Newcastle University

COMPUTING SCIENCE

Panopticon: A Parallel Video Overview Technique

Dan Jackson and Patrick Olivier

TECHNICAL REPORT SERIES

No. CS-TR-1362 December 2012

TECHNICAL REPORT SERIES

No. CS-TR-1362 December, 2012

Panopticon: A Parallel Video Overview Technique

D. Jackson and P. Olivier

Abstract

Panopticon is a novel method of arranging a moving video sequence that displays multiple sub-sequences in parallel. The technique provides a consistent spatio-temporal layout for context, while presenting a rapid overview of the entire sequence to the user, and yet does so in a way that allows any sub-sequence to be followed without interruption. Furthermore, this output can be generated offline as a highly efficient repeated animation loop. Panopticon can be used directly on source video as a new form of video abstract. Moreover it could be used to augment existing video skims with additional context.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

JACKSON, D., OLIVIER, P.

Panopticon: A Parallel Video Overview Technique
[By] D. Jackson, P. Olivier

Newcastle upon Tyne: Newcastle University: Computing Science, 2012.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1362)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1362

Abstract

Panopticon is a novel method of arranging a moving video sequence that displays multiple sub-sequences in parallel. The technique provides a consistent spatio-temporal layout for context, while presenting a rapid overview of the entire sequence to the user, and yet does so in a way that allows any sub-sequence to be followed without interruption. Furthermore, this output can be generated offline as a highly efficient repeated animation loop. Panopticon can be used directly on source video as a new form of video abstract. Moreover it could be used to augment existing video skims with additional context.

About the authors

Dan Jackson works as a Senior Research Associate for the School of Computing Science at Newcastle University. He studied an MEng Computer Systems and Software Engineering Degree with the Department of Computer Science at The University of York.

Patrick Olivier is based in Culture Lab and leads the Cultural Technologies theme in which computer scientists and electronic engineers are engaging with researchers in the arts and humanities both in the development of new technology and research methods. He is also a member of the Informatics Research Institute and the Institute of Ageing and Health. His research interests spans aspects of human computer interaction (especially for pervasive computing and computing of older people and the cognitively impaired), computer graphics, and artificial intelligence. Patrick has an undergraduate degree in Physics (Natural Sciences, Cantab, 1989) and a PhD is in computational linguistics (UMIST, 1998). Prior to coming to Newcastle (2004), he was a lecturer at the University of Wales, Aberystywth and the University of York. In York he founded Lexicle Limited where he led the development of the world's first commercial 3D embodied conversational agent.

Suggested keywords

PARALLEL VIDEO
VIDEO ABSTRACTION
VIDEO SUMMARIZATION
VIDEO BROWSING

Panopticon: A Parallel Video Overview Technique

Dan Jackson, Patrick Olivier
{dan.jackson, patrick.olivier}@ncl.ac.uk
School of Computing Science, Newcastle University, UK.

ABSTRACT

Panopticon is a novel method of arranging a moving video sequence that displays multiple sub-sequences in parallel. The technique provides a consistent spatio-temporal layout for context, while presenting a rapid overview of the entire sequence to the user, and yet does so in a way that allows any sub-sequence to be followed without interruption. Furthermore, this output can be generated offline as a highly efficient repeated animation loop. Panopticon can be used directly on source video as a new form of video abstract. Moreover it could be used to augment existing video skims with additional context.

Keywords

Parallel video, video abstraction, video summarization, video browsing.

INTRODUCTION AND BACKGROUND

The volume of digital video data has grown dramatically in recent years. By the end of 2012 video will account for half of consumer Internet traffic and, by 2016, 1.2 million minutes of video is forecast to transfer over the Internet every second [9]. Fundamentally, video is a collection of image frames which, when played, are displayed sequentially at a specified frame rate. Achieving good perception of motion typically requires 24-30 frames per second, with the result that one hour of footage can contain 100,000 frames. Supporting applications such as video browsing [2], selection and editing within this information-dense resource requires interface technologies that provide efficient representation of video content. The ability to present a visual overview of video data that can be rapidly comprehended is of particular importance.

Video abstraction techniques generate a new, shorter, sequence of frames, while attempting to maximize the relevant information preserved. Typically, these abstractions are driven from automatic key frame identification, with videos often segmented into scenes, shots and frames [8]. There are two primary types of presentation for video abstracts: moving and non-moving [4]. In a moving abstract, or *video skim*, image sequences are chosen to produce a set of highlights, or a summary sequence is generated to give an impression of the whole video [6]. In a non-moving abstract, or *video summary*, a storyboard is produced from a set of salient images which are either directly extracted from the source video, or are synthesized aggregate mosaics. The storyboard is typically presented in a grid layout with time increasing (not

Figure 1. (a) A simple web-based Panopticon player (left), (b) focus+context Panopticon viewing application (right).

necessarily uniformly) left-to-right, top-to-bottom, allowing the user to see the representative frames simultaneously while maintaining their temporal relationship. In this way, storyboards are able to provide users with a rapidly comprehendible context, but at the expense of losing the motion information from the video.

