國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia

College of Electrical Engineering ＆ Computer Science

National Taiwan University

Master Thesis

縮時環場視訊的穩定

Time-lapse Panoramic Video Stabilization

沈超

Chao Shen

指導教授：洪一平博士

Advisor: Yi-Ping Hung, Ph.D.

中華民國 104 年 7 月

July, 2015

國立臺灣大學碩士學位論文
口試委員會審定書

縮時環場視訊的穩定

Time-lapse Panoramic Video Stabilization

本論文係沈超君（學號 R02944045）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國一百零四年七月十三日承下列考試委員審查通過及口試及格，特此證明

口試委員：

洪一平（簽名）

（指導教授）

歐陽明　　邱志義

莊永裕

所　長：逄愛君

致謝

能夠完成這篇論文，首先要感謝的就是指導教授洪一平。洪老師總是能在最短的時間內找出我研究的不足，以及指點思考方向，讓我可以很快的找到問題和解法。從中，老師教會了我如何找尋研究方向，思考和解決問題。還要特別感謝口試委員歐陽明教授，莊永裕教授，邱志義教授給與的寶貴意見，使得我最後的研究和論文更加完整。在研究過程中，還要感謝實驗室的同學，學長姐，學弟妹們的陪伴和支持，感謝大源，秉軒在找題目時候的幫助和支持，感謝彥齊，冠翔，歆如，冠捷等在研究過程中的陪伴和鼓勵。最後要非常感謝我的父母，感謝你們的精神和物質上的支持和鼓勵。謝謝。

中文摘要

在虛擬實境用戶體驗中，使用基於圖片渲染的環場視訊越來越普遍，通過縮時來觀看已經錄好的冗長的視訊则是用戶的普遍選擇。然而，縮時會放大視訊原本就存在的抖動，沉浸式觀看抖動的視訊更會使得用戶頭暈和不舒服。

我們的目標是產生穩定的縮時環場視訊。我們第一個提出了一種可以獲得穩定的縮時環場視訊的方法。我們首先闡述一個基本的視訊穩定的方法框架以及如何改進使得可以減輕視差帶來的問題。然後我們提出適用於環場視訊穩定的方法，適用於縮時視訊穩定的方法以及縮時環場視訊穩定的方法。最後我們回顧最終的完整的處理方法以及驗證效果。虛擬實境用戶體驗中使用穩定的縮時環場視訊可以得到很好的效果。

關鍵字：縮時視訊、環場視訊、視訊穩定、影像變形、相機路徑、虛擬實境

Abstract

Image-based rendered panoramic videos are used in virtual reality user experiences commonly. Long videos recorded in situations and speed up watching by a created time-lapse video is a natural choice for users. Videos will be further shaking because of the time warping. Immersion viewing shaking videos are bring dizzy and uncomfortable for users.

Our goal is the time-lapse panoramic videos stabilization. Our paper propose the first method which can perform time-lapse panoramic videos stabilization well. We propose a traditional video stabilization framework and how to use Laplacian mesh warping method to correction the parallax error of the frames first. Then we optimize the proposed framework to fit panoramic videos well. After that, we extend the basic framework by an optimized frames selection method to generating time-lapse videos. Finally, we go through our approaches and evaluation by using several input videos. Using stabilized panoramic videos in virtual reality user experiences provide good results.

Key words: Time-lapse video, Panoramic video, Video stabilization, Image warping, Camera path, Virtual reality

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Virtual reality devices developing improved much today. It is useful for immersive experience for users. It allow the user visit distant or hardly reachable places such as sea floor, sky, and even outer space. It also allow the user feel difficult behaviors experience such as skiing, climbing and even flying. Virtual reality devices become more and more popular recent years and have a bright future.

Panoramic video based virtual reality dynamic scene is created easily. It perform a realistic feeling for users than 3D modelling based scene. The first step for image-based rendering is generating panoramic videos to use. Fortunately, digital cameras become more and more cheap, small and easy to use today. We also are convenient to combine multiple cameras as a system to get the panoramic video. It can be attached to out-vehicles, computers, phones, and wearable cameras. 360Heros[1] is a good example to hand-free and always-on to allow the users record videos in many special situations such as skiing, climbing and even sky driving and create panoramic videos by video stitching.

While more and more panoramic videos are recorded, watching through this video is a challenge for users. The first reason is the video recorded is too long to boring. Second, video shaking with natural motion disturbing caused the watching uncomfortable. Speed up is a natural choice for users to fast browsing the videos. A naive fast forward video

[1] Capture life in 360 video, http://www.360heros.com/

method is skip fixed frames to show as a time-lapse video. But the time-lapse videos will be further disturbing viewing for users because of its speed up. Traditional video stabilization can not perform a result well for them. In addition, immersion viewing shaking videos is rejected for users because it brings intensive dizzy and uncomfortable.

Traditional video stabilization methods [1–6] can be divided three steps: motion estimation, motion compensation, and image composition. They can perform a stabilized result well for general non distortion egocentric videos. But they are not suitable for generated time-lapse videos stabilization. Existed time-lapse video stabilization approaches [7, 8] add a pre-step before traditional video stabilization framework's three steps to optimize frames selection to generate time-lapse videos first. But they only considered the egocentric videos time warping. Existed panoramic video stabilization method [9] follow traditional video stabilization steps and use different motion estimation methods to adapt panoramic videos stabilization. It can not perform a good result for time-lapse videos. There is no existed approach framework to handle the time-lapse panoramic video stabilization problems.

Our goal is creating and stabilization the time-lapse panoramic videos. This paper propose the first method which can perform time-lapse panoramic video stabilization well. We propose an optimize frame selection method by consider camera motion averaging, video times scaling uniform and subdivision camera paths stabilization for panoramic videos. This approach measure the influence of the panoramic local distortion for stabilization. In addition, It guarantee the global compression ratio of the time and maintain the total camera motion for the final time-lapse video. We propose an optimize single camera path method for time-lapse video by adaptive Gaussian smooth considered the time-lapse video and corresponding original video' s velocity and acceleration characteristics changing. We consider to use 2D Laplacian mesh warping to generate subdivision camera paths for panoramic video stabilization. We summarize our approaches frameworks to handling the time-lapse, panoramic and time-lapse panoramic videos stabilization and evaluate the effectiveness of our approaches using serval input videos.

The rest of paper is organized as follows. We survey related works in chapte 2. We propose a basic video stabilization framework and introduce how to use Laplacian mesh warping method to correction the parallax of the frames in chapte 3. Then we perform an optimize method for panoramic video stabilization in chapte 4. In chapte 5, we propose a new time-lapse video stabilization method suitable for egocentric videos and improve it to fit panoramic videos. We summarize our approaches frameworks and evaluate in each chapter.

Chapter 2

Related Work

Hardware-based or software-based approaches are two main kinds of approaches for traditional videos stabilization.

Hardware based approaches used the sensors to collect camera motion informations. Multiple works [10, 11] has showed how to use sensors to directly measure the camera motion during capturing and how to use the camera motions to video stabilization and rolling shutter correction. Hardware based approaches utilizing onboard gyros can be quite successful with specialized hardware at capture time. However, heavy specialized hardware is not convenient to take in many situations because it occupy too much loading for quadcopter and is a burden for people in sports. In addition, it can not be apply to existing videos.

Most software based video stabilization methods [1–3, 5, 6, 12, 13] follow a framework include three steps: motion estimation, motion compensation, image composition.

- Motion estimation get the previous frame to current frame transformation and accumulate the transformations to get the image motion trajectory. Software video stabilization techniques can be roughly divided as 2D and 3D methods through this steps using different kinds of approaches. 3D methods use structure from motion (SfM) method to estimate camera paths in 3D apace for stabilization [12]. 3D methods reproduce the real 3D camera paths. However, the motion model estimation is less robust in various degeneration such as feature tracking failure, motion blur, camera zooming. 2D methods common to use a 2d transform matrix to represent the camera motion [5]. The best advantage of 2D methods is robust and fast because of only calculation a linear transformation in processing. [6] considered remove the foreground influence in transforms calculation by an classification iteration of optical flow.

- Motion compensation smooth the trajectory generated above and use the smooth trajectory to generate a new set of previous to current transformations. Many methods have been proposed for the goal of this step: remove high-frequency jitters from the estimated camera motion. For example, Litvin [14] construct a physics-based state-space model of these interframe motion parameters and use recursive Kalman filtering to perform stabilized camera position estimation. Grundmann [13] present a novel algorithm for automatically applying constrainable, L1-optimal camera paths to generate stabilized videos by removing undesired motions.

- In image composition, most approaches use each original frame to generate a stabilized frame by applying a new transformation in each original frame. It is too weak to handle the parallax problems because a full-frame warp can not model the scene without depth information. Recent stabilized methods let a frame warping as a Laplacian mesh distorting to localize features influence for the final results. Liu [12] is the first technique that can perform 3D video stabilization for dynamic scenes by weighted Laplacian mesh distorting to preserve contents. Liu [5] present bundles camera paths model which maintains multiple, spatially-variant camera paths. This kind of methods can ease the parallax problems for the result.

