Stanford CS248A: Computer Graphics: Rendering, Geometry, and Image Manipulation
Exercise 6

Understanding Path Tracing

Problem 1:

Consider the path tracing algorithm to compute global illumination discussed in class (slide 59 of lecture 13). The code is copied below for convenience, WITH ONE MODIFICATION INDICATED IN THE CODE COMMENT BELOW: **When sampling the bounce direction, the implementation ALWAYS CHOOSES AS RANDOM RAY DIRECTION, REGARDLESS OF THE CURRENT BSDF**.

```
Spectrum PathLo(Ray ray) {
  Spectrum Lo = 0, beta = 1;
  int depth = 0;
  while (true) {
    Intersection isect = scene->Intersect(ray);
    Vector3f wo = -ray.d;
    if (depth == 0) Lo += isect.Le(wo);

    BSDF bsdf = isect.GetBSDF();
    Lo += beta * ReflFromDirectLighting(bsdf, wo);

    Spectrum fr = bsdf.Sample_f(wo, &wi, &pdf);  // ASSUME THE IMPLEMENTATION OF Sample_f
                                                 // ALWAYS CHOOSES A RANDOM RAY DIRECTION
                                                 // FROM THE HEMISPHERE ABOVE THE SURFACE
                                                 // NORMAL WITH PROBABILITY 1/2PI.

    beta *= fr * Dot(wi, isect.N) / pdf;

    float q = 0.25;
    if (randomFloat() < q) break;
    else beta /= (1-q);

    depth++;
    ray = Ray(isect.P, wi);
  }
  return Lo;
}
```

Consider a scene that contains only point light sources (lights that are infintesimally small points). Why is it the case that although the code above is an unbiased global illumination ray tracer, with extremely high probability, perfect mirror surfaces in the scene will appear black?

Questions about Color

Problem 2:

A. It's nearing the end of the quarter and to blow off some steam you decide to go out clubbing to celebrate the end of CS248A! You check the club's website and learn that tonight is "yellow light night", where the entire dance floor is illuminated in yellow-looking light that is emitted from light sources that have red, green, and blue primaries. Your friend, who is in a glum mood, says, "I find it hard to party because I'm so sad that CS248A is almost over! I wish I could wear black tonight to show off my feelings, but I only have red shirts and blue shirts to choose from." You tell your friend, "Oh you can still look like you are wearing black!" Which shirt to you advise your friend to wear, and why?

B. Give one reason why color representations that explicitly separate the luminance (brightness) and chroma components of a color (e.g., hue-saturation-brightness (HSB) or Y'CbCr) can be useful color representations compared to RGB.

C. Imagine the human visual system could directly measure and interpret the full spectrum of incident light. (That is, your brain received and used full spectral information $L(\lambda)$ rather than just the response of S,M,L-cones). Why would this change to human perception make recording and displaying digital images and rendering pictures far more challenging? (Hint: consider reproducing the appearance of a real world scene on a display. The word metamer might be useful.)

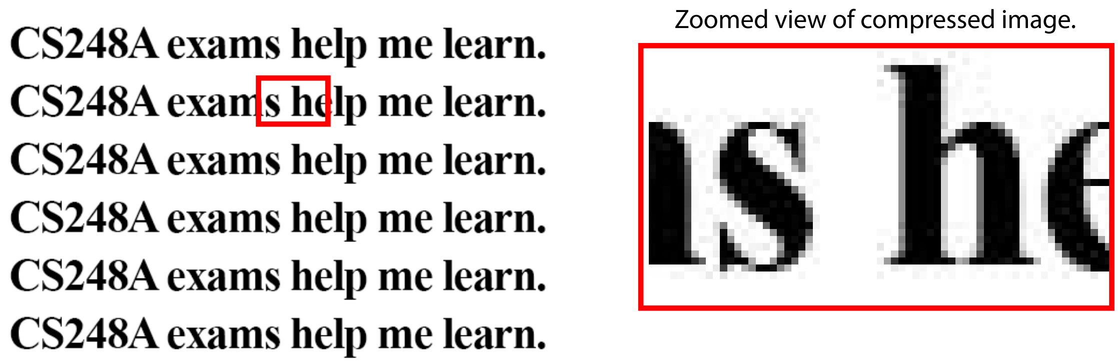
D. Describe the biological reason why, even though a spectrum may contain power over all wavelengths, human perception of color is only three-dimensional. We'd like to see the phrase "response function" in your answer.

E. Imagine if all three types of cone cells in your eye had *the same spectral response function*. If this were the case, would you have color vision (the ability to differentiate different colors)? Why or why not?

