International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

Video Mining for Creative Rendering

Mei Chen

***Abstract*—**More and more home videos are being generated with the ever growing popularity of digital cameras and camcorders. For many home videos, a photo rendering, whether capturing a moment or a scene within the video, provides a complementary representation to the video. In this paper, a video motion mining framework for creative rendering is presented. The user's capture intent is derived by analyzing video motions, and respective metadata is generated for each capture type. The metadata can be used in a number of applications, such as creating video thumbnail, generating panorama posters, and producing slideshows of video.

***Keywords*—**Motion mining, semantic abstraction, video mining, video representation.

I. Introduction

To preserve precious memories of life, people record vast amount of video using movie cameras and camcorders. Recently, many digital cameras have also implemented video capture functionality, capturing video clips up to VGA resolution. In a few years, compact digital cameras will be able to capture hours of high definition digital videos.

Currently, there is no good way to enjoy or manipulate one's home videos. For example, with a two-hour video digitized and sitting on the hard disk, the only way to know what is in the video is to open up the media player, play it from the beginning, or move the scrolling bar to an arbitrary point and play it from that time on. Some DVD software can generate chapters while authoring a DVD disc. Yet the chapters are only represented by the first frame where there is a scene change. It is difficult to get a quick *sound bite* of a home video, let alone doing anything creative with it. Therefore, albeit taken with much affection and anticipation, most home videos end up sitting on the shelf or on the disk, rarely revisited, fading into oblivion. The motivation of this research is to help people achieve more enjoyment and better experience from their home videos.

There has been much research on video analysis, especially around video highlight detection, keyframe extraction, video summarization, and automatic video editing. Instead, we are proposing a new framework targeting at a different application scenario: mining video motion to automatically derive an appropriate representation for various video segments. The derived representation can be a panorama, a collection of keyframes, or a close-up of an object of interest.

Our research theme came from the following observation: people capture home videos for a variety of purposes. One of the main purposes is to capture action and sound, in which case many trophy shots representing a fleeing moment have been captured on home videos. Another one is to capture the environment, such as a panoramic view on top of a mountain. Still another purpose is to capture an object that has caught the attention of the videographer, such as a close-up of a humming bird perching on a tree branch. The goal of our research is to automatically detect these capture types, classify them into the right category, and generate the appropriate metadata that can be used for further rendering.

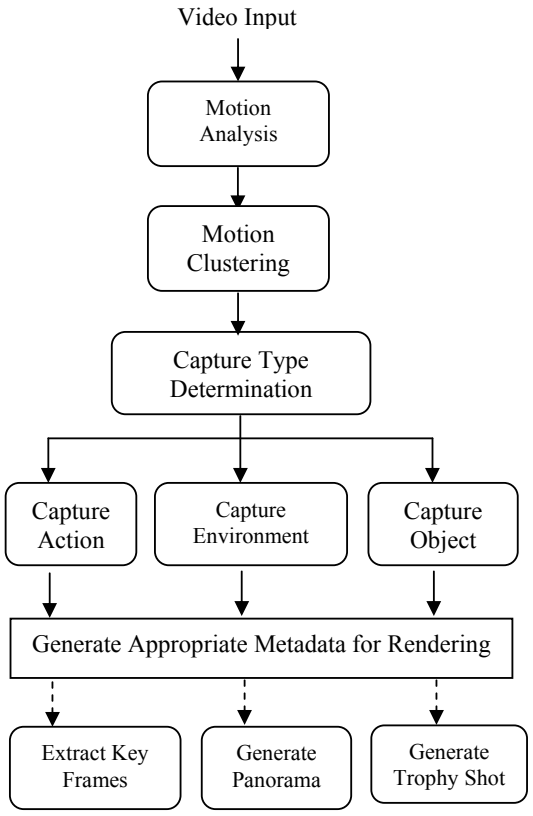


Fig. 1 Overall framework for the video motion mining systems for creative rendering

