Video Carving

Billy Chen[†1] and Pradeep Sen[‡2]

[1]Microsoft, Redmond, WA
[2]Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM

Abstract

We present a technique for summarizing a video into a short segment, while preserving the important events in the original. While many techniques remove whole frames from the video stream when condensing it, we observe that these deleted frames need not come from a single time step. More generally, deleted frames are "sheets" through the space-time volume. This leads to an algorithm whereby sheets are incrementally carved from the video cube to shorten the length of a video. The problem of finding these sheets is formulated as a min-cut problem, whose solution can be mapped to a sheet. We show results by creating short, viewable summaries of long video sequences.

Categories and Subject Descriptors (according to ACM CCS): I.4.0 [Image Processing and Computer Vision]: General

1. Introduction

We live in an age of digital media, where everyone with a cell phone effectively has a video camera in their pocket. One of the problems with having convenient access to these acquisition devices is that we now generate too much digital media, which complicates the task of sorting through it all to find the important content. Many of us face this problem already; we have hard drives full of photographs and raw videos that we promise ourselves that we will someday clean up and organize. This is a particularly significant problem with video, since raw, uneditted footage consists of lots of time where nothing important happens with only a few short moments of interest in between.

The problem of extracting the key information from a video is also particularly important problem in security and surveillance applications. A 24-hour video from a security camera in a parking lot only contains a handful of valuable seconds when a thief breaks a car window to steal the radio. If the time of the crime is unkown, a security guard desiring to catch a glimpse of the thief in action would have to watch the entire video (which is impossible since it is 24 hours long) or watch it in extreme fast-forward (in which case the actual portion of the video when the crime occurred might be skipped).

As we shall discuss in Section 2, several techniques have been proposed to condense long video into a shorter and more useful synopsis. Most common are "downsampling" or "fast-forwarding" schemes where the video is cut-down in size by extracting only every nth frame. However, this simple approach sometimes fails to capture a rapidly moving object since the temporal samples might miss the actual object (an example of temporal aliasing).

In this work we present a novel scheme to take a long video stream with m frames and condense it into a short viewable clip with n frames (where $n << m$) that preserves the most important information. While most approaches prune down the video size by eliminating whole frames from the video stream, we observe that each deleted frame does not have to consist of pixels from a single time step. Instead, we think of the frames to be deleted as "sheets" within the space-time volume where each pixel on the sheet has one and only one time step, but different pixels can have different time steps (Figure 1). We can therefore think of a single frame of video as a sheet where all of the pixels have the same time coordinate.

Our algorithm repeatedly carves out sheets of smallest importance until the desired video size is reached. To do this, we draw on the work by Avidan and Shamir on seam carv-

† bill.chen@microsoft.com
‡ psen@ece.unm.edu

© The Eurographics Association 2008.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

ing [AS07]. Analogous to our 2-D sheets, they identify 1-D low-energy "seams" that are carved out from the image to reduce its horizontal or vertical resolution. As in our work, their seams are not straight but rather follow the contours of low-gradient regions in the image. One of the important differences between our approaches is that they use dynamic programming to find the optimal seams while we use a min-cuts algorithm find the optimal cut within the 3D grid of pixels. After doing the process repeatedly, the result is a shorter video that preserves important information.

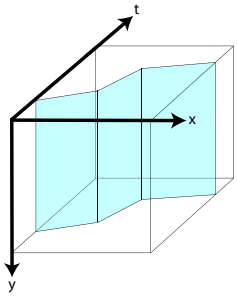

Figure 1: *A "sheet" in the space-time volume. Each pixel of the space-time volume, when projected forward or backward in time projects to a unique position on the sheet.*

2. Related Work

Several techniques have been proposed to create video summaries. As we mentioned earlier, the simplest technique is to simply play the video faster. This is accomplished by skipping frames and averaging the remaining ones. Unfortunately, fast activities may be the lost in the process. To avoid this problem, techniques have been developed that identify activities and adaptively adjust the frame rate [JPH03, DPR*03, BM07].

Recently, in contrast to the frame-based approaches above, object-based approaches have been proposed [PRAGP07, RAPP06]. These techniques represent activities as 3D objects in the space time domain (e.g. video cube) and seek a tighter packing of these objects in the time axis. Identifying these 3D objects relies on accurate segmentation of each frame, after which a packing problem must be solved. Our technique effectively performs the inverse: instead of segmenting objects in the video cube and packing them, we incrementally remove empty regions between objects.

