RTRACE(1G)

NAME
    rtrace - ray tracing program

SYNOPSIS
    rtrace [options] scene image [>log]

AUTHOR
    Antonio Costa, INESC-Norte, 1989 1992

DESCRIPTION
    rtrace command performs ray tracing from a SFF text file
    which must describe a scene with objects, lights, surface
    definitions, textures, etc. This scene file must be in the
    appropriate format, as described bellow.
    Basically, ray tracing is an algorithm for the creation of
    realistic images from the geometry and attributes of
    objects, lights, etc. This ray tracer supports several
    types of objects (sphere, box, bicubic patch, cone,
    cylinder, polygon, triangle and text), CSG and list opera-
    tions, different light types, reasonable surface defini-
    tions, lots of textures, several anti-aliasing schemes,
    depth of field, stereoscopic image creation and so on...

OPTIONS
    [wimage-width] [himage-height]
    The parameters image-width and image-height define the image
    size. Defaults are 256x256.

    [Aaliasing-threshold]
    [Sshading-threshold]
    [Tambient-threshold]
    The parameters aliasing-threshold (pixel supersampling),
    shading-threshold (shading rays propagation) and ambient-
    threshold (ambient rays distribution caching) control the
    image quality (0-best, 1-poor). Defaults are 0.05, 0.01 and
    0 (no ambient threshold). Good ranges are 0.1-0.03, 0.01-
    0.001 and 0.01-0.00001, respectively.

    [aantialiasing-mode]
    The parameter antialiasing-mode chooses adaptive supersam-
    pling antialiasing (0-default), semi-adaptive supersampling
    antialiasing (1) or normal supersampling antialiasing (2-
    should be used with nonzero focal apertures).

    [Bmaskfile]
    The parameter maskfile creates a file with a background
    mask, suitable for mixing images (it is like an alpha chan-
    nel).

    [ccluster-size]
    The parameter cluster-size controls the enclosing of objects
    (number of grouped objects per cluster) in the object
    hierarchy. Use a low value for sparse scenes, a high value
    for dense scenes (4-default).

    [dambient-levels]
    The parameter ambient-levels defines the number of shading
    levels (shading tree depth) in which ambient lighting
    calculations will be done through ray distribution (0-
    default, ie, no ray distribution). Use low values!

    [Dambient-samples]
    The parameter ambient-samples defines the maximum number of
    distributed rays to be used in ambient lighting calculations
    (16-default). Again, use with care.

    [iintersect-mode]
    The parameter intersect-mode chooses, in adaptive supersam-
    pling antialiasing, between testing all scene objects (1) or
    only the objects found at the pixel corners and inside (0-
    default; this greatly reduces CPU time, but with very small
    objects, it sometimes fails).

    [Iintersect-adjust-mode]
    The parameter intersect-adjust-mode avoids some problems
    with invalid self-intersections (1) (0-default). Scenes
    with text objects should be traced with this parameter equal
    to 1.

    [jjittering-mode]
    The parameter jittering-mode chooses jittered sampling (1)
    or not (0-default). Sometimes, activating it produces better
    images from scenes with small tricky details.

    [llighting-mode]
    The parameter light-mode controls the generation of shadow
    rays through non-opaque objects: 0-none (default), 1-
    partial, 2-full. If a scene has translucent objects, to
    obtain realism one should use 1 or 2 (better).

    [mshading-mode]
    The parameter shading-mode chooses between shading models:
    0-normal phong, 1-strauss (default but slower) (note: this
    model was developed by Paul Strauss of SGI).

    [nnormal-mode]
    The parameter normal-mode controls the correction of surface
    normals, so that it points against the incident ray: 0-
    always (default), 1-only inside objects. With "correct"
    objects, it is good to use 1.

    [znormal-check-mode]
    The parameter normal-check-mode controls the correction of
    surface normals when textures that modify the normal are
    used, as they may sometimes create strange surface effects.
    This tends to happen if the scale of the normal perturbation
    is big. 0-no correction (default), 1-correction.

    [Rrawfile]
    The parameter rawfile creates a raw image file, without any
    antialiasing (all defects show up!).

    [psampling-levels]
    The parameter sampling-levels controls the amount of sam-
    pling levels inside pixels: 0-none ... 3-maximum (default).
    A reasonable value is 2 for high resolutions, but for small
    ones 3 gives better (and slower) results.

    [sshading-levels]
    The parameter shading-levels establishes a maximum shading
    tree depth (default is 8). When a scene has
    transparent/reflective objects, it may be important to lower
    this parameter, or else the tracing never stops. In the
    other cases, there should be no problem allowing it to be
    big.

