Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Hongyuan Cai

Entitled
Video Anatomy: Spatial-temporal Video Profile

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Jiang Yu Zheng
Chair

Elisha Sacks

Mihran Tuceryan

Voicu Popescu

Xavier Tricoche

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Jiang Yu Zheng

Approved by: Sunil Prabhakar / William J. Gorman 03/22/2013
Head of the Graduate Program Date

VIDEO ANATOMY: SPATIAL-TEMPORAL VIDEO PROFILE

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Hongyuan Cai

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2013

Purdue University

West Lafayette, Indiana

To My Parents

ACKNOWLEDGMENTS

This study and graduate dissertation could not be completed without the kind attention and careful guidance from my adviser Dr. Jiang Yu Zheng. His serious scientific attitude, the spirit of rigorous scholarship and his work style of always pursuing improvement deeply influenced and inspired me. From the topics selection to the final completion of the dissertation, Dr. Zheng was always careful to give me guidance and tireless support. Over these six years, Dr. Zheng not only carefully guided me on academic field, but also took great care and be concerned about my life here, here I would like to extend my sincere thanks and great respect to my adviser Dr. Zheng. He is a generous man, a dedicated mentor and a great professor.

Sincere thanks should also go to my committee members Dr. Voicu Popescu, Dr. Mihran Tuceryan, Dr. Elisha Sacks, and Dr. Xavier Tricoche, for serving as the members of my Ph.D. thesis committee and supporting my academic goals. It is precisely because of their help and support that I can overcome the difficulties one by one until the successful completion of the dissertation.

This dissertation will be completed in time, now I feel unable to calm down. From the beginning of this dissertation subject to the successful completion, many honorable teachers, classmates, friends and department staffs gave me speechless help throughout my graduate study. All their kind support is gratefully acknowledged. Finally, I would also like to thank my family who educated me to be strong in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Cai, Hongyuan Ph.D., Purdue University, May 2013. Video Anatomy: Spatial-Temporal Video Profile. Major Professor: Jiang Yu Zheng.

A massive amount of videos is uploaded on video websites, smooth video browsing, editing, retrieval, and summarization are demanded. Most of the videos employ several types of camera operations for expanding field of view, emphasizing events, and expressing cinematic effect. To digest heterogeneous videos in video websites and databases, video clips are profiled to 2D image scroll containing both spatial and temporal information for video preview. The video profile is visually continuous, compact, scalable, and indexed to each frame. This work analyzes camera kinematics including zoom, translation, and rotation, and categorize camera actions as their combinations. An automatic video summarization framework is proposed and developed. After conventional video clip segmentation and video segmentation for smooth camera operations, the global flow field under all camera actions has been investigated for profiling various types of video. A new algorithm has been designed to extract the major flow direction and convergence factor using condensed images. Then this work proposes a uniform scheme to segment video clips and sections, sample video volume across the major flow, and compute the flow convergence factor, in order to obtain an intrinsic scene space less influenced by the camera ego-motion. The motion blur technique has also been used to render dynamic targets in the profile. The resulting profile of video can be displayed in a video track to guide the access to video frames, help video editing, and facilitate the applications such as surveillance, visual archiving of environment, video retrieval, and online video preview.

1 INTRODUCTION

1.1 Objectives

Digitized visual memory and its sharing can assist in such human activities as cognition, decision-making, location finding, process execution, and social activity. With the explosive increase of video data and sharing to the web, smooth video browsing, editing, retrieval, and summarization are highly in demand. Increasing numbers of digital cameras and cell phones along with large storage devices have created huge video archives available for cataloging personal experiences. Moreover, tremendously large video datasets are recorded for experiments and surveillance at research institutes, business sites, public areas, and critical infrastructures. Small wearable cameras are available to law enforcement, health care, and retail establishments for constant recording of daily events and people. The resulting footage is defined as egocentric videos. The real challenging problem now is how to conveniently view and navigate video data and how to effectively use the video information for various applications. Vast amounts of time has been used in searching and screening video. Tools that can automatically find the most relevant content according to our interests, specified manually or learned from the viewing history, are desired. Two common approaches in accessing the large dataset of video so far are interactive browsing and automatic retrieval.

How can the viewer have a glance at an entire video clip and then index to each individual frame from a video digest? Because dynamic video frames have overlaps on scenes, the reduction of redundant pixels from a video clip to a 2D image belt becomes possible in video summarization. We can thus analyze the camera motion information in the video to sample static scenes only once. Such a sampling dramatically reduces

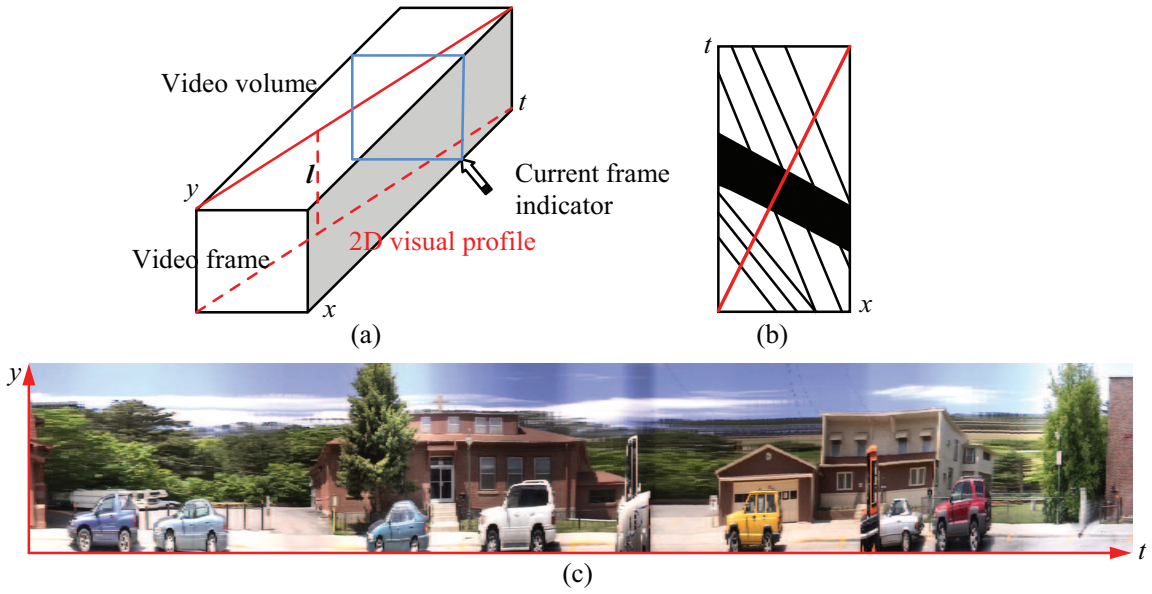


Figure 1.1.: Video volume and a possible cutting of diagonal slice across the major flow in the video clip. (a) Video data volume with an indicator indicating the current frame. The blue rectangle shows a frame plane. The video data volume has two spatial dimensions x and y, and one temporal dimension t indicating the order of frames. (b) a condensed image showing flow traces of scenes, (c) a generated profile of a video clip from a camera translating sideways. The red plane in (a) and the red line in (b) show the sampling plane formed by moving a sampling line and the path of the sampling line.

the data size and the influence from the camera motion. This work creates a 2D profile from raw video data, which is an image belt that contains one axis as the timeline, and the other indicating a spatial dimension in the video frame. It is a novel view of video from side of the video volume (instead of the conventional method that looks from the front) that can (1) index to each frame for video editing, (2) provide a view of entire scene space that a video captures for browsing. The sampling (slice cutting) strategy avoids image matching, flow segmentation, and other complex procedures to achieve the robustness. The image belt can be embedded as a video track in video production software, displayed in web-page for browsing, and used as an intrinsic

video space for retrieval. Moreover, the created video summary keeps the temporal order of video and is scalable in time starting with a resolution higher than spatial or temporal indexing. Figure 1.1 displays a possible setting of such a profile cutting in the spatial-temporal video volume.

1.2 Contributions

The first major contribution of this dissertation is to solve the fundamental problems on

1. Whether the profile is possible to be extracted for all types of camera motions.
2. How to obtain the profile of video shot/clip from a general camera motion.
3. What kinds of information are presented in the profile.
4. How to acquire the profiles of video clips automatically and efficiently.

To achieve above goals, we design a path of sampling line in the video volume to yield a planar or curved cutting slice in order to reveal the video content in the video volume. It is implemented by sweeping a sampling line across the video volume. Our criteria to cut a profile from video are to

1. Include all the stable background space that a video clip captures.
2. Show meaningful shapes and identities.
3. Reduce the distortion of target scenes in the generated profiles, because the profile obeys a different scene projection from the normal perspective projection.

We analyze the typical camera works (motion styles), their underlying kinematics, and the generated optical flow in the video to design the cutting and slicing strategy. For automatic profile acquisition, this work further develops an efficient algorithm to detect the global flow in the video using an intensity condensing approach. Our slicing of the video volume is designed to cut through every frame in the video volume

Figure 1.2.: Profile of consecutive clips of a concert video from YouTube. (Top) Vertically condensed image (will be explained later) from the video where clip boundaries are visible. (Lower) Profile cut from the trajectory in the condensed image above. The horizontal axis indicates the frame position in the clip. This profile shows both temporal and space/shape information in the clip as well.

so that the continuous profile indexes to frames, which is impossible for the key frame approach. Following our designed cutting strategy, the spatial information such as static environment and dynamic targets in the video is also visualized in the profile, although some deformation and changes in spatial ordering are brought in. Moreover, the created video summary keeps the temporal order of video and is scalable in time starting with a resolution higher than spatial or temporal indexing. Figure 1.2 gives an example of such profiles.

The second major contribution of this work is to realize an automatic video profiling for the video database. It solves the problems on

1. The segmentation of sections in clips corresponding to smooth camera operations.

2. Understanding major motion by detecting global flow.

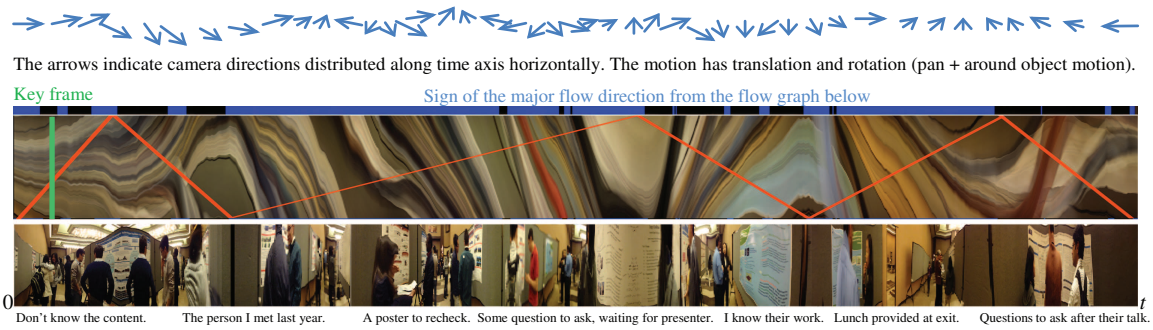


Figure 1.3.: Profile of a highly dynamic video clip taken from a wearable camera. The viewer pans left and right while walking through a conference poster session. Upper part is a flow graph with a sampling trajectory determined from blue bars that indicates flow directions. The profile reveals entire scenes in the temporal domain subject to some changes in aspect ratio and minor shaking. The profile can index to frame number along the time axis. Notes are tagged in temporal order. No scene in the video is missed in this presentation of video.

3. Real time profiling of video clips, and normalization of the shape in the video profile for the interface.

Figure 1.3 shows such an example of video profile from a camera that is performing multiple actions in walking along a poster aisle in a conference. More general types of camera operations as well as their combinations are tackled in this work. It enables a fast scanning of the video database for profiling variety of clips with different camera motions.

The generated profile recovers the more efficient and intrinsic *scene space* by removing the global camera motion (Fig.1.4). Most of the methods so far use discrete video frames over a partial scene space. The overwhelmed optical flow extracted in every frame may not reflect the true scene dynamics but only caused by the camera motion. The global view of scene space and temporal order preserved in the profile will provide the critical information for classifying videos of the same scene even from different camera actions. The 3D video volume is thus reduced to a 2D profile. All

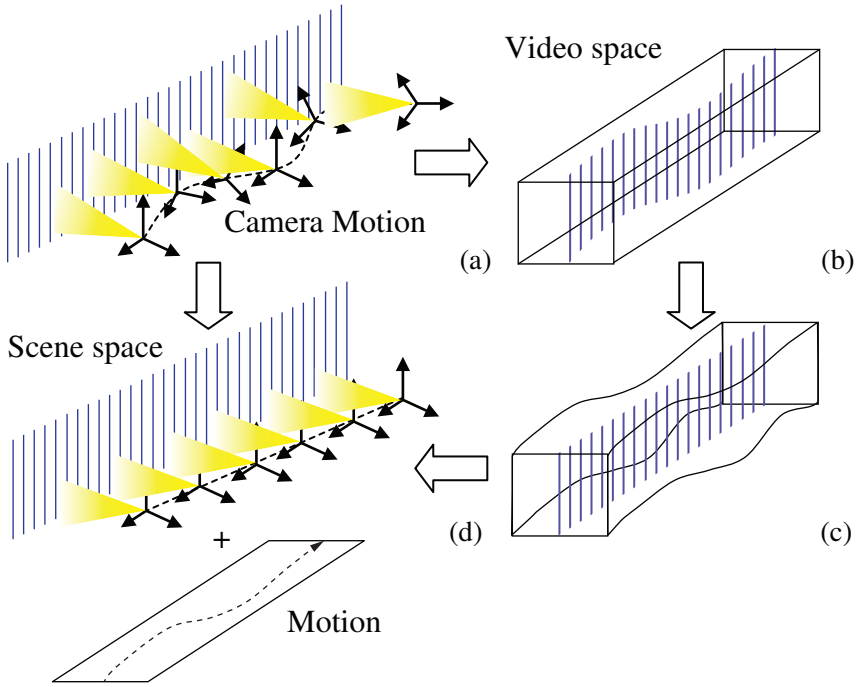


Figure 1.4.: Cutting slice to maintain scene space. (a) Dynamic camera moving in scene space records a video volume, (b) located slice in video volume reflects non-redundant scenes in space, (c) a restored slice without influences from camera motion, (d) ideal camera projection of scenes.

the scenes stably appearing in the video clip are included in the profile once for preview. We do not create multiple copies of objects as onion-skin or strobe image [1–3], because it may generate confusion with a group activity. Inversely, by enforce the temporal order on the slice cutting, a scene point visible at a time must appear in the corresponding frame as well as its adjacent frames in the clip. This facilitates further access to the frames.

The advantage of the profile of video lies in its aspects of: (I) compact size, (II) reflecting temporal information, such that dynamic events even appearing in the same space can be listed in temporal order without overlapping, which is impossible for a spatial mosaicing method that only aims at enlarging the field of view. Annotation of dynamic events can be easily done along the time axis. (III) preserving shapes

to some extent, (IV) embracing static background and dynamic foreground, and (V) robustness in processing all types of videos. The acquisition of video profile will produce stable results based on our robust major flow extraction and a designed cutting strategy. This will allow us to compare profiles efficiently before examining videos, even in near duplicated video detection.

In contrast to previous works, the proposed video profile is easy to be embedded into video software to enhance video editing, retrieval, analysis, and visualization in general. It is a spatial-temporal slice that can overcome the problems of the previous indexing methods in resolution, camera motion types, and robustness. The proposed framework avoids image matching, scene segmentation in each frame, and other time-consuming procedures in mosaicing to achieve the robustness, and is a global approach depending on explicit camera motion styles, in contrast to the inter-frame optimization that achieves the spatial integrity locally [4] but may get problem on comprehensive camera action such as around-object-rotation with the camera focusing on a target.

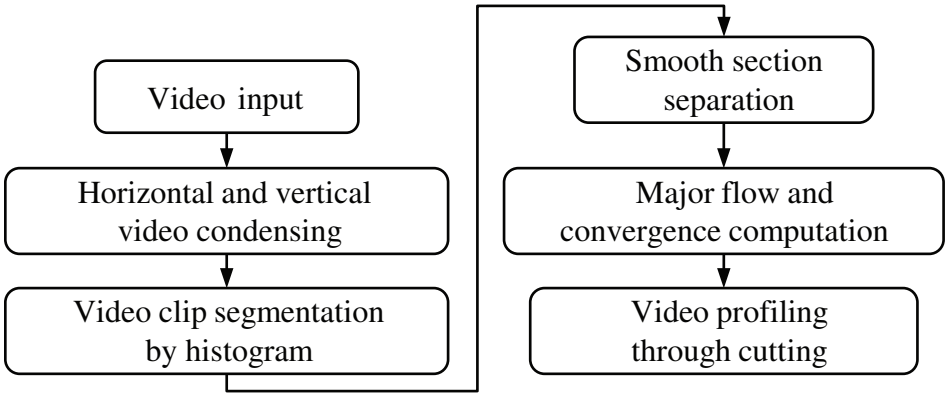


Figure 1.5.: The overall flow chart of the system.

1.3 Organization

The next chapter comprehensively surveys the spatial and temporal indexing techniques. As the flow chart of Fig.1.5 indicates, Chapter 3 analyzes the camera motions, presents major and minor flow, and introduces a new technique to estimate the camera motion effectively and efficiently. Chapter 4 will address the segmentation of video clips to sections with smooth or monotonic camera motion. Chapter 5 gives an efficient method to automatically understand and identify the camera motion including major flow and convergence factors. Chapter 6 proposes a uniformed framework for video volume cutting approaches on a general camera alignment and motion that might be the combination of simple motions. We then apply the method to video clips with composite camera motions or concatenated video clips. Chapter 7 addresses two major shape improvements for generated video profile as effective post-processes. Chapter 8 describes the experiment, provides results on various videos, builds GUI for PC and mobile devices, which is followed by a discussion in Chapter 9 and conclusion in Chapter 10.

2 RELATED WORK

In light of the video deluge on the video sharing website and surveillance databases, the need to represent the video content raises a fundamental problem: is there a way to give the viewer effective preview and, since video itself is a sequential data, the way of fast indexing/accessing. The video itself can be viewed as a 3D spatial-temporal data volume which contains one temporal dimension and two spatial dimensions. Among the studies conducted so far, the indexing can be categorized into *Spatial Indexing* methods and *Spatial-temporal Indexing* methods (Fig.2.1), which will be surveyed in the following sections. This work tries to take advantage of spatial-temporal profile by designing the path of scanline based on the flow characters in the video volume, along which the generated profile can compress and reveal most of the repeating video contents in discrete frames.

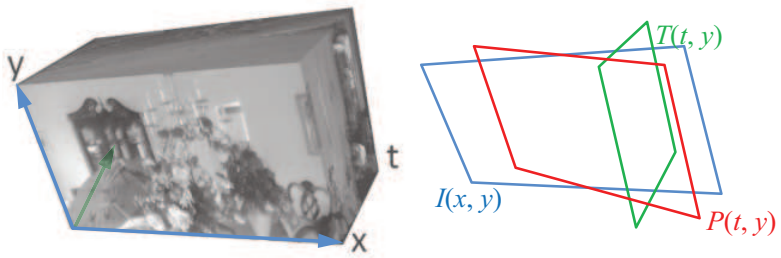


Figure 2.1.: Spatial-temporal volume of a video shot/clip, and spatial frame $I(x, y)$ in blue, temporal slice $T(t, y)$ in green and a spatial-temporal slice $P(t, y)$ in red for video profile.

2.1 Spatial Indexing of Video

Most of the video indexing uses *key frame*s for video clips/shots and their collection as story boards or scenes, and it is fine for most of the static TV programs that switches between static scenes. The *spatial indexing* presents key frames and its extension on video frame mosaicing extend the *field of view* to *multi-perspective view* in a larger spatial domain. Various algorithms have been developed to extract stable and representative frames for key frames in a *storyboard* or *tapestry*. 2D panoramas are generated by stitching camera rotating video frames on either static or dynamic scenes. A *montage* image overlaps dynamic actions from a static camera in the spatial domain. These methods selectively mosaic regions from different frames into one summary image.

2.1.1 Static Methods

The static methods are among the earliest attempts developed to index the video. They are mainly focusing on the presentation of the entire static background and emphasizing the story of the video. There are four major techniques in this category.

