Title
Dense Spatial Pyramid Mesh Warping for Registering Moving Cameras in 3D Scene Map

Author
Yang, Jiadi

Publication Date
2015

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Dense Spatial Pyramid Mesh Warping for Registering Moving Cameras in 3D Scene Map

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Jiadi Yang

2015

Abstract of the Thesis

Dense Spatial Pyramid Mesh Warping for Registering Moving Cameras in 3D Scene Map

by

Jiadi Yang

Master of Science in Computer Science

University of California, Los Angeles, 2015

Professor Song-Chun Zhu, Chair

We propose a robust multi-modal method for automatically registering a moving camera (e.g. mounted on a robot) in 3D scene map. Our approach takes advantages of both Global Positioning System (GPS) and visual sensors to obtain high-precision geographic location for a moving camera at each time. The proposed method distinguishes from past works in the following three aspects: i) we introduce a spatial pyramid mesh warping method to obtain dense correspondences between consecutive frames, which can be used to remove unexpected camera motion for robust registration ; ii) we introduced a robust feature tracking method to tracking feature points in consecutive frames; and iii) we utilize a continuous polynomial function to describe camera motion w.r.t time, which can be solved by minimizing the errors of interpolating both visual observations and GPS locations. We evaluate the proposed method on a set of challenging videos for both stabilization and registration tasks. Results with comparisons to other popular methods showed that our method is capable of achieving high-quality results under various challenges, e.g. lighting changes, motion blurs, scene noises etc.

The thesis of Jiadi Yang is approved.

Ying Nian Wu

Demetri Terzopoulos

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2015

Table of Contents

List of Figures

Acknowledgments

First, I would like to express my sincere gratitude to my advisor Prof. Song-Chun Zhu for the support of my study and research, for his patience, enthusiasm and immense knowledge. I would not have been able to finish the thesis without the help and advice from my thesis committee members.

I would like to thank my fellow teammates in UCLA Center for Vision, Cognition, Learning and Art. In particular, I want to extend my gratefulness to Xiaobai Liu, who led me into this project and provided inspiring discussions and help with his rich experience.

I would also like to thank my parents for supporting me throughout my life. Without their love and encouragement I could not have finished the thesis.

CHAPTER 1

Introduction

Accurately registering a moving sensor in 3D scene map is capable of providing scene context for high-level video understanding tasks, e.g. behavior analysis, action recognition etc. The basic objectives are two folds: registering a video frame geographically and estimating camera parameters (e.g. viewing angles) at each time. Traditional methods utilize optical flow algorithm [LYTa] [IMH05] to estimate correspondences between interest points in consecutive frames, from which relative camera motions can be estimated. Although impressive results achieved, the geo-registration problem still remain challenges, especially for noisy scenarios, because of two major issues: i) for each pair of frames, it is difficult to get accurate feature correspondences while encountering motion blurs, low-resolution, or frequent foreground intersection; ii) while the errors are accumulated over several frames, the trajectory estimation shall drift with arbitrarily large errors.

In this work, we develop a robust method to address the above-mentioned issues. Figure. 1.1 (a) shows a frame from the video sequence captured by a moving camera. Figure. 1.1 (b) plots the estimated camera trajectories and the Field of View (FOV) on the map. The original GPS points (smoothed and interpolated) are plotted for comparisons.

One crucial step of our method is to stabilize input videos captured by a moving camera, e.g. being mounted on a bicycle or a robot, i.e. to remove the high-frequency camera motion. These unconscious movements bring in motion blurs and ambiguities which challenges the later visual measurement steps. In this work, we propose a stabilization algorithm based on spatial pyramid mesh warping, which first generates frame-to-frame correspondences between interest points, and then smooths camera trajectory

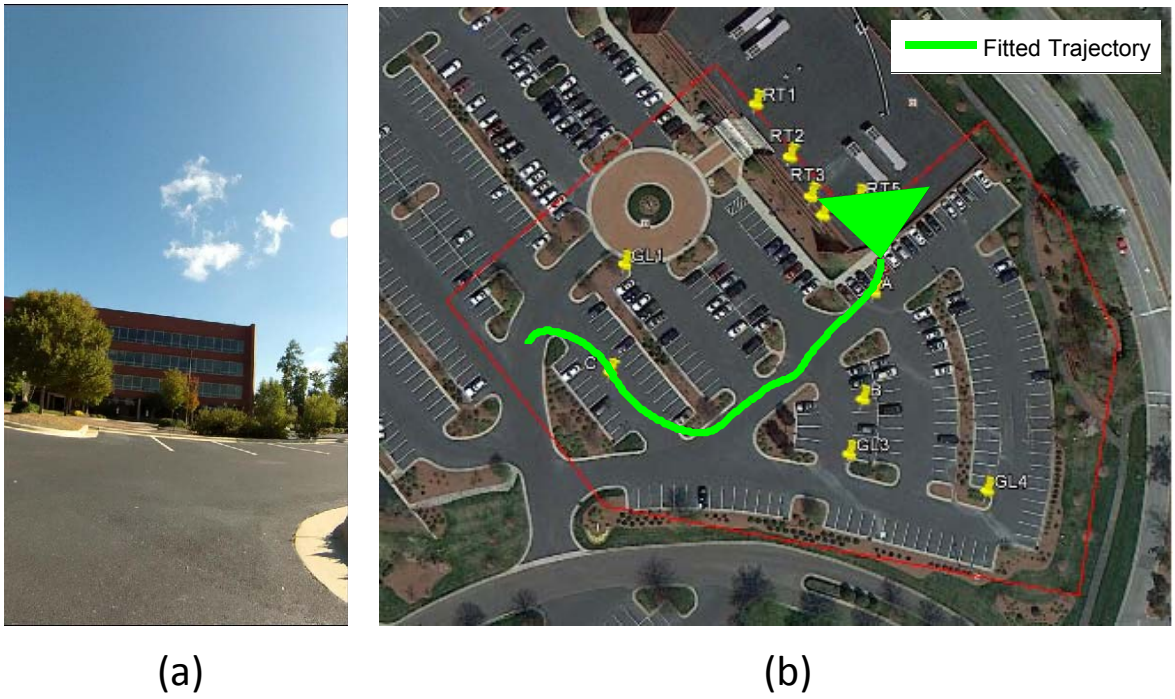

Figure 1.1: Registering moving cameras in 3D scene map. (a) input video frame captured by a moving camera (mounted on a bike); (b) scene map overlaid with estimated camera trajectory.

over time to get stabilized video frames. Our warping algorithm starts with detecting and matching interest points, based on which we warp uniform grid mesh in one frame into the next frame such that local shapes (e.g. triangles) are preserved. Moreover, we perform mesh warping over multiple scales simultaneously and impose cross-scale consistency. To the end, this multi-resolution strategy is able to i) exploit both short-range and long-range information to preserve visual content; and ii) automatically deal with local regions without interest points detected. We shall demonstrate that our method can work well against scenes noise, motion blurs, repetitive patterns and other challenges.

