Integrating Dashcam Views through Inter-Video Mapping

Hsin-I Chen Yi-Ling Chen Wei-Tse Lee Fan Wang Bing-Yu Chen
National Taiwan University
{fensi,yiling,wlee,fan,robin}@cmlab.csie.ntu.edu.tw

Abstract

In this paper, an inter-video mapping approach is proposed to integrate video footages from two dashcams installed on a preceding and its following vehicle to provide the illusion that the driver of the following vehicle can see-through the preceding one. The key challenge is to adapt the perspectives of the two videos based on a small number of common features since a large portion of the common region in the video captured by the following vehicle is occluded by the preceding one. Inspired by the observation that images with the most similar viewpoints yield dense and high-quality matches, the proposed inter-video mapping estimates spatially-varying motions across the two videos utilizing images of very similar contents. Specifically, we estimate frame-to-frame motions of each two consecutive images and incrementally add new views to form long-range motion representation. On the other hand, we dynamically infer spatial-varying motions across the two videos that propagated from local feature correspondences to trajectories. In this way, the observed perspective discrepancy between the two videos can be well approximated by our motion estimation. Once the inter-video mapping is established, the correspondences can be updated incrementally, so the proposed method is suitable for on-line applications. Experiments with real-world challenging videos demonstrate the effectiveness of our approach.

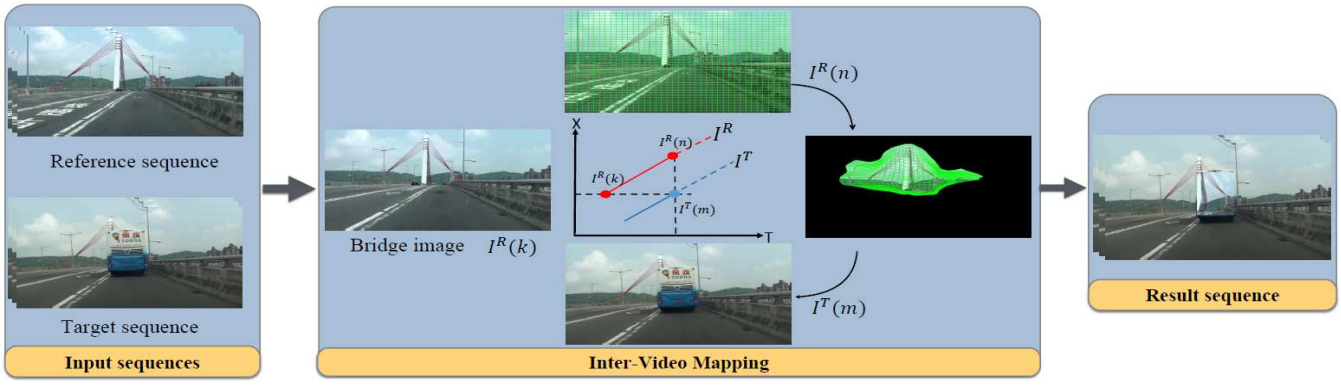
1. Introduction

It is quite common that a driver's view is obscured by a large preceding truck, resulting in increased reaction time due to poor visibility . Although dashcams (dashboard cameras) have achieved massive popularity, and future vehicles may be enabled with vehicle-to-vehicle (V2V) communication to access the videos feed from the preceding vehicles [9], superimposing the videos without adjustment for fitting the driver's viewpoint leads to visually misaligned views, which may cause more distraction to the driver.

The problem of transforming views with great perspective variations between two different dashcams installed on a preceding vehicle and its following vehicle was first investigated by Chen *et al*. [3]. Under this environment, a practical solution needs to tackle at least the following three challenges: (1) It must model the foreshortening effect induced by arbitrary time-shifts between the two dashcams; (2) It must model the parallax owning to different camera locations; (3) It must maintain the temporal coherence for stable and comfortable viewing experience. Ideally, a faithful novel view can be generated if we have dense 3D structures of the scene. However, obtaining such a dense reconstruction from 2D images is extremely challenging in terms of both accuracy and efficiency. In [3], perspective discrepancy and parallax are dealt with by estimating spatially-varying warping functions from feature trajectories. However, obtaining long trajectories is particularly hard in driving scenarios due to large textureless regions, and the tracked points are sparsely distributed, leaving large image regions without close and relevant warping constraints. Besides, their work naïvely extended an image stitching method [23] to videos, which leads to temporal incoherence, causing flickering and waving artifacts.

In this paper, a novel inter-video mapping framework is proposed to integrate video footages from two dashcams installed on a preceding and its following vehicle to provide the illusion that the driver of the following vehicle can see-through the preceding one, as shown in Figure 1. This is quite challenging because the concurrent views of the two videos have large perspective change and share only a small number of common features since the video captured by the following vehicle is partially occluded by the preceding one. To establish the correspondences between the two videos, we first find a bridge frame in the reference (preceding) video that looks similar with the current frame of the target (following) video, so that the correspondences can be established through the relationships between the bridge frame and the current reference and target frames, respectively.

