Interactive Video Cubism

Sidney Fels
Dept. of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
+1 604 822-5338
ssfels@ece.ubc.ca

Kenji Mase
ATR MIC Research Laboratories
2-2 Hikaridai, Seika-cho
Soraku-gun, Kyoto, Japan 619-02
+81 (0) 774 95-1440
mase@mic.atr.co.jp

ABSTRACT

This paper presents an interactive video visualization system. In this visualization video data is considered to be a block of three dimensional data where frames of video data comprise the third dimension. The user can manipulate and see a cut plane through the video data. The visualization leads to images that are aesthetically interesting as well as being useful for image analysis.

Keywords

Video visualization, interactive volumetric visualization, video cube, epipolar planar analysis.

1. INTRODUCTION

We introduce a new technique for visualizing video data. In this novel scheme, video data is considered to be a volume of data. The dimensions of width and height are the usual X and Y axis of a frame of video data. The third dimension is derived from the layering of frames of video data sequentially in time as shown in the diagram in Figure 1. Figure 2 shows a cube of real video data. For example, a normal video frame is 640X480 pixels and changes at 30 frames per seconds. Thus, if we capture 100 frames of video we have a block of 640X480X100 pixels. (To save memory though, we use frames that are only 212x160 pixels.) Normal video viewing can be considered a cut plane that is parallel to the X-Y axis and advancing from the first frame to the last frame along the T axis as shown in the diagram in Figure 3. Figure 4 shows a cut through the real video cube in the T dimension which is just a typical frame of video data.

Now, imagine rotating the cut plane to a different location and moving it. For example, consider moving the cut plane so that it is parallel to the X-T axis and advancing it along the Y dimension. At each cut you are seeing all of the X dimension values for all the frames at a given position in the Y dimension as shown in the diagram in Figure 5.

Figure 6 shows an example from the real video cube where the cut plane is parallel to the X-T axis and is positioned near the top of the video data i.e., Y is around 120 pixels out of a possible 160 pixels.

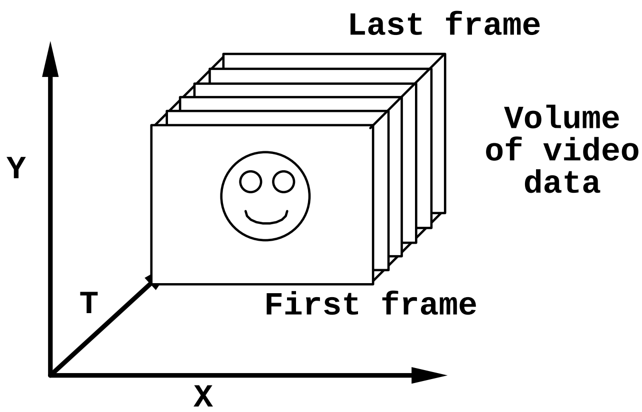

Figure 1: Video frames are stacked in order to form a volume of video data. In video cubism the viewer can manipulate a viewing plane to cut through this volume of video. An example of the video cube with real video data is shown in Figure 2.

Figure 2: 180 frames of 210x160 video data are stacked to form a cube. Each side is a texture so that pixels are interpolated when necessary.

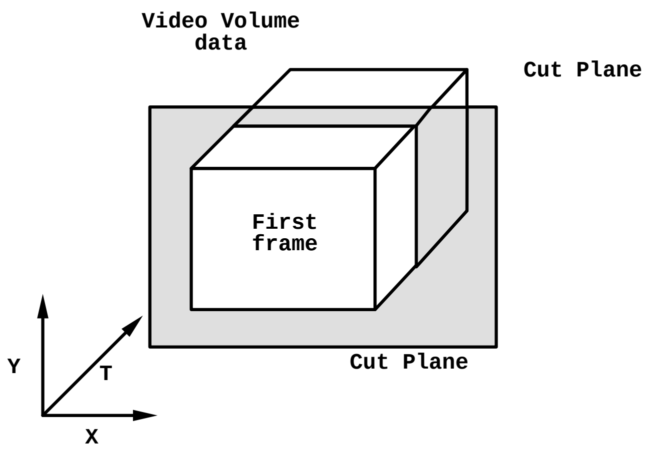

Figure 3: Cutting the video cube parallel to the X-Y axis shows just a regular video frame located at some moment in time. The image from this cut plane can be seen in Figure 4.