To address the drifting issue, we introduce a multi-modal method to exploit both GPS locations and visual observations. Despite being reliable to access, GPS locations are usually noisy, sparse (available at a time-step of a few seconds), with large errors (up to 10- meters on average) and it might not work for certain scenarios (e.g. with occlusions, close to building etc.) or bad weathers. On the other hand, we could measure relative

camera motion from cross-frame correspondences which are continuous, less sensitive to environment (e.g. weather), or other challenges. Therefore, it is natural to fuse both cues for robust camera geo-registration. In this work, we describe a camera trajectory as a polynomial piece-wise smoothing function of camera position over time, and introduce a robust fitting method to interpolate the synchronized geographical points and measured camera positions. We shall demonstrate that the fusion of two modalities could obtain better geo-localization results than utilizing individual modality.

The rest of this paper is organized as follows. In Chapter 2, the relationships to the previous literature are discussed and summarized. In Chapter 3 we first introduce a robust video stabilization method and then present the proposed camera registration method. In Section 4.3 we evaluate the proposed method on challenging videos for both video stabilization and 3D registration of moving cameras. In Chapter 5 we summarize this work by discussing the limitations of our approach and remarking the future directions.

CHAPTER 2

Related Works

This work is closely related to three research streams in computer vision and graphics.

Video-Stabilization or motion compensation aims to remove high-frequency camera motion, or unexpected image motion caused by unintentional move of the camera itself. The past efforts can be divided into two categories: 2D methods and 3D methods. The first category of algorithms directly estimates 2D frame-to-frame transformations. Igarashi et al. [IMH05] proposed a as-rigid-as-possible formulation for warping images to encourage each triangle in the original mesh to undergo a rigid transform (rotation+shift). This pioneering work was extended in different aspects, including moving-least-squares method [SMW], matrix form [ZCH09] and as-similar-as possible [LYTb], and applied in image re-targeting [ZCH09], video re-targeting [MOG06], panoramas and content-preserving rotation [HCS12]. Recent works proposed to warp grid mesh such that local shapes ,e.g. straight lines [CAA10] [CC12], triangles [LGJ], are preserved in the new mesh. Chen et al. [CLH08] used polynomial curves to describe camera motion. Grundmann et al. [GKC12] used a bundle of homograph to describe camera motion and formulate motion estimation as a L_1-norm minimization problem [GKC12]. Yasuyuki et al. [MOG06] proposed a method for filling missing image area while stabilizing video jitter videos. The proposed method belongs to this category and follows the shape-preserving methodology. In contrast, we perform mesh warping in multiple scales and enforce cross-scale consistency which can access both long-range and short-range information.

The second category of stabilization methods aims to recover 3D camera motion which usually requires a full 3D reconstruction of the scene and the camera trajectory. Buehler et al. [BBM01] proposed to utilize structure-from-motion method for estimating camera

motion. Fitzgibbon et al. [FWZ05] proposed to render novel viewpoints along camera trajectories based on the input video frames. These 3D methods can achieve high-quality stabilization results. But they usually require long-term tracking of feature points and are sensitive to scene noises. Liu et al.[LGJ] built a robust system to utilize both 2D estimations and 3D reconstruction and achieved impressive results.

Visual Odometry is to measure relative camera motion from visual frames. The key step of VO is to match keypoints (e.g. SURF [BTG08] or SIFT [LYTa]) in consecutive frames. Recently there was significant improvement in visual odometry with development of real applications, e.g. self-driving cars. In particular, Zhang and Singh [ZS14] proposed to measure visual odometry from lidar sensors in real-time and achieved the state-of-the-arts results on public benchmark KITTI [GLU12]. Badino et al. [BYK13] introduce a stereo system that integrates feature correspondences between multiple frames instead of a single frame to improve measurement accuracy. Geiger et al. [GZS11] and Song et al. [SC14] applied Structure-from-motion method over monocular video sequences to estimate camera motion. The later work [SC14] proposed to estimate the ground plane and its scale change while a vehicle-mounted camera is moving. In contrast with stereo system, monocular videos are usually more challenging because of the drifting issue, which is however the commonly used environment in real applications. In this work, we work on monocular videos and propose to fuse Visual Odometry with noisy GPS locations to avoid drifting issues.

Geo-localization of a moving intelligent platform is usually achieved by Simultaneous Localization and Mapping (SLAM) method or its variants [SSS] [LSC02]. Recently there has been another stream [WKR] [Dav] [SBS07] [VZS12] [ZS10] that utilize geo-tagged scene images to help localize a moving sensor. Although great successes achieved, such a system suffers from two limitations: i) it is very time-consuming to prepare the geo-tagged images; ii) the current view of the sensor might be arbitrarily different from the pre-store images due to the environment changes. Another shortcoming of these methods is the low-precision of geo-localization (up to few meters, or even worse than GPS). In this work, we proposed to localize a moving platform with meter-level precision. To do