Kamali [9] propose the first and only omnidirectional video stabilization method. The main contribution of this paper is propose a structure from motion method to create the camera path of omnidirectional video. Then, they use 3D spherical Laplacian Triangle mesh distorting by local features to stabilization. While this image stitching based structure from motion to generate a single camera path too coarse to represent local shaking.

This paper aims the goal of robust high-quality result for panoramic videos. We propose a subdivision camera paths motion model like Liu [5] to panoramic video stabilization. Each different area in panoramic frame have different level of deform and we generate a camera path to fit the local area.

Time-lapse video stabilization techniques can also be decided in 2D and 3D methods. Kopf [15] first to consider this problem and use a sophisticated 3D method to solve it. They used structure-from-motion method to create the 3D scene and obtain the camera path first. Then, they propose a new camera path smooth method to generate new smooth camera path. Finally, they propose an algorithm to choose multi original frames to merge a new frame to generate new smooth time-lapse video. This approach works well when there is sufficient camera motion and parallax in a scene, but has difficulty where the camera motion is small or purely rotational, as the depth triangulation in the structure from motion step is not well constrained. Structure from motion also have difficulties to get results when the scene is dynamic. This approach also has very high computational cost that over one minutes pre frame.

Latest works consider 2D methods to time-lapse video stabilization because of low cost, fast and robust. They add a frame selection step before traditional video stabilization methods three steps to create a lime-lapse video first. Joshi [8] present an optimize algorithm in frame selection step to create time-lapse videos real time. Pole [7] also propose a new optimize frame selection algorithm in frame selection step before traditional stabilization framework. Both of them propose a method to optimize frames selection to creating time-lapse video and choosing an existed method to stabilize created time-lapse video. But, both of above methods to frames selection can not strict guarantee the compression ratio of time and space for the final time-lapse video. In addition, Their methods are not suitable for the panoramic videos time warping and stabilization.

This paper propose an optimize frame selection method to generate time-lapse videos and propose a camera paths optimization method in video stabilization steps. We can perform videos with dynamic scenes well. We considered camera motion averaging and video times scaling uniform in time-lapse processing. We guarantee the global compression ratio of the time and maintain the total camera motion for the final time-lapse video.

Chapter 3

Basic Video Stabilization Framework

In this section, we introduce the basic video stabilization framework for dynamic scenes first. Then, we propose how to use Laplacian mesh warping method to correction the parallax of the frames.

3.1 Dynamic Scenes Handling

In Motion Estimation, We get previous to current frame transformation by Calculation optical flow of sparse features setted by the iterative Lucas-Kanade method with pyramids. This result matrix H only represent 2D rigid transform, no scaling, no sharing, to fast and robust enough.

$$H = \begin{pmatrix} R_{0,0} & R_{0,1} & T_0 \\ R_{1,0} & R_{1,1} & T_1 \end{pmatrix} \tag{3.1}$$

We define d_x represent the motion of image contents in x axis on the image plane, the same as d_y represent the motion of image contents in y axis and d_a represent the angle of the image contents rotation on the image plane. We obtain d_x, d_y and d_a values from 2D homography matrix H.

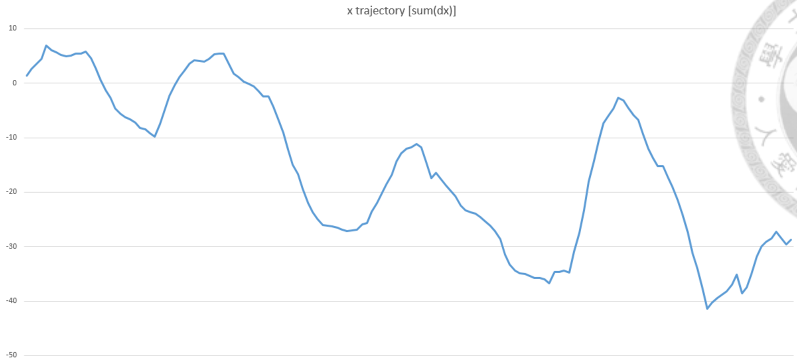


Figure 3.1: An example of the original trajectory of image contents generated by original frame transforms

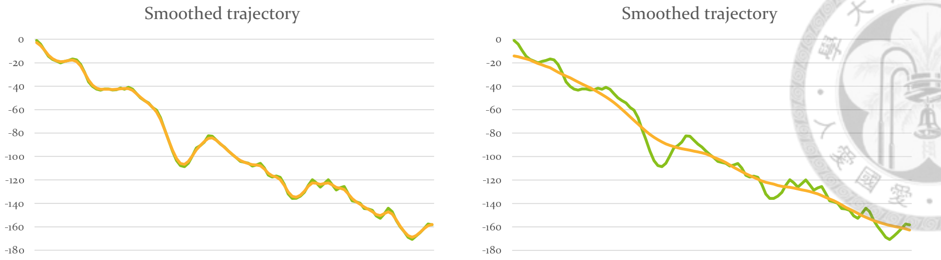
$$d_x = T_0 \tag{3.2}$$

$$d_y = T_1 \tag{3.3}$$

$$d_a = \arctan \frac{R_{00}}{R_{11}} \tag{3.4}$$

We will explain the image contents motion smooth in x,y axis and then discuss the image rotation smooth in the end in this section. Figure **??** show the d_x transformation from previews to current frame sequence as an example. The next step use the accumulated d_x,d_y transformations to calculate a related image contents motion trajectory in x and y axis. We use $(0, 0)$ as the start image contents motion position and apply the transformations one by one to obtain the trajectory in x,y axis and the intermediate position of this trajectory. The final trajectory we use is shift the trajectory to let the intermediate position in $(0, 0)$. Figure **??** show the calculated x original trajectory from the original transformation showing in Figure **??**.

The trajectory is a rather abstract quantity that it not represent the real camera motion directly. It is a 2D rigid transforms in a plane can not afford the representation of the camera motion. We not calculated the 3D camera trajectory directly because the structure from motion approaches are too slow and much restriction to suitable multiple situation robust. We are thinking about the opposite direction, the image contents motion camera taken also represent the camera motion, we use image contents 2D motion and rotation in



(a) $w = 60, \sigma = 1$ (b) $w = 60, \sigma = 5$

Figure 3.2: Smoothed trajectories by different parameters

image plane to mapping the camera motion in 3D space. The trajectory represent here all means the 2D image contents motions and rotations.

In motion compensation, our goal is smooth the trajectory. Recent methods [9, 12] consider Gaussian filter to smooth the trajectory. we also use Gaussian smooth method in this basic video stabilization framework. We should pay attention to the Gaussian kernel size and sigma value. Large kernel size and sigma make the video more smooth but take much more black areas to the result videos. In addition, large kernel size and sigma can not maintain original image motion user active operations. It’ s difficult for users to setup this parameters. Figure 3.2 show the smooth trajectory of x axis with large and small kernel and sigma.

The original trajectory and the corresponding smooth trajectory are used to generate the new set of previews to current transformations in x,y axis which is the final we use to transform the videos frames to smooth. We use $d_{smooth-x}$ and $d_{smooth-y}$ to represent the smoothed image contents motion in x,y axis. They are obtained by the correspondent d_x,d_y added the difference between the original trajectory and the smooth trajectory.

$$d_{smooth-x} = d_x + (Traj_{smooth-x} - Traj_x) \tag{3.5}$$

$$d_{smooth-y} = d_y + (Traj_{smooth-y} - Traj_y) \tag{3.6}$$

The smooth homography matrix used to image warping can be created by the smoothed image contents motion transformation $d_{smooth-x}$,$d_{smooth-y}$ in x,y axis and image rotation $d_{smooth-a}$. Apply the list of new transformations to the original video frames can get the stabilized videos.

$$H_{smooth} = \begin{pmatrix} \cos d_{smooth-a} & -\sin d_{smooth-a} & d_{smooth-x} \\ \sin d_{smooth-a} & \cos d_{smooth-a} & d_{smooth-y} \end{pmatrix} \tag{3.7}$$

We handle the image contents rotation d_a as same as the image contents motion in x,y axis. While we observe a phenomenon that the image contents rotation changed small in most video sequence. But the changing of image contents rotation influence users feeling of video shaking much. So, using large Gaussian kernel and sigma, even using a L1 line fit or a fixed rotation value can perform a good result in video stabilization.

Following all above steps can perform egocentric video stabilization well. But this basic framework can not work with panoramic videos or time-lapse videos. Panoramic video frames should not have a global image contents motion and rotation because it represent the omnidirectional for the camera position. Each local area in panoramic video has its local contents motion and rotation trajectory. So, we can take the panoramic frame warping as a Laplacian mesh distorting in video stabilization. The next chapte 4 will explain a method to update this basic framework to panoramic video stabilization. Time-lapse videos amplify the shaking of frames because the contents motion and rotation in videos speed up. We update this basic video stabilization framework detail in chapte 5 to perform time-lapse videos well.