1. Key Frame: A key frame in animation and film-making is a drawing that defines the starting and ending points of any smooth transition [5]. The key frame technique is among the earliest attempts for content-based video analysis. The techniques that can be applied to still image can be used directly on video represented as a selected sequence of images (key frames). The video is first temporal-partitioned in sub-sequences which contains a homogeneous action in time and space for indexing purposes. The partitioned segments are usually called *shots*. In each homogeneous partition, a key frame is selected based on the color distribution, usually the temporal centroid of the camera shot. These key frames are then laid out in temporal order with hard borders to represent the content of the video [6]. This technique is considered as the early transition

from the still image analysis to the video analysis. Most of the video sharing website [7] and video players are still using it as preview of the video content.

2. Key Frame Mosaic: The mosaicing methods can be viewed as an spatial extension to the key frame methods. The key frames of videos taken by panning (rotating) or translating cameras usually share some common scenes in a single camera shot. The mosaicing techniques find ways to stitch key frames together based on the partial similar information to form a single result. The mosaicing usually expands the field of view of the key frame [8, 9]. A good example is that a panorama image stitched from images taken in various directions gives the viewer a surrounding view of the scenes captured. The key frame mosaicing methods allow fast clustering of scenes into physical settings, as well as further comparison of physical settings across videos.

3. Storyboard/montage: In this type of technique, the key frames connected do not necessarily have overlaps of physical scenes. A camera transition from one scene to another is allowed. Outlines, arrows, and text describing are used to annotate the motion in the scene and transition between camera shots if there are any. In storyboards, a significant time interval of the video content can be expressed all at once [1]. A similar method is called the video tapestry. In the tapestry, there are no hard borders between discrete moments in time, and a user can zoom smoothly into the image to reveal additional temporal details. It's roughly chronological, presenting events in a spatial order that corresponds to their temporal order in the film [10].

4. Scene Manifold: This technique scans a sampling line within the space-time volume of the video to guarantee the least image distortions possible. The scanline traces the scene outline. Every local neighborhood within the manifold formed by the scanline resembles some image patch. The shortest path of the movement of scanline is solved to produce the globally optimal solution based on spatial scene appearance. Constraining appearance rather than geometry

gives rise to numerous new capabilities, such as dealing with camera parallax. Any small part of it can be seen in some image even though the manifold spans across the whole video. Thus it can deal seamlessly with both static and dynamic scenes, with or without 3D parallax [4].

2.1.2 Dynamic Method

The dynamic methods mainly augment the static methods with certain presentation of the foreground. A method called Motion Panorama projects the foreground objects on the mosaic and on the video frame are briefly discussed below.

First introduced in [11], the motion panorama generalize the static panorama method. This technique can only be used with camera zoom and pan/tilt. It uses frame-to-frame alignment as a combination of feature-based, rough motion segmentation, and color-based direct method. Based on this, the dynamic building of a background representation as well as an efficient segmentation of each image such that moving regions of arbitrary shape and size are overlaid in temporal order on the static background. The technique is also introduced in [12] for qualified web videos which meets three criteria as the author proposed. However, the former paper claims that the static portions of the scene are not necessarily dominant because of smaller number of feature points used, while the later requires that the background is dominant in a video. Similarly, [13] creates a dynamic narrative, which could be played and skimmed in real-time. Graph cut technique is also used to composite the narratives from different camera shots which is similar to video tapestry. [14] describes an approach for simulating apparent camera motion through a 3D environment. A single multi-perspective panorama is used to incorporate multiple views of a 3D environment as seen from along a given camera path. When viewed through a small moving window, the panorama produces the illusion of 3D motion. In addition, [15] uses inter-frame motion estimation to build an image mosaic that completely stabilizes the camera movement to create a panoramic image, and then animates a virtual

camera that views this mosaic. In this fashion, the casually captured videos can be post-processed to improve apparent camera movement caused by hand shaking and bumpy camera movement.

2.1.3 Drawbacks of the Spatial Indexing Techniques

There are several drawbacks of the spatial indexing methods:

1. Key frame is a coarse representation of the content of the video.

2. Lack of temporal order in such an integrated video space. It can index to a clip rather than to a frame directly, which is not further useful for video editing.

3. Some summary aims at visualizing motion by duplicating targets. It becomes cluttered if the video clip lasts long or targets have a high complexity [3].

4. Camera motions are only limited to static and pan. If the motion parallax varies as in a translating video, it can only succeed on the scenes with homogeneous depth or color [16]. This limits such approach to be applied to general video database.

5. The background matching and foreground segmentation are not robust for complex and dynamic scenes. Instantaneous events and non-rigid shapes such as fire, smoke, water, and so on may cause more problems. Most of these mosaicing only work on single shots of video so far. A typical one on real video database is in [17], which packed two types of video icons: panoramic (pan/tilt) and key frame icons together in a space-efficient manner.

2.2 Spatial-temporal Indexing of Video

In contrast, the *spatial-temporal indexing* strictly reflects temporal information by collecting a pixel line (small image patches) from each frame, which achieved results for camera rotation and translation with a fixed slit and dynamic slits.

2.2.1 Fixed Slit Methods

In the fixed slit methods, the slit set are static in spatial domain (remains in the same position and orientation in each frame) or in spatial-temporal volume (the slit forms fixed shape spatial-temporal manifolds). Two fixed slit methods are briefly discussed below.

1. Route Panorama: [18] creates a route panorama by scanning scenes continuously with a fixed virtual slit in the camera frame to form image memories. For each camera image in the video sequence, a vertical slit view (or image memory) is copied at a fixed position and pasted together consecutively to form a long, seamless 2D image belt. The 2D image belt can be transmitted via the Internet, enabling end users to easily scroll back and forth along a route. The process of capturing a route panorama is as simple as recording a video on a moving vehicle and can be done in real time. The generated image belt with its consecutive slit views pieced together has much less data than a continuous video sequence. A special type of Charge-Coupled Device sensor called line sensor reads temporal data from the device array continuously and forms a 2D image profile. Compared to most of the sensors in the current sensor networks that output temporal signals, it delivers more information such as color, shape, and event of a flowing scene. On the other hand, it abstracts passing objects in the profile without heavy computation and transmits much less data than a video. [19] revisits the capabilities of the sensors in data processing, compression, and streaming in the framework of wireless sensor network. Sensor setting, shape analysis, robust object extraction, and real time background adapting

have been studied to ensure long-term sensing and visual data collection via networks. All the developed algorithms are executed in constant complexity for reducing the sensor and network burden. A sustainable visual sensor network can thus be established in a large area to monitor passing objects and people for surveillance, traffic assessment, invasion alarming, etc.

2. Adaptive Manifold: In this technique, thin strips (scanline) are projected multi-perspectively from the images onto manifolds which are determined dynamically based on the motion of the camera. Manifold mosaicing can be performed by computing the manifold explicitly from the ego motion of the camera obtained from auxiliary sensors, and projecting the frames onto that manifold. Alternatively, this projection can be done implicitly by the process of cutting and warping strips, and without explicit computation of the manifold. Manifold mosaics represent the entire environment of a video shot in a single, static, image. This single image can be used as a summary of the video clip for video browsing, or as a compressed representation of the shot which can be approximately re-generated from the mosaic given the stored motion parameters. While the limitations of mosaicing techniques are a result of using predetermined manifolds, the use of more general manifolds overcomes these limitations [20].

2.2.2 Dynamic Slit Method

In the dynamic slit methods, the movement of the sampling slit can be adaptive to the scene change. The position of the sampling strip varies as a function of the explicit input camera location. The new images that are generated this way correspond to a new projection model defined by two slits, termed the Crossed-Slits (X-Slits) projection. In this projection model, every 3D point is projected by a ray defined as the line that passes through that point and intersects the two slits. The intersection of the projection rays with the imaging surface defines the image. The author claims that X-Slits mosaicing provides two benefits. First, the generated mosaics are closer

to perspective images than traditional pushbroom mosaics. Second, by simple manipulations of the strip sampling function, the user can change the location of one of the virtual slits, providing a virtual walkthrough of a X-slits camera. This can be done without recovering any 3D geometry, so that no camera calibration is needed [21].

2.2.3 Drawbacks of the Spatial-temporal Indexing Techniques

The shortcomings are as follows:

1. The created image from a short clip with fast motions has a low temporal resolution.

2. It generated views with a different projection from normal perspective projection.

3. It requires deshaking on the original video [15] or on the generated image [22,23] for a non-smooth camera motion.

4. Motion type has to be given in advance, even though it can handle more types of motion than spatial mosaicing. There is no work so far that can sort out video database to identify a type of camera motion each clip was taken. Besides 2D video summaries, a volume visualization method has been proposed for summarizing video sequences [24]. However, the scene type is limited to static camera case and interaction is required for exploring the details.

2.3 Other Methods

There are two other existing methods that, instead of generating summarization images, show either a movie with special post-processing effects, or a shortened version of movie to reduce human effort on surveillance video analysis.

1. Dynamosaic: This technique explores the manipulation of time in video editing, which allows the users to control the chronological time of events. These

time manipulations include slowing down (or postponing) some dynamic events while speeding up (or advancing) others. Time manipulations are obtained by first constructing an aligned space-time volume from the input video, and then sweeping a continuous 2D slice (time front) through that volume, generating a new sequence of images for dynamic scenes. To avoid artifacts, the problem of finding optimal time front geometry was formulated as one of finding a minimal cut in a 4D graph, and solve it using max-flow methods [25].

2. Video Synopsis: Video synopsis is an effective tool for browsing and indexing of surveillance videos. It provides a short video representation, while preserving the essential activities of the original video. The activity is condensed into a very short period video by simultaneously showing multiple activities, even when they originally occurred at different times. The synopsis video is also an index into the original video by pointing to the original time of each activity. Video synopsis can be applied to create a synopsis of an endless video streams, as generated by webcams and by surveillance cameras. However, viewing such a synopsis may seem awkward to the non-experienced viewer [2].

3 CAMERA MOTIONS AND CORRESPONDING FLOWS

3.1 Camera Ego-motions

How could a 3D video volume be transferred to a 2D image profile while making it inclusive and representative? The consecutive frames of a video have large overlaps of scenes. Therefore, reducing redundant pixels in a clip becomes possible in video summarization. A videoed space contains some static background and dynamic foreground. With various camera operations including static camera, the scene space is projected to the *video space*. If the camera motion can be extracted, the intrinsic scene space can be recovered in the image profile without the pixel redundancy.

In video databases, most of videos have intentional camera operations rather than random waving. For such a smooth camera operation, we can use a global motion appearing in the video for efficient video profiling. Assuming static background patterns B_i, $i = 1, 2, \ldots$ and dynamic foreground patterns $F_j(t)$, $j = 1, 2, \ldots$ are in the space. They are interchangeable depending on which one dominating the *field of view*. A camera can be static or undergo ego-motions such as zoom f, rotation R, translation T, and their combinations (Fig.3.1). Through the camera ego-motion $K_{R,T,f}$, a scene has relative 3D motion $V = R \times (B_i, F_j) + T$ with respect to the camera where (B_i, F_j) is its location in the camera coordinate system. A composite camera motion with R, T, and f can be categorized as its high level operations such as pan/tilt, rail/vehicle motion, orbiting (focusing and moving around object with simultaneous translation and rotation), crane motion (orbit motion plus free camera direction and zooming), forward moving or zooming, and so on. On the other hand, dynamic scenes in a static field of view may further reveal a variety of motions themselves that can be classified as directional motion (e.g., marathon crowds) or diversified motion (e.g., random

walking people in a shopping mall). In general, we can describe the camera kinematics in Fig.3.2, which yields typical camera works (dotted boxes). The camera works generate distinct optical flows that can be classified as diversified flow or directional flow in the field of view. This categorization helps us design a general cutting strategy to obtain the profile of video clips. Video clips with typical camera motions and the composite camera motions were examined in previous published papers [26, 27].

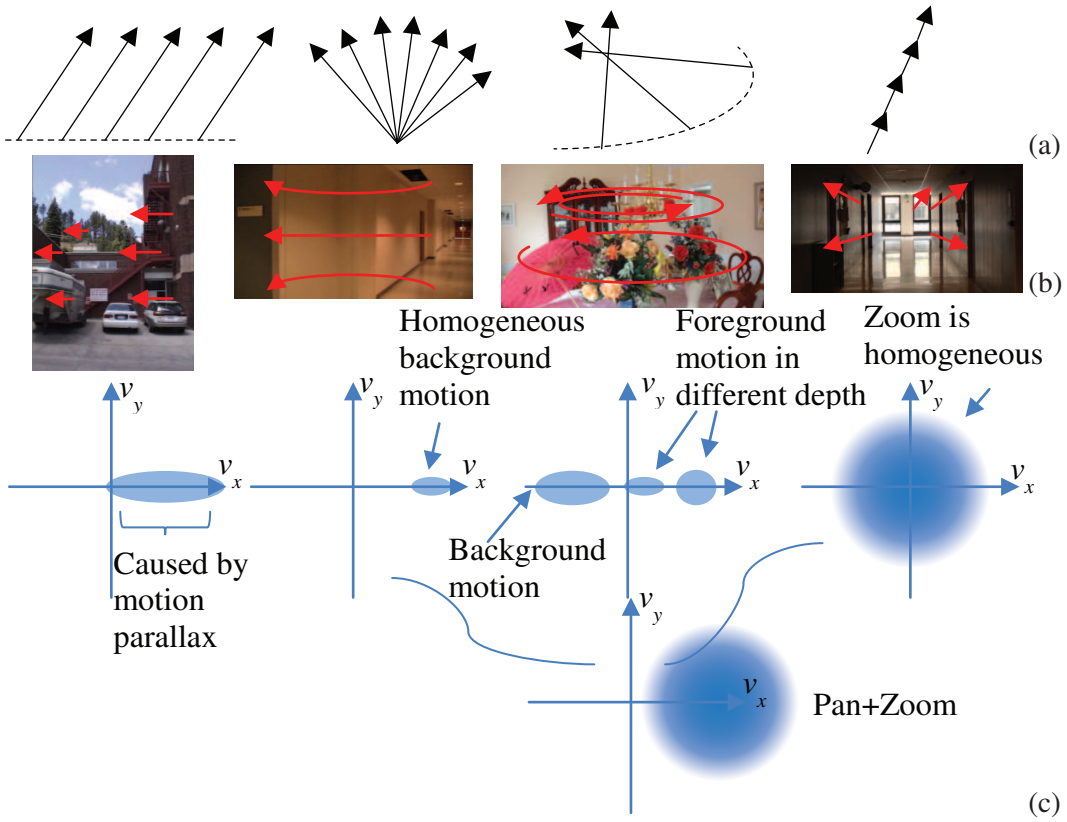


Figure 3.1.: Typical camera operations translation, pan, around-object, and zoom, from left to right in each column. The arrows in (a) indicate the movement of camera axis for translation, pan/tilt, around object rotation and zoom in/out. The optical flow directions are also indicated in the field of view in (b). The distributions of motion vectors projected to a video frame are also illustrated briefly in (c). (c) also shows the distribution of composite pan + zoom camera operation which shows the additive property of the optical flow from different motions.

3.2 Major Flow and Minor Flow

Now, let us examine some common properties of the categorized motion. The video is obtained as $I(x, y, t) = K_{R,T,f}(B_i, F_j)$ and the optical flow in the video volume is denoted as $u(x, y, t) = (u_x, u_y, u_t)$, indicating the motion component of a feature in the frame during time u_t (one or more frames observed). The image flow is normally affected by 1) the intentional camera direction, 2) unintentional shakings, and 3) unpredictable movement of target. For a video clip with flow generated from a smooth camera ego-motion or a directional movement of target crowds (usually lasting for 0.5 or more seconds in a video database), we can specify an global flow vector $V \in R^3$ in the spatial-temporal video volume, as the overall evaluation of distinct optical flow (Fig.3.3). Denoting a video clip or shot by C, and the optical flow vector

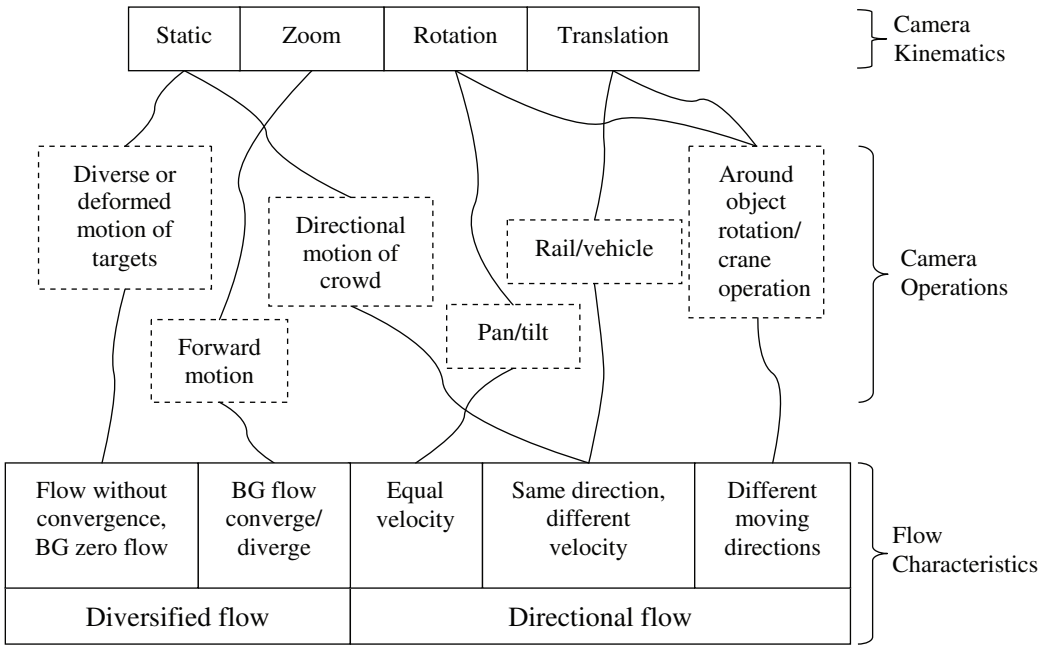


Figure 3.2.: Categorizing camera motions, typical camera works (operation) and optical flow styles in three levels.

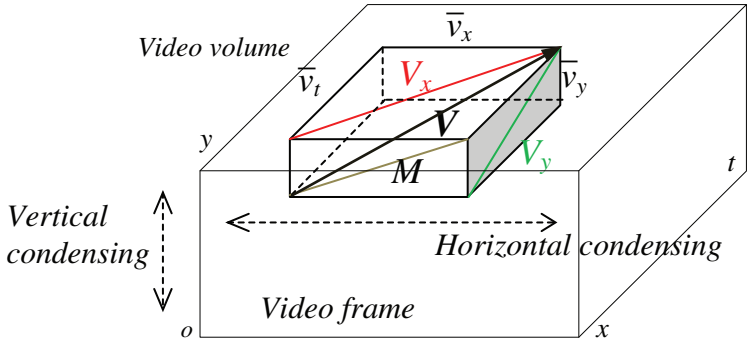


Figure 3.3.: Global flow vector $V(\bar{v}_x, \bar{v}_y, \bar{v}_t)$ and its projections in condensed images in x and y directions. V_x and V_y are projections of V to $x-t$ and $y-t$ plane.

at each point by $u(x, y, t)$ in the video clip, the global flow in the corresponding C is defined as

$$V = (\bar{v}_x, \bar{v}_y, \bar{v}_t) = \frac{1}{N} \sum_{x,y,t \in C} u(x, y, t) \tag{3.1}$$

where $||u(x, y, t)|| = 1$, and N is the number of high contrast points in the clip. Its projections along x, y, and t directions are $V_x = (\bar{v}_x, \bar{v}_t)$, $V_y = (\bar{v}_y, \bar{v}_t)$, and $M = (\bar{v}_x, \bar{v}_y)$, respectively, as depicted in Fig.3.4. V shows the direction as well as the speed of scene shift in consecutive frames, if the scenes have some common motion or the camera motion is smooth. In the implementation, it's not necessary to compute the detailed optical flow vectors, because of the computational costs and errors in noisy video or videos lack of features and textures.

Depending on the impact of the flow on the video, between global flow projections V_x and V_y, the one with the major impact is treated as a *major flow*, the other one is considered as *minor flow*. The minor flow is sometimes from the camera motion caused by hand shaking, unstable walking, and vehicle waving during video capturing, or from intentional camera motion that has less impact. Its effect is visible in our profile as tilt changes during pan, translation, and zoom. Shaking in minor flow can be kept in the profile to reflect the dynamics of the camera, or can be removed by

video deshaking before or after profiling [25]. The minor motion will not be used for determining slice cutting.