This idea of incrementally removing regions is inspired by Avidan and Shamir's work on seam carving for image resizing [AS07]. To resize an image, they incrementally remove seams, which are 8-connected paths through the image. Their algorithm computes the seam that blends most with its surrounding using dynamic programming, so that removing it leaves little visual artifacts in the resulting image. Our approach extends seam carving to sheet carving in the 3D video cube. Our formulation leads to a simple technique for incrementally removing sheets to summarize video.

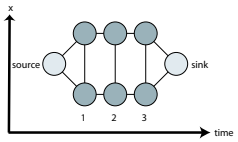

Figure 2: *The min-cut formulation. This example represents a 2-pixel video with 3 frames.*

Complementary to summarizing video is the task of video retargetting for different output resolutions. Recent work [WGCO07] form sparse linear systems with constraints, mapping pixel locations to new ones in the resized video. While this formulation could also be used to shorten a video, our main contribution is the generalization of a video frame to a sheet in the video-cube. We present one way to remove such sheets using a min-cut formulation.

3. Video Carving

A long video can be summarized through video carving by incrementally removing 2D sheets from the video cube to reduce its total time. This is analogous to removing 1D seams from a 2D image to change its width or height. We observe that many seam carving ideas extend naturally to 3D. Instead of a 2D image, the input is a 3D video cube and the 1D seam becomes a 2D sheet. Since we wish to reduce the video length (as opposed to its width or height), the sheet must fully cut across the xy-plane of the video cube as shown in Figure 1.

To compute this sheet, we use a min-cut formulation. By creating an appropriate graph of video pixels and augmenting it with source and sink nodes, we can find the min-cut of this graph and therefore compute the corresponding sheet to remove from the video cube.

First, we define a node for each pixel of the video cube. Nodes have edges to their top, bottom, left, and right neighbors. They also have edges to nodes in the same pixel location in the next and previous frames. In addition, a source and sink node are connected to all the nodes in the first and last frame, respectively. Figure 2 shows this construction for a 2D space-time volume.

Next, edge weights are computed using a measure of spatio-temporal difference. This way, a min-cut will traverse through regions of low difference (e.g. high similarity). When the low-difference sheet has been found and removed, the resulting video will have few visual artifacts since the removed pixels will be similar to their surroundings both spatially and temporally. In our work we use the gradient as a measure of spatio-temporal difference:

$$e(V) = \sqrt{\frac{\partial V}{\partial x}^2 + \frac{\partial V}{\partial y}^2 + \frac{\partial V}{\partial t}^2} \qquad (1)$$

Finally, we find a min-cut on this graph and compute a

© The Eurographics Association 2008.

corresponding sheet that has the property that it has only one temporal value at every projected pixel location. This is a similar restriction to that of seams, since vertical seams, for example, can only have one seam pixel per horizontal row [AS07]. To do this, we first find the set of nodes, S, that have edges that cross the min-cut. We then use a "front-surface" strategy to determine which nodes to remove: for each pixel location, we project it along the time-axis of the video cube, from the first frame to the last frame. The first node $n \in S$ we encounter will be the pixel we remove from the video cube. Figure 3 illustrates this process. This procedure assumes that every pixel location can be projected onto the min-cut surface. It is simple to show that for any graph with the above topology, a min-cut surface is always visible to a pixel location along the time-axis.

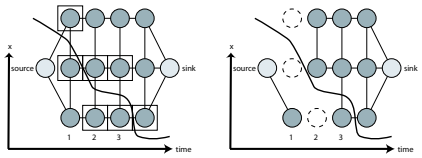

Figure 3: *Extracting the sheet from the minimum cut. On the left, the nodes adjacent to the cut (shown with squares) are considered for removal. On the right, the dashed nodes are removed and the remaining nodes are packed along the time axis. The video is now one frame shorter.*

Once a sheet is removed from the video cube, the remaining pixels are packed to cover the empty space. Because every pixel location had one and only one frame removed, the total video cube is shortened by one frame.

4. Implementation

To compute the min-cut algorithm on our graph, we use Boykov and Kolmogorov's maxflow code [BK04]. However, the memory requirements of storing the entire data structure can be significant. We store the video stream as a 3D doubly-linked grid of of pixels with each "pixel" storing the color and gradient information as well as pointers to its neighbors, resulting in a structure 40 bytes in size per pixel. Since 32-bit Windows gives applications only 2GB of total memory – the remaining 2GB is reserved for the operating system – this limits the maximum number of pixels in our graph to about 50 million. For a 720×480 video at 30 frames per second, this only yields about about 150 frames (5 seconds), which is unacceptable.

