SCN2SFF(1G)

NAME
     scn2sff - format conversion program

SYNOPSIS
     scn2sff [options] {SCNfile|-} [>SFFfile]

AUTHOR
     Antonio Costa, INESC-Norte, 1991 1992

DESCRIPTION
     scn2sff command performs conversion between a text format (SCN) suitable for scene descriptions to another more simple text format (SFF) that is accepted by the rtrace ray-tracer program. The SCN text file describes objects, lights, surface definitions, textures, etc. The scene format is described bellow.

OPTIONS
     [C]
     The parameter C tells the program to preprocess the input file through the UNIX standard preprocessor (/usr/lib/cpp with option -P).

     [M]
     The parameter M tells the program to preprocess the input file through the M4 preprocessor (/usr/bin/m4).

     [P"preprocessor command"]
     The parameter P tells the program to preprocess the input file through the command described (for example, P"/usr/lib/cpp -P -Dabc").

RESTRICTIONS
     None for the moment.

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

DESCRIPTION
     Comments start with % ; or # characters and continue to the end of the line (so there are no nested comments).

     The commands are processed from the start of the scene, and some have a global effect until they are changed or the scene is complete. Such commands are refraction, surface, transformations and textures; as commands can be nested by grouping, any command defined inside a group is removed when that group is finished.

     Example:

```
surface matte white        % current surface is matte white
refraction 1.1             % current refraction index
transform rotate x 15      % transformation
group                      % start of a group of commands
  surface matte red        % new current surface
  transform scale 2        % transformation
  sphere 0 0 0 1           % sphere object
ungroup                    % end of group
% back to matte white surface, refraction 1.1
% scale transformation is removed
```

ENTITIES

The main entities are:

integer - it can be a simple number, an integer expression enclosed in parenthesis or the int function applied to any real expression.

real - a number, a function or an expression enclosed in parenthesis.

color - a triplet of RGB real values between 0 and 1 (in certain cases, it is allowed to be greater than 1 or negative; called color extended) or a name (like red, blue, etc).

point - a triplet of XYZ values (numbers, functions or expressions).

vector - similar to point, but the 3 components cannot be simultaneously equal to 0.

filename - a set of characters with no blanks between.

expression - anything enclosed in parenthesis. Operators are +, -, *, /, ^ (exponentiation) and | (remainder).

function - there are many functions available: int sin cos tan asin acos atan sqrt rtod dtor exp log abs max min. There also some functions that operate with vectors and return a number (dotvector) and some that return a vector or point (normvector addvector diffvector scalevector crossvector). There is also an operator mono that converts from a value to 3 identical values (good for specifying monochromatic colors).

GENERAL COMMANDS

The main commands are:

eye (from) point - default {5,0,0}.

look (at) point - default {0,0,0}.

up vector - default {0,1,0}.

angle (fov) horizontal [vertical] - half aperture view in degrees (default 22.5 degrees).

background color - the color of the background, at infinite distance (default light_sky_blue).

ambient color - the diffuse light that illuminates the whole scene (default is {0.1,0.1,0.1}).

refraction (ior) index - default is 1.

group ... ungroup - anything enclosed is only defined inside the block, ie, it does not apply outside.

LIGHT COMMANDS

The commands for definition of light sources are:

light point point [color_extended] - default color for lights is white.

light directional vector [color].

light spot point vector color_extended [angle [factor]] - the light illuminates inside a cone defined by the angle (default 45 degrees) and the transition can be sharp if factor is near 1 or smooth if factor >> 1 (default 1).

light extended point color_extended radius samples - a spherical light (it is sampled by samples^2 rays).

SURFACE COMMANDS

The commands for definition of surfaces are:

surface color [diffusion specularity phong metalness [transparency]] - phong and metalness are values, the others are colors (defaults {0.9,0.9,0.9} {0.1,0.1,0.1} 3 0 {0.1,0.1,0.1} or transparency only {0,0,0}).

surface strauss color smoothness metalness [transparency] - all colors (default transparency is {0,0,0}).

surface matte color - all diffuse surface.

surface plastic color smoothness phong - surface with big diffusion, small specularity and small phong factor.

surface metal color smoothness phong - surface with small diffusion, big specularity, big phong factor and metalness factor equal to 1.

surface dielectric color transparency refraction - transparent surface with no diffusion, some specularity, large phong factor and null metalness.

surface glass color transparency - transparent surface with refraction index equal to 1.52, approximately.