For example, in Fig.3.4, the viewer could understand that the major flow in camera pan is horizontal and the minor flow is vertical. In a zooming shot, the major flow is relatively small but the variance of motion vectors is large. For a static camera shooting deformable action of persons or a random crowd in a place, the variance of motion vectors is small. For other camera actions, a major flow can be estimated from directional motion vectors in the frames.

3.3 Motion Estimation in Condensed Images

We use condensed images to perform the task of automatic profiling, instead of computing optical flow $u(x, y, t)$ explicitly and then summarizing the motion vectors for global flow direction (so far uses *Principle Component Analysis* [28] on the optical flow [29,30]). The reason is because the cost of flow computing for large video database is high and the results are unstable for scenes with deformation, water, fire, etc. and

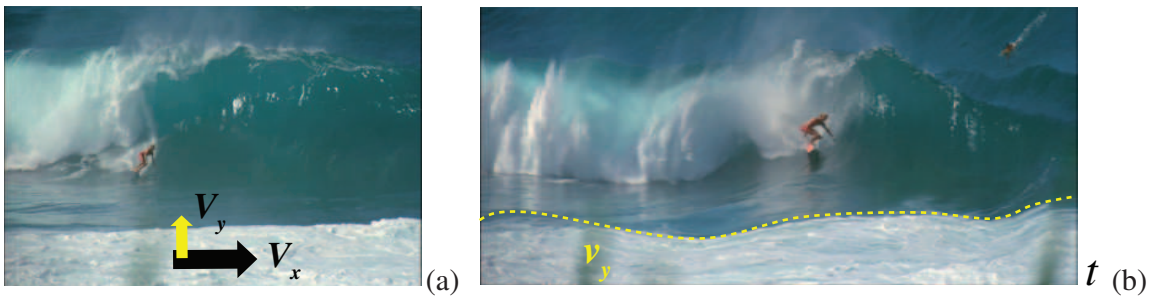


Figure 3.4.: An example of global flow projections in right camera pan as if we are looking at the spatial-temporal video volume from front. (a) This example shows a real right panning video with a directional major flow V_x facing right and disturbance minor flow V_y. (b) The generated video profile with a vertical sampling pixel line from left to right. The effect of disturbing minor flow $V_y = (\bar{v}_y, \bar{v}_t)$ on the video profile over time is marked as a dashed yellow trace.

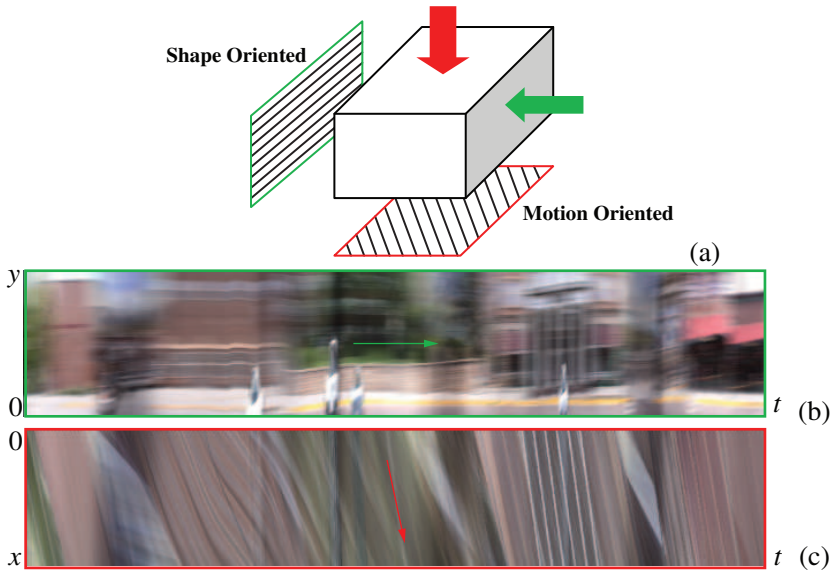


Figure 3.5.: Two condensed images from a horizontally translating camera. (a) Condensing a video volume to images in x (green arrow) and y (red arrow) directions. (b) Shape-oriented condensed image with stationary blurred shapes, (c) Motion-oriented flow graph with many motion traces in it. The time axes are both horizontal. The traces and their global directions are shown as green and red arrows.

scenes without many features. Alternatively, this work uses video condensing to images to perform the task. A *flow graph* has been proposed as one of the condensed images to show reliable motions as major flow in traces across frames, and it also achieves efficiency in obtaining the global motion. Another one, called shape-oriented condensed image, embeds the shape distortions introduced by camera shaking which will be used in Section 7.3 for deshaking of generated video profile.

Two condensed images as in vehicle video sequences [31] are employed here. For simplicity, the condensed images along the x and y directions in the frame have been collected (Fig.3.5), as

$$C_y(t,x) = \frac{1}{h}\sum_{y\in C} I(x,y,t) \quad C_x(t,y) = \frac{1}{w}\sum_{x\in C} I(x,y,t) \tag{3.2}$$

where w and h are the frame width and height. Long or high-contrast features aligning with the condensing direction shows their distinct traces in the resulting image, while those features in other directions are blurred out. If a clip is from a static camera, i.e., $|\bar{v}_x| \approx |\bar{v}_y| \approx 0$, both condensed images only contain traces aligning with the time axis. However, if $\bar{v}_x$ or $\bar{v}_y$ has a relatively large length, either condensed image will show motion traces non-parallel to the time axis. Figure 3.5b,c are two condensed images from a video captured by a translating camera. Condensed in y direction in the frame, the traces in Fig.3.5c show the flow direction as traces, while condensed in x direction, Fig.3.5b poses the stationary blur [32] on features other than x direction but keeps features in x direction sharp. The waves on the condensed features in x direction show the shaking in minor flow (in this example, V_y) in the clip. It can be found that, if V is parallel to neither x nor y axis (V slanted), both condensed images contain motion and shape information such as traces, features with stationary blur, and waved linear features. Depending on the dominant information that $C_y(t, x)$ and $C_x(t, y)$ contain, this work refers to one as *motion-oriented image* (flow graph) and the other as *shape-oriented image*. The motion-oriented one displays more motion traces of features in the video, while the shape-orientated one shows more blurred shape than traces. Determining which condensed image is motion-oriented will allow us to select the horizontal or vertical cutting line in the clip.

3.4 Stationary Blur in Shape Oriented Condensed Image

Here we briefly describe a phenomenon of the *parallel projection* [18, 33] for the video profile in the shape oriented condensed image. In the parallel projection image, there exists a special blur effect along the time axis, named *stationary blur* [34, 35] in contrast to the *motion blur* in the spatial image. Ideal *Plane of Sight* [18] are infinitely thin and infinitesimally dense for a video profile along the camera movement. Nevertheless, this can only be approximated by a high resolution line sensor with a fast sampling rate. A video camera, however, has pixel lines with a certain physical width.

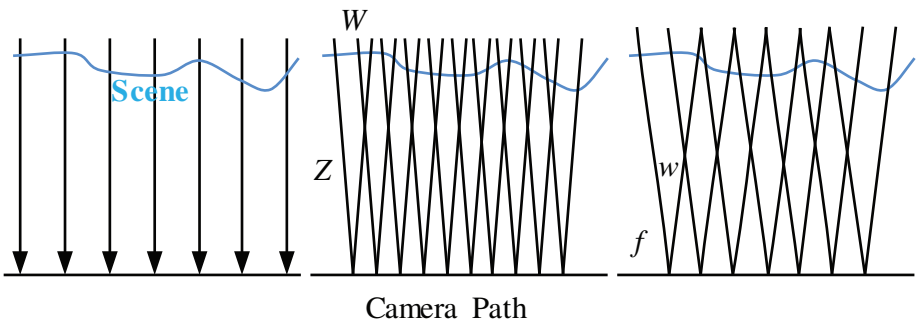


Figure 3.6.: Sampling scenes on a camera path (top view) under an ideal parallel projection and narrow perspective projections respectively in obtaining a video profile.

The ray through a pixel is in a shape of cone realizing narrow perspective projection at each position on the path (Fig.3.6) or similarly at each direction of camera pan. The distant scenes are averaged as they are projected towards the profile through the cone. As the camera shifts horizontally, consecutive cones overlap partially beyond a certain far range. This causes the temporal burring over consecutive pixel columns in the profile. Imagine we have a three dimensional camera coordinate system whose origin is at the center of projection and whose Z axis is along the optical axis as shown in Fig.3.7. Let us model the stationary blur optically for our extension of it in the next section. Denote the color distribution of a scene (B_i, F_j), $i, j = 1, 2, \ldots$. If w is the slit width ($w = 1$ pixel for the video profile), the sampling cone through the slit has a width of $W = Zw/f$ at a surface point $P(X, Y, Z)$. The sampling cone averages the colors at the surface with rectangular function

$$\text{Cone}(X, W) = \begin{cases} 1/W & |X| < W/2 \\ 0 & |X| > W/2 \end{cases} \tag{3.3}$$

Hence, the video profile $P(t, y)$ is obtained formally from

$$P(t, y) = (B_i, F_j) \oplus \text{Cone}(X, W) \tag{3.4}$$

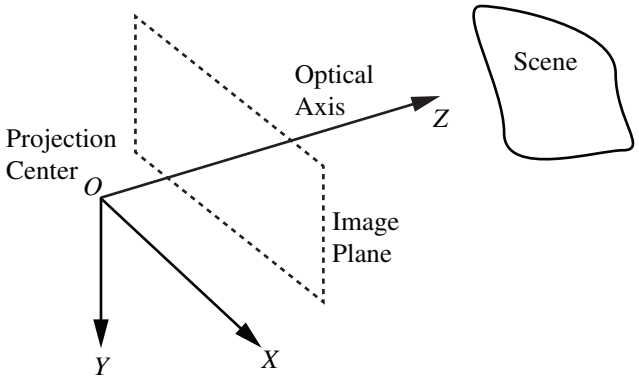


Figure 3.7.: Camera coordinate system

In shape oriented condensed image, the intensities are averaged horizontally in each frame to generate 1D intensity profiles. This enhances such a blur effect by enlarging w to multiple pixels for accumulating color spatially in each frame. We found that both distant and horizontal features in the 3D space appear as long traces in the shape oriented condensed image. This can be used as reliable evidence for the shaking detection and removal, which will be introduced in Section 7.3.

4 SEGMENTATION FOR SIMPLE CAMERA OPERATIONS

Video clips are segmented using traditional histogram differentiation [36]. For the better analysis of the flow characteristic and the generating of temporal video profile, a clip was further separated to sections, each with a monotonic camera operation/motion. In the condensed images, the flow characteristics was explored by examining extractable traces.

By detecting temporal discontinuity in $C_y(t,x)$ and $C_x(t,y)$ using their temporal histograms $\sum_x C_y(t,x)$ and $\sum_y C_x(t,y)$, video clips with continuous camera motions are successfully segmented. Further, the discontinuity was found in the flow direction so that sections with homogenous camera motions are obtained. This is particularly important for the profiling (even necessary for mosaicing if a clip is long). Its partial derivatives could be computed in a condensed image as

$$\Delta_t(C_y(t,x)) = \partial C_y(t,x)/\partial t \quad \Delta_x(C_y(t,x)) = \partial C_y(t,x)/\partial x \tag{4.1}$$

with a differential operator. The traces are selected at the peak points of the gradient $\mathrm{grad}(C_y(t,x))$, and their directions, denoted as unit vectors $g = (g_t, g_x)$, are extracted from $\Delta_t C_y$ and $\Delta_x C_y$ as

$$g(t,x) = (gt(t,x), gx(t,x)) = \frac{(-\Delta_x(C_y(t,x)), \Delta_t(C_y(t,x)))}{\mathrm{grad}(C_y(t,x))} \tag{4.2}$$

Further, it's forced that $g(t,x) = -g(t,x)$, if $g_t(t,x) < 0$, because a motion vector either from a positive or negative edge should always be along the t axis in C_y, i.e., $g_t > 0$.

The average trace direction $v(t)$ ($v_x(t)$ or $v_y(t)$ in Fig.5.1) was estimated at each time t where $v(t) = (\sum_x g_x(t,x))/q$, q is the number of trace points at time t. Obvious changes were found in its sign to segment a clip to sections with smooth camera motions. The same processing could also be applied to $C_x(t,y)$ as well to find the possible separation of clips according to the change of camera motion in the y direction.

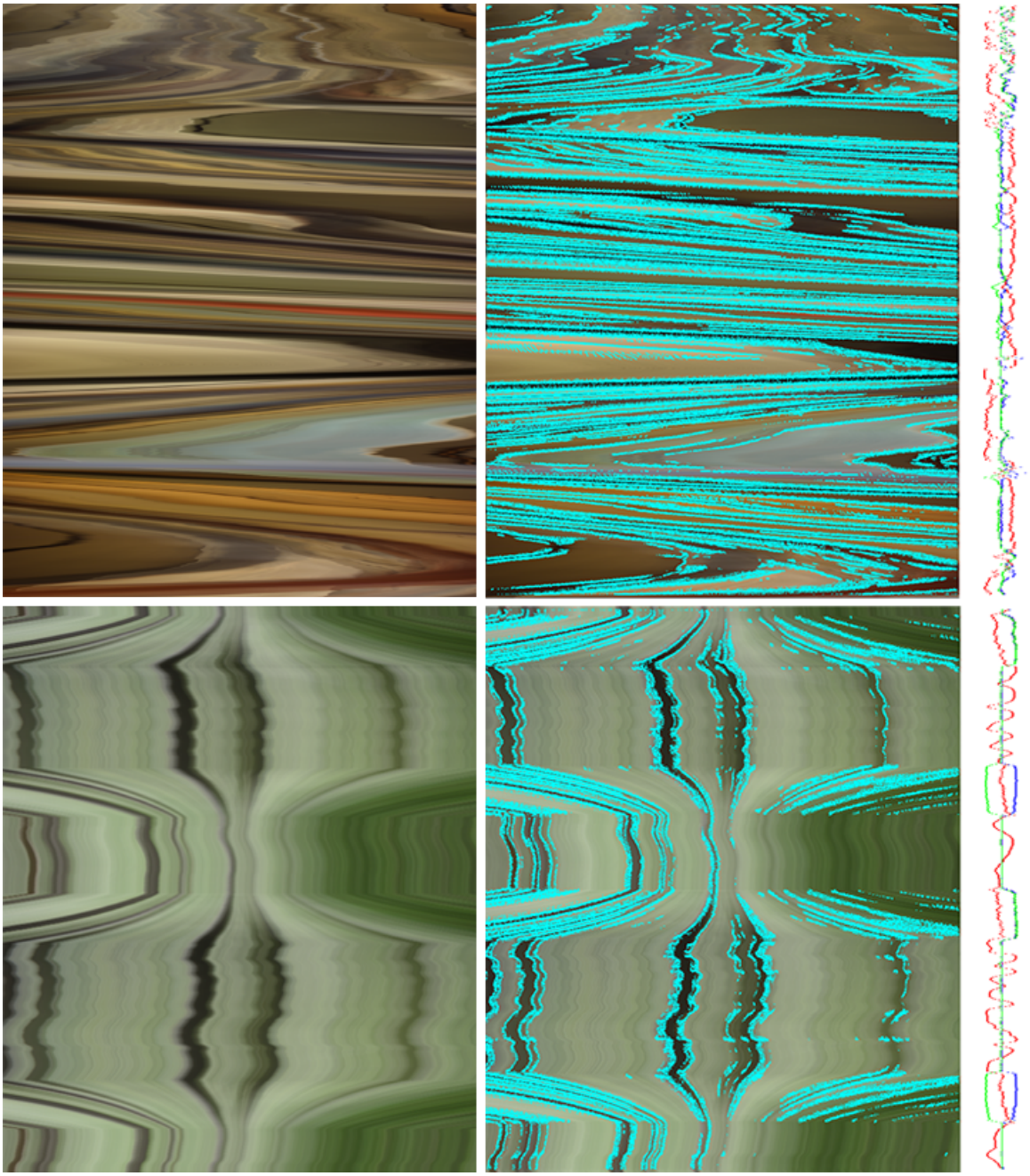


Figure 4.1.: Computing major flow and variance in a condensed image of two clips for segmenting sections as pans, static, and zooms. High gradient positions on traces are marked in cyan for motion estimation. The value of $v(t)$ ($v_x(t)$ or $v_y(t)$), $\sigma_v(t)$ and $\kappa(t)$ (convergence factor explained later) are displayed in red, blue, and green curves respectively. The time axis is vertical and downward.

Further, the variance $\sigma_v(t)$ of $g(t,x)$ is computed over time. As shown in Fig.4.1, a larger $\sigma_v(t)$ suggests a static camera or zooming. A distinct $g(t,x)$ indicating a directional motion always has a small variation $\sigma_v(t)$, which allows segmentation of sections. This work used this information to separate the diversified motion section from directional motion section. Same operation is for the other condensed image. With the sequence of $v(t)$ and $\sigma_v(t)$, a video clip can be segmented to sections with the rules listed in Table 4.1.

A median filter is further used to merge the short sections into large ones in order to remove noises in motion and obtain distinct camera movements with clear intentions. The variations in such a section are then the camera shaking to be removed later. A result of segmentation of smooth motion sections is shown in Fig.1.3 in blue bars. A convergence factor $\kappa(t)$, which will be defined later, is similar to $\sigma_v(t)$ but indicates convergence or divergence of the flow. The result of $\kappa(t)$ is also shown in green curve in Fig.4.1.

Table 4.1: Determine the camera operation based on trace direction and direction variance.

	small $\sigma_v(t)$	*large* $\sigma_v(t)$
small $v(t)$	Static camera and scene	Zooming
large $v(t)$	Camera pan	Pan+zoom, translation

5 GLOBAL FLOW COMPUTATION

5.1 Extracting Major Flow for Profiling

The relationship of global flow V and its projections are illustrated in Fig.5.1. To compute the major flow in a section with smooth camera operation, this work proposes a straightforward yet reliable method in the condensed images. In the condensed image, motion vectors at strong traces vote for a global flow vector $G = (\eta_t, \eta_x)$ as the estimate of V_x, i.e.,

$$G = \sum g(t,x)/n \quad \text{or} \quad (\eta_t, \eta_x) = (\sum g_t(t,x)/n, \sum g_x(t,x)/n) \tag{5.1}$$

where n is the number of accumulated high contrast trace points in the section.

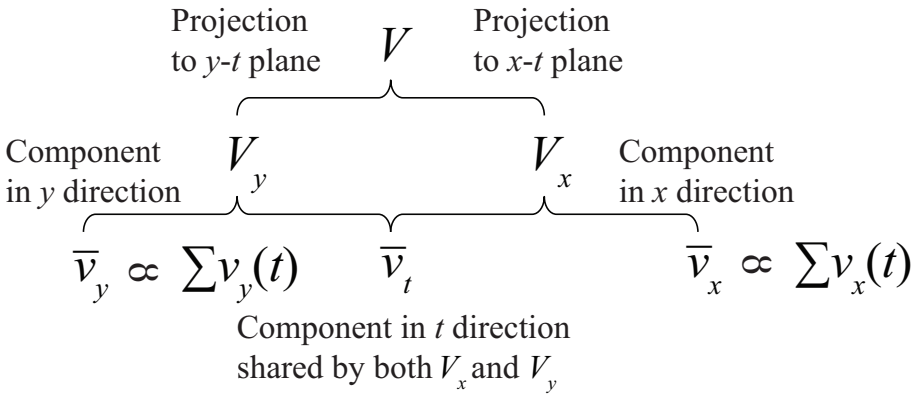


Figure 5.1.: The relationship among V, V_x, V_y, $\bar{v}_x$, $\bar{v}_y$, and $\bar{v}_t$, as well as $v_x(t)$ and $v_y(t)$. Note that $v_x(t)$ and $v_y(t)$ are the directions of traces in each time instance vary in t direction.