Video editing packages (for example, Final Cut Pro X [1]) use a simple version of a static storyboard summary: a linear filmstrip layout of uniformly-sampled frames, increasing in time from left to right. Each cell is generally rendered as a smaller, "thumbnailed", representation of the original video frame, and this interface provides an overview of a video section while supporting rapid seeking within it. However, a static presentation of images gives a poor overview of action within a video sequence – salient frames could be omitted and important information from the animation may be missing.

This paper discusses the concept of animating video sub-sequences in parallel, and introduces *Panopticon*, a novel method of arranging parallel video to provide a consistent spatio-temporal layout that presents a rapid overview of the entire sequence to the user in a way that allows any sub-sequence to be followed without interruption. The technique provides a new form of video abstract when used directly on video, and could be used to augment existing video skims with additional context.

PARALLEL PRESENTATION OF VIDEO

Displaying frames from a video in parallel is a powerful method of providing the user with a rapid overview of the video and, with an appropriate spatial layout, it maintains the context of each part of the video. To overcome the limitations of conventional static layouts, each of the cells may be animated by advancing the frame indices presented to the user over time. An example of a how such a dynamic presentation of a linear layout changes over time is shown in Figure 2.

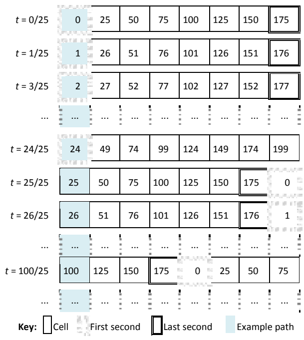

Figure 2 An example dynamic linear layout; video frame indices shown at time, *t* (25 fps, 1s cell interval, 8s duration). The example path through the video is uninterrupted, but the beginning and end of the sequence move around.

As time progresses, the video shown in a given cell will catch up to the time index originally displayed in the cell to the right (in the above example, this will occur after 1 second). So that an observer may continue to follow one cell of video without interruption, the video index within each cell can continue on in sequence (an example path is indicated). However, with this approach, the initial temporal layout, with the start of the video at the left of the layout and the end at the right, is then lost.

Alternatively, at this "wraparound" point, each cell can be reset to its original frame index to preserve the temporal layout (see Figure 3). However, this presents the observer with a spatial discontinuity if they were following an individual cell – they have to suddenly transfer their attention to the adjacent cell (e.g. the example path is discontinuous at the 25th frame).

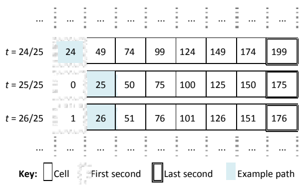

Figure 3 The example dynamic 1D layout restarting; video frame indices shown at time, *t* (25 fps, 1s cell interval, 8s duration). The beginning and end of the sequence remain still, but the example path through the video is interrupted.

Panopticon provides a layout technique which allows both uninterrupted viewing of a sub-sequence and consistent temporal layout. Moreover, a class of layouts is suitable for efficient offline computation.

PANOPTICON

Panopticon presents multiple parts of a video sequence in parallel while overcoming the limitations of simple parallel video layouts. The key concept lies in recognizing that the spatial layout that gives context to these animated storyboards can be applied as a continuous function of time – mapping each frame index from the video to a displayed position and, potentially, other rendering attributes.

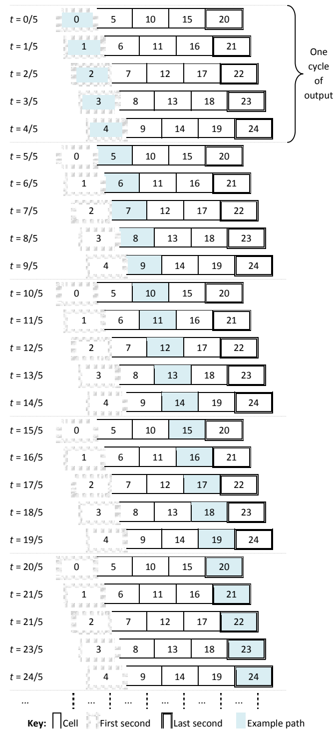

Figure 4 Panopticon uniform-interval 1D layout; frame indices shown at time, *t* (5 fps, 1s cell interval). An example, uninterrupted path through the 5 repeated outputs is shaded.