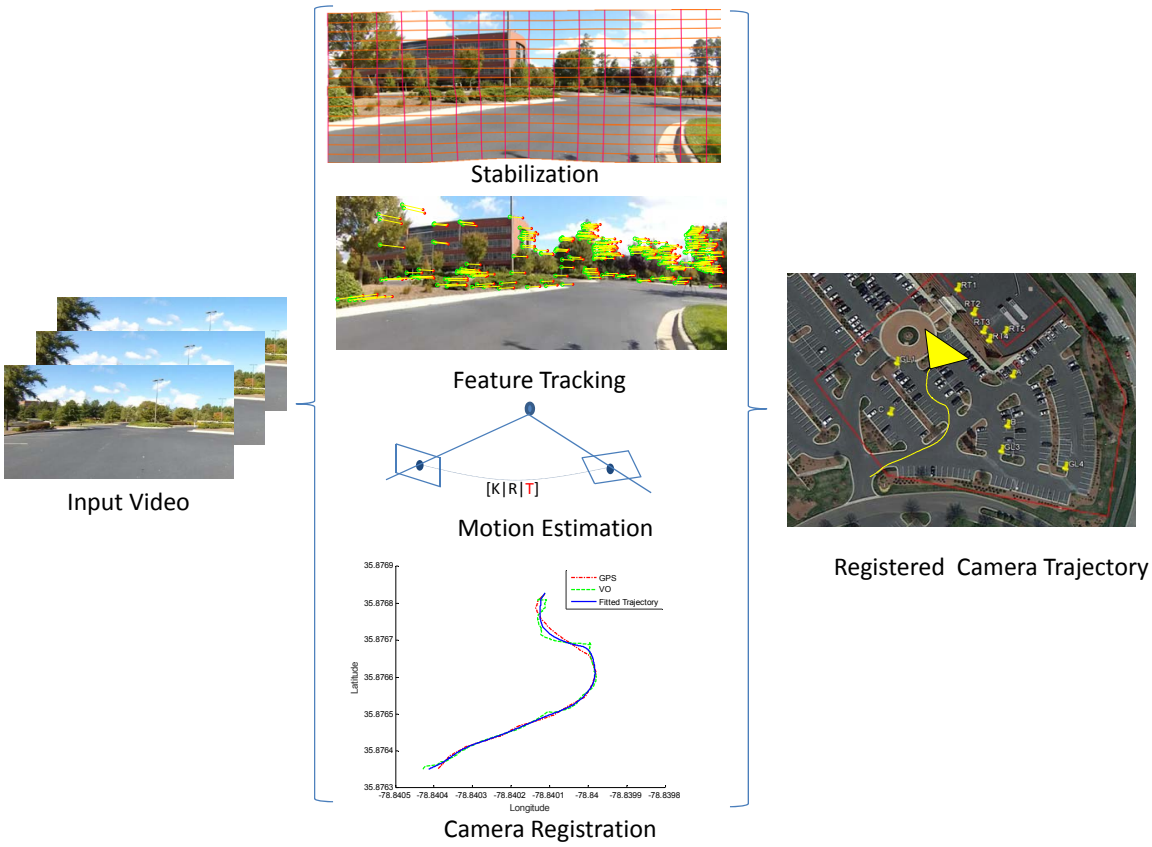

Figure 2.1: Flowchart of our approach. Input: video sequence captured by a moving camera; Output: registered camera trajectory in 3D scene map. The proposed algorithm includes four major steps: stabilization, feature tracking, motion estimation, and camera registration.

this, we utilize both GPS locations and visual odometry (robustly estimated) for robust inference.

CHAPTER 3

Geo-Registration of Moving Cameras

In this chapter, we first introduce the pipeline of our approach and then describe in details each step.

3.1 Overview of Our Approach

Fig. 2.1 summarizes the sketch of the proposed registration algorithms. It takes monocular video sequence and noisy GPS locations as inputs and estimates accurate camera position in 3D scene map at each time-step. Our method starts from detecting interest points (e.g. SURF) and proceeds the following four major steps:

- *Video Stabilization* is used to remove high-frequency camera motion which is usually generated, taking hand-held cameras for instances, by unconscious shaking or movement. This step works as a pre-processing for the 3D camera geo-registration.
- *Feature Tracking* We tracking each detected points into the next frame to build correspondences. We formulate this task as an energy minimization problem and solve it efficiently using belief propagation (BP) algorithm
- *Motion Estimation* We apply the standard Visual Odometry algorithm [Nis03] over frame-to-frame correspondences to estimate camera movement at each time-step.
- *Camera Registration* We first register the estimated continuous camera motion to the noisy discrete GPS locations by computing a similarity deformation. Then we introduce a piece-wise smoothing function to describe the camera 3D location changes over time. In particular, we use B-Spline in this work to interpolate the

warped camera motion and GPS locations simultaneously.

We conduct the above-mentioned components sequentially to obtain the camera geo-location. In practice, since the video-stabilization step also needs feature correspondences, we can iterate the first two steps multiple times to boost system performance. In this work, we simply run the whole pipeline once.

3.2 Spatial Pyramid Mesh Warping for Video Stabilization

We propose a spatial pyramid mesh warping method to represent the motion between two consecutive frames, which can be used for stabilizing video sequence [LYTb] captured by a moving camera. At each frame, we detect interest feature points and match them into next frames. As in [LGJ], we extract camera motion by warping uniform grid mesh, instead of features, between consecutive frames. Each feature point shares the same homograph determined from the motion of the four enclosing vertices. This motion model makes a tradeoff between global homograph and per-pixel optical flow.

Different from past works that define a uniform grid mesh, in this work, we introduce a spatial pyramid mesh as illustrated in Fig. 3.1. Each grid cell is divided into four cells in a recursive way. The camera motion is thus described by the cross-frame deformation of the grid mesh at each scale while preserving cross-scale consistency. Note that encoding motion in spatial pyramid is crucial since some of the cells (at a certain scale) might not include sufficient features.

Given two consecutive frames, We estimate the camera motion by minimizing three energy terms: a data term for matching appearance features; a regularization term of shape-preserving constraint (as-similar-as-possible [IMH05]) and a consistency constraint across different scales. The former two terms are defined over individual scales while the last term is defined to enforce constraints across scales.

Data Term At each scale s, let $(p, q) \in \Upsilon^s$ denote the matched feature pair from the frame t to t+1. We rewrite p as the bilinear interpolation of the four vertices of the enclosing grid cell: $p = V_p w_p$, where the matrix V_p include four cell coordinates as

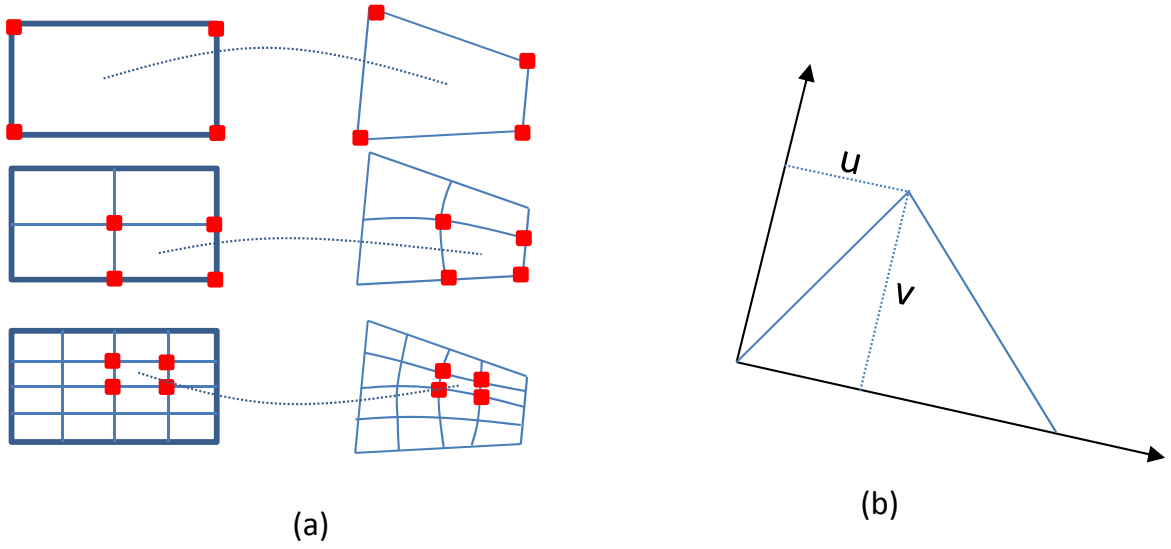

Figure 3.1: Mesh Grid for video stabilization. (a) spatial pyramid of a cell grid; (b) a vertex in the warped mesh grid is represented in the local coordinate system (u,v) of its opposite edge .

column, and whose w_p is a encoding vector and $\sum_i w_p(i) = 1$. The feature q should share the same encoding coefficients as p once the grid V_p is warped into the next frame. Let $\bar{V}_p$ denote the warped grid, the data term is defined as:

$$F^d(\bar{V}^s) = \sum_{(p,q)\in\Upsilon^s} \|q - \bar{V}_p w_p\|^2 \tag{3.1}$$

where $\bar{V}^s$ pools all the warped mesh vertices at the scale s.