According to (3.1), V_x and V_y can be obtained by accumulating flow components $u(x, y, t) = (u_x, u_y, u_t)$ as

$$\begin{aligned} V_x &= \tfrac{1}{n}\textstyle\sum_{y\in C}(u_t(x,y,t), u_x(x,y,t)) \quad \text{motion/shape oriented} \\ V_y &= \tfrac{1}{n}\textstyle\sum_{x\in C}(u_t(x,y,t), u_y(x,y,t)) \quad \text{shape/motion oriented} \end{aligned} \tag{5.2}$$

Although $G = (\eta_t, \eta_x)$ is collected from a subset of points in the clip, they have the consistent motion as those points for V_x that are blurred and ignored after condensing, as the camera operation caused motion patterns in the video are not irrelevant random motion [26,27]. Thus, vectors G and V_x have the similar direction but different scales. In addition, our process does not use the traces orthogonal to the time axis so that the non-physical movements such as instantaneous events such as lighting changes, explosion, and some special effects are excluded in the major flow computation.

In the same way, the above computation is applied to the other condensed image $C_x(t, y)$ so that a global vector $H = (\rho_t, \rho_y)$ can be obtained for V_y as G. Vectors G and H precisely reflect the directions of V_x and V_y. Since V_x and V_y are projections from a same V, we need to normalize V_x and V_y so that $|\bar{v}_t| = 1$, the lengths of G and H are thus normalized using η_t and ρ_t to estimate V_x and V_y

$$V_x = \bar{v}_t G/\eta_t = (\bar{v}_t, \bar{v}_t\eta_x/\eta_t), \quad V_y = \bar{v}_t H/\rho_t = (\bar{v}_t, \bar{v}_t\rho_y/\rho_t) \tag{5.3}$$

The relation of $|\bar{v}_x|$ and $|\bar{v}_y|$ determines the acute angle or obtuse angle of V to a frame edge so as to select sampling line L_x or L_y. A larger $|\bar{v}_x|(> |\bar{v}_y|)$ means a faster flow in horizontal direction, for which $C_y(t, x)$ is the motion-orientated image for slice cutting. In opposite, $C_x(t, y)$ is treated as the motion-oriented image. According to the projection of V in the motion-oriented image V_x (or V_y), the diagonal direction of cutting trajectory $x(t)$ (or $y(t)$) is determined to intersect V_x or V_y so as to include all the scenes into the profile. Figure 5.2 gives the results of this method to find the direction of major/minor flow for various videos.

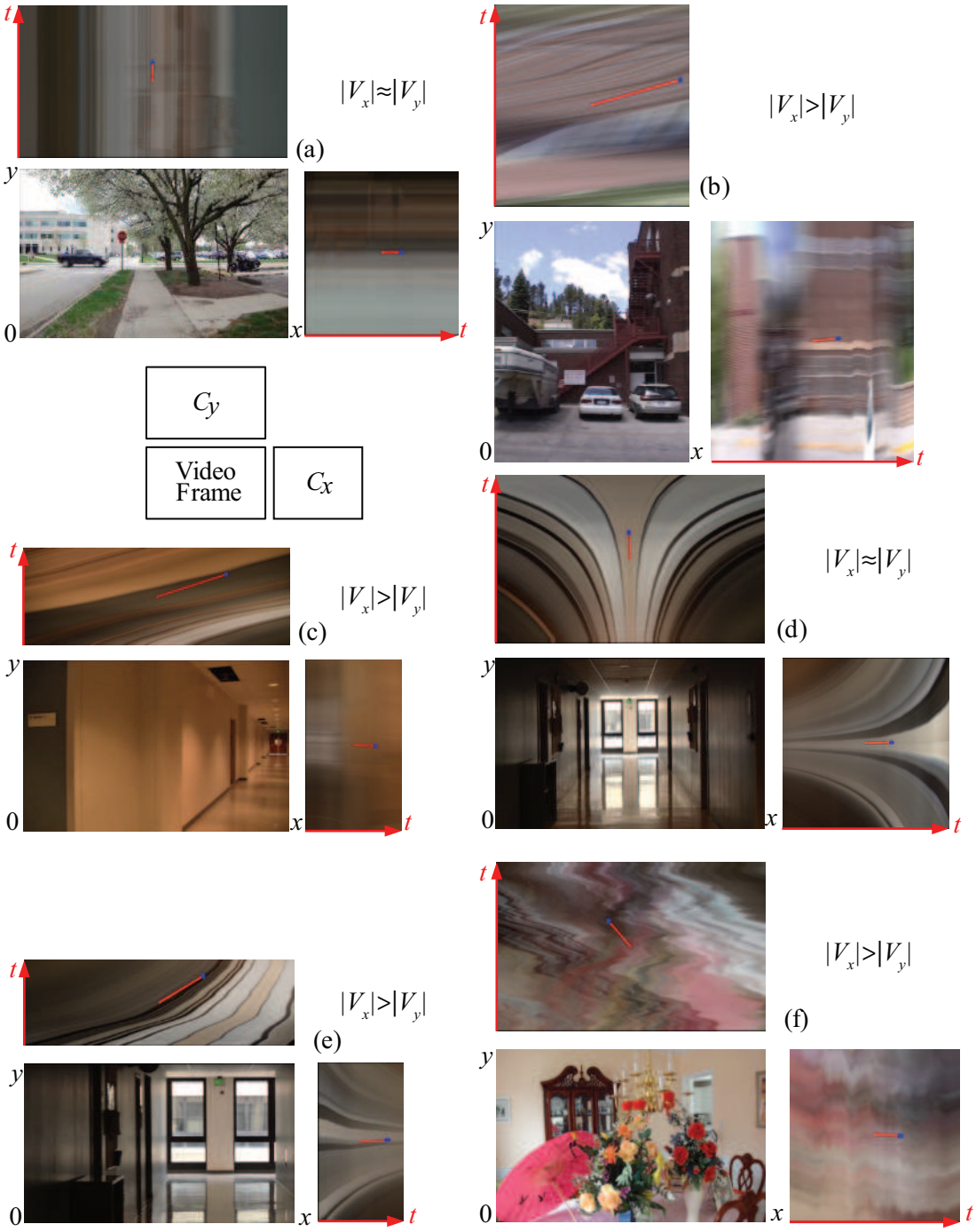
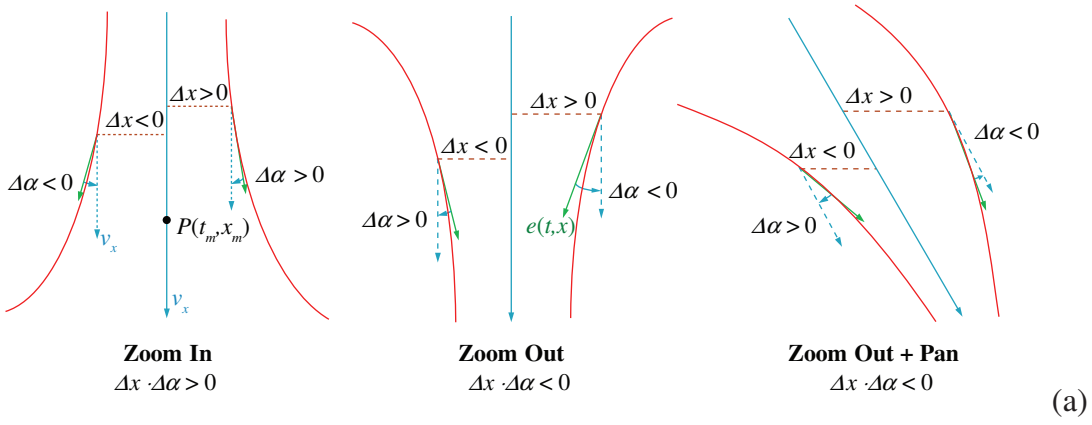


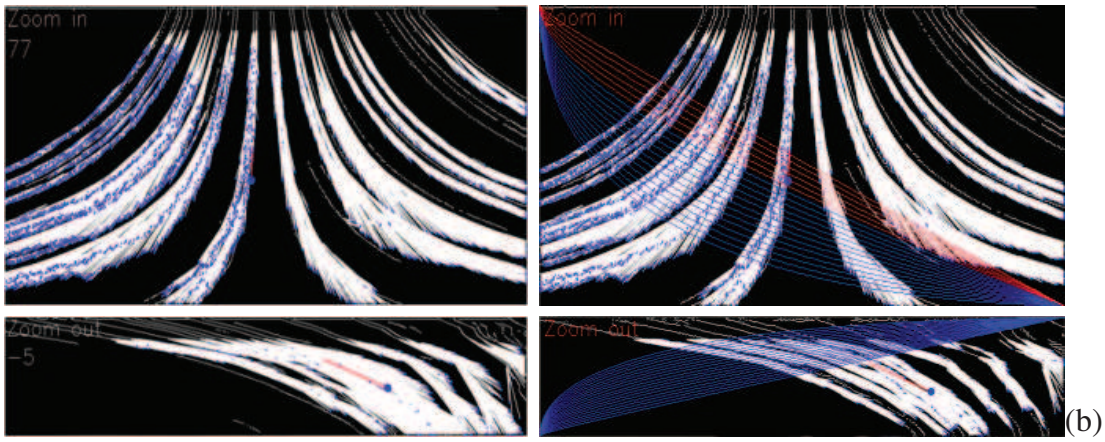
Figure 5.2.: The major flow directions of video clips detected in condensed images $C_y(t, x)$ and $C_x(t, y)$. The estimated normalized projectionsV_x and V_y are plotted in red arrows with blue tips. Note that their projections on t axis are scaled to the same length. (a) Static camera, (b) camera translation, (c) panning, (d) zoom, (e) around-object motion, (f) pan plus zoom.

5.2 Estimating Convergence Factor

In addition to the major flow direction, a convergence/divergence factor [37] which is similar as $\sigma_v(t)$, denoted by $\kappa(t)$, characterizes the zoom effect as in Fig.5.2d, e in each section. this work computes this factor only in flow graphs that is more efficient than computing optical flow and comparing their eigenvalues (unable to figure out



(a)



(b)

Figure 5.3.: (a) The computation of convergence factor from strong traces, where $\triangle x = e(t,x) - \bar{v}_x$ and $\triangle\alpha = x - x_v(t)$. (b) Experiment of convergence factor computation from strong edge points in the flow graphs in Fig.5.2d, e. and planned quadratic curve candidates for a profile (blue: without scene recurrence, red: with scene recurrence). White needles show the tangent vectors on traces and the blue dots show their tips. The vertical axis is the time. Estimated V_x (V_y) are indicated in left column with red arrows.

zooming direction even zoom itself is detectable). At a strong edge point (t, x_i) in a section of flow graph, e.g., $C_y(t, x)$, the angle of a motion vector of trace point, $e_i(t, x)$, $i = 1, 2, \ldots$, is computed from its gradient $g(t, x) = (g_t, g_x)$. A median point $p(t_m, x_m)$ is computed in $C_y(t, x)$ from the positions of all the qualified edge points with a median filter (see Fig.5.3a). Through $p(t_m, x_m)$, a reference line $x_v(t)$ along major flow V_x (Fig.3.3) divides all the edge points. Then, the zooming effect at each point is calculated as $(e(t, x) - \bar{v}_x)\text{sign}(x - x_v(t))$. It takes positive value for divergence flow (zoom-in) and negative value for convergence flow (zoom-out). For the convergence factor at time t, the convergence factor is

$$\kappa(t) = \frac{1}{n}\sum_i (e_i(t, x_i) - \bar{v}_x) \cdot \text{sign}[x_i - (x_m + \bar{v}_x(t - t_m))] = \begin{cases} < 0 & \text{converge} \\ > 0 & \text{diverge} \end{cases} \tag{5.4}$$

gives the degree of convergence/divergence in value, where n is the number of points involved in computation. $|\kappa(t)|$ reflects the degree of flow convergence/divergence or how fast the scene is zoomed out/in. Figure 4.1 shows $\kappa(t)$ over time in green curve. The convergence factor κ in the entire section is the average of $\kappa(t)$.

6 CUTTING VIDEO VOLUME FOR PROFILE

6.1 A General Cutting Strategy for Temporal Mode

The profile not only has the advantage to include more scenes for browsing as mosaicing, but also has a dimension of time for indexing to a particular frame form a clicked/selected position. To facilitate multimedia indexing and transmission, some image properties of perspective projection may be sacrificed as in panoramas. With the successfully segmented sections of video volume $I(x, y, t)$, this work performs global sampling to obtain their 2D profiles denoted as either $P(t, x)$ or $P(t, y)$, to guarantee a single occurrence of a scene in the profile except occlusion. As illustrated in Fig.6.1, the profile reveals all the scenes in the video for retrieval and display subject to certain shape deformation. Through the profile, say $P(t, y)$, a video section can be temporally indexed to a frame via t, rather than mosaicing frames into a space. Instead of composing mosaic by segmenting B_i and F_j in $I(x, y, t)$, a moving pixel line L_y or L_x is used to sample the video volume either vertically or horizontally for the image belt $P(t, y)$ or $P(t, x)$, respectively. In order to record shape of B_i and F_j in the profile, the sampled slice in the volume should cut against $v(t)$, rather than

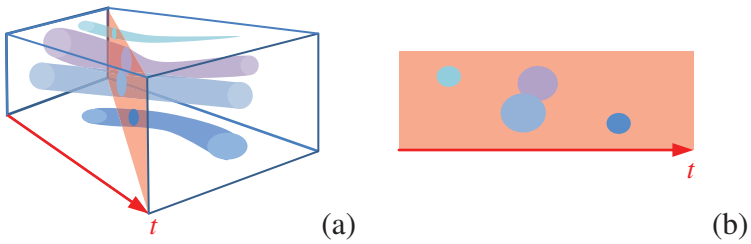


Figure 6.1.: Profiling by cutting across flow in the video volume. (a) Video volume with flow and cutting slice, (b) video profile.

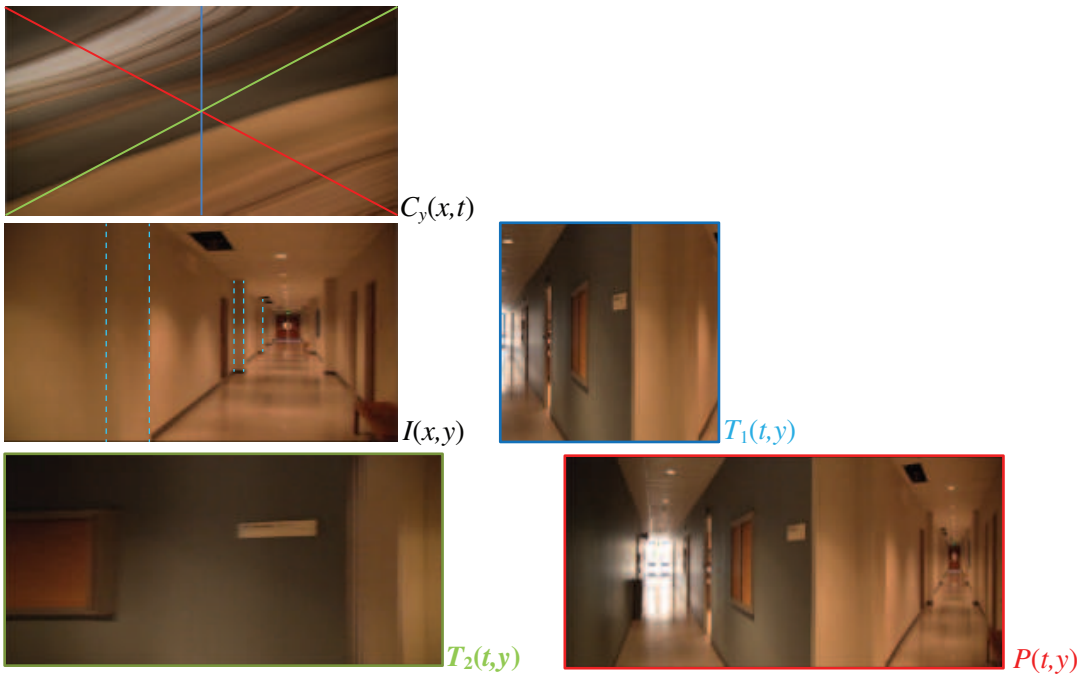


Figure 6.2.: A flow graph $C_y(x,t)$ condensed vertically from a panning video clip, and possible cuts of video. The most inclusive and sharp one is profile $P(t,y)$ cut against the flow. The diagonal cutting $T_2(t,y)$ along the motion traces makes a narrow view in a blurred image, which is not meaningful as a video summary. Key frame $I(x,y)$ and simple indexing $T_1(t,y)$ at image center are not inclusive for a scene space.

aligning with $v(t)$ that yields traces in the profile, as shown in Fig.6.2. The global cutting method is as following.

- The sampling line is parallel to an axis of image frame, mostly parallel to structure lines in the scenes (dotted blue lines in $I(x,y)$ of Fig.6.2), to keep the shape integrity, as $P(t,x)$ or $P(t,y)$ are displayed in regular window [26]. The line more orthogonal to the major motion is selected, i.e., select L_y to sample the video vertically if $|\bar{v}_x| > |\bar{v}_y|$, or select L_x otherwise. This extends line stitching [4] to both x and y directions.

- After aligning the sampling line, say L_y, it is moved along a diagonal trajectory $x(t)$ in the volume intersecting the global flow V, i.e.,

$$p(t, y) = \text{sampling}(I(x, y, t)|x(t)) \tag{6.1}$$

  The diagonal $P(t, y)$ obtains sharp scenes than cutting along V_x and, at the same time, map all the scenes stably visible in the video into $P(t, y)$. It does not cut back and forth in the volume with size-varied patches [4], because $P(t, y)$ should reflect a consistent temporal scale in temporal mode.

- If the major flow is accompanied with convergence or divergence effect due to a zooming operation, the sampling curve will be bent towards the enlarged frame in the clip so as to prevent scene blurring and recurrence in the profile. The value of convergence factor determines a curved or straight trajectory, as well as the direction of bending.

The bending is for the purposes of (i) emphasizing zoomed scenes in the profile, (ii) improving the scene distribution in the profile, (iii) avoiding recurrence of B_i in the profile, (iv) adding motion blur to F_j in the profile, and (v) possible profile animation of the profile. The bending degree depends on $\kappa(t)$ computed above.

For an easier control in the context of automatic implementation and the simplicity in form, a quadratic Bezier curve is used for $x(t)$ from one corner to the diagonal one to cut the major flow. $x(t)$ is bended towards the end frame if $\kappa > 0$, and towards the start frame if $\kappa < 0$, and linear if $\kappa \approx 0$. The bending degree increases if $|\kappa|$ is large. Figure 5.3b shows the computation of κ as well as a sequence of Bezier curves, from which a curve is selected to avoid scene recurrence and reduce the motion-blur in the generated video profile.