3.2 Parallax Handling

We first represent the warping-based subdivision camera paths generation methods framework for parallax handling following [5, 12] first. Then, we introduce the detail of Laplacian mesh warping approach.

Parallax phenomenon in video stabilization lead the features motion difference for each frame. This influence the Homography matrix calculation accuracy and stabilization results. Camera subdivision can perform each sub camera only consider the features motion in camera surrounded area to local homography calculation accuracy. Using local accurate smoothed frame transformation can reduce the parallax problems influence of the video stabilization method result. Partial video frame stabilization and single camera path smooth follow the basic video stabilization framework detailed above. The key improve the framework here is how to combine the multiple camera paths smoothly by Laplacian warping detailed the following steps.

The first step is to create a subdivision camera paths mesh. We split a frame as a $n»n$ mesh. The size of the mesh is same as the video frame size. Each vertex of the mesh represent a local camera. The expected situation is getting the local camera path in each mesh vertex by local features of the part of the video directly. But in fact the obvious features distribution of any frame is not uniformed. The camera paths generated directly in feature rich region of frames is used to generate feature poor region of frames camera paths by Laplacian mesh warping.

After subdivision camera paths mesh generated, each region in the single input frame is warped to generate a single output frame. For each region of the frame, it is a rectangle region constraint by four camera position. We already know the four camera position before and after warping. So, find the homography of each single region of the frame and combine all the region of each frame to generate the final warped frame is the way we chose in image composition. The next formula show the homography H calculated by four boundary points. For the region I, (x_i, y_i), (x_{i+1}, y_{i+1}), $(x_{i+mcols}, y_{i+mcols})$, $(x_{i+mcols+1}, y_{i+mcols+1})$ are the four boundary points of the frame region. $mcols$ represent the number of mesh columns.

$$\begin{pmatrix} x_i^{'} & x_{i+1}^{'} & x_{i+mcols}^{'} & x_{i+mcols+1}^{'} \\ y_i^{'} & y_{i+1}^{'} & y_{i+mcols}^{'} & y_{i+mcols+1}^{'} \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} x_i & x_{i+1} & x_{i+mcols} & x_{i+mcols+1} \\ y_i & y_{i+1} & y_{i+mcols} & y_{i+mcols+1} \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{3.8}$$

3.2.1 Laplacian Mesh Warping

The subdivision camera paths can represent as a 2D Laplacian mesh. Let $M = (V, E)$ be a triangles mesh where V are the set of vertices of the mesh and E are the set of edges of the mesh. Each vertex $V_i \in M$ is represented by absolute cartesian coordinates, denoted by $V_i = (x_i, y_i, z_i)$ and $Z_i = 0$ in 2D Laplacian mesh here.

Considering the Laplacian mesh deforming can be in different views such as vertices, edges or faces. We adopt the mesh warping approach can be as multiple steps optimized problems following the [5, 12, 16] which considered the triangles deforming. It Generate an intermediate result by minimize the error metric of mesh warping prevents shearing and non-uniform stretching but permits rotation and uniform scaling first. After that, It takes this result and adjusts the scale of each triangle.

- **Scale Free Construction**:

  For a corresponding triangle V_1, V_2, V_3, we calculated the relative coordinates x_{01}, y_{01} in the related coordinate system create by vector V_0,V_1 to represent V_2 in this coordinates system.

$$V_2 = V_0 + x_{01}\overrightarrow{V_0V_1} + y_{01}R_{90}\overrightarrow{V_0V_1}, \quad R_{90} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{3.9}$$

  We defined the deformed triangle as $V_1^{'}, V_2^{'}, V_2^{'}$. The $V_2^{desired}$ also can be defined in the relative coordinates system created by $V_0^{'}, V_1^{'}$.

$$V_2^{desired} = V_0^{'} + x_{01}\overrightarrow{V_0^{'}V_1^{'}} + y_{01}R_{90}\overrightarrow{V_0^{'}V_1^{'}}, \quad R_{90} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{3.10}$$

  The error associated about this triangle vertex V_2 is means the difference between the desired vertex position by the relative local triangle warping and actual vertex position by the global optimized mesh deforming results.

$$E_{V_2} = \|V_2^{desired} - V_2^{'}\|^2 \tag{3.11}$$

  This results is suitable for represent all the vertices in this mesh represented by each triangle related and we can express this in matrix form.

$$E_{1(V')} = V^{'T} M V', \quad V^{'} = (V^{'}_{0x}, V^{'}_{0y}, ..., V^{'}_{nx}, V^{'}_{nx})^T \tag{3.12}$$

According to the theory of Laplacian mesh deforming, the matrix M has multiple solution because the rank of M is not enough to consider to solve this equation. The camera paths generated directly in feature rich region of frames as the constraint points of the Laplacian mesh which can fix the relation rows and cols in matrix M. The feature poor region of frames camera paths here is also means the free vertices of the mesh. So, the minimize problem can solved by the partial derivatives of the function $E_1(V^{'})$ because constraint vertices existence.

This step not consider the scale changing of the mesh to fast calculation enough. We need to adjust the scale of the mesh for the final results.

- **Scale Adjustment**:

  We split the position handling and rotation handling processing in this step. A minimize methods used to optimize the triangle position fitting. After that, an edge based minimize error methods used to optimize the triangle rotation and get the final results.

  The intermediate triangles mesh only use to refer in this step because of its large unwanted scaling. We fit the original triangles to the intermediate triangles by a minimize error measure approach. We defined the fitted triangle $(V_1^{fit}, V_2^{fit}, V_3^{fit})$ for the original triangle (V_1, V_2, V_3) and calculate its position by the following functional:

$$E_{2(V_{tri}^{fit})} = \sum_{i=1,2,3} \|V_i^{fit} - V_i^{'}\|^2 \tag{3.13}$$

  This can extend to the whole mesh. We can minimize E_2 by setting the partial derivatives of its free variables to zero. After that, the system will calculate the final vertices positions by given the constraint vertices and minimize the free vertices position between the resulted now and the fitted before. We defined the original triangle vertices (V_1, V_2, V_3) and the related fitted vertices $(V_1^{fit}, V_2^{fit}, V_3^{fit})$ calculated above steps. We define the minimize error equation:

$$E_{3(V_i'')} = \sum_{(i,j)=(0,1),(1,2),(2,0)} \|\overrightarrow{V_i''V_j''} - \overrightarrow{V_i^{fit}V_j^{fit}}\|^2 \tag{3.14}$$

We evaluate the error of each edge instead of the vertex because we consider the rotation of each triangle and ignore the position. We minimize E_3 by setting the partial derivatives of its free vertices to zero too.

Finally, summarize all the mesh vertices V are divided free vertices and constraint vertices. Free vertices means this camera position can not calculate the camera motion by features points directly and the constraint vertices can. Mesh deforming approach using the known vertices motion to calculate the unknown vertices motions.

This 2D Laplacian mesh deforming method perform a fast enough method to generate all subdivision cameras motion. We predefine the subdivision mesh size before running this algorithm. Then, we can find that all above parameter matrix used in minimize error equation can obtained because it is only decided the mesh defined. When the vertices position matrix updated by constraint vertices position changed, we can fast obtain the free vertices deforming results because of only need a simple matrix multiply.

Chapter 4

Panoramic Processing

This section propose a panoramic video stabilization method. It is based on Laplacian mesh warping. We introduce the panoramic video characters, the difficulties of panoramic video stabilization and propose the detail of our panoramic video stabilization method first. Then, we summarize our approaches frameworks and evaluate it.

4.1 Panoramic Video Stabilization

Panoramic video stabilization intuitive method created the 3D camera path by structure from motion. Traditional structure from motion method has much restricts. Panoramic frames only can be as six traditional photos and are not benefit for this steps stable. Kamali [9] proposed a new structure from motion method used in frames stitching to create the 3D camera path to video stabilization. However, 3D camera path calculated by features of frames is not reflect the true camera motion because the parallax problem is much serious in panoramic videos camera path generating. Outdoor panoramic videos include the ground plane and the sky at the same time often. The difference between the distance from camera to the ground plane and the distance from camera to the sky is too large to fit a single camera path for panoramic frame.

Traditional 2D video stabilization methods is stable and fast in use. However it is not suitable for handling the panoramic frames sequence. The features motion in different part of the frame must be difference in panoramic frames contents relations in 3D space. [5,12]

propose a method to use Laplacian mesh warping to reduce the parallax effective of the video stabilization. It performed a good result under the promise of the features motion description accurately. It is easy in video frames are not distortion. While the panoramic frame is rectangle size and can considered as a texture for a sphere in space. The texture mapping lead to the panoramic frames vision contents distortion. The features motion in panoramic frames combine the reason of the shaking and mapping. In addition, the distortion of the panoramic also influence the feature extracting in this step.

