Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California, USA

DETC2005-84393

Visualizing the Time Motion of Planar Mechanisms with Volume Methods

Mike Bailey
Oregon State University
mjb@engr.oregonstate.edu

Abstract

Two dimensional, or planar, mechanism design is a mainstay of Mechanical Engineering modeling and analysis. An important part of the design process is the visualizing of the motion of the mechanism. This paper describes a novel approach to visualizing the time motion of a planar mechanism – turning the time dimension into a spatial dimension. All three dimensions (x,y,time) are then treated as a 3D volume. From there, we use interactive volume visualization techniques, including slicing and thresholding. As is seen, this method is able to produce new insights into planar mechanism motion, particularly when more than one mechanism is working cooperatively.

Keywords: Visualization, Mechanism Design, CAD, Computer Graphics

Introduction

Two dimensional, or planar, mechanism design is a mainstay of Mechanical Engineering modeling and analysis. It is more complex than it looks. Determining link lengths, connections, and ground positions to produce a desired mechanical motion can be quite time-consuming. There has been considerable research on this topic (e.g., Bhatt03, Hicks01, Nagchaudhuri00). There are also many commercial programs to aid in the analysis of such mechanisms (for example, [ADAMS05]). There are fewer programs to aid in the synthesis of such mechanisms. Indeed, often the synthesis process consist of taking an initial guess, analyzing the mechanism's motion, and then adjusting the parameters accordingly until the analysis shows that the motion is correct within the design tolerance.

In the world of planar mechanisms, fourbar linkages have been used extensively to create certain types of mechanical motion. Fourbar mechanisms are so named because they consist of three links pinned in series, with the first link and third link each pinned to ground. This sounds simple, but in fact, extending the middle link can create an amazingly rich variety of motions, surely one to fit any need. Figures 1 and 2 below show two different fourbar mechanisms, each with a very different motion path at the "peak" of the middle link. Note that the lengths of the links have not changed from Figure 1 to 2, just the location of the "peak".

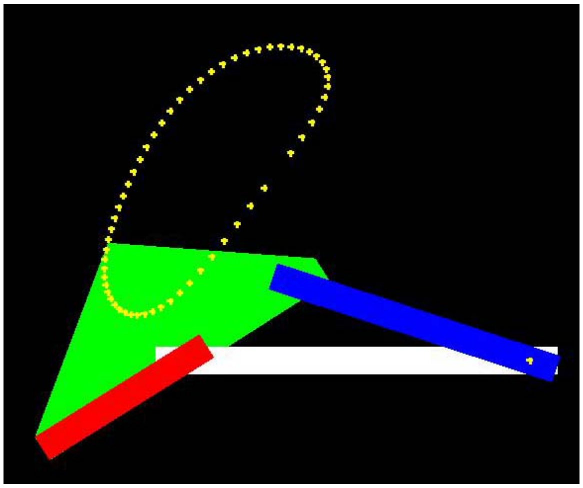

Figure 1: Trace Plot of the "peak" of a fourbar linkage

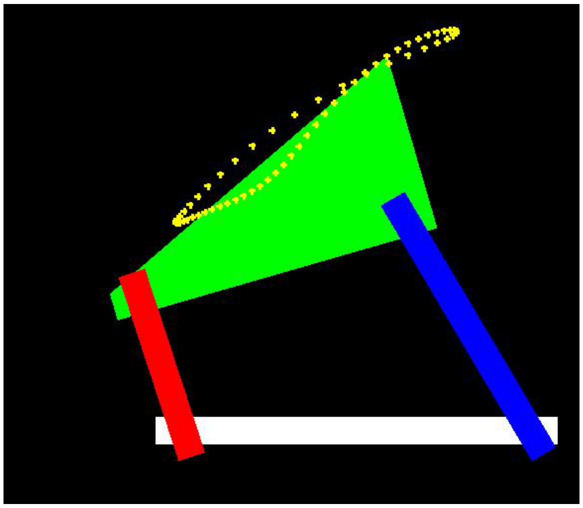

Figure 2: Changing the location of the "peak" can dramatically change the motion

This is not to say that fourbar linkages only became useful in the age of the computer graphics display. In fact, Hrones and Nelson [Hrones51], in the 1950s, used manual methods to painstakingly create a very large atlas of possible showing the motion paths generated by many permutations of fourbar link lengths and peak locations.

Many techniques have been brought to bear on the problem of letting a designer know how a particular fourbar mechanism will behave. The technique used in Hrones and Nelson, and later one of the earliest computer graphics techniques, was the *trace plot*. This consists of having the peak leave a small marker behind as it goes through its motion cycle. This was shown in Figures 1 and 2. As pointed out in Hrones and Nelson, the markers not only show the path, the distance between the markers also shows the relative speed over the range of motion.