Shape-Preserving We utilize the shape-preserving term in [LGJ] that requires a triple of neighbor vertices follow a similarity transformation. In particular, each vertex is represented in a local image coordinate system formed by the vector between the other two vertices and its orthogonal vector. Fig. 3.1 (b) illustrates such a coordinate system. Let $(i, j, k) \in \Lambda^s$ index three vertices that form a triangle in the mesh at the scale s, we first calculate the local coordinates u_i, v_i in the original mesh such that:

$$p_i = p_j + u_i(p_k - p_j) + v_i R_{90}(p_k - p_j) \tag{3.2}$$

where $R_{90} = [0, 1; -1, 0]$. We expect the warped vertices q_i share the same local coordi-

nates as p_i and thus have the following shape-preserving term:

$$F^r(\bar{V}^s) = \sum_{(i,j,k)\in\Lambda^s} \|q_i - q_j + u_i(q_k - q_j) + v_i R_{90}(q_k - q_j)\|^2 \tag{3.3}$$

Note that for each vertex, we sum the above shape-preserving term over all eight triangles for robust inference against noises or occasional errors.

Cross-scale Consistency The warped grid vertices at different scales should be consistently solved as illustrated in Fig. 3.1 (a) where the four vertices in the top row are inherited by the other two scales. Let $(i, j) \in Q$ denote a pair of vertices that locate on the same position yet on different scales in the original mesh. We define the cross-scale consistency as:

$$F^c(\bar{V}) = \sum_{(i,j)\in Q} \|q_i - q_j\|^2 \tag{3.4}$$

Thus, we integrate Eq.s (3.2) (3.3) (3.4) to form the following quadratic function to minimize:

$$\arg\min_{\bar{V}} \sum_s F^d(\bar{V}^s) + \lambda^r F^r(\bar{V}^s) + \lambda^c F^c(\bar{V}) \tag{3.5}$$

which can be solved analytically. In this work, we use three scales of mesh: $16 \times 16, 32 \times 32, 64 \times 64$.

Note that each feature point will have different encoding coefficients at different scales since the encoding vertices are different. In this way, at the coarser scale, the warping of a grid pool feature matches from longer range than that at the finer scale. Enforcing consistency across scales will lead to adaptively fuse information from both long and short ranges. After jointly solving the three scales of mesh, we simply use the mesh at the finest scale to stabilize video frames as in [LGJ].

To make the motion estimation more robust, we apply RANSAC [FB81] to fit a global affine transformation between two frames and discard the outliers. This helps to get rid of the fast-moving foreground objects and mismatched feature points.

After having the warped grid vertices, we estimate local homography $H_i(t)$ in each

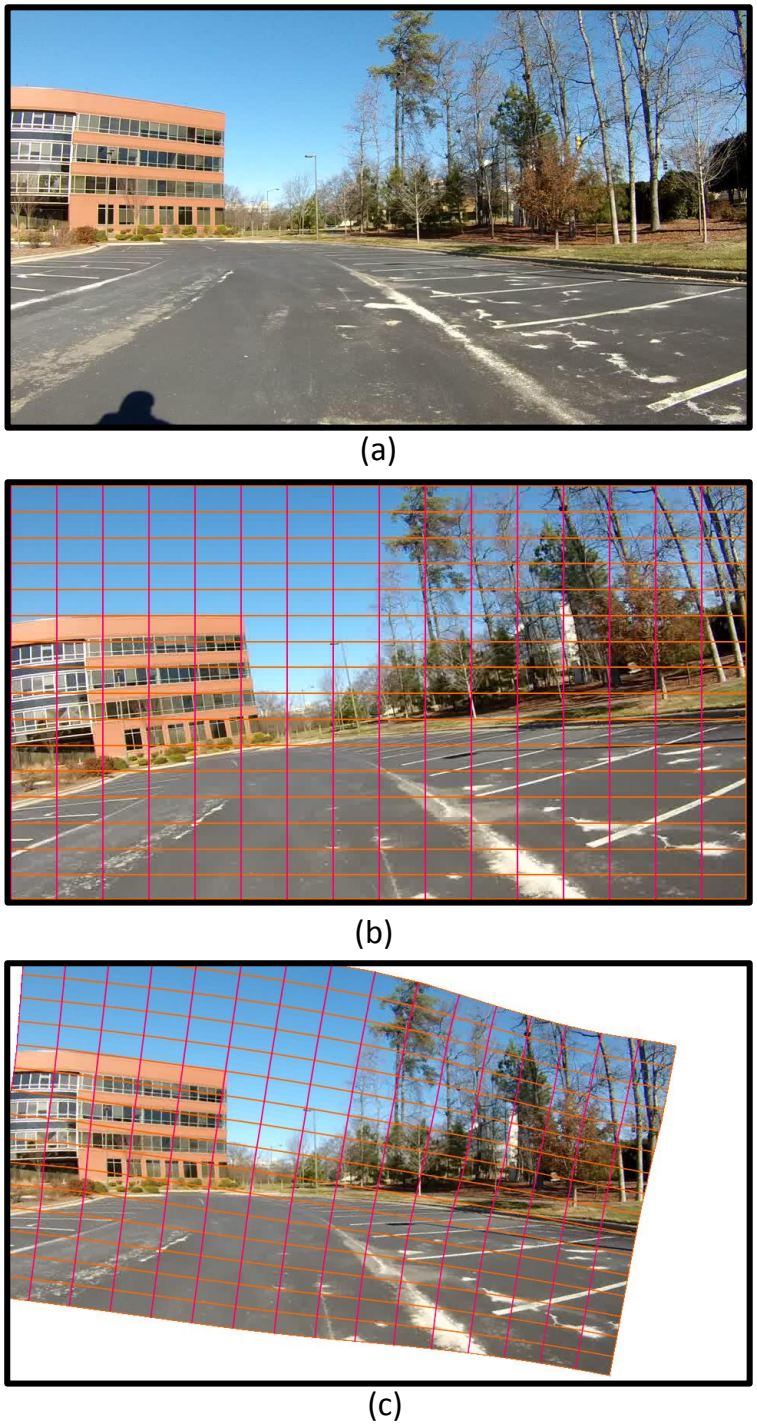

Figure 3.2: Warping to previous frame. (a) frame t; (b) frame $t+1$; (c) frame $t+1$ warped by the estimated local homography (contents outside frame t have been cropped).

grid cell i of frame t by solving the linear equation:

$$\bar{V}_i = H_i(t)V_i \tag{3.6}$$

where V_i and $\bar{V}_i$ are the four grid vertices before and after the warping.

Fig 3.2 shows the warping result. For more obvious visual effect, we choose frame $t+1$ and frame t with a difference of 1 second in time and display the mesh at the scale of 16×16. Note how Fig 3.2 (c) is aligned with Fig 3.2 (a).

3.3 Robust Feature Tracking

Once video frames are stabilized, we extract local SURF (Speeded Up Robust Features) points [BTG08] in a frame and track them into the next frame. SURF descriptor is a sparse feature detection and feature extraction. In this work, we set the strongest feature threshold to be 1000, the number of octave to be 3, and the number of scale levels to be 4.