Another method of panoramic video stabilization is each original video stabilized first. Then, using all the stabilized original video to generate the panoramic video. Each Synchronous frame in different videos need to calibration. We can not accept the flicker taken by the error of calibrations between frames for the panoramic video. That’ s why traditional video calibration to generate panoramic video method is choose one or multiple frames for each video synchronized to create a template. All frames of the videos use this template to calibration. It is fast and stable.

Our goal is stabilize the panoramic video robust. We propose a subdivision camera paths for panoramic video stabilization. Split the influence of the image distortion by texture mapping is the core idea of this processing.

While the panoramic frame features mapping is effected by the distortion. We can obtain the features motion by video shaking in original videos not distortion. So, to obtain the right motion of shaking, we obtain the features motion between two frames by optical flow calculation for each original video.

We need to know the mapping of each feature between the original videos and the panoramic video. Fortunately, we apply the traditional panoramic video calibration which use a fixed mapping template. This means the features we extracted in original frames can applied to panoramic frames easily. Note that this mapping succeed in surround the center of original video because of the panoramic frame deforming. A panoramic frame with right motion features generated now.

We separate the influence of panoramic deforming and obtain the right feature motion by original video shaking now. Subdivision camera paths is a good solution for panoramic video stabilization. We split the whole panoramic frames to multiple areas and generate smooth camera path for each area to stabilization. It fit the observation that different part of panoramic frame must have different pixel motions. For each subdivision camera path, we follow the basic framework to stabilization. We need to use the original features motion results to calculate the homography matrix. The 2D motion trajectory represented the subdivision camera motion generated by the sequence of transformations in feature rich area in panoramic frame. This result is not the final camera position pixel motion in panoramic video. The intensity of panoramic deforming in subdivision camera position effect the final motion represented.

The panoramic deformed frame mapping to a sphere called UV mapping. For any point $P(d_x, d_y, d_z)$ on the sphere , UV normalized coordinate in the range of $(0, 1)$ calculated as follows:

$$u = 0.5 + \frac{\arctan 2(d_z, d_x)}{2\pi}$$
$$v = 0.5 - \frac{\arcsin d_y}{\pi}$$

We find the intensity of pixels deforming in panoramic frames is same if they have same latitude. The intensity of pixels deforming in panoramic frames fit the sin function of longitude. The relationship of the intensity of each pixels latitude and longitude fit the above normalizations. Using this above relationship of U,V to adjust the calculated intensity of the camera motion perform the right camera points position adjustment in panoramic frames. Defined the camera position point P original motion by shaking motion vector is (x, y) and the adjust result is $(x^{'}, y^{'})$, the adjustment as follows:

$$x^{'} = x \cdot \alpha \cdot \sin\theta \tag{4.1}$$

$$y^{'} = y \cdot \beta \cdot \sin\theta \tag{4.2}$$

Note α, β is the ratio of the original frame size and the panoramic frame size in rows and columns directions. The $sin\theta$ function is fit the influence of latitude.

We obtain the right constraint points motion for Laplacian mesh. Then, we use Laplacian mesh warping approach generating all subdivision camera trajectory. The challenge is maintain the omnidirectional view in mesh warping. Parallax reducing laplacian mesh warping can not maintain the rectangle shape of the frame. It caused the field of view reduce. We can accept this FOV changing in traditional video stabilization. Panoramic video stabilization must maintain the panoramic view of frame at any time. We propose a subdivision camera paths generating approach maintain the panoramic views by Laplacian mesh warping.

We reform the subdivision camera paths mesh according to the panoramic frame characteristic. The left and right mesh edge points have corresponding relationship. We considered pair of points in left and right edge have same latitude as the same points and combine their neighbours influence in calculating. The top edge of the mesh points correspond to the same point in space sphere. We considered all the top edge mesh points as a same point and combine all their neighbours influence in mesh warping calculation. The vertices on bottom edge of mesh handled the same approach of the top edge vertices. Then, we obtain all the vertices new positions. In image compensation, we define the filled area is same as the original panoramic frame area. Tilling outside the area defined all can draw back to the filled area because the edge extend relations defined above. This guarantee using the original panoramic pixels filling the filled area is enough. We obtain a same size warped panoramic frame to make sure the panoramic views.

4.2 Panoramic Processing Evaluation

We propose general panoramic video stabilization approach in above section. Figure 4.1 show the whole processing of this panoramic video stabilization.

We use 360Heros to fix the six Gopro Hero 4 cameras. Fixed camera position in video catching is useful for video calibration. The frames from difference video frame sequences catching at the same time calibration can use the same calibration template because of the fixed camera positions. Once we obtain the videos, videos synchronization is very impor-

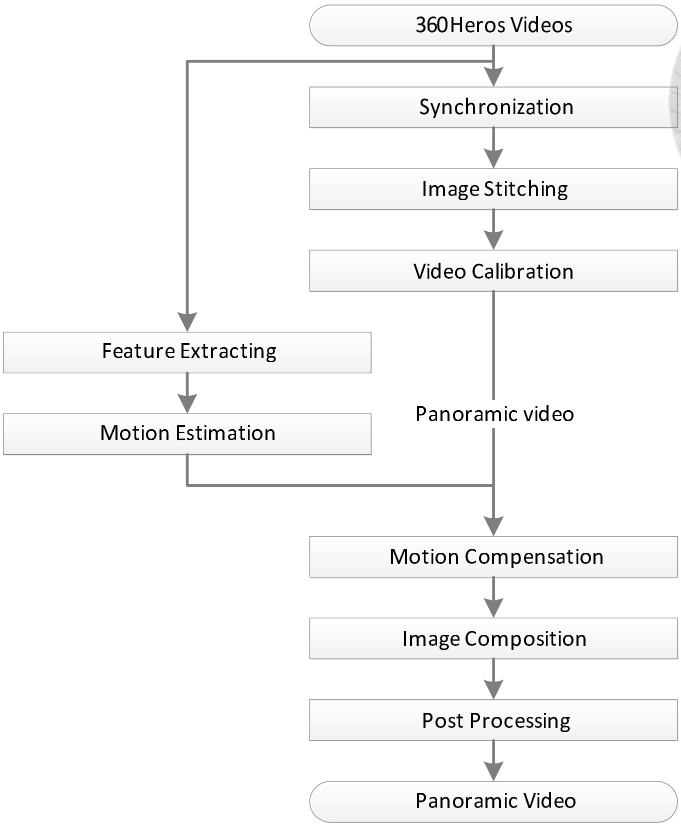


Figure 4.1: Panoramic Video Stabilization Framework

tant in this processing. Incorrect video synchronization influence all the frame sequence and caused the panoramic have the local motion. We are sensitive to the junctions between original videos in watching panoramic videos because incorrect connection parts of the panoramic frame take the intensive illusion. We can not tolerate the Synchronization error even only one frame. Synchronization by audio can not suitable for our application. We use motion estimation functions in VideoStitch to obtain the high accuracy video Synchronization. VideoStitch is a good tool that can generate a panoramic video fast and robust. We use this software generating a panoramic video without brightness adjustment and stabilization.

Panoramic video generation is the first step in this framework. We apply our panoramic video stabilization method to the generated panoramic video. Our panoramic video stabilization method also follow the three steps in basic video stabilization framework. But, We should extract the sparse features and calculated the features motion by optical flow for each original video before basic three steps. In addition, we also need to find the features mapping between the each original video and the panoramic video. Then, we accept the idea of Liu [5]. Single camera motion can not fit the contents motion in panoramic video. camera subdivision is particular suitable for panoramic video. Contents motion in panoramic video is partial similarity. Each sub camera motion can be represented the partial contents motion. We create the subdivision camera paths mesh and use the features motion calculated in original video to optimize all the subdivision camera paths. We are warping the panoramic frame by the adjusted key points represented the camera positions.

4.2.1 Panoramic Analysis

The subdivision camera paths mesh size is influence the result of video stabilization. The surround area are more large, the contents motion in this area are more difference. Sparse vertices distributed in this mesh may be too coarse to represent the motion of the surround area. Density vertices distributed in the mesh may be not perform a result well. The surround area may be too small to have enough feature points to calculate the camera path directly. Less constraint camera paths influence the others camera path calculation result. We defined a mesh suitable for panoramic frame by the size of $13 * 13$. It give a suitable size of each cell and a suitable sub mesh to map to the original video frames. Figure 4.2 show the mapping between the original frames and the panoramic frame. This correspondence is fixed if the template is chosen to panoramic video generation. Figure 4.2 also show the correspondence of the subdivision camera paths between the original frames and the panoramic frame.

4.2.2 Compare With VideoStitch

kamali [9] is the only paper propose a stabilization method suitable for panoramic video. But, they can not show their results in website. VideoStitch provide a panoramic video stabilization functions to use. We compared the results between our method and the VideoStitch software.

