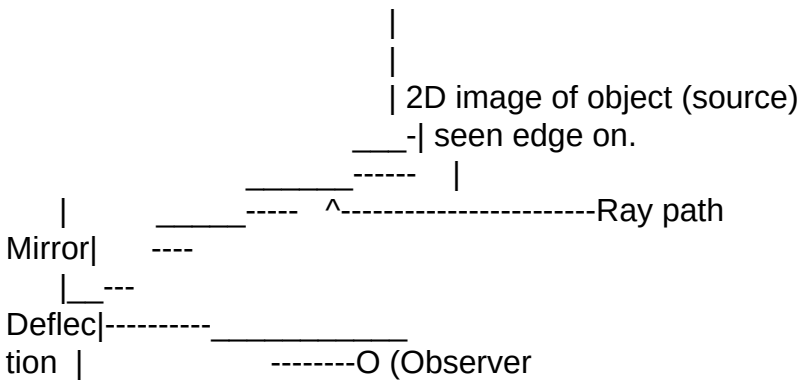
Subject: Re: mirror distortions
Posted by steinhh on Mon, 31 Oct 1994 14:55:20 GMT
View Forum Message <> Reply to Message

In article <1994Oct29.164623.11269@timessqr.gc.cuny.edu>, rmb@CUNYVMS1.GC.CUNY.EDU (Robert Braham) writes:
|>
|> A quickie:
|>
|> Are there procedures available (or do you know of people who
|> have done this) that can provide optical distortion models
|> for images (arrays) in IDL? "Portraits in convex mirrors",
|> etc.?

I'm not aware of any general routines for doing this, but the general method is (reverse) raytracing (which I've been doing a lot of, but only for gravitational lenses, which are very convenient when it comes to raytracing).

Imagine that we have the following scenario (and pardon the ASCII graphics...):



Now, the trick is to map rays coming from the observer, via the mirror (or through the lens, for transparent lensing) onto the image. Obviously, the reverse direction is exactly symmetric, so the calculated paths are also the paths of light rays coming from the image (source) via the mirror to the observer.

Thus, for every point on the mirror, a unique hit point on the image can be found. Now, what will be seen (by the observer) at any specific point of the mirror, is the pixel value of the image at the hit point (on the image).

So the following skeleton should work:

mir_x = <Two-dimensional array describing the x position of each point
  on the mirror>

```
mir_y = <Ditto for the y position of each mirror point>

hit_x = x_hit_point(mir_x,mir_y)   ; Functions mapping each point (x,y)
                                   ; on the mirror onto a point (hit_x,hit_y)
hit_y = y_hit_point(mir_x,mir_y)   ; on the source according to the desired
                                   ; lens type. (Returns 2D arrays).
                                   ; Should return the pixel addresses, e.g.,
                                   ; the array subscripts of the source image

image = <2D array of same dim. as e.g. mir_x> ; Initialize background color

ix = where( <hit_x inside the source> and <hit_y inside the source>)

image(ix) = source_image(hit_x,hit_y)

tvscl,image    ; Here you have it. This is what the observer sees
               ; when looking at the mirror.
 ------------------------------------------------------------ -----------
```

Now, the crucial thing here is the geometrical mappings x_hit_point() and y_hit_point(), which depend on the setup of the lens system.

Remember that the observer position could also be included explicitly in the calculation (it is always included *implicitly*)..

Stein Vidar
