TDA361/DIT220 Computer Graphics

EXAM

(Same exam for both CTH- and GU students)

Friday, January 13th, 2017, 08:30 - 12:30

Examiner

Ulf Assarsson, tel. 031-772 1775

Permitted Technical Aids

None, except English dictionary

General Information

Numbers within parentheses states the maximum given credit points for the task. Solutions shall be clear and readable. Too complex solutions can cause reductions in the provided number of points

Questions to examiner during exam

will be possible approximately one hour after the start of the exam. If anything is unclear – remember what has been mentioned on the lectures, in the slides and course book and do your best.

Grades

In order to pass the course, passed exam + exercises (or exam + project) are required. The final grade is calculated from the exam grade. The exam is graded as follows

CTH: 24p ≤ **grade 3** < 36p ≤ **grade 4** < 48p ≤ **grade 5**

GU: 24p ≤ **G** < 45p ≤ **VG**

Max 60p

Grades are announced by the LADOK system ~3 weeks after the exam

Solutions

will be announced on the course home page.

Review

Opportunity to review the correction of your exam is provided on Friday, February 10th at 12.00, room 4116 (next to my office room), 4th floor (west corridor), EDIT-building.

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

Question 1

a) **[1p]** What does the vertex shader do? I.e., what is its typical purpose?
Answer: (Computes vertex position and) projects from 3D to 2D. Precomputes per-vertex values, e.g., per-vertex lighting.

b) **[2p]** Explain what the z-buffer is used for and how it works.
Answer: Use a buffer called the z or depth buffer to store the depth of the closest object at each pixel found so far. As we render each polygon, compare the depth of each pixel to depth in z buffer. If less, place shade of pixel in color buffer and update z buffer.

c) **[2p]** The geometry stage: List all the spaces that a 3D vertex (typically) will be transformed through on its way from model space to screen space.
Answer: (0: model space/object coordinates), 1: world space, 2: camera/view space/eye coordinates, 3: projection space/homogeneous coordinates/clip space/unit cube, (4: normalized device coordinates/after perspective division/homogenization step), 5: screen space (window coordinates).

d) **[1p]** What is a rigid body transform?
Answer: Only rotations and translations (no change of size or shape).

e) **[2p]** Assume that you are creating a system where you have a rotation matrix **R**, a scaling matrix S, and a translation matrix T, to define an object's transform. For a vertex **v** of the object, show how you compute the transformed vertex **v'**. (Use the recommended order of the matrices. A one-line answer **v'**=... is enough.)
Answer: v' = (**TRS**) **v**

f) **[2p]** State a simple 4x4-matrix **M** that performs a projection. Also explain which plane your matrix projects onto and what type of projection it is.
Answer: Plane z=0. Orthographic (parallel) projection. M = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

Question 2

a) **[1p]** What is the difference between supersampling and multisampling?
Answer: multisampling only runs the fragment shader once per fragment (not once per sample-inside-each-pixel). I.e., multisampling shares computations, e.g. executes fragment shader for only one sample but takes several depth samples.

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

b) **[1p]** Draw the Quincunx pattern and state the weights per sample.
Answer: 1 sample in each of the four pixel corners (with weight=1/8) and 1 sample in the pixel center (with weight=0.5)

c) **[2p]** Assume that your graphics hardware supports the tent filter and sinc filter. You want maximum quality when enlarging your textures. Which filter would you choose? Motivate for points.
Answer: The sinc-filter is ideal. It uses all texels for each sample and cuts too high frequencies.

d) **[2p]** Which types of filters are common in real-time computer graphics for minification of 2D textures? (Each listed wrong filter will give negative score. Sum will however not go below 0p.)
Answer: nearest (box), biliear filtrering (tent) with and without mipmapping. Trilinear filtering with mipmapping, anisotropic filtering

e) **[2p]** What is a mipmap hierarchy?
Answer: A prefiltered hierarchy of halved (in both x- and y-) resolutions where each texel is the average of the 2x2 texels at the corresponding position of the lower level.

f) **[2p]** How do you prevent fully transparent parts of for instance a billboard to update the z-buffer when the billboard is drawn to the frame buffer (with standard z-test and updating enabled)?
Answer: you kill the fragments in the fragment shader, if alpha is zero.

Question 3

a) **[2p]** Assume that you have enabled backface culling. How does the hardware determine if a triangle will be backfacing or frontfacing? (Drawing an image can be useful.)
Answer: The vertices v0, v1, v2, are projected onto the screen and if they appear in anti-clockwise order (right-hand side rule), the triangle is (by default) assumed to be front facing. Otherwise, backfacing. (The order could be reversed by specifying CW-order.)

b) **[2p]** To draw transparent objects using for instance OpenGL, you typically divide the triangles in two groups: the transparent ones and the opaque ones. 1) Which of these two groups needs to be sorted, 2) why, 3) and in which order?
Answer: 1) the transparent triangles, 2) for correct blending, 3) back-to-front.

c) **[1p]** Normally, you would want to draw all geometry that is in front of the camera. So, why is a near and far plane used for the view frustum?
Answer: Near plane is used to avoid a degenerate projection plane. Far plane can be used to get better z-precision in the z buffer.

d) **[4p]** Describe a method for a ray-polygon intersection test. You may assume that the polygon is convex.
Answer: See lecture Intersections: Convert to a point-in-2D-polygon test and use the *crossing number algorithm / even-odd rule*.