We formulate feature tracking as an energy minimization problem.

Let I^t, I^{t+1} denote two consecutive frames, i indexes SURF points in I^t, and (x_i, y_i) is the coordinate of the i^{th} point. Let $(i,j) \in \varepsilon$ denote two neighbor points. Our goal is to estimate a motion field $(\Delta x_i, \Delta y_i)$ such that the following objectives are maximized:

$$\begin{aligned} E(\{\Delta x_i, \Delta y_i\}) = & \sum_i \|I^t(x_i, y_i) - I^{t+1}(x_i + \Delta x_i, y_i + \Delta y_i)\|^2 \\ & + \lambda \|(\Delta x_i, \Delta y_i)\|^2 + \beta \sum_{(i,j)\in\varepsilon} [\|\Delta x_i - \Delta u_j\| + \|\Delta y_i - \Delta v_j\|] \end{aligned} \tag{3.7}$$

which includes three terms: appearance discrepancy, displacement and spatial smoothness regularization. Similar method has been used in [LYTa] to obtain dense correspondences between partially overlapped images. In contrast, we aim to track interest points over time while preserving local spatial geometry. We adopt the Loopy belief Propagation algorithm to optimize Eq. (3.7) [SZS08].

3.4 Motion Estimation

We briefly review the Visual Odometry (VO) algorithm [Nis03] that includes three major steps. First, we compute the essential matrix from feature correspondences using epipo-

lar constraints between two frames. The essential matrix contains the camera motion parameters up to an unknown scale factor for the camera translation. The minimal case solution involves five correspondences and we use the efficient implementation proposed by Nister [Nis03]. Second, we extract rotation matrix and translation (up to an unknown scale) from the estimated essential matrix. With orthogonal constraints over the rotation matrix, there are in general four decompositions for one essential matrix yet only one of them is plausible (i.e. all points are in front of the camera). Nister et al. proposed an efficient way to identify the correct decomposition. Third, we compute relative scale and rescale the translation accordingly.

Note that the re-scaling step requires additional assumption (e.g. fixed camera height and pitch angle) or calculation (e.g. estimating 3D positions of interest points), which are not reliable in practice. In the following, we shall introduce a registration method which can work well without knowing the scale factor and making those assumptions.

3.5 Camera Registration

We register visual odometry with GPS locations over time by computing a similarity transformation (i.e. rotation, scale, translation) between these two types of trajectories. Let M denote the similarity transformation matrix, $\bar{x}_i, \hat{x}_i$ denote homogeneous coordinates of the VO point and the GPS location at time i, respectively. Thus, we have the following least square problem:

$$\arg\min_{M} \|M * \bar{x}_i - \hat{x}_i\|^2 \tag{3.8}$$

The matrix M is used to register the estimated VO point at each time step with the GPS locations which are not available at each time.

Fig 3.3 plots GPS points in red and the warped VO points in red. As shown, the two shape are complementary to each other and the broken segments missed by GPS are recovered by VO points. Note that the warped VO shape still has significant deviations with the GPS shape, especially in areas of complex background structure (e.g. repetition patterns), and thus we need a robust method to fuse these two shapes.

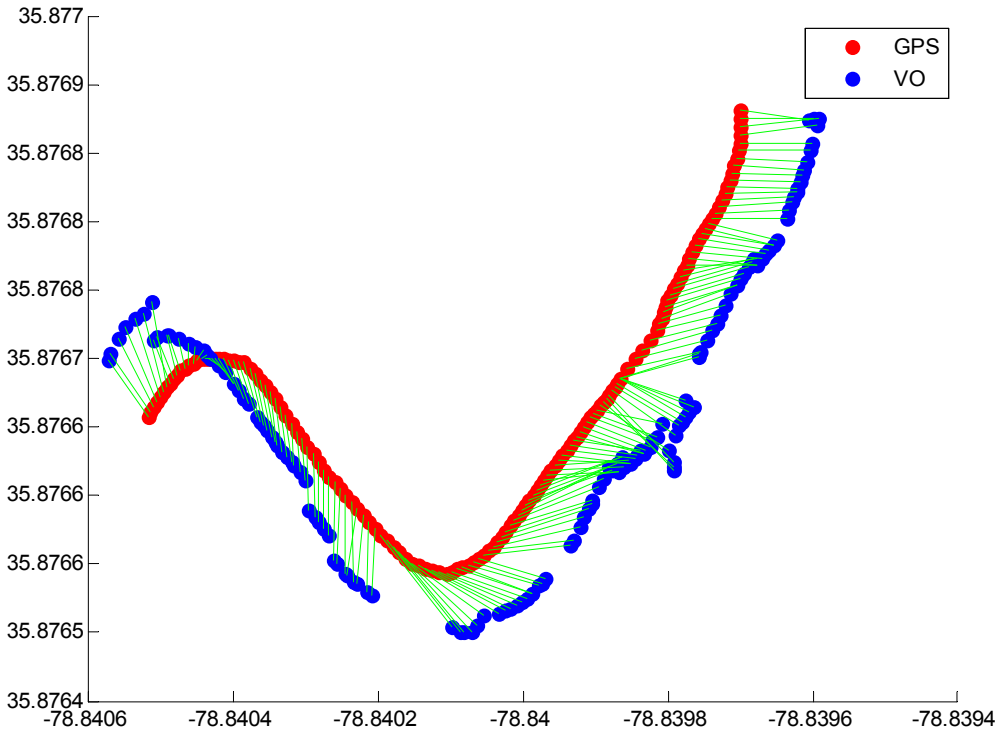

Figure 3.3: Warped Visual Odometry(VO). We linearly interpolate the GPS locations (in red) and compute similarity transform to the VO points (in blue, only matched points are plotted).

Our method represents the camera trajectory as a continuous polynomial smoothing function of 3D positions w.r.t time $\tau : t \rightarrow x_t$. In particular, we consider B-Spline whose first derivative and second derivative are both continuous. These high-order continuous constraints are used to enforce smoothness over the desired camera trajectory. Let d denote the order of B-Spline and we set $d = 3$ in this work. Let $B_l(t)$ denote the quadratic basis function, the spline function $\tau(t)$ can be written as a linear combination of basis functions:

$$\tau(t) = \sum_l \alpha_l B_l(t) \tag{3.9}$$

where the basis functions B_l can be directly obtained given time interval and order d.

We use one single spline function to interpolate both the warped VO point and the

original GPS location at each time-step, formulated as follows:

$$\arg\min_{\{\alpha_l\}} \sum_l \sum_t \|M * \bar{x}_t - \alpha_l B_l(t)\|^2 + \gamma \|\hat{x}_i - \alpha_l B_l(t)\|^2 \tag{3.10}$$

where γ is a constant. In order to address noises and errors, we utilize the RANSAC method to alternately estimate inlier points and the parameter α_l.

CHAPTER 4

Experiments

4.1 Video Dataset

We test our method on our own dataset of videos taken from a parking lot. The dataset includes videos taken from a driving car and videos taken from a riding bike together with the GPS position. We test our pipeline on segments of these videos when the vehicle is moving. Each video segment lasts for about 30-60 seconds with a frame rate of 30. Bike videos have a 1080×1920 resolution. Car videos have a 1920×1080 resolution. Videos taken from cars have their lower part occupied by the car's front hood so we crop out the lower 1/3 of the frames. The GPS coordinates are recorded at 1Hz.