The VideoStitch can remove the contents high frequency shaking in panoramic video. Our method also can remove the contents high frequency shaking in panoramic video too. The difference between the method proposed here and the VideoStitch is showing in the

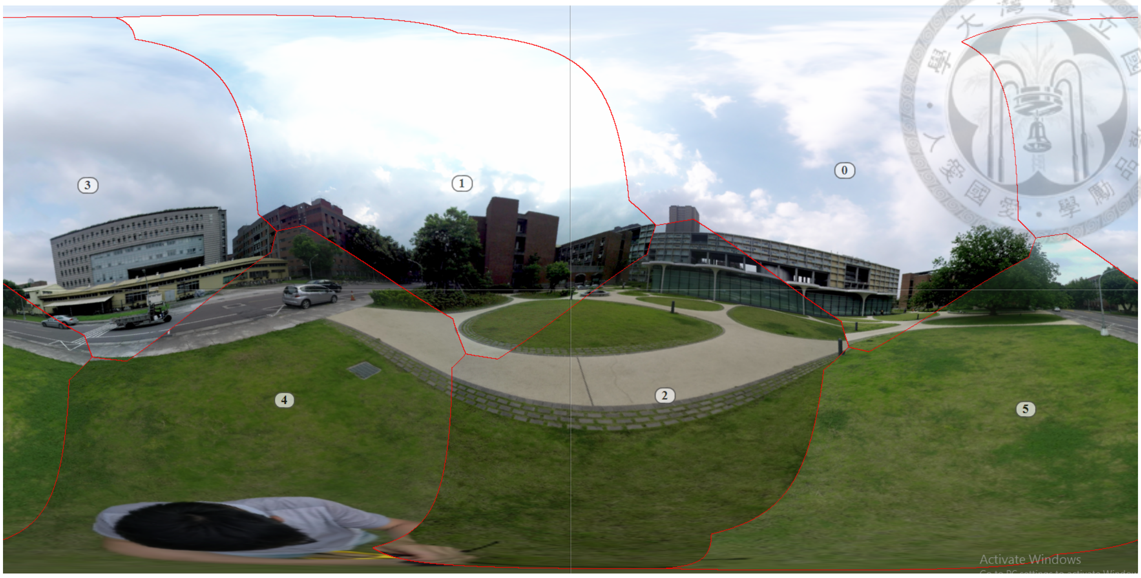


Figure 4.2: Mapping between Original Frames and Panoramic Frame

original frame connecting border in the panoramic video. VidioStitch can not explain the method they used to panoramic video stabilization. But, we guess they stabilize each original video or adjust the calibration template to panoramic video stabilization because we observed the stabilized panoramic video by VideoStitch have different degree of stability in different area mapping the different original frame. This cause the connection border shaking in the result.

Chapter 5

Time-lapse Processing

Our goal is to create time-lapse videos at any speed up ratio with no constraints on the scene content or camera motion. In addition, it should be suitable for multiple camera systems. We propose an optimize frame selection method by consider camera motion averaging and video times scaling uniform first. We guarantee the global compression ratio of the time and maintain the total camera motion for the final time-lapse video. Then, we propose an optimization of single camera path method for time-lapse videos by consider to the characteristic of velocity and acceleration in the corresponding original video. This time-lapse video stabilization are suitable for egocentric videos, We summaries our method framework and evaluate it. Finally, we improve the above time-lapse video stabilization approach we proposed to suitable multiple cameras system to generate panoramic videos.

5.1 Optimal Frame Selection

Given an input video represented as a frame sequence $F = 1, 2, , N$, we defined a time-lapse video F^c is a sub frame sequence which all frames selected from F we expected. The optimized image contents motion path means the chose frames contents motion between the frames.

We formulate the time-lapse frame selection optimization problems as a minimize path costs choose problem. We propose how to measure the costs first.

5.1.1 Shakiness Cost

An optimal frame-to-frame translation is where both frames aligned well and have significant overlap. The first criteria measure the path costs is total of neighbour frames transformation. Given two frames in the original video F_i, F_j. Define $T(i,j)$ is the homography matrix measure the transformations between F_i, F_j. We obtain the sparse feature points by optical flow method. We use the standard RANSAC method on the sparse feature points to calculate the T. The shakiness cost function as follows:

$$S_{i,j} = \begin{cases} C_o(i,j) & C_r(i,j) < \tau_c \\ \gamma & C_r(i,j) \geq \tau_c \end{cases} \tag{5.1}$$

Note that the C_r term is equivalent to the average of all measurement 2D features projection errors. We used this term to check the frame shaking motion too large or not. τ_c is the threshold of the average projection error. If the C_r term is too large than threshold to accept, we give a large γ cost means that we prefer not to choose this frame in path optimize unless no others can choose. Because large costs means frames are less overlap for the final time-lapse video. The large costs causes the optimization to avoid choose the frame to generate time-lapse video. If C_r term in acceptable region, we use the frame geometry center point projection error as the final cost for this term C_o which can represent this frame rotation and position. The C_o, C_r terms defined as follows:

$$C_r(i,j) = \frac{1}{n}\sum_{p=1}^{n} \|(x_p, y_p)_j^T - T(i,j)(x_p, y_p)_i^T\|_2 \tag{5.2}$$

$$C_o(i,j) = \|(x_0, y_0)^T - T(i,j)(x_0, y_0)^T\|_2 \tag{5.3}$$

5.1.2 Velocity Cost

The matching costs measure the time-lapse frames smooth. However, achieve the target speed-up ratio is more important for the time-lapse video. Recent methods considered the time-lapse video generating from the first frame in the original video and they considered this term by frame serial numbers velocity and acceleration from the first frame. But the first frame influence the optimize path choose in this term. In addition, their methods can not guarantee the speed-up caused frame decrease ratio of original frames is right. We consider the first frame of the time-lapse video can be one of the first $v/2$ original frame. The v defined the ratio of speed up times. If we fixed the first frame already, the probability of each frame in the original video chose to generate the time-lapse video by velocity and accelerate can costed. The perfect chose by this cost term is

$$F_i^c = F_0^c + kv$$

and the defined costs for this term is follows:

$$C_i = \begin{cases} \|i - (i_0 + kv)\|_2^2 & |i - (i_0 + kv)| < v \\ \tau_s & others \end{cases} \tag{5.4}$$

Note the i_0 is the first chose frame in original video. τ_s is the threshold to suggest the frame choose only consider the frame within v distance to the perfect choose. This term decide only depend on its position in original frame sequence.

5.1.3 Appearance Cost

We propose this term to measure the camera uniform motion in space. Record original videos may not contain a uniform moving speed. Slight record video camera motion influence the time-lapse video contents smooth showing. Global features flowing is a 2D representation of the camera motion. We consider uniform the 3D camera motion between two selected frames by uniform the 2D optical flow values between two selected frames. We defined the total values $G(x, y)$ by accumulate the feature motions calculated by optical flow for all chose frames.

$$G = \sum_{i=0}^{N-1} g_{i-1,i} \tag{5.5}$$

The cost of this term is represent as:

$$V_{i,j} = K_{flow} - G/N \tag{5.6}$$

We defined the K_{flow} represented the current frame motion. We prefer each frame motion is similar to the average frame motion of this choose frame. This term is only consider the chose frame sequence.

5.1.4 Optimal Frame Selection

We consider the path optimization using graphical model. Minimize functions of the form:

$$\hat{w}_{1...N} = argmin_{w_{1...N}} \left(\sum_{n=1}^{N} U_n(w_n) + \sum_{n=2}^{N} P_n(w_n, w_{n-1}) \right) \tag{5.7}$$

Set up the costs for traversing graph – each path from left to right is one possible configuration of world states. Unary term $U_n(w_n)$ costs only related by itself. We consider the appearance costs as the Unary term in this solution. Appearance costs only decided by its position in frame sequence. We can pre-known the costs if the first frame and speed up rate are defined. Then, we can create the graphical model by the Unary term defined. Each Unary state we want to choose means a frame in time-lapse video. The total of states is fixed already and this proof our time-lapse method make sure the speed up ratio. Each state have $2v$ units represent original frames. We need to choose a frame from sub frames sequence on each state. Unary term has its own costs only related itself frame. We consider the appearance costs as the Unary probability as follows

$$Pr(x_n|w_n) = C_i \tag{5.8}$$

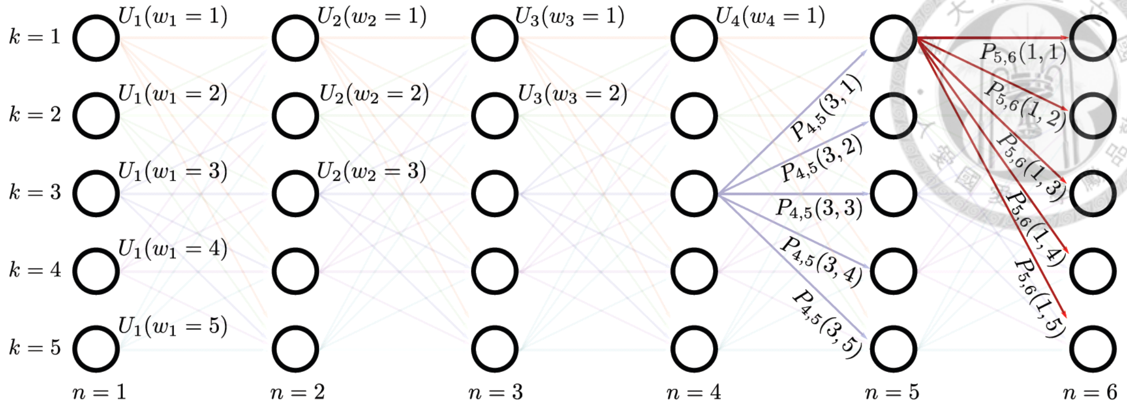


Figure 5.1: Used directed hidden Markov chain model

The velocity cost is only related its own position in original frame sequence and can split multiple section as the state of the graphical. This term represent the high costs this frame has, also represent the low probability this frame chose to generate time-lapse video.