6.2 Cutting Clips from Simple Camera Motions

Now, let us apply the above profiling method on various video clips generated from simple and composite camera motions as in Fig.3.2 in order to validate the design of

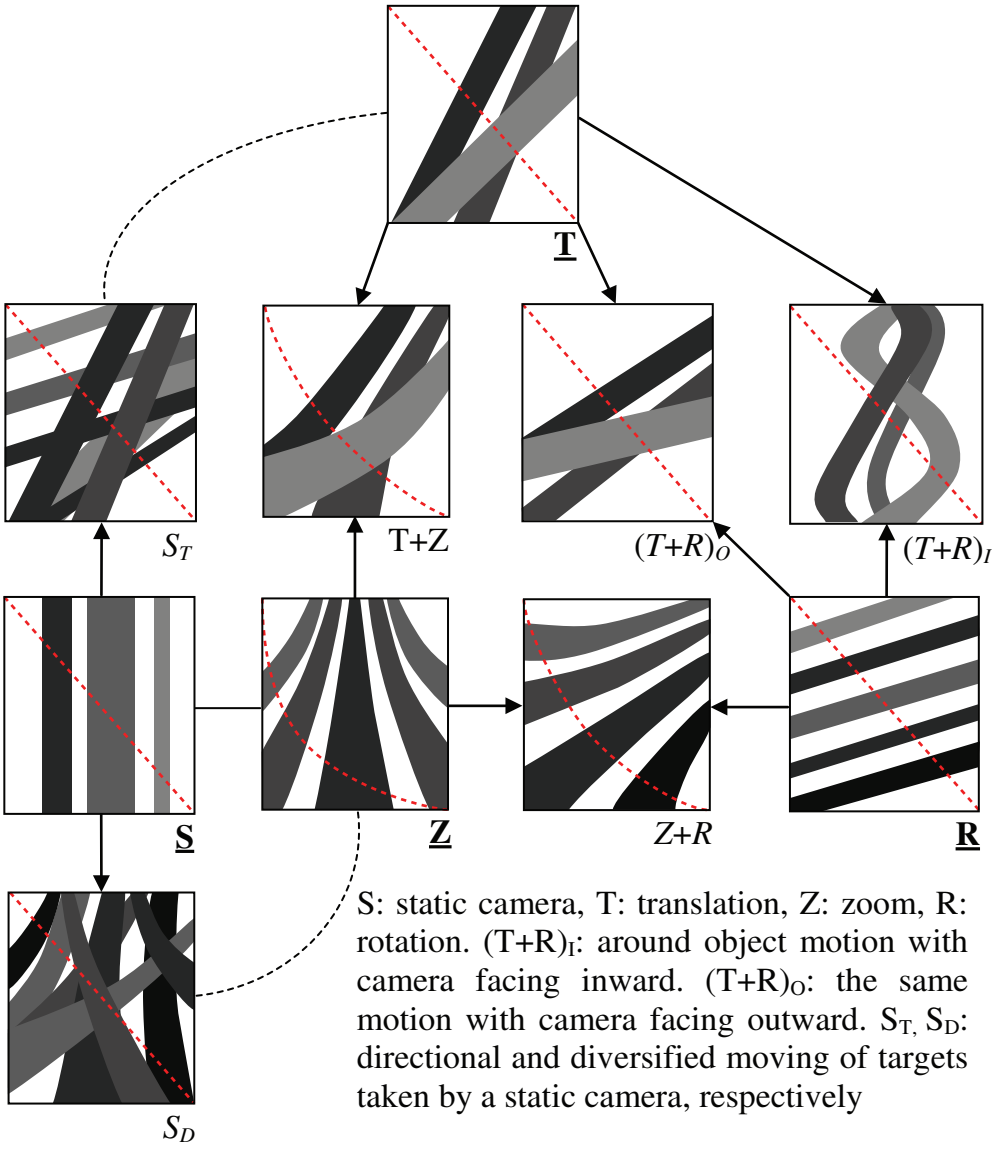
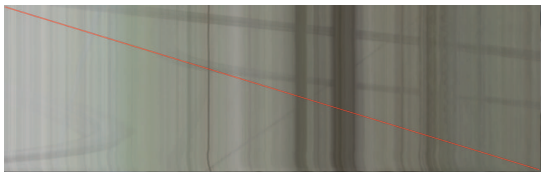


Figure 6.3.: Possible cutting trajectory (dashed red lines) for major motion traces (grey belts) in the condensed images. The time axes are downward vertically. The camera motion is rightward if it is not static and zooming.



(a) Flow graph and cutting trajectory. The time axis is downward.

(b) End frame of the clip $I(x,y)$

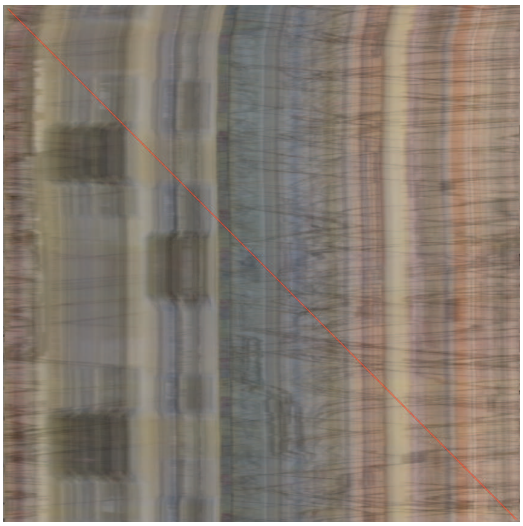
(c) Profile $P(t,y)$

Figure 6.4.: Video took beside a street. In the profile, the background stays the same as that in each frame.

our algorithm. After computing the major flow direction and aligning the sampling line, we examine the motion traces in the motion-oriented condensed image for slice cutting. Fig.6.3 illustrates the flow characteristics from all types of camera motions, assuming the major flow is horizontal, i.e., the camera motions are horizontal, for the simplicity in explanation. Simple camera motions such as zoom, rotation, and translation are abbreviated as Z, R, and T in bold font, and their possible combinations are put in between Z, R, and T. The motion traces are depicted and the diagonal slice cuttings are indicated by $x(t)$ in dashed red lines. For a static camera, a diversified flow S_D is similar to case Z, while a directional flow S_T is inherently similar to case T, if the foreground flow is dominant.

6.2.1 Profiles from Static Camera

For a static camera shooting mild motion such as a talk show, the direction of major flow vector mainly from the static background is almost parallel to the time axis, i.e., $|\bar{v}_x| \approx |\bar{v}_y| \approx 0$. We cut a vertical slice (consistent to gravity) across the



(a) Flow graph and cutting trajectory. The time axis is downward.

(b) End frame of the clip $I(x,y)$

(c) Profile $P(t,y)$

Figure 6.5.: Video took in a shopping mall. Shoppers and camera shaking can be observed in the profile.

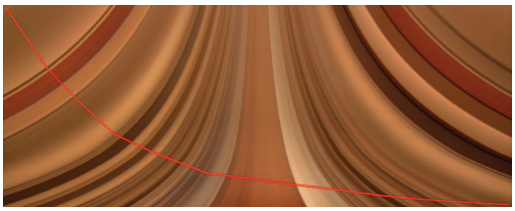
Figure 6.6.: Profile obtained from a surveillance camera capturing video with an infinite length.

video volume diagonally (Figs.6.4, 6.5). A diagonal slice in the volume is longer than the frame width; the profile image, $P(t, y)$, has a better resolution than key frame when it is scaled up along the timeline. If the camera shoots directional flow, e.g., a surveillance camera monitors people and vehicles through a path (Fig.6.6), a sampling line parallel to the dominant structure lines in the scenes is set to cut the flow for the profile as in [19]. The profile shows the shapes and time of arrival of passing targets.

If a camera is shooting diversified flow, we can set multiple sampling lines at pathways where major flows occur. With this profile, a surveillance video lasting for many hours can be briefly browsed for locating a time for a person in the video or to count the total number of passages.

6.2.2 Zoom In/Out

Extending from a static camera, camera zoom yields a flow expanding from a *Focus of Expansion*. Considering the gravity direction projected in the frame, we can



(a) Flow graph and cutting trajectory. The time axis is downward.

(b) End frame of the clip $I(x,y)$

(c) Profile $P(t,y)$

Figure 6.7.: Video took in a hallway. In the profile, the zooming effect can be found, the whole scene get trapezoidal from left to right.

specify the major flow direction as horizontal and set a vertical line to sweep the video volume from either left or right. The captured scenes thus will have distortion in the profile as in Fig.6.7. The cutting curve $x(t)$ is bended in such a way to preserve the resolution of the enlarged portion in the zooming. The scaled shape along time axis in the profile indicates the zooming up action in the video clip. If the scene is zoomed out, we can obtain a time-flipped curve in the condensed image for cutting.

Along a planned Bezier curve, the tangent calculated from the formula is compared with the tangent of motion trace at each crossing point to ensure that it's larger angle than the trace angle (Fig.5.3b).

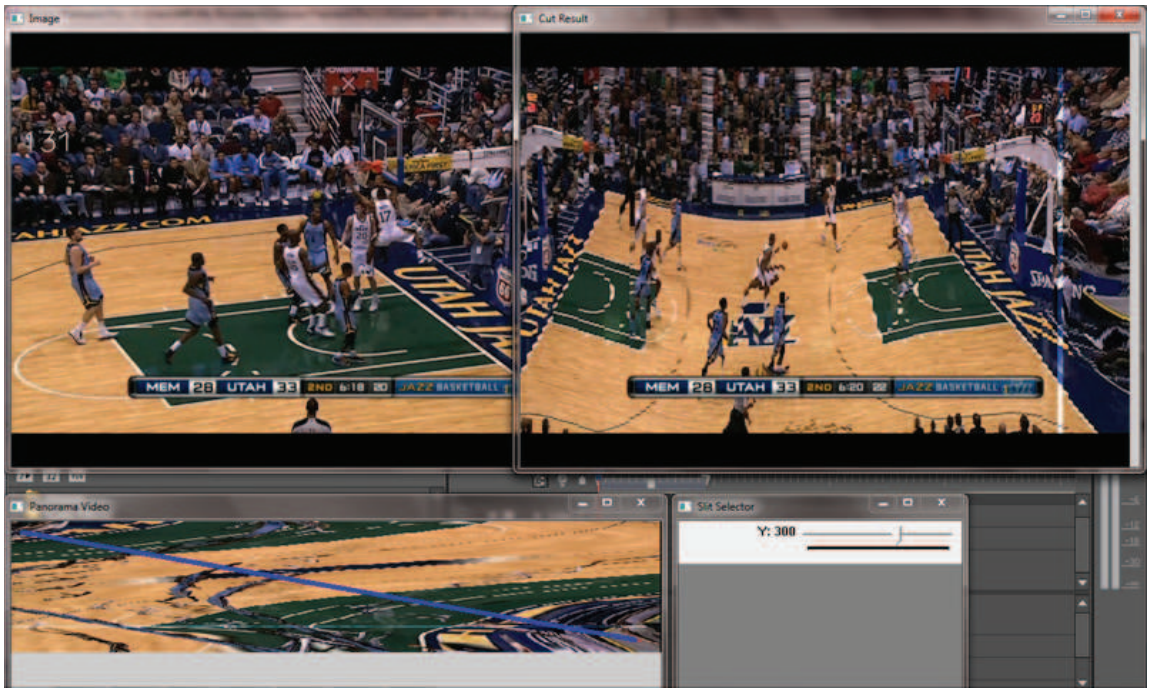


Figure 6.8.: Screen shot of the software processing a video from a panning camera from left to right and the generated profile. (top left) End frame, (bottom) a flow graph, (top right) profile.

6.2.3 Pan/tilt Clip

Panning appears most frequently in video to increase the field of view and track a target. The major flow traces are homogeneous (parallel) as in Fig.6.3R. The generated profile even works on deformable scenes as in Fig.6.8 where matching based spatial mosaicing is incompetent. Although the generated profile is bumpy, it reflects the minor flow caused by tilting and can be rectified through deshaking. Similarly, tilting clips can be processed in a symmetric way for a profile $P(t, x)$.

6.2.4 Translating Camera

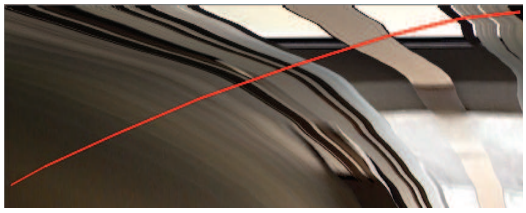
The camera translation is always visible in movie shots captured by vehicle/rail sets [18,33]. Such shots can also be captured from planes, ships, cars, etc. The camera translation in a sideway direction creates a parallel flow field in the field of view with non-homogeneous motion parallax due to varied depths in the scene (Fig.3.5c and Fig.6.3T). A vertical line can scan the major flow diagonally in the clip to form a profile as in Fig.1.1. If the camera is translating not purely sideways, it creates a flow field expending from focus of expansion, which has both effects of translation and zoom as analyzed in [33]. A vertical line cuts the video frame and generates a forward aspect view in the profile. The profile obeys a *parallel-perspective projection* that is different from a perspective projection (Fig.6.9).

Figure 6.9.: Profile of a vehicle-borne video while the camera is translating on a path as a smooth curve.

6.3 Profiling Videos with Composite Camera Motions

6.3.1 Cutting Clips of Composite Camera Motion for Profile

A camera can zoom during translation (T+Z) (Fig.6.3). A translating camera with its optical axis along the path is a case of T+Z. On a circular path usually obtained with a crane arm or on the rail, the camera can face inward as $(T+R)_I$ to focus on a target, where the background generates the major flow. If the camera faces outward, i.e., $(T+R)_O$, it generates flow more similar to translation with motion parallax. It is well known that the flow from a composite camera motion is the combination of the flows from simple motions, according to the additive property of the optical flow from different motions. As shown in Fig.6.3, most of the composite flows have a consistent direction in the condensed image. Even if the image velocities (trace orientations) vary, the flows are mostly inverted to the camera moving direction. According to our algorithm, the designed slice cutting for a composite camera motion is a plane or curved surface across the flows, as indicated in red lines in the figures. A slice



(a) Flow graph and cutting trajectory. The time axis is downward.

(b) End frame of the clip $I(x,y)$

(c) Profile $P(t,y)$ showing zoom out

Figure 6.10.: Pan plus zoom out and its profile.

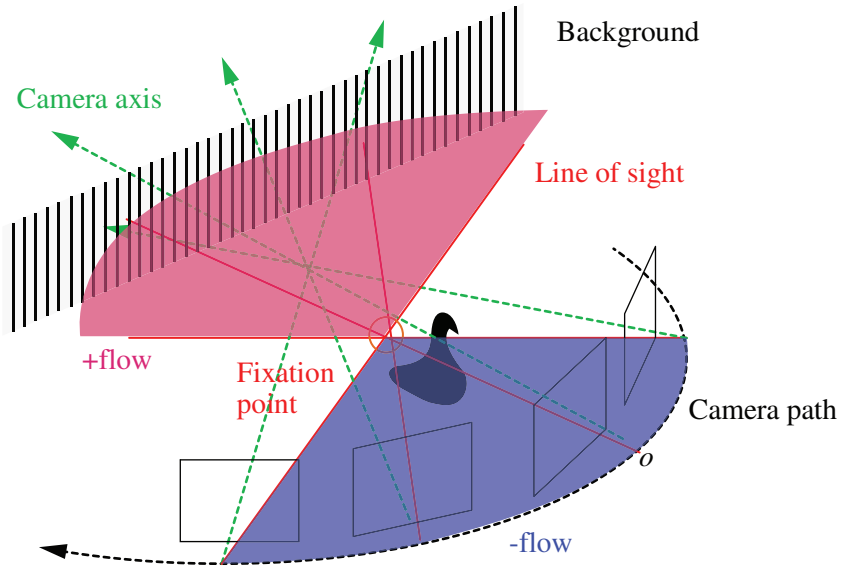


Figure 6.11.: Camera path in around object motion and rays focusing on a target. Scenes at different ranges show different flow directions (as +flow and –flow).

passes all the traces once to include scenes that stably appearing in the video clip. As an example, Fig.6.10 is a profile from a clip captured from camera panning while zooming.

Orbiting (around object) camera is also a camera work frequently adopted in shooting static objects such as a sculpture in museum, a performer on stage, etc. showing their various aspects. The camera usually focuses on a target during its motion along a circular path. The motion has simultaneous translation and rotation (Figs.6.12, 6.13).

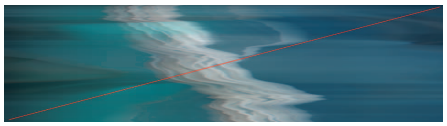
In an orbiting video, the path center has zero optical flow. For the space beyond the path center, its projected flow in video is in the same direction as the camera moving direction. However, the space closer than the center is projected as flow opposite to the camera moving direction (Fig.6.11). The condensed image shows twisted flow traces of the foreground target. Because background is usually larger than the foreground target in the field of view and it determines the major flow, our algorithm cuts slice across background to show the entire space as depicted in Fig.6.3.

(a) Flow graph and cutting trajectory. The time axis is downward.

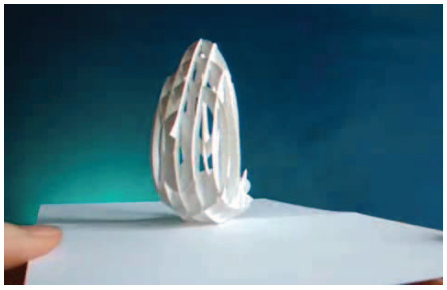
(b) End frame of the clip $I(x,y)$

(c) Profile $P(t,y)$

Figure 6.12.: A car videoed from its surrounding during a fast movement. The profile includes two sides of background.



(a) Flow graph and cutting trajectory. The time axis is downward.



(b) End frame of the clip $I(x,y)$

(c) Profile $P(t,y)$

Figure 6.13.: Rotating object in front of the camera and its profile.

The foreground target is also cut and the width is extended (target is emphasized), and the order is reversed in the profile. This selection of slicing direction is more reasonable than the opposite way which extends a partial background and squeezes the foreground in the profile. Figure 6.12 shows an example in which the focused car taken by a camera on another moving car.

We deal with general around-object motion along a smooth path and camera rotation. Equivalently, a camera moving on a straight rail rotating towards a focused target can be considered similarly. Further, a rotating target in front of a static camera can be treated as this type of motion [27], as shown in Fig.6.13.

6.4 Visualize Dynamic Foreground

In this section, we are aiming at present both shape and motion information in a video profile. To avoid the shape being destroyed completely, we employ motion blurring-enhancing approach [38, 39] in the profiling. If we extend the exposure time of an image, dynamic objects are motion-blurred, because the intensity at each point is accumulated temporally. Static objects have consistent intensities over time and their average are still sharp. People can perceive the motion information when the profile is displayed with motion blur.

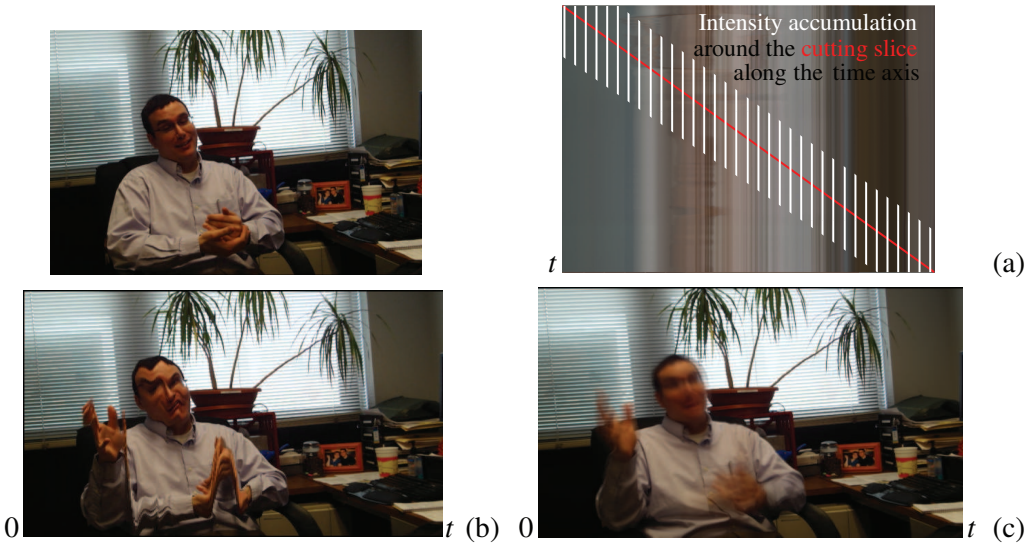


Figure 6.14.: Motion blur created in video profile for representing dynamic foreground. (a) Accumulating intensities along time axis during slice cutting. (b) The result from a simple slice cutting without motion blurring. (c) The video profile with motion blur and sharp background. The video frame is on the top-left.

If the cutting speed is slow due to a long video clip as in Fig.6.14a, even a mild motion of foreground might be single-slice profiled with a distortion. We increase a degree of motion blur by temporal averaging around the cutting slice. As shown in Fig.6.14a, we increase the width of the slice with thickness of γ in the video volume for averaging. Given global flow direction V, which is a stable result, temporal averaging in the flow direction is

$$P(t,y) = \frac{1}{\gamma} \sum_{\tau=-\gamma/2}^{\gamma/2} I(x(t) + \bar{v}_x\tau, y(t) + \bar{v}_y\tau, t + \bar{v}_t\tau) \tag{6.2}$$

Thus, Fig.6.14b can be improved by motion blur as the result in Fig.6.14c by setting $\gamma = 35$ frames.

This accumulation has two effects. It motion-blurs the dynamic foreground object with a different flow direction from the background (major flow), and enhances the background in the profile that may be motion blurred in each individual frame. Another result in camera panning case is given in Fig.6.15.