Hence, there are two major procedures in the proposed approach. In the first procedure, perspective adaptation is performed to locally and adaptively alter the projected shape within the reference (preceding) video. Inspired by



(a) (b) (c)

Figure 1. An overview of our inter-video mapping approach. Given I^R and I^T as input (a), for each incoming image of I^T, we seek its visually closest image in I^R as the bridge image. To estimate the dense mapping function between the reference and the target image, we perform long-range motion estimation between the reference and the bridge image, together with the estimation of a non-linear geometric transformations that related the image-coordinates of the bridge and the target image (b). The projected shape and size of the reference image is adjusted and stitched into the occluded region to generate the final composite image (c).

the observation that dense and accurate correspondences are essential for establishing high-quality local warps to account for non-linear motions, and successive video frames with most similar viewpoints provide such information. We estimate spatial-varying motions between consecutive frames and incrementally accumulate them for long range motion estimation. This yields a more accurate local warp model that enables the viewpoint to be adapted to the bridge frame. In the second procedure, with the aid of rich correspondences between the bridge and the target frame, we estimate spatially-variant motions to align video frames captured from different camera locations. Additionally, our approach dynamically associate feature correspondences to the point trajectories, allowing warping frames sequentially while keeping temporal coherence without resorting to solving spatio-temporal optimization.

Contribution. We investigate the problem of perspective adaptation across two independent moving dashcams. Our solution is an one-pass process, exploiting rich correspondences from video frames with very similar viewpoints, to register two videos with large geometric variations. Our approach operates in an incremental fashion, gradually compiling robust local motions into long-range motion representations as new frame becomes available, that makes our system very efficient and suitable for on-line applications. Finally, our approach is also comprehensively evaluated on several real-world scenarios: scenes containing non-trivial parallax effects and camera zooming. The results demonstrate the effectiveness of the proposed approach.

2. Related Work

Motion Estimation is an important research topic for computer vision. Sparse feature tracking and dense optical flow are two major approaches for estimating camera motions. In the former approach, features are detected in one video frame, and tracked independently in the rest of the video [19]. In the later one, a flow vector is estimated for every pixel in one video frame, indicating the transition of the pixel in the next frame. As revealed in [18], sparse feature tracking can establish long-range correspondences (*e.g.* up to hundreds of frames), but typically only a limited amount of feature points can survive for long-range tracking. On the other hand, dense optical flows reveal more about the scene motion, but the flow field computed in a frame-by-frame manner is hard to reliably propagate to distant frames. Some hybrid solutions combine the two approaches to obtain spatially-dense and temporally-smooth trajectories [18, 13, 8]. However, such methods still have the difficulty providing satisfied solutions for constructing dense motion tracks in a truly long-range fashion, especially in the case with large perspective change and several noisy moving objects.

Video Alignment aims to establish temporal mapping between two video sequences while maximizing their content similarity. Diego *et al*. [5, 6] leveraged GPS information and posed the video sequence synchronization as a MAP inference problem. Evangelidis *et al*. [7] designed a quad descriptor and solved for spatio-temporal mapping by aggregating votes through a multi-scale scheme. Both of the above two methods assume that the videos to be aligned are captured along similar trajectories, resulting in little or no parallax. Under this condition, the transformations between the corresponding frames from the two sequences can be well modelled by a single homography. However, dashcams may be installed differently between vehicles, resulting in videos of larger perspective changes and parallax.

Our work describes the deformation of each pair of corresponding frames using local-variant motions to account for parallax induced by different camera positions.

Image and Video Warping is important for many computer vision and graphics applications, *e.g.*, image and video resizing [22, 21]. For video stabilization, Liu *et al*. [14] proposed the content-preserving-warping (CPW) technique, which computed spatial-varying warps induced by recovered 3D scene structures to synthesize stable novel views. Liu *et al*. [15] adopted the as-similar-as-possible warp to estimate bundle camera paths designed for space-time path smoothing. Inspired by their work, we also model the motion between consecutive frames using mesh-based representation, but aim to provide accurate local warps for registering frames within a video with large geometric variations.

3. Inter-Video Mapping

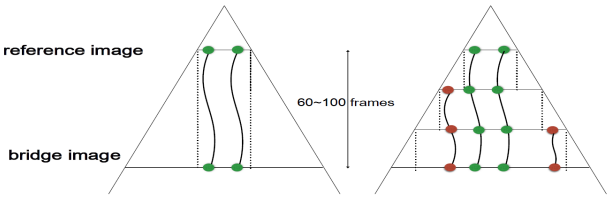
3.1. Overview

Denote the reference and target sequences captured by the preceding vehicle and its following vehicle as I^R and I^T, respectively. Our goal is to establish *dense* and *continuous* mapping between the concurrent frames $I^R(n)$ and $I^T(m)$ in I^R and I^T, where m and n are the corresponding frame indexes. It is extremely difficult due to the large geometric variations and occlusions between them. Based on the observation that feature correspondences can be most reliably established between frames with similar viewpoints, the proposed inter-video mapping method tackles this problem in two stages by utilizing a *bridge image* $I^R(k)$ in I^R, which is the visually closest frame to $I^T(m)$. Firstly, *intra-video mapping* is performed *within* I^R to realize long-range motion estimation by utilizing rich frame-to-frame correspondences and a warping-based motion model (Figure 2). Secondly, *cross-video mapping* is performed *between* I^R and I^T to build spatially dense and temporally coherent mapping by *trajectory transfer*.

Combined together, inter-video mapping achieves high-quality registration of $I^R(n)$ and $I^T(m)$. In addition, it can accomplish view integration by replacing the occluded region in $I^T(m)$ with the visual elements in $I^R(n)$, respecting the perspective variation between $I^T(m)$ and $I^R(n)$. The final synthesized image is denoted by $\hat{I}^T(m)$. Figure 1 illustrates the overview of the proposed method, and in the following sections we first explain the preprocessing stage, and then intra- and cross-video mapping algorithms in more details.

