VIDEO STABILIZATION FOR VEHICULAR APPLICATIONS USING SURF-LIKE DESCRIPTOR AND KD-TREE

Keng-Yen Huang, Yi-Min Tsai, Chih-Chung Tsai, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering
National Taiwan University, Taipei, Taiwan
{kyhuang,ymtsai,cctsai,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

This paper describes a method to stabilize video for vehicular applications based on feature analysis. An investigation on camera motion model is conducted. Harris features are extracted under the proposed resolution adaptation scheme. Besides, features are described with SURF-like descriptor. For feature matching, KD-tree with best-bin-first search significantly reduces the matching time. A damping filer is utilized to model and predict the unwanted oscillation. 93.1% correct rate in average is achieved in divergent driving conditions. Only 0.114 second is required to process a frame at resolution 1280×960. The provided benchmark shows outperformance of the proposed method.

***Index Terms*—** Video stabilization, damping filter, KD-tree, motion analysis, and feature extraction

1. INTRODUCTION

As vehicular safety systems become more and more popular these years, many applications such as advanced driver assistance systems (ADAS) are proposed to protect drivers from car accidents. Vehicular video-based processing (VVP) is to provide a more intelligent assistance and reduce the cost of the whole system. However, severe oscillation may cause blur problems and result in low accuracy of VVP . To eliminate the oscillation effects, video stabilization is adopted as a preprocessing stage.

Video stabilization mainly relies on global motion estimation (GME). GME attempts to estimate the global motion and separates the motion into intentional motion (IM) and unwanted motion (UM). By subtracting the UM, a stabilized video can be obtained from a shaky condition.

Conventionally, video stabilization is used to solve jitter in hand-held devices. In [1], camera motion is transformed into frequency domain. UM belongs to higher frequency which can be removed through frequency decomposition. Block-based solutions are commonly employed to estimate global motion. For example, motion vectors are utilized to estimate global motion through Newton-Raphson's method [2]. By analyzing the video content [3] and choosing the proper processing resolution, the bulky computation of block-based approaches can be reduced. For high-speed processing, extracting the edge and projecting the content from 2D to 1D are proposed in literatures [4, 5]. Moreover, Kalman filter [6] is applied to separate the estimated motion into UM and IM. SIFT feature is gradually utilized for tracking moving background [7]. While in vehicular applications, these algorithms are either time-consuming or degraded in accuracy. For this purpose, lane line monitoring [8] utilizes lane lines and the vanishing point to stabilize video. However, it is restricted to the evidence of lane lines.

This paper is organized as follows. We briefly describe the characteristics of camera motion in Sec. 2. Proposed methods are detailed in Sec. 3. In Sec. 4, experimental results are discussed. The contribution is summarized in Sec. 5.

2. CAMERA MOTION ANALYSIS

Knowing the camera motion is beneficial to construct a proper motion model for the oscillation. Assume that the image plane is parallel to XY plane with location $Z = -F$. Besides, the optical axis is coincident with the Z-axis and the image formation is simply described by the projection matrix in Eq. 1,

$$\Pi = \begin{pmatrix} -F & 0 & 0 & 0 \\ 0 & -F & 0 & 0 \\ 0 & 0 & -F & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

$$I^{'} = \Pi I \mapsto (-F\alpha/\gamma, -F\beta/\gamma, -F, 1)^T \tag{2}$$

where $I = (\alpha, \beta, \gamma, 1)^T$ is denoted as the world homogeneous coordinates of a point in 3D space and $I^{'}$ is the homogeneous coordinates in 2D image plane.

Suppose that the optical axis is not coincident with Z-axis because of camera motion. The transformation matrix T_M is used to transform the optical axis to Z-axis. The inverse transform T_M^{-1} is applied on the world coordinates before I is projected to image plane. Consequently, the new homogeneous coordinates in image plane $I^{''}$ is calculated by $I^{''} = \Pi T_M^{-1} I$.

