NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Science

Examination paper for TDT4230 Graphics & Visualization

Academic contact during examination: Theoharis Theoharis

Phone: 73591447

Examination date: 16.5.2014

Examination time (from-to): 09:00-13:00

Permitted examination support material:

No written or printed materials allowed.
Simple (non-scientific) calculators are allowed.

Other information:

Time allowed: 4 hours.
The point scores for each question part are shown in the text. Total points: 50
READ EACH QUESTION THROUGH CAREFULLY BEFORE BEGINNING YOUR ANSWER.
ANSWER ALL QUESTIONS. SHOW ALL YOUR WORKING.

Language: English

Number of pages: 4 (including this page)

Checked by:

Date Signature

1. Theme: Parametric Curves [10pts]

Given the DeCasteljau triangle:

$$\begin{tikzcd}
\mathbf{p}_0 = \mathbf{p}_0^0 \arrow[r, "1-t"] & \mathbf{p}_0^1 \arrow[r] & \mathbf{p}_0^2 \arrow[r] & \mathbf{p}_0^3 = \mathbf{P}^3(t) \\
\mathbf{p}_1 = \mathbf{p}_1^0 \arrow[r] \arrow[ur, "t"] & \mathbf{p}_1^1 \arrow[r] \arrow[ur] & \mathbf{p}_1^2 \arrow[ur] & \\
\mathbf{p}_2 = \mathbf{p}_2^0 \arrow[r] \arrow[ur] & \mathbf{p}_2^1 \arrow[ur] & & \\
\mathbf{p}_3 = \mathbf{p}_3^0 \arrow[ur] & & &
\end{tikzcd}$$

a. What degree Bezier curve is this triangle made for? [2pts]

b. What is the Bezier curve point for t=0.5 given $\mathbf{p_0}=[0,0]^T$, $\mathbf{p_1}=[4,8]^T$, $\mathbf{p_2}=[8,16]^T$, $\mathbf{p_3}=[16,32]^T$? [4pts]

c. If the [0..1] range is sampled at 100 points to produce the curve, how many multiplications in total are used, assuming that the curve is in 3D space? [4pts]

2. Theme: Visualization Principles [10pts]

In the context of visualization:

a. Describe a filtering technique used to remove input data noise. [2pts]

b. What is a transfer function? [3pts]

In the following transfer function:

$$i_{\text{out}} = f_{\text{quant}}\left(f_{\text{contrast}}\left(i_{\text{in}}\right)\right),$$

$$f_{\text{contrast}}(x) = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot v_{\max},$$

$$f_{\text{quant}}(x) = x_{\min} + \frac{(x_{\max} - x_{\min})}{N} \cdot \left\lfloor N \cdot \frac{x - x_{\min}}{x_{\max} - x_{\min}} + \frac{1}{2} \right\rfloor$$

c. what is x_{min} and x_{max} and what do $f_{contrast}$ and f_{quant} do? [5pts]

3. Theme: Illumination [10pts]

In the Phong illumination model:

a. what is the reflection vector $\vec{\mathbf{r}}$, where is it used and what can it be replaced by?. [3pts]

b. what is the difference between Gouraud shading and Phong shading? [3pts]

In the ambient occlusion equation:

$$I_a(\mathbf{p}) = k_a I_a w(\mathbf{p}),$$
$$w(\mathbf{p}) = \frac{1}{\pi}\int_{\Omega} \mu(d(\mathbf{p},\theta_i,\phi_i))\cos\theta_i d\vec{\omega}$$

c. describe the output of the *d* function? [2pts]

d. what does *w*(*p*) represent? [2pts]

4. Theme: Visualization Algorithms [10pts]

Given the following implementation of the Marching Cubes algorithm:

```
Void MC()
{
        for (i= 0; i<maxcubeI; i++)
                for (j= 0; j<maxcubeJ; j++)
                        for (k= 0; k<maxcubeK; k++)
                        {
                                l1=get_label (i,j,k);
                                l2=get_label (i+1,j,k);
                                ...
                                l8=get_label (i+1,j+1,k+1);
                                index=l1++l2++l3++l4++l5++l6++l7++l8;
                                bindex=map_2_basic_index(index);
                                transform=map_2_basic_trans(index);

                                surface_list= precomputed_surfaces(bindex,transform^{-1});
                                for (p=0; p<num_vertices(surface_list); p++)
                                        compute_precise_edge_position(p,cube_field_values(i,j,k));
                                for (p=0; p<num_vertices(surface_list) p++)
                                        compute_normal(p, cube_field_values(i,j,k));
                        }
}
```

a. What does the command `compute_precise_edge_position(p,cube_field_values(i,j,k));` do? [2pts]

b. Why are there 8 labels l1…l8? [2pts]

c. What is transform? [2pts]

d. What is bindex? [2pts]

e. What is the output of the algorithm? [2pts]

5. Theme: Ray Tracing [10pts]

Given the following function of a ray tracer:

```
Color raytrace( Ray r, int depth, Scene world, vector <Light*> lights )
{
	Ray *refl, *tran;
	Color color_r, color_t, color_l;

	if ( depth > MAX_DEPTH )
		return backgroundColor;
	int hits = findClosestIntersection(r, world);
	if ( hits == 0 )
		return backgroundColor;
	color_l = calculateLocalColor(r, lights, world);
	if (r->isect->surface->material->k_refl > 0)
	{
		refl = calculateReflection(r);
		color_r = raytrace(refl, depth+1, world, lights);
		delete refl;
	}
	if (r->isect->surface->material->k_refr > 0)
	{
		tran = calculateRefraction(r);
		color_t = raytrace(tran, depth+1, world, lights);
		delete tran;
	}
	return color_l + color_r + color_t;
}
```

f. Which statement of this code is the most expensive to execute and what is its time complexity in terms of the number of rays *R* and objects *N*? [2pts]

g. What does the statement `raytrace(tran, depth+1, world, lights);` do? [2pts]

h. Which additional recursion termination condition can you think of? [2pts]

i. Which subroutine of `raytrace` includes the consideration of shadow rays and what do these rays do? [2pts]

j. How can the ray-object intersection tests be sped up using space subdivision? [2pts]