3.2. Preprocessing

To obtain the occluded regions in I^T, we first contour the preceding vehicle in the first frame of I^T (or a rear vehicle detector using HoG features [4] can be utilized) and then exploit the robust object tracking method using a collaborative model [24] to update its positions. SIFT keypoints and descriptors are detected and extracted in both I^R and I^T [16]. In addition, we apply the standard KLT feature tracker [19] to obtain a set of feature trajectories in both I^R and I^T. The tracked features lying on moving objects are ruled out by epipolar constraints [11] and a simple and effective heuristic that all features belonging to the static scene always move away from the image center under the driving scenario.



(a) (b)

Figure 2. Feature tracking is long-range but sparse (a). With the aid of frame-to-frame correspondences, our motion model is dense and robust (b).

4. Intra-Video Mapping

In this section, we first describe a motion model based on image warping, which have been successfully used to model camera motion in video stabilization [14, 15] and then extend it for long-range motion estimation between $I^R(n)$ and the bridge image $I^R(k)$ (the selection of $I^R(k)$ will be introduced in Section 5). Unlike [14], we do not rely on 3D reconstruction (*i.e.*, structure-from-motion) to recover camera path, which is computationally infeasible for an online-application.

4.1. Warping-based Motion Model

To model the *backward* motion between consecutive frames $I(t)$ and $I(t-1)$, the source frame $I(t)$ is first divided into an $n \times m$ uniform grid mesh, as illustrated in Figure 3. Given a set of feature trajectories linking $I(t)$ and $I(t-1)$, the motion can thus be represented by a warped version of this grid which best aligns the corresponding feature points. Guided by these sparse displacements, the unknown warped mesh is obtained by requiring the matched features (*e.g.*, p and $\tilde{p}$ in Figure 3) to share the same bilinear interpolation of the four corners of the enclosing grid cell after warping. At the i-th grid cell, the warping from frame t to frame $t-1$ introduces a homography $H_i(t)$, which can be determined from the motion of the four enclosing vertices. Thus, the warping-based motion model is actually a set of spatially-variant local homographies defined on a 2D grid.

To solve for the unknown mesh vertices, we perform mesh optimization by minimizing the following energy terms: a data term for matching features, a smoothness term for *regularizing* the deviation of mesh deformation, and a

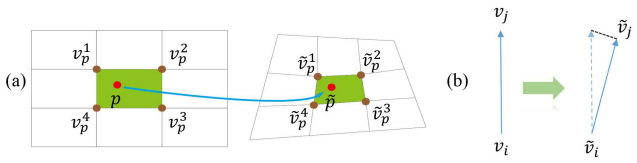


Figure 3. Warping-based motion model. (a) The feature pair $(p, \tilde{p})$ should be represented by the same bilinear interpolation of the enclosing vertices after warping. (b) Line bending energy measures the deviation of an edge after warping.

line preserving term which encourages collinearity of landmark points lying on salient line structures after warping.

Data term As shown in Figure 3, assume $p, \tilde{p}$ are the tracked features along a feature trajectory from frame t to frame $t-1$. The feature p can be represented by a 2D bilinear interpolation of the four vertices $\mathcal{V}_p = [v_p^1, v_p^2, v_p^3, v_p^4]$ of the enclosing grid cell: $p = \mathcal{V}_p \omega(p)$, where $\omega(p) = [w_p^1, w_p^2, w_p^3, w_p^4]^T$ are interpolation weights summing up to 1. For each feature p, we impose a constraint to enforce the warped mesh vertices $\tilde{\mathcal{V}}_p = [\tilde{v}_p^1, \tilde{v}_p^2, \tilde{v}_p^3, \tilde{v}_p^4]$ to approximate the corresponding feature $\tilde{p}$ with the same interpolation weights. Therefore, the data term is defined as

$$E_D(\tilde{\mathcal{V}}) = \sum_p ||\tilde{\mathcal{V}}_p \omega(p) - \tilde{p}||^2. \quad (1)$$

Smoothness term We use the same line bending term as in [22] to ensure there is not much content distortion. It requires the orientation of every edge to be similar before and after mesh deformation (as illustrated in Figure 3(b)), and is defined as

$$E_S(\tilde{\mathcal{V}}) = \sum_{(v_i, v_j) \in E} \|(\tilde{v}_i - \tilde{v}_j) - l_{ij}(v_i - v_j)\|^2, \quad (2)$$

where E represents the edges in the mesh, v_i and v_j are the endpoints of an edge, $l_{ij} = \|\tilde{v}_i - \tilde{v}_j\| / \|v_i - v_j\|$ is the length ratio of the edges before and after deformation. The length of $\tilde{v}_i - \tilde{v}_j$ is approximated by the pre-warped mesh using global homography.

Line preserving term The line preserving energy encourages three consecutive samples points on each line to remain colinear after mesh deformation as in [12],

$$E_L(\tilde{\mathcal{V}}) = \sum_{l \in L} \beta_l \sum_{i=1}^{n} \|\omega(s_i^l)\tilde{\mathbf{V}}_{s_i^l} - \frac{\omega(s_{i-1}^l)\tilde{\mathbf{V}}_{s_{i-1}^l} + \omega(s_{i+1}^l)\tilde{\mathbf{V}}_{s_{i+1}^l}}{2}\|^2, \quad (3)$$

where L are the line segments detected by the LSD algorithm [10]. Each line is represented as a set of samples as $l = \{s_0^l, ..., s_{n+1}^l\}$, where s_i^l is a sample point on l. β_l is the weight proportional to $l's$ length before warping.

Linear combination of the three energy terms forms our energy function $E(\tilde{\mathcal{V}})$:

$$E(\tilde{\mathcal{V}}) = E_D(\tilde{\mathcal{V}}) + \lambda_S E_S(\tilde{\mathcal{V}}) + \lambda_L E_L(\tilde{\mathcal{V}}), \quad (4)$$

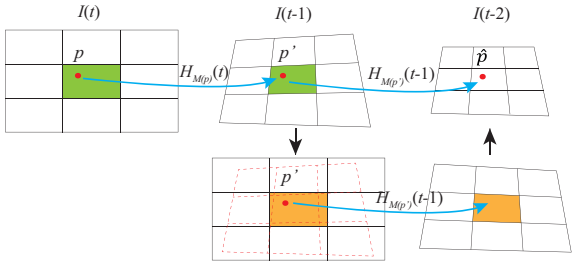


Figure 4. Local homographies are aggregated to achieve long-range motion estimation.

where λ_S and λ_L are weighting coefficients balancing the importance of each term and are set as 1 and 100, respectively. The above minimization problem is quadratic and can be solved using a standard sparse linear solver.

4.2. Long-range Motion Estimation

Recall that the above motion model consists of a set of local homographies $H_i(t)$. In [15], the local homographies of the corresponding grid cells between consecutive frames are *concatenated* to generate a bundle of spatially-varying camera paths. These camera paths are then smoothed for video stabilization. For video stabilization, it is sufficient to cascade the frame-to-frame camera motions, because the novel views to be synthesized possess similar viewpoints and camera locations with the original ones. However, in our case, the goal is to synthesize the occluded region in $I^T(m)$ with the content of a distant frame $I_R(n)$. As a result, we take the strategy of *aggregating* the local motions to achieve long-range motion estimation.

Denote $M(p)$ as a function that determines the grid cell where a feature point p belongs. $M(p)$ is determined by the frame-to-frame motion described in the previous section. The backward motion from $I(t)$ to $I(t-1)$ can thus be expressed as

$$p' = H_{M(p)}(t) \cdot p. \quad (5)$$

To obtain the position $\hat{p}$ in frame $I(t-2)$, the local homography corresponding to the grid cell enclosing the displaced feature p' is applied.

$$\hat{p} = H_{M(p')}(t-1) \cdot p'. \quad (6)$$

As illustrated in Figure 4, it is straightforward to iteratively apply the above process to propagate the motion from previous frames to a distant frame. Note that this model is particularly suitable for incremental update and online applications. Specifically, when a new frame $I(t+1)$ arrives, the previous motion models can be reused and only the warping between $I(t+1)$ and $I(t)$ needs to be derived.

5. Cross-Video Mapping

For each $I^T(m)$, we aim to: (1) seek its visually-closest frame $I^R(k)$, which maximizes some measure of spatial

(a) (b) (c)

Figure 5. An example of bridge image selection. Target image (a). GPS aligned image (b). The bridge frame identified by our method (c).

alignment quality with $I^T(m)$, and (2) establish a dense mapping between $I^T(m)$ and the identified bridge image $I^R(k)$.

5.1. Bridge Image Selection

To identify the bridge image $I^R(k)$, we first find the initial nearest neighbor $I^R(k_0)$ using GPS information and include the frames within a local temporal window centered at $I^R(k_0)$ as candidates: $\{I^R(k')|k_0 - \delta < k' < k_0 + \delta\}$. To measure the alignment quality between $I^T(m)$ and each candidate $I^R(k')$, we perform an image content similarity analysis by SIFT feature matching. Denote $\mathcal{M}(m, k')$ as the matched feature set, $\mathbf{x}$ and $\mathbf{x}'$ as a feature pair lying in $I^T(m)$ and a candidate image $I^R(k')$, respectively. The visual similarity between $I^T(m)$ and $I^R(k')$ is measured by the following score function $\Phi(m, k')$, which takes both the Euclidean distance and descriptor similarity between matched features into account, favoring matches that are close in image space as used in [17, 7, 20],

$$\Phi(m,k') = \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{M}(m,k')} e^{(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{\sigma_s})} e^{(-\frac{\|\mathbf{d}(\mathbf{x})-\mathbf{d}(\mathbf{x}')\|}{\sigma_d})}, \tag{7}$$

where $\mathbf{d}(\mathbf{x})$ denotes the SIFT descriptor of $\mathbf{x}$, and σ_d, σ_s are the average distances between feature pairs in spatial and feature domains, respectively. The final bridge image $I^R(k)$ is determined by minimizing $\Phi(m, k')$ over the candidate image set.

$$k = arg\min_{k'} \Phi(m,k'), \quad k' \in (k_0 - \delta, k_0 + \delta), \tag{8}$$

where δ is fixed to 30 in all experiments. Figure 5 demonstrates an example of selecting the bridge image by using image similarity analysis and GPS information. Based on the assumption that the speeds of the vehicles do not vary drastically, bridge image selection is performed periodically after a fixed time interval and the subsequent bridge images are chosen in chronological order to meet the computational demand of online application.