[t<u>texture</u>-<u>mode</u>]
The parameter <u>texture</u>-<u>mode</u> allows the definition of
texture(s) for the objects: 0-no textures (default), 1-with
textures defined inside objects field, 2-with textures
defined after objects field. As textures may consume much
CPU time, they should be activated only for final images.

[v<u>view</u>-<u>mode</u>]
The parameter <u>view</u>-<u>mode</u> chooses the view mode: 0-normal
(default), 1-left eye, 2-right eye.

[P<u>focal</u>-<u>aperture</u>]
The parameter <u>focal</u>-<u>aperture</u> defines the focal aperture of
the camera (default 0.0, ie, pinhole camera). If different
than zero, there is depth of field, and so adaptive super-
sampling antialiasing will not work well.

[F<u>focal</u>-<u>distance</u>]
The parameter <u>focal</u>-<u>distance</u> defines the focal distance of
the camera (default is the distance from the eye point to
the look point).

[E<u>stereo</u>-<u>separation</u>]
The parameter <u>stereo</u>-<u>separation</u> controls the separation
between the left and the right eye. If negative, it
represents a percentage of the gaze distance.

[O<u>output</u>-<u>format</u>]
The parameter <u>output</u>-<u>format</u> chooses between the PIC format
(0-default) or the PPM format (1).

[V<u>verbose</u>-<u>mode</u>]
The parameter <u>verbose</u>-<u>mode</u> supresses any messages (0) or
shows listing of parameters (1), previous plus statistics
(2-default) or previous plus a line by line echo of the
tracing (3-default on DOS and transputers).

The <u>scene</u> data internal syntax (SFF) is described bellow.

The <u>image</u> file will contain the ray traced image. The image
file has a 4 byte header composed of <u>width</u> LS and MS bytes,
<u>height</u> LS and MS bytes and RGB byte triplets starting in the
upper left corner line by line to the lower right corner.

RESTRICTIONS
    The straightforward use:

      rtrace demo.sff demo.pic

    is not recommended, as ray tracing usually takes lots of CPU
    time to execute.

    So, it is better to do:

      rtrace demo.sff demo.pic >demo.log &

    or then use <u>nice</u> (1) or similar strategies.

BUGS
    No bugs known. They have to be hidden deep somewhere, as
    usual.

DESCRIPTION

SFF (Simple File Format) description follows. This is a very
crude ASCII format, almost like if generated by a lexical
analyser. The idea is to have other programs create scene
descriptions in more sophisticated ways, and then feed the
tokenized results to this program. So, it behaves accord-
ingly to the UNIX philosophy: one program for one task.
Complaints are not wellcome!...

There is a reasonable scene language available (SCN) that
allows the creation of scenes with much more flexibility;
the converter is called 'scn2sff' and works directly with
this program.

Note: the ^(circunflex) character represents start of line.

```
[Start of File]
^... Comment
^Eye(X Y Z)
^Look(X Y Z)
^Up(X Y Z)
^View_angle(H V) [1,89 degrees]
^... Comment
^Background(R G B)
^Ambient(R G B)
^... Comment
^Light_type(Type) Position(X Y Z) Bright(R G B) ...
     |                               |
     |               /---------------------------/
     |               |
     V               V
 1-POINT:
 2-DIRECTIONAL: Direction(X Y Z) Angle(La) Light_Factor(Lf)
 3-EXTENDED:    Radius(R) Samples(N)
^Etc
^<NL>
^... Comment
^Surface_type(Type) Color(R G B) ...
     |                       |
     |       /---------------------/
     |       |
     V       V
 1-: Dif(R G B) Spec(R G B) Phong(Pf) Metal(Mf) Trans(R G B)
 2-: Smoothness(R G B) Metalness(R G B) Transmission(R G B)
^Etc
^<NL>
^... Comment
^Object_type(Type) Surface_ID(S) Refraction(Re) ...
     |                             |
     |               /-----------------------/
     |               |
     V               V
 1-SPHERE:          Center(X Y Z) Radius(R)
 2-PARALLELIPIPED:  Center(X Y Z) Size(X Y Z)
 3-PATCH:           Origin(X Y Z) Scale(X Y Z) Filename(...)
 4-CONE/CYLINDER:   Base(X Y Z) Base_Radius(Rb) Apex(X Y Z)
                    Apex_Radius(Ra)
 5-POLYGON:         Origin(X Y Z) Scale(X Y Z) Filename(...)
 6-TRIANGLE:        Origin(X Y Z) Scale(X Y Z) Filename(...)
 7-TEXT:            Filename(...)
or
 64-TEXTURE:        see below
 65-TRANSFORMATION: Object_ID(I)
                    Transform(X1 Y1 Z1 W1 ... X4 Y4 Z4 W4)
 66-CSG 0:          Surface_ID(S) Refraction(Re) (Union-0
```

```
Sub-1 Int-2)
    CSG 1:          Next CSG member
    CSG 2:          End of CSG
 67-LIST 0:         Surface_ID(S) Refraction(Re)
    LIST 1:         End of List
^Etc
^<NL>
^... Comment
^Texture_type(Type) Object_ID(I)
  |                 Transform(X1 Y1 Z1 W1 ... X4 Y4 Z4 W4)
  |                 ...
  |                     |
  V                     V
 0-NULL:
 1-CHECKER:    Surface_ID(S)
 2-BLOTCH:     Scale(K) Surface_ID(S) [Filename(...) or -]
 3-BUMP:       Scale(K)
 4-MARBLE:     [Filename(...) or -]
 5-FBM:        Offset(K) Scale(K) Omega(K) Lambda(L)
               Threshold(K) Octaves(O)
               [Filename(...) or -]
 6-FBMBUMP:    Offset(K) Scale(K) Lambda(L) Octaves(O)
 7-WOOD:       Color(R G B)
 8-ROUND:      Scale(K)
 9-BOZO:       Turbulence(K) [Filename(...) or -]
 10-RIPPLES:   Frequency(K) Phase(K) Scale(K)
 11-WAVES:     Frequency(K) Phase(K) Scale(K)
 12-SPOTTED:   [Filename(...) or -]
 13-DENTS:     Scale(K)
 14-AGATE:     [Filename(...) or -]
 15-WRINKLES:  Scale(K)
 16-GRANITE:   [Filename(...) or -]
 17-GRADIENT:  Turbulence(K) Direction(X Y Z)
               [Filename(...) or -]
 18-IMAGEMAP:  Turbulence(K) Mode(K) Axis(X Y) Filename(...)
 19-GLOSS:     Scale(K)
 20-BUMP3:     Scale(K) Size(K)
^<NL>
^... Comments
[End of File]
```

