CS 184/284A | **Name:** ____________________

Spring 2016: Midterm | **SID number:** ____________________

3/14/16 | **WebAcct login:** ____________________

Time Limit: 80 Minutes

Note: this 2016 midterm has had 16 points of Camera and Lens questions added to it that did not appear in the original exam.

This exam contains 17 pages (including this cover page) and 5 problems. Check to see if any pages are missing. Enter all requested information on the top of this page, and put your initials on the top of every page, in case the pages become separated.

This exam is closed book, except for one 8.5×11 page of notes (double sided).
Show your work on each problem. Partial credit will be given.

- Note that the exam is 80 minutes long and there are 80 total points. This is to help you budget your time effectively.
- There is intended to be a large range in the difficulty of the problems. In particular, there are some easier problems scattered throughout the test, so don't spend too long on any one problem until you've at least read through the whole midterm.
- Answer each question in the space provided.

Problem	Points	Score
1	16	
2	22	
3	22	
4	20	
5	16	
Total:	96	

1. (Total : 16points) Warm Up: Miscellaneous Short Questions

   (a) (2 points) Your third assignment computes BRDF values in a coordinate system that is constructed using a normal vector to the surface and two other arbitrarily chosen perpendicular unit vectors. This means that the coordinate system has some unknown rotation around the surface normal. Which type of BRDF **cannot** be properly calculated given this fact?

   (i) Ideal specular.
   (ii) Ideal diffuse.
   (iii) Glossy specular.
   (iv) Anisotropic.

   Solution: (iv) depends on the rotation of the coordinate system with respect to the surface, but none of the others do. You used (i) and (ii) in your assignment, and (iii) could have been added.

   (b) (3 points) When rendering a Cornell box scene with a single area light overhead, Joe noticed that his shadows are very noisy. Which option would **not** make Joe's shadows less noisy?

   (i) Decrease the size of the area light.
   (ii) Make the area light twice as bright.
   (iii) Increase samples per pixel.
   (iv) Increase samples per light.

   Solution: (ii) will increase variance in the partially shadowed pixels by 4x.

   (c) (4 points) Each of these scenes presents rendering challenges. **Pick one** and explain **at least two** of the difficulties you would encounter when rendering your chosen scene, considering both its geometric and material attributes.

   (i) A person's head complete with skin and hair.
   (ii) A forest full of trees and plants.
   (iii) A city street with office buildings and cars.

   Solution: Example Answers:

   (i) Subsurface scattering of the skin; complicated light transport simulation and geometric intersection of the hair primitives.

   (ii) Translucent leaves and plants with subsurfacescattering; high geometric detail with millions of primitives; multiple levels of detail if viewing the forest from above so that the view goes all the way to the horizon; hard to denoise if viewing the dark interior of forest shadowed by many leaves.

   (iii) Complex BRDF effects on car paint; many-bounce interreflections from reflective building and car windows; potential level of detail issues if looking along a long street to the horizon; high geometric detail if many buildings viewed from above.

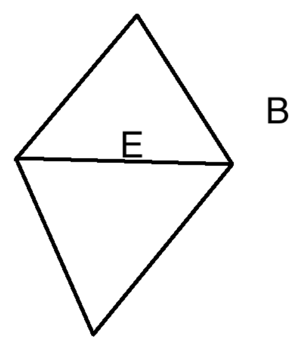
(d) (3 points) Give an example of a scene where a bounding volume hierarchy would perform much better than a uniform grid acceleration structure and explain why.

Solution: A galaxy scene where objects are far away from each other.
Example from class: detailed teapot inside an empty stadium.

(e) (4 points) Joe is trying to implement extra credit in his second assignment, and he wants to check if an edge is on the boundary of his mesh. He decides that he can just check whether the two vertices connected by that edge lie on the boundary. Help Joe understand the error of his ways by drawing a counterexample 2D mesh where at least one nonboundary edge connects two boundary vertices. Mark boundary faces with the letter B, and mark the nonboundary edge connecting two boundary vertices with E.