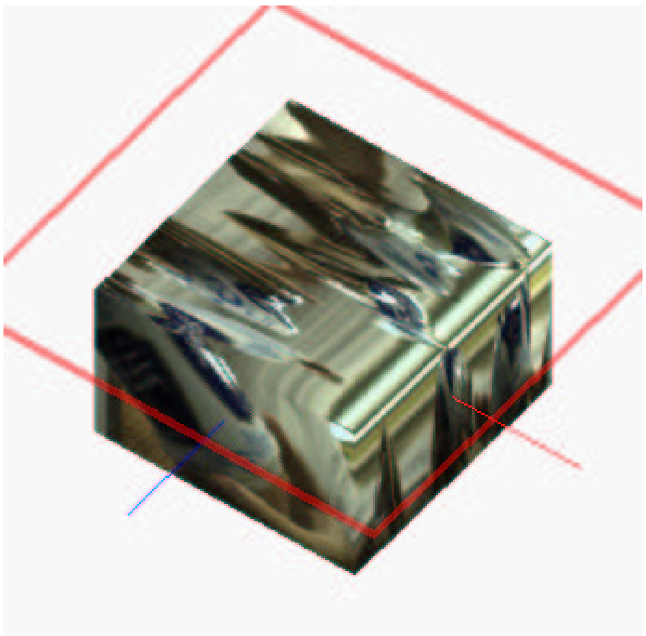

Figure 4: Cutting the video cube parallel to the X-Y axis shows just a regular video frame located at some moment in time.

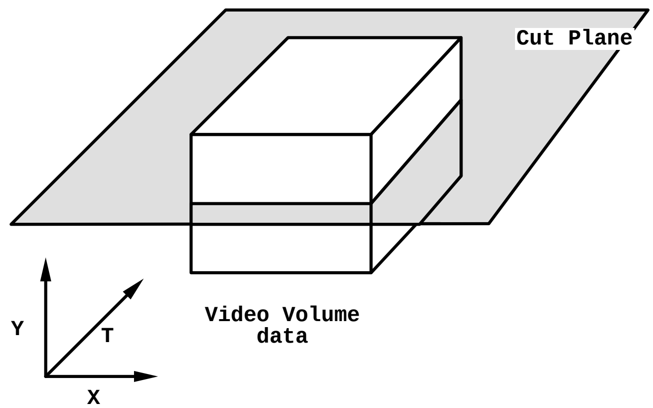

Figure 5: Cutting the video cube parallel to the X-T axis shows the video data displayed across all of X for all the frames for a given value of Y. The image that forms in the actual video cube is shown in Figure 6.

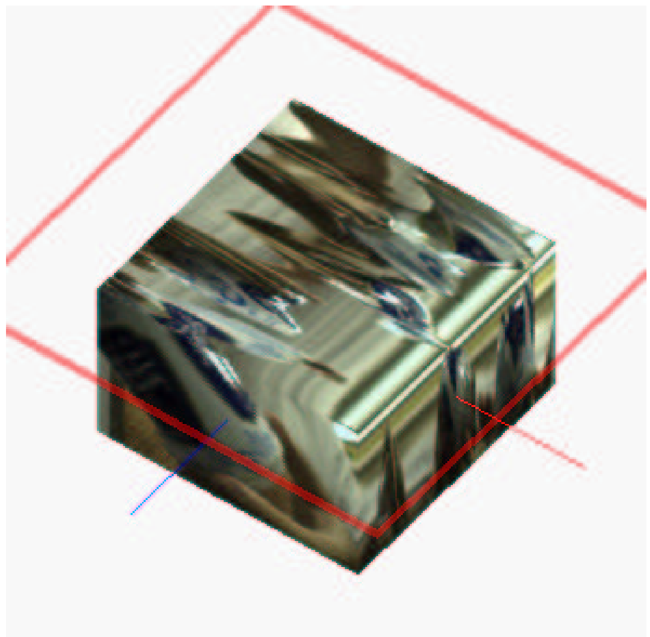

Figure 6: A horizontal cut through the video cube shows all the X pixels across all the frames for a fixed Y.

2. Related Work

Viewing video data along the X-T axis and Y-T axis has appeared in several forms in the literature. Most recently[4], has developed the technique, called the {\it tx-transform}, for use with film. In their work, the different cut planes are always along the basis axis and is used for creating aesthetically interesting dynamic viewpoints of film data. The work, *The Invisible Shape of Things Past* [5] also represents video as a three dimensional object where the topology is determined by the characteristics of the video camera. In this arrangement, the time axis is also displayed spatially depending upon the movement of the camera. The aesthetic character of the images provides an intriguing aspect of navigating the virtual architecture presented in the work as well as allowing interaction with the whole video sequence using a spatial metaphor.