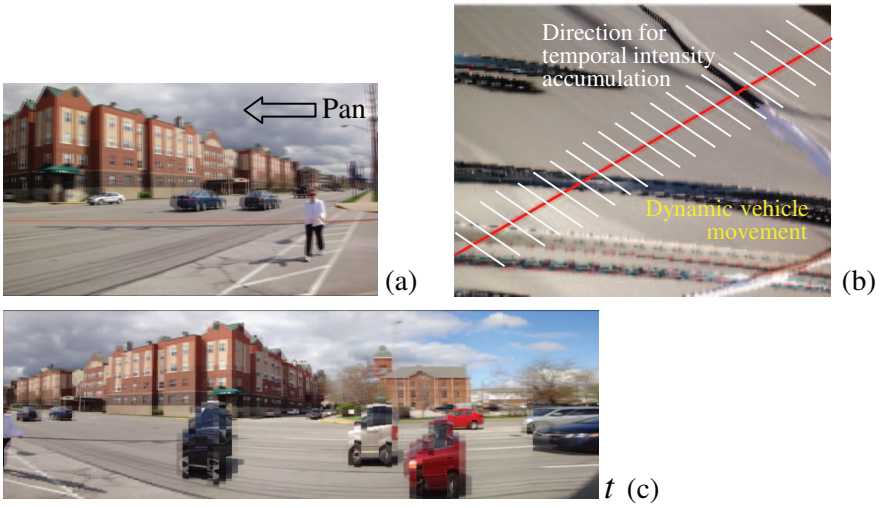


Figure 6.15.: Motion blurring for dynamic foreground and rotating background. (a) A video frame during camera panning left. (b) A flow graph and averaging intensities along the background flow direction. (c) Scaled video profile in time with sharp background and blurred cars from the diagonal cut in (b).

Furthermore, if the camera involves translation, the motion parallax is then not homogeneous in each frame. It depends on the object depth from the camera. We can only compute the dominant motion parallax at each time instance for intensity accumulation along that direction. This means that only the object in the dominant parallax (depth) will be clear and objects off the depth will have a certain degree of motion blur.

7 SHAPE IMPROVEMENT OF THE GENERATED VIDEO PROFILE

This chapter uses information in both the motion-oriented and shape-oriented condensed images for the rectification of video profiles as a post-process. In the motion-oriented image, the motion traces of major vertical features are kept. The major flow shows the speed and direction of major camera operation. The inconsistency of the traces in the motion-oriented image are brought in by the speed variation of the camera operation. This issue will be taken care of by the introducing of shape mode for the video profile, which can be seen as linearizing a curved trace to correct the video profile with less distortion from the speed variation. This can improve the aspect ratio of major scenes in the video profile to be close to the perspective projection. In the shape-oriented condensed image, a feature reveals two motion components. The degree of blur in the horizontal direction is related to the image velocity of the feature [34], and the deviation in the vertical direction provides the camera shaking evidence. The first one is difficult to measure because the feature may mixture with the neighboring ones, while the second one exhibits the shaking parameters of the camera apparently for us to rectify the video profile. We will introduce a method that makes the positive use of blur as an effective filter to rule out the unreliable features for wave straighten. Note that these two methods only work on the camera operations that generate directional flows. The diversified flow are hard to model, since it's caused either by inconsistent camera zoom that varies camera by camera, or by unpredictable foreground crowd.

Figure 7.1.: The profile (top) and spatial mosaic (bottom) of videos with continuous camera operation (pan right with tilt up) with camera pan as major operation. The continuous up-tilt operation can be seen from the profile (top) as the decline of the structure line toward right. The spatial mosaic (bottom) cannot fit in the time line of video editing software due to the irregular shape caused by various camera operations.

7.1 Information Captured in the Video Profile

Since our method is to use a scan line to sweep against the major flow, all the background scenes appear once in the profile. The shape distortion information in the temporal direction is analyzed as follows.

1. If the major flow is much larger than minor flow (Fig.3.3). This is the most case.

   (a) In the case that the minor flow exists in addition to the major flow, the structure line will be deformed by the minor flow projection, i.e., for the horizontal structure line in a pan+tilt video, the profile shows deformed structure line. This phenomenon is shown in Fig.7.1.

(b) Camera shaking in the minor flow is recorded in the profile as the minor flow. The profile shaking can be rectified if necessary, based on shape orientated condensed image (Section 7.3).

2. Depending on the camera motion (if it is horizontal leftward), the generated profile may be spatially inverted (Fig.8.1), although it is perfectly correct in the temporal domain. The shape mode display can horizontally flip the profile if it is requested. This is because our profile is forced to align with the time axis rightward.

3. Background aspect ratio deformation

   (a) If the cut is fast (because the clip is short), most of the background scenes will be recorded in a good shape in the profile. The shape in the video profile is similar to the video frame.

   (b) If cutting is slow

      i. If the flow is slow, the profile can be scaled narrowly in the temporal direction. Static background is deformed locally with a temporal scaling as compared to its original shape in frame. However, this is tolerable because the displayed video track is originally scalable along the time axis for editing and browsing in most video software. A shape mode of profile (Section 7.2) is prepared to rectify this for a better display (Fig.7.3).

      ii. If the flow is fast. Another effort is to design cascade cutting to avoid slow cutting. This is introduced in Section 7.4.

4. Foreground motion may be inconsistent with background motion.

   (a) Most of the time, it's fine if foreground has small motion against background (almost as background).

(b) If the slice cutting is fast, i.e., cutting speed is large, the foreground is merged into the background and the shape distortion is insignificant even when the relative motion is large as demonstrated in Figs.8.2, 8.3, 8.4.

(c) If the slice cut is slow

i. If foreground moves quickly, e.g., face expression, articulate movements and rotation, and minor flow in shooting the clip, the target shape may be damaged in the profile as motion traces.

ii. If a foreground target moves also slowly (e.g., camera focuses on it), the target is extended in the profile. Inversely, a target is squeezed in the profile if it passes the field of view quickly. These effects match the videographer's intention to emphasize or ignore targets. This can be improved if the shape mode display is triggered on, if the traces of foreground are sufficiently distinct. The way to solve this problem is through the averaging of slices in the direction of V_x or V_y over a small range so that a motion-blurred foreground is obtained in the profile (Figs.6.14, 6.15) as suggested in [27]. This is particularly effective on moving people taken by a static camera (Fig.8.3). Other method is also under exploration.

The motion information is partially contained in the profile.

1. The temporal information is accessible along the horizontal axis for video editing (specifying frames). The temporal order of the profile is consistent with the camera moving direction, rather than the original spatial order. A scene may have reversed order from what is observed in the video. We notify this effect with color underlines as in Fig.8.1.

2. A camera zoom action is recognizable from scene structure scaling along the time axis (Fig.6.7c).

7.2 Display Profile in Shape Mode

The cutting is determined by the length of the section, which yields the profile different from the perspective projection. The resulting video profile thus has shape distortions. For a more pleasant experience for viewers, this work further introduces a shape mode for the profile in addition to the temporal mode that strictly follows time code.

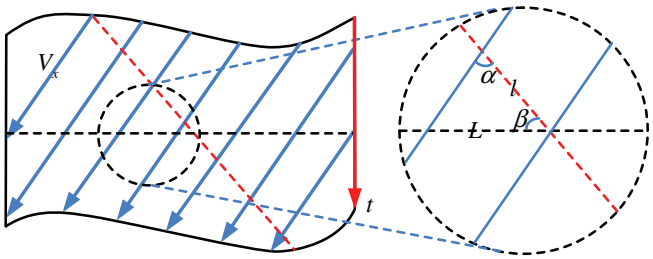
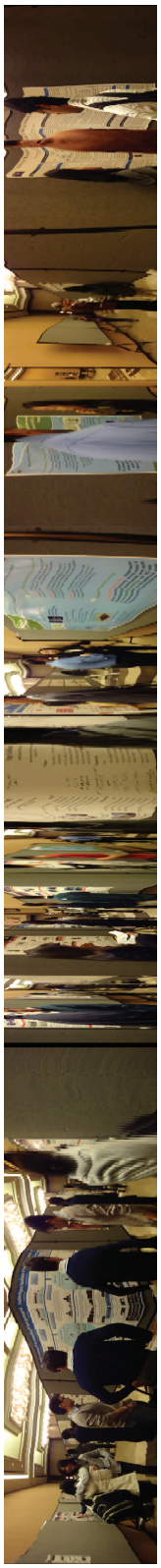


Figure 7.2.: Temporal scaling of profile for better shape

To preserve more spatial information, the resulting video profile needs to be resized according to the angle between the cutting path and the major flow direction. As shown in Fig.7.2, if the cutting length denoted as line segment $l(t)$ can be locally scaled to the same length as in the image shown as line segment $L(t)$, the shape can be preserved better in the profile. In triangle in Fig.7.2 formed by cutting segment $l(t)$ (several pixels), its corresponding scene length $L(t)$ in the shape mode profile, and the major flow V_x/V_y, the corresponding scene length is

$$L(t) = l(t)\frac{\sin\alpha(t)}{\sin(\alpha(t)+\beta)} \tag{7.1}$$

where $\alpha(t)$ is the angle between $l(t)$ and V_x/V_y, and β is the angle between $l(t)$ and the frame plane. Both are known angles computed already. The shape mode is only applied to the profile of the camera motion with directional flows, i.e., translation or panning. Figure 7.3 shows such a result to normalize the profile for a better shape of scenes.



Temporal mode display of profile.

Shape mode of profile is scalable in the temporal domain. The video contains forward walking accompanied with pans.

Figure 7.3.: Local scaling of temporal mode to shape mode of Fig.1.3.

7.3 Shaking Removal in Video Profile

Another effort to improve the shape of profile is to rectify the profile without shakings, although the footprint of shaking might be useful in video editing and evaluation. We examine the shape-orientated condensed image and use the local traces of stationary blurring to rectify the up-and-down motions in the profile. A straightening technique [23] is applied to curved lines in such condensed image. Figure 7.4 shows such results of minor flow reduction in the profiles for a better visualization.

Here we use the positive aspect of the *stationary blur* [34, 35] to remove unstable motions in the video profile. The generating of shape-oriented condensed image (Fig.3.5b) automatically enhances the long-lasting features named *lighthouse features* for revealing the camera shakings and suppresses irrelevant features. By tracking the trajectory of such lighthouse features continuously in the condensed images, a video profile can be rectified and normalized at the sub-pixel level in a continuous way. In addition, the small data size processed achieves the efficiency and robustness in generating good-quality video profile.

7.3.1 Shaking Embedded in Shape-oriented Condensed Images

It's found that both distant and horizontal features in the 3D space appear as long traces in the shape-oriented condensed image. In Fig.7.5, the horizontal window structure (top) also forms long connected traces (bottom). The waving of the traces gives good evidences of the camera tilt changes. On the contrary, the vertical features such as lines and points in the frames are largely stationary-blurred.

If a portion of a horizontal line on a building is occluded by a small object, the horizontal line will be nicely connected in the condensed image. This will allow for longer horizontal lines to be tracked and straightened. As compared to the line tracking based video profile deshaking, this solves the problems of feature tracking in video profile [40] and the matching of confusion patterns in *Shape from Motion* [41] and image stitching method. Repetitive patterns such as windows and building decora-

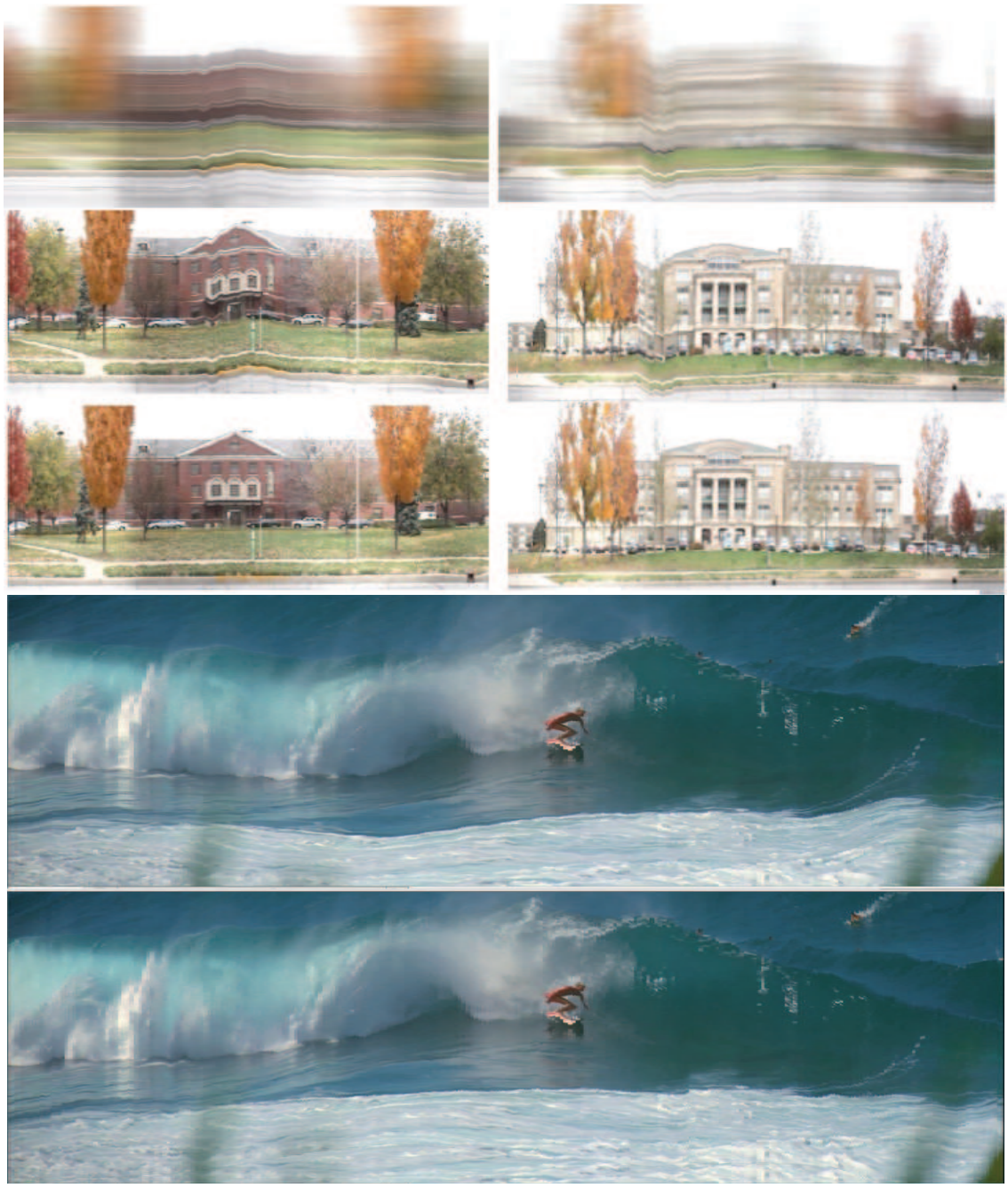


Figure 7.4.: Reducing shakings affected by irregular camera movement for better display (top) shape-oriented condensed image, (middle) profiles with minor flow shaking, (lower) Profiles with shaking removed.

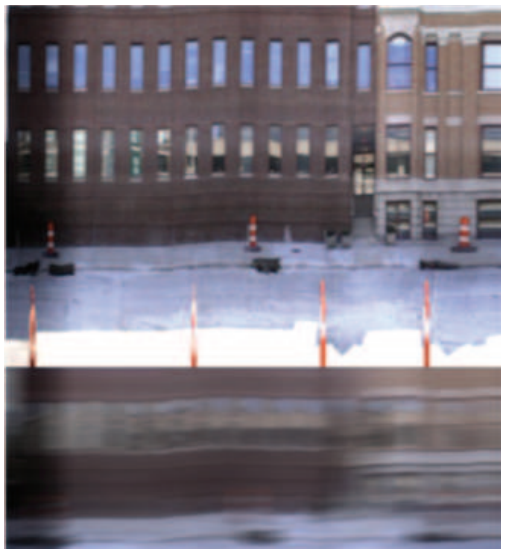


Figure 7.5.: The video profile at top with shape-oriented condensed image at the bottom

tions that are interrupted in line tracking in the video profile. They are connected as longer lines in the condensed image, because the vertical features are blurred out. Also, slanted long lines on roofs will not be selected mistakenly as references for rectifying video profile, because they are also blurred out.

This work first filters the shape-oriented condensed image vertically to detect the edge and then track continuous edge points horizontally with two-level thresholds. Dense traces thus are detected at the sub-pixel level (1/3 pixels) as in Fig.7.6. Instead of using many unreliable features and their average for estimating shaking parameters between frames, this work focuses on a few curves from features visible for a long period, which is more favorable in tracking the camera tilt.

Generally, a distant feature (with large Z) stays in the video for long time as reliable references of camera motion. This is because it gives reliable orientation information. The distant and widely visible lighthouse feature is more distinct than a conventional landmark feature that is only unique in contrast to its surroundings. Denoting the length of a 3D horizontal segment by L, starting from X, its appearing scope along the path in the condensed image is

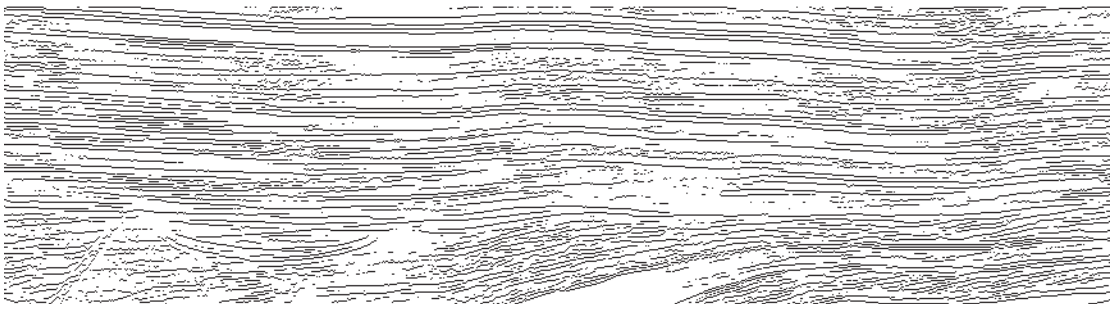


Figure 7.6.: Tracking of dense traces (non-crossing) in the shape-oriented condensed image.

$$(X - Zw/f, X + L + Zw/f) \tag{7.2}$$

according to the convolution property in (3.3)(3.4). The segment length in the $P(t, y)$ for tracking is computed as

$$\triangle t = \frac{m\left(L + \frac{2Zw}{f}\right)}{V} \tag{7.3}$$

which is much longer than its length mL/V in the video profile. This proves that a long or distant feature (either large L or Z or both) under a strong stationary blur will provide a reliable evidence (long period of $\triangle t$) for motion/shaking detection. On the other hand, closer features have relatively higher image velocities, which appear short and less stationary-blurred in $C_x(t, y)$.

The condensed image has revealed the motion characteristics of video in an intuitive way. The stationary blur effect caused by accumulating the pixels which makes a distant scene last longer in the condensed image than in the video profile. The deshaking of the video profile can thus be done by tracking horizontal traces in the

condensed images for finding shaking parameters, and then applying the correction parameters back to the video profile for aligning vertical columns.

7.3.2 Local Deshaking Based on Trace Tracking

In the shape-oriented condensed image, a feature reveals two motion components horizontally and vertically. The degree of blur in the horizontal direction is related to the image velocity of the feature [34], while the deviation in the vertical direction provides the camera shaking evidence. The first one, i.e., the blurred degree, is difficult to measure because the feature may mix with the neighboring ones, while the second one, i.e., the deviation, exhibits the shaking parameters of the camera apparently for rectifying the video profile.

Let us estimate the deviation related to the camera shaking. Assume that the local frame has horizontal features F_k, $k = 0, 1, 2, \ldots$ at height y, and the lengths of F_k are x_k, respectively. The tangent of a trace in y direction in the condensed image $C_x(t, y)$, if detectable, is

$$\begin{aligned}\frac{\partial C_x(t,y)/\partial t}{\partial C_x(t,y)/\partial y} &= \frac{\sum_{x=-w/2}^{w/2} \frac{\partial I}{\partial t}}{\sum_{x=-w/2}^{w/2} \frac{\partial I}{\partial y}} = \frac{x_1 I_t^{(1)} + x_2 I_t^{(2)} + \ldots + x_k I_t^{(k)} + \ldots}{x_1 I_y^{(1)} + x_2 I_y^{(2)} + \ldots + x_k I_y^{(k)} + \ldots} \\ &= \frac{x_1 I_y^{(1)} \frac{I_t^{(1)}}{I_y^{(1)}} + x_2 I_y^{(2)} \frac{I_t^{(2)}}{I_y^{(2)}} + \ldots + x_k I_y^{(k)} \frac{I_t^{(k)}}{I_y^{(k)}} + \ldots}{x_1 I_y^{(1)} + x_2 I_y^{(2)} + \ldots + x_k I_y^{(k)} + \ldots} \\ &= \frac{\sum_k x_k I_y^{(k)} v^{(k)}}{\sum_k x_k I_y^{(k)}} \end{aligned} \tag{7.4}$$

where $I_t^{(k)}$ and $I_y^{(k)}$ are temporal and spatial partial derivative of feature k.