5.2. Trajectory Transfer

Although SIFT feature matching has been employed to guide the selection of $I^R(k)$ and can be readily used as the mapping between $I^R(k)$ and $I^T(m)$, performing view integration with these independently constructed match sets

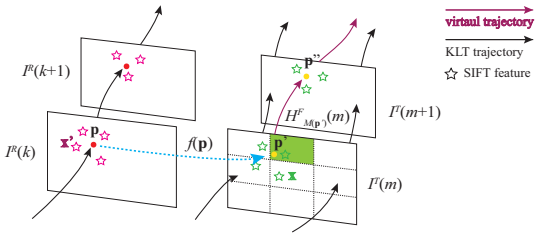


Figure 6. Trajectory transfer. For each long trajectories in I^R, we estimate a corresponding virtual trajectory in I^T by SIFT feature matches and the motion model in I^T.

will introduce temporal incoherence between consecutive frames. To tackle this problem, we describe a novel technique called *trajectory transfer* to estimate continuous feature correspondences. In a nutshell, trajectory transfer aims to associate each feature trajectory in I^R with a *virtual trajectory* in I^T, whose motion is estimated by the motion model described in Section 4.1. Similar to standard feature tracking methods, trajectory transfer is composed of a *detection* and *tracking* phase.

Detection Recall that the KLT feature trackers give us a set of feature trajectories in both I^R and I^T. For each current bridge image $I^R(k)$, the trajectories which have lengths greater than 3 and do not have a corresponding virtual trajectory are collected. Here, we exclude the short trajectories which usually occur in textureless regions and are highly unreliable. Assuming that the current position of a valid trajectory is $\mathbf{p}$ in $I^R(k)$, we initialize a corresponding virtual feature tracker by *detecting* a new virtual point $\mathbf{p}'$ in $I^T(m)$ as a weighted combination of the SIFT features $\mathbf{x}$ in $I^T(m)$ as

$$\mathbf{p}' = f(\mathbf{p}) = \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{M}(m,k)} \alpha^{\mathbf{p}}_{\mathbf{x}'}\mathbf{x}, \tag{9}$$

whose weights are defined as below,

$$\alpha^{\mathbf{p}}_{\mathbf{x}'} = e^{(-\|\mathbf{p}-\mathbf{x}'\|^2/\sigma^2)}. \tag{10}$$

The above weighting function gives higher confidence to the SIFT features $\mathbf{x}'$ in $I^R(k)$ that are closer to $\mathbf{p}$ and the scale parameter σ is defined as the average distance between $\mathbf{p}$ and $\mathbf{x}'$.

Tracking For each current target image $I^T(m)$, we have a set of virtual feature trackers with their current locations $\mathbf{p}'$. These virtual points are *tracked* by estimating the *forward* motion between $I^T(m)$ and $I^T(m+1)$ using image warping. The KLT feature trajectories in I^T provides us rich and robust information to estimate the frame-to-frame motion. Specifically, the position of $\mathbf{p}'$ is updated by $\mathbf{p}'' = H^F_{M(\mathbf{p}')}(m)\mathbf{p}'$, where $H^F_{M(\mathbf{p}')}(m)$ encodes the local motion from $I^T(m)$ to $I^T(m+1)$.

The above trajectory transfer algorithm allows us to establish spatial correspondences across two videos which are also temporally continuous and is illustrated in Figure 6.

6. View Integration

For each incoming image pair, the visual content of $I^R(n)$ is transferred to $I^R(k)$ using the aggregated motion model (Section 4.2) to form a new image $\hat{I}^R(n)$ which recover the occluded region. With the continuous point correspondences established between each $I^R(k)$ and $I^T(m)$, view integration can be accomplished by solving for the spatially-varying warps described in Section 4.1 to register $\hat{I}^R(n)$ to $I^T(m)$ to fill in the occluded region. Specifically, we use the spatial mappings generated by virtual trajectories as the data energy. We also introduce an additional *temporal regularization* constraint to those grid cells without any spatial mappings, which is defined as below,

$$E_T(\tilde{\mathcal{V}}) = \sum_i \|\tilde{v}_{i,k} - \tilde{v}_{i,k-1}\|, \tag{11}$$

where $\tilde{v}_{i,k}$ denotes the unknown mesh vertex position in $I^T(m)$ and $\tilde{v}_{i,k-1}$ is the corresponding warped position in $I^T(m-1)$. The above constraint respects the previous warping result when there are no corresponding feature trajectories available to guide the warping. By combining all the energy terms, we obtain the following energy minimization problem:

$$E(\tilde{\mathcal{V}}) = E_D(\tilde{\mathcal{V}}) + \lambda_S E_S(\tilde{\mathcal{V}}) + \lambda_L E_L(\tilde{\mathcal{V}}) + \lambda_T E_T(\tilde{\mathcal{V}}), \tag{12}$$

where $\lambda_S, \lambda_L, \lambda_T$ are set as $1, 1, 3$, respectively. Notice that standard texture mapping can then be exploited to generate the warped images given the original and deformed mesh vertices.