In [1], they describe epipolar-plane analysis for tracking object in motion. In this work, the cut plane images through the video cube are are analysed for straight lines or hyberbolic curves to track objects during camera motion on a mobile robot. In their work, they consider the effect of moving a camera in straight lines relative to a fixed scene. The epipolar plane is a cut plane through the video data. The location and shape of the cut plane is a function of the camera motion. For example, latera camera motion perpendicular to where the camera is pointing corresponds to a horizontal slice. As they observe, camera rotations and translations can lead to different epipolar topology and thus are not always along a single cut plane. In this situation, a non-planar cut would be needed to track the pattern created by a stationary object. In video cubism, the camera and the objects are free to move. The complex patterns that form are due to the plane cutting through the epipolar lines allowing multiple representations of the spatiotemporal data.

Interestingly, the reverse of the epipolar image formation could be used to simulate camera movement from images without the need for rendering. For example, a lateral camera movement corresponds to straight lines along horizontal cut planes. The set of equations formed for each pixel could then be computed and solved for changing camera positions. Likewise, the same technique could be used for more complex camera and object motions or deformations.

The main distinction this work has is that the cut plane used to view the video data can be moved to any angle and position in real-time. This provides an opportunity to interactively explore the video cube from many different angles to get both aesthetically interesting static images as well as motion effects. Currently, only a single cut plane is supported, thus the investigation is like being able to move a window around the video cube to see all sides as well as inside the video data; hence the name video cubism.

3. Video Cubism

The video cube system has three main parts, the video cube data, the virtual cube and the cut plane. The video cube data is formed from frames of video data. The virtual cube is the representation of the video data in virtual coordinates. Finally, the cut plane is the window that is used to cut through the virtual video cube that in turn displays the corresponding video data.

3.1 The Video Data

The first component of the system is the video cube data. The complete video data is stored in memory as a 3D array consisting of a sequence of frames of video data. Currently, we are using RGB values for video frames that are 212x160pixels. Using the full 3D array representation made addressing individual video data (vixels) that are on the cut plane simple. Other formats could be used but the corresponding equation may need to be modified. The video data is used as a texture that is mapped onto the appropriate face in the video cube.

3.2 The Virtual Video Cube

The video data is represented by a virtual video cube. The video data is mapped onto the cube with the appropriate scaling to convert from the virtual video cube coordinates to the array indices of the video data. The dimensions of the cube can be selected arbitrarily since the texture mapping will stretch the video data so that it fits the shape accordingly. This is similar to the technique used in [2]. In our implementation, we use a virtual cube that is 1.0x0.75x1.0. The Y axis is set to maintain a 4:3 aspect ratio so that the video frames do not looked stretched in the Y direction. The Z dimension is set to 1.0 so that the time axis looks the same length as the X axis. Of course, if we have fewer than 212 frames of data the time dimension will be stretched. Likewise, if we use more than 212 frames the data will be compressed to fit. In the images in this paper 180 frames (6 seconds) of video data are used. The cube can be rotated interactively and the corresponding intersection with the cut plane displayed. Figure 1 shows the video cube rotated without any cut plane intersection.

The video cube is represented as a set of 12 line segments. The line segments are used to determine the intersection with the cut plane whenever the cube is rotated or the cut plane is moved (see subsection 3.3).

3.3 The Cut Plane

The cut plane is represented by its scalar equation (**Ax + By + Cz + D = 0**). Anytime the video cube is moved or the cut plane is moved the intersection of the cut plane and the video cube is recalculated. The corresponding intersection points is a polygonal face. The appropriate video data which should appear on the intersecting face is calculated and placed in a texture. This texture is mapped onto the face.

The scalar form for the cut plane was chosen since it is relatively simple to find the intersection points between the cut plane and the 12 line segments which make up the video cube. For each line segment between $P_1 = (x_1,y_1,z_1)$ and $P_2 = (x_2,y_2,z_2)$ the intersection equation:

$$\mathbf{u} = \frac{\mathbf{A}x_1 + \mathbf{B}y_1 + \mathbf{C}z_1 + \mathbf{D}}{\mathbf{A}(x_1 - x_2) + \mathbf{B}(y_1 - y_2) + \mathbf{C}(z_1 - z_2)}$$

is applied where **u** determines the point of intersection. If **u** is between 0 and 1 then the plane intersects the line segment. The only other condition of interest is when **u** is 0 in which case the line is either on the plane or is parallel to it. For the case where it is on the line we consider the two end points to be intersecting points.