Graphics animation has also been a big plus. As graphics systems went from static systems, such as plotters and direct view storage tubes, to dynamic devices such as random scan and later raster scan devices, complex mechanisms could be easily seen in motion. One problem with this was that the moving mechanism simulation look great if you were in front of the screen, but reverted to a static display if you wanted to publish results. A clever compromise has been to use Hollywood motion blur techniques, which can be easily implemented using the OpenGL Accumulation Buffer [Shreiner04] as shown below in Figure 3.

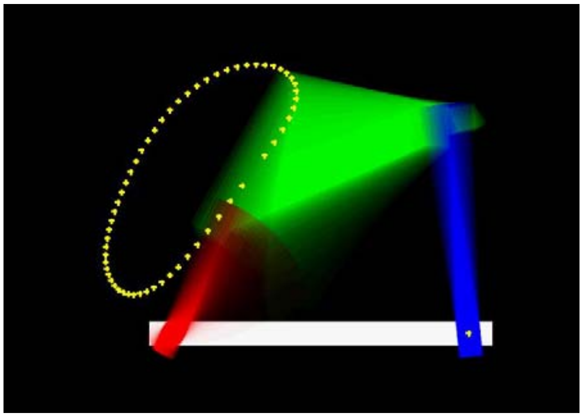

Figure 3: Motion Blur to show a mechanism in motion

Time as a Third Dimension

Unfortunately, all of these methods to model the motion of a planar mechanism involve some sort of compromise. The trace plot only shows the movement of one point, but shows it for all times. The dynamic animation shows the position of the entire mechanism, but only for a single time step. In this project, we were interested in looking at ways of visualizing the entire mechanism over its entire motion cycle.

To do this, we turned to *direct volume visualization*. Volume visualization is a computer graphics technique that is used to understand the patterns in a 3D cloud of points. Direct volume visualization is most closely affiliated with medical imaging, where the 3D cloud of points comes from stacking layers of 2D images, such as typically come from a CAT or MRI scan. Each scan slice is composed of small dots, known as picture elements, or *pixels*. When those slices of pixels are stacked up, each pixel is given a certain thickness. This turns it into a volume element, or *voxel*.

The object of direct volume visualization is to make sense of that cloud of voxels. This happens by tracing rays through the volume, performing weighted averaging of teach voxel's color along the way. For many years, this was done in software [e.g., Westover90] and recently has been able to take advantage of hardware acceleration [Pfister99, Bailey01].

Figures 4 and 5 show the use of volume visualization to show the time motion of a fourbar linkage. Figure 4 shows a "front" view, looking in through all time steps, so that the resulting image looks like a motion blur of the entire range of motion. Figure 5 shows a better view showing more of the time motion of the mechanism.

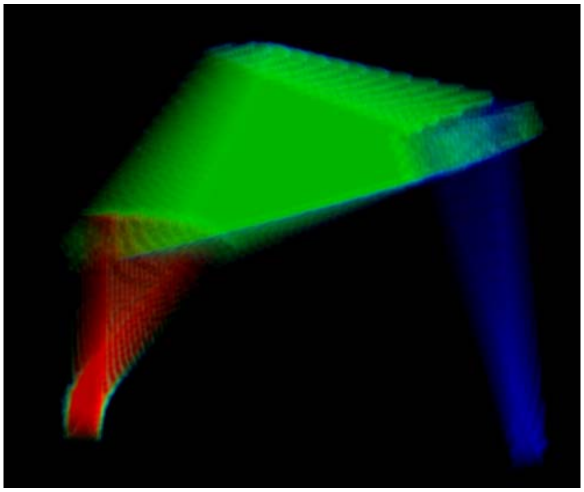

Figure 4: "Front" view of the motion volume. Time recedes in the distance.