The overall weight of the edge between nodes means frame i and frame j in different states w_n and w_{n-1} is given by:

$$Pr(w_n|w_{n-1}) = \alpha \cdot S_{i,j} + \beta \cdot V_{i,j} \tag{5.9}$$

Note the α, β are represented the importance of the related term costs in the overall weight of the edge. Nodes represented the sub frame sequence may be have the same frame in neighbour states in our graphical model. We add zero weights edges to avoid the repeat frames chose at the same time. We then use directed hidden Markov chain model to compute the shortest path represented have global largest probability to generate the time-lapse frames. Working through this graph model compute maximize probability to reach each node. Keep going until we reach the last column, find the maximum probability and trace back to see how we got there. This is the maximum probability time-lapse frame sequence we got. Figure 5.1[1] is a picture of this model.

[1] vision: models, learning and inference. copyright 2011 Simon J.D. Prince

5.2 Single Camera Path Optimization

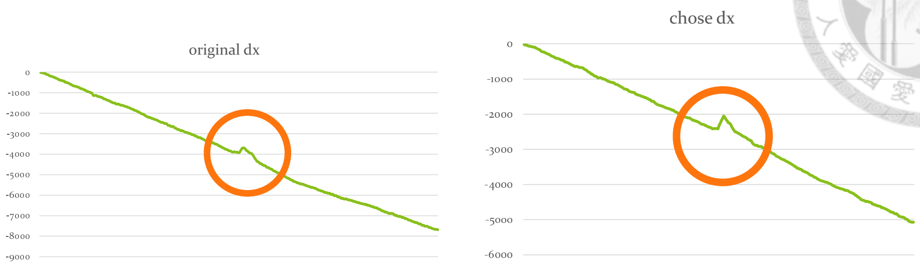
We propose a general algorithm to generate a time-lapse video stabilized relatively. We propose a complete time-lapse video stabilization method here. Time-lapse video amply the shaking effect of the original video. We explain how to model the enlarged shaking phenomenon first. Then, we propose a camera path optimization approach to stabilization.

First, we follow the theory of [13] to analysis the original video and its related time-lapse video. We analysis the camera paths to be composed of the following path segments. A constant path term represent a static camera motion. $D_p(t) = 0$ be the differential operator to smooth. A velocity term represented a camera panning or dolly shot. $D_p^2(t) = 0$ means this term to smooth. An acceleration term represent the ease in and out transition between static and panning cameras motion. $D_p^3(t) = 0$ to smooth it.

For a video frame sequence $I_1, I_2, I_3, , I_n$. Defined each pair frames (I_{t-1}, I_t) is associated with a linear motion model H_t. From now on, we considered the camera path can as follows:

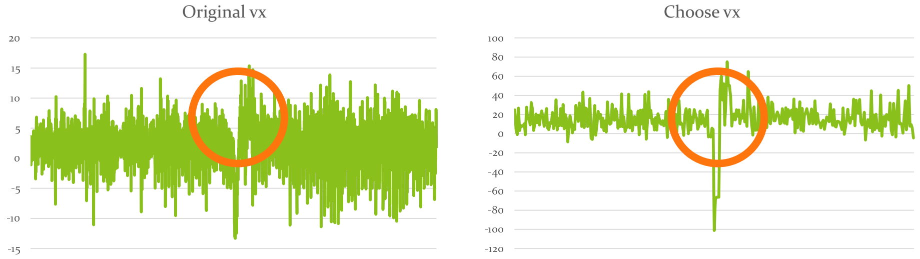
$$F_t = F_{t-1} H_t \Rightarrow F_t = H_1 H_2 ... H_t \tag{5.10}$$

F_t can calculated by matrix multiplication iteratively. We generate the original video and its related time-lapse video frames position trajectory, velocity and acceleration transformations. Figure 5.2, figure 5.3 and figure 5.4 show an example of this analysis. Time-lapse video frames total movement distance drop a little with the original frames. But the proportion of frames and motion distance improved significantly. In addition, we observed the standard deviation of the frame motion increased much especially in large motion in original video frames sequence. This cause the time-lapse video’ s velocity and acceleration values sensitive for the original large motion. This large always not the camera shaking because we filtered in frame choose step. It represent the camera motion actively. It is always conclude multiple frames motion continues but sharp in time-lapse video frames sequence. We need to maintain this sharp camera motion to increase the overlap area in stabilized video.



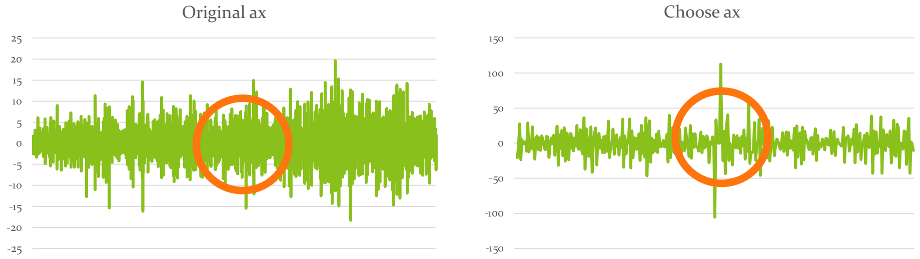
(a) Original Position Trajectory (b) Time-lapse Position Trajectory

Figure 5.2: Position



(a) Original velocity Trajectory (b) Time-lapse velocity Trajectory

Figure 5.3: Velocity



(a) Original Acceleration Trajectory (b) Time-lapse Acceleration Trajectory

Figure 5.4: Acceleration

We propose an adaptive Gaussian smooth to camera paths approach to balance the overlap area and camera smooth. The smoothed camera trajectory is much more fit the trend of original camera trajectory. Gaussian smooth is common used [8, 12] in camera path smooth in video stabilization. A Gaussian smooth kernel can proposed as follows:

$$G_{1D}(x,\sigma) = -\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$$

The σ determines the width of the Gaussian kernel. The difference velocity and acceleration information of the original video frames sequence and the time-lapse video frames sequence can be used to define the Σ at the frame t. The function σ_t represent the σ at the frame t is

$$\sigma_t = G_\alpha\left(k_1\left\|V_c - \frac{1}{num(S)}\sum_{o\in S_t} V_o\right\| + k_2\left\|A_c - \frac{1}{num(S)}\sum_{o\in S_t} A_o\right\|\right) \tag{5.11}$$

k_1, k_2 represent the weight of velocity and acceleration term for sigma. V_c represent the velocity of this frame smoothed. S means the sequence of time-lapse frames sequence. We consider the difference between the current frame velocity and the time-lapse frame sequence average velocity to define the current frame σ. We consider the same approach to the acceleration. G_σ is a conversion function to restrict the final result to mapping to a range of σ value.

We also consider the influence of windows of discrete Gaussian. We follow the idea of bilateral filter [5, 17] and design it by this functions :

$$w_t = G_\beta\left(\sum_{r\in\omega_t} G_1(\|r-t\|)G_2(\|C(r)-C(t)\|)\right) \tag{5.12}$$

$\|r-t\|$ represent the distance between frame r and frame t in the time-lapse video frames sequence serial number. $\|C(r)-C(t)\|$ represent the transformation distance between the two frames in the time-lapse video frame. G_1, G_2 are Gaussian function to smooth the contribution of the two terms. ω_t is the largest windows this approach used and G_β is a mapping function that the total weight result is mapping to the right region of

Videos	Frames	Ours costs	Naive costs	Decrease rates
1	361	242.092	322.33	0.751069
2	1723	270.352	565.495	0.47808
3	1931	334.108	527.066	0.633902
4	2769	622.742	808.335	0.770401
5	3121	210.662	430.451	0.489398
6	3430	540.893	808.795	0.668764

Table 5.1: Table of Costs Analysis

the window size.

5.3 Time-lapse Video Stabilization Evaluation

Figure 5.5 shows the time-lapse framework of the egocentric video. We take the Gopro Hero 4 to the chest, bicycles and quadcopters to capture the 2.7K Full HD videos. For each original video, we extract frame features by calculate optical flow to measure the shaking costs. We compute the flow for features of the first input image using the Lucas and Kanade algorithm [18].

