VISUALIZATION OF VIDEO MOTION IN CONTEXT OF VIDEO BROWSING

Klaus Schoeffmann, Mathias Lux, Mario Taschwer, Laszlo Boeszoermenyi

Institute of Information Technology, Klagenfurt University, 9020 Klagenfurt, Austria
{ks,mlux,mt,laszlo@itec.uni-klu.ac.at}

ABSTRACT

We present a new approach for video browsing using visualization of motion direction and motion intensity statistics by color and brightness variations. Statistics are collected from motion vectors of H.264/AVC encoded video streams, so full video decoding is not required. By interpreting visualized motion patterns of video segments, users are able to quickly identify scenes similar to a prototype scene or identify potential scenes of interest. We give some examples of motion patterns with different semantic value, including camera zooms, hill jumps of ski-jumpers, and the repeated appearance of a news speaker. In a user study we show that certain scenes of interest can be found significantly faster using our video browsing tool than using a video player with VCR-like controls.

Index Terms— Video Browsing, Video Exploration, Motion Visualization, Video Abstraction, Video Surrogates

1. INTRODUCTION

A common task in video browsing is to find similar scenes within a video. In such a use case a user identifies a scene-of-interest and tries to find similar scenes by using domain heuristics inferred from already found scenes. Visualization schemes representing a large amount of information about the video at once help users to find a way through the video and speed up the browsing task. Therefore, video browsing heavily relies on powerful visualizations to allow for effective visual analysis by the user.

We present a novel method to visualize motion information of a video based on motion vector classification of H.264/AVC compressed bit streams. Working in compressed domain allows very fast analysis and enables on-demand visualization for video browsing scenarios. We represent motion in videos based on motion vector histograms (one per frame) using a newly introduced motion vector classification scheme. We further employ a transformation of the histograms to the HSV color space for the purpose of visualization. This transformation pertains both similarity and dissimilarity of scenes and, therefore, allows users to find similar scenes by simple visual inspection. For example, if a scene features a lot of fast motion to the right it will be represented by an intense red-magenta color. If a second scene shows less distinctive motion to the same direction a less saturated tone of red will be used for visualization. A change in the direction of the motion would result in a change of the hue of the visualization. Visualization is employed as means for interactive navigation through a video and allows users to (1) interlink high-level semantics with the visualization of low-level motion information for a specific segment and to (2) immediately jump to that segment by a simple mouse click. In an evaluation we show that users can identify scenes of interest more efficiently in a typical video browsing scenario with our visualization than with a standard soft video player - the ”poor man’s” video browsing tool.

2. RELATED WORK

Many *video browsing* approaches have been already presented in the literature, some of them are described in [1, 2, 3, 4]. For example, Adams et al. [1] presented a video browsing approach based on a *tempo function* which is computed of camera motion, audio energy, and shot length. Campanella et al. [4] proposed a video browsing tool which uses visualization of MPEG-7 features like dominant color, motion intensity, edge histogram, etc. Motion is also important in the area of *video retrieval*. According to Su et al. [5], ”most existing content-based video retrieval systems do utilize motion as one of the features when executing a search process”. However, to the best of our knowledge, no video browsing approach has been proposed so far, which uses meaningful visualization of motion in a video in order to allow a user to deduce content semantics and, thus, quickly identify possibly interesting segments.

3. MOTION CLASSIFICATION

We use motion vector information contained in H.264/AVC bit streams to create a motion histogram for every frame of a compressed video sequence. Motion classification is a low-complexity task, as it does not require full video decoding. In fact, the complexity is dominated by entropy decoding, which consumes 22 to 42 percent of the full decoding workload [6]. Motion vector prediction and motion classification add only about 8 percent to this workload, according to measurements

using our H.264/AVC decoder implementation. Although our approach could be applied to earlier video coding standards as well, H.264/AVC allows for a more precise motion classification due to the concept of macroblock partitions resulting in up to 16 motion vectors per macroblock. Moreover, neighboring macroblock partitions need not be predicted from the same reference frame. As Su et al. noted [5], motion vectors represent the sum of camera motion, object motion, and noise introduced by the motion estimation algorithm of the encoder. Consequently, our approach will visualize both camera motion and object motion, but will suffer from noisy motion vectors.

The classification scheme considers forward-prediction by a single motion vector per macroblock partition only. Motion vectors representing backward-prediction or a second prediction for bi-predicted partitions are ignored for now, but may be subject of future work. Intra-coded macroblock partitions of predicted frames are considered as still motion using a motion vector of $(0, 0)$. The motion histogram of an intra-coded *frame*, however, is duplicated from the previous frame to support object motion crossing a GOP boundary.

