6.837 Introduction to Computer Graphics
Final Exam

Tuesday, December 20, 2011 9:05-12pm
Two hand-written sheet of notes (4 pages) allowed

NAME:

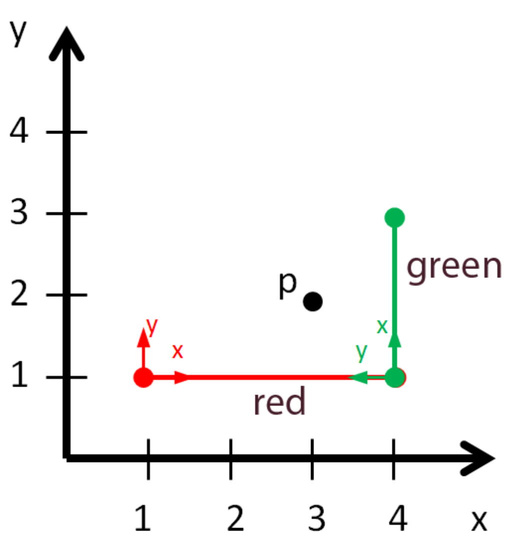
1	/ 17
2	/ 12
3	/ 35
4	/ 8
5	/ 18
Total	/ 90

1 SSD [/17]

In this problem we are going to animate a simple character using linear blend skinning.

1.1 Computing Bind Pose

We are given a skeleton and a skin mesh in a bind pose. Our character has only two bones (red and green) and we are interested in only one mesh vertex, p. See the diagram below.

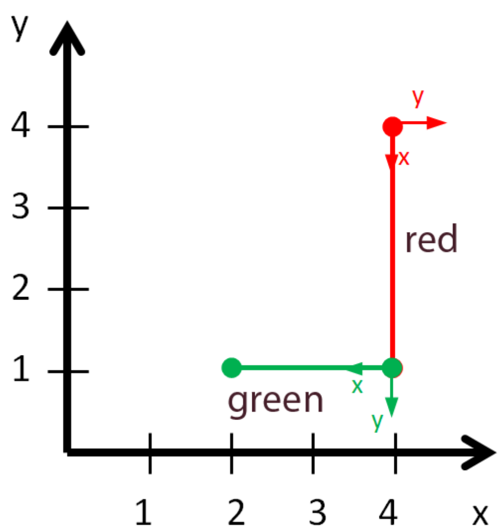


Compute rigid-transformation matrices $\mathbf{B}_{red}$ and $\mathbf{B}_{green}$ that transform mesh vertices from local bone coordinate system to the global coordinate system. [/4]

Then compute $\mathbf{p}_{red}$ and $\mathbf{p}_{green}$, the bone space coordinates of vertex $\mathbf{p}$ relative to the red and green bones, either geometrically or by inverting $\mathbf{B}_{red}$ and $\mathbf{B}_{green}$. [**/4**]

1.2 Bone Transformations

We have transformed each bone of this character according to the diagram below.



Compute matrices $\mathbf{T}_{red}$ and $\mathbf{T}_{green}$ that transform mesh vertices from local bone coordinate system to the global coordinate system. [**/4**]

1.3 Computing Vertex Positions

Using previously computed $\mathbf{p}_{red}$ and $\mathbf{p}_{green}$ and the new bone matrices $\mathbf{T}_{red}$ and $\mathbf{T}_{green}$, determine the transformed positions of vertex $\mathbf{p}$ in the global coordinate system, both for the red and green bone.

[**/4**]

Given that the weights for the red and green bone are 0.5, compute the final transformed vertex position in the global coordinate system. [**/1**]

2 Shading [/12]

Suppose we have a sphere centered at the origin, $x^2 + y^2 + z^2 = r^2$. There is a light source at (a,b,c). Generate a formula for finding the color at any point (x,y,z) on the surface of the sphere, assuming that there is diffuse reflection. Define any additional terms you introduce. [**/12**]

3 Ray Tracing [/35]

3.1 Refraction [/6]

Recall that the formula for the outgoing angle of a refracted ray is:

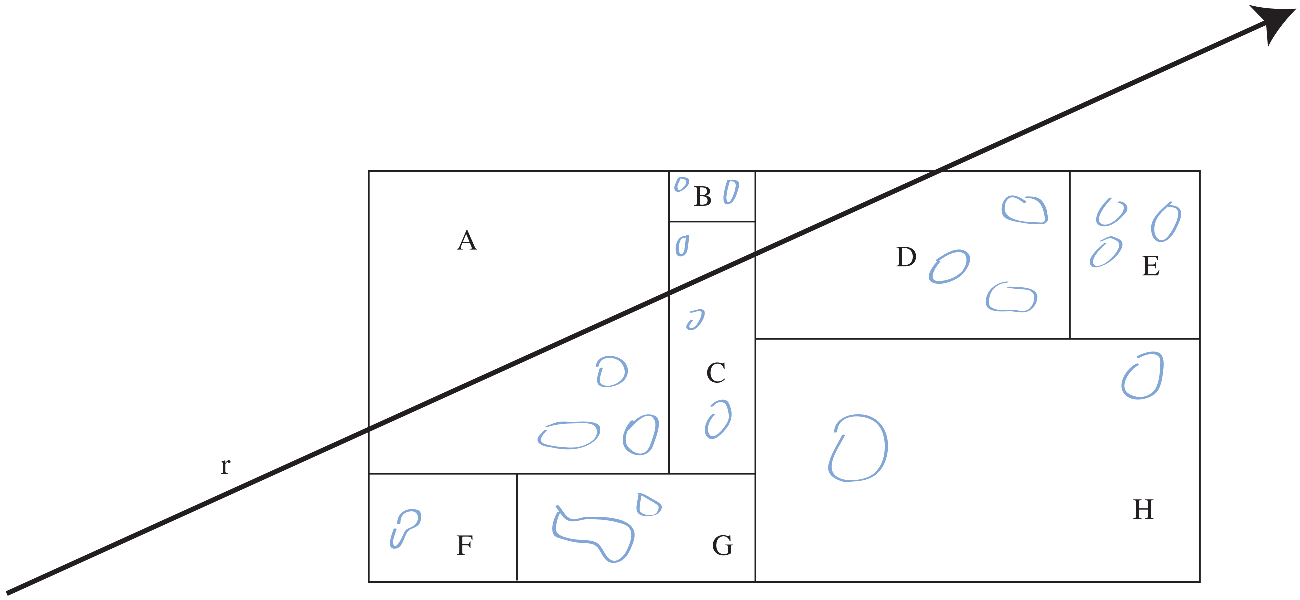
$$T = \left[\eta_r (N \cdot I) - \sqrt{1 - \eta_r (1 - (N \cdot I)^2)}\right] N - \eta_r I$$

What is the name of the physical phenomenon that causes the term under the square root to be negative? [**/3**]

How should we deal with the transmitted ray in such a case? [**/3**]

3.2 Kd-tree [/13]

Below is the representation of a given 2D Kd-tree with the leaves indicated by upper-case letters. We have drawn some leaf geometry in blue for motivation, but you do not need to consider it, albeit to notice that the particular ray r does not have any intersection with the scene.



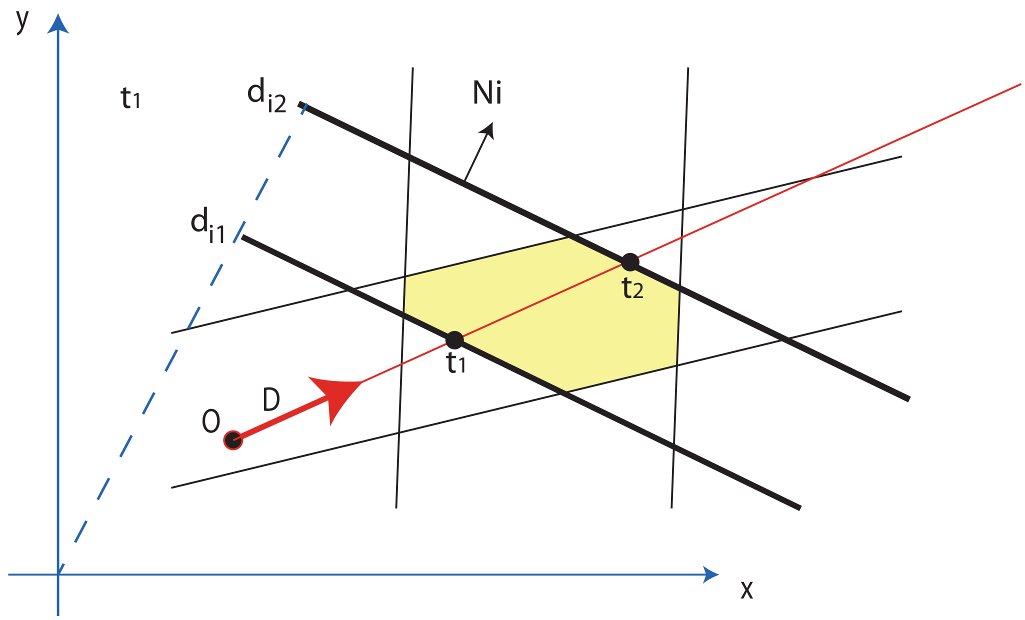
Draw the corresponding tree structure. [/5]

We now consider the traversal of this kd-tree for the ray r, as it would happen for ray-tracing acceleration. For warm up, draw the four intersections with the sides of the bounding box of the tree that occur during the initialization of the traversal. [/2]

We now want you to show the order in which ray-plane intersections are computed for the efficient hierarchical traversal of the tree. Draw a cross at each intersection point and write its order as a number next to it. NB: we want the order in which intersections are calculated, not the order along the ray. Make sure you use a smart traversal that only visits relevant nodes and that the order can enable early termination if appropriate. [/6]

3.3 Ray slab intersection [/16]

We seek to compute the intersection between a ray and a convex object defined as the intersection of a set of slabs. Slabs are the space between two parallel planes (see figure). A slab with index i is defined by a normal N_i and two real numbers d_{i1} and d_{i2}. The axis-aligned bounding boxes we studied in class are a special type of such objects where the three slabs have axis-aligned normals. We want to adapt the fast ray-box intersection algorithm to handle general slabs. We parameterize our ray as $P(t) = O + tD$ where O is the origin and D the direction. You can assume that the ray is going in the positive direction (i.e. t_1 is always smaller than t_2) and you should not worry about the ray being parallel to a plane or starting inside the slab.



First, we consider a single slab with index i. Write the equation for t_1 and t_2, the intersection parameters for the first and second plane delimiting this slab. [/6]

We now turn to the intersection of the ray and the CSG intersection of N slabs. We initialize t_{start} and t_{end} with the values for t_1 and t_2 given by the first pair of planes. Write pseudocode to update t_{start} and t_{end} with the values t_1' and t_2' for a new pair of planes. [**/6**]

Finally, after they have been updated to take into account all slabs, give a criterion on t_{start} and t_{end} that determines if the intersection between the ray and the volume is non-empty. Do not worry about whether the slab is in front or behind the origin.

[**/4**]

4 Rasterization [/8]

We want to implement two-scale rasterization where rectangular groups of pixels are quickly declared fully inside or fully outside a triangle. Assume you are given the three edge equations so that a 2D point P inside the triangle respects $P \cdot N_i - d_i > 0$ for $i = 0..2$. A rectangular region is described by its four corners P_j for $j = 0..3$.

What is the condition for the full rectangle to be entirely inside the triangle? [/**4**]

Things are more tricky for the test to be fully outside. A naïve solution would be to say that all four corners fail the edge tests. Find a counter example. [/**4**]

5 Graphics hardware [/18]

List one form of task vs. data parallelism in graphics hardware. [/4]

Example of task parallelism:

Example of data parallelism:

Attribute the following properties to either graphics hardware or CPU (we recommend against using the acronym GPU because we might have a hard time distinguishing your Gs and Cs :-(
[/6]

- optimized for latency

- latency hiding

- extremely long pipeline (1000 stages)

Would the following algorithm be implemented in a vertex or pixel shader? [/8]

SSD skinning

Phong shading

Blend shapes

Shadow map query

6.837 Computer Graphics
Fall 2012