Each original video we used the follow processing to generate stabilized time-lapse video and generate naive time-lapse video. We compared the shaking costs drop down between our method and the naive method. This features informations are used to costs modeling and frame selection to generate the time-lapse video. The retain frame features also conserved to use in the motion estimation step to calculate time-lapse video transformations and generate camera motion trajectory. Then, the steps are same as the basic video stabilization steps detailed in chapte 5.

To evaluate the effect of our approach. We create multiple videos by taking in different kinds of scenes. For example, The video is taken on opening lawn have the wide vision including large area sky, large area flat glass ground and plants or others near horizon. The video record the corridor scene having the composition of perspective. Taking videos on the road have the parallax problems between the front features and the side features. We take much video include multiple kind of scenes in the national taiwan university campus.

Table 5.1 shows the examples costs compare between the naive time-lapse videos and

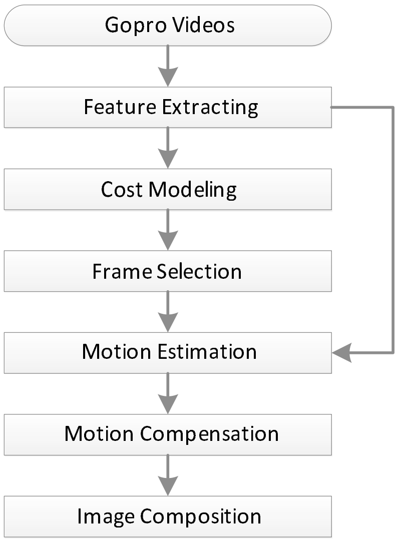


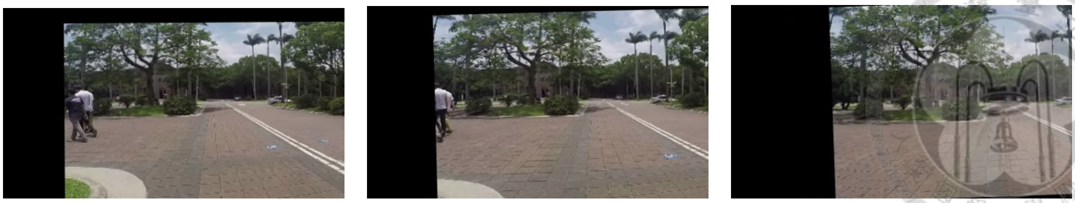
Figure 5.5: Time-lapse Video Stabilization Framework

the time-lapse videos created by our approach. We choose different kind of scenes to analysis include taken the ourdoor wide vision scene video, the video taken on the road, the video with large angle of rotation, record the corridor scene's video. We observation that the costs we defined dropped over the one fourth by our method compare with the naive frame selection.

We defined three kinds of costs which influence the frame selection. We should adjust the parameter α, β in the edge costs equation to balance the influence of the stabilization, velocity, appearance costs. The best combination we defined is $\alpha = 10$ and $\beta = 1$ to fit the three costs result values range equal. We can adjust the β to defined the camera motion uniform is important or not for your application.

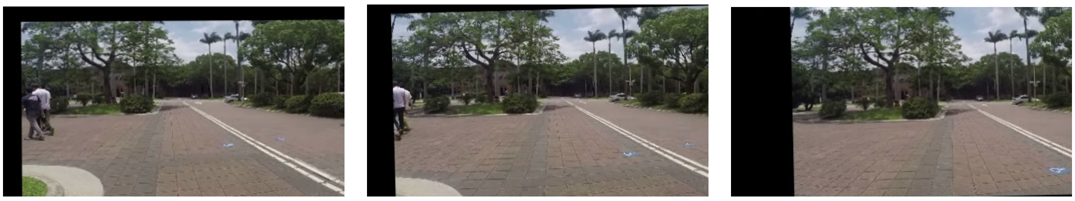
5.3.1 Single Camera Path Optimization Analysis

We propose an adaptive gaussian smooth method to image motion trajectory. According to the characteristic of the original video's velocity, acceleration and the related time-lapse video's velocity, acceleration. It recognize the positive wide angle camera rotation and retain the camera original motion at that position. The key of this adaptive adjust is the characteristic map to the gaussian filter parameters. In experiment, consider the σ_t, we define the $k_1 = 1$ and $k_2 = 2$ to balance the contribution of velocity and acceleration. In the formula of w_t calculating, we define the $G_1 = 1$ and $G_1 = 0.1$ to normalize the



(a) (b) (c)

Figure 5.6: Gaussian Filter



(a) (b) (c)

Figure 5.7: Adaptive Gaussian Filter

values. G_α and G_β used to mapping the normalized contribution to the gaussian filter parameters region defined. we set $\sigma = 5$ and $w = 60$ to initial the gaussian filter. we defined the G_α and G_β is a linear mapping in experience. linear mapping is enough distinguish the σ, w represented positive rotation motion or not. Figure 5.6 show the fixed gaussian trajectory smooth result in a position with positive wide angle rotation. we set the $\alpha = 5$ and $w = 60$ in this example. Figure 5.7 show the adaptive gaussian trajectory smooth result in the same position with Figure 5.6. Less empty information area and stabilization enough in this situation by using our adaptive gaussian filter.

5.4 Time-lapse Panoramic Video Stabilization

The approach above proposed is a general method suitable for time-lapse egocentric video generating and stabilization. Record multiple videos synchronization to generate panoramic videos also too long to boring. We improve the above time-lapse method to suitable for panoramic videos.

We suppose the multiple camera system synchronized already. We defined the frame i from camera k is F_i^k. Multiple frames from different camera at each time we obtain as an entirety to calculate the stability costs to influence the frame choose. We combine all the frames shakiness costs in the panoramic original video frame selection. The velocity and

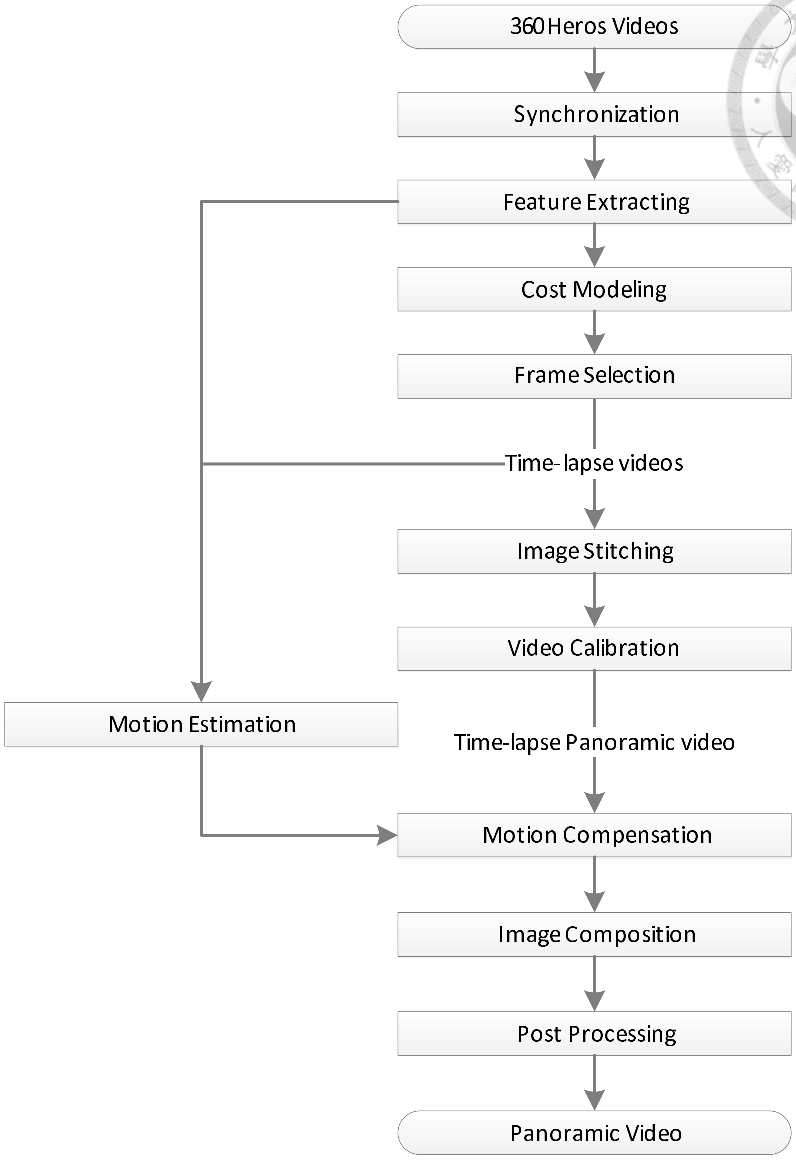
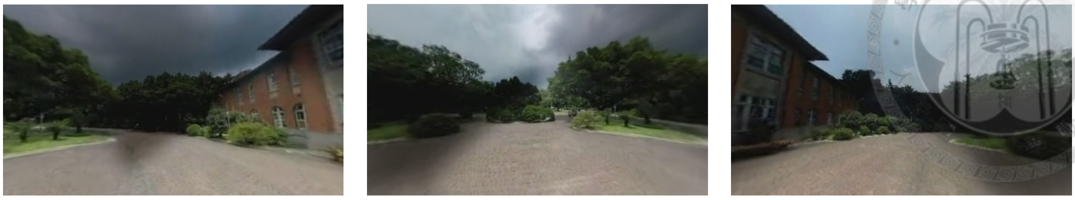


Figure 5.8: Time-lapse Panoramic Video Stabilization Framework

the appearance costs unchanging in this panoramic time-lapse model. Then, we use the directed hidden Markov model to find the less costs path to choose the frames to generate the final time-lapse panoramic video.