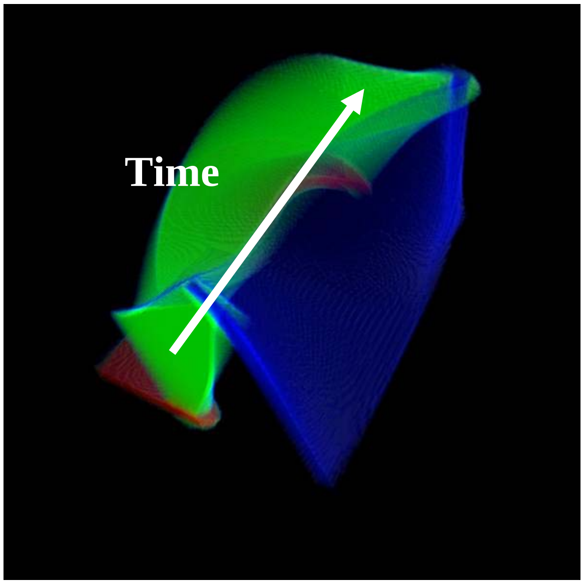

Figure 5: Overhead view of the time volume

Creating the Volume

The volume could have been created procedurally, that is, we could have created the 3D array of voxels and then computed which voxels were occupied by which elements of which mechanism at which time. Instead, we took the easier route and had the graphics system create it for us. Our own OpenGL graphics program created the 2D figures in this paper. OpenGL [Shreiner04] is an industry-standard graphics Application Programming Interface (API), and is the most popular one in the area of scientific and engineering visualization. We modified this program to upload the scene pixels after each frame of the animation, using the OpenGL **glReadPixels()** routine. Then, then the images were layered on top of each other to create the time volume.

This, by itself, turned out to be inadequate. Normally, when displaying a 2D scene, OpenGL would place new scene geometry on top of, and hiding, existing scene geometry. Thus, the appearance of the volume would be dependent on the order in which it was created. Instead, we wanted to see where scene elements overlapped. To do this, we drew the scene with the OpenGL blending turned on with the color addition function enabled. This caused the drawing process to add a new color to an existing color instead of replacing it. This resulted in a much more information-rich scene.

Dual Mechanisms

Besides providing s better understanding of the motion of a single planar mechanism, the volume method also has real advantages in designing and modeling multiple planar mechanisms that need to function in nearly the same space. Figure 6 shows two fourbar linkages each with a different "peak" geometry. Such combinations can be used quite effectively to model materials handling and other close-coordination design situations.. Clearly this won't work, as the two middle sections collide. From this view, it can be seen that one solution to the collision problem is to reposition one of the linkages farther away in X or Y.
Figure 7 shows the 3D volume view of the situation.

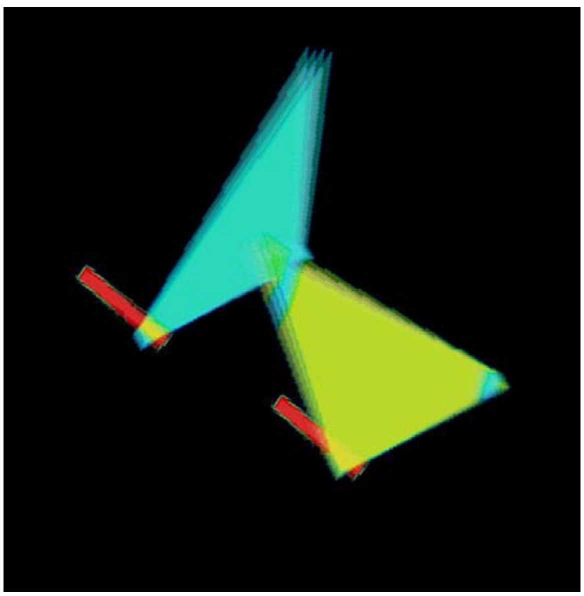

Figure 6: Dual fourbar linkages which collide during their range of motion

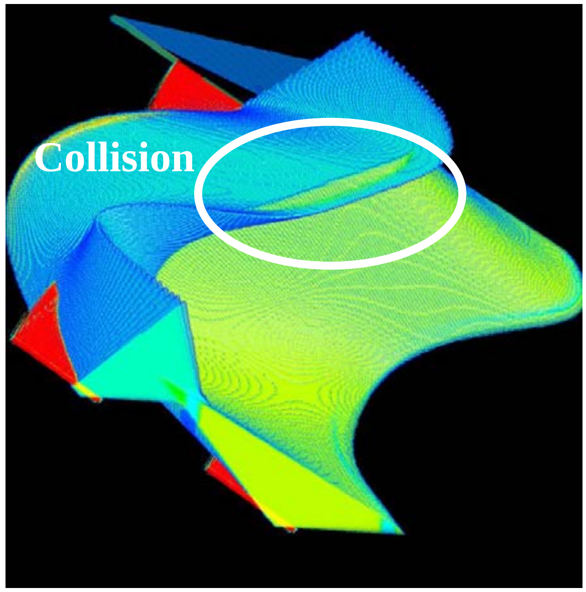

Figure 7: Collision detail in time-volume space

The volume view allows us to treat time exactly the same as a spatial dimension. In this view, we can see that another way to solve the collision problem is to "slide" the right-hand mechanism in time. This is the same as making the input angle of one mechanism out of phase with the other. We can get a good first guess by eyeballing Figure 7, or we can measure it exactly within the volume. In this case, the answer is –45 degrees. Figures 8 and 9 show the results of applying this phase difference. Each figure shows the resulting dual motion volumes. Figure 9, the top view, especially shows how nicely separated the two envelopes are. It also shows that the two mechanisms can be moved even more spatially closer together with no collision problems.