Compressing Images for the Web

Problem 3:

A common rule taught to graphics designers and web developers is “never save images as JPG files if they contain text”. Below is an image that contains text, along with a zoomed view. Please briefly summarize why a compressed JPG file exhibits these compression artifacts for images that contain text. Mention key properties of the input image in your answer.



Diving into a Path Tracer

OPTIONAL PRACTICE PROBLEM 1:

Consider the following pseudocode for a path tracer. It is the same code shown in lecture. It employs Russian Roulette to terminate paths with probability q.

```
// return radiance along ray 'ray'
Spectrum PathLo(Ray ray) {
  Spectrum Lo = 0, beta = 1;
  int depth = 0;
  while (true) {
    Intersection isect = scene->Intersect(ray); // find scene intersection
    Vector3f wo = -ray.d;
    if (depth == 0)         // *** QUESTION IS ABOUT THIS LINE ***
      Lo += isect.Le(wo);

    BSDF bsdf = isect.GetBSDF();

    // accumulate reflectance due to direct lighting
    Lo += beta * ReflFromDirectLighting(bsdf, wo);

    // generate new ray direction wi, and evaluate BSDF given wo and wi.
    Spectrum fr = bsdf.Sample_f(wo, &wi, &pdf);

    // update path throughput before next step along path
    beta *= fr * Dot(wi, isect.N) / pdf;

    float q = 0.25;
    if (randomFloat() < q)
       break;         // terminate path
    else
       beta /= (1-q); // update path throughput

    depth++;
    ray = Ray(isect.P, wi);
  }

  return Lo;
}
```

A. Notice the line of code with the comment `*** QUESTION IS ABOUT THIS LINE ***`. Please explain why the algorithm only accumulates surface emission into the path's output radiance if the path depth is 0. Note that path depth = 0 means this is a camera ray.

B. In your own words, why is it generally a good idea to special case the sampling of direct lighting (like in the code above), rather than implement the simple form of path tracing psuedocoded on slice 46?

A Highly Irregular Rasterizer

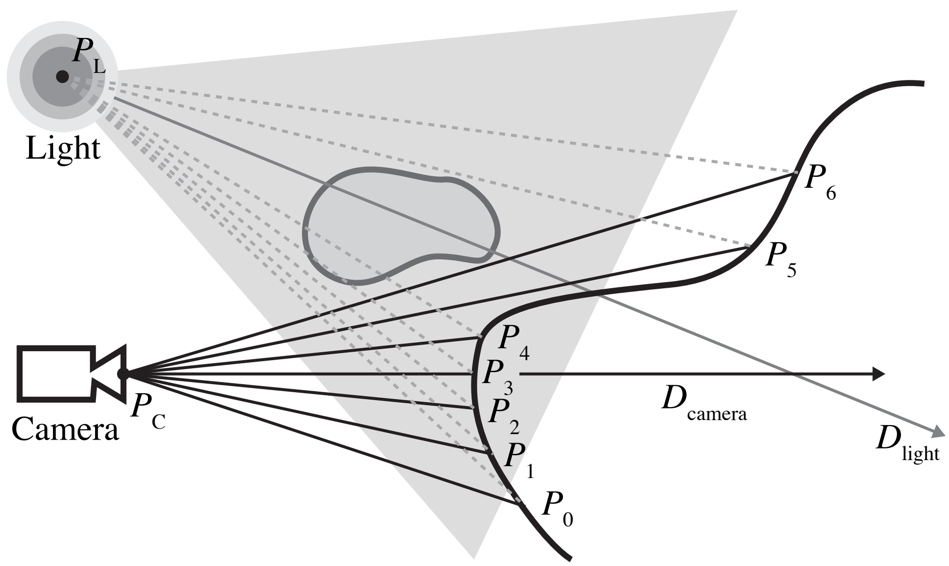
OPTIONAL PRACTICE PROBLEM 2:

Imagine that you have a special kind of rasterizer which doesn't evaluate depth/coverage at uniformly spaced screen sample points, instead it evaluates depth/coverage **at a list of arbitrary 2D screen sample points provided by the application**. An example of using this rasterizer is given below. In this problem you should assume that depths returned by `fancyRasterize` are in WORLD SPACE UNITS.

```
vector<Point2D>  myPoints;  // list of 2D coverage sample points: in [-1,1]^2
vector<Triangle> geometry;  // list of scene triangles in WORLD SPACE
Transform worldToCam;       // 4x4 world space to camera space transform
Transform worldToLight;     // 4x4 world space to light space transform
Transform perspProj;        // 4x4 perspective projection transform


// this call returns the distance to the closest scene element from the camera
// for all points in myPoints (assume infinity if no coverage)
vector<float> depths = fancyRasterize(geometry, myPoints, worldToCam, perspProj);
```

You are now going to use `FancyRasterize` to render images with shadows. Consider the setup of a camera, scene objects, and a light source as illustrated below.