Prior research on video motion mining focused on analyzing human motion in video sequences. There has not been a solution provided for automatic repurposing of various segments in a video clip. In the case of video panorama

Manuscript received July 31, 2005.
Mei Chen is with Hewlett Packard laboratories, Palo Alto, CA 94304 USA (phone: 650-857-4762; fax: 650-857-2951; e-mail: mei.chen@ hp.com).

generation, although there has been extensive research, most work deal with rendering only, assuming that the whole video clip is taken with the intent for panorama creation. However, in the reality of consumer home videos, a segment that is suitable for panorama rendering is often embedded within a long video sequence of various content. Using existing technologies, the user would have to manually cut out the relevant video segment in order to have a panorama created, which requires the inconvenience of using a video editing application. The advantage of our system is that it automates the above process, as well as having the ability to discover and repurpose video excerpts that are suitable for other kinds of rendering, such as action prints.

The rest of the paper is organized as follows. Section 2 presents the video mining framework, followed by detailed algorithm description in section 3. Section 4 and 5 discuss implementation issues and experimental results. Then section 6 concludes with discussions and future research directions.

II. System Overview

Fig. 1 shows the overall framework of our video mining system. We first perform motion analysis to determine motion types between neighboring frames, e.g. whether it is panning or zooming. We then cluster frames with similar motion types in a hierarchical manner. This will identify segments of video frames with similar motion types. Depending on each specific motion type, we classify the corresponding video segment into a respective capture type, and generate the appropriate metadata that can be used for further rendering or manipulation. For example, if the camera motion of a video segment is a *steady panning* motion type, it indicates that the user is trying to capture the environment. We store that metadata, together with the frame-to-frame image motion that can be used for panorama creation. If the camera motion is a *zooming-in* motion type followed by a *still* motion type, it suggests that there is something the user is interested in, and trying to capture at greater detail. We generate the metadata to indicate the frame of interest that the user can extract or print, possibly with some type of frame enhancement, such as the one described in [3]. If there is considerable object motion, it is inferred that the user is trying to capture some action. In this case, we then generate the metadata that can be used to extract representative frames from the video segment to produce action shots.

Using this process, we enable the rendering of a collection of representative photos from videos, in the form of panorama images, detail shots, or action sequences. Together they form a synopsis of the whole video. To browse a video file, one can either do a slide show of the photo collection, or do a layout of the photos.

III. Video Motion Mining for Creative Rendering

The video motion mining algorithm consists of four main steps: it first analyzes the image motion between each pair of consecutive video frames; secondly, it finds clusters of adjacent video frames with similar camera motion; then it determines a motion type for each cluster of video frames; and lastly, it inferres capture types based on the motion type classification, and generates the metadata that can be used for the corresponding renderings. Fig. 2 shows a flow diagram of the algorithm.

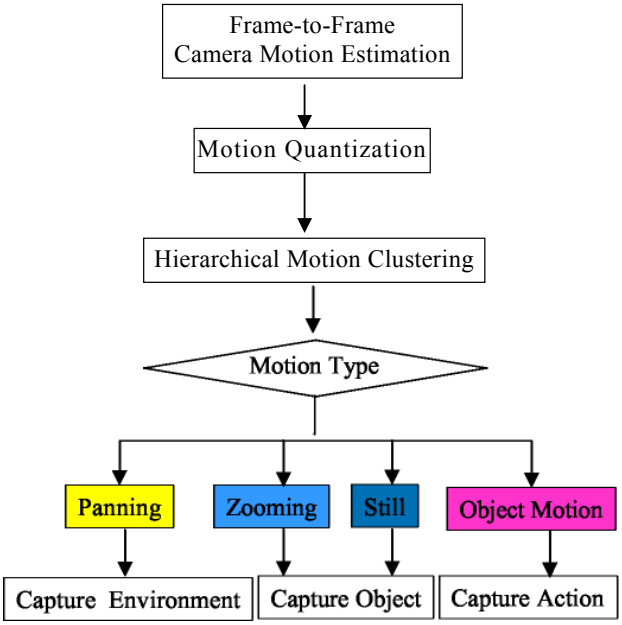


Fig. 2 Flow diagram of video motion mining algorithm

A. Camera Motion Analysis

The algorithm starts with frame-to-frame camera motion analysis. For camera motion analysis, we adopt the affine motion model, and compute it from optical flow estimates. For optical flow estimation, we use a gradient based, iterative technique that has been proven to be robust under a variety of image motion scenarios [1]. For affine model computation, we adopt the least squared error (LSE) regression method described in [2].