Once all the intersecting points are collected in a list any duplicates are removed (possibly from line segments which lie on the plane). The list of points is then ordered to create a convex polygon. The ordering is done by first sequentially ordering points that are on the same cube face. If the points are not on the same face, then the next point in order is selected using the smallest distance between points.

The sorted list of intersecting points gives a planar intersecting polygon. The intersecting polygon is rotated to lie in the X-Y plane. The bounding box of the polygon is calculated and used to iterate over the polygon texels; that is the location of each point in the intersecting polygon stated in 2D texture coordinates. For each point in the bounding box that lies inside the polygon we rotate the point back onto the cut plane and calculate the appropriate vixel from the video data. The vixel is copied into the two dimensional texture array.

Finally, for each of the intersecting points the corresponding point in the texture space is calculated so that when the intersecting polygon is drawn the correct texture region is applied.

With only simple optimisation the slowest update occurs within 40msec (25 frames/sec) on a four R10000 processor Onyx. This provides sufficient speed for real-time interaction with the cube and the cut plane. We are currently using only 2D texture mapping as this is more portable; however, using 3D textures will make the mapping much simpler.

3.4 Interaction Controls

The main interaction controls include rotation and translation of the whole scene, the video cube or the cut plane. In the simplest version, we use sliders for all the controls. For the cut plane, only translations along the normal make sense since the plane is infinite in width and length. It is also possible to animate the cut plane translation along the normal. Thus, if the cut plane is positioned parallel to the X-Y axis, normal video play results. Off axis positioning provides interesting animation effects. Figure 7, Figure 8 and Figure 9 show some of the stimulating images that result. The imagery changes in real time which animates the images providing deformations of objects depending upon their motion and the camera motion.

We have also performed some experiments using Polhemus 6 DOF trackers; one sensor is embedded in a cardboard box and the other is attached to a cardboard frame with the middle cut out. The box is calibrated so that it directly rotates the video cube. Likewise the frame is calibrated to correspond to the cut plane orientation and position providing absolute control. The interface is difficult to use as the user's hands often get in the way of each other. This is because when the one hand holding the box goes through the cut plane frame it is difficult to continue to manipulate the box since the user's hand movement is now constrained by the frame. One solution we are considering is to use a relative orientation and translation mechanism such as found in[3].

One success we had with the absolute orientation and positioning mechanism was when we held the cut plane fixed and allowed the user to move the video cube using the box only. Moving the box in and out of the cut plane allowed users to explore interesting animation effects of rotating cut planes. For example, in Figure 8 the user is seeing the rigid can being deformed by having a cut plane that extends partially through the time axis as well as the X and Y axis. By moving the box around this deformation is animated for striking effects.

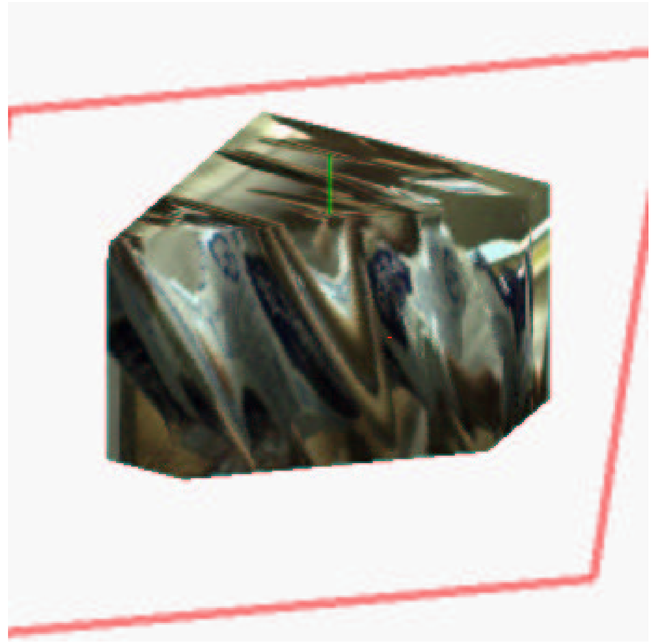

Figure 7: Example of a cut plane.