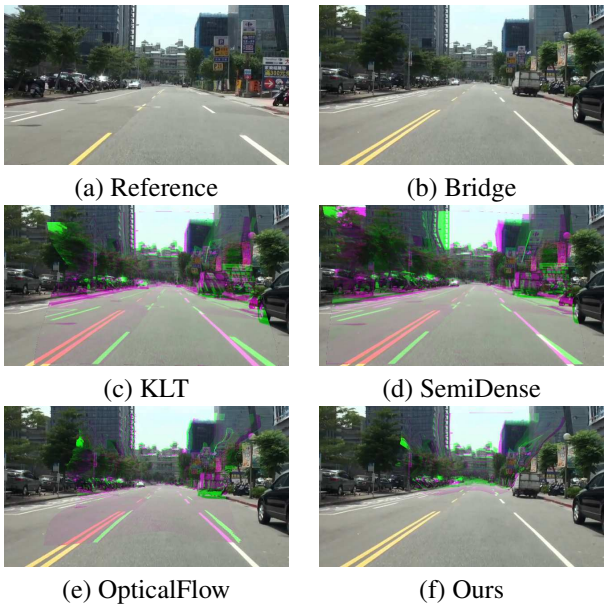
7. Experimental Results

In this section, we first compare the proposed method with several related methods, and then discuss the computation efficiency and limitations of the proposed method. See the accompanying videos for better visual comparison between the proposed method and previous methods.

7.1. Comparisons

The performance of the proposed method and various related techniques were evaluated on three data sets captured under real-world driving scenarios with different traffic conditions: HIGHWAY, CITYROAD and BRIDGE. All test sets consist of two sequences collected from a preceding and its following vehicle. These sequences exhibit large depth variations, contain dynamic moving objects and different illumination conditions and are thus very challenging for motion estimation and video alignment.

Comparison with motion estimation methods In this experiment, we compare the performance of registering the reference image $I^R(n)$ to the bridge image $I^R(k)$ by the proposed intra-video mapping method and existing state-of-the-art motion estimation methods, including trajectory-based approaches, *i.e.*, *Kanade-Lucas-Tomasi* feature tracker (KLT) [19] and *Semi-Dense Point Tracker* (SemiDense) [8], and *optical flow* (OpticalFlow) [1]. The feature correspondences obtained from KLT and SemiDense are used to compute a warped image of $I^R(n)$ by the image warping technique described in Section 4.1. Our result is obtained by the aggregative motion model described in Section 4.2. The result of OpticalFlow is obtained by copying the pixel values from $I^R(n)$ with the frame-to-frame motion flows. The registration quality is visualized by fusing the G channel of $I^R(n)$ and (R,B) channels of $I^R(k)$. As a result, the misaligned regions are highlighted with lawn-green and hot-pink colors.

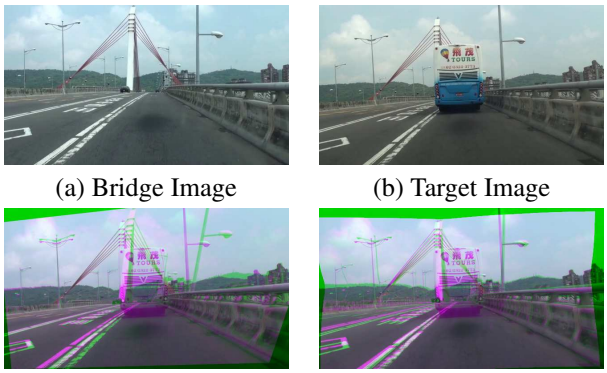


(a) Reference (b) Bridge (c) KLT (d) SemiDense (e) OpticalFlow (f) Ours

Figure 7. Comparison of various motion estimation approaches. (a) and (b) are the reference and the bridge images. (c)∼(f) shows the differences between the bridge and the warped images by each method. Misaligned regions are highlighted with lawn-green and hot-pink colors.

Figure 7 depicts the aligned images obtained from the CITYROAD data set. The result of KLT tracking (Figure 7(c)) did not well align the image pair due to the limited number of available long trajectories to guide the image warping. Although SemiDense may produce considerably more feature matches between neighboring frames, it inevitably loses many feature trajectories due to the long distance between $I^R(n)$ and $I^R(k)$ and the influence of moving objects. As a result, it also did not perform well in aligning the image pair (Figure 7(d)).

The result by OpticalFlow showed less misalignments, suggesting that the usage of dense per-pixel tracking could better account for parallax since more local warp constraints are established. However, it still suffers from the problem of



(a) Bridge Image (b) Target Image

(c) ECC (d) Ours

Figure 8. Comparison of global homography (ECC) and spatially-varying warping model (Ours) to perform cross-video alignment. Obviously, spatially-varying model can better account for parallax.

(a) Reference image (b) Target image (c) Our result

(d) Reference image (e) Target image (f) Our result

Figure 9. An example of view integration at road intersection.

drifting and error propagation, thus misalignment still presented (Figure 7(e)). Figure 7(f) demonstrates the result by intra-video mapping with considerably less misalignment. The warping-based motion model encodes the per-pixel movement within a mesh grid by a local homograhy estimated from robust frame-to-frame feature correspondences. In addition, the local motions in textureless regions lacking of sufficient features are regularized by mesh optimization. Aggregated together, intra-video mapping achieves robust long-range motion estimation, resulting in superior performance over both sparse and dense tracking methods.

Comparison with global homography model We compare the performance of aligning bridge and target images by the proposed cross-video mapping and a global homography-based approach, Enhanced Correlation Coefficient model (ECC) [7]. Figure 8 shows the warped results between the bridge and the target images from the BRIDGE sequence. Similarly, the alignment quality is visualized by image fusion. Apparently, a single global homography is not sufficient to model the parallax between two dashcams. Our method can better handle parallax by using a spatially-varying warping model.

Comparison with state-of-the-art methods The proposed inter-video mapping is compared with two baseline view integration systems implemented by state-of-the-art algorithms. The first baseline (HV-MDLT) performs robust feature matching [2] and image stitching [23] between the reference and target images. The second baseline system (KLT-MDLT) [3] employs two-pass image warping and stiching [23] mediated by a bridge image similar to ours, which is selected by using only GPS information.

