Frame Hierarchy: A Framework for Level-of-Detail Video Visualization

Neta Sokolovsky Jihad El-Sana Michael Elhadad

Department of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel

Abstract

The advances in video acquisition and storage technologies have increased the quantity and size of the generated video data for various applications. The video data is usually processed by human being as tedious time-consuming task. In this paper we are presenting a novel approach to generate a multiresolution representation of a video data – frame hierarchy – that provides a level-of-detail visualization in an adaptive manner. Frame hierarchy is constructed off-line from the original video data based on a given visual distance metric. At real-time the generated data structure is used to guide the selection of an appropriate level-of-detail representation of the video data based on various control parameters.

Frame hierarchy is a general framework for level-of-detail video visualization that supports various navigation patterns and extends the available video visualization techniques. It enables passive visualization as well as interactive visualization where the extracted levels of detail change interactively to comply with the visual error and the user requests.

1. Introduction

The advances in video acquisition and storage technologies have boosted the availability and accessibility of video data in various applications such as entertainment, security, and remote monitoring. The generated video data is usually processed by human being and probably forms one of the most time-consuming tasks. With the notified exception of entertainment, users would like to reduce or even eliminate the expensive human processing time especially for applications such as security and monitoring video data. For that reason, automatic video processing has attracted the interests of researchers over the last decades. A rich collection of algorithms and techniques to analyze pictorial data and extract various statistical indicators have been developed. However, there is a general lack of effective techniques to convey complex statistical information intuitively [7]. In addition, most of these techniques are not reliable and robust enough to replace the human user in making decision based on the available video data.

In this paper we are presenting a novel approach to generate a multiresolution representation of a video data – *frame hierarchy* – that provides a level-of-detail visualization in an adaptive manner. This multiresolution hierarchy is generated off-line from the original video data based on a given visual distance metric. Then, the generated frame hierarchy is used at real-time to guide the selection of an appropriate level-of-detail representation of the video data based on various control parameters such as visualization time and visual error tolerance.

Frame hierarchy is a general framework for level-of-detail video visualization that supports various navigation patterns and extends the available video visualization techniques. In addition, it enables passive visualization as well as interactive visualization where the extracted levels of detail change interactively to comply with the visual error and the user requests.

In the rest of this paper we briefly review related work on video processing focusing on video segmentation and multiresolution hierarchy. Then, we introduce our approach for level-of-detail video visualization followed by discussing our current implementation and experimental results. Finally we draw some conclusions and suggest directions for future work.

2. Related Work

In this section we shall briefly review related work in video processing, image hierarchy, and level-of-detail visualization in general.

In the context of computer graphics multiresolution hierarchy for level-of-detail rendering has been used to allow various levels of detail to smoothly coexist over different regions of the same surface [5, 9].

Related work on the field of image organization include dimensional reduction and image clustering based on the visual features of the input images [4, 12]. Chen *et al*. [3] and Barnard *et al*. [1] have grouped images hierarchically based on features extracted from these images. Benitez and Chang [2] have developed methods to organize and browse annotated images using multimedia networks that represent knowledge about the images.

The field of automatic segmentation or shot detection for video data involves the detection of cut, fade/dissolve, and camera operation. Shot detection is a basic building block and an important semantic unit

of a video data. We refer the interested reader to the review by Lèfevre *et al.* [11]. However, the statistical results provided by the automatic video segmentation are not easily comprehensible and may require sequential viewing. In addition, the readability and robustness of these approaches are not guaranteed for all different cases without manual calibration [7].

To utilize the human factor several approaches have been developed to obtain an overview of the video by extracting important features and information. Yeo and Yeung [13] have used a browsing technique for viewing a video as flipping through a book. Hertzmann and Perlin [8] and Klein *et al*. [10] have used volume rendering approaches to create artistic visual effects from video data.

Gareth and Chen [7] have presented a methodology for capturing videos and extracting features using volume visualization techniques. A video dataset is composed of a series of 2D images and thus can be considered as a volume dataset. They have examined several image comparison metrics for computing relative and absolute differences between video frames.

Finkelstein *el al*. [6] have presented the representation for time-varying image data called multiresolution video. This representation allows varying spatial and temporal resolutions in different parts of video. Thus, the user can view the video in arbitrary image resolution and speed. As well, multiresolution video provides a control lossy compression of video data.