OBJECT COMMANDS

The commands for objects are of the form
     object object_data
or else with local commands that apply only to itself of the form
     object [attributes ... data] object_data.

sphere center radius.

box center sizes - this is an axis-aligned box.

cube center size - again it is axis-aligned.

cone apex base base_radius - closed cone.

cone open apex base base_radius.

cylinder apex base radius - closed cylinder.

cylinder open apex base radius.

cone truncated apex apex_radius base base_radius - closed.

cone truncated open apex apex_radius base base_radius.

wedge point point point depth - defined by a triangular face and depth (face is defined counterclockwise so that depth is measured in the opposite direction of Rigth Hand Rule thumb; this convention also applies to other objects).

tetra point point point point.

prism depth number_vertices point ... point - closed prism.

prism open depth number_vertices point ... point.

pyramid depth number_vertices point ... point - closed pyramid.

pyramid open depth number_vertices point ... point.

pyramid truncated open depth apex_scale number_vertices point ... point - it is an open pyramid with the apex scaled by apex_scale in relation to its base (if 0 it is an open pyramid, if 1 it is a prism).

disc center normal radius.

ring center normal outer_radius inner_radius.

patch point ... point (12) - a bicubic patch is defined by its corners and 8 exterior points, arranged in this manner:
```
                 11  12
               7  8  9  10
               3  4  5  6
                  1  2
```
Normal points according to Right Hand Rule using corners 4-5-9-8.

patch file [point [point]] filename - a group of patches stored in a file; first point is a translation and second is a scale.

polygon number_vertices point ... point - a polygon (can be concave, but does not have holes).

polygon file [point [point]] filename - a group of polygons stored in a file; first point is a translation and second is a scale.

triangle point point point.

quadrangle point point point point.

triangle normal point vector point vector point vector - a triangle with normals in its vertices.

triangle normal file point point filename - a group of triangles with normals in the vertices stored in a file; first point is a translation and second is a scale.

torus outer_radius section_radius start_angle end_angle [outer_samples section_samples] - A closed torus is centered in {0,0,0} and lies in the XZ plane. 0 degrees is in the X direction and the angle increases counterclockwise.

torus open outer_radius section_radius start_angle end_angle [outer_samples section_samples] - An open torus.

text3d file filename - a group of text primitives stored in a file; each primitive is described by lines and arcs and is extruded (similar to a prism, in a certain way).

csg begin - start of a CSG primitive, ie, left component.

csg next - right component of a CSG primitive.

csg end - end of a CSG primitive.

list begin - start of a list primitive (no nesting allowed).

list end - end of a list primitive.

TRANSFORMATION COMMANDS

A transformation may be defined globaly or inside a block, and it is post-concatenated with previous transformations. If inside a block, when the block is terminated the transformations defined inside it are removed. Also, when a transformation is an attribute of an object or texture it only exists for that entity.

transform none - removes all transformations.

transform scale factor [factor factor].

transform translate point.

transform rotate x angle.

transform rotate y angle.

transform rotate z angle.

transform rotate axis angle.

transform general point point point [point].

TEXTURE COMMANDS

A texture is basically a modification of the surface characteristics of an object, a modification of the normal vector in the intersection point or the modification of the intersection point itself. It is possible to apply transformations to textures, and even keep them independent from the object transformations.

texture none - remove all defined textures.

texture scale factor [factor factor].

texture translate point.

texture rotate x angle.

texture rotate y angle.

texture rotate z angle.

texture rotate axis angle.

texture general point point point [point].

texture local - generate all the transformations necessary to access the object directly, without considering any object transformations previously defined.

checkers surface [transform] - a chessboard-like pattern of the current surface and the defined surface.

blotch scale surface [filename] [transform] - A spray-like mixture of 2 surfaces (the current and the defined). The scale controls the mixture. If a filename is given, it is interpreted as a color palette, and it must contain 256 triplets of RGB values in the range 0 to 255 (this format is equal for all the textures that have a filename parameter, except imagemap).

bump scale [transform] - A normal-modifying texture.

marble [filename] [transform] - A marble-like texture.

fbm offset scale omega lambda threshold octaves [filename] [transform] - A fractal brownian motion texture that changes diffusion and specularity.

fbmbump offset scale lambda octaves [transform] - a texture that modifies the normal.

wood color [transform] - A texture imitating wood (default color is brown).

round scale [transform] - strange texture that modifies diffusion and specularity.

bozo turbulence [filename] [transform].

ripples frequency phase scale [transform] - a texture that imitates ripples (small sinusoidal perturbations of the surface).

waves frequency phase scale [transform] - a texture like waves (multi-interfering sinusoidal perturbations of the surface).

spotted [filename] [transform] - small color spots.

dents scale [transform] - small modifications of normal that imitate dents.

agate [filename] [transform].

wrinkles [transform] - a texture that modifies normal imitating wrinkles.

granite [filename] [transform].

gradient turbulence direction [filename] [transform] - This texture produces a variation of color following direction given.

imagemap turbulence mode u_axis v_axis filename [transform]

- An image-mapping texture. Mode parameter controls tiling of texture (0-yes, nonzero-no). The u_axis and v_axis specify the internal texture axis from the 3D axis (1-X, 2-Y, 3-Z). A filename must be given, because it is the image that will be drawn on the surface (the format of the image is the rtrace format PIC).

gloss scale [transform] - Glossy-like texture that changes diffusion, specularity and phong factor.

bump3 scale size [transform] - A normal-modifying texture. Changes intersection point, so may produce strange results!