A. Assume you use a traditional rasterizer to compute the depth of the closest scene element at each screen sample point. In the figure, the closest point visible under each sample when the camera is placed at position P_C and looking in the direction D_{camera} is given by P_i. All points in the figure are given in **world space!**

Assume you are given a *world space to light space* transform `worldToLight`. (Light space is the coordinate space whose origin in world space is P_L and whose -Z axis is in the direction D_{light}.) Describe an algorithm that computes, for each point P_i, if the point is in shadow from a point light source located at P_L. Your algorithm accepts as input an array of world space points P_i, world space points P_C, P_l, and has access to all variables listed in the example code. The algorithm should call `fancyRasterize` only once. (No, you are not allowed to just implement a ray tracer from scratch.)

Hint: Be careful, `fancyRasterize` wants points in 2D (represented in a space defined by the $[-1, 1]^2$ "image plane") so your solution will need to describe how it converts points in world space to a list of 2D sample points in this plane. **This involves transformation, perspective projection via `perspProj`, then convert from a homogeneous 3D representation to 2D.**

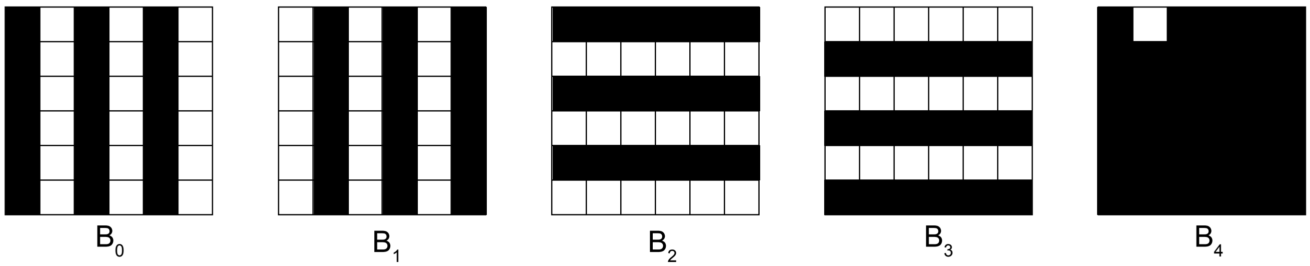
B. Does the algorithm you gave above generate "hard" or "soft" shadows? Why? (You can answer this question even if you did not correctly answer part A—just assume a solution that does what was asked in part A exists.)

C. Prof. Kayvon quickly looks at the algorithm you devised above and waves his hand dismissively. He says, "remember I told you in class that shadow mapping is such a hack", it only yields an approximation to ray traced shadows. The CAs jump in and shout, "Wait a minute here, this algorithm seems to compute the same solution a ray tracer would to me!" Who is correct? Why?

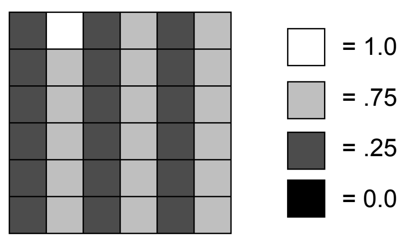
A Weird Compression Scheme

OPTIONAL PRACTICE PROBLEM 3:

In class we talked about how JPG compression represents 8×8 pixel image patches in the 2D *cosine basis*. Now consider a very different image representation scheme that represents 6×6 pixel patches in terms of a linear combination of these five base patches.



A. Consider representing the following 6×6 image in terms of the base patches above. What are the coefficients of the image under this representation? Explain why, **for the specific case of this image**, we have devised a very efficient image compression scheme. (Hint: what is the size of the 6×6 image in the pixel basis? What about the size of the representation in terms of the image patches above?)



B. Although the compression scheme in part A can be very efficient for some images, the problem with the scheme is that is cannot accurately represent all 6×6 images. Draw one example of an image that cannot be represented as a combination of the provided "basis" images.

Image Compression

OPTIONAL PRACTICE PROBLEM 4:

A. In the image compression lecture I showed you an example where I added a significant amount of noise to an image and, as a result, the compressed size of the resulting JPG compressed file grew substantially. (Assume both files were compressed with the same JPG quality setting, or in other words, the same quantization matrix.) Describe why the image with more noise compresses less.

B. Which image do you think will be compressed to a smaller file size using JPG compression? Please describe why. Your explanation should reference the state of the coefficient matrix after the quantization step. What properties of each image make one more compressible than the other?

A

B