Our approach is based on one-frame-distance motion vectors predicting from frame $n-1$ to frame n (in presentation order). A multiple-frame-distance motion vector $\mu_{k,n} = (x, y)$ of a macroblock partition in frame n, predicting from frame k to frame n with $k < n - 1$, is linearly interpolated to a one-frame-distance motion vector μ:

$$\mu = \frac{\mu_{k,n}}{n-k} = \left(\left\lfloor \frac{x}{n-k} + \frac{1}{2} \right\rfloor, \left\lfloor \frac{y}{n-k} + \frac{1}{2} \right\rfloor \right) \quad (1)$$

The one-frame-distance motion vectors of a particular frame are classified into a motion histogram with $K + 1$ bins, modelling K equidistant motion direction intervals (bins $b \in \{1, \ldots, K\}$) and a separate bin ($b = 0$) for zero-length motion vectors, as illustrated in Figure 1 for $K = 12$. We define the *direction* of a motion vector $\mu = (x, y) \neq (0, 0)$ with length $|\mu|$ as:

$$\omega(\mu) = \begin{cases} \arccos \frac{x}{|\mu|} & \text{if } y \geq 0 \\ 2\pi - \arccos \frac{x}{|\mu|} & \text{if } y < 0 \end{cases} \quad (2)$$

Note that $0 \leq \omega(\mu) < 2\pi$ and that $\omega(\mu) = 0$ corresponds to direction *right*. A motion vector μ is assigned to a bin of a $(K + 1)$-bin motion histogram by the following equation:

$$b(\mu) = \begin{cases} 0 & \text{if } \mu = (0, 0) \\ 1 + \left(\left\lfloor \omega(\mu) \frac{K}{2\pi} + \frac{1}{2} \right\rfloor \bmod K \right) & \text{otherwise} \end{cases} \quad (3)$$

This formula has been chosen such that all motion directions differing from direction *right* ($\omega = 0$) by less than π/K (modulo 2π) are assigned to the same bin (bin 1). The same condition holds for the other main directions *left*, *up*, and *down*, too, if and only if K is a multiple of 4.

Finally, the *motion histogram* of a video frame consisting of R pixels is defined to comprise the following information about bins $b \in \{0, 1, \ldots, K\}$:

- *motion direction* D_b: the relative amount of pixels of the frame being predicted by a motion vector of bin b. More precisely, if P_b denotes the set of macroblock partitions predicted by motion vectors of bin b, and if $|p|$ denotes the size of partition p in pixels, then

$$D_b = \frac{1}{R} \sum_{p \in P_b} |p| \quad (4)$$

- *motion intensity* I_b: the median length of all motion vectors of bin b. If $D_b = 0$, then I_b is defined to be 0. Note that I_0 is always 0.

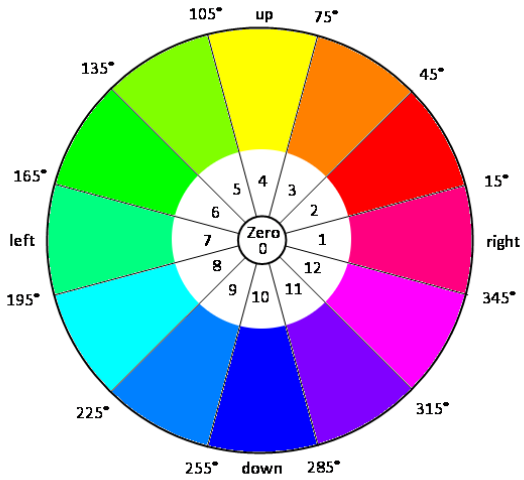

Fig. 1. Motion vector classification for a motion histogram with 13 bins ($K = 12$).

If D_b and I_b are represented by 16-bit data types each, the minimal storage requirement of motion histograms amounts to $4(K + 1) - 2$ bytes per frame, as I_0 does not need to be stored. For our user study described in section 5 we used motion histograms with 13 bins ($K = 12$) as a feasible compromise between desired level of detail and required storage space. In this case, motion histograms require 50 bytes per frame, that is, 4.29 MB for a one hour video with 25 fps.

4. MOTION VISUALIZATION

The motion histograms defined in the previous section are used for a novel and powerful visualization scheme of video motion based on the HSV (Hue, Saturation, Value) color space [7].

For every video frame a fixed-length vertical line L is painted, resulting in a colorized rectangular diagram of the video sequence. Each line L is composed of $K+1$ line segments L_b corresponding to the bins of the frame's motion histogram. The line segment length, denoted by $|L_b|$, and its HSV color are defined as follows:

$$|L_b| = D_b \cdot |L| \quad (5)$$

$$H = \left((b-1)\cdot\frac{360}{K}+330\right) \bmod 360 \quad (6)$$

$$S = \begin{cases} 0 & M=0 \text{ or } b=0 \\ 100 & \text{otherwise} \end{cases} \quad (7)$$

$$V = \begin{cases} 100 & M=0 \text{ or } b=0 \\ I_b \cdot \frac{100}{M} & \text{otherwise} \end{cases} \quad (8)$$

where $M = \max\{I_b \mid b = 0, 1, \ldots, K\}$.