We use four parameters to model the camera motion $C = C(F, \alpha, \beta, \gamma)$, where F is the camera's focal length, and α, β, γ are the three rotation angles along the X, Y, Z axes, respectively, as illustrated in Fig. 3. For an arbitrary point P(x, y, z) in the three dimensional world, it will be projected onto a point Q(u, v) in the camera's imaging plane, following the transformation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} zoom & rotation & pan_x \\ rotation & zoom & pan_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = A \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

where zoom, rotation, pan_x and pan_y are four parameters

decided by the camera motion C. The affine model is estimated by using least squared error LSE regression as discussed in [2]:

$$A=(X^TX)^{-1}X^TU,\quad X=\begin{pmatrix}x_1 & x_2 & \dots & x_N\\ y_1 & y_2 & \dots & y_N\\ 1 & 1 & \dots & 1\end{pmatrix},\quad U=\begin{pmatrix}u_1 & u_2 & \dots & u_N\\ v_1 & v_2 & \dots & v_N\end{pmatrix}$$

After affine motion estimation, each pair of successive frames is associated with a set of motion parameters indicating the camera motion at that instant. Based on these parameters, the algorithm proceeds to extract higher-level semantic meaning over moderate time spans.

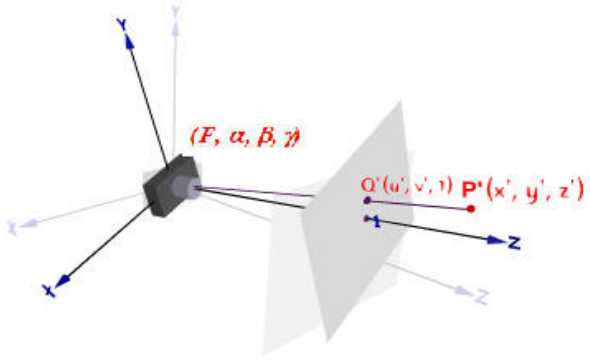


Fig. 3 Camera motion geometry

B. Motion Quantization

The computed camera motion is typically represented as floating point parameters, and to extract semantic meaning from them requires appropriate interpretation of these numbers. Moreover, the estimated motion may contain spurious or insignificant results, such as the videographer's hand tremor, and such noise should be filtered out prior to semantic extraction. To fulfill the above requirements, we quantize the computed camera motion parameters into a number of magnitude levels and directions. In particular, the magnitude of frame-to-frame panning motion along X or Y axis is classified into three levels: *negligible, steady,* and *large*; whereas the panning direction is binned into eight quadrants, namely *up, down, left, right, left up, left down, right up, right down*. Similarly, the magnitude of frame-to-frame zooming motion is classified as *negligible or significant,* with the direction identified as *in* or *out*.

This quantization of motion parameters enables meaningful association of the frame-to-frame, instantaneous camera motion within a time span, as well as the extraction of the dominant component within a composite camera motion.

C. Motion Clustering

After camera motion quantization, the next step in video motion mining is to uncover the underlying associations between the frame-to-frame, instantaneous camera motion. To achieve this goal, we adopt a hierarchical clustering technique to group video frames based on their motion similarities.

At the bottom of the hierarchy, video frames within a pre-determined, short, temporal distance can be grouped together based on quantized camera motion similarity. That is, adjacent frames sharing the same quantized camera motion are merged into a unit of longer time span. This step finds in the video clip groups of frames within which the quantized camera motion is consistent. This lays the foundation for further abstraction.