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

e) **[1p]** Describe a test which determines whether or not a sphere and a plane intersect.
Answer: Insert the sphere center into the plane equation. If the result is smaller than the radius, there is an intersection.
I.e., **n•c** + d <= r.

Question 4

a) **[3p]** Write pseudo-code for hierarchical view frustum culling.
Answer:

b) **[2p]** Describe a top-down approach to build an Axis Aligned BSP-tree for a scene.
Answer: Create minimal AABB around the whole scene. Split along an axis. Recursively split the 2 parts alongs a new axis. Terminate if empty node, #triangles < threshold or level >= max recursion depth. (Axis aligned: x-,y-,z-axis are the split planes.)

c) **[1p]** What is a shadow cache? Explain what it is good for and how it works.
Answer: pointer to previous intersected triangle (primitive) by the shadow rays. Null, if last shadow ray had no intersection. Reason: speedup, potentially avoiding tree traversal.

d) **[4p]** Describe the structure of a typical ray tracer by using the functions main(), trace(), shade(), and findClosestIntersection(). I.e., describe what these functions do and which functions they call (who calls who).
Answer: main()-calls trace() for each pixel.
Trace(): returns the color of the closest intersected point. Calls findClosestIntersection() and then calls shade() for the point.
Shade(): computes color + calls trace recursively for reflection/refraction ray.

Question 5

a) **[2p]** What is a BRDF? (You can answer by stating what the abreviation BRDF stands for and assuming *f*() is a BRDF and describe its input parameters and what it returns.)
Answer: Bidirectional Reflection Distribution Function. The function $f(\omega_i, \omega_o)$ returns how much of the radiance from the incoming direction ω_i that is reflected in the outgoing direction ω_o.

b) **[3p]** Describe what makes Path Tracing efficient.
Answer: Spawning many rays at each bounce would result in a ray tree. Most work is spent on the many rays at the bottom of the tree which have least impact on the pixel. Path tracing only follows one randomly chosen spawned ray per bounce, resulting in a ray path. Instead, many paths are traced per pixel. The advantage is that an equal amount of rays is traced at each level of bounces. Better balance between spent work and importance.

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

c) **[2p]** Describe the advantages and disadvantages of shadow maps vs shadow volumes. You should mention at least a total of four important bullets (0.5p per unique bullet).
Answer: SM Pros: any rasterizable geometry, constant cost per rasterized fragment from the camera's view (basically just a texture lookup), fast.
SM Cons: jagged shadows / resolution problems, biasing.
SV Pros: sharp shadows.
SV Cons: 3 or 4 rendering passes (and thus often slower than shadow maps), lots of polygons and fill.

d) **[3p]** Describe the shadow map algorithm.
Answer: 1. Render a shadow (depth) map from the light source.
2. Render image from the eye. For each generated pixel, transform/warp the x,y,z-coordinate to light space and compare the depth with the stored depth value in the shadow map (at the pixel position (x,y).
If greater → point is in shadow.
Else → point is not in shadow.
(Bias/offset is necessary due to disretization and precision problems.)

Question 6

a) **[1p]** Sketch one non-continuous curve and one C^0-continuous curve (and mark which is which).
Answer:

c) **[2p]** In which ways are NURBS more general than B-Splines?
Answer: control points can be set at non-uniform intervals and they can have different weights.

d) **[1p]** Assume **p**=(4,9,0,2). Perform the homogenization step on **p**.
Answer:

e) **[1p]** Manually normalize the vector **x**=(0,4,1).
Answer:

f) **[2p]** There are 4 main taxonomies of hardware, based on where in the GPU-pipeline sorting in screen space is done. Which are these four? (Names are enough. You do not need to describe them if you don't want to.)
Answer: Sort-first: sort the triangles spatially before the geometry stage.
Sort-middle: sort the triangles spatially after the geometry stage.
Sort last fragment: sort the rasterized fragments spatially.
Sort last image: Have a frame buffer per pipeline. Merge the frame buffers after full rendering.
(See lecture 12.)

g) **[2p]** Assume that you are a hardware designer of a modern GPU working at some company. What would be the main point on your wish list: being able to increase the core clock frequency or being able to increase the memory bandwidth? **Motivate!**
Answer: A possible answer: you would typically want to increase the bandwidth. See graphics hardware lecture regarding bandwidth usage.

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY