SIGGRAPH2006

Introduction to RenderMan

Saty Raghavachary

Talk outline

RenderMan basics

- origins
- 'RI Spec' – brief tour
- workflow
- scene (RIB) files
- shaders (RSL)
- resources

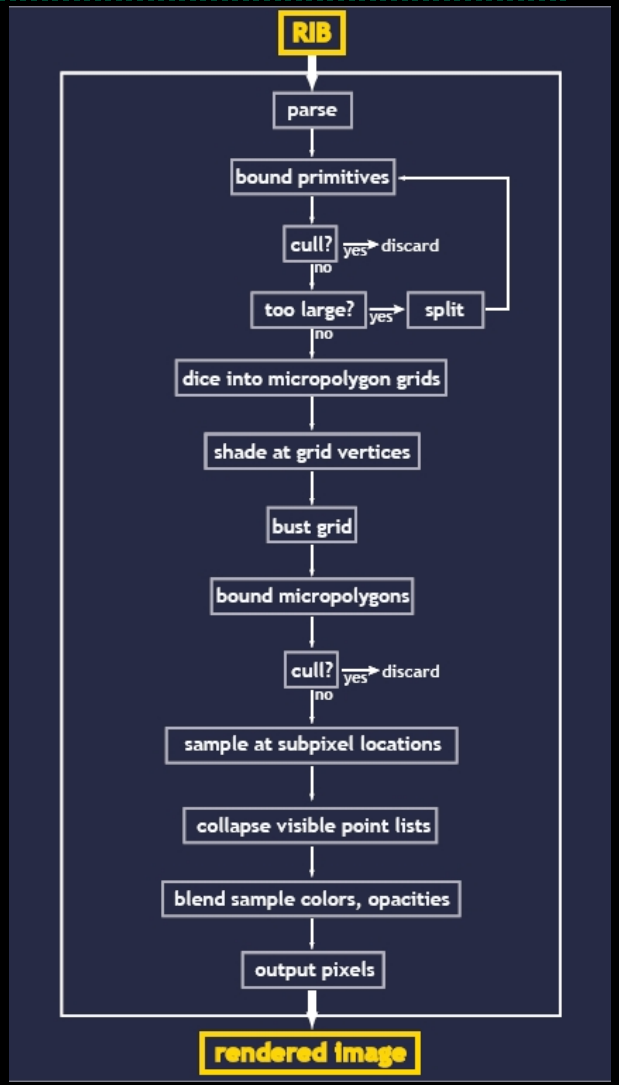
Origins, history

- Univ. of Utah – Ed Catmull
- Lucasfilm, Pixar - “REYES” ('87)
- Photorealistic RenderMan ('89), RSL ('90)

Origins, history

REYES

- all objects get broken down into micropolygons
- micropolygons are shaded
- results are combined to produce output pixel colors and opacities



Origins, history

SIGGRAPH2006

Pixar's short films

- The Adventures of Andre and Wally B. (1984)
- Luxo Jr. (1986)
- Red's Dream (1987)
- Tin Toy (1988)
- Knickknack (1989)
- Geri's Game (1997)
- For the Birds (2000)
- Mike's New Car (2002)
- Boundin' (2004)

The Adventures of André & Wally B. | Red's Dream | Tin Toy | Knick Knack

Origins, history

Origins, history

A Bug's Life
A.I.
Aladdin
Apollo 13
Atlantis: The Lost Empire
Babe: Pig in the City
Balto
Batman Forever
Batman Returns
Beauty and the Beast
Bicentennial Man
Black Hawk Down
Casper
Cast Away
Cats & Dogs
Chicken Run
Clear & Present Danger
Cliffhanger
Contact
Death Becomes Her
Demolition Man
Dinosaur
Evolution
Fight Club
Final Fantasy: The Spirits Within
Finding Nemo

Forrest Gump
Free Willy
Gladiator
Harry Potter and the Chamber of Secrets
Harry Potter and the Sorcerer's Stone
Hollow Man
How the Grinch Stole Christmas
Indian in the Cupboard
Inspector Gadget
Interview with a Vampire
Iron Giant
Jetsons
Jumanji
Jurassic Park: The Lost World
Jurassic Park 2
Jurassic Park 3
Men in Black
Men in Black II
Minority Report
Miracle on 34th Street
Mission to Mars
Monsters, Inc.
Moulin Rouge
Nutty Professor II: The Klumps

