

Full Name: \_\_\_\_\_

Andrew Id: \_\_\_\_\_

## 15-462/662, Fall 2016

### Midterm Exam

Oct XX, 2016

#### Instructions:

- This exam is CLOSED BOOK, CLOSED NOTES (with the exception of your one post-it note).
- You MUST answer Problem 1, and can choose any 5 out of the remaining 6 questions. Please circle the questions you choose on the front page.
- The exam has a maximum score of 100 points. Answering the extra (non-circled) question can earn you up to 5 points of extra credit.
- If your work gets messy, please clearly indicate your final answer.

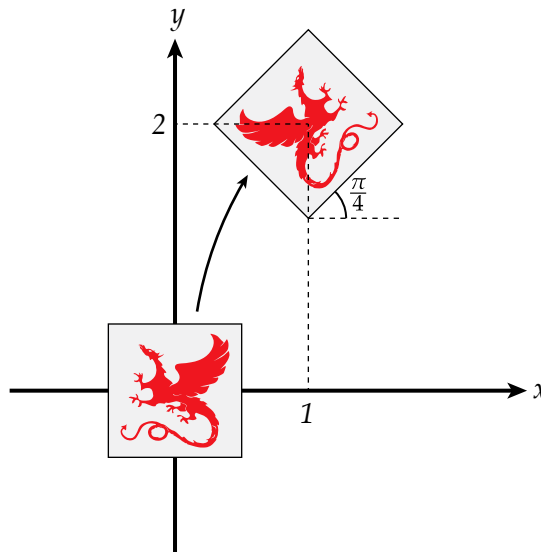
Problem	Your Score	Possible Points
1		20
2		16
3		16
4		16
5		16
6		16
7		16
Total		100

## Getting the Party Started: Miscellaneous Short Problems

### Problem 1. (20 points):

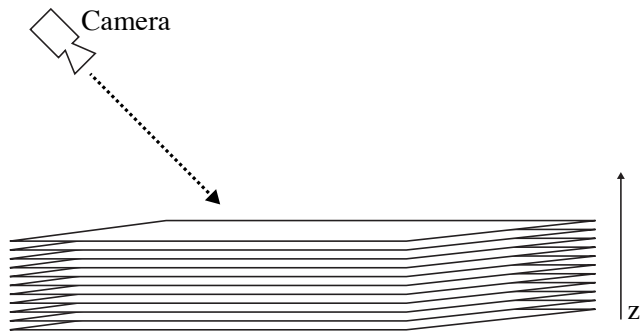
- A. (4 pts) You graduate from CMU and start making serious bank at your graphics startup that dominates the niche, but surprisingly lucrative, mesh simplification market. To reward yourself you buy a limo and take it to the shop to receive a window tint. Tint is applied by fixing 10% opaque ( $\alpha = 0.1$ ) layers of tint to your windows (more layers results in more light attenuation). If you want to have your windows reduce the amount of light entering the limo *by at least 20%*, what is the minimum number of layers of tint required?

- B. (4 pts) Describe in words any sequence of operations whose composition describes the overall transformation indicated by the figure below. (You do **not** need to write the matrices!)



C. (4 pts) In class we discussed the “wagon wheel” effect, where rapidly spinning tires on a moving car appear to spin backwards when viewed on TV or in a 30 fps video. Prof. Kayvon (correctly) claims that you’ve never witnessed this effect with your own eyes when looking at a moving car in broad daylight. (If you’ve seen the wagon wheel effect in the real-world, it would only have been outside at night under street lights, or indoors under cheap fluorescent bulbs). What is the cause of the wagon wheel effect, and why can it not be observed by the naked eye in daylight? (Optional: for kicks, why *might* it be observed by the naked eye if the scene is illuminated at night?)

D. (4 pts) You’ve just landed a job implementing the renderer for the latest *World Series of Poker* video game, and you’re trying to determine whether to use a rasterizer or a ray tracer. Consider the scene illustrated below with a camera looking down at a 45 degree angle on a deck of 5000 cards stacked in the +Z direction.



Consider the difference between rendering this scene using a rasterizer and a ray tracer: specifically think about the cost of determining what card is visible at each screen sample (or along each camera ray in the ray tracer). Do you think determining occlusion using the Z-buffer algorithm or via ray tracing is a better solution in this scenario? (Your answer can assume that the ray tracer has prebuilt a BVH or uniform grid to organize scene geometry.) Why?