According to the definition of $|L_b|$, motion direction bins covered by the majority of motion vectors of a frame will also visibly dominate the line L representing the frame. Motion direction bins are represented by different colors (H values), and motion intensity is mapped to the brightness (V values) of line segments L_b. However, no motion (bin 0) is represented by white ($S = 0, V = 100$). The color mapping has been chosen such that if K is a multiple of 4, directions *up* and *down* map to simple color names (yellow and blue, respectively, see Figure 1), but any other rotated view of the hue circle would be appropriate, too.

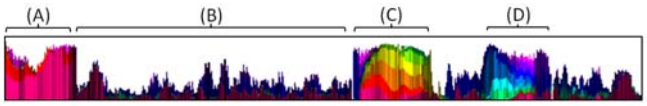

Fig. 2. Visualization ($K = 12$) applied to 25 seconds of a news video. The numbers denote scenes as follows: (A) fast approaching car from the right, (B) an interview (with small amount of slow motion), (C) fast zoom originating from right above, (D) fast zoom originating from bottom left

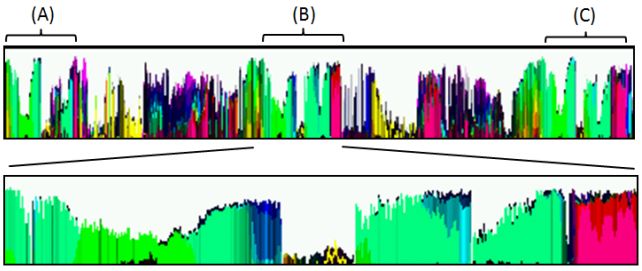

Fig. 3. Visualization ($K = 12$) applied to 150 seconds of a ski jumping competition video. The jumps of three athletes are shown at (A), (B), and (C).

This visualization scheme results in a motion plot which has several nice characteristics with respect to a particular scene of the video sequence. For qualitative analysis or comparison it is immediately visible to a user:

(a) how much motion a particular scene contains at all (length of L_0 - a fully white bar means no motion);

(b) how much motion a particular scene contains in a particular direction (length and color of L_b); and

(c) how intensive (fast or slow) the motion for a particular scene is (overall brightness of L).

Note that the property (b) also allows a user to identify video segments exhibiting motion in all/several directions, which usually constitutes a *zoom/pan* camera operation. The user can also immediately see if the zoom happens *fast* or *slow* (by brightness). Moreover, a user is able to quickly identify scenes with similar object motion due to similar visualization, even without fully understanding the visualization scheme. The visualization example shown in Figure 2 shows a camera-pan, two zooms and a few parts containing only little and slow motion of a news video.

For the purpose of video browsing we propose to show the motion visualization in two combined diagrams: an *overview diagram* with low-details and a *detailed diagram* with higher details according to a user-defined *zoom window*. The user can select position and size of the zoom window in the overview diagram. Both diagrams allow interactive navigation through a video in a way that a mouse click immediately starts playback from the corresponding position. This idea, which is described in more detail in [8], is demonstrated in Figure 3 where an overview for 150 seconds is shown above and a detailed view for a contained segment of 15 seconds is shown below. The overview visualization has been down-scaled to the required width using a nearest neighbor algorithm. The figure visualizes motion in a ski jumping video for a segment containing three jumps of competitors. The jumps from the hill are visualized as greenish "V"-like patterns (A,B,C in the figure).

5. USER STUDY AND EVALUATION

A user study has been conducted to evaluate the performance of the video motion visualization, when used for the purpose of searching. 16 subjects have been tested, 10 men and 6 women. The average age of the subjects was 28.69; the youngest subject was 24, the oldest one was 40. All subjects characterized themselves to have *good computer skills*, 14 of them were computer science students. Two search tasks had to be done, one with a standard video player (VLC v0.9.2) and the other one with our own tool. Both tasks were executed with two different videos, whereas both the chosen tool and the chosen video have been permutated with the number of subjects with a Latin square principle to avoid familiarization effects. Before starting a task a subject got an introduction into the usage of the related tool by watching a three-minutes

tutorial video. After that, we showed the subject a scene of interest and told him/her that (s)he has to find all the other similar scenes in the video, whereas the number of such scenes were also told. Thus, when our tool has been chosen for a task a subject also knew how the scene of interest is visualized in the motion visualization panel directly shown below the video panel. Our implementation used two motion panels, an overview panel and a detailed panel for a user-defined zoom window, as described at the end of the previous section (compare to Figure 3).