1. Valid ranges of data

RGB must be in [0,1[(Note: RGB brightness of lights may be between]-300,300[; negative values mean to not attenuate with distance).

XYZ must be in [-10000,10000]

Factor must be in [0,300[

Filename must a valid filename for the operating system, or then '-', in which case data is read from the standard input or the current SFF stream.

2. Patch specification

File format for PATCH (bicubic 4-sided patch):

```
[Start]
^Patch_1_Index(1 2 3 4 5 6 7 8 9 10 11 12)
^Patch_2
^Etc
^<NL>
^Patch_Index_1_Coords(X Y Z)
^Patch_Index_2_Coords(X Y Z)
^Etc
^<NL>
^...
[End]
```

3. Polygon specification

File format for POLYGON (n-sided planar polygon):

```
[Start]
^Polygon_1_Vertex_Number Polygon_1_Index(1 2 3 ...)
^Polygon_2
^Etc
^<NL>
^Polygon_Index_1_Coords(X Y Z)
^Polygon_Index_2_Coords(X Y Z)
^Etc
^<NL>
^...
[End]
```

4. Triangle specification

File format for TRIANGLE (3-sided polygon with vertex normals):

```
[Start]
^Triangle_1_Vertice_1(X Y Z) Normal_1(X Y Z)
        Vertice_2(X Y Z) Normal_2(X Y Z)
        Vertice_3(X Y Z) Normal_3(X Y Z)
^Triangle_2
^<NL>
^...
[End]
```