- E. (4 pts) You place an object **that is a perfect mirror** reflector in a scene and try to render the object using the starter code for Assignment 3. For those that haven't started (grrrr...) the starter code evaluates the reflectance equation as follows:

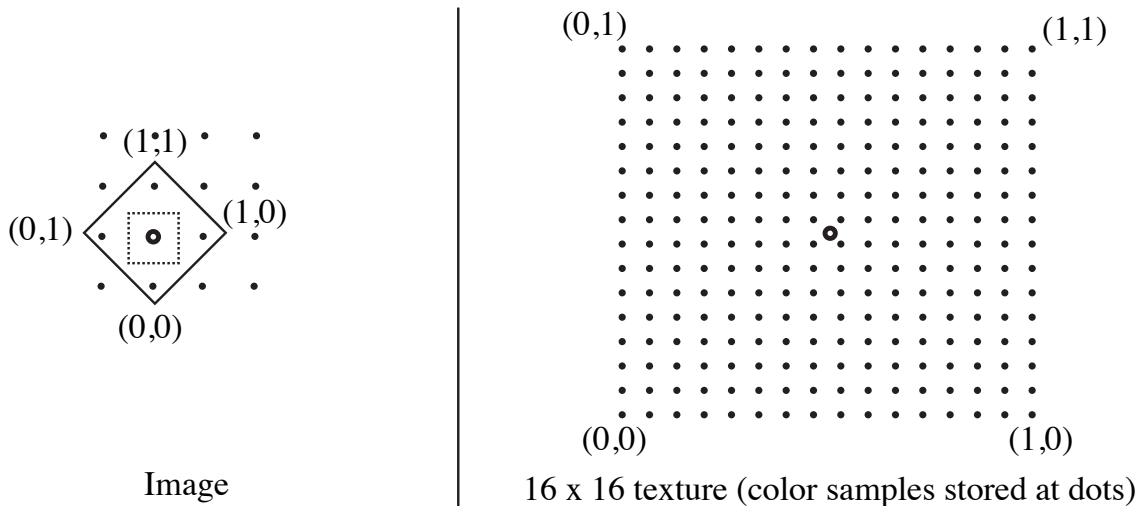
```
// to compute reflectance in the direction Wo
Spectrum Lo = 0;
for all reflection estimate samples:
    wi = generate random direction about surface normal
    Li = determine incoming light from direction wi
    Lo += brdf(w0, wi) * Li * cos(theta) / pdf;
return TWO_PI * Lo / num_samples;
```

Assume that the number of illuminance samples is four and the scene is lit with a uniform hemispherical light source so that incident radiance from any direction  $w_i$  is the constant  $L$ . You render a picture and the reflective object is black in nearly all pixels. Why?

## Texture Mapping

### Problem 2. (16 points):

Consider rendering the quadrilateral shown at left in the figure below. Per-vertex texture coordinates are given, and the dots indicate the position of screen sample points. Now consider the computation to compute the color of the quadrilateral at the highlighted screen sample point, which requires a texture lookup into the  $16 \times 16$  texture shown at right. The location in texture space of the desired sample from the texture function is also shown in the figure.



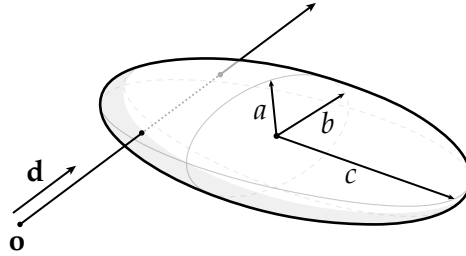
- A. (5 pts) Imagine you implement the texture lookup using a bilinear resampling (“bilinear filtering”) of the texture map. Assuming the texture image contains details such as many sharp edges, what might be the problem with this approach?

B. (6 pts) On the figure on the previous page, draw the bounds of the region in texture space corresponding to the dotted rectangle surrounding the shaded screen sample point (at left). If you wanted to properly filter (avoid aliasing) the texture sampling result, briefly describe an accurate, but slow algorithm for computing the resampled value of the texture at this specified point?

C. (5 pts) Consider using a mip-mip to accelerate the algorithm you proposed in the previous subproblem. What is one quality problem that might arise from using this more efficient solution?

