Stained-Glass Visualization for Highly Condensed Video Summaries

Patrick Chiu, Andreas Girgensohn, Qiong Liu
FX Palo Alto Laboratory, 3400 Hillview Ave., Bldg. 4, Palo Alto, CA 94304, USA
{chiu, andreasg, liu}@fxpal.com

Abstract

This paper presents a method for creating highly condensed video summaries called Stained-Glass visualizations*. These are especially suitable for small displays on mobile devices. A morphological grouping technique is described for finding 3D regions of high activity or motion from a video embedded in x-y-t space. These regions determine areas in the keyframes, which can be subsumed in a more general geometric framework of* germs *and* supports*: germs are the areas of interest, and supports give the context. Algorithms for packing and laying out the germs are provided. Gaps between the germs are filled using a Voronoi-based method. Irregular shapes emerge, and the result looks like stained glass.*

1. Introduction

It is challenging to browse video on small mobile devices such as PDAs and cellphones. The small screen restricts the amount and size of the content that can be displayed. Existing techniques for visualizing video summaries are not designed for small screens and do not work well on them. A popular method is to use a storyboard with keyframes selected from the video using content analysis algorithms (e.g. see [3], [9], [10]). Such layouts may have same or different sized rectangular images. When viewed on a small screen, it can be difficult to see what is in the images (see Fig. 1a).

The idea of *Stained-Glass visualization* is to find regions of interest in the video and to condense the keyframes into a tightly packed layout. The result is that the people and objects in these regions appear larger and become easier to see, as shown in Fig. 1b. The stained-glass effect comes from non-rectangular region boundaries that emerge from a Voronoi-based algorithm for filling the spaces between the packed regions.

2. Creating Stained-Glass Video Summaries

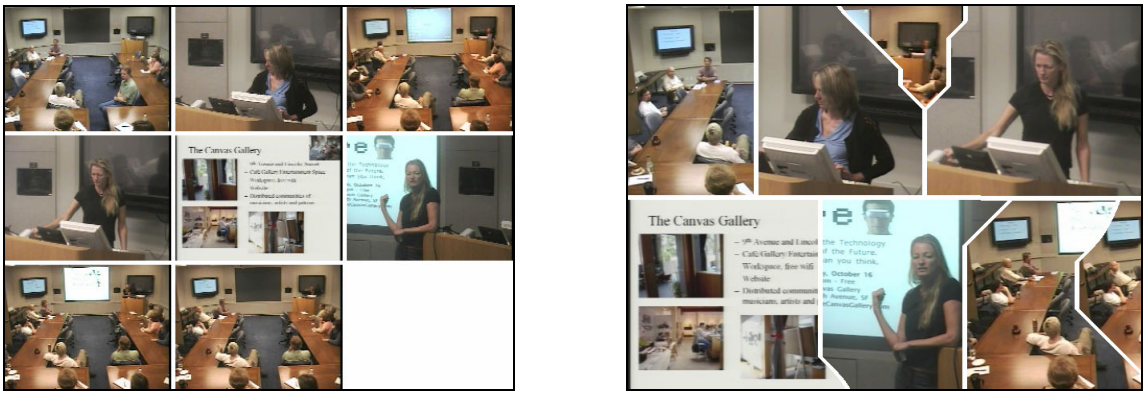
There are several major steps in constructing a stained-glass video summary:

1) Segment the video into video clips.
2) Find regions of interest in the video clips.
3) Layout the regions of high importance.
4) Fill the spaces between regions.

We go into each of these in detail.

2.1. Segment the video into video clips

The video is segmented into video clips so that each clip consists of successive frames that are similar. This can be done using standard techniques such as color histograms [2] or variations of pixel-wise differences [8].



(a) **(b)**

Fig. 1. Two visual summaries of a staff meeting video. Both have the same set of keyframes and bounding boxes. (a) A generic visualization. (b) A Stained-Glass visualization.

Final version. To appear in 2004 IEEE International Conference on Multimedia and Expo (ICME 2004).

Sometimes the camera on/off information is available, providing true shot boundaries that can help produce better clips.

2.2. Find regions of interest in the video clips

A video can be regarded as a 3D volume in *x-y-t* space by stacking the frames along the time axis (Fig. 2). We look for 3D regions with relatively high activity or motion in this space. Each sampled pixel is assigned a velocity. The velocity can be obtained by calculating the change in luminance between video frames.