Camera motion can be classified into two categories, rotational motion and translational motion. For rotational motion around X-axis (*pitch*), the overall projection is formulated as follows,

$$T_{M_{pitch}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \cos\phi & 1 & -\sin\phi & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

$$\begin{aligned} I^{''} &= \Pi T_{M_{pitch}}^{-1} I \\ &= (-F(\alpha\cos\phi - \gamma\sin\phi), -F\beta, \\ &\quad -F(\alpha\sin\phi + \gamma\cos\phi), \alpha\sin\phi + \gamma\cos\phi)^T \end{aligned} \tag{4}$$

When the rotation angle ϕ approaches zero, the difference between $I^{'}$ and $I^{''}$ is given by,

$$I^{''} - I^{'} = (F\phi, 0, 0, 0) \tag{5}$$

	Hand-held	Vehicle-mounted
Depth variation	Insignificant, modeled as one motion model	Single model is not enough
Object position	Insignificant	Different positions may have different motions
Motion separation	Frequency segregation	Not knowing the frequency threshold

Table 1. Comparison of three primary factors on video stabilization when camera is mounted on a hand-held or a vehicular platform.

The Eq. 5 reveals that the induced transformation $F\phi$ in image plane is like a translational displacement. This inference is also applied to Y-axis (*yaw*). If the camera is rotated along Z-axis (*roll*) with angle ϕ, the same ϕ is observed in image plane.

As for translation, if a camera's translational motion is (a, b, c) in world coordinates. The displacement $I^{''} - I^{'}$ in image plane is computed by Eq. 6 and Eq. 7 accordingly,

$$\begin{aligned} I^{''} &= (-F(\alpha/\gamma - a/\gamma + O(\alpha c/\gamma^2)), \\ &\quad -F(\beta/\gamma - b/\gamma + O(\beta c/\gamma^2)), -F, 1)^T \end{aligned} \tag{6}$$

$$\begin{aligned} I^{''} - I' &= -F(a/\gamma + O(\alpha c/\gamma^2), \\ &\quad b/\gamma + O(\beta c/\gamma^2), 0, 0) \end{aligned} \tag{7}$$

where γ is the distance between the observed point and the camera. Therefore, from Eq. 7, unless the translational motion of camera is in the order of γ, the introduced difference in image plane is insignificant. This means the camera must be moved larger than 10 center meters to get noticed during frame transition, which is not possible in regular hand-held situations.

However, in vehicular applications, camera motion along Z-axis c becomes large. From Eq. 7, if c is large, the position of the point (α , β) becomes a dominant factor in forming the displacement $I^{''} - I^{'}$. In addition, points with different depth γ introduce different motion vectors. The comparison between the hand-held and vehicle-mounted video is shown in Table 1.

3. PROPOSED ALGORITHM

The algorithm flow is shown in Fig. 1. Firstly, current resolution is down-sampled to the desired one by resolution adaptation. Secondly, Harris features with SURF-like descriptors are extracted from the down-sampled frame after RGB-to-Gray color conversion. Feature points are matched by constructing KD-tree with best-bin-first search. Lastly, through a damping filter, UM is subtracted from the calculated global motion and the stabilized result is produced.

3.1. Resolution adaptation

Frames are processed under two resolutions, original full resolution (R_1) and subsampled lower resolution (R_2). Every frame is initially processed under R_2 and the primary estimation is produced. If the primary estimation is similar to the prediction from previous frame, the result is passed to motion prediction stage as the final estimation, otherwise, it generates a new estimation under full resolution.

From Eq. 7, points away from image center conduct larger displacements in image plane when high-speed driving. Thus, the central area provides a speed invariant and lower depth variation circumstance.

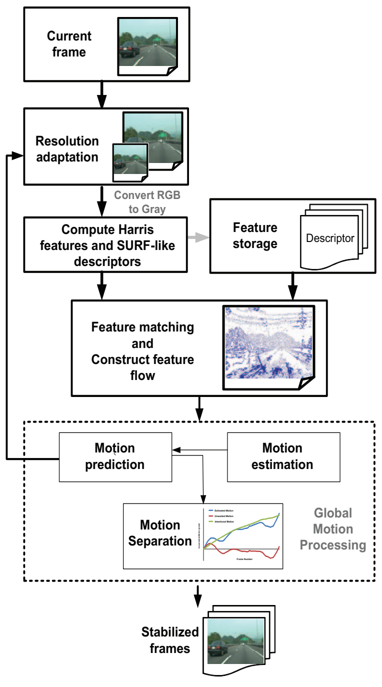

Fig. 1. Block diagram of the proposed process.

3.2. Feature extraction and matching

Points are extracted as features by applying Harris matrix A described in Eq.8.