To proceed to the next level of this hierarchical clustering, an average camera motion is computed for each group of video frames. This average camera motion is then quantized, and used to merge groups into larger clusters of video frames of similar camera motions. Specifically, groups within a predetermined temporal distance can be combined based on quantized camera motion similarity. During this operation, groups with durations longer than a predetermined threshold will function as *anchors*, whereas the other groups are merged into these anchors.

This process can be iterated to generate an even smaller set of video segments within which the camera motion is consistent. The *collective* camera motion of such video segment can be computed as an average of the camera motion of its member groups, weighed by each group's temporal duration. Each video segment is associated with a motion type.

D. Motion Type Determination

Once a video clip is consolidated into segments of consistent camera motion, a *motion type* is determined for each video segment, based on its quantized camera motion. The current motion types of consideration are: *panning, zooming, object motion,* and *still*.

The classification of the panning, zooming, and still motion types is rather straightforward. In particular, if a video segment has *negligible* panning or zooming motion, then it will have *still* motion type; if a video segment has dominant panning motion, then it will have *steady panning* or *fast panning* motion type, depending on its quantized parameters; similarly, if a video segment has dominant zooming motion, it will have *zooming* motion type. For the object motion category, the following rules are used: if a video segment's camera motion does not belong to any of the above motion types, and the magnitude and/or directions of local motion vectors vary beyond predetermined thresholds, the video segment will be assigned the motion type *object motion*.

E. Capture Type Classification

The last step of video motion mining is to infer meaningful higher level semantics such as *capture types*, from the above motion types. Specifically:

(1) If a video segment's motion type is *zooming*, the algorithm checks if it is followed by a video segment of *still*

motion type, and if the still motion lasts longer than a predetermined length of time. If both are true, the algorithm will determine that the user was to zoom in to *capture an object* of interest. It then records the indices of the video frames at the beginning and the end of the still motion segment as metadata, as well as the estimated frame-to-frame image motion during that period. The metadata can later be used to extract a video frame with the object of interest; or, if desired, it can also be used to initialize a multi-frame super-resolution enhancement process to enhance a frame with the object of interest to a higher spatial resolution [3].

(2) If a video segment's motion type is *steady panning*, the algorithm will determine that the user was trying to capture a panoramic view of the scene. It will record the indices of the video frames at the beginning and the end of the steady panning segment as metadata, together with the estimated frame-to-frame image motion during that period. This metadata can be used to extract the panoramic segment; or, when desired, this metadata can be used to evoke a panorama rendering engine to generate a panorama of the scene.

If a video segment is associated with a *fast panning* motion type, the algorithm will infer that the user was quickly moving the camera to a new scene, and had little interest in the intervening views.

(3) If a video segment's motion type is *object motion*, the algorithm will infer that the user was recording an object or objects in action. It then records the indices of the frames at the beginning and the end of that video segment as metadata, as well as the estimated frame-to-frame image motion. This metadata can be used to create a slideshow or a collage of the action clip; or, if desired, it can be fed into a keyframe extraction module to select action shots from that video segment.

IV. Implementation Notes

In order to make the process conducive for embedded system implementation, we have made significant effort to optimize the algorithm. The series of measures we have taken at different stages of the procedure include: temporal and spatial down-sampling, the use of look-up tables, the simplification of routine algorithms, and the repetitive use of intermediate results, etc.

V. Experiments

We have tested the system on 31 home video clips with various motion scenarios, including panning left/right, zooming in/out, and action scene. For motion quantization and clustering, the system succeeded in most cases except for a long panning segment: it did not completely filter out the videographer's hand tremor, and separated that one panning sequence into two segments. For motion and capture type determination, the system performed well except for two occasions: in one clip it missed a zooming capture because the duration of the still motion following the zooming motion was below the predetermined threshold; in another clip, it misclassified an object motion type because that was a long shot, and the objects were too small to pass the predetermined local motion threshold. Fig. 4(a) shows an example of the semantic content classification of a video clip.