Outbreak
Patch Adams
Pearl Harbor
Pitch Black
Planet of the Apes
Pocahontas
Shark Tale
Sinbad: Legend of the Seven Seas
Sleepy Hollow
Small Soldiers
Space Cowboys
Species
Speed
Spider Man
Spirit: Stallion of the Cimarron
Star Trek VI
Star Wars Episode I: The Phantom Menace
Star Wars Episode II: Attack of the Clones
StarQuest
Starship Troopers
Stuart Little
Stuart Little 2
Terminator II
The Abyss
The Adventures of Rocky & Bullwinkle
The Beach

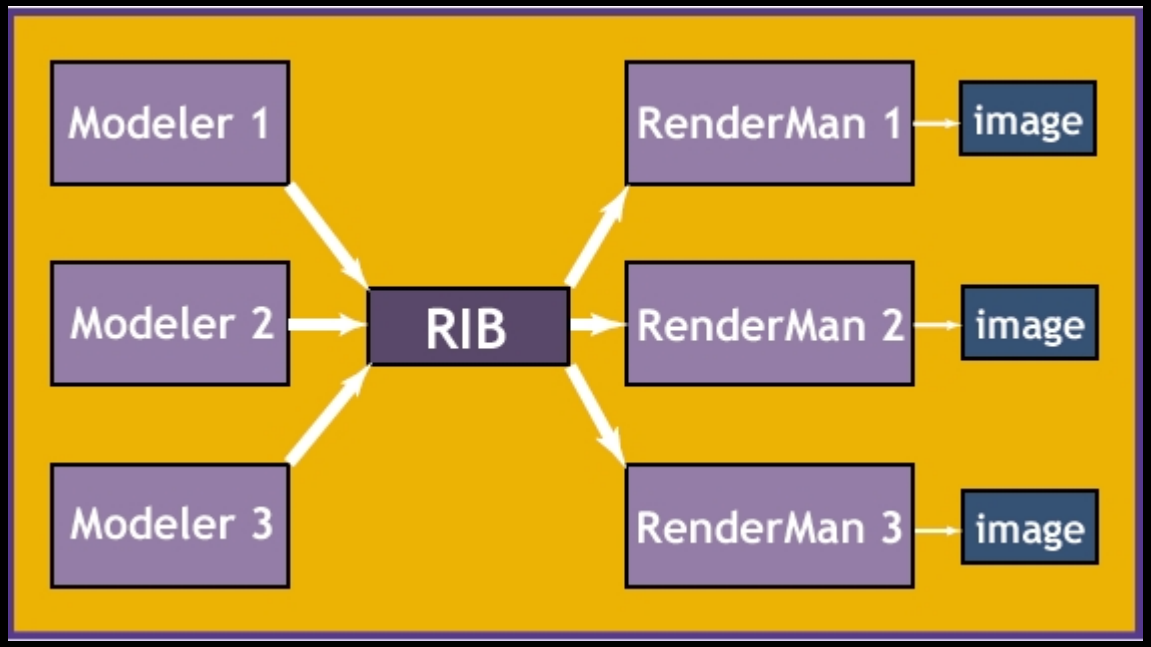
The Incredibles
The Jungle Book
The Lion King
The Lord of the Rings: Return of the King
The Lord of the Rings: The Fellowship of the Ring
The Lord of the Rings: The Two Towers
The Mask
The Matrix
The Mummy
The Mummy Returns
The Perfect Storm
The Prince of Egypt
The Road to El Dorado
The Scorpion King
The Time Machine
The World is Not Enough
Titanic
Tomb Raider
Toy Story
Toy Story 2
True Lies
Vanilla Sky
What Dreams May Come
Young Sherlock Holmes

LOTS of movies use RenderMan!

RI Spec

What is RenderMan?