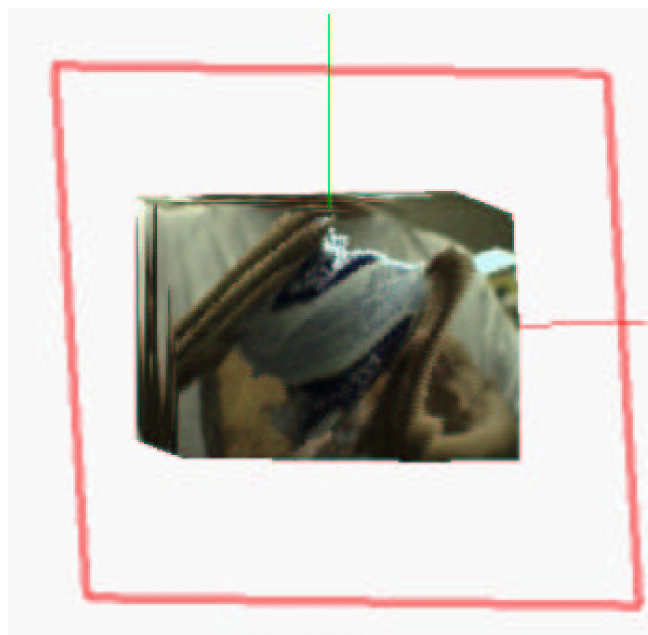

Figure 8: Example of a cut plane.

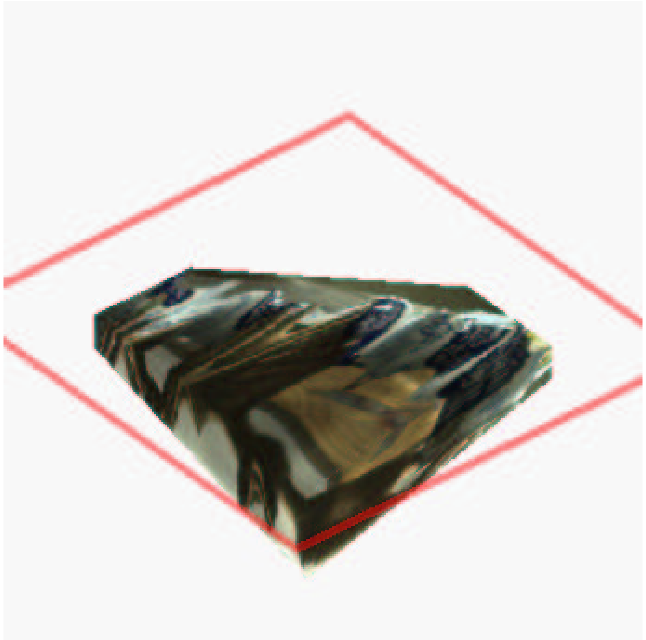

Figure 9: Example of a cut plane.

3.5 Conclusions and Future Work

With video cubism it is possible to interactively explore video data in all three dimension simultaneously. The main purpose for this project is to explore some of the aesthetics of looking at video data from a variety of perspectives. The images can be abstract or concrete depending upon the orientation and position of the cut plane as well as the movement of the camera and the object in the video data.

When the video data is of the user, the cut plane is like a mirror that allows the user to inspect themselves in three dimensions; width, height and time, at the same time. This perspective of seeing all dimensions simultaneously makes video cubism artistically interesting.

One interesting direction (suggested by Poupyrev, personal communication) is to use the cut plane techniques to do image based object deformation, video morphing and camera motion. We plan to investigate these possibilities.

We also plan to continue investigating ways to explore the dynamic imagery possible with the video cube. To continue the video cubism theme, one idea is to have multiple connected cut polygons though the video space that topologically fit on a plane. These connected polygons could then be placed on the plane and display on a large screen. This configuration would give a temporal kaleidoscope. Connecting this to real-time video would provide an interesting aesthetic experience as found in the Iamascope[1]. Non-planar cuts through the cube will also be interesting.

Video cubism is a novel way to explore video data for both aesthetic imagery and analysis.

4. ACKNOWLEDGMENTS

Many thanks to Ivan Poupyrev for helpful discussion.

5. REFERENCES

[1] Bolles, R.C., Baker, H. H., and Marimont, D.H. Epipolar-plane image analysis: An approach to determining structure from motion. International Journal of Computer Vision, 1(1):7-55, 1987.

[2] Fels, S.S., and Mase, K. Iamascope: A graphical musical instrument. Computers and Graphics, 2:277-286, 1999.

[3] Poupyrev, I., Otsuka, T., Weghorst, S, and Ichikawa, T. Amplifying rotations in 3D interfaces. In ACM SIG Computer Human Interaction (CHI'99), pp. 256-257, 1999.

[4] Reinhart, M. tx-transform. Available through http://www.tx-transform.com/frame_e.htm, 1998.

[5] Sauter, J. and Lusebrink, D. The invisible shape of things past. Available through http://www.artcom.de/projects/invisible_shape/welcome.en, 1997.