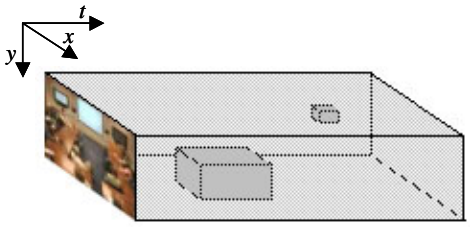


Fig. 2. Video in *x-y-t* space. Dark boxes represent regions of high activity or motion.

For video applications, it is important to use the residual motion obtained by subtracting the motion of the camera or background. There are various types of camera motion such as pan, tilt and zoom, and methods to detect these have been developed (e.g. [6], [8]). As a simple approximation, we subtract off from each pixel's motion the average motion of the pixels in its frame.

Next, for each pixel in *x-y-t* space, the magnitude of the velocity is binarized to values of 0 or 1. A pixel having magnitude above average, with respect to its video segment, is set to 1; otherwise, it is set to 0. We call a pixel with value 1 a *1-pixel*.

The regions are constructed from the 1-pixels. For 2D bitmap images, well-known morphology algorithms have been proposed that take the set of 1-pixels and through erosion and dilation operations produce blobs (e.g. [7]). These blobs can have highly irregular shapes. Computationally, this approach is much more expensive for a 3D volume associated with a video. To keep down the space complexity, we do not progressively change 0-pixels into 1-pixels through erosion and dilation. Instead we perform morphological grouping to grow regions by working with only the initial set of 1-pixels.

During the region growth process and later in the layout step, we also need to compute geometric properties such as intersection and containment of regions. To keep the geometric operations efficient, we use 3D rectangular boxes to represent the regions.

Grouping is performed for each video segment. It starts with a segment's set of 1-pixels, which are just degenerate groups. If two groups are adjacent, they are merged into a larger group provided that they do not satisfy the stopping conditions. The stopping conditions keep the groups from spreading too thin, and are based on density and volume.

The density should not be allowed to decrease too much after a merge. More precisely, let $d(A)$ be the density of a group A, which is the number of 1-pixels in A divided by the total number of pixels contained in the bounding box of A. Let $d(W)$ be the average density of the whole video segment. We merge two groups A and B into C only if $d(C) > d(W)$.

The volume should not expand too much when two groups are merged. Let $v(A)$ be the volume of the bounding box of a group A. Then for groups A and B, we take their intersection K. If $v(K)/v(A) < 1/2$ and $v(K)/v(B) < 1/2$, we do not merge A and B.

The result of the iterative merging process is a forest of trees, where each tree represents a group, and the leaves of the tree are 1-pixels. The trees are not binary; each node can have more than two children. The bounding box of each tree is a region.

2.3. Layout the regions of high importance

At this point, the video has been segmented into clips, and 3D regions have been computed for each clip. In a clip, there is often a single dominant region. This is typically the case for our corpus of staff meeting videos. Examples are: (1) a person at the podium talking and gesturing, (2) people sitting in a section of a room moving around a little bit. For this reason, we take the dominant regions from each clip of the video as the set of regions to be laid out. Associated to each region is a keyframe, which we take to be the first frame of the region's clip.

We now introduce the following terminology:

- ***Germ***: A germ is the intersection of the *x-y* projection of the region and the keyframe.
- ***Support***: A germ's support is an image area that contains the germ.

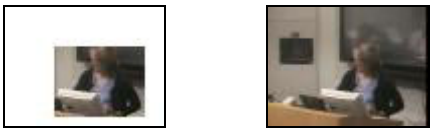


Fig. 3. A germ and its support.

In the simplest case with box-shaped regions, the germ is a sub-rectangle of the keyframe and the germ's support is the whole keyframe (Fig. 3). In general, a germ and its support can be irregularly shaped, and it is possible to extend a germ's support beyond the keyframe's bounds using video mosaic algorithms to create larger panoramas (see [1], [8]). Note that the method in [8] packs whole panoramas and leaves empty gaps between them; in contrast, our approach packs germs that are sub-areas and later use the support to fill in the gaps. Thus, it is crucial to identify the germs and supports.