$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{8}$$

where I_x and I_y are denoted as partial derivatives. If both eigenvalues λ_1 and λ_2 of A are large, this point is claimed as a feature. M_c in Eq. 9 is used to determine the values of eigenvalues.

$$\begin{aligned} M_c &= \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 \\ &= det(A) - \kappa \cdot trace^2(A) \end{aligned} \tag{9}$$

where κ is a tunable parameter ranging from 0.04 to 0.15. If M_c is larger than a designed threshold η, it implies λ_1 and λ_2 are large and features are decided accordingly. Lowering the value of η is to increase the number of feature candidates. Miss matching caused by inconsistent feature selection between frames is ultimately diminished.

Features are represented with the SURF-like [9] descriptor. An extracted feature and its surrounding region form a descriptor vector V. The surrounding region is further split into 2×2 sub-regions with a sub-region descriptor v illustrated in Fig. 2(a). For each sub-region, Haar wavelet responses are computed in horizontal direction (d_x) and vertical direction (d_y). Afterwards, the wavelet responses are summed up and form a set of entries to v. Absolute values are also calculated to bring in the polarity of the intensity changes. Hence, each sub-region has a four-dimensional descriptor vector v described in Eq. 10. Examples are shown in Fig. 2(b). This results in a full descriptor vector V with 16 dimensions for one feature.

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \tag{10}$$

The feature descriptors of the previous frame is clustered by building KD-tree. Every entry represents a dimension in the K-entry

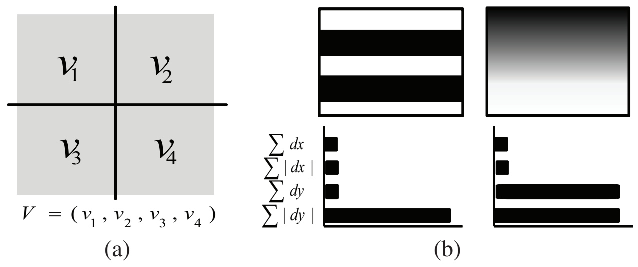

Fig. 2. (a) Descriptor vector V is composed of four sub-regions' vectors v_1,v_2,v_3, and v_4. (b) Sub-region descriptor illustrations.

tree structure. A feature is then randomly selected to be the partition node. In all descriptors of selected features, the separating dimension with the largest deviation is chosen from the 16 entries (dimensions) of V.

Each feature extracted from the current frame traverses the built KD-tree to find its corresponding point with best-bin-first search. The feature flow field is constructed when all feature points are matched.

3.3. Global motion processing

Global motion processing consists of three modules, motion estimation, motion prediction, and motion separation. For motion estimation, Eq. 11 states that the translational component is the average of the sampled feature motion vectors (FMV). A FMV stands for the displacement between the two matched features.

$$ME_x = \frac{1}{N}\sum_{i=1}^{N} FMVx_i\ ,\ ME_y = \frac{1}{N}\sum_{i=1}^{N} FMVy_i \qquad (11)$$

where N is the total number of FMV in the feature flow field.

In motion prediction, a damping filter is proposed to model the UM as a damping spring described in Eq. 12.

$$UM(f) = O_A exp(\frac{2\xi\pi f}{T_F})\cos(\frac{2\pi f}{T_F} + \psi) \qquad (12)$$

where UM(f) is denoted as the function of unwanted oscillation at a certain frame f and O_A is the maximum oscillating amplitude appeared in image plane of an oscillation. ξ is the damping coefficient, ψ is the initial condition of each oscillation, and T_F is denoted as the damping period in terms of frame number. ξ and T_F are dependent on the vehicles' shock absorbers. These parameters are automatically generated after experiencing an initial oscillation. Through modeling the UM, global motion (GM) is separated and refined to only intentional fraction, which suggests a stabilized video.

4. EXPERIMENTS

The program is run on a PC with Pentium IV 2.8GHz CPU equipped. The original resolution of video sequences is 1280×960. Different environments are tested, including highway, tunnel, and complicated city streets. There are more than 500 frames in each case. Processing resolution R_1 is 1280×960, while R_2 is 320×240. Correct estimation is defined as 1 pixel tolerance between the estimation and the ground truth.

