# APPENDIX

# Shading

E.H. Blake

*Centre for Mathematics and Computer Science*
*P.O. Box 4079, 1009 AB Amsterdam, The Netherlands*

## 1. What is shading?

Raster displays are very simple but versatile instruments for showing realistic moving pictures. They allow one to assign any one colour out of several million possible colours, individually, to any one of the million or so dots on the display screen. These dots, which are the picture elements or *pixels*, are redrawn every sixtieth of a second, and if the picture is in motion they actually change value about twenty times a second 'real-time' computer animation.

To exploit the potential of these devices various algorithms have been developed to assign the correct values to the pixels. If we consider the problem of displaying, or *rendering*, three-dimensional objects which are modelled (mathematically) in the computer then there are a number of operations which have to be performed:

- the objects have to be transformed geometrically and projected from three-dimensions onto the two-dimensions of the display
- the surfaces which are obscured by other objects or which would lie beyond the limits of the display 'window' have to be removed
- the visible surfaces have to be rendered on the display in a way which takes into account the modelled light sources.

The last operation involves *shading*. Shading mimics the effect of light from the modelled environment on the visible surface. Shading is the process which assigns values to the pixels of a visible surface by approximating the laws of optics. If objects are modelled as smooth surfaces it is often desirable to add textures to the rendered surface to achieve greater realism.

## 2. Shading techniques

Since the introduction of raster displays numerous shading techniques have been developed. There are very many pixels which have to be calculated repeatedly and so there is a trade-off between realism, with its associated computational cost, and speed.

The most realistic pictures are achieved by two rather expensive algorithms: *ray-tracing* and *radiosity* (see plate VII). Ray-tracing involves tracing the path of a light ray backwards from the pixel where it is going to end up, through its reflections and refractions from the surfaces in the environment, to the light source from which it originated. Radiosity algorithms calculate the diffuse light energy levels on all interreflecting surfaces of the model. Both techniques offer highly realistic pictures, though not perfect, and are very expensive computationally.

Gouraud shading and Phong shading (named after their originators) are less expensive techniques which often provide adequate levels of realism. They are particularly suited for many computer-aided design and scientific visualization applications.

The brightness of the surface depends on the characteristics of the light, physical properties of the surface, and geometry. The important aspects of the geometry are captured by various vectors which vary depending on position: the surface normal (perpendicular), the light direction, and the viewing direction. Light is reflected from a smooth (mirror) surface at the same angle to the normal as the incident ray. We will call this specular reflection. A real surface is not completely smooth and so diffuse light will also be reflected in all directions with the intensity depending only on the light direction and the surface orientation. It should be obvious that specular reflection is localized and produces 'highlights', while diffuse effects are more spread out.

If we consider a shape which has been approximated by means of planar polygons (see plate II) then we can still define normals at the vertices of the polygon which depend on the normals of the original figure. The directions of the light sources and the viewing direction can also be found at the vertices.

> In Gouraud shading we use these vectors to calculate the intensity of the pixels at the vertices of the polygon. The *intensity values* are then linearly interpolated across the polygon. See plate III.
>
> In Phong shading the vectors are found in the same way but it is the *vectors* which are interpolated across the polygon. At each pixel the shading calculations are performed to find its intensity. See plate IV.

Gouraud shading is cheap but it can lead to undesirable artifacts and omissions. If a highlight lies in the middle of a polygon then it will be missed with this technique. If things are moving then the highlight will be missed in some cases and not in others, and this produces a very noticeable scintillation. The other problem with Gouraud shading is that the rate of change of intensity makes sharp jumps between adjoining polygons. The intensities are continuous but not their derivatives. Unfortunately the eye is sensitive to this discontinuity: the effect produced is known as Mach banding.

Phong shading ameliorates, although it does not eliminate, the problems with Gouraud shading. The drawback is that several vectors have to be interpolated. The interpolation of a vector is a great deal more complicated than that of a scalar in that it involves divisions and square root calculations.