Figure 10 depicts view integration results from the HIGHWAY, BRIDGE, CITYROAD sequences. HV-MDLT did not align images well due to the mismatches caused by large viewpoint changes, different lighting conditions and the presence of occlusion. In the case of KLT-MDLT, it suffers from the lack of long feature trajectories thus significant misalignment remains. Its performance degrades in textureless regions, such as ground since there are too few features available to guide the warping. As discussed earlier, inter-video mapping exploits frame-to-frame correspondences to establish dense and continuous mapping across videos. It achieves more plausible spatial alignment as shown in Figure 10(e) and temporal coherence of the synthesized sequences. See the highlighted regions in Figure 10 and the accompanying videos to compare the performance differences. Some more view integration results are shown in Figure 9.

7.2. Computational time

Our approach is implemented in C++ with the OpenCV and SiftGPU[1] libraries, and run on a PC with Intel $i7$ 3.4 GHz processor, 16G RAM and a GeForce GTX 750 graphics card. All the test sequences are of 640×360 resolution and the meshes used in our motion models are fixed to 40-by-40 grids. Due to the nature of incremental aggregation, intra-video mapping is very efficient and achieves 27 fps. SIFT keypoint detection runs in near real-time speed in GPU. Bridge image selection is computationally intensive since it involves extensive SIFT feature matching against a set of candidate images. It currently runs in about 1 fps but does not need to be performed for each frame.

7.3. Limitation

Our method relies on accurate moving object feature detection, which remains a very challenging problem in computer vision. The inconsistent motion from features not belonging to the static scene will typically result in corrupted meshes when estimating the motion model. Epipolar constraints [11] did not work well for all types of scene in our test cases. Although the heuristic described in Section 3.2 assisted us to eliminate most features lying on moving objects, it still cannot handle dynamic objects with similar motion as the static scenes. Our method is not suitable for curved roads due to the substantial change in viewing direction. The occluded region in $I^T(m)$ may not be seen in

