extraction with Shadow Recognition," *Proceedings, IEEE 22nd International Conference on Digital Signal Processing (DSP)*, London, UK, pp. 1-5, Aug. 2017.

[21] D. Nistér, O. Naroditsky, and J. Bergen, "Visual Odometry," *Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, pp. 652-659, 27 Jun. to 2 Jul. 2004.

[22] Chuanqi Cheng, Xiangyang Hao, Zhenjie Zhang, and Mandan Zhao, "Monocular Visual Odometry Based on Optical Flow and Feature Matching," *Proceedings, IEEE the 29th Conference on Chinese Control And Decision (CCDC)*, Chongqing, China, pp. 4554-4558, May 2017.

[23] Haifeng Li, Hongpeng Wang, and Jingtai Liu, "Monocular Visual Odometry Using Vertical Lines in Urban Area," *Proceedings, IEEE the 32nd Conference on Chinese Control (CCC)*, Xi'an, China, pp. 5676-5681, Jul. 2013.

[24] Hoi-Kok Cheung, Wan-Chi Siu, Steven Lee, Lawrence Poon, and Chiu-Shing Ng, "Accurate Distance Estimation Using Camera Orientation Compensation Technique for Vehicle Driver Assistance System," *Proceedings, IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, pp. 231-232, Jan. 2012.

[25] Niko Sünderhauf, Peer Neubert, and Peter Protzel, "Are We There Yet? Challenging SeqSLAM on a 3000 km Journey Across All Four Seasons," *Proceedings, Workshop on Long-Term Autonomy*, *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, pp. 102-115, May 2013.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceedings, the 25th International Conference on Neural Information Processing Systems (NIPS),* Lake Tahoe, Nevada, USA, pp. 1097-1105, Dec. 2012.

[27] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," *Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, pp. 886-893, Jun. 2005.

[28] Y Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," *Proceedings, the 22nd ACM International Conference on Multimedia*, Orlando, Florida, USA, pp. 675-678, Nov. 2014.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "Imagenet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, Dec. 2015.

[30] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox, "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT," *arXiv preprint arXiv:1405.5769v1*, pp. 1-9, May 2014.