4.2 Implementation

We first calculate the spatial pyramid mesh warping based on tracked feature correspondence. Each frame is then warped to a smoothed path to get the stabilized video. We do visual odometry on the stabilized video to get the camera trajectory. Then the camera trajectory is aligned with the GPS and together fit a cubic spline as the smoothed trajectory. Finally we register the fitted trajectory to the map. For videos taken from a driving car, the camera is mounted above the dashboard. For videos taken from a riding bike, the camera is mounted on the handlebar and thus is shaky and suffers more serious jittering.

For feature tracking, we detect 500-800 SURF [BTG08] features and track them through the frames. When current visible features are fewer than 50% we reacquire new features to track. In spacial pyramid mesh warping we use three scales of mesh: 16×16,

32×32, 64×64. We apply an adaptive regularization parameter similar to [LYTb] when estimating the warping mesh to adapt to various image contents. For each frame we try 10 values of the adaptive weight α from 0.5 to 5 and automatically choose a value that minimizes the total error. The system tends to choose a large α when the data term is weak, for example when there are large areas with insufficient features or when the camera suffers serious motion blurs. The system will choose a small α when there are reliable features distributed over the whole image.

We implement our method in Matlab R2014a and we run our method on an Intel i7 2.8GHZ Quad-Core machine with 16G RAM. For a video of 1920×1080 resolution, the warping estimation takes 150 milliseconds per frame, synthesizing the stabilized video takes 450 milliseconds per frame and the visual odometry takes about 350 milliseconds per frame. Spline fitting and map registering takes less than 30 milliseconds per frame.

4.3 Results

Below we show two results from videos taken from a riding bike and one result from a video taken from a driving car. For each video we show (a) tracking results with red dot for the tracked feature in previous frame, green circles for current position of those features and yellow lines connecting them; (b) original mesh grid of the previous frame; (c) mesh grid of the current frame warped to the previous frame; (d) Trajectory calculated from visual odometry aligned to geographic coordinates (in cyan), GPS position (in red) and the smoothed trajectory (in green); (e) The final registration result on the map. For more obvious visual effect, we show results calculated from two frames which differ by 0.5 second in time instead of using two consecutive frames.

Videos taken from a riding bike involves a lot of sudden camera rotation and translation which often causes the visual odometry to fail if we directly apply VO. After stabilizing the video, we get rid of the fast movement and obtain a reasonable VO trajectory.

The fast counterclockwise rotation of the camera can be seen from tracking result in

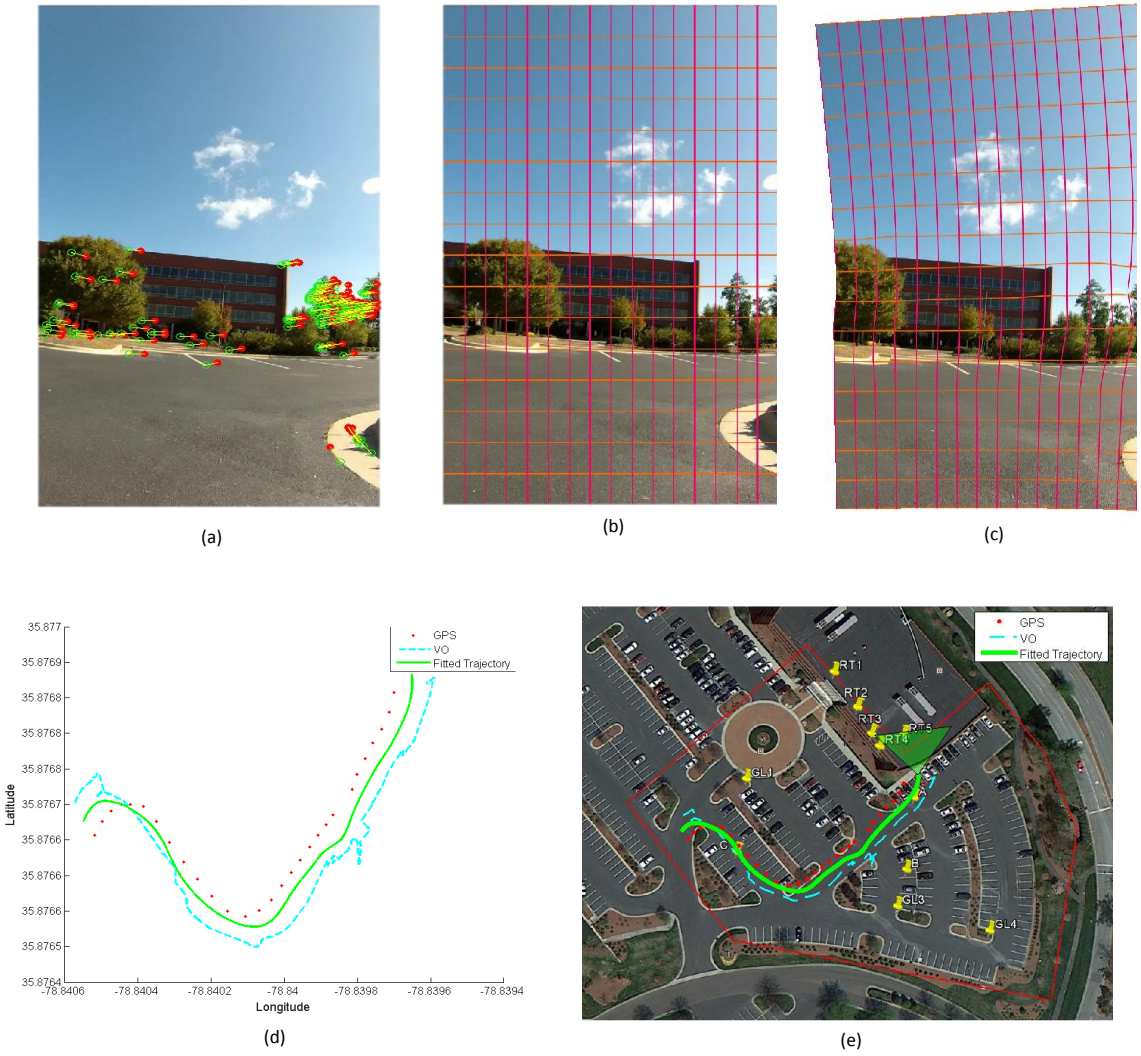

Figure 4.1: Intermediate and final results of a video taken from a bike.

Fig 4.2 (a) and the rotated mesh grid in Fig 4.1 (c). The visual odometry results are not so desirable as seen from Fig 4.2 (d) because the camera is facing the sun. But with the polynomial smoothing we get a reasonable good trajectory.

The camera mounted on a car suffers less shaking and jittering. As seen from Fig 4.3 (c), the deformation of the mesh is relatively small. However due to the error in GPS, the fitted camera trajectory is not very accurate. As in Fig 4.3 (e), the camera path goes across the trees by the road.