Some reminders: For a vertex to be on the boundary, there must exist at least one halfedge pointing to that vertex and also pointing to a boundary face. For an edge to be on the boundary, there must exist at least one halfedge pointing to that edge and also pointing to a boundary face. Boundary faces are allowed to have more than 3 edges.

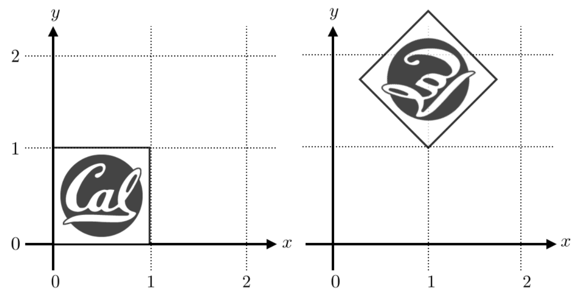
Solution:



The simplest case.

2. (Total : 22points) Rasterization Pipeline.

   (a) (4 points) Describe a sequence of basic transforms (translate, rotation, reflection) that you could execute in order to produce the following transformation. Note: you do *not* have to write the matrices for these basic transforms, only their parameters (for example, a single angle for rotation).



Before After

Solution: Multiple answers possible. One answer: reflect about the y-axis, rotate by 45 degrees clockwise, translate by $(1, 1)$.

(b) (5 points) Write down a 3×3 matrix representing the affine transformation from question 2(a) in homogeneous coordinates.

Solution:

$$\begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & +\frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

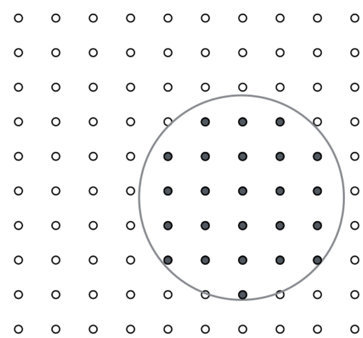
The simplest method to calculate this is to look at the effect of the matrix applied to the 3 homogeneous vectors $(0,0,1)^T, (1,0,1)^T$, and $(0,1,1)^T$.

(c) (5 points) Write out C++ code implementing a function

```
void rasterize_circle(float x_c, float y_c, float r);
```

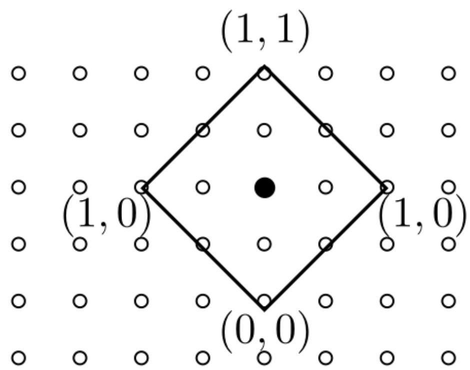
where the first two arguments provide the center of the circle and the third gives the radius. Assume access to a helper `rasterize_pixel(int x, int y)` that generates a fragment at pixel (x, y) after checking that the given point lies inside the valid range of the canvas. Perfect syntax and efficiency are not important.

An example of the desired result is given in the following figure, where each open dot represents a pixel where no fragment is generated, the large outline represents the circle to be rasterized, and the filled dots represent the pixels where a fragment is generated.