This is the first paper propose a complete time-lapse panoramic video stabilization framework in Figure 5.8. We use 360Heros to fix the position relation of the six Gopro Heros 4 cameras recording the videos. Synchronization also is very important for this whole processing. We accept the motion synchronization method VideoStitch provided to the six related original videos. Note that we need to try motion synchronization many times to get a stable results. When we obtain the synchronization original videos, we

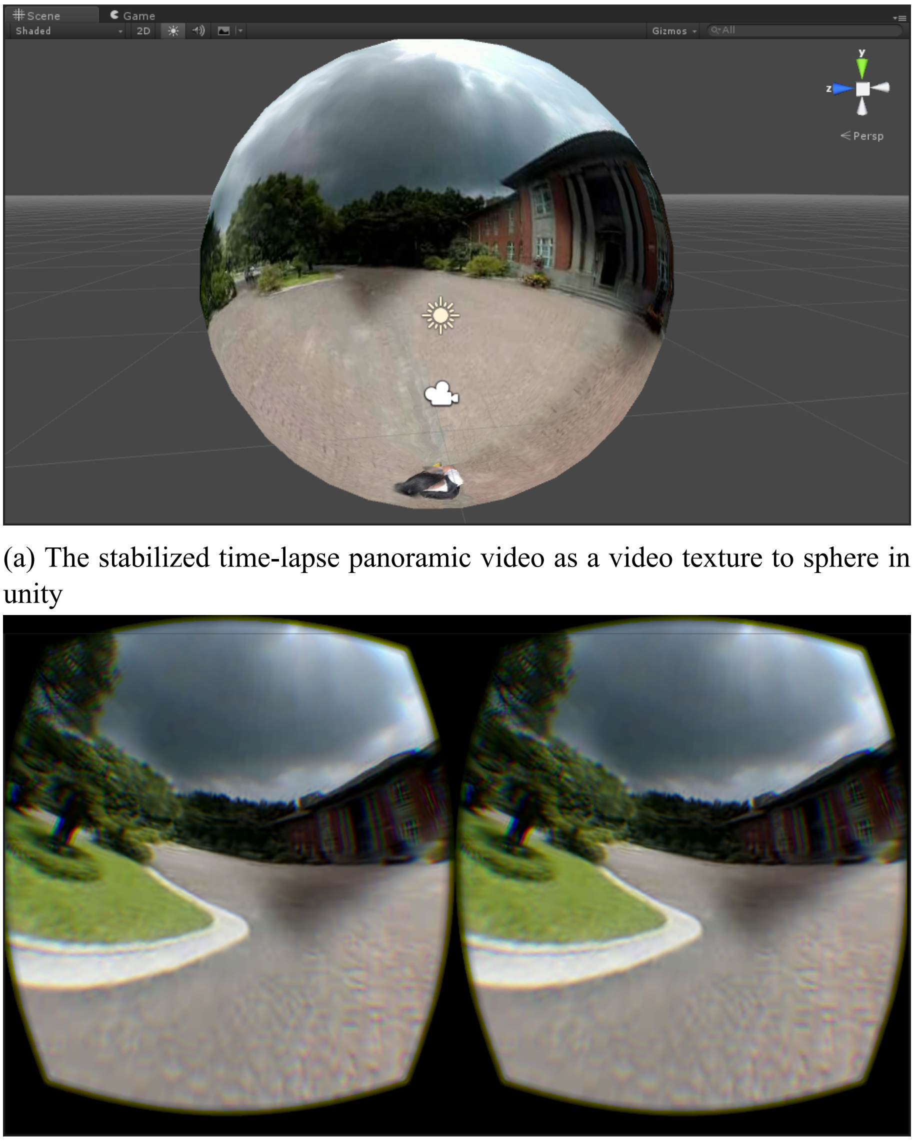


(a) View 1

(b) View 2

(c) View 3

Figure 5.9: Kolor Showing



(a) The stabilized time-lapse panoramic video as a video texture to sphere in unity

(b) Oculus showing the virtual reality effects

Figure 5.10: An example of using stabilized time-lapse panoramic video in virtual reality user experience.

create the time-lapse original videos first. We use the method proposed in chapte 5 create the time-lapse original videos. The key for this proposed method is guarantee the selected frames numbers in all synchronization original video frame sequence is same. This make sure we can use a calibration template to generate the time-lapse panoramic video with the time-lapse original videos. We include the time-lapse original videos to VideoStitch and generate the time-lapse panoramic video by its video calibration functions. We can apply the method chapte 4 proposed to stabilized the time-lapse panoramic video in the end.

To evaluation the result of time-lapse panoramic video stabilization. We prepare two tools to watching the result videos. The first is KolorEyes 1.4. It provide a quick method to watch the panoramic videos we generated. We can preview our time-lapse panoramic stabilized video in this tool. We verify our results maintain the panoramic view and have good result in original frame calibration showing in Figure 5.9.

We also create a virtual reality tool to take the immersive experience to users. We create a 3D sphere in unity and the warped video frames sequence as a texture to mapping to the inside sphere. We put the oculus camera in the center of this sphere in unity by oculus unity developing toolkits. Users can immersive experience our panoramic scenes with the time warping. Figure 5.10 show the examples user watched in oculus.

Chapter 6

Conclusions

We are the first proposed a time-lapse panoramic video stabilization approach. In this paper, We elaborate a basic video stabilization framework and propose how to use Laplacian mesh warping method to correction the parallax of the frames first. Then, we provide three video stabilization frameworks. The first is the complete time-lapse egocentric video stabilization. The second is the complete panoramic video stabilization framework. The third framework we proposed is used for time-lapse panoramic video stabilization.

In time-lapse video stabilization, we propose an optimize frame selection method by consider camera motion averaging, video times scaling uniform and bundles camera paths stabilization for panoramic videos. It guarantee the global compression ratio of the time and maintain the total camera motion for the final time-lapse video. We also propose an optimize single camera path method for time-lapse video by adaptive Gaussian smooth considered the time-lapse video and corresponding original video’ s velocity and acceleration characteristics changing. In panoramic video stabilization, we consider to use 2D Laplacian mesh warping to generate subdivision camera paths mesh for panoramic video stabilization. We perform a time-lapse panoramic videos stabilization result well by optimize the above methods we proposed. We evaluate the effectiveness of our approaches using serval input videos in the end.

Bibliography

[1] Yasuyuki Matsushita, Eyal Ofek, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 50–57. IEEE, 2005.

[2] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(7):1150–1163, 2006.

[3] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung. Video stabilization using robust feature trajectories. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1397–1404. IEEE, 2009.

[4] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):4, 2011.

[5] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78, 2013.

[6] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 4209–4216. IEEE, 2014.

[7] Yair Poleg, Tavi Halperin, Chetan Arora, and Shmuel Peleg. Egosampling: Fast-forward and stereo for egocentric videos. *arXiv preprint arXiv:1412.3596*, 2014.

[8] Neel Joshi, Mike Toelle Wolf Kienzle, and Michael F. Cohen Matt Uyttendaele. Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics (TOG)*, 2015.

[9] Mostafa Kamali, Atsuhiko Banno, Jean-Charles Bazin, In So Kweon, and Katsushi Ikeuchi. Stabilizing omnidirectional videos using 3d structure and spherical image warping. In *IAPR Conference on Machine Vision Applications*, pages 177–180, 2011.

[10] Neel Joshi, Sisil Mehta, Steven Drucker, Eric Stollnitz, Hugues Hoppe, Matt Uyttendaele, and Michael Cohen. Cliplets: juxtaposing still and dynamic imagery. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 251–260. ACM, 2012.

[11] Alexandre Karpenko, David Jacobs, Jongmin Baek, and Marc Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR*, 1:2, 2011.

[12] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3d video stabilization. In *ACM Transactions on Graphics (TOG)*, volume 28, page 44. ACM, 2009.

[13] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 225–232. IEEE, 2011.

[14] Andrey Litvin, Janusz Konrad, and William C Karl. Probabilistic video stabilization using kalman filtering and mosaicing. In *Electronic Imaging 2003*, pages 663–674. International Society for Optics and Photonics, 2003.

[15] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)*, 33(4):78, 2014.

[16] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3):1134–1141, 2005.

[17] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[18] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.