## Rendering an Egg

### Problem 3. (16 points):



Consider an ellipsoid with radii  $a, b, c > 0$ , which can be expressed explicitly as

$$(\theta, \varphi) \mapsto (a \cos(\theta) \cos(\varphi), b \cos(\theta) \sin(\varphi), c \sin(\theta))$$

for  $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$  and  $-\pi \leq \varphi \leq \pi$ . It can also be expressed implicitly as

$$\mathbf{x}^T D \mathbf{x} = 1,$$

where  $\mathbf{x} = (x, y, z)$  and

$$D = \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{bmatrix}.$$

- A. (4 pts) Suppose you wanted to visualize the ellipsoid either by tracing rays or by rasterizing points. Which representation would you use in each case? Why?
- B. (7 pts) Now consider a ray  $\mathbf{r}(t) := \mathbf{o} + t\mathbf{d}$ . At what times  $t$  will this ray intersect the ellipsoid? Give an explicit formula in terms of  $\mathbf{o}$ ,  $\mathbf{d}$ , and  $D$ . When is this formula valid? (*Hint: quadratic formula!*)

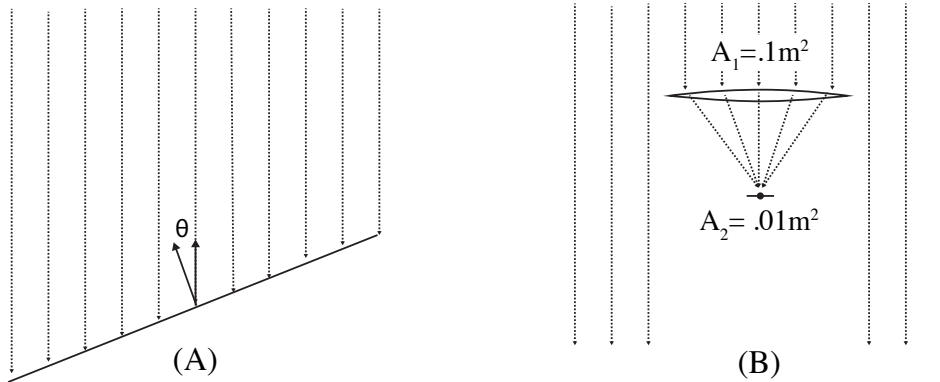
C. (5 pts) Suppose now that the ellipsoid is rotated via a matrix  $U$ . What's a **simple** way you can find the intersection with the rotated ellipsoid, *without* redoing all of your work from part (B)?



## Magnifying Glass

### Problem 4. (16 points):

Consider unidirectional sunlight falling on a tilted plane as shown in part (A) at left in the figure below (due to the large distance to the sun, all incident light is from the same direction). The plane is oriented with angle  $\theta$  relative to the incoming beam, and a measurement of **irradiance** on the plane yields the value 500 Watts/meter<sup>2</sup>.



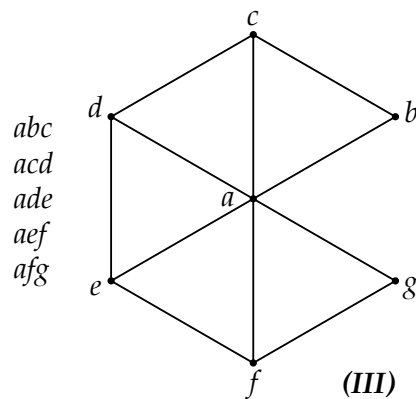
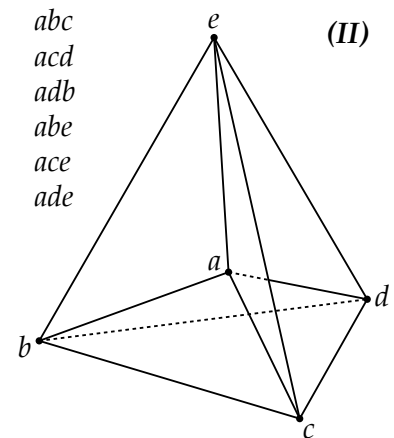
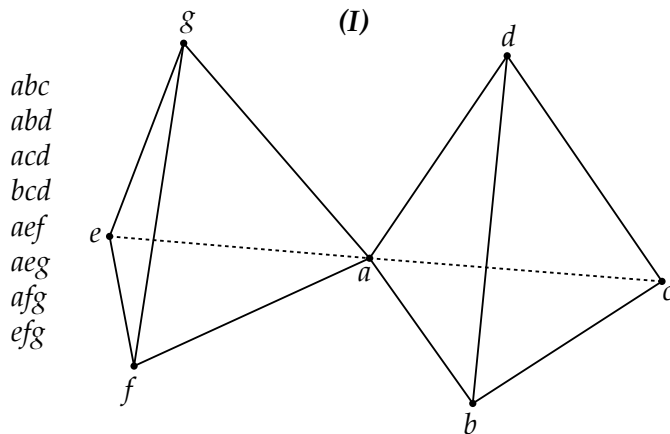
A. (8 pts) If you tilt the plane so that it is normal to the incoming beam, give an expression for the new irradiance on the surface (in terms of  $\theta$ )? Does surface irradiance increase or decrease?