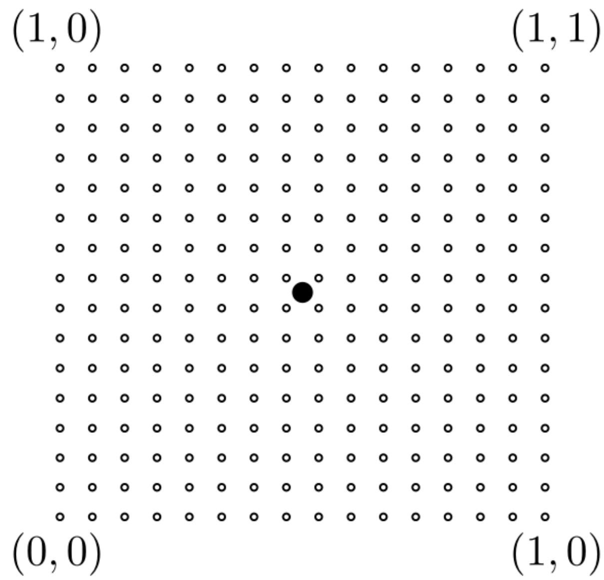


Solution: Iterate over each pixel (x, y) in the image. Test if $(x + 0.5 - x_c)^2 + (y + 0.5 - y_c)^2 < R^2$, and if so we consider the pixel covered and we generate a fragment for it. (Omission of the 0.5 offsets is ok.)

(d) (3 points) Consider drawing a small, texture-mapped quadrilateral with texture coordinates at each vertex as shown below left, covering the pixel grid as shown. The texture that is mapped onto this quadrilateral is the 16×16 texture as shown bottom right.



Image



16×16 Texture (color samples stored at dots.

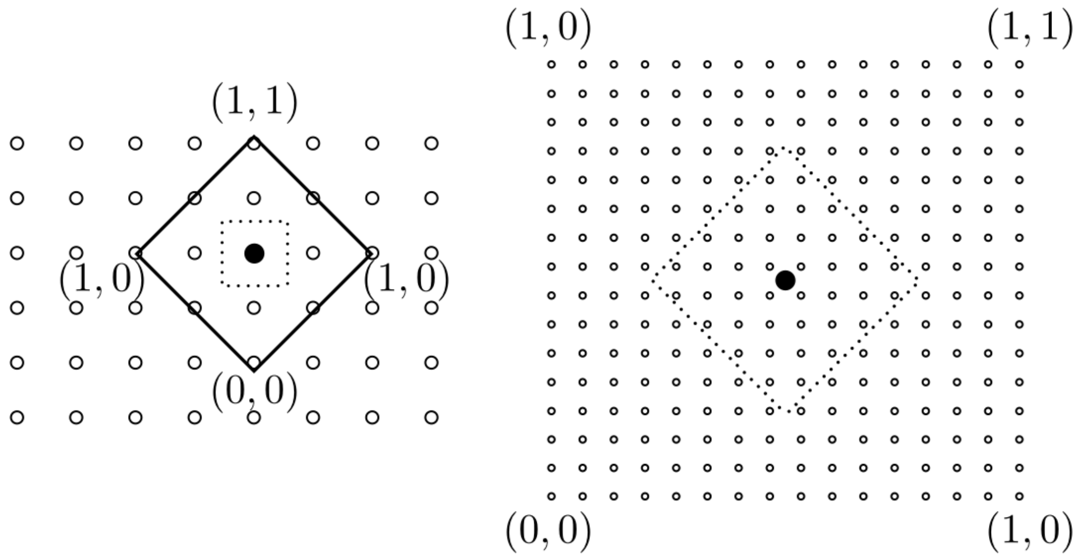
Consider the sample position on the output image shown by the black circle on the left. The corresponding location in the texture is shown on the right. If we use nearest texel sampling, *briefly* describe what kind of problems may occur and why they occur.

Solution: Aliasing! This is an example of highly minified texture mapping. If there are high-frequency features in the texture, such as fine lines, this may result in aliasing artifacts, such as moire, skipped lines, etc.

(e) (5 points) Continuing the previous question, consider a one-pixel square centered on the black sample on the left. Determine the area on the texture corresponding to this one-pixel square, and *outline this area on the texture grid on the right.*

Now describe a method for creating a high quality antialiased sample for the black sample position on the left, disregarding any concerns about efficiency.

Solution:



Image

16×16 Texture (color samples stored at dots.

We can average the texels shown in the diamond region on the texture.

Partial credit provided for mention of mipmapping.

3. (Total : 22points) Geometry

   (a) (5 points) Figure 1 shows some local structure of a mesh. To contract the edge connecting vertices V_0 and V_2, we need to first update some fields of those elements shown in the figure, then delete vertex V_2, halfedges $e_1, e_2, e_4, e_5, e_7, e_8$, and faces f_0, f_1. Fill in the blanks for how the following fields should be updated in order to minimize changes in the whole operation.

   $next(e_{11}) =$ ______ $halfedge(f_3) =$ ______
   $halfedge(V_0) =$ ______ $face(e_3) =$ ______
   $vertex(e_{12}) =$ ______.

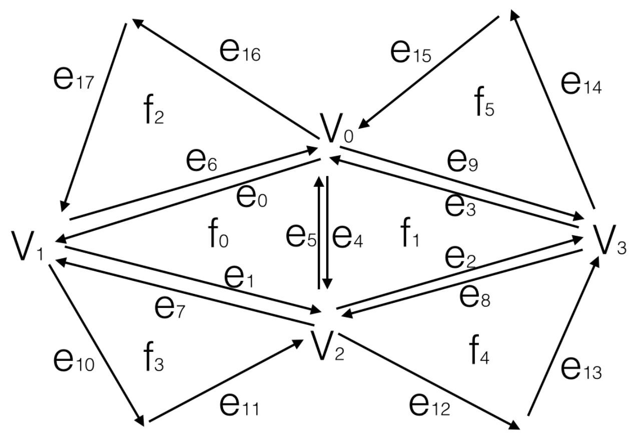


Figure 1: Edge Contraction

   Solution: $next(e_{11}) = e_0$ $halfedge(f_3) = e_0/e_{10}/e_{11}$
   $halfedge(V_0) = e_0/e_9/e_{12}/e_{16}$ $face(e_3) = f_4$
   $vertex(e_{12}) = V_0$.

   (b) (3 points) In figure 2, A, B, C, D are the 4 control points of a 2D Bezier curve $f(t), t \in [0, 1]$. Draw $f(\frac{1}{4})$ using de Casteljau's algorithm.

   Solution:

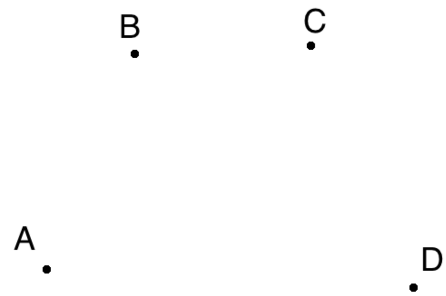
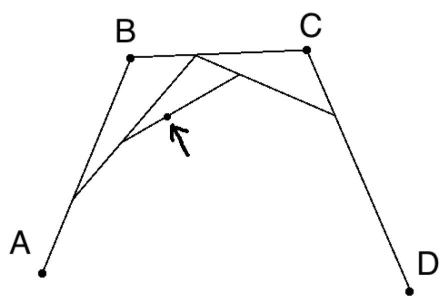


Figure 2: Bezier



(c) (6 points) Referring again to the diagram in the previous question, verify that the general point $f(t)$ obtained by de Casteljau's algorithm can be written in the following form:

$$f(t) = (1-t)^3 A + 3(1-t)^2 tB + 3t^2(1-t)C + t^3 D.$$

Solution:

$$AB = (1-t)A + tB$$
$$BC = (1-t)B + tC$$
$$CD = (1-t)C + tD$$
$$ABC = (1-t)AB + tBC = (1-t)^2 A + 2t(1-t)B + t^2 C$$
$$BCD = (1-t)BC + tCD = (1-t)^2 B + 2t(1-t)C + t^2 D$$
$$f(t) = (1-t)ABC + tBCD = (1-t)^3 A + 3(1-t)^2 tB + 3t^2(1-t)C + t^3 D$$

(d) (4 points) The 2D Delaunay triangulation algorithm maximizes the minimum angle of all triangles by flipping edges of a 2D mesh. In each iteration of the iterative version of 2D Delaunay triangulation algorithm, we need to traverse across all edges and flip some edges based on a condition. Figure 3 shows 3 examples of meshes before and after Delaunay triangulation. What condition do you think we should use to decide whether to flip a given edge?