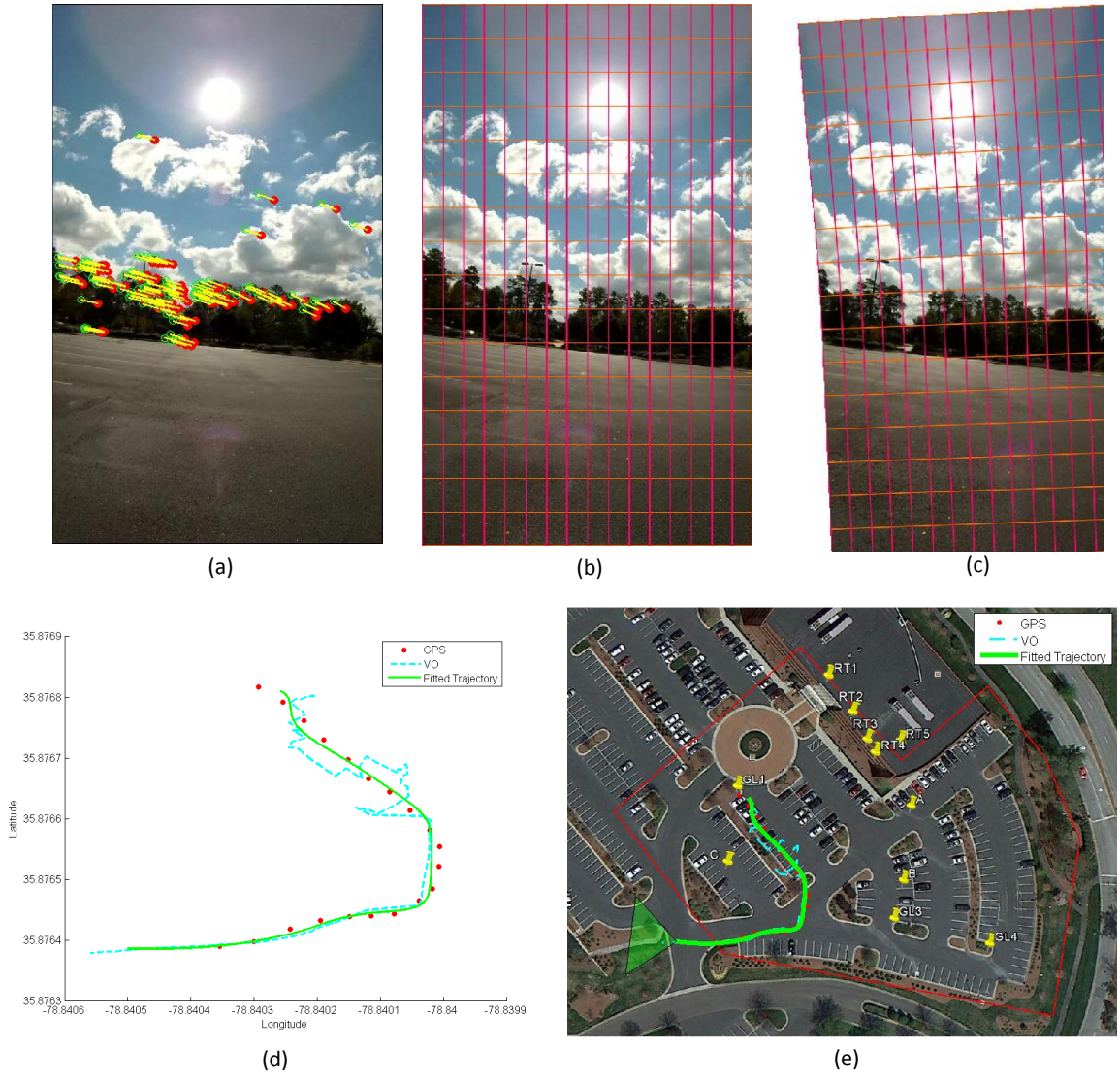

Figure 4.2: Intermediate and final results of a video taken from a bike.

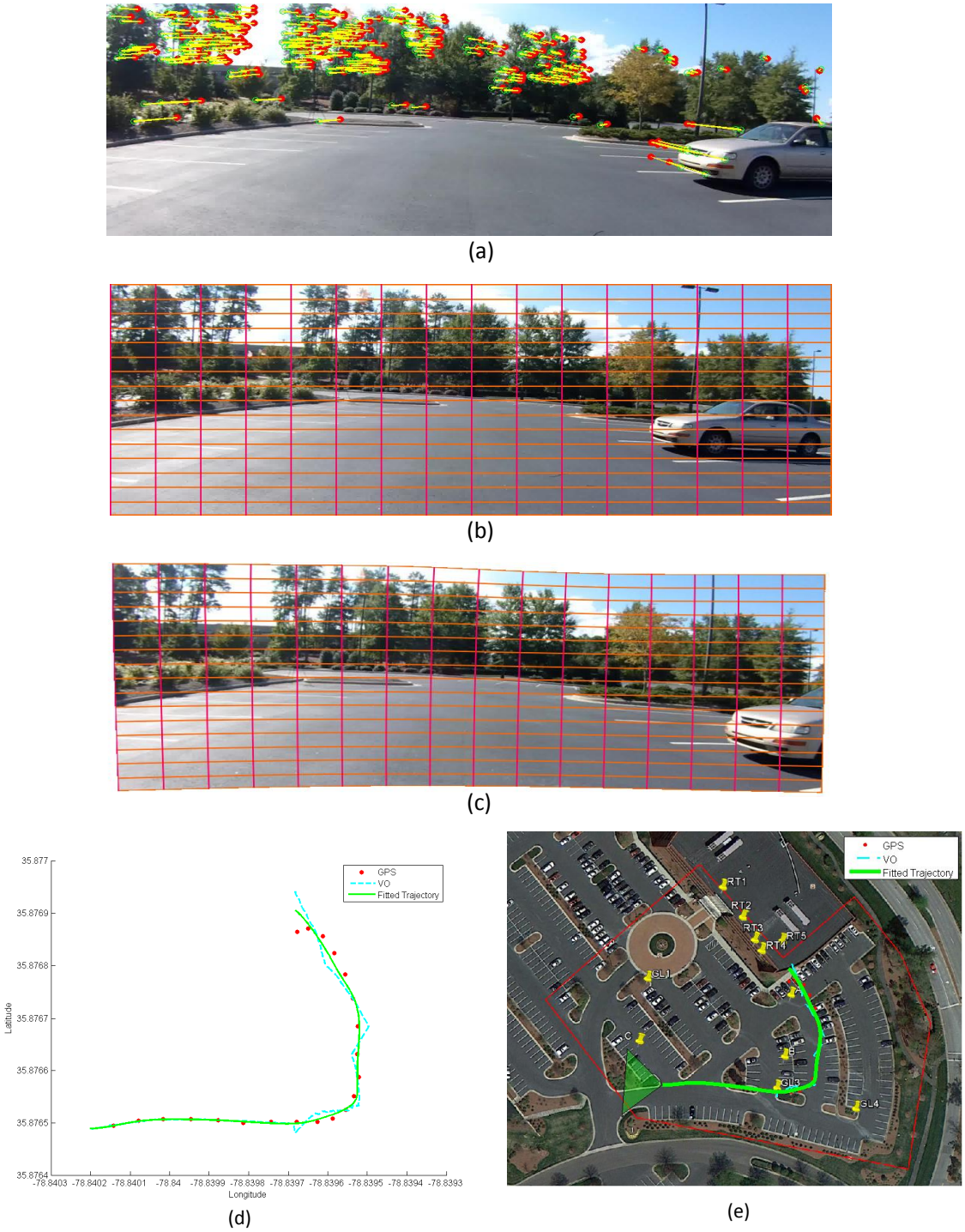

Figure 4.3: Intermediate and final results of a video taken from a car.