To create the output, (a *Panopticon*), a set of cells are used, each at a different index within the video. The cells are animated with time: each is updated so that it displays the appropriate frame from the video sequence, and each has the transform function applied to determine where, and how, to display the video frame. Where the transform function is chosen to produce spatial-continuity in its output, the scheme allows an observer to easily follow any sub-sequence of the video.

Let us consider a simple Panopticon with a uniform-interval linear layout, where a cell's horizontal position is proportional to the input video time, the cells are sampled at a regular interval, and the interval and output scaling are chosen to avoid overlap. Figure 4 illustrates an example time sequence. The spatio-temporal consistency is clear: the first few frames are always on the left, the last few are always on the right. As the cells are displayed in parallel, after one second the user has been shown an overview of the whole video. Meanwhile, it can be seen that any sub-sequence can be visually followed by the user without interruption – even the entire 25-frame sequence (path highlighted). In this way, the limitations of simple parallel video layouts have been overcome.

Looped output

The transform function has the effect of adjusting the position of each of the cells so that, by the time they reach the frame index at which an adjacent cell started displaying its content, they have gradually moved to the same location that the adjacent cell started from. Figure 4 demonstrates that the output from the example sequence repeats every five updates: the user has been shown a complete overview of all 25 frames after just one second, and will be shown the same overview repeatedly. This follows from the fact that the set of cell time values repeat over a shorter interval than the original video sequence and, as the transformation function produces values that are only time-dependant, the resulting output will exactly repeat over this interval.

Importantly, this single cycle of five different outputs represents exactly all that is required to be shown to the user. This could be computed once, stored as a new summary video (of the length of the interval), and repeatedly displayed in an animation loop. Such computation could be performed offline for a video, which would then allow efficient access to an overview of the video – the Panopticon output can be accessed and played as a single short video loop. It would be of particular benefit in scenarios where bandwidth-limited or other resource-constrained devices were used, where decoding multiple frames of a video in parallel (even if a thumbnail-sized video) may not be feasible.

Extending to two-dimensional layouts

The Panopticon technique may be extended to a two-dimensional layout, similar to that used for storyboards. An example uniform grid layout of cells, temporally distributing the cells left-to-right and top-to-bottom, is illustrated in Figure 5.

t = 0/5

0	5	10	15	20	25	30	35
40	45	50	55	60	65	70	75
80	85	90	95	100	105	110	115
120	125	130	135	140	145	150	155
160	165	170	175	180	185	190	195

t = 1/5

1	6	11	16	21	26	31	36
41	46	51	56	61	66	71	76
81	86	91	96	101	106	111	116
121	126	131	136	141	146	151	156
161	166	171	176	181	186	191	196

t = 2/5

2	7	12	17	22	27	32	37

...

Figure 5 Panopticon using a uniform grid layout: video frame indices (25 fps, 1 second cell interval)

This layout keeps the start of the video in the upper-left, and the end in the lower-right. The observer can always adjust their position in the video by five frames by changing their focus to the left or right of the current cell, and 40 frames by changing their focus above or below the current cell. This transform function is generally spatially continuous, so the observer can watch a sub-sequence at any point, but it does have spatial discontinuities at the left and right sides of the layout. One way to reduce the impact of these is to duplicate the cell disappearing from one row on to the start of the next, allowing the observer more time to adjust their gaze between rows.

Parameter selection

When considering a regularly-sampled layout at fixed output dimensions, there is a relationship between the number of cells and the interval at which they repeat at. A smaller interval reduces the time taken for a complete overview, but will require more cells, and so the cell size will become smaller. Conversely, a larger cell size would require fewer cells, but the interval would have to increase.

In a regularly-sampled linear layout, the time interval between each cell (I_c) is affected by the duration of the source video (T_s) and the number of cells shown (N_x):

$$I_c = T_s \,/\, N_x$$

The cell width (C_w) for a given output width (O_w) must have room for the translation equivalent to the width of one cell, and the cell height is scaled to produce the same aspect ratio as the source (dimensions: $S_w \times S_h$), thus:

$$C_w = O_w \,/\, (N_x + 1)$$

$$C_h = C_w \cdot S_h \,/\, S_w$$

The number of cells would typically be chosen from the set of numbers up to some usable maximum (that produces the smallest practical output) which produces an interval as close as possible to a preferred interval, so as to maximize the cell size for shorter clips.