We used the following two test videos: (1) a 92 minutes long evening show (Austrian version of *"Who Wants To Be a Millionaire"*); with four scenes of interest (which are *"ask the audience*" lifelines) with an average length of 32 seconds and an average distance of 22 minutes and 54 seconds, (2) a 23.5 minutes long recording of a *ski jumping* competition with 17 scenes of interest (which are jumps of athletes) with an average scene length of 22 seconds and an average distance of 40 seconds. The tests have been performed on Windows Vista 32 Bit SP1 with a screen resolution of 1280x1024 pixels. The H.264/AVC encoded videos had a resolution of 512x288 pixels; a time limit of 10 minutes has been used for each task.

The user study has clearly shown that a user is able to remember motion visualization of a particular scene and find similar scenes with it (i.e. similar visualizations), also if visualization of a similar scenes differs marginally.

In comparison to the VLC player, users of our video browsing tool were able to solve the search task 3.72 times faster for video-1 (94 secs vs. 351 secs), which is only 26.78 percent of the time required by VLC users. For video-2, users of our tool were only 1.14 times faster than VLC users (189 secs vs. 215 secs), i.e. in 87.9 percent of the time required by VLC users. However, it should be noted that the second video can be regarded as nearly a best-case for browsing with VCR-like navigation features, since searched scenes are very close to each other (i.e. scene $k+1$ starts 40 seconds to the end of scene k, in average). Thus, a fast-forward with highest possible speed (e.g. 8x) in the VLC player would be a strategy to find all 17 searched scenes in about 176.25 seconds (in fact, several subjects used that search strategy). Moreover, due to the high number of searched scenes and relative short duration of the whole video (26.5 percent of the content was part of a searched scene), random hits in video-2 were more probable than in video-1 (only 2.34 percent). The short duration of the entire video is also advantageous for browsing with VLC due to the resolution of the seeker bar.

The user study has, thus, demonstrated that even for video sequences especially well suited for player-like navigation, a user can be faster with our tool. In the usual case, as indicated by the first test scenario, the gain is significant. A comparison with more advanced video browsing tools and in different application domains is subject of further study.

6. CONCLUSIONS

We have presented a novel motion visualization method that uses motion histograms transformed to the HSV color space. Our scheme considers both intensity and direction of motion and visualizes that information in a compact form. It is both detailed enough to pertain similarity of similar scenes in visualization and compact enough to keep it simple and understandable to humans. Thus, it can be used for video browsing tools as visual navigation means helping users to quickly find scenes with specific motion characteristics (e.g. zooms/pans) or similar scenes due to similar visualization. Our user study has shown that a user can find scenes that (s)he searched for significantly faster for given scenarios than by using a usual video player and linear search.

7. REFERENCES

[1] B. Adams, S. Greenhill, and S. Venkatesh, "Temporal semantic compression for video browsing," in *Proc. of the 13th Int. Conference on Intelligent User Interfaces*. ACM New York, NY, USA, 2008, pp. 293–296.

[2] L. Chen, G.C. Chen, C.Z. Xu, J. March, and S. Benford, "EmoPlayer: A media player for video clips with affective annotations," *Interacting with Computers*, vol. 20, no. 1, pp. 17–28, 2008.

[3] R. Villa, N. Gildea, and J.M. Jose, "FacetBrowser: A User Interface for Complex Search Tasks," in *Proc. of the 16th Annual ACM Int. Conference on Multimedia 2008*. ACM Press, Vancouver, British Columbia, Canada, 2008, pp. 489 – 498.

[4] M. Campanella, R. Leonardi, and P. Migliorati, "An intuitive graphic environment for navigation and classification of multimedia documents," in *Proc. of the 2005 IEEE International Conference on Multimedia and Expo, ICME 2005, July 6-9, 2005, Amsterdam, The Netherlands*. 2005, pp. 743–746, IEEE.

[5] C.W. Su, H.Y.M. Liao, and K.C. Fan, "A motion-flow-based fast video retrieval system," in *Proc. of the 7th ACM SIGMM Int. Workshop on Multimedia information retrieval*. ACM New York, NY, USA, 2005, pp. 105–112.

[6] M. Holliman and YK Chen, "MPEG Decoding Workload Characterization," in *Proc. of Workshop on Computer Architecture Evaluation using Commercial Workloads*, 2003, pp. 23–34.

[7] Alvy Ray Smith, "Color gamut transform pairs," *SIGGRAPH Comput. Graph.*, vol. 12, no. 3, pp. 12–19, 1978.

[8] K. Schoeffmann and L. Boeszoermenyi, "Video Browsing Using Interactive Navigation Summaries," *7th International Workshop on Content-Based Multimedia Indexing, CBMI 2009*, June 2009.