[1] http://cs.unc.edu/∼ccwu/siftgpu/

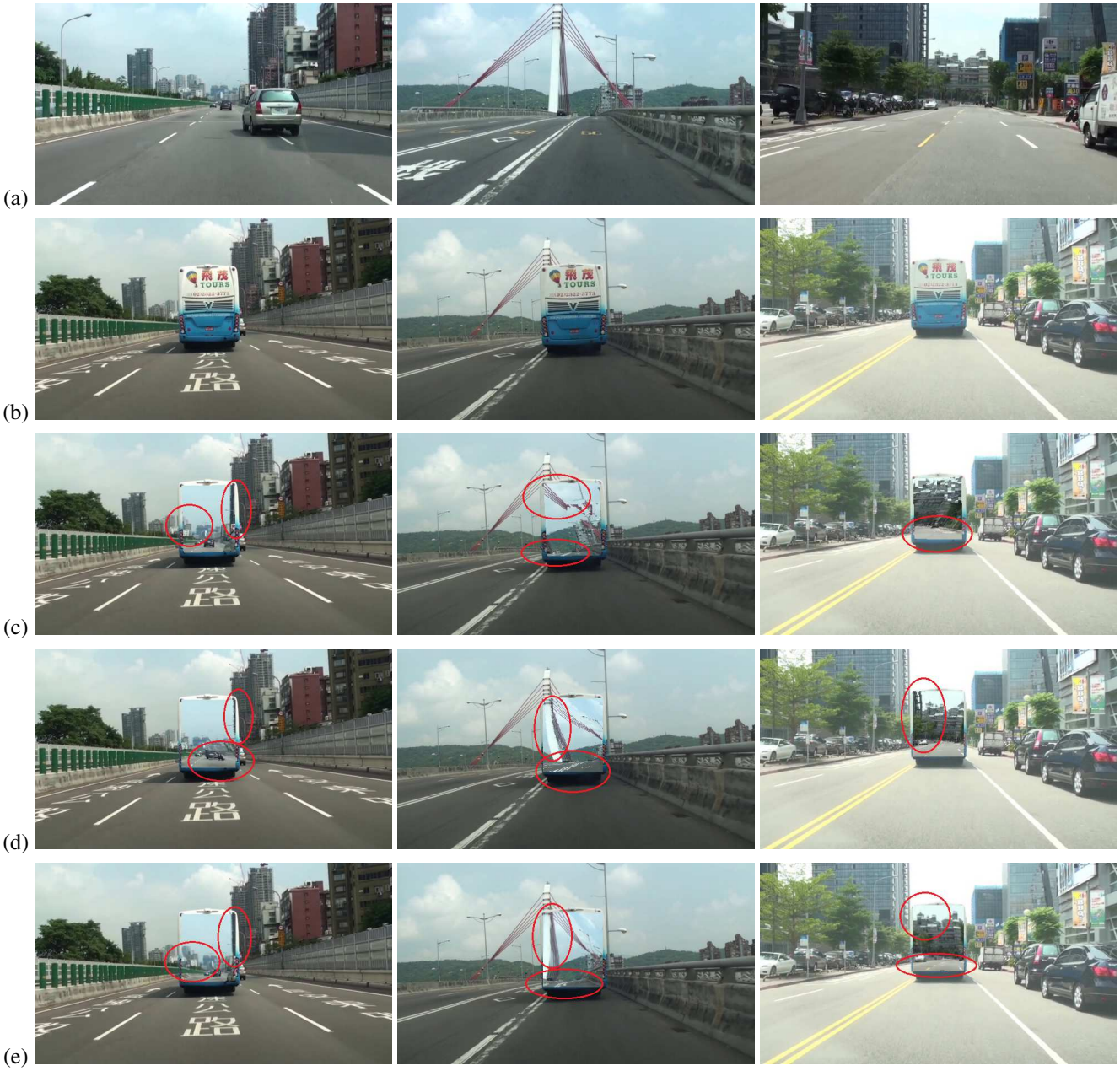


Figure 10. (a)(b) shows three examples of reference and target image pairs from the HIGHWAY, BRIDGE and CITYROAD sequences, respectively. The composite images by HV-MDLT, KLT-MDLT and the proposed method are shown in (c)∼(e), respectively.

$I^R(n)$ and little visual content can be transferred to unveil the occlusion.

8. Conclusion and future work

In this paper, we presented an inter-video mapping method for view integration between two dashcams. The proposed method can model long-range motions and thus applicable to videos with considerable depth variations. It is efficient based on incremental motion estimation and produces temporally smooth results without resorting to spatio-temporal optimization. For future work, we would like to investigate the possibility of integrating views from multiple reference sequences.

Acknowledgment We thank Hsin-Mu Tsai and his team for supporting data collection. This work was supported in part by Ministry of Science and Technology, Taiwan, National Taiwan University and Intel Corporation under Grants MOST103-2911-I-002-001, NTU-ICRP-104R7501, and NTU-ICRP-104R7501-1.

References

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. Eur. Conf. Comput. Vis.*, pages 25–36, 2004.

[2] H.-Y. Chen, Y.-Y. Lin, and B.-Y. Chen. Robust feature matching with alternate hough and inverted hough transforms. In *Proc. Conf. Comput. Vis. and Pattern Recognit.*, pages 2762–2769, 2013.

[3] S.-C. Chen, H.-Y. Chen, Y.-L. Chen, H.-M. Tsai, and B.-Y. Chen. Making in-front-of cars transparent: Sharing first-person-views via dashcam. *Computer Graphics Forum*, 33(7):289–297, 2014.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Conf. Comput. Vis. and Pattern Recognit.*, pages 886–893, 2005.

[5] F. Diego, D. Ponsa, J. Serrat, and A. M. López. Video alignment for change detection. *IEEE Trans. on Image Processing*, 20(7):1858–1869, 2011.

[6] F. Diego, J. Serrat, and A. M. López. Joint spatio-temporal alignment of sequences. *IEEE Transactions on Multimedia*, 15(6):1377–1387, 2013.

[7] G. Evangelidis and C. Bauckhage. Efficient subframe video alignment using short descriptors. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 35(10):2371–2386, 2013.

[8] M. Garrigues, A. Manzanera, and T. M. Bernard. Video extruder: a semi-dense point tracker for extracting beams of trajectories in real time. *Journal of Real-Time Image Processing*, pages 1–14, 2014.

[9] P. E. R. Gomes, F. Vieira, and M. Ferreira. The see-through system: From implementation to test-drive. In *Proceeding of IEEE Vehicular Networking Conference 2012*, pages 40–47, 2012.

[10] R. Grompone, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 32(4):722–732, 2010.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[12] Y.-H. Huang, T.-K. Huang, Y.-H. Huang, W.-C. Chen, and Y.-Y. Chuang. Warping-based novel view synthesis from a binocular image for autostereoscopic displays. In *Proceedings of IEEE International Conference on Multimedia and Expo 2012*, pages 302–307, 2012.

[13] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung. Video stabilization using robust feature trajectories. In *Proc. Int'l Conf. Comput. Vis.*, pages 1307–1404, 2009.

[14] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3D video stabilization. *ACM Trans. on Graphics*, 28(3):44:1–44:9, 2009.

[15] S. Liu, L. Yuan, P. Tan, and J. Sun. Bundled camera paths for video stabilization. *ACM Trans. on Graphics*, 32(4), 2013.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, 2004.

[17] P. Sand and S. J. Teller. Video matching. *ACM Trans. on Graphics*, 23(3):592–599, 2004.

[18] P. Sand and S. J. Teller. Particle video: Long-range motion estimation using point trajectories. In *Proc. Conf. Comput. Vis. and Pattern Recognit.*, pages 2195–2202, 2006.

[19] J. Shi and C. Tomasi. Good feature to track. In *Proc. Conf. Comput. Vis. and Pattern Recognit.*, pages 593–600, 1994.

[20] O. Wang, C. Schroers, H. Zimmer, M. H. Gross, and A. Sorkine-Hornung. VideoSnapping: Interactive synchronization of multiple videos. *ACM Trans. on Graphics*, 33(4):77:1–77:10, 2010.

[21] Y.-S. Wang, H. Fu, O. Sorkine, T.-Y. Lee, and H.-P. Seidel. Motion-aware temporal coherence for video resizing. *ACM Trans. on Graphics*, 28(5):127:1–127:10, 2009.

[22] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. on Graphics*, 27(5):118:1–118:8, 2008.

[23] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving DLT. In *Proc. Conf. Comput. Vis. and Pattern Recognit.*, pages 2339–2346, 2013.

[24] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Proc. Int'l Conf. Comput. Vis.*, pages 1838–1845, 2012.