Timespace Reconstruction of Videosequences

Juraj Hájek
hajekj@decef.elf.stuba.sk

Department of Computer Science and Engineering
Slovak University of Technology
Bratislava / Slovakia

Abstract

The main object of this paper is to explain the method of the time-space reconstruction of video-sequences. Time-space reconstruction of video-sequences is a method for visualization time-dependent image data in 3D space. Principles of this method are described in this paper and some potential problems will be analyzed. Also basic possibilities of implementation are described.

KEYWORDS: time-space reconstruction, video-sequence, 3D model, multiplanar reprojection, sign language.

1 Introduction

There exist many application where we need to detect moves or deformations of objects. Monitoring of moves or deformations may be improved in computer animation - monitoring of moves of real human or other real model. It can be also used for computer analysis of sign language to help deaf-and-dumb people.

It is very inefficient and unpractical to search changes between video frames by comparing every frame to each other. This is the reason, why we need to use some method for visualization of this data. One of these methods is to create a volume model in 3D space. This can be very useful when we need (for example) to detect collisions of object or changes in position of objects.

After doing this kind of visualization we don't need to compare separated frames, but we can see trajectories of objects in 3D space. The visual appearance of this object is the same as objects created by sweeping in 3D modeling, but generating shape is changing in time. 3D visualization is related to some another methods of creating volume models (for eg. making volume models from computer tomography in medicine or visualization of dynamic processes in industry). We can also use this method for visualization of this kind of data, but there exists many specialized applications in this field.

2 Conception of video data in 3D space

We can imagine video in 3D space like slices of cuboid with sizes $x \times y \times t$. Dimension t is equal to length of video and x and y are dimensions of each video frame. Briefly, we can imagine a video like a box made from sequences of photographs as shown on Fig. 1. This conception is very simple and clear.

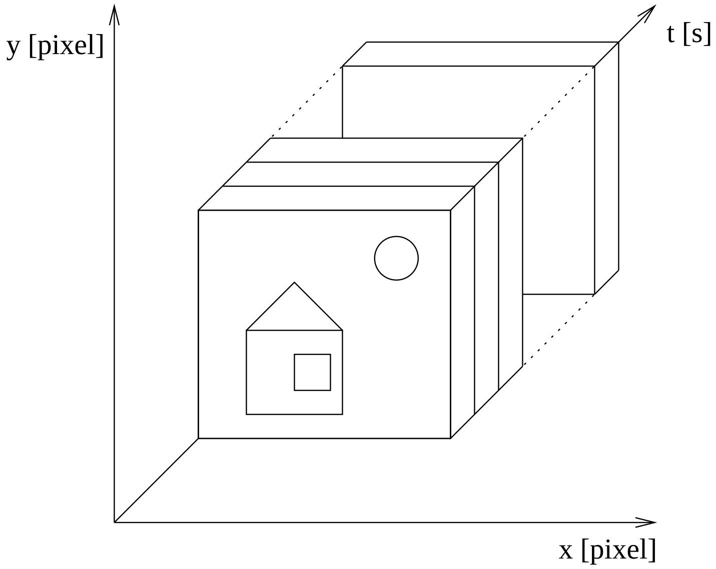

Figure 1: Conception of video-sequence as a 3D object.

Try to imagine, that we can interactive manipulate this object. We can cut this object in many different ways and watch different time dependencies in dynamic image. Fig. 2 shows such a solid object as we described. Figs. 3, 4 and 5 are different angled cuts of this object. Using this, it is easy to detect in which time object on scene pass through boundary defined by x or y.

Another example of video-sequence visualization is Fig. 6. Source for visualization is a sequence of images with a rotating circle. The projection of these images to 3D space is shown thereafter.

With this example we can see the trajectory of a moving object. With this technique we can watch trajectories or deformations of many objects at once. It makes the visual detection of changes between frames much faster than "by the frame by frame" method.