CHAPTER 5

Conclusion and Discussion

Moving camera registration can help providing better scene context for video understanding tasks including action recognition and behavior analysis. Geo-registration often have the issues of inaccurate correspondence and accumulated drifting errors over frames. To address the motion blurs, low resolution and repetitive patterns that are often encountered in motion estimation process, this thesis proposes a video stabilization method based on spatial pyramid mesh warping to get a stabilized video. To solve the accumulated drifting issue, we introduce a method to exploit and fuse both GPS information and visual odometry results. While GPS data is sparse and often suffers fluctuation and noise, motion estimating based on visual observation are continuous but is difficult to recover the true scale. Exploiting both cues help us get a robust camera geo-registration.

For videos taken from shaky camera, our stabilization method successfully remove the unintentional high-frequency camera motion which almost always cause visual odometry to get bad results. Our spatial pyramid mesh warping utilizes both short-range and long-range information to preserve the visual content. This is effective when the image has large areas of insufficient interest point (sky and the ground) and against noise. Utilizing GPS positions helps to correctly scale the camera trajectory without reconstructing the 3D scene and thus makes our method less sensitive to scene noises.

However, there are still several issues that cause errors in the motion estimation. Visual odometry will generate bad results when the camera is facing the glaring sun and when there are rolling shutter effect (like when facing a building) which our model does not address. In the stabilization step, our method relies only on 2D information to remove unintended camera motion which requires correctly estimating the image background

motion. Since we use RANSAC to pick the inlier interest points which are believed to be in the background, the estimation will fail when there are many foreground objects moving. Future work could be done to use more sophisticated approaches to robustly estimate the background motion, or use 3D stabilization methods for more complex scenes.

References

[BBM01] C. Buehler, M. Bosse, and L. Mcmilian. "Nonmetric image-based rendering for video stabilization." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[BTG08] H. Bay, A. Ess T. Tuytelaars, and L. Van Gool. "SURF:Speeded Up Robust Features." *Computer Vision and Image Understanding*, **110**(3):346–359, 2008.

[BYK13] H. Badino, A. Yamamoto, and T. Kanade. "Visual odometry by multiple-frame feature integration." In *International workshop on Computer Vision for Autonomous Driving at ICCV*, 2013.

[CAA10] R. Carroll, A. Agarwala, and M. Agrawala. "Image warps for artistic perspective manipulation." In *SIGGRAPH*, 2010.

[CC12] C.-H. Chang and Y.-Y. Chuang. "A line-structure-preserving approach to image resizing." In *International Conference on Computer Vision and Pattern Recognition*, 2012.

[CLH08] B. Chen, K. Lee, W. Huang, and J. Lin. "Capturing intention-based full-frame video stabilization." In *Computer Graphics Forum*, 2008.

[Dav] A. J. Davison. "Real-time Simultaneous Localisation and Mapping with a Single Camera." In *IEEE Conference on Computer Vision*.

[FB81] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Commun. ACM*, **24**(6):381–395, June 1981.

[FWZ05] A. Fitzgibbon, Y. Wexler, and A. Zisserman. "Image-based rendering using image-based priors." *International Journal of Computer Vision*, **63**(2):141–151, 2005.

[GKC12] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. "Calibration-free rolling shutter removal." In *ICCP*, 2012.

[GLU12] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[GZS11] A. Geiger, J. Ziegler, and C. Stiller. "StereoScan: dense 3D reconstruction in real-time." In *Intelligent Vehicles Symposium*, 2011.

[HCS12] K. He, H. Chang, and J. Sun. "Content-aware rotation." In *International Conference on Computer Vision*, 2012.

[IMH05] T. Igarashi, T. Moscovich, and J. Hughes. "As rigid-as-possible shape manipulation." *ACM Trans. Graph*, **24**(3):1134–1141, 2005.

[LGJ] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. "Content-Preserving Warps for 3D Video Stabilization." *ACM Trans. Graph*, **28**(3).

[LSC02] H. Lim, S. Sinha, M. Cohen, and M. Uyttendaele. "Real-Time Image- Based 6-DOF Localization in Large-scale Environments." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2002.

[LYTa] C. Liu, J. Yuen, and A. Torralba. "SIFT Flow: dense correspondence across scenes and its applications." *IEEE Transaction on Pattern Recognition and Machine Intelligence*.

[LYTb] S. Liu, L. Yuan, P. Tan, and J. Sun. "Bundled Camera Path for Video Stabilization." *ACM Transactions on Graphics*.

[MOG06] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Shum. "Full-frame video stabilization with motion inpainting." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **28**(7):1150–1164, 2006.

[Nis03] D. Nister. "An efficient solution to the five-point relative pose problem." In *International Conference on Computer Vision and Pattern Recognition*, 2003.

[SBS07] G. Schindler, M. Brown, and R. Szeliski. "City-Scale Location Recognition." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[SC14] S. Song and M. Chandraker. "Robust Scale Estimation in Real-Time Monocular SFM for Autonomous Driving." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[SMW] S. Schaefer, T. McPhail, and J. Warren. "Image deformation using moving least squares." *ACM Trans. Graph.*

[SSS] N. Snavely, S. Seitz, and R. Szeliski. "Photo tourism: exploring photo collections in 3D." *ACM Transactions on Graphics*.

[SZS08] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. "A comparative study of energy minimization methods for markov random fields with smoothness-based priors." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(6):1068–1080, 2008.

[VZS12] G. Vaca-Castano, A. Zamir, and M. Shah. "City scale geo-spatial trajectory estimation of a moving camera." In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[WKR] B. Williams, G. Klein, and I. Reid. "Automatic Relocalization and Loop Closing for Real-Time Monocular SLAM." *IEEE Transactions on Pattern Recognition and machine ingelligence*.

[ZCH09] G. Zhang, M. Cheng, S. Hu, and R. Martin. "A shape preserving approach to image resizing." In *Computer Graphics Forum*, 2009.

[ZS10] A. Zamir and M. Shah. "Accurate Image Localization Based on Google Maps Street View." In *European Conference on Computer Vision*, 2010.

[ZS14] J Zhang and S. Singh. "LOAM: Lidar Odometry and mapping in real-time." In *Robotics: science and Systems conference*, 2014.