For two dimensional grid layouts, the method is similar, but the output height (O_h) must be taken into consideration. For each potential number of horizontal cells (N_x) up to the maximum, the height of each cell (C_h) is calculated as above, and the number of vertical cells (N_y) is calculated as:

$$N_y = \lfloor O_h \,/\, C_h \rfloor$$

The interval of the two-dimensional layout is calculated:

$$I_c = T_s \,/\, (N_x \cdot N_y)$$

Finally, the configuration is chosen that gives a cell interval closest to the desired interval.

Interaction

The Panopticon layout can be used in applications to support the user in activities such as browsing, seeking and selecting ranges in video. The overview displays low-resolution video thumbnails, and does not provide any direct support for other modalities of video files (such as speech, subtitles or music) – a user may want to be able to ‘peek’ into the original video at a higher resolution, with the audio track. We suggest two approaches to allow peeking into the video. One approach is to allow the user to directly select a cell from the Panopticon, and provide a larger interactive viewer that plays the video and audio from the selected time, while a ‘playback head’ indicates the currently played position as an overlay on the Panopticon. Figure 1a shows a screenshot of a demonstration application with this behavior.

Another approach also provides detailed video and audio at any point, but does so in-situ while maintaining the surrounding context by distorting the adjoining cells as if seen through a fisheye lens – a *Focus + Context Display* [7]. Figure 1b shows a screenshot of a demonstration application with this behavior.

The ability of the Panopticon technique to support preview, seeking, and range selection, lends itself to a range of potential applications. For example, consumer television viewing could benefit from visual seeking on a digital video recorder. The ability to pre-render the Panopticon allows the potential for fine-grained scene selection, even on resource constrained, read-only technologies such as DVD or Blu-ray. Other applications where quick review of large spans of video information is required, for example, 'life logged' data from a wearable camera, recorded video surveillance footage, or assisting video editors with their initial 'Edit Decision Lists' – first cuts from raw video footage.

CONCLUSION

Motivated by shortcomings in existing methods for providing an overview of video, we have discussed the concept of animating video sub-sequences in parallel, and introduced Panopticon, a novel method of arranging parallel video. This technique has four important benefits by: presenting a rapid overview of the video sequence; providing a consistent spatio-temporal layout; enabling any sub-sequence of the video to be followed without interruption; and allowing the output to be generated as a highly efficient repeated animation loop.

Much work remains in fully exploring this new technique. User studies should be designed that test the assumed benefits of this approach, and could be based on previous assessments of video abstraction [2,3]. Despite the multimodal nature of video data, additional data such as speech, music, and subtitles are not directly represented within the Panopticon. Augmenting the display with techniques such as those used in the Audio Visual Spiral [5] could be investigated.

Our examples have only shown the specific case where there is a regular time interval between each cell, the cells are presented in a single row or uniform grid, and the cells move horizontally. More general uses of the transform function need exploring. For example, it could be used with existing video abstraction techniques to show larger, slower moving cells where they represent more salient images.

The Panopticon technique could even be extended to cover live sources of video, and could provide an overview of the recent past in when watching live television or surveillance footage. So that the most recent image is always displayed, rather than absolute time being shown spatially, 'time relative to now' could be used – at a fixed time offset, the cells would not move.

Panopticon can be used directly on source video as a new form of video abstract. Moreover it could be used to augment existing video skims with additional context.

REFERENCES

1. Apple Final Cut Pro X. http://www.apple.com/finalcutpro/
2. Ding, W. and Marchionini, G. A Study on Video Browsing Strategies. Tech. Report CS-TR-3790, UMIACS-TR-97-40, CLIS-TR-97-06, University of Maryland. 1997.
3. Li, F., Gupta, A., Sanocki, E., He, L., Rui, Y. Browsing Digital Video. In *Proc. ACM CHI 2000*. Hague, ACM Press (2000).
4. Li, Y., Zhang, T., and Tretter, D. An overview of video abstraction techniques. Tech. Rep. HP-2001-191, HP Laboratory. July 2001.
5. Mason, A., Evans, M., and Sheikh, A. Music Information Retrieval in Broadcasting: Some Visual Applications. BBC Research White Paper WHP 166, June 2008.
6. Pfeiffer, S., Lienhart, R., Fischer, S. and Effelsberg, W. Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 345-353, Dec. 1996.
7. Robertson, G.G. and Mackinlay, J.D. The document lens. In *Proc. UIST 1993*, ACM Press (1993), 101-108.
8. Truong, B.T. and Venkatesh, S. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1). (2007) pp. 1-37.
9. The Zettabyte Era. Cisco Visual Networking Index White Paper, 2012. http://www.cisco.com/en/US/solutions/collateral /ns341/ns525/ns537/ns705/ns827 /VNI_Hyperconnectivity_WP.pdf