3. Our Approach

A video stream is typically modeled as a finite-length sequence of synchronized still images and audio. We usually refer to these still images as frames which are uniformly distributed along the time axis. The term *frame rate* is used to define the number of frames displayed over one second which is usually constant for video data. We define the *frame resolution* for a video data *V* as the number of frames used to describe *V*.

The idea of level-of-detail visualization has been used in the fields of computer graphics and image processing in various contexts. It provides a mechanism to select different instances/levels of the visualized dataset. These levels form a hierarchical structure and differ in the amount of information they provide, the memory size, and the computation power they demand.

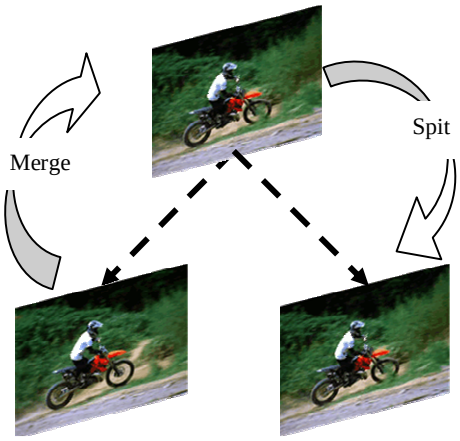

Fig. 1: Frame merge and frame split operations.

In the context of video visualization levels of detail provide a novel mechanism to select appropriate samples of the given video segment in an adaptive fashion. The level-of-detail hierarchy is constructed in an off-line process, and in real-time this hierarchy is used to guide the selection of the appropriate levels of detail in an adaptive manner.

3.1. Levels of Detail Construction

Reducing the time required to examine a video segment is the main motivation for using levels of detail for video visualization. The idea of level-of-detail video visualization is based on reducing the number of visualized frames while keeping the visual information as close as possible to the origin. The construction of these levels of detail is achieved by simplifying the video data. The simplification of a video data *V* is performed by reducing the number of frames representing *V* which corresponds to decreasing the frame resolution. One could view the simplification process as resampling the video data. We are interested in performing video simplification/sampling in an adaptive manner to reduce the visual effect of the introduced error. To enable dynamic change of the visualized level of detail we need to generate a multiresolution hierarchy that encodes the various levels of detail in an adaptive manner. We shall refer to this multiresolution hierarchy as *frame hierarchy*.

The visual distance between two images is defined as their difference with respect to a given metric. Measuring the distance in the image space has attracted the interest of researchers over the last decades for various applications. Nevertheless, the ultimate image-distance metric has not been found. The main reason of the absence of such metric is that different applications tend to view the distances between images differently.

The different levels of detail are encoded and stored in a frame hierarchy which is constructed bottom-up in an off-line process. The construction algorithm starts by computing the visual distance between adjacent frames and storing these distances in a heap. Then, at each stage it performs the following steps:

1. Extract the shortest distance from the heap.
2. Replace the two selected frames by a new frame which could be one of the two frames or a weighted blending or montage of these frames.
3. The newly created frame forms the parent of the two selected frames in the frame hierarchy. Currently we use the first frame as the parent.
4. The distances between the new frame and its adjacent frames are computed and the heap data structure is updated accordingly.

The construction is completed when (a) the heap is empty, (b) the frame's number has reached a predefined threshold, or (c) the minimal visual distance has exceeded a predefined threshold. The levels which are close to the leaves of the trees correspond to high resolution in terms of frame/second, and the levels which are close to the roots correspond to low resolution. The constructed frame hierarchy is a general framework that encodes all the levels of detail of the given video data in a resolution up to one frame.

3.2. Real-Time Visualization

The constructed frame hierarchy is used in real-time visualization to guide the selection of the important frames with respect to the pre-computed visual distance metric, visualization parameters, and time constraints. A set of frames that form a breadth cut on the frame hierarchy defines a sequence of images which provide a valid level of detail. We shall refer to this set of frames as the active frame list (see Fig. 2).