B. (8 pts) Now consider the setup in part B of the figure where a magnifying glass with area  $A_1 = 0.1\text{m}^2$  has been placed about a small surface of area  $A_2 = 0.01\text{m}^2$ . Assume the surface of the magnifying glass and the surface of the receiving surface are flat planes that oriented directly toward the beam, and assume that the receiving surface is sufficiently small that it receives energy uniformly about its surface. Give an expression for the irradiance at points on the small receiving surface. (Hint: don't overthink this one, just consider the definition of irradiance! No integrals are required in this computation.)

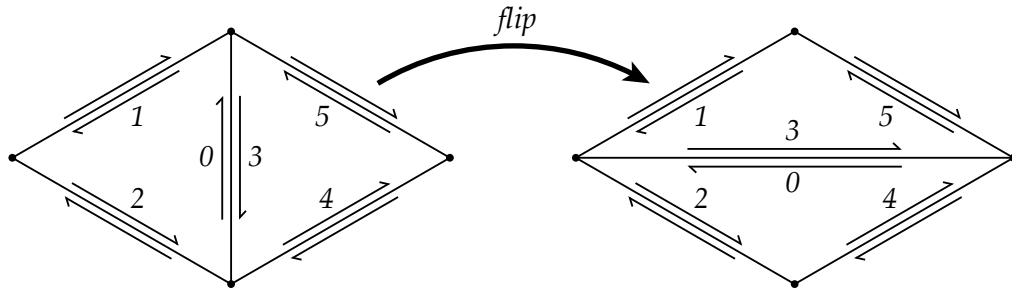
## Meshes and Manifolds

### Problem 5. (16 points):

- A. (5 pts) A triangle mesh has *boundary* if at least one of its edges is contained in only one triangle. A triangle mesh is a *manifold* if (i) every edge is contained in one or two triangles, and (ii) every vertex is contained in a single loop or fan of triangles. Using these definitions, indicate whether each of the examples below are manifold and/or have boundary. If a mesh is nonmanifold, specify one of its nonmanifold vertices or edges. If a mesh has boundary, specify one of its boundary edges.



- B. (5 pts) Consider a simplified halfedge data structure that does not explicitly encode vertices, edges, or faces—each halfedge keeps track of only its “next” and “twin” halfedges. This data is stored in two fixed-length arrays  $N$  and  $T$  of size  $2E$ , where  $E$  is the number of edges in the mesh. In particular, suppose we give each halfedge a unique index in the range  $0$  to  $2E - 1$ . Then for any halfedge  $k$ ,  $N[k]$  is the index of its next halfedge, and  $T[k]$  is the index of its twin. Update these arrays to reflect the edge flip below, using as few assignments as possible.



- C. (6 pts) Consider the same simplified halfedge data structure as in the previous problem. We can make this data structure even simpler by adopting the convention that halfedges  $2p$  and  $2p + 1$  are always twins, for  $p = 1, \dots, E - 1$ . This way, we don't even have to store the array  $T$ —just the array  $N$  of “next” pointers. In this way, any permutation of the integers  $0, \dots, 2E - 1$  specifies a valid manifold polygon mesh. Verify this claim by drawing a picture of the mesh defined by the  $N$  array below. (*Hint: draw the individual faces first, then think about how they connect up.*)

$k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$N[k]$	2	8	4	14	6	12	0	10	15	7	9	5	11	3	13	1