For feature matching, the advantage of KD-tree is reducing the number of feature it has to search. The efficiency is evaluated by measuring the average percentage of searched features (traversed

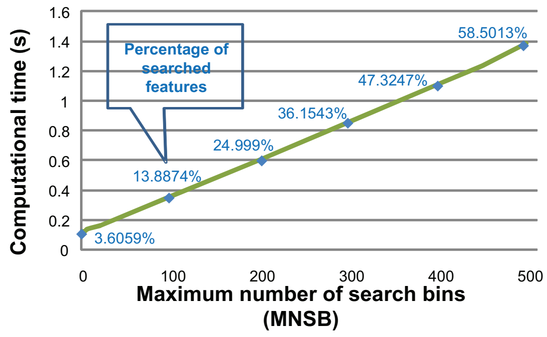

Fig. 3. Computation time in different maximum number of search bins. The blue points indicates the corresponding percentage of searched features in a KD-tree.

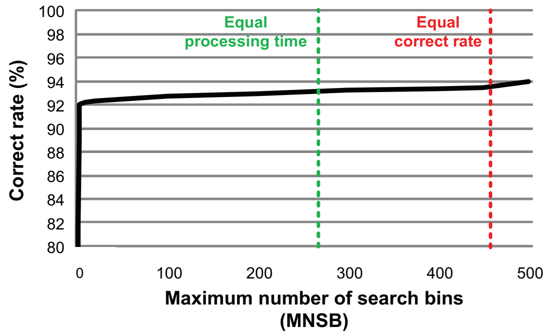

Fig. 4. Correct rate of different maximum search bins. The green line marks the equal processing time boundary and the red line marks the equal accuracy bounday.

nodes) and the processing time per frame. The major factor influencing the searched feature percentage is the maximum number of search bins (MNSB). MNSB defines the maximum number of bins when a feature traverses its corresponding node in KD-tree. As shown in Fig. 3, the computational time and the percentage of traversed nodes are positively proportional to MNSB.

On the other hand, linear search is a straightforward baseline strategy. It leaves out the time to construct the KD-tree. In Fig. 4, the black curve describes the correct rate in different MNSB. The green line marks the bounday where linear search and KD-tree spend the same processing time. The red line highlights the boundary where both achieve equal correct rate. The result reveals that the ordinary KD-tree without enhancement takes more time to reach the same correct rate of linear search. However, with best-bin-first search, correct rate quickly attains 92% at MNSB = 4. Only 3.7% of features are traversed and the computational time is reduced by 84% compared to linear search. Fig. 5 shows the processing time comparison between several approaches. The proposed method unrolls the exceptional performance in reducing computational efforts.

An example of motion separation is shown in Fig. 6. Estimated motion (blue line) is divided into IM (green line) and UM (red line) by using the damping filter. Oscillation period T_F is initialized to 20 frames. It differs from variant kinds of cars' shock absorbers and is adjustable. With resolution adaptation, computation time is reduced more than 60% while only 1.4% drop in correct rate compared to full-frame processing.

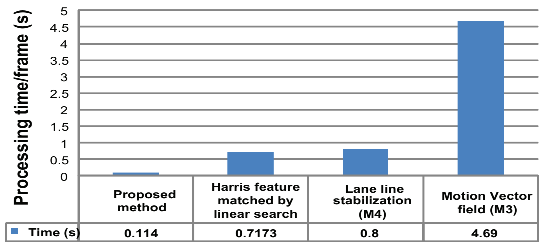

Fig. 5. Computation time of different algorithms.

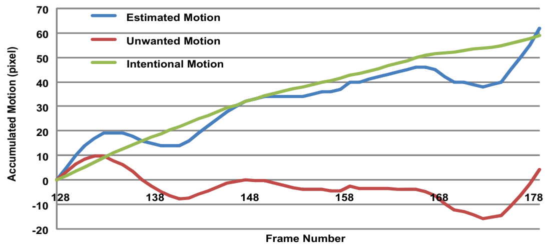

Fig. 6. Motion separation using damping filter. T_F is equal to 20 with ξ is 0.025.

Four state-of-the-arts are benchmarked and listed below.

