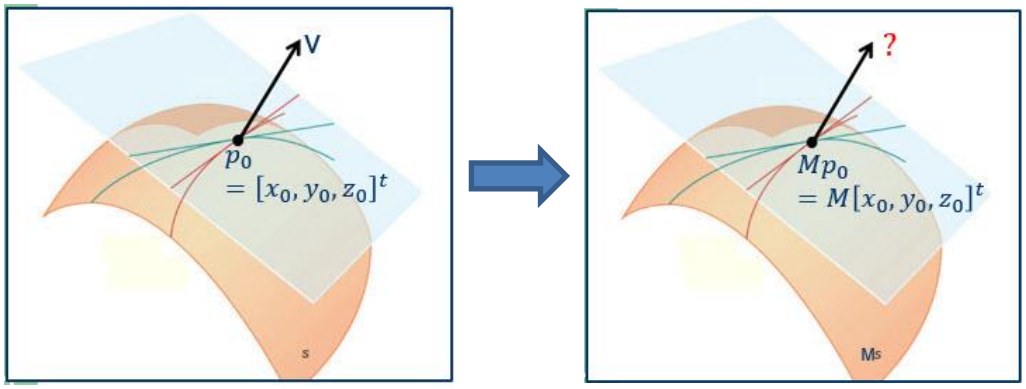
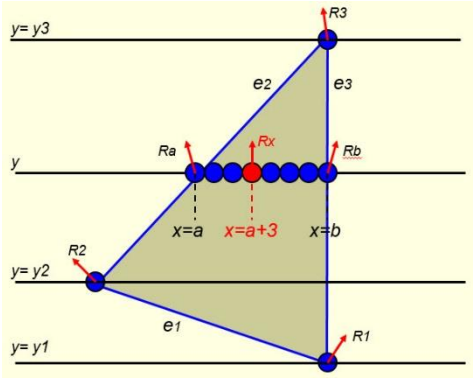
CS 535 Computer Graphics
Homework Assignment 5 (40 points)
Due: 11/01/2024

1. Given a point p_0 of a surface *S* with normal vector *V* (see the left figure below). If the surface *S* is transformed by an MV matrix *M* to a new location MS, then the normal vector of p_0 at its new location (see the right figure below) would be $(M^{-1})^t V$ where *V* is represented as a column vector. Why? (5 points)



2. Given the normal vector of a surface at a given point N and an incident ray L, we need to compute the specular reflection vector R at that point to compute its shade. Develop an incremental method to compute the specular reflection vectors for points of a triangle, assuming R1, R2 and R3 at the three vertices of the triangle are known to us. This method (also called **Fast Phong Shading**) is used by the Rasterization stage of the openGL pipeline. (5 points)



3. Gouraud shading (intensity-interpolation shading) and Phong shading (normal-interpolation shading) can both be used to eliminate intensity discontinuities when rendering a polygonal mesh. However, Gouraud shading could generate the so-called ***Mach band effect*** and Phong shading would not. Phong shading, nevertheless, is not as efficient as Gouaud shading because Phone shading has to compute the specular reflection vector R for each point of the polygon mesh, an expensive process. One alternative is to use the bisector of the angle between L and V, H, to compute the color/intensity, the so called Blinn-Phong shading. Another alternative is to use the above incremental method (Question #2) to estimate the specular reflection vector for each point of the triangle. Is Blinn-Phong shading a better choice? why or why not? (10 points)

4. The ***shadow volume based*** 'shadow generation' algorithm can be integrated with the scan-line hidden surface elimination process so that we can do hidden surface elimination and shadow generation at the same time. Can the ***shadow volume based*** 'shadow generation' algorithm be integrated with the Z-buffer method so we can also do hidden surface elimination and shadow generation at the same time, and how or why not? (10

points)

5. The ***shadow map based*** 'shadow generation' algorithm is easy to implement. But it has a potential problem. What is it? What is the reason for getting this potential problem? To overcome this problem, one way is to increase the resolution of the display surface. However, this is not really a solution, only a way to reduce the seriousness of the problem. What is a real solution to this problem and why would it work? (10 points)

- Solutions must be typed (word processed) and emailed to me both as a pdf file and a word document before 23:59 on 11/01/2024.
- Please name your files as:
  CS535_HW5_2024f_LastName.docx / CS535_HW5_2024f_LastName.pdf