## BVH Refit

### Problem 6. (16 points):

One of the nice properties of a BVH is that it can be efficiently updated (“refit”) when a primitive in the bounding volume hierarchy is moved. This refit modifies the BVH so that the BVH property is maintained: the bbox of a node is the bound of the primitives contains in all child nodes. The refit **does not modify the structure of the BVH – only the bounding boxes of the nodes change.**

In the pseudocode below you’re given a definition of a BVHNode and interfaces for getting the bounding box of a primitive. Please implement the function `refitBVH(node)` below which should update bounding boxes in the BVH tree so that the BVH property holds **assuming that only scene primitives in the node node has moved.** To keep things simple, we’ve given you a pointer to the parent node `parent` in the BVH. You can assume that the `left` and `right` pointers of an interior are always non-NULL. A full-credit answer yields a BVH with the tightest bounds possible for the current tree.

```
struct BBox {
    void clear();           // resets bbox to empty
    void union(const BBox& b); // enlarges bbox to include volume in b
};

struct Primitive {
    BBox getBBox() const; // returns primitive's world-space axis-aligned bbox
};

struct BVHNode {
    BVHNode* left, *right;
    BVHNode* parent; // parent node in the BVH: NULL if root

    BBox bbox; // nodes bbox, you need to update this!

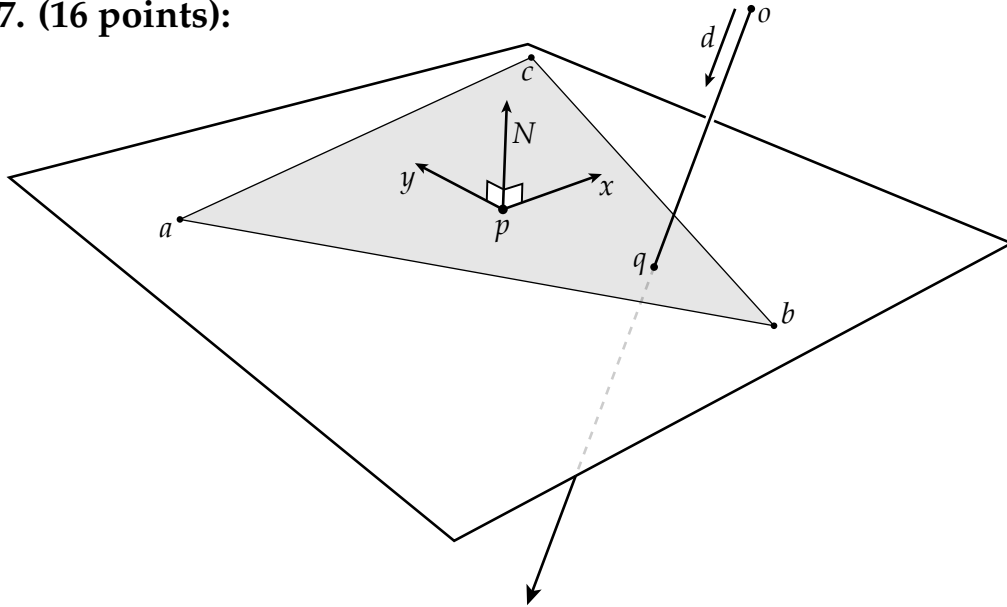
    int numPrims; // 0 if node is interior node, non-zero otherwise
    Primitive* prims; // prims[i]->getBBox() returns bbox of i'th prim in node
};

void refitBVH(BVHNode* node) {

}
}
```

Intersection

**Problem 7. (16 points):**



A. (4 pts) Suppose you have a plane passing through the point  $\mathbf{p}$  with unit normal  $\mathbf{N}$  and two orthogonal unit tangent vectors  $\mathbf{x}$  and  $\mathbf{y}$ . Write an equation for the plane in both implicit and explicit form.

B. (4 pts) Suppose you use the implicit plane equation to intersect a ray  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$  with the plane containing the triangle  $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ . If  $\mathbf{q}$  is the point of intersection, how would you test whether it is inside the triangle? Try to be as detailed as possible.

C. (4 pts) Now suppose you use the *explicit* plane equation to compute the ray-plane intersection. What equation(s) do we have to solve, for which variables? What are some pros and cons of computing the intersection this way?

D. (4 pts) Briefly describe how you might use your ray-triangle intersection routine to perform a triangle-triangle intersection test. In this setting, what is the significance of the value of  $t$ ? To keep the answer short, you can assume that triangles are not parallel.