Frame hierarchy allows several visualization patterns to process video data: uniform, visual error driven, and dynamic. A uniform pattern is defined based on the user time constrains and the selected active frames have the same distance from their roots; a visual error based pattern selects the frames which have similar visual error; and a dynamic pattern allows the user to change the frame resolution in real-time.

The visualization starts with the roots of the forest as the active list. Playing the video corresponds to displaying the active frames one by one. The selected level of detail is updated by the frame merge/split operations which decrease/increase the frame resolution respectively (see Fig. 1). The frame merge operation is performed by replacing the two children frames by their common parent frame. The split operation is performed by replacing the active frame by its two children frames in the frame hierarchy.

3.3. Interactive vs. Non-Interactive

Visualizing a level-of-detail video data could be done in two modes – passive and interactive. In the passive mode the user provides the initial constraints which include the time interval, the error value, and/or the frame change behavior. Then, she/he sets back and watches the playing video which is generated based on the user parameters. In the interactive mode the user provides the initial constraints, but she/he can increase or decrease the active level of detail at each frame in real-time.

Interactive visualization allows the change of level of detail in two ways – major and minor. The changes on the major level of detail maintain the same level of detail for the consecutive frames until the next change on the major level of detail. In contrast, the changes on the minor level of detail vanish after few frames and the selected level of detail of the consecutive frames returns to follow the major level of detail.

3.4. Extracting Appropriate Frames

For a given frame hierarchy and a set of user constraints, we would like to select the appropriate level of detail – the set of frames that provides the best approximation. We shall refer to the time constrains as the frame budget since we do not change the frame duration. Note that a node n on the frame hierarchy represents frames that correspond to the descendants of n. We would like n to represent its descendants faithfully. Therefore, we minimize the maximal visual distance between the frames and their representative.

We select the appropriate frames as the set of nodes which have similar maximal visual distance from the nodes in their sub-tree. In real-time visualization the visual error is computed to determine if there is a need to increase or reduce the resolution by going down or up the tree respectively.

3.5. Estimating Frame Budget

To accurately estimate the play time of video data, it is necessary to determine the appropriate level of detail and compute its size. However, it is often required to estimate the play time dynamically in real time.

We have developed an algorithm that traverses the frame hierarchy and stores the remaining time to complete the video data at each possible level of detail. The algorithm starts from the last root and traverses the frame hierarchy right-to-left inorder manner. In such traversal the algorithm sweeps through the hierarchy from right to left (analogous to backward in the frames order). The algorithm calculates the appropriate number of remaining nodes for each node. The remaining-nodes counters are used at real time to quickly identify the selected level of detail. The algorithm determines the first frame of the appropriate level of detail by starting from the first root and inorder traversing the frame hierarchy until it reaches the node which remaining-nodes counter matches the visualization time budget. In addition, the algorithm determines the remaining time at each node during passive as well as interactive traversal.

4. Implementation and Results

We have implemented our approach in C++ over windows XP and adapt Microsoft DirectShow [14] for video processing and playing. Our proto-type system consists of two parts – the off-line preprocessing that constructs the frame hierarchy and a real-time video visualization system.

We have implemented several visual metrics which are based on various image-distance algorithms. Most of these algorithms have been developed for shot detection and video segmentation (for detail refer to [11]). These algorithms could be divided into the three categories: pixel-to-pixel relation – measure the distance between the corresponding pixels in the images; block-based relation – divide the images into blocks

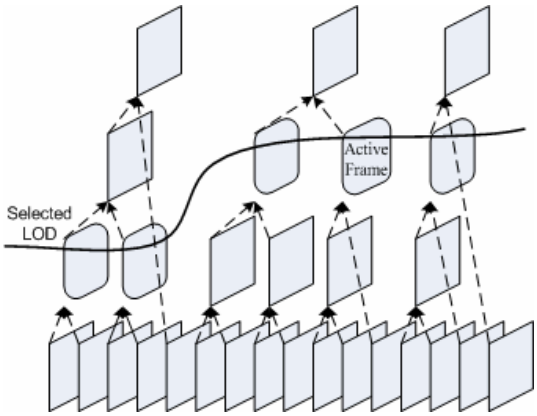

Fig. 2: A selected level of detail: active frames appear with rounded corners.

and perform a comparison between them; and histogram relation – use the histograms of the two images to derive a distance between the images.