- an interface between modelers and renderers



- idea is to be able to mix-and-match them

RI Spec

What's in the Spec

- core capabilities (eg. hierarchical graphics state, antialiasing, programmable shading language)
- advanced/optional capabilities (eg. motion blur, depth of field, global illumination)
- C/C++ bindings and RenderMan Interface Bytestream (RIB) bindings
- RenderMan Shading Language (RSL)

RI Spec

Who implements the Spec?

- Pixar – Photorealistic RenderMan (PRMan) - v.13
- Dot C Software – RenderDotC
- Paul Gregory and team - Aqsis
- SiTex Graphics – AIR
- Okan Arikan – Pixie
- 3Delight
- ... several others ...

RI Spec

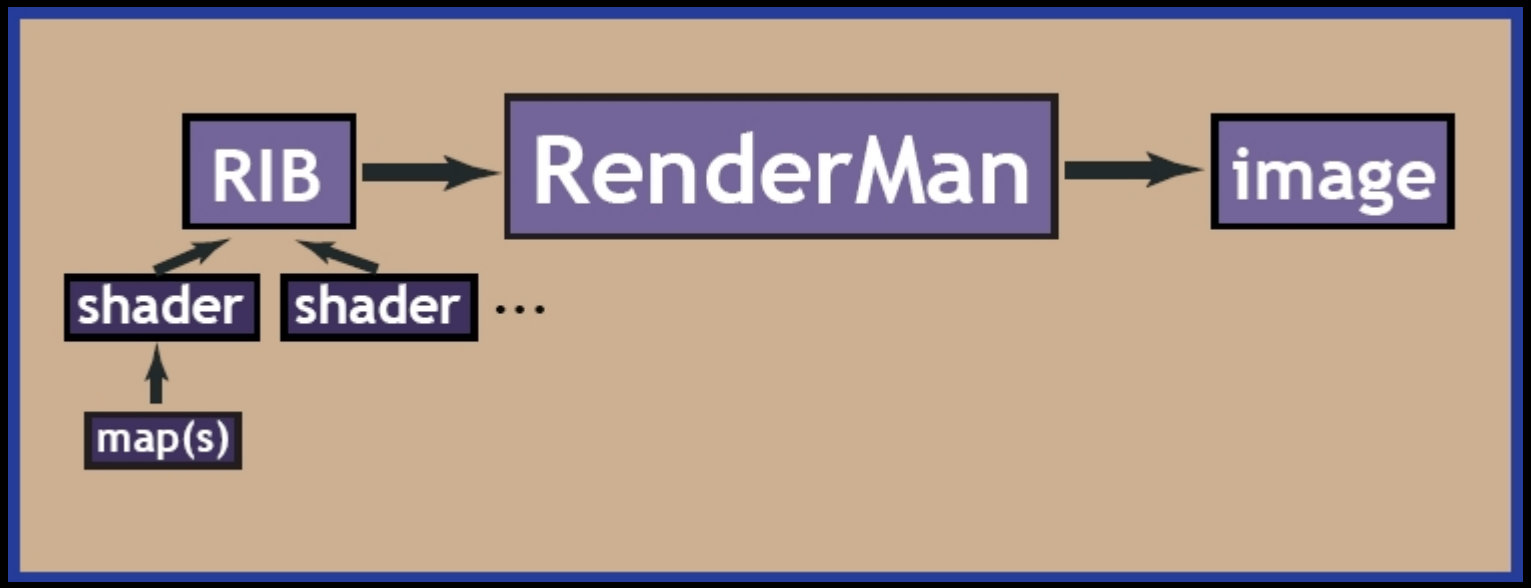
Version numbers

- 3.0 – 1988
- 3.1 – 1989, revised in 1995
- 3.2 – 2000
- 3.2.1 – Nov. 2005
- 3.3 – in the works

Read the Spec thoroughly!

