# Real-Time Shadows Report

# Description of shadow algorithm implemented

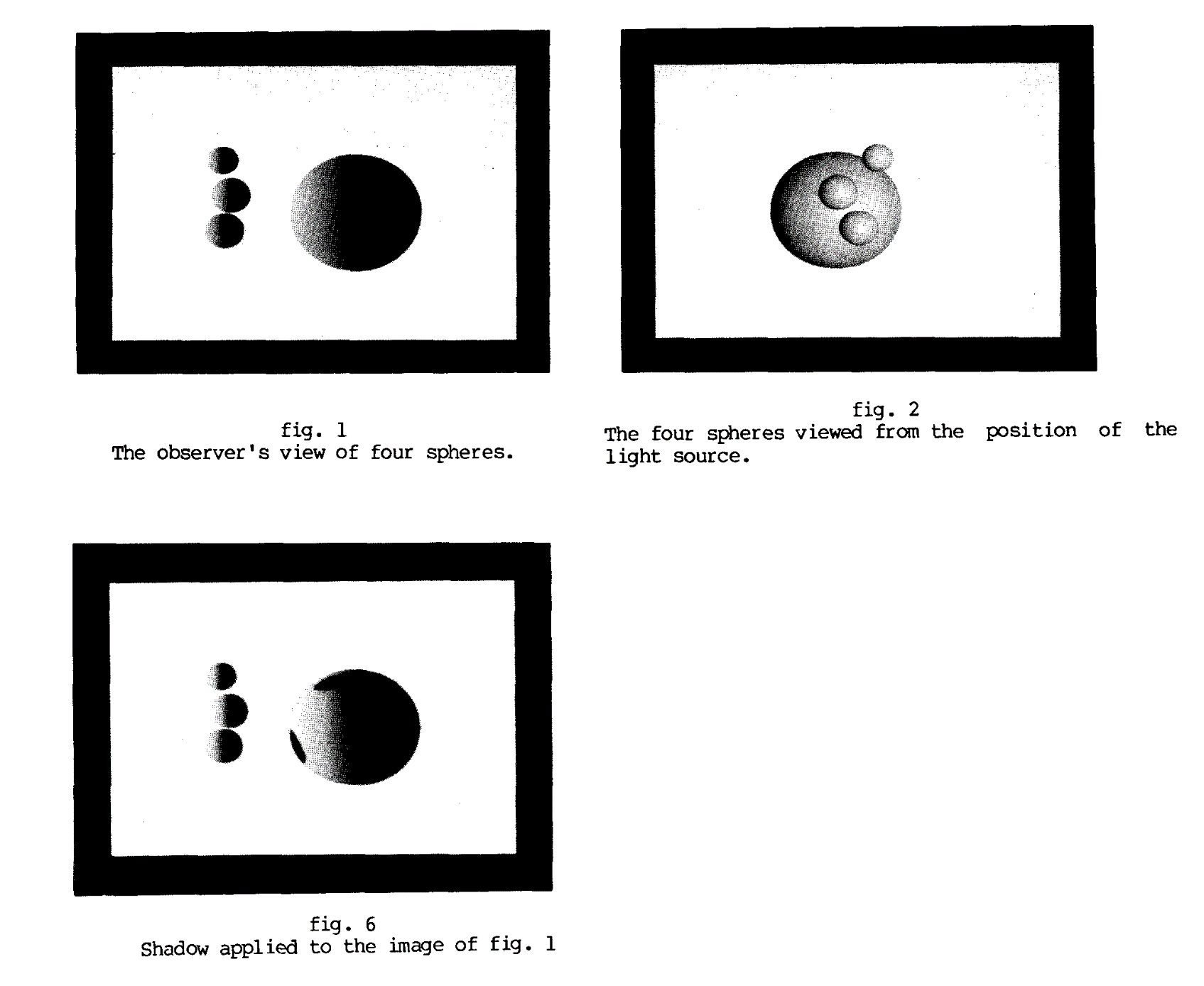
fg

# Advantages and disadvantages of different shadow algorithms

Two of the common algorithms used to implement dynamic shadows are shadow mapping (which was used in this implementation), and shadow volumes. Shadow mapping is used to add shadows to an existing 3D scene. The algorithm was first presented by Williams (1978). The main goal of the algorithm was to enable shadows to be cast on curved planes, as opposed to the flat planes required by simpler methods like planar shadows. The first step in shadow mapping is to construct a view of the scene from the perspective of the light source. Williams states that it is not necessary to calculate the shading values (colours); only the Z-values for each object are required. This data will be stored in the Z-buffer (or depth buffer).

The next step of the algorithm is to construct a view of the scene from the perspective of the “camera”, or observer viewpoint. Whenever the values for a pixel in the scene are calculated, it is necessary to transform the pixel’s X, Y, Z coordinates into light space and compare against the depth buffer, in order to check if the pixel is visible from the light source. If the pixel is not visible to the light source, then it is in shadow and should be shaded darker than normal. If the pixel is visible to the light source, then it is shaded as normal.

Williams (1978) provides an image of the output of the shadow mapping algorithm. The first image shows the observer’s view of the scene, the second image shows the image rendered to the depth buffer from the perspective of the light, and the third image shows the final output, with the shadowed regions calculated based on the pixel visibility.



Williams (1978) states a number of advantages for the shadow mapping algorithm. The first is that objects in the scene do not need to be sorted in order for the algorithm to work correctly, so the algorithm is useful for rendering complex scenes with many objects. The second advantage is that the algorithm is efficient; it scales linearly as the depth complexity of the scene increases. Williams states that in general, the rendering cost of the algorithm is roughly double that of rendering the scene normally (but not quite double, since there is no need to shade pixels or render to the screen when rendering to the depth buffer from the light’s perspective).

Another advantage of the shadow mapping algorithm is that it is flexible; performance can be adjusted by changing the resolution of the shadow map. A low-resolution shadow map will be faster to create, but the shadow quality will be reduced. Shadow mapping is based around rendering a depth map to a texture,

When compared to shadow volumes,

quality of shadow dependent on resolution of shadow map

*Shadow Volumes pros/cons*

*Generic – as many lights and objects as needed*

*Produces self shadowing*

*Precise shadow shapes in all kinds of geometry.*

*Very CPU intensive, especially as polygon counts of the shadow casting objects get high*

*Produces only "hard shadows".*

## References

Williams, L. (1978) ‘Casting curved shadows on curved surfaces’, *ACM SIGGRAPH Computer Graphics,* 12(3), pp. 270-274.