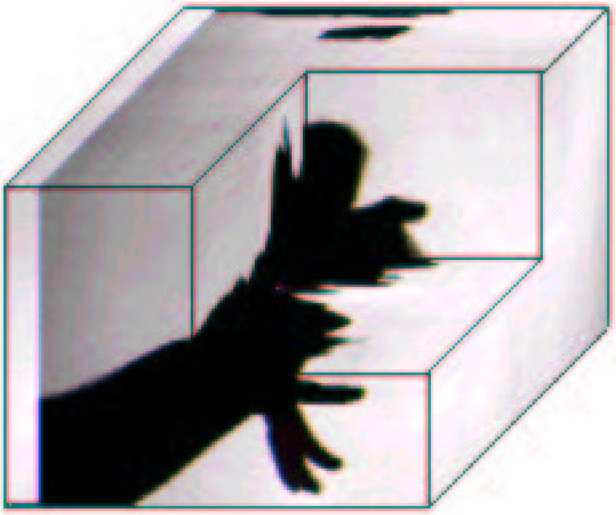

Figure 2: Solid 3D object created from source video file. (Shadows of hands on the wall)

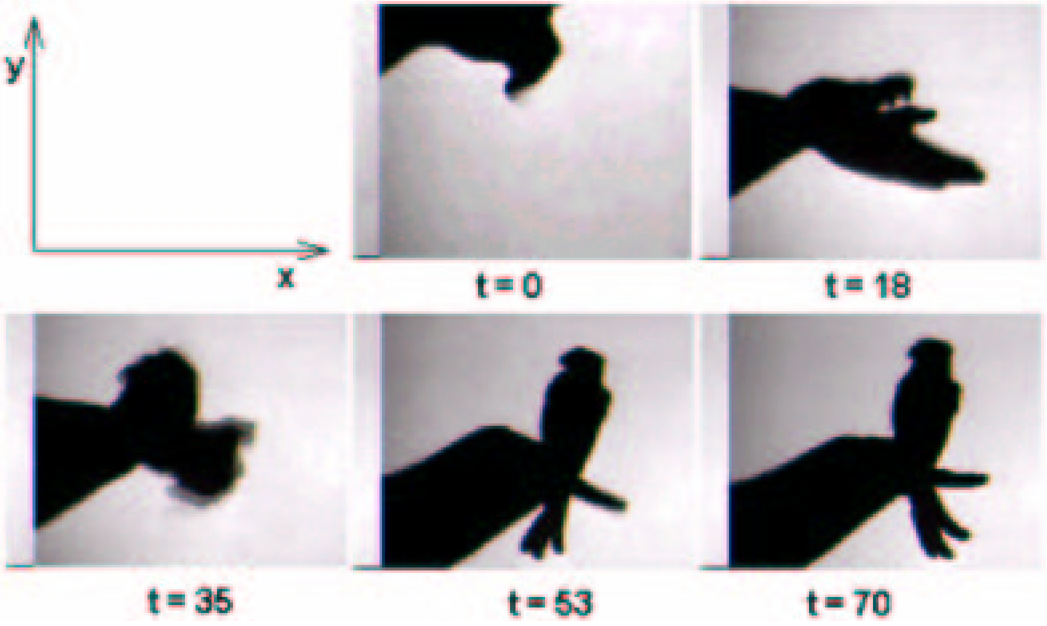

Figure 3: Slices of original 3D object with constant time.

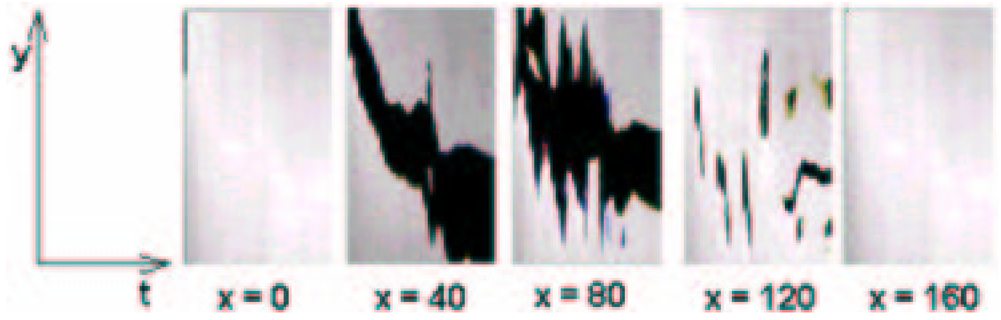

Figure 4: Slices of original 3D object with constant x dimension.