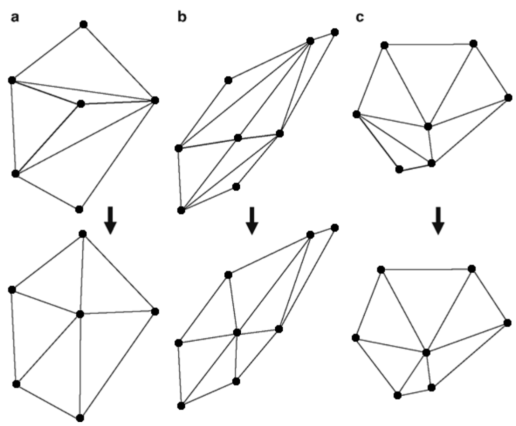


Figure 3: Delaunay Triangulation

Solution: Find the 6 angles of the 2 adjacent triangles of the edge before and after the flip. Flip an edge if it will make the minimum angle bigger.

Or flip an edge when the circumscribe circle of one of its two adjacent triangles contains all vertices of the other adjacent triangle.

(e) (4 points) When we transform an object, different transformation matrices should be applied to its vertices and normals. Show that if the transformation matrix for an object is M, then the transformation matrix for its normals should be M^{-T} as in figure 5. Remember that X^{-T} denotes the transpose and inverse of matrix X. (Hint: show that if you take any vector that is perpendicular to the original normal, the transformed vector and transformed normal will still be perpendicular.)

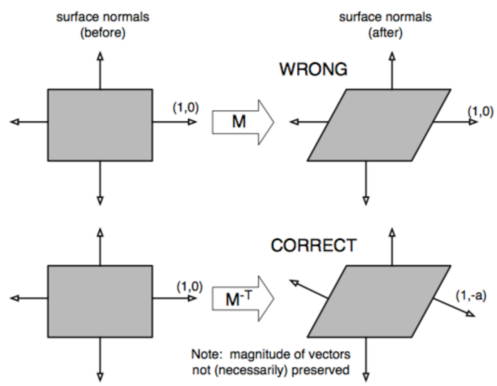


Figure 4: Normal Transformation

Solution: For any v such that $n^T v = 0$, then we see that

$$(M^{-T} n)^T (Mv) = n^T (M^{-1} M v) = n^T v = 0$$

so the transformed normal and transformed vector are still perpendicular.

4. (Total : 20points) Rendering

   (a) (5 points) In order to beat the curse of dimensionality, we turn to Monte Carlo integration with random samples. With Monte Carlo, we use the standard deviation (square root of variance) as a measure of expected noise levels.

   Prove that when taking n independent random samples $X_1, \ldots, X_n$ with the same variance σ^2 and letting $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, the standard deviation of $\overline{X}$ is

$$\sqrt{var(\overline{X})} = \frac{\sigma}{\sqrt{n}}.$$

Solution:

$var(\overline{X}) = var(\frac{1}{n}\sum\limits_{i=0}^{n} X_i) = \frac{1}{n^2} var(\sum\limits_{i=0}^{n} X_i)$

Because $X_1, ..., X_n$ are iid,

$var(\sum\limits_{i=0}^{n} X_i) = n\sigma^2$

$var(\overline{X}) = \frac{\sigma^2}{n}$

$\sqrt{var(\overline{X})} = \frac{\sigma}{\sqrt{n}}$

(b) (6 points) Considered the function $p(x) = \frac{1}{x^2}$. This is a probability density on the interval $[1, \infty)$ because it is nonnegative and integrates to one.

Write out C++ code implementing a function

```
float SampleInvSq();
```

which samples a point in the interval $[1, \infty)$ with probability density p. You are given access to a helper function `RandFloat()` that samples a random floating point number in the range $[0, 1]$. Perfect syntax is not important.