Given that we can obtain keyframes to represent actions in a video segment, panoramas to represent the environment, and high resolution photos to represent important shots, we can generate various output types to represent the entire video. One such example is to generate a video thumbnail page. The important advantage we have over a regular video thumbnail page is that we can give a much better indication of the content of the video – what actions took place, the surroundings captured, and the important frames. Moreover, such an output can be printed out as a story book, representing highlights and details of the video in high quality.

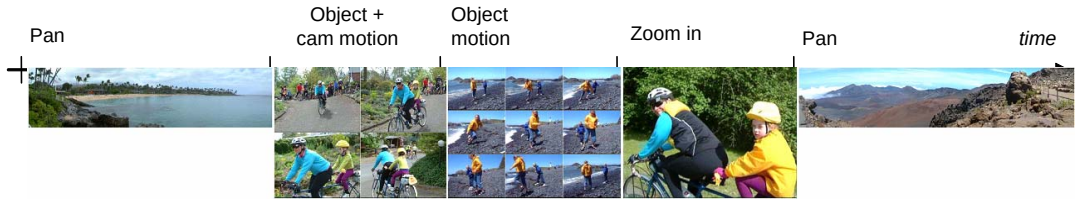
Fig. 4(b) is an example of such a video thumbnail page. On the upper row, the image on the left is a super-resolution enhanced version of a video frame which gives a clear view of the two riders in the video clip. In the middle and on the right side are two keyframe sets extracted from two video segments with action, with the number of keyframes being four and nine, respectively. On the lower row are two panorama pictures produced from two video segments with panning motions. Overall, this video thumbnail page provides a complete, succinct, and diversified view of the content in the video, including stories, actions, sceneries, as well as details.

VI. Conclusion

We have presented a framework for video motion mining to provide a complementary user experience. Our technology automatically computes frame-to-frame camera motion of a video sequence, conducts statistical analysis of the video motion profile to extract semantic interpretation of the video content, and discovers interesting patterns for creative rendering. Our current system recognizes three capture types: capturing of object actions, capturing of the environment, and capturing of objects in detail. Based on these capture types, the system generates the metadata that can be used for rendering respective image representations for these capture types, such as keyframes, panoramas, and detail shots.

This technology can be widely used in video understanding and management applications such as video summarization, editing, annotation, browsing, printing and retrieval. One application scenario is to have it running in the background as soon as the user uploads a video, and discover interesting patterns for creative rendering; and, if desired, automatically generate the respective representation for the user to enjoy.

In the future, we plan to conduct user studies to make the algorithm less sensitive to preset thresholds, and to extend the framework to include more motion capture types for more output representations, such as panorama capture with panning in swathes.



(a)

(b)

Fig. 4 (a) An example of the video motion mining result, (b) An example of a video thumbnail page

REFERENCES

[1] J. Bergen, P. Anandan, and K. Hanna, "Hierarchical model-based motion estimation", ECCV, 1992.J. Park, N. Yagi, K. Enami, K. Aizawa, and M. Hatori, "Estimatino of camera parameters from image sequence for model based video coding", IEEE Trans. On Circuit and System for Video Technology. Vol. 4, No. 3, June 1994.

[2] R. L. Rardin, *Optimization in Operations Research*, Prentice Hall, 1998, ISBN: 0023984155.

[3] M. Chen, "Dynamic Content adaptive super-resolution", Int. Conf. Image Analysis and Recognition, Sept. 2003.

[4] [Polana92] Ramprasad Polana and Randal C. Nelson, Recognition of Motion from Temporal Texture, Proc. IEEE Conference on Computer Vision and Pattern Recognition, Champaign, Illinois, June 1992, 129-134.

[5] [Kim97] Eung Tae Kim, Jong Ki Han, Hyung-Myung Kim, "A Kalman-filtering method for 3D camera motion estimation from image sequences," Proceedings of ICIP 97, vol. 3, pp. 630-633, Oct. 1997.