For post-processing deshaking, dense traces are tracked from edges for local jitters in $C_x(t, y)$ and then $P(t, y)$. To find a shaking location from traces, two median filters are applied vertically first and horizontally then on edge traces. The first filter obtains a common vertical shift from multiple traces at any time instance, i.e.,

$$\triangle \hat{y}(t) = \text{median}(\triangle y_1(t), \triangle y_2(t), \ldots, \triangle y_n(t)) \tag{7.5}$$

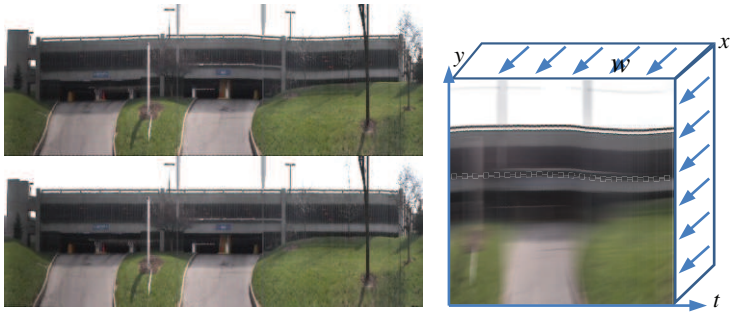


Figure 7.7.: A condensed image with visible motion traces. (left) Original $P(t, y)$ and rectified $P(t, y)$ with curve removed. (right) Enlarged condensed image $C_x(t, y)$ in a span of $P(t, y)$ shows curved building traces.

assuming the number of lines, n, is more than a threshold. This process works when multiple traces provide common evidence of shaking in a short period. The second median filter has a large horizontal span N along the time axis (a large number of frame) for removing the sharp sparks as noises in the vertical shift distribution, to obtain the vertical shift, y'(t), for deshaking at each position, i.e.,

$$y'(t) = \text{median}(\triangle\hat{y}(t+\tau)),\ \tau \in [-N, N] \tag{7.6}$$

The move of the vertical column of video profile for correction is at pixel level such that the $P(N, y)$ and $C_x(N, y)$ will not be affected in resolution at this stage. Figure 7.7 is such an example where a rectified section is displayed in local video profile. Although N is large, we have used a revised median filter algorithm [19] for consecutive input data of a large sequence to achieve the median filtering in linear complexity (i.e., $O(n)$, instead of $O(n \log n)$ for median filtering by a general sorting algorithm). That ensures the deshaking process moving forward at a constant speed regardless the window size N.

7.3.3 Global Wave Reduction Based on Lighthouse Features

In addition to local jitters, we further straighten large structures in the video profile according to waved traces of lighthouse features from horizontal lines or distant points in the scene. The waves are caused by driving on inclined road surfaces. As demonstrated in Fig.7.8, long curves in $C_x(t, y)$ are tracked with a low threshold. A set of continuous traces $r_i(s_i, e_i),\ i = 1, 2, 3 \ldots$ are obtained with length $e_i - s_i$ as the process moves forward sequentially. Then, we straighten such curves successively in $C_x(t, y)$, resulting in the difference between original traces and straightened one for the video profile deshaking.

In the implementation, the following steps are performed. (1) During the tracking of a trace, its length is counted and, at the end, the length information is labeled backward onto the entire trace. To guarantee a robust deshaking by referring to global and static features, the traces shorter than a threshold are ignored. We found that a single reliable trace from a lighthouse feature yields a much better result in the video profile deshaking than using multiple noisy traces. (2) For every position t, the longest trace that covers the position is marked in $C_x(t, y)$. For all the traces $r_k,\ k = 1, 2, 3, \ldots$ covering t, i.e., $t \in [s_k, e_k]$, there exists a trace j that satisfies

$$\text{length}(r_i(t)) \geq \text{length}(r_k(t)) \tag{7.7}$$

(3) A sequence of non-overlapped longest traces $r_j(t)$ are followed for rectifying waved video profile, i.e., several consecutive longest traces $r_j(t),\ j = 1, 2, 3, \ldots$ cover the entire video profile.

The wave rectification of the video profile has two modes. (I) One is to generate straightened scenes while keep the video profile within the image frame, which is suitable for image visualization. (II) The other is to recover the true scene height by accumulating the motion from the beginning of path. The generated video profile can easily drift out of the window frame due to the error accumulation or the path on a hill road. Our video profile deshaking obeys mode I to generate piecewise straight video profile. The curved traces of longest segments are precisely located in $C_x(t, y)$ and are straightened based on end points. The pixel transformation is then applied

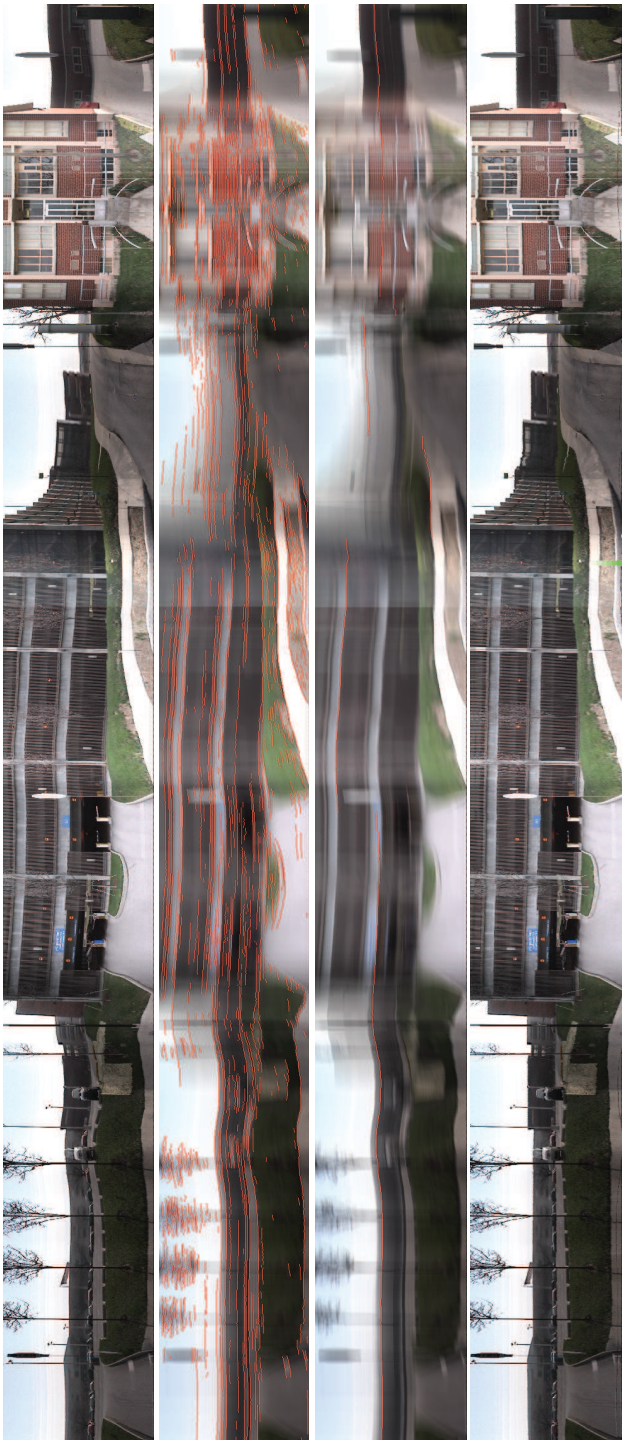


Figure 7.8.: The process to rectify video profile with distance feature long lasting in the shape-oriented condensed image $C_x(t, y)$. Video profile, C_x with tracked traces (red), the longest traces (red) in C_x, and video profile after straightening according to the longest traces are displayed from left to right. The method works for wide area.

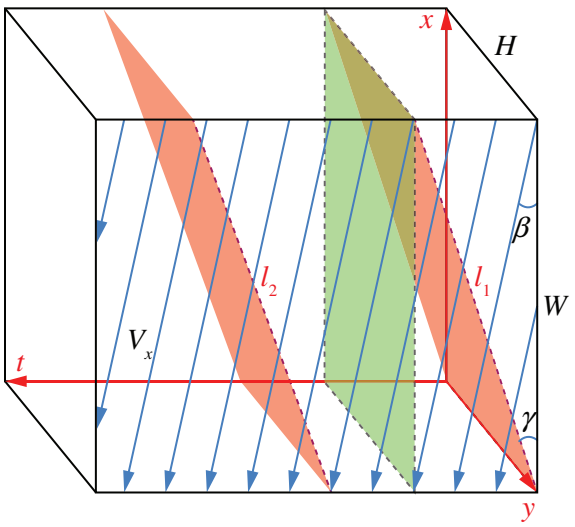


Figure 7.9.: Evaluation of the video profile in terms of the aspect ratio. W and H are the width and height of the video frame. l_i are the paths of the sampling line. Angle β indicates the direction of the major flow direction. Angle γ indicates the direction of the sampling line. The green plane indicates a frame of the video.

to the video profile accordingly. As a result, we solve the waving problem of video profile by straightening traces of lighthouse features over long periods. This is difficult by matching consecutive 2D frames that may cover only partial structures in the scenes [42]. Figure 7.8 demonstrates such an example of correcting the waved video profile based on the long lasting lighthouse features in the shape-oriented condensed image. It works on wide and large depth area.

7.4 Cascade Cutting for Acceptable Aspect Ratio of Profile

In the triangle formed by the sampling plane (Fig.7.9), the major flow, and the frame plane, the projected length is $w = W\sin\beta/\sin(\gamma + \beta)$. The aspect ratio of the video profile is thus determined by the angle γ formed by the sampling plane and the frame plane. If we choose an acceptable aspect ratio as λ, we have $H/w = \lambda$.

$$\frac{W \sin\beta}{H \sin(\beta+\gamma)} = \frac{1}{\lambda} \tag{7.8}$$

Expand the equation and divide both sides by $\sin\beta$ ($\beta \in (0, \pi/2]$), we can get

$$\lambda W = H\cos\gamma + H\cot\beta\sin\gamma \tag{7.9}$$

Since angle β shows the direction of the major flow, we have $\cot\beta = |\bar{v}_x|/|\bar{v}_t|$ with $\bar{v}_t^2 + \bar{v}_x^2 = 1$. We can obtain the expression of γ as

$$\begin{aligned}
\sin\gamma &= \frac{\lambda W\cot\beta \pm \sqrt{H^2 + H^2\cot^2\beta - \lambda^2 W^2}}{H\cot^2\beta + H} \\
&= \frac{\lambda W\sin\beta\cos\beta \pm \sin\beta\sqrt{H^2 - \lambda^2 W^2\sin^2\beta}}{H} \\
\cos\gamma &= \frac{\lambda W \mp \cot\beta\sqrt{H^2 + H^2\cot^2\beta - \lambda^2 W^2}}{H\cot^2\beta + H} \\
&= \frac{\lambda W\sin^2\beta \mp \cos\beta\sqrt{H^2 - \lambda^2 W^2\sin^2\beta}}{H}
\end{aligned} \tag{7.10}$$

after solving (7.9) as a quadratic equation with respect to $\gamma \in [0, \pi/2)$, considering $\sin^2\gamma + \cos^2\gamma = 1$. In (7.10), we need to have $\lambda \leq H/(W\sin\beta)$.

If the angle γ is too large, the preset aspect ratio λ may be invalidated. In this condition, we may further split the video volume to smaller segments and set more sampling planes based on the strategy introduced before, from one end of the segment to the other. There are two options for the starting position of the new sampling plane. One is to start the new plane at the end frame of the current plane. This might introduce duplicated scenes near the connections of two consecutive profiles. The other is to start the new plane at the frame projected from the end frame of the current plane along the major flow direction. This approach might give better result than the first method in the sense that it won't duplicate the scene. But the result might suffer from inaccurate estimation of major flow due to local waves.

8 EXPERIMENTS

8.1 Generating Profiles

As a video is read in, it is scanned for color condensing. With the two condensed images, we segment video to clips according to the discontinuity of color distribution as traditional approaches [36] and then to sections with smooth camera motions as shown in Fig.4.1. The blue curves show the variance of the horizontal component of the trace vector. A relatively larger variance suggests a zoom operation or component. The averaged flow direction from accumulated trace angles is shown with red curves along the time axis. A median filter of size $3\sigma_v$ is applied to remove the noises caused by foreground moving objects. In each segment, two components of the major flow vector are computed and compared from the gradient values on traces. This information is then used in selecting slice alignment and the cutting direction in the motion-oriented condensed image. From the convergence/divergence factor computed in the condensed image, the bending curve is determined to generate the profile accordingly.

We have examined our method on hours of videos in profiling them. A video clip may contain a back-and-forth panning that is a concatenation of our simple and smooth camera movements, as can be found in Fig.8.1a. Some non-trivial results are shown in Figs.8.1b, 8.2, 8.3, 8.4.

The computation time is $T(\text{3D}) + T(2 \times \text{2D}) + T(\text{2Dslice})$ for the intensity voting to condensed images, filtering for the aggregate motion vectors, and slice cutting in the video clip, respectively. It is much less than filtering the video volume itself for optical flow. Software has been developed to perform this task on PC. The testing video clips are mostly from YouTube and other web video service providers. If a

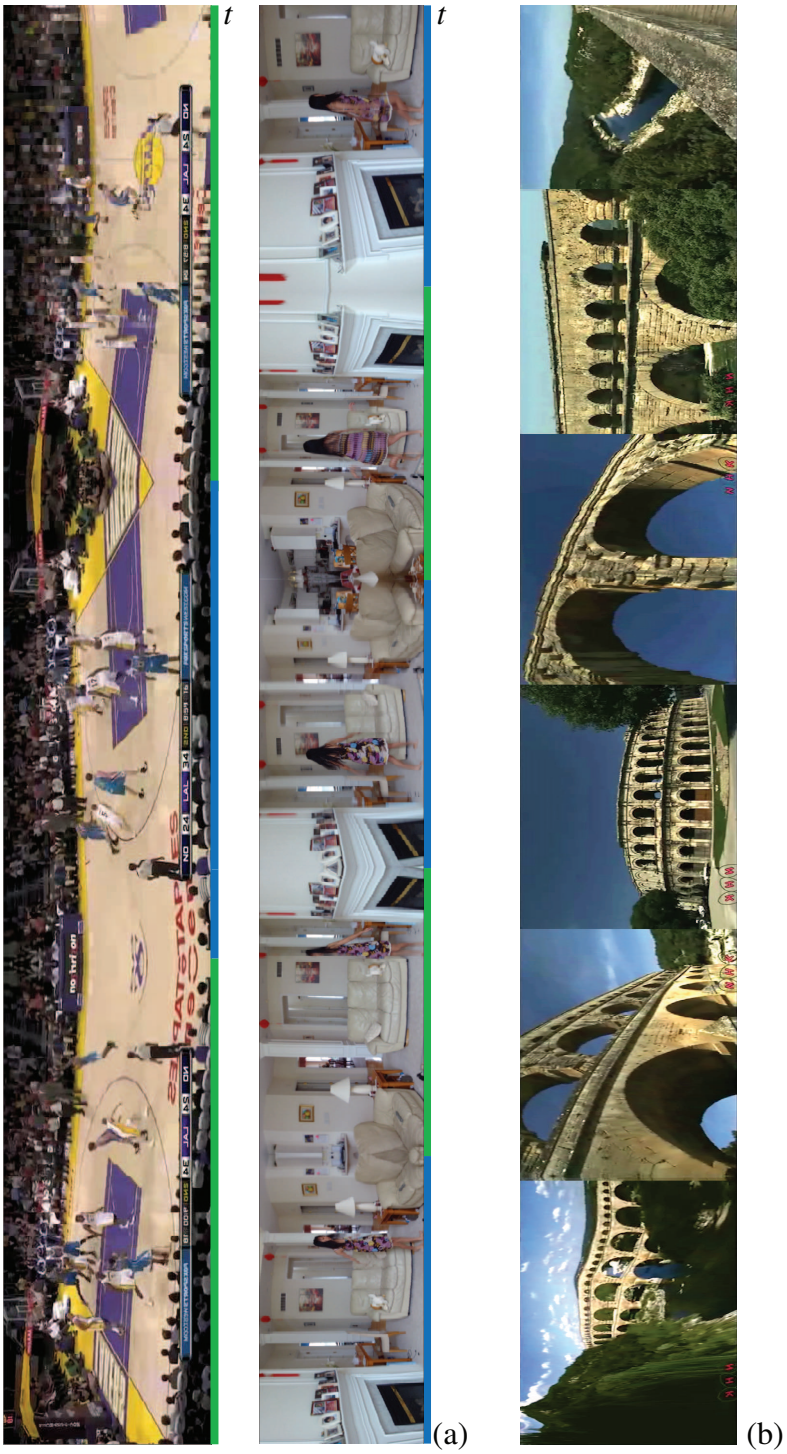


Figure 8.1.: (a) Back-and-forth panning on a basketball game and a dancing girl. (b) Profile of a video capturing a world heritage in Roman. There are pan operations and a forward moving (similar as zoom) operation on an airplane in the video.

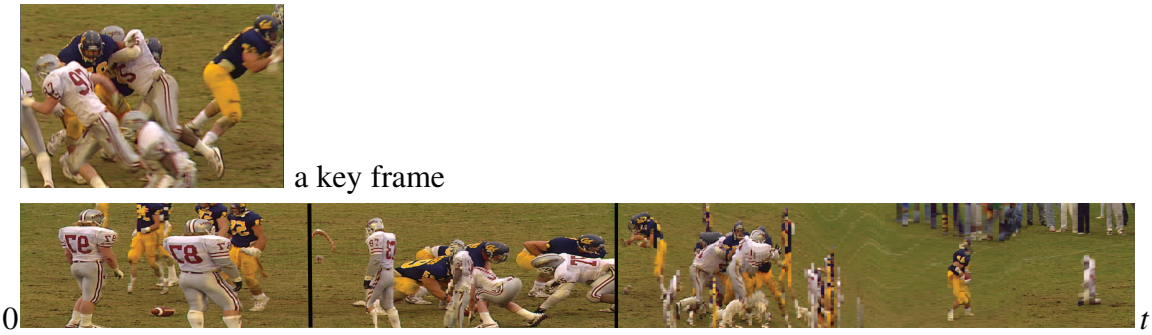


Figure 8.2.: Large camera motion following the crowded actions. The profile from three consecutive pans shows the game progress in the temporal domain. A key frame is also attached. It contains true shapes but is hard to know context in the video.

clip has severe shaking, the deshaking technique can smooth the video prior to the profiling. We have developed software to perform the video profiling on a laptop PC (Dell XPSL511Z) in real time (processing 35 frames per second).

8.2 GUI for Video with Profile

The profile of video makes the video track in the video software and web visible. This allows the user to search the scenes of interest effectively before watching the video itself. We have explored various interfaces of using video profile to enhance the video browsing, searching, and comparison.

Along with the video window and operation buttons, an associated video profile is displayed in the video track. It is constructed to be scalable and scrollable in time for scene search in the video. The frame indicator on the profile is synchronized with the frame in the video. Users can interact with the profile by using mouse on PC and finger on mobile devices. By specifying a scene, the corresponding frame in the video is pulled out in the video display window. By indicating a range of frames in the profile, users can replay, copy and paste the video segment.

Figure 8.3.: A wall of temporal profiles contains a sports ceremony. The profile cutting method works successfully in general. Although deformation happens in profiles for the discussed reasons, there is no difficulty to identify the scenes. The camera operations are mostly camera pan and crane motion.

Figure 8.4.: A wall of temporal profiles contains the TV program of an MTV. It's easy to identify the singers and dancers. Each video segment with smooth camera operation lasts a short time so the resulting video profile is more similar to a set of key frames.

In addition to the frame-profile pair display for browsing and editing, we also display profiles of a large video set consecutively in a large window called video wall as shown in Fig.8.5. The wall is scalable and scrollable as well. It provides a function to locate scenes and allows users to quickly compare video clips briefly to find duplications. If a position is clicked in the video wall by mouse, the corresponding frame further pops up near the location specified. The most powerful function is to display the frame in a separate window side-by-side with the wall window, where sweeping the mouse position over the profiles realizes a fast video forward in the frame window.