5. An example

```
[Start of File]
View
25 25 7        - Eye point
0 0 0          - Look point
0 1 0          - Up vector
30 30          - View angles
Colors
0.196 0.6 0.8  - Background (Sky Blue)
0.1 0.1 0.1    - Ambient light
Lights
1 0 60 60 0.9 0.9 0.9 - Point Light 1
1 20 40 -7 0.9 0.9 0.9 - Point Light 2
<NL>
Surfaces
1 0.6 0.8 0.196 0.99 0.99 0.99 0 0 0 0 0 0 0 0
1 0.9 0.9 0.9 0.5 0.5 0.5 0.5 0.5 0.5 50 1 0 0 0
1 0.5 0.5 0.5 0.1 0.1 0.1 0.1 0.1 0.1 200 0.7 0.8 0.8 0.8
1 0.9 0.2 0.2 0.99 0.99 0.99 0 0 0 0 0 0 0 0
<NL>
Objects
5 1 1.0 0 0 0 15 15 15  - Polygon
4 1 2 3 4
<NL>
1 0 1
1 0 -1
```

```
-1 0 -1
-1 0 1
<NL>
2 2 1.0 0 2 0 7 2 3      - Parallelipiped
2 3 1.5 0 5 10 3 5 3     - Parallelipiped
1 4 1.0 7 15 -7 3        - Sphere
<NL>
Textures
2 1 2 0 0 0 0 2 0 0 0 0 2 0 0 0 0 1 0.4 4
4 2 5 0 0 0 0 5 0 0 0 0 5 0 0 0 0 1
5 4 10 0 0 0 1 10 0 0 0 1 1 10 0 0 0 0 1 0 0.6 0.5 2 0.1 6
<NL>
Demo / 11-OCT-1989 / Antonio Costa
[End of File]


To ray trace without textures, do:

  rtrace demo.sff demo.pic >&demo.log


else, do:

  rtrace t2 demo.sff demo.pic >&demo.log


Another example with INESC symbol:

[Start of File]
View
45.0 45.0 81.0  - Eye point
45.0 45.0 -81.0 - Look point
 0.0  1.0  0.0  - Up vector
30 30           - View angles
Colors
0.196 0.6 0.8 - Background (Sky Blue)
0.3 0.3 0.3   - Ambient
Lights
1  0.0 100.0 100.0 1 1 1 - Light 1 (White)
1 90.0 100.0 100.0 1 1 0 - Light 2 (Yellow)
<NL>
Surfaces
1 0.557 0.420 0.137 0.8 0.7 0.7 0.2 0.3 0.3 30 0.8 0 0 0
1 0.137 0.420 0.557 0.5 0.5 0.6 0.5 0.5 0.4 5 0.2 0 0 0
1 0.600 0.800 0.200 0.9 0.9 0.9 0.0 0.0 0.0 1 0 0 0 0
<NL>
Objects
1 1 1.0 10.0 09.5 0.0 4.5 - Sphere
1 1 1.0 10.0 26.5 0.0 4.5
1 1 1.0 20.0 63.5 0.0 4.5
1 1 1.0 20.0 80.0 0.0 4.5
1 1 1.0 40.0 09.5 0.0 4.5
1 1 1.0 40.0 26.5 0.0 4.5
1 1 1.0 40.0 43.5 0.0 4.5
1 1 1.0 50.0 80.0 0.0 4.5
1 1 1.0 60.0 53.0 0.0 4.5
1 1 1.0 70.0 09.5 0.0 4.5
1 1 1.0 70.0 43.5 0.0 4.5
4 2 1.0 10.0 30.0 0.0 1.5 10.0 70.0 0.0 1.5 - Cylinder
1 2 1.0 10.0 70.0 0.0 1.5
4 2 1.0 10.0 70.0 0.0 1.5 17.5 77.5 0.0 1.5
4 2 1.0 12.5 12.0 0.0 1.5 20.0 19.5 0.0 1.5
1 2 1.0 20.0 19.5 0.0 1.5
4 2 1.0 20.0 19.5 0.0 1.5 20.0 60.0 0.0 1.5
4 2 1.0 22.5 61.0 0.0 1.5 37.5 46.0 0.0 1.5
4 2 1.0 37.5 12.0 0.0 1.5 30.0 19.5 0.0 1.5
1 2 1.0 30.0 19.5 0.0 1.5
4 2 1.0 30.0 19.5 0.0 1.5 30.0 33.5 0.0 1.5
1 2 1.0 30.0 33.5 0.0 1.5
4 2 1.0 30.0 33.5 0.0 1.5 37.5 41.0 0.0 1.5
4 2 1.0 30.0 26.5 0.0 1.5 36.5 26.5 0.0 1.5
4 2 1.0 40.0 47.0 0.0 1.5 40.0 70.0 0.0 1.5
1 2 1.0 40.0 70.0 0.0 1.5
4 2 1.0 40.0 70.0 0.0 1.5 47.5 77.5 0.0 1.5
4 2 1.0 42.5 12.0 0.0 1.5 50.0 19.5 0.0 1.5
1 2 1.0 50.0 19.5 0.0 1.5
4 2 1.0 50.0 19.5 0.0 1.5 50.0 43.0 0.0 1.5
1 2 1.0 50.0 43.0 0.0 1.5
4 2 1.0 50.0 43.0 0.0 1.5 57.5 50.5 0.0 1.5
4 2 1.0 67.5 12.0 0.0 1.5 60.0 19.5 0.0 1.5
1 2 1.0 60.0 19.5 0.0 1.5
4 2 1.0 60.0 19.5 0.0 1.5 60.0 33.5 0.0 1.5
1 2 1.0 60.0 33.5 0.0 1.5
4 2 1.0 60.0 33.5 0.0 1.5 67.5 41.0 0.0 1.5
5 3 1.0 0.0 4.0 0.0 200.0 200.0 200.0 - Polygon
4 1 2 3 4
<NL>
 1.0 0.0  1.0
 1.0 0.0 -1.0
-1.0 0.0 -1.0
-1.0 0.0  1.0
<NL>
<NL>
End
INESC Logo / 23-FEB-1989 / Antonio Costa
```