In order to process videos of larger sizes, we take the input video and break it up into smaller video subsets, each which can fit entirely within memory. We then extract a single frame from each subset with the min-cut algorithm before proceeding to the next one. Therefore, after the first pass through the entire video is finished, we have removed as many frames as there were video subsets. We continue making passes through the video removing frames until the video reaches the desired size.

5. Results

In this section, we present the results of our technique applied to video test data. For comparison, we compare our approach to the common technique of reducing video length by downsampling, which keeps only every nth frame to reduce the video by a factor of n.

As can be seen from the accompanying video and from Figures 4 and 6, video carving preserves important information that is not in the fast-forwarded version. For example, the street video shows an empty street most of the time, which is reflected in the downsampled video. However, the video-carved version shows the street more busy, with cars passing more frequently. Figure 5 shows how two objects that are temporally separated can be automatically composited together by our technique if the condensed video is short enough. Finally, Figure 7 shows a visualization of a single video sheet by showing the removed pixels in three frames that are included in the sheet.

However, our video carving technique has artifacts that show up as "motion tails" following rapidly-moving objects. These are caused by video sheets that traverse the path of the object, placing it with a previous image of itself on the same frame. These artifacts are the direct cause of having to use a small subset of the video during processing because of memory limitations. Since each video subset that was processed was only a few seconds long and required the removal of a video sheet, our algorithm was often forced to remove frames across a moving object even though there were other places the min-cut would have been better. This problem would go away if we could load larger blocks of video at a time for processing.

Figure 5: *The left two frames are from the original video and are located 1:15 apart. Our algorithm combines the motorcycle and the car into the single frame (right) when the 10 minute clip was reduced to a couple of seconds. However, the fact that a car and motorcycle went by is preserved.*

Figure 7: *Visualizing removed pixels. Each frame above has pixels removed. Although pixels are removed from multiple frames, the total length of the video is reduced by 1 frame.*

© The Eurographics Association 2008.

Figure 4: ***Hallway***. *This 8 minute video from a security camera in a hallway shows a student walking (top row is fast-forwarded, the bottom row is the result of video carving). The student vanishes in the fast-forward version, while the video-carved result preserves his motion through the hallway. To reduce motion tails, both videos were averaged using a 10-frame window.*

Figure 6: ***Street***. *This is a portion of an 8 minute video of a calm suburban street (top is fast-forwarded, bottom is the result of video carving). Fast-forwarding shows little activity, while video carving captures the movement of all the vehicles.*

6. Conclusions and Future Work

Based on our results, video carving seems like a promising technique to shorten long videos. However, there are several avenues for future work. First, we might reduce the motion tails in the condensed video by processing larger blocks of video at one time. By using a hierarchy of multi-resolution frames, we could reduce memory consumption and process larger blocks as well as accelerate our algorithm. In addition, it would be of interest to be able to enforce temporal order in the final video. Because of the way the video sheets can have pixels from different times, the carving process might actually change the order at which events happen in the video. If we identify these events, it may be possible to assign a penalty for reordering them. Finally, because we do not use any object information during processing, the carving of video sheets can cause discontinuities to appear as objects move. This is analogous to the artifacts of seam carving when a vertical seam is removed from a diagonal line which results in a line that no longer lines up on either side of the seam.

By carving out low-gradient video sheets from a long video, we are able to produce a much shorter version that preserves important information, even going as far as compositing objects together that happen different times in the same frame. The technique presented could certainly benefit applications where a long video must be viewed to try to catch short, but important events.

References

[AS07] AVIDAN S., SHAMIR A.: Seam carving for content-aware image resizing. *SIGGRAPH* (2007).

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI* (2004).

[BM07] BENNETT E., MCMILLAN L.: Computational time-lapse video. *SIGGRAPH* (2007).

[DPR*03] DIVAKARAN A., PEKER K. A., RADHAKRISHNAN R., XIONG Z., CABASSON R.: *Video summarization using MPEG-7 motion activity and audio descriptors*. Tech. rep., Mitsubishi Electric Research Laboratory, 2003.

[JPH03] JOJIC N., PETROVIC N., HUANG T.: Scene generative models for adaptive video fast forward. *ICIP* (2003).

[PRAGP07] PRITCH Y., RAV-ACHA A., GUTMAN A., PELEG S.: Webcam synopsis: Peeking around the world. *ICCV* (2007).

[RAPP06] RAV-ACHA A., PRITCH Y., PELEG S.: Making a long video short: Dynamic video synopsis. *CVPR* (2006).

[WGCO07] WOLF L., GUTTMANN M., COHEN-OR D.: Non-homogeneous content-driven dideo-retargeting. *ICCV* (2007).

© The Eurographics Association 2008.