For the layout algorithm, the input is:

- The germs $G = \{g_i\}$, $0 \le i < n$, ordered by time.
- The corresponding supports $S = \{s_i\}$, $0 \le i < n$.
- The canvas rectangle R to place the layout.

We begin the layout by determining the maximum scale factor for the germs such that they can be packed in rows filling the canvas rectangle. We iteratively adjust the scale factor and find "line breaks" in the ordered set of germs. (This is analogous to laying out a paragraph of text with word-wrap.) The line breaks are chosen so that the packing of the ordered germs, line by line from left to right, results in a packing whose bounding rectangle is closest to the aspect ratio of the canvas rectangle.

Initially, each row height is set to the height of its tallest germ. Additional vertical canvas space is distributed evenly among the rows without making any of the rows taller than the supports in them. Each row is divided into cells containing one germ that are initially as wide as the germs in them. Additional horizontal space is distributed just like the vertical space. Each germ is centered in its cell. Germs might be shifted to allow their support to cover the whole cell. Fig. 4 shows cell borders as dashed lines and illustrates the placement of the germs in the cells. Several of the germs are pushed against the borders because they come from the edges of the keyframes.

Fig. 4. After layout, before fill.

2.4. Fill the spaces between regions

To fill the spaces between regions, we take the Voronoi regions of the germs. The Voronoi regions partition the canvas into disjoint areas corresponding to the germs: a point (x, y) on the canvas belongs to the germ closest to it (d_0 in Fig. 6). If this point belongs to the support of the nearest germ, the pixel is assigned the same value as the pixel of the germ's support. Otherwise, we depart from the Voronoi regions and look for the nearest germ whose support contains the point and use the pixel value from there. If no such germ exists, we assign the pixel the canvas background color.

To delineate the images in the visualization, we put borders around the areas from different images. Fig. 4 shows the placement of these borders in solid lines.

More rounded areas can be generated using different distance functions in the fill process. We consider two functions derived from the distance between a pixel and the center of a germ (d_1 in Fig. 6). To include the size of a germ in the distance function, we construct two circles centered at the center of the germ. The diameter of the first circle is the average of the germ's width and height (radius r_1 in Fig. 6). The diameter of the second circle is the germ's diagonal (radius r_2 in Fig. 6). The first distance function $\max(0, d_1 - r_1)$ generates moderately rounded borders. The second distance function d_1/r_2 generates highly rounded borders (see Fig. 6). These distance functions can be combined in a weighted average to vary the degree of roundedness. Note that the boundaries of the supports still force some straightness in the borders.

3. Preliminary Tests

After developing the algorithms, we tested them on the three most recent video recordings of staff meetings at our lab. In a typical staff meeting, several people would make announcements and give brief presentations (project updates, trip reports, etc.) Slide images are often shown on a large wall display. A human camera operator shoots the video, capturing views of the room, the podium, and the wall display.

For the three videos that we tested, the Stained-Glass summarizations looked qualitatively alike. It did a good job of finding and framing the different speakers (e.g. Fig. 3). For selecting keyframes with a speaker, it did better than our earlier method based on time-constrained color histogram clustering (see [4], [9]). The reason for this is evident in Fig. 4: the 2nd and 4th keyframes show two different persons on the same background but their color histograms are not sufficiently dissimilar.

Our current algorithm did a poor job of finding keyframes with slides, because most slides contain little motion. A simple way to remedy this is to select additional keyframes; e.g. from video segments that are longer than one or two standard deviations from the average. Or one can combine the keyframes selected by motion analysis with those selected by color histogram clustering.

3.1. A computation in detail

Next, we examine the computation for one of the videos in more detail. This video, which is used to

illustrate this paper, is 14:04 in length. Sampling every 0.5 sec, 1,688 frames are extracted. Each frame is further subsampled evenly to obtain a 16×12 array of pixels. The segmentation step produced 20 segments.

In the region finding step, 8 of the 20 segments have above average motion density. On these 8 segments, binarization of the high velocity pixels led to an average of 544 1-pixels per segment, ranging from 106 to 1,722. This is quite sparse in the *x-y-t* space, with an average density of 0.042.

The average segment has 6 non-degenerate groups, and each segment has a dominant group with a substantially larger volume than the rest. Half of the dominant groups contained over 90% of the 1-pixels in its segment (Fig. 5).