Workflow

RenderMan dataflow/pipeline



Do modeling, layout, animation, effects

and lighting/surfacing elsewhere (eg. in Maya)

Workflow

Sources of RIB files

- 'by hand', using a text editor
- by running a program – bindings exist for Java, Python, Perl, Tcl, Ruby..
- standalone scene converters, eg. mi2rib
- native output, eg. from Blender
- translator plugins – eg. for Maya, we have MTOR/Slim, RfM, MayaMan and Liquid

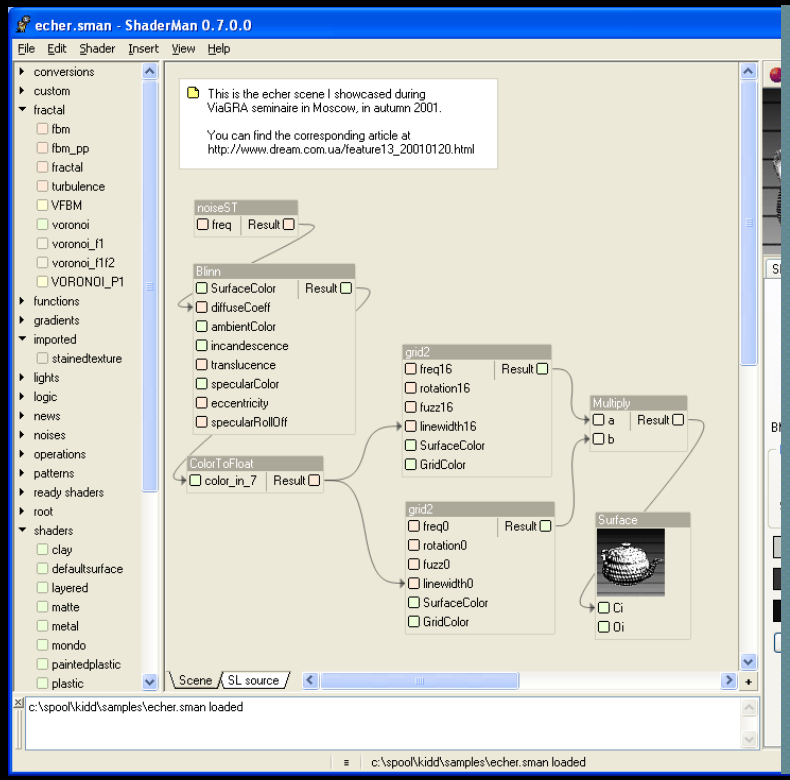
Workflow

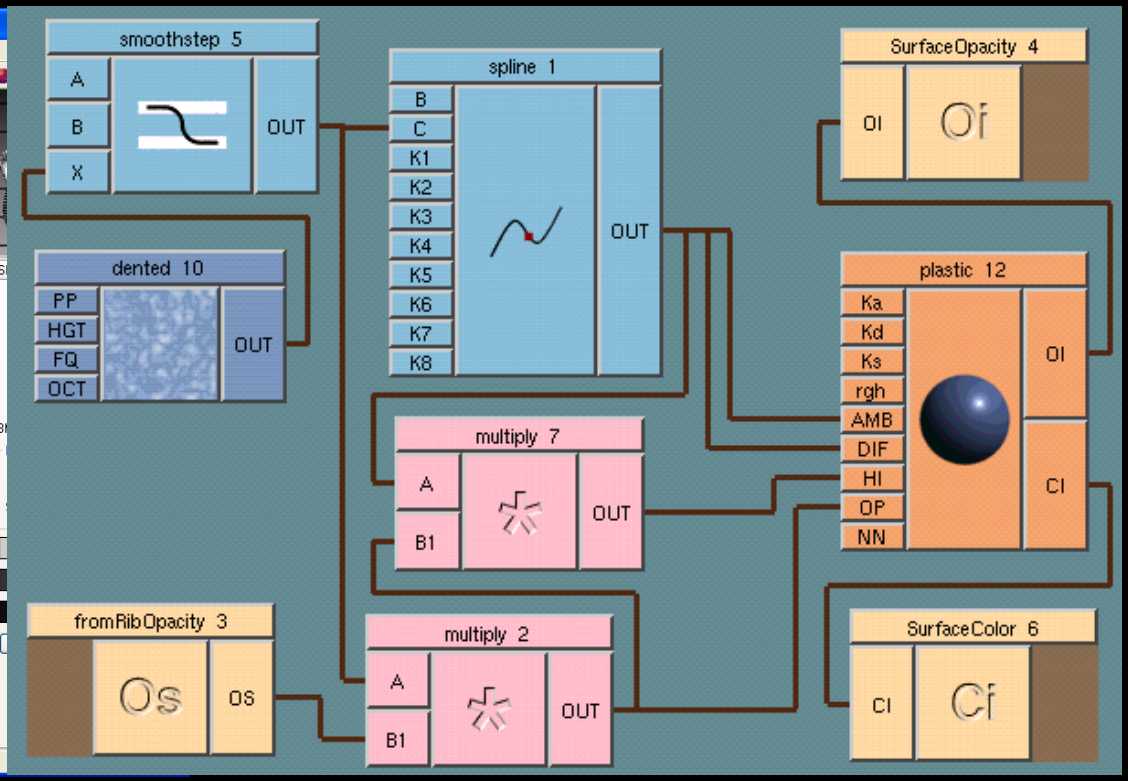
Sources of shaders

- from scratch, 'by hand'
- plugin interface, eg. 'Slim' for Maya
- integrated into host program, eg. 'RenderMan for Maya' plugin for Maya
