

**Spring 2007**  
**22C:251 Advanced Computer Graphics (Image Synthesis)**  
**Assignment 4**

**Due: Before class on Tuesday, April 3rd**

**Goal:** Implement one (or two if you are a group) interesting things to extend your ray or path tracing code. You should choose one (or more) that relate to your final project for the class.

**Problem 1:** (50 points) Implement one of the following things if you are an individual, or two if you are working in a group.

1. An object instance class.
  - This will allow you to load a single complex object (like the Buddha) and render it in many different locations and orientations in the scene.
  - For full credit, you should be able to load instanced objects from a file.
  - For full credit, you must either be able to load nested instances from a file *or* be able to dynamically update the instance matrices to allow for simplistic motion.
2. The ability to render motion blur.
  - The idea is real cameras are open for more than an instant, so objects may move during that period.
  - You must be able to define via your scene file:
    - (a) Which object move.
    - (b) How they move (this can be very simple, like “along this line”).
    - (c) How fast they move.
    - (d) How long the “camera” is open.
  - You then need to associate a “time” with each ray (in addition to a (x,y) location on screen), as a particular ray may hit an object at one time, but not another.
  - To get a smooth motion blur, you’ll need to sample multiple times per pixel, where each sample uses a different time.
3. A new material class based upon either the Ward, Lafortune, or Ashikhmin-Shirley BRDF models. (You may implement two different BRDF models as your two choices, if you are a group).
  - Look at the “Physically Based Rendering” or “Advanced Global Illumination” books or talk to me if you cannot find details about these models.
4. Implement texture mapping.
  - This should probably be implemented as a new material class (which may be derived from an existing material type).
  - For full credit, you must implement 2D texture mapping for all primitive types (triangles, spheres, and either infinite planes or parallelograms).
  - For full credit, you must also implement at least one of: 3D texture (where a pixel’s color depends on the value stored in a 3D volume), bump mapping (where the texture modifies the surface normal), composite material types (where the texture specifies what type of material is present at that location – so white in the texture might mean Lambertian and black might mean a mirrored surface), or a light probe textured background (using light probes from <http://www.debevec.org>, and you must use the high dynamic range probes).