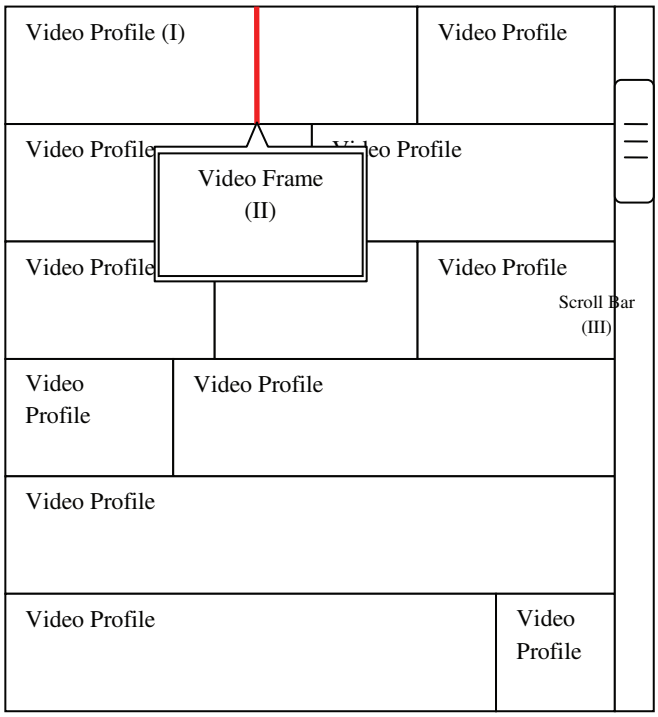


Figure 8.5.: Video Wall Display with profiles of a long video with the indication of the functions on the top.

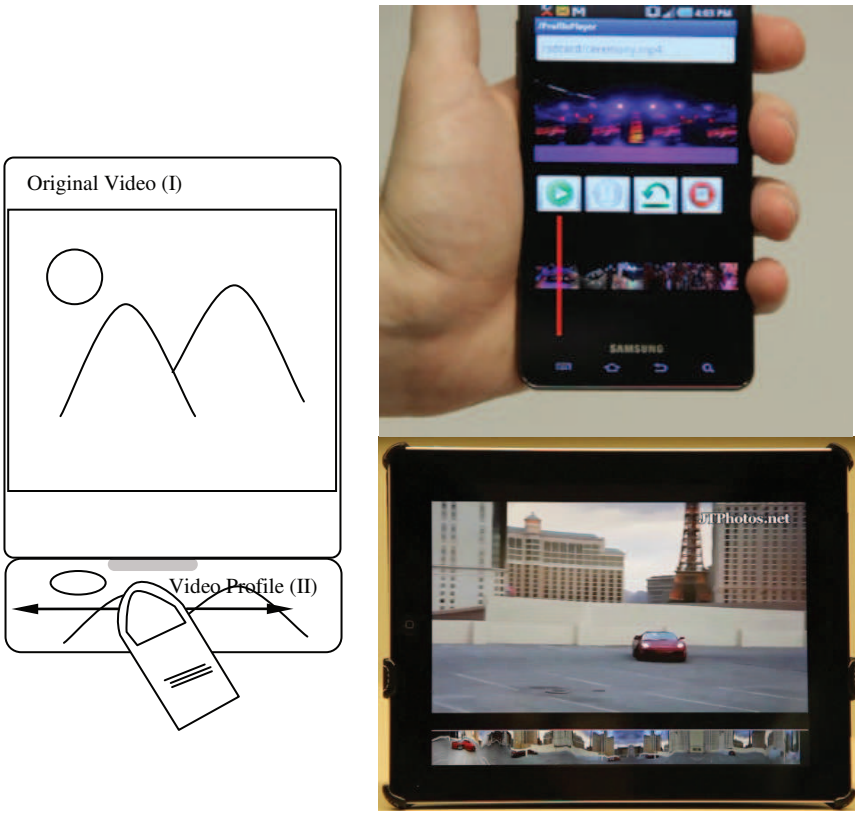


Figure 8.6.: Video profiles on various mobile devices such as Android phone and iPad. The profiles of videos displayed underneath the video frames are synchronized with the video.

We have developed software for PCs, iPad, and Android phones to examine the effectiveness of the video profile in helping video access as shown in Fig.8.6. We used Java binding with OpenCV in the video wall. For Apple iPad, we coded with Objective-C on Apple and used the software xCode to test and debug the code. A simple drag and drop interface was created to bring code into the iPad.

9 DISCUSSION

Many TV programs are the concatenation of clips from static cameras. If the clip is short, the resulting profile is not very different from the key frame itself. But the profile has a higher temporal resolution than key frame because the longer diagonal line than the frame width. Because the $|\bar{v}_x|$ and $|\bar{v}_y|$ are small for static camera and pure zoom, both horizontal and vertical cutting can be considered. We perform a vertical slice for a profile.

Figure 9.1.: Long video profile before and after rectifying waved image. The horizontal axis is time. There are repeating patterns on the brick building.

Compared to the spatial mosaicing, the motion and temporal information is accessible along the time axis in the profile. If the camera motion is slow or the section after smooth camera operation is short, our video profile is almost identical to key frames (Figs.6.4, 8.4), since the slice forms a small angle with the video frame and the segmentation process takes into consideration the camera shot transition and camera operation change.

The generated profile is almost identical to the stitching-based method for camera pan (Fig.9.2). Plus, our method has a better and smooth connection between consecutive pixel lines. Image stitching has artifacts at boundaries or seams, whatever an effort to find invisible seams is or re-projection from 3D is made. Our method pro-

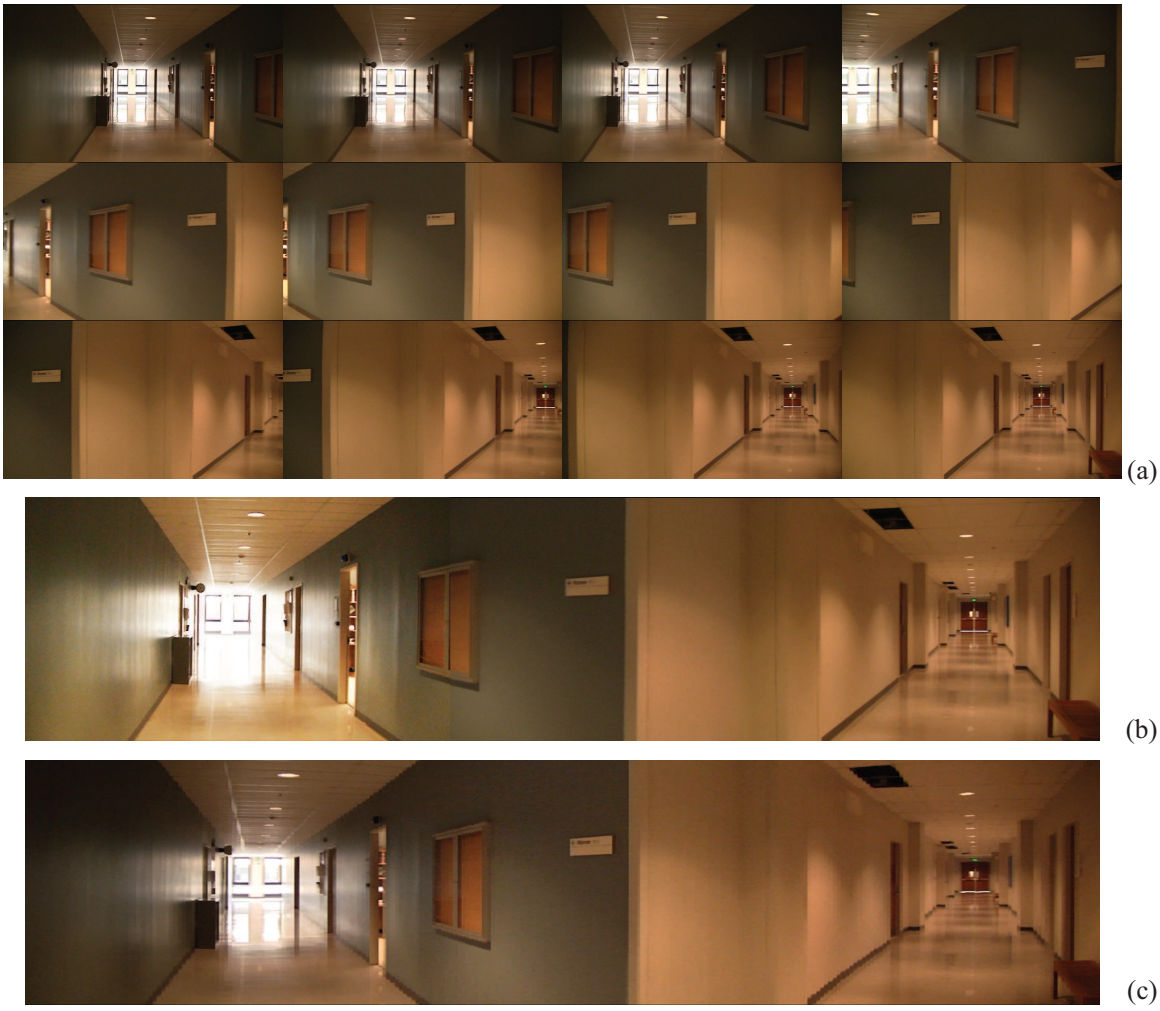


Figure 9.2.: The comparison between the spatial method and video profile method in a video of camera pan. (a) Key frame method (b) Mosaic method (c) Our slice cutting method.

duces a long image directly before we rectify it. On the other hand, image stitching obtains a long view after many steps of segmentation, matching/optical flow identification, patch optimization, etc. Image stitching has fundamental problems in dealing with the different motion parallax or disparities at the same place. This is not a problem for our method as can be seen in Fig.9.1 from a traveling camera along a long route.

At the current stage, our profile may not be used purely for registering actions of a person. Users are directed to see video itself. Although the graphics rendering approach has achieved artistic video annotation, the methods only work on static and rotating camera so far [12, 17]. The scene segmentation may not succeed if the scene complexity increases. On the contrary, this work aims at automatic video profiling for general camera motion for indexing video database. A profile needs to present true scenes for video retrieval. Multiple copies of targets may cause confusion. Although [4] using linear patches scanning approach creates a less distorted shape, the optimization is extremely time-consuming because of the ignorance of the global flow direction. Based on our test, on a 2000 frame video shown in Fig.3.5 costs more than one hour by using [4] in generating the mosaic. Moreover, the temporal resolution and scale is not consistent due to the fact that a large shape determines the stitching size of the patches (a car may still be squeezed temporally due to a large background).

Since our profile includes more complete scenes in the video than key frames, it can be used for video comparison for duplicated clips at a coarse level more efficiently than comparing video volumes. In this sense, the profile can be used as a reliable intermediate video representation for retrieval. For a profile from surveillance video with traffic and people through a location (S_T in Fig.6.3), one can have a glance at a target before checking the video. In the profile, we can even count passing people in a group activity such as marathon and parade, which is easier than counting in overlapped frames because the data in profile is non-redundant.

10 CONCLUSION

This work addresses a general framework of automatic profiling of video volumes for video digests. Based on the analysis of camera motion and global flow, a uniformed algorithm has been implemented on simple and combined camera motions, which theoretically guarantees the profiles from various video clips. The global motion of camera has been estimated efficiently with two condensed images for determine the slice cutting, and the automatically generated 2D profiles containing both temporal and spatial information. Besides the background scenes, the moving foreground objects are registered as motion-blurred shapes to express the motions and relative positions. The profiling method is global, more robust and faster than mosaic-based methods. Post-processing is also employed to improve the display of the video profile. It can automatically map video database to profiles to facilitate video browsing and editing. GUI on both PC and hand held devices are designed and developed to prove the usefulness of proposed video profile.

LIST OF REFERENCES

LIST OF REFERENCES

[1] D.B. Goldman, B. Curless, D. Salesin, and S.M. Seitz. Schematic storyboarding for video visualization and editing. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 862–871. ACM, 2006.

[2] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1971–1984, 2008.

[3] Y. Caspi, A. Axelrod, Y. Matsushita, and A. Gamliel. Dynamic stills and clip trailers. *The Visual Computer*, 22(9):642–652, 2006.

[4] Y. Wexler and D. Simakov. Space-time scene manifolds. In *Tenth IEEE International Conference on Computer Vision*, volume 1, pages 858–863. IEEE, 2005.

[5] wikipedia.org. Wikipedia. `http://www.wikipedia.org`.

[6] B. Janvier, E. Bruno, T. Pun, and S. Marchand-Maillet. Information-theoretic temporal segmentation of video and applications: multiscale keyframes selection and shot boundaries detection. *Multimedia Tools and Applications*, 30(3):273–288, 2006.

[7] Google Inc. Youtube. `http://www.youtube.com`.

[8] A. Aner and J. Kender. Video summaries through mosaic-based shot and scene clustering. *European Conference on Computer Vision (ECCV)*, pages 45–49, 2006.

[9] A. Rav-Acha, G. Engel, and S. Peleg. Minimal aspect distortion (mad) mosaicing of long scenes. *International Journal of Computer Vision*, 78(2):187–206, 2008.

[10] C. Barnes, D.B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics (TOG)*, 29(4):89, 2010.

[11] A. Bartoli, N. Dalal, and R. Horaud. Motion panoramas. *Computer Animation and Virtual Worlds*, 15(5):501–517, 2004.

[12] F. Liu, Y. Hu, and M.L. Gleicher. Discovering panoramas in web videos. In *Proceedings of the 16th ACM International Conference on Multimedia*, pages 329–338. ACM, 2008.

[13] C.D. Correa and K.L. Ma. Dynamic video narratives. *ACM Transactions on Graphics (TOG)*, 29(4):88, 2010.

[14] D.N. Wood, A. Finkelstein, J.F. Hughes, C.E. Thayer, and D.H. Salesin. Multiperspective panoramas for cel animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 243–250. ACM Press/Addison-Wesley Publishing Co., 1997.

[15] M.L. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 5(1):2, 2008.

[16] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 853–861. ACM, 2006.

[17] Y. Taniguchi, A. Akutsu, and Y. Tonomura. Panoramaexcerpts: extracting and packing panoramas for video browsing. In *Proceedings of the Fifth ACM International Conference on Multimedia*, pages 427–436. ACM, 1997.

[18] J.Y. Zheng. Digital route panoramas. *IEEE Multimedia*, 10(3):57–67, 2003.

[19] J.Y. Zheng and S. Sinha. Line cameras for monitoring and surveillance sensor networks. In *Proceedings of the 15th International Conference on Multimedia*, pages 433–442. ACM, 2007.

[20] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1144–1154, 2000.

[21] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):741–754, 2003.

[22] G.R. Flora and J.Y. Zheng. Adjusting route panoramas with condensed image slices. In *Proceedings of the 15th International Conference on Multimedia*, pages 815–818. ACM, 2007.

[23] H. Cai, J.Y. Zheng, and H. Tanaka. Acquiring shaking-free route panorama by stationary blurring. In *IEEE International Conference on Image Processing*, pages 921–924, 2010.

[24] G. Daniel and M. Chen. *Video visualization*. IEEE, 2003.

[25] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaicing: Mosaicing of dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1789–1801, 2007.

[26] H. Cai and J.Y. Zheng. Video anatomy: cutting video volume for profile. In *Proceedings of the 19th ACM International Conference on Multimedia*, pages 1065–1068. ACM, 2011.

[27] J.Y. Zheng, H. Cai, and K. Prabhakar. Profiling video to visual track for preview. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2011.

[28] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[29] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. *European Conference on Computer Vision (ECCV)*, pages 1–14, 2008.

[30] M. Hu, S. Ali, and M. Shah. Detecting global motion patterns in complex videos. In *19th International Conference on Pattern Recognition*, pages 1–5. IEEE, 2008.

[31] J.Y. Zheng, Y. Bhupalam, and H.T. Tanaka. Understanding vehicle motion via spatial integration of intensities. In *19th International Conference on Pattern Recognition*, pages 1–5. IEEE, 2008.

[32] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Evolving time fronts: Spatio-temporal video warping. In *SIGGRAPH*, 2005.

[33] J.Y. Zheng, Y. Zhou, and P. Mili. Scanning scene tunnel for city traversing. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):155–167, 2006.

[34] J.Y. Zheng and M. Shi. Scanning depth of route panorama based on stationary blur. *International Journal of Computer Vision*, 78(2):169–186, 2008.

[35] J.Y. Zheng and M. Shi. Removing temporal stationary blur in route panoramas. In *18th International Conference on Pattern Recognition*, volume 3, pages 709–713. IEEE, 2006.

[36] Y. Murai and H. Fujiyoshi. Shot boundary detection using co-occurrence of global motion in video stream. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[37] R.C. Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, 1989.

[38] M. Potmesil and I. Chakravarty. Modeling motion blur in computer-generated images. In *ACM SIGGRAPH Computer Graphics*, volume 17, pages 389–399. ACM, 1983.

[39] G.J. Brostow and I. Essa. Image-based motion blur for stop motion animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 561–566. ACM, 2001.

[40] J.Y. Zheng. Stabilizing route panoramas. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 348–351. IEEE, 2004.

[41] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[42] E. Zheng, R. Raguram, P. Fite-Georgel, and J. Frahm. Efficient generation of multi-perspective panoramas. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 86–92. IEEE, 2011.

[43] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, 1996.

[44] M.M. Yeung and B.L. Yeo. Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5):771–785, 1997.

[45] J.Y. Zheng and S. Tsuji. Generating dynamic projection images for scene representation and understanding. *Computer Vision and Image Understanding*, 72(3):237–256, 1998.

[46] H.H. Baker and R.C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3(1):33–49, 1989.

[47] M.H. Kolekar and S. Sengupta. Semantic concept mining in cricket videos for automated highlight generation. *Multimedia Tools and Applications*, 47(3):545–579, 2010.

[48] J.Y. Zheng, Y. Fukagawa, T. Ohtsuka, and N. Abe. Acquiring 3d models from rotation and highlights. In *Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing*, volume 1, pages 331–336. IEEE, 1994.

[49] J.Y. Zheng and M. Shi. Mapping cityscapes into cyberspace for visualization. *Computer Animation and Virtual Worlds*, 16(2):97–107, 2005.

[50] K. Schoeffmann and D. Ahlstrom. Similarity-based visualization for image browsing revisited. In *IEEE International Symposium on Multimedia (ISM)*, pages 422–427. IEEE, 2011.

[51] Michal Irani and P Anandan. Video indexing based on mosaic representations. *Proceedings of the IEEE*, 86(5):905–921, 1998.

[52] Jianke Zhu, Steven CH Hoi, Michael R Lyu, and Shuicheng Yan. Near-duplicate keyframe retrieval by nonrigid image matching. In *Proceedings of the 16th ACM International Conference on Multimedia*, pages 41–50. ACM, 2008.

[53] Heung-Yeung Shum and Li-Wei He. Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 299–306. ACM Press/Addison-Wesley Publishing Co., 1999.

[54] Paul Rademacher and Gary Bishop. Multiple-center-of-projection images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 199–206. ACM, 1998.

[55] Harpreet S Sawhney, Serge Ayer, and Monika Gorkani. Model-based 2d&3d dominant motion estimation for mosaicing and video representation. In *Fifth International Conference on Computer Vision*, pages 583–590. IEEE, 1995.

[56] virtualdub.org. Virtualdub. `http://www.virtualdub.org`.

[57] opencv.org. Opencv. `http://opencv.org`.

VITA

VITA

Hongyuan Cai received his B.E. of Computer Science and Technology from Beijing Forestry University, China, in 2007. He entered the doctoral program of Computer Science in Purdue University in 2009. He received his Ph.D degree from Purdue University in 2013. He has received multiple awards from the University and travel grants from major international conferences including ACM Multimedia and IEEE ICME. His research interests include computer vision and multimedia computing. He has actively engaged in research involving different projects including Panorama Planning, Route Panorama Rectification, In Car Video, and Spatial-temporal Video Summarization. He has coauthored more than 10 research articles. He is also a reviewer for IEEE Transactions on Intelligent Transportation Systems and IEEE Transactions on Multimedia. He was also a reviewer for ICPR 2012, IROS 2011, IEEE R&A 2011, and IEEE ICME 2011. After graduation, he joined Synopsys.