- standalone 'connect the boxes' UI programs, eg.:
  - ShaderMan
  - ShadeTree (defunct)

Workflow

ShaderMan, ShadeTree





Workflow

Sources of maps

- digital paintings, scans, photos..
- prior renders, ie. from a previous 'pass'

TIFF images for texturing need to be converted to map format, eg. .tex for PRMan

Types of maps:

- texture
- environment/reflection
- normal
- shadow/deep shadow
- photon
- irradiance cache
- occlusion
- brick

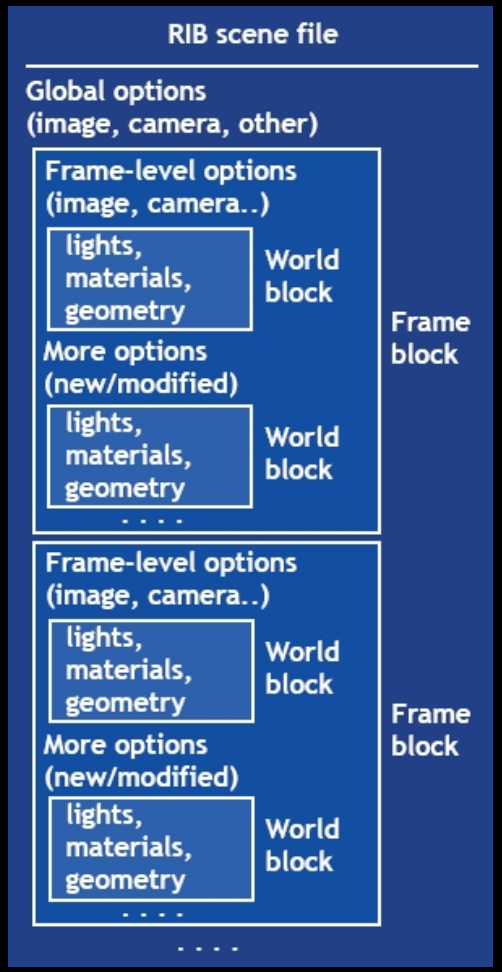
Workflow

Ways of extending RenderMan

- shaders
- DSO shader plugins, “shadeops”
- standalones/DSOs to output RIB
  for use with the 'Procedural' RIB call
- display driver DSOs