M1: Pixel-based diamond search
M2: Curve warping [4]
M3: Motion vector field with Newton-Raphson's method [2]
M4: Lane line and vanishing point stabilization [8]

In Table 2, all algorithms claim about 90% correct rate in highway. However, some shortages appeared in other conditions. In tunnel and city street, 1D information (M2) is insufficient to achieve high accuracy. Large homogeneous area like tunnel degrades the performance of pixel-based diamond search (M1) and motion vector field (M3). Vanishing point-based (M4) stabilization can be only applied to conditions with clear lane lines. For computational time analysis, the proposed method attains 0.114 second to process a frame. Conventional pixel-based approaches (M1) and block-based motion vector (M3) are slower than other approaches. Curve warping (M2) is the fastest by processing frame in 1D, but only 50% in correct rate is achieved. At last, examples of the proposed video stabilization are shown in Fig. 7.

5. CONCLUSIONS

We present an efficient video stabilization method targeted at vehicular applications. Resolution adaptation accelerates the stabilization procedure. SURF-like descriptors enhance the robustness of Harris

Methods	Correct rate (%)			Processing time per frame (s)
	Highway	Tunnel	City street	
M1	97.4	53.71	92.65	8.27
M2	89.74	46.3	40.32	0.0322
M3	94.85	52.26	91.26	4.69
M4	95.4	N/A	N/A	0.8
Proposed	95.17	91.4	92.31	0.114

Table 2. Comparison between different methods.

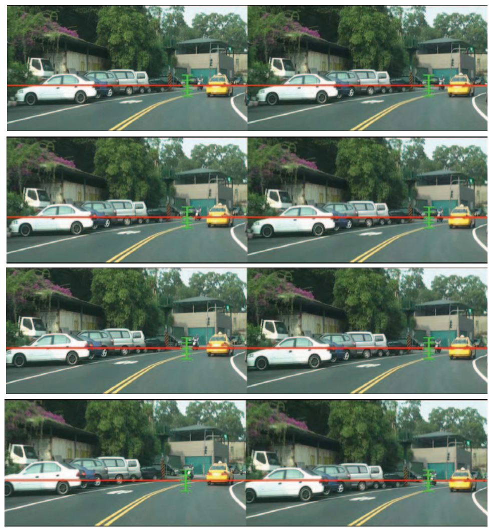

Fig. 7. Examples for video stabilization. Left columns are original four consecutive frames during oscillation. Right columns are corresponding stabilized results.

features. Constructing KD-tree with best-bin-first search greatly decreases the computation time in feature matching. A damping filter is utilized to remove unwanted oscillation calculated from feature motions. Results show the identical accuracy under different conditions and report decent performance.

6. REFERENCES

[1] Etay Mar Or and Dmitry Pundik, "Hand motion and image stabilization in hand-held devices," *IEEE Transactions on Consumer Electronics*, vol. 53, pp. 1508–1512, November 2007.

[2] Yeping Su, Ming-Ting Sun, and Vincent Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 232–242, February 2005.

[3] Aziz Umit Batur and Bruce Flinchbaugh, "Video stabilization with optimized motion estimation resolution," in *Proc. of International Conference on Image Processing(ICIP)*. IEEE, 2006, pp. 465–468.

[4] Angelo Bosco, Arcangelo Bruna, Sebastiano Battiato, Giuseppe Bella, and Giovanni Puglisi, "Digital video stabilization through curve warping techniques," *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 220–224, May 2008.

[5] A.J. Crawford, H. Denman, F. Kelly, F. Pitie, and A.C. Kokaram, "Gradient based dominant motion estimation with integral projections for real time video stabilization," in *Proc. of International Conference on Image Processing(ICIP)*. IEEE, 2004, vol. 5, pp. 3371–3374.

[6] Marius Tico and Markku Vehvilainen, "Constraint motion filtering for video stabilization," in *Proc. of International Conference on Image Processing(ICIP)*. IEEE, 2005, vol. 3, pp. 569–572.

[7] Chih-Yuan Chung and Homer H. Chen, "Feature-based full-frame image stabilization," in *Proc. of International Symposium on Multimedia*, 2007, pp. 100–106.

[8] Yu-Ming Liang, Hsiao-Rong Tyan, Shyang-Lih Chang, Hong-Yuang Mark Liao, and Sei-Wang Chen, "Video stabilization for a camcorder mounted on a moving vehicle," *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 1636–1647, November 2004.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Proc. of European Conference on Computer Vision (ECCV)*. IEEE, 2006.