We are currently using DirectShow [14] from Microsoft to extract the frames from AVI movies. The extraction in a non-sequential manner seems to take most of the running time.

Table 1 shows the preprocessing time of two video segments for three types of visual metrics. Video1 is an outdoor video data, and Video2 is an indoor video data. The running time of our off-line algorithm to construct the frame hierarchy is usually proportional to the number of frames in the video data. The running time mainly depends on the frame extraction and the visual error metric – the time for simple pixel-to-pixel comparison is relatively faster than blocks or histogram based metrics.

Table 1: Frame hierarchy construction time (off-line).

Dataset	Frame Count	Frame Resolution	Metric Name	Time minute
Video1	730	640x480	Pixels	16.1
Video1	730	640x480	Blocks	18.9
Video2	960	320x240	Pixels	17.4
Video2	960	320x240	Histogram	20.8

We have also found that the quality of the video data, kind of data in terms of the images, and the relation between them determine the appropriate visual metric for frame hierarchy representation. At real-time the frame extraction is not expensive as in the preprocessing stage since we practically perform a smart frame skipping. In addition, we pre-fetch frames ahead of time to maintain adequate frame rates.

5. Conclusions and Future Work

We have presented a novel multiresolution hierarchy – frame hierarchy – for video visualization. It enables short time processing to extract features and review video data based on the coherence among frames, review time, and user specification. The frame hierarchy is constructed off-line and used in real-time to guide the selection of the appropriate frames with respect to a given visual constraints. The system allows interactive as well as passive visualization of a video data. The interactive visualization allows users to increase or reduce the selected level of detail in real-time.

The scope of future work includes implementation and testing more visual metrics and trying to categorize them for the different types of video data. In addition, we hope that incorporating the audio data in the frame hierarchy could improve the results.

We believe that most of the previous video visualization algorithms and techniques could be incorporated in our framework. The recent work by Gareth and Chen [7] could be easily integrated at any selected level of detail to reduce the processing time and improve the visualization process. In addition, frame hierarchy could be used as the infrastructure to develop editing and real-time video visualization tools.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. I. Jordan. Matching words and pictures. Submitted to JMLR.

[2] A. B. Benitez and S. F. Chang. Multiresolution organization and browsing of images using multimedia knowledge networks. *ADVENT Technical Report #007 Columbia University*, 2003.

[3] J. Chen, C. A. Bouman, and J. C. Dalton. Hierarchical browsing and search of large image databases. In *IEEE Trans. On Image Processing* Vol. 9, No. 3, 2000.

[4] S. Craver, B. L. Yeo, and M. Yeung. Multilinearization data structure for image browsing. *IS&T/SPIE-1999*, Vol. 3656, pp. 155-166, 1999.

[5] J. El-Sana and A. Varshney. Generalized view- dependent simplification. *Computer Graphics Foru*m, 18(3), pp. 83– 94, 1999.

[6] A. Finkelstein, C. E. Jacobs and D. H. Salesin. Multiresolution video. In *Proceedings of SIGGRAPH 96*, pp. 281-290, 1996.

[7] D. Gareth and M. Chen. Video Visualization. In *IEEE Visualization*, pp. 409– 416, 2003.

[8] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. In *Proceedings 1st International Symposium on Non-Photorealistic Animation and Rendering*, pp. 7-12, 2000.

[9] H. Hoppe. "Progressive meshes". In *Proceedings of SIGGRAPH '96*, pp. 99-108, 1996.

[10] A. W. Klein, P. J. Sloan, A. Finkelstein, and M. F. Cohen. Stylized video cubes. In *Proceedings ACM SIGGRAPH Symposium on Computer Animation*, pp. 15-22, 2002.

[11] S. Lefèvre, J. Holler, and N. Vincent. A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. *RFAI publication: Real Time Imaging*, to appear.

[12] Y. Rubner, L. J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. *DARPA Image Under-standing Workshop*, pp. 661-668, 1997.

[13] B. L. Yeo and M. Yeung. Retrieving and visualizing video. In *Communication of the ACM* 40, 12, pp. 43-52, 1997.

[14] http://msdn.microdoft.com/library. DirectShow: Media streaming architecture for Windows.