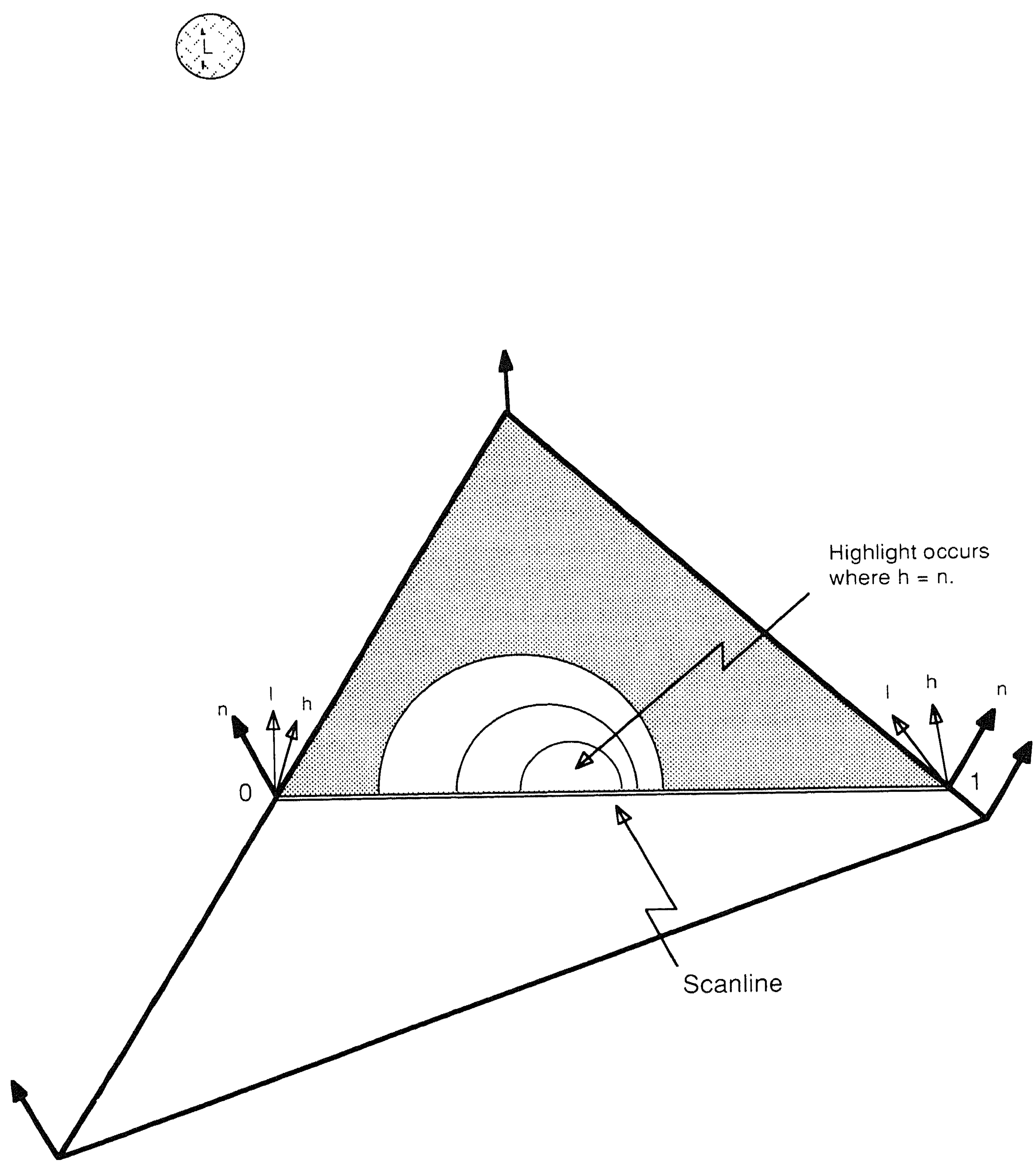


FIGURE 1.
The vectors involved in shading a single scanline. The diagram shows a triangle which is shaded line by horizontal line. The shading process has been interrupted about halfway through. The top part has been done and it is busy with a scanline which happens to cut a highlight. **n** is the normal used for shading, it is specified for the 3 vertices of the triangle. **l** is the direction of the light source (which is shown at the top of the figure). The vector **h** is halfway between the light and the viewer. In this figure the viewer is assumed to be somewhere above the page. The scanline proceeds from point 0 to point 1.

Shading techniques exhibit a clear trade-off between computational cost and realism. Realism is needed to convey information accurately but interaction and animation also require that a certain minimum processing speed be provided for every user. There is thus a continual pressure to provide realism which is computationally cheap.

### 3. Fast Phong shading

Gouraud shading can be implemented in hardware and so the high speeds necessary for real-time graphics can be achieved. In order to deal with the problems outlined above however one either has to make the polygons being shaded very small (and lose the speed advantage in that way) or switch to real-time Phong shading.

In the 'raster graphics on the basis of VLSI' project at the CWI new algorithms have been developed which allow Phong shading to be implemented in hardware. The essential idea is that interpolation on the unit sphere can be used instead of vector interpolation. This technique avoids complex calculations which have bedevilled Phong shading up to now.

The algorithm requires the incremental quadratic interpolation. This means that it is not only attractive for hardware interpolation but also offers advantages over existing software techniques on general purpose computers. See plate V.