I enjoyed reading this article. Having some experience in designing an early 2D to 3D representational drawing program, I never considered some of the details brought up in this research paper. When dealing with a change of viewing perspective, many new (to me) issues must be addressed. Some of the difficulties that were expressed had been considered and overcome in earlier applications such as stroke width. Why is stroke width not proportional to the distance from the viewing source scaled from the original size?

Since this is the first exposure for me to this type of drawing systems, I can’t really comment about the usability and control mechanisms. I think their rotational algorithm for bridged billboards might be flawed. In the application I worked on, when objects where “linked” to one another, a new virtual parent object was created that allowed scaling, rotation and other like permutation on the group as a whole, with the objects maintaining all relationships with one another. In general, the parent object keeps track of the extents of all its children. When rotation occurred, it was based upon the center of all objects in relation to each individual object. This is a hard concept to explain without given a graphical example.

I definitely like the way terrain mapping was implemented. It was unexpected, but very logical after a brief explanation. I would be interested in seeing a live demonstration of this system, as describing it in written for is difficult for simple systems that are interactive, much the less complex systems for new concept explorations.

In the article, times are given for the generation of scenes. This time data bears no relevance without also disclosing the specifications of the system that it is running on. Why is there no information about the platform the system works on? There is also no discussion as to what language the system was written in.