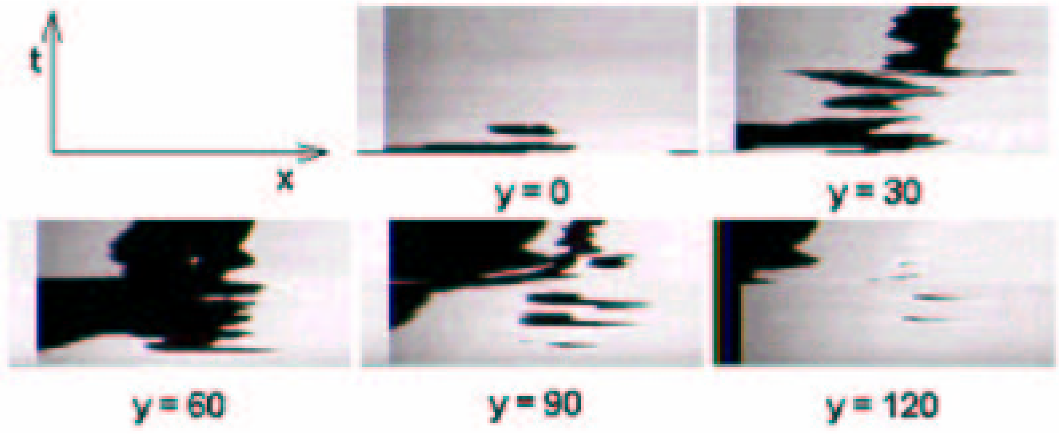

Figure 5: Slices of original 3D object with constant y dimension.

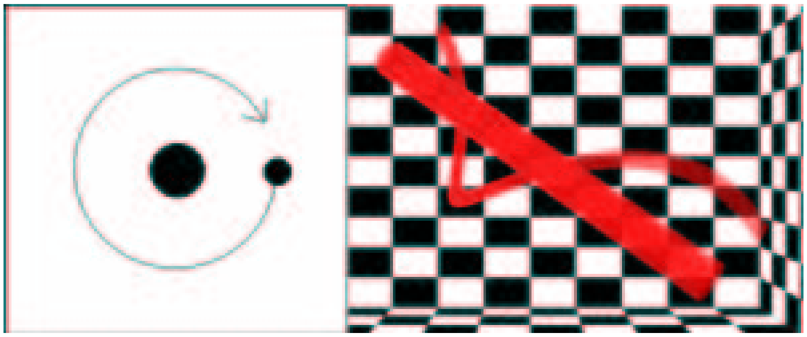

Figure 6: Visualization of the rotating object.

3 Demands on video visualization system

System for video-sequence reconstruction should have support for processing of video data from some standard format. It should have support for interactive viewing of volume models. Performance and speed are very important aspects, because a vaste amount of data is processed.

In real application for visualization complementary tools for video preprocessing are also needed (filters, basic tools for video cutting).

4 Architecture

The basic schema of system for video visualization is shown on Fig. 7.

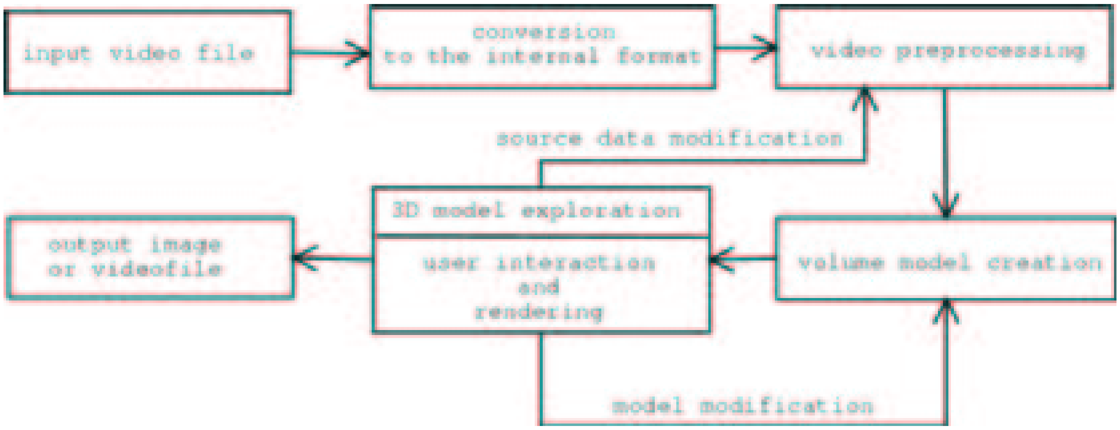

Figure 7: Basic schema of system for video visualization.