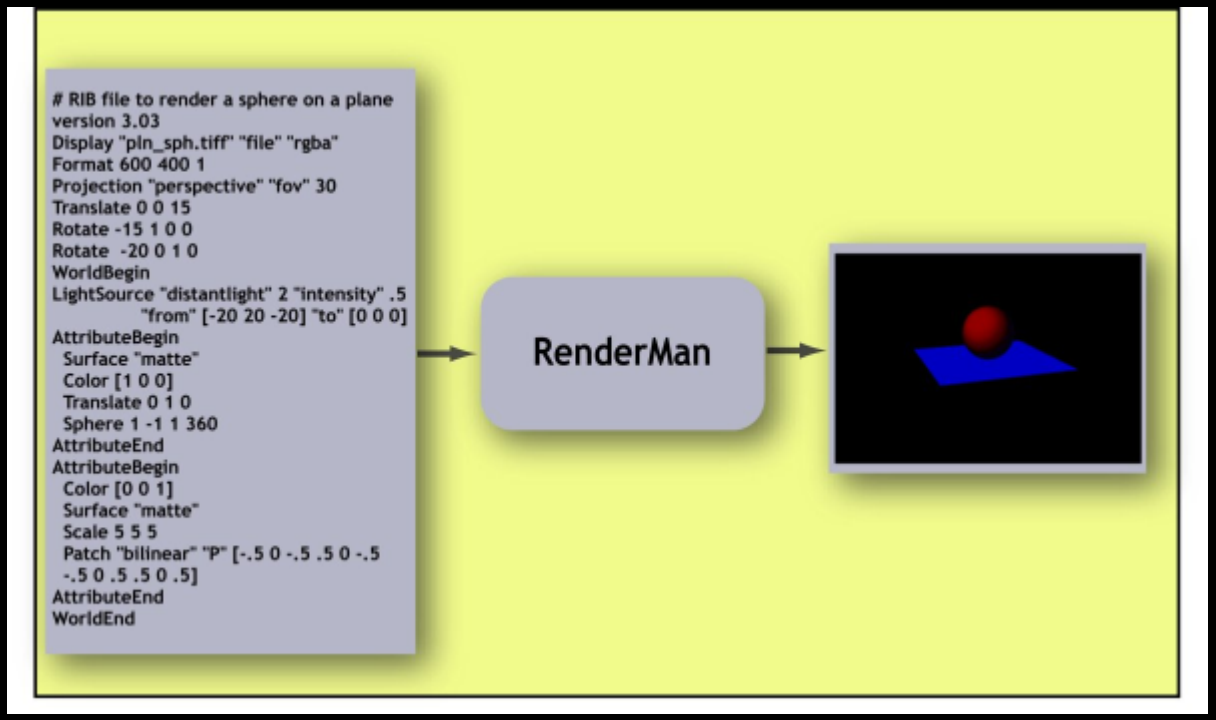
RIB files

- scene description files
- simple, declarative style, eg.
  `Projection "perspective" "fov" 54.3`
- no loops, branches, function calls etc.
- can be ASCII or binary
- usually one file per frame of animation



RIB files

A very simple RIB file and output:



PRMan render command: `render sph_pln.rib`

RSL

Language highlights

- C-like syntax
- types: float, string, color, point, normal...
- operators: +,-,*,/,%,==...
- control statements: if(), for()..
- rich collection of built-in functions
- user-definable functions
- DSO shadeops (shader plugins)
- pre-processor directives, eg. #include

RSL

Built-in function categories

- mathematical, eg. `asin()`
- geometric, eg. `normalize()`
- color, eg. `ctransform()`
- matrix, array, eg. `arraylength(), determinant()`
- string, eg. `concat()`
- shading, lighting, eg. `diffuse(), trace()`
- texture mapping, eg. `texture(), shadow()`
- message passing, info., eg. `surface(), rayinfo()`

RSL

Five categories of shaders

- surface
- displacement
- atmosphere
- light
- imager

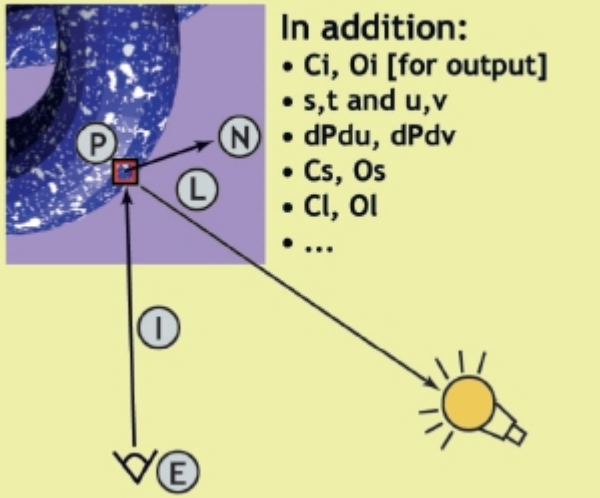
Shader source needs to be compiled, eg. (PRMan):

```
shader myshader.sl
```

RSL

SIGGRAPH2006

Surface shader global variables



In addition:

- Ci, Oi [for output]
- s,t and u,v
- dPdu, dPdv
- Cs, Os
- Cl, Ol
- ...