EXAMPLES

Here are some simple examples:

```
%%%%% example 1
% light source
light point 4 3 1
% surface
surface matte red
sphere 0 0 0 1
% another surface (replaces previous)
surface plastic blue mono 0.3 0.3
sphere 3 -0.4 0.4 0.2
% another surface
surface plastic yellow mono 0.9 0.9
% transformations for next object(s)
transform rotate y rtod(atan(1))
transform translate 3 -0.4 -0.4
box 0 0 0 0.1 0.1 0.3
% remove previous transformation(s)
transform none
% another surface
surface green mono 0.8 mono 0.2 10 0.3
cone 3 0.1 0 3 -0.4 0 0.2
surface matte white
csg subtraction begin
    sphere 0 0 0 1
    csg next
    box 0 0 0 1.1 0.4 0.4
csg end
```

This example is correct, although it does not take full usage of SCN, ie, the capability of defining locally the attributes. It could be rewritten:

```
%%%%% example 2
% light source
light point 4 3 1
% now all objects have local attributes
sphere
    surface matte red
    data 0 0 0 1
sphere
    surface plastic blue mono 0.3 0.3
    data 3 -0.4 0.4 0.2
box
    surface plastic yellow mono 0.9 0.9
    % local transformations
    transform rotate y rtod(atan(1))
    transform translate 3 -0.4 -0.4
    % this object is defined in a local coords system
    % the translation puts it in the right place
    data 0 0 0 0.1 0.1 0.3
cone
    surface green mono 0.8 mono 0.2 10 0.3
    data 3 0.1 0 3 -0.4 0 0.2
csg subtraction surface matte white
    data begin
    sphere 0 0 0 1
    csg next
    box 0 0 0 1.1 0.4 0.4
csg end
```

To produce an image from any of these examples, the example should be stored in a file (suppose example.scn) and then execute

```
  scn2sff example.scn|rtrace w512 p2 A0.1 - example.pic
```

to create the image. If the SCN file contained any cpp preprocessor directives, then

```
  scn2sff C example.scn|rtrace w512 p2 A0.1 - example.pic
```

would do.

A complete demo example follows:

```
[Start]
% example to be traced with parameters like
% w512 p2 A0.1 t1 I1 - good quality
% or then
% w512 p3 A.05 t1 I1 j1 - very good quality

%%%%% start
eye 5 2 2
fov 20
background light_sky_blue
ambient mono 0.2 % dark grey

light point 3 5 4 white

surface matte red % default surface

%%%%% a simple CSG example
csg subtraction begin
% no attributes for this CSG, so it uses the attributes
% of its nodes...

  % left node
  csg subtraction
    % attributes of this CSG object
    surface matte white
    texture scale 0.2
    checkers surface matte mono 0.3 translate 0.1 0.1 0.1
    data begin

    box 0 0 0 1 1 1

  csg next

    box 0 0 0 1.01 0.5 0.5
    list begin
      % a cylinder must be enclosed in a list, because it is
      % not a closed object, but 3 objects joined together
      cylinder 0 1.01 0 0 -1.01 0 0.5
    list end

  csg end

csg next
```

```
       % right node
       sphere 1 1 1 0.5 % default surface assumed
       sphere 1 1 -1 0.5
       sphere surface matte blue data 1 -1 1 0.5
       sphere surface matte blue data 1 -1 -1 0.5

     csg end

     %%%%% some 3D text
     text3d file surface matte yellow
       data csg.t3d % data is in file

     %%%%% end
     [End]

     The csg.t3d file contents could be:
     [Start]
     SPACING 0.1
     ORIENTATION 0 0 -1 0 1 0 1 0 0
     ENCODING abc.ppe
     FONT zurichcg.ppf
     SCALE 0.4 0.4 0.2
     AT 1.25 1.5 1.6 "Antonio Costa"
     FONT renfrew.ppf
     SCALE 0.4 0.4 0.1
     AT 1.1 -0.85 1.1 "/copyright/1992"
     AT 1.1 -1.3 1.1 "INESCn"

     # there must be an empty line in the end
     Description:
     SPACING is letter spacing
     ORIENTATION defines how the text appears
      - 1st: text direction vector (left to right)
      - 2nd: vertical vector
      - 3rd: depth vector
     ENCODING associates logical character names to glyph numbers
     FONT is the file where the 2D glyphs are defined
     SCALE controls scaling along ORIENTATION vectors
     AT is baseline lower left position of text plus text
     (quoted)
     [End]

HISTORY
     Copyright (C) 1991 1992 by Antonio Costa.
     Permission is granted to use this file in whole or in part
     for any purpose, educational, recreational or commercial,
     provided that this copyright notice is retained unchanged.
     This software is available to all free of charge by
     anonymous FTP.

     23-Jul-92  Antonio Costa at INESC-Norte
          Release 1.3.1
          acc@asterix.inescn.pt acc@basinger.inescn.pt
```