First of all, it is useful to convert the input file (or its parts) into independent internal format for easier manipulation with frames later. After that, images are preprocessed. The reduction of colors may be performed, or data may be passed through high-frequency filter to reduce scene complexity. Finally, if visualization method does not map the image data directly into 3D space, a volume model may be created and we can render output image.

5 Possible solutions

There are more alternatives how to implement a system for the visualization of video-sequences. We can choose different methods for creating a volume model or different projection methods. These alternatives shall be described in next sections.

5.1 Visibility of spotted objects

Real video-sequences (not trivial computer animations) contain a large amount of unimportant data (objects on background, noise, etc.). This is the reason why it is very difficult to watch spotted objects. These parts of image must be reduced or removed completely.

Opacity of the object can be defined through their importance. Borders of visualized objects can be identified from brightness of pixels in individual frames or colour. It is useful to preprocess input images, for. eg. histogram adjustment or filtering.

Identification of objects and "empty space" may be very difficult, especially in complex scenes. It is easier to prepare suitable scene than process unsuitable data. By this reason, it is recomended to prepare scene before turnig it into video media, if it is possible (simple one coloured background, etc.). In the case that this is not possible, we can use some trics, like to fix infra diodes on moving parts of objects or coloured table-tennis balls.

Borders of objects in scene may be defined also manually, but it is not possible in the large count of frames. Process of object identification can be also automatized by artificial intelligence.

Fig. 8 showns grayscale animation, which is used for later visualization. This animation has well defined borders of objects, so we do need to preprocess the images and a volume model can be created directly.

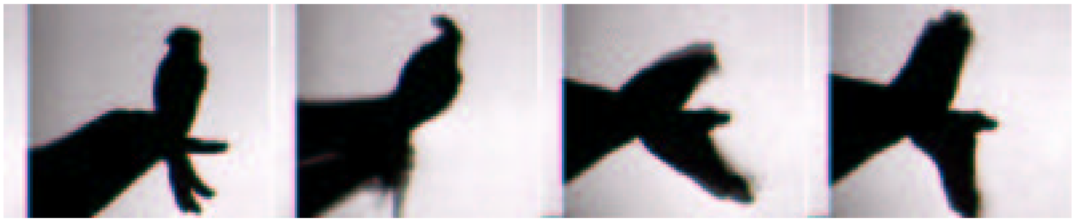

Figure 8: Shadows on the wall. Source animation for volume visualization.

On Fig. 9 is 3D model of this animation. Objects with different brightness has different opacity. Different colors are used for following frames for better visibility.

On Fig. 10 is the same model as in Fig. 9, but background is removed by threshold. On the next figure is other view to the same dataset.

5.2 Volume visualization methods

The fundamental algorithms are of two types: direct volume rendering (DVR) algorithms and surface-fitting (SF) algorithms.

Algorithms for volume visualization can be classified into two fundamental categories [3]:

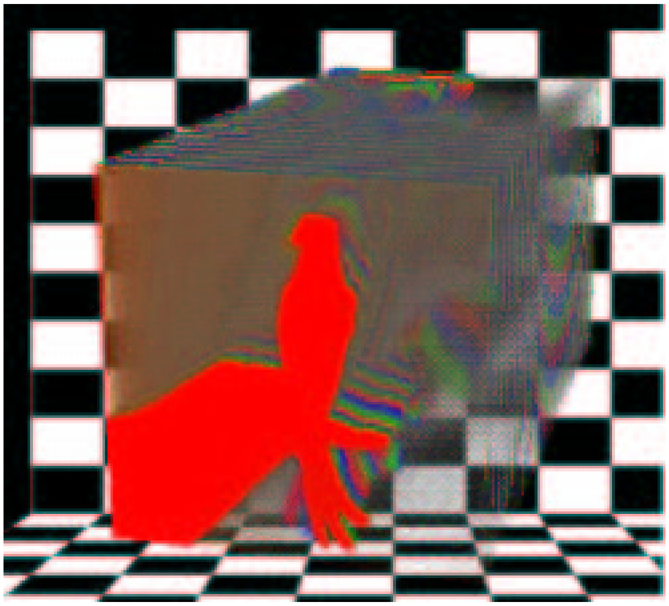

Figure 9: 3D model created from source animation.