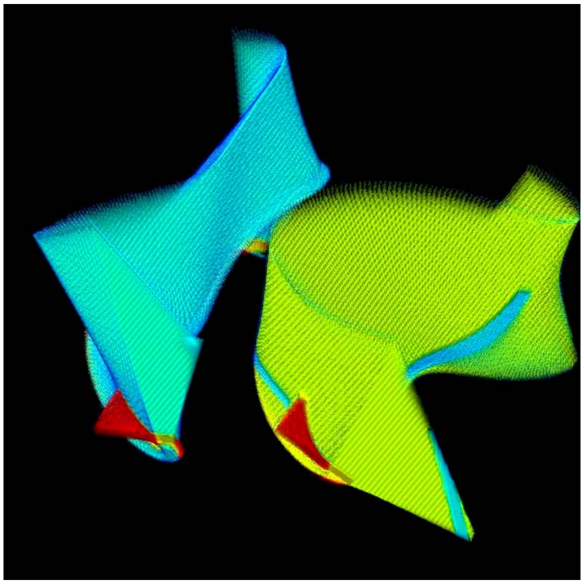

Figure 8: New time motion volumes showing good separation

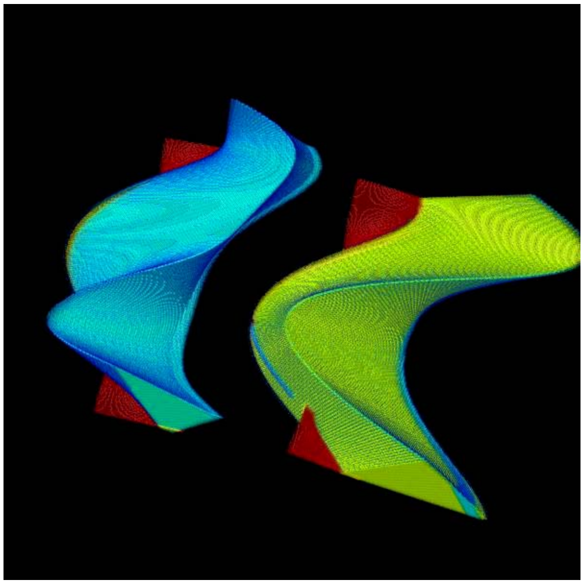

Figure 9: Overhead view of the time motion volumes

Conclusions

One of the lessons we have learned in scientific visualization is that there are really no rules. Any method that provides more insight into a scientific or engineering analysis is fair game. This method represents another trick in the bag. By representing time as a dimension no different from 2D space, certain insights can be obtained. This provides useful insight into understanding the time motion of a single planar mechanism. It has also been especially useful for coordinating the motion of multiple mechanisms.

However, our volume interaction and display system shows these volumes much better than static images alone do. The key point is that this method doesn't just declare there to be a collision, it *characterizes* it, and shows what is necessary to alleviate it.

In addition, we also take advantage of other visualization tricks, such as stereographics, to add even more to our understanding of the 3D time volumes.

References

ADAMS05
`http://www.mscsoftware.com/products/products_detail.cfm?PI=413`

BAILEY01
Mike Bailey, "Interacting with Direct Volume Rendering", *IEEE Computer Graphics and Applications*, Volume 21, Number 1, January/February 2001, pp. 10-12.

BHATT03
Rajankumar Bhatt, Chin Pei Tang Leng-Feng Lee, and Venkat Krovi, "Web-based Self-Paced Virtual prototyping Tutorials," *Proceedings of ASME 2003 Design Engineering Technical Conferences*, Chicago, IL, September 2-6, 2003.

HICKS01
B.J. Hicks, A.J. Medland, and G. Mullineux, "A constraint-based approach to the modeling and analysis of packaging machinery", *Packaging Technology and Science*, Volume 14, Issue 5 , Pages 209 – 225.

HRONES51
Hrones & Nelson, *Analysis of the Fourbar Linkage*, MIT Press, 1951.

NAGCHAUDHURI00
Abhijit Nagchaudhuri, "Dynamic Modeling and Analysis of a Crank Slider Mechanism", *Proceedings of ASEE 2000 Conference*.

PFISTER99
Hanspeter Pfister, Jan Hardenbergh, Jim Knittel, Hugh Lauer, and Larry Seiler, "The VolumePro Real-Time Ray-casting System", *Proceedings of SIGGRAPH 99*, pp. 251-260.

SHREINER04
Shreiner, Woo, Neider, and Davis, *OpenGL Programming Guide*, Addison-Wesley, 2004

WESTOVER90
Lee Westover, "Footprint Evaluation for Volume Rendering", *Computer Graphics* (Proceedings of SIGGRAPH 90), Volume 24, Number 4, pp. 367-376.