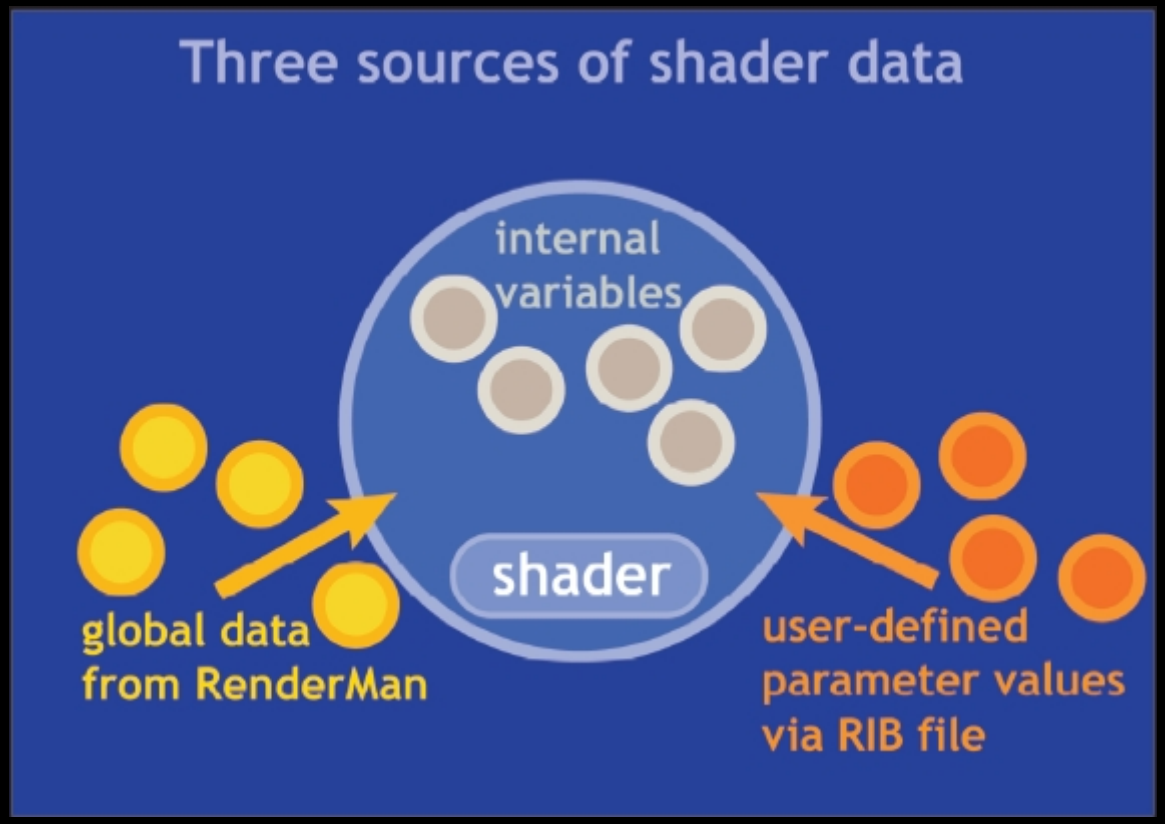
Information based on the RiSpec 3.2. Copyright Pixar Animation Studios

Surface Shader Variables

Variable Name	Type	Storage class	Description
Cs	color	varying	Surface Color described on the RIB file
Os	color	varying	Surface opacity described on the RIB file
P	point	varying	Position of shaded surface
dPdu	vector	varying	Derivative (tangent) of the surface position along u
dPdv	vector	varying	Derivative (tangent) of the surface position along v
N	normal	varying	Surface shading normal
Ng	normal	varying	Surface geometric normal
u,v	float	varying	Surface parameters
du,dv