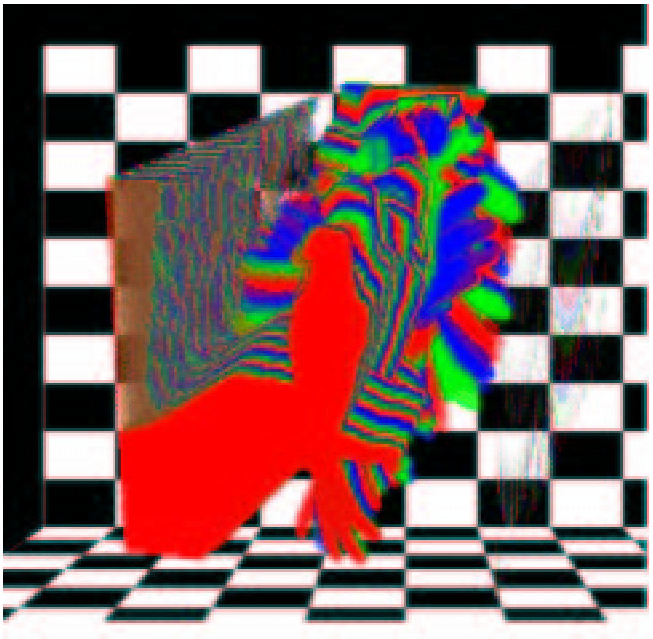

Figure 10: 3D model of animation with removed background.

- direct volume rendering (DVR)
- surface fitting (SF)

The third category of algorithms is also sometimes specified. It consists of different fast algorithms suitable for interactive rendering.

DVR methods map elements of 3D object directly to the screen space. Volume raytracing is typical algorithm in this category . They don't use any geometric primitives for intermediate representation. They are slow, because the entire dataset must be traversed on each rendering. Rendering may be progressive. Firstly, image with low resolution is rendered. Secondly, image with better resolution may be created. DVR algorithms are good for amorphous objects, for example clouds, fog and fluids.

SF methods do not map elements of volume object directly to the screen space, but they map planar polygons or patches into object contours. This process is sometimes called iso-surfacing. The best known algorithm of this category is "marching cubes" algorithm. The source dataset must be traversed only once, after that we can manipulate with polygonal model. SF methods are faster than DVR methods. The reason being that is, that after obtaining surfaces we can use conventional rendering methods. This way, existing hardware accelerators can be used.

It is not necessary to use DVR and SF methods, if the prime object of visualization is only exploring the volume model. This is also a primary object of visualization of video-sequences. The result of this knowledge is, that we can use some of the simpler and faster interactive methods. Interactive methods are usually easy to understand and implement (for example wire model). They are fast, but the output has lower quality. Also information value of rendered image is limited.

Probably the best method (and the most intuitive) method for video visualization is multiplanar reprojection ("slicing"). A volume model is created of multiple slices - planar polygons. These slice represent frames of video. The problem of rendering is reduced to texture mapping. If a higher quality of output image is requested, it is also possible to extend this method to interpolation of voxels between the slices. If a volume model is created by multiplanar reprojection then the visibility of objects can be simply implemented as alpha-blending of polygons.

5.3 Choosing type of projection

Perspective projection does not give good results, it is caused by the deforming of objects. It is better to choose an orthogonal projection, especially if we need to interpret the data exactly. Perspective projection causes problems in detection of scale changes of objects.

Perspective viewing can also cause ray divergence problems in some DVR algorithms.

6 Conclusions and future works

The described system for time-space reconstruction of video-sequences has only experimental status. This system can be used (except research) in computer animation or in sign language analysis.

At the present time, base framework for video visualization has been estabilished and and we continue implementing of complex application with the set of complementary tools for video cutting and image processing. This study is extension of works [2] and [4].

Acknowledgments

Especially I would like to thank my supervisor doc. Ing. Martin Šperka for his advice as well as his many helpful comments and suggestions. This work was researched under his leadership as a part of master thesis on Department of Computer Science and Engineering at FEI SUT in Bratislava, Slovakia.

References

[1] J. Koloszár D. Jocha. *Interactive Virtual Colonoscopy.* CESCG see [doplnit adresu], 2001.

[2] J. Hájek. *Časopriestorová rekonštrukcia videosekvencií (vytvorenie modelu).* KIVT FEI STU Bratislava, 2001.

[3] G. Scott Owen. *Volume Visualisation And Rendering.* SIGGRAPH see http://www.siggraph.org/education/materials/HyperVis/volume/volume.htm, 1999.

[4] V. Pavlišin. *Časopriestorová rekonštrukcia videosekvencií (prehliadanie).* KIVT FEI STU Bratislava, 2001.