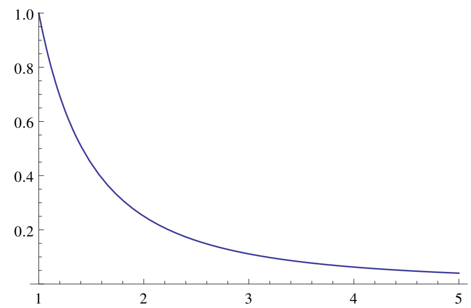


Figure 5: Plot of $\frac{1}{x^2}$

Solution:

1) Get CDF:

$\int_1^x \frac{1}{t^2} dt = -\frac{1}{t}\Big|_1^x = 1 - \frac{1}{x}$

2) Inverse:

$y = 1 - \frac{1}{x}$

$x = \frac{1}{1-y}$

3) Code:

```
float SampleInvSq()
{
    return 1/(1-RandFloat());
}

or

float SampleInvSq()
{
    return 1/RandFloat();
}
```

(c) (4 points) Suppose I have a cube that measures 1 m on each side, and I cover all 6 faces with solar panels that are 100% efficient. At noon, the sun lies directly overhead, casting rays of light perpendicular to the ground such that any point has incident irradiance 1 kW/m^2.

If I go outside at noon, what is the maximum energy in kilojoules (kilowatts are kilojoules per second) that my cube can collect in one second?

(i) 1 kJ
(ii) $\sqrt{2}\ kJ$
(iii) $\sqrt{3}\ kJ$
(iv) 2 kJ

Solution: When the cube's longest diagonal is perpendicular to the ground, the cube will have projected area $\sqrt{3}\ m$ perpendicular to the sunlight, so it will collect $\sqrt{3}\ kW$ of power.

(d) (5 points) Consider an empty Cornell box (a cube with 5 faces filled and one empty, with the camera facing into the box and a single white area light on the ceiling). All four of its walls and its floor and ceiling are perfectly diffuse with the same fixed reflectance values for all wavelengths of light (i.e., gray). Joe renders the box with his assignment 3 program using full path tracing (paths terminated only by Russian roulette) to get both direct and indirect illumination. Now, Joe divides the perfectly diffuse BRDF of the walls and ceiling by two and multiplies the intensity of the light by two, then renders a new image. Will the two images be different? Justify your answer.

Solution: Not the same. One bounce image will look the same. But for multiple bounces, $L(K)^n \neq 2L(K/2)^n$ when $n > 1$ and $0 < K < 1$.

5. (Total : 16points) Cameras and Lenses

(a) (4 points) Which of the following cameras has the widest horizontal field of view on its focal plane? You can assume the cameras are all focused at infinity.

(i) Cellphone camera with 6mm wide sensor and 5mm focal length lens.

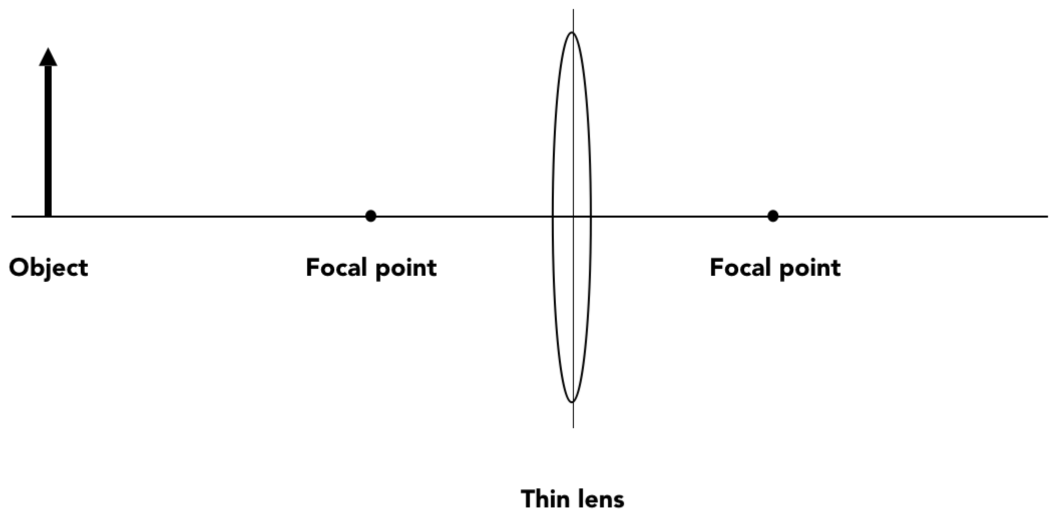
(ii) Full-frame camera with 36mm wide sensor and 50mm focal length lens.