The areas of the germs are much smaller than the original keyframes, averaging 34% of the keyframe area. Only 2 of 8 are larger than 50% (Fig. 5).

In the layout step, the germs are scaled up further by a factor of 1.50 to cover all the gaps.

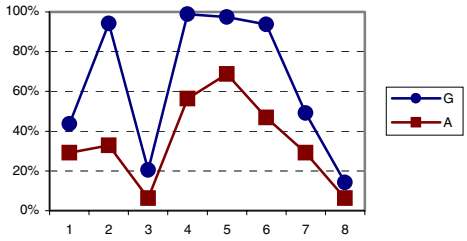


Fig. 5. G is percent of 1-pixels in dominant group. A is percent of keyframe area.

4. Conclusion

We presented a new way to create highly condensed visual summaries of videos. The algorithms in their most basic forms have yielded good results so far, although further refinement and testing on different genres of videos is necessary to make them more robust. While the application that motivated this work is for small displays, these visualizations can help make efficient use of screen real estate for displays of all sizes. We also plan to apply Stained-Glass visualization to images and photo collections.

Acknowledgements. We thank Lynn Wilcox for valuable comments on this work.

5. References

[1] A. Aner and J. Kender, "Video summaries through mosaic-based shot and scene clustering," *Proc. European Conference on Computer Vision,* May 2002.

[2] J. Boreczky and L. Rowe, "Comparison of video shot boundary detection techniques," *SPIE Conf. On Storage and Retrieval for Image and Video Databases IV,* San Jose, January 1996, vol. 2670.

[3] M. Christel, D. Winkler, and C. Taylor, "Multimedia abstractions for a digital video library," *Proc. ACM Intl. Conf. on Digital Libraries (DL '97),* pp. 21-29.

[4] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," *Proc. 1999 IEEE Intl. Conf. on Multimedia Computing and Systems,* vol. 1, 756-761.

[5] Y. Li, Y. Ma, and H. Zhang, "Salient region detection and tracking in video," *Proc. Intl. Conf. on Multimedia and Expo (ICME 2003).*

[6] N. Peyrard and P. Bouthemy, "Motion-based selection of relevant video segments for video summarization," *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME 2003).*

[7] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, 1982.

[8] Y. Taniguchi, A. Akutsu, and Y. Tonomura, "PanoramaExcerpts: extracting and packing panoramas for video browsing," *Proc. ACM Multimedia '97,* pp. 427-436.

[9] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video Manga: Generating semantically meaningful video summaries," *Proc. ACM Multimedia '99,* pp. 383-392

[10] M. Yeung and B. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans. Circuits and Syst. Video Technol.,* 7, 5 (Oct. 1997), 771-785.

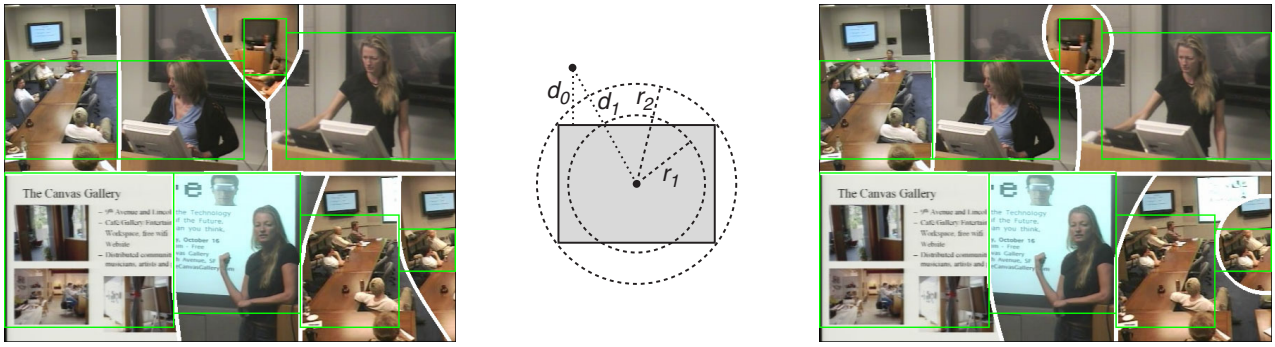


Fig. 6. Rounded borders generated by different distance functions.