Raytra Scene File Format

CS4160

October 24, 2022

1 Command File Language

Note: We will provide test files, but you can (and should) make your own as well. The command file consists of a series of lines which describe geometry, camera, lights, or materials.

Only this last one is context-sensitive: a defined material applies to any geometry that is defined subsequently, until it is replaced by another material definition.

All commands are unique from a single letter - the only variant of this is that the "lights" command ("l") has a first argument which says which kind of light to make (note also that lights don't take a material definition - the light color and intensity is encoded in 3 floats).

You may extend the command language as you wish, in order to accomplish any "extras" you implement, but you must stay backwards-compatible; *i.e.*, you must be able to read and render from a source file in the format below, so that we may test your renderer's basic capabilities.

All points, scalars, or vectors are given as floats, with distances in meters. r/g/b values are encoded as floats with range [0 1] for material colors, and light color and intensity are both encoded in the r g b triple, with minimum 0 and unbounded maximum (although it's reasonable to choose 1 has a nominal value).

Here are the possible commands:

Comment:

/ Any line starting with / should be ignored

Geometry:

/ sphere at position x y z with radius r:
s x y z r

/ triangle with counterclockwise point order:
t x1 y1 z1 x2 y2 z2 x3 y3 z3

/ plane with normal n and scalar value d:
p nx ny nz d

/ triangle mesh
w meshpath

Camera:

/ camera at position [x y z] looking in direction [vx vy vz], with focal length d,
/ an image plane sized iw by ih (width, height) and number of pixels pw ph.
c x y z vx vy vz d iw ih pw ph

Lights: (note second parameter to denote which kind of light)

/ a point light at position x y z, color & intensity coded as r g b
l p x y z r g b

/ a directional light with direction vx vy vz and color & intensity coded as r g b
l d vx vy vz r g b

/ the ambient light (there will be, at most, only one of these):
l a r g b

Materials:
When a new material is defined, all subsequently-created geometry will be assigned that material.

/ Phong material with solid color:
/ defined by diffuse components [dr dg db], specular components
/ [sr sg sb], ideal specular components [ir ig ib], and with "roughness"
/ or Phong exponent "r"
m dr dg db sr sg sb r ir ig ib

/ textured Phong material:
/ defined by texture id "ti", diffuse components [dr dg db], specular components
/ [sr sg sb], ideal specular components [ir ig ib], and with "roughness"
/ or Phong exponent "r"
/ (note: the texture referenced by this material must be defined first)
n ti dr dg db sr sg sb r ir ig ib

/ dielectric Phong material:
/ defined by index of refraction ior and attenuation components [dr dg db]
d ior dr dg db

Textures:

/ image texture with id "ti" and path "path".
/ "flipx" and "flipy" are integer values in $\{0, 1\}$, specifying whether to flip the image
/ horizontally or vertically, respectively.
i ti flipx flipy path

Options:

/ for your own additions and coding, you may add option flags on a single line in this way
o myopt1 myopt2 etc.

(for example, you might want to be able to switch shadows on/off by including a "shadows" option. Leaving the "shadows" option out will tell your renderer not to do the shadow computation, etc.)