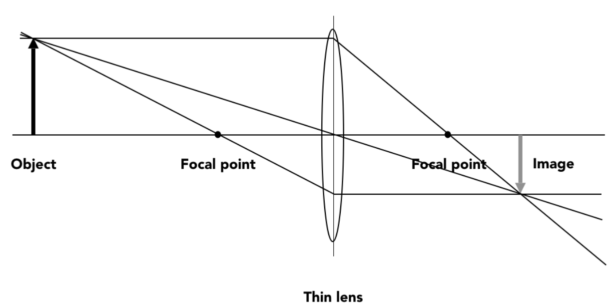
(iii) APS-C sized camera with 24mm wide sensor and 12mm focal length lens.

(iv) They all have the same horizontal field of view.

Solution: The answer is (iii), because it has the widest angular field of view and we are focused at infinity.

(b) (4 points) Given the object and thin lens shown below, use Gauss' ray construction method to draw the image that would appear.

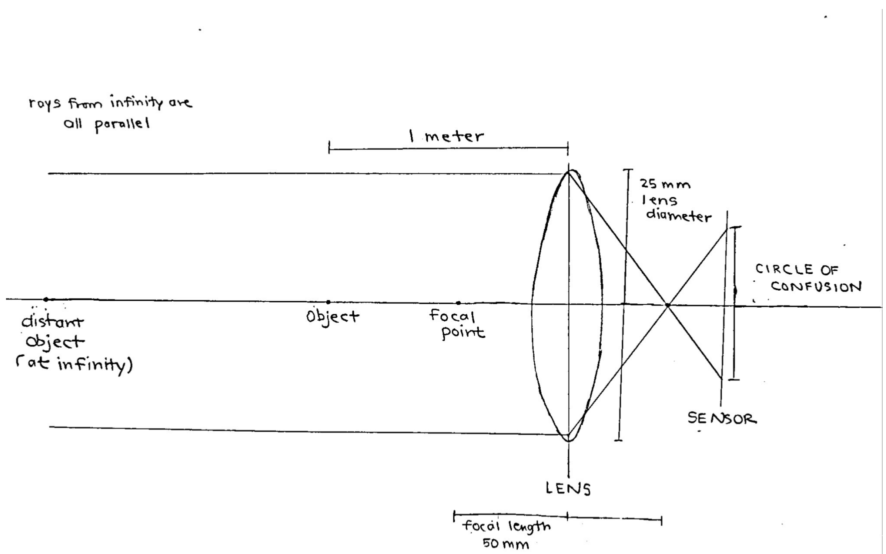




Solution:

(c) (8 points) Assuming a regular single-lens camera with an ideal thin lens with 50 mm focal length, with an f/2.0 aperture, and which is focused 1 meter away. Now consider objects very far away (i.e. at optical infinity).

i. Draw a ray diagram representing this problem that includes the following items and labels them: a distant object, an object at 1 meter, lens, focal point, sensor, and rays from distant objects and label the circle of confusion. Your drawing should be neat and clear, but does not need to be drawn to scale.



Solution:

ii. What is the size of the circle of confusion (in mm) that these distant objects make on the sensor surface?

(i) $2.0 \times \left(\frac{1}{\frac{1}{50.0}-\frac{1}{1000.0}} - 50.0\right)$ mm

(ii) $\frac{1}{2.0} \times \left(\frac{1}{\frac{1}{50.0}-\frac{1}{1000.0}} - 50.0\right)$ mm

(iii) $2.0 \times \left(\frac{1}{50.0} - \frac{1}{1000.0}\right)$ mm

(iv) $\frac{1}{2.0} \times \left(\frac{1}{50.0} - \frac{1}{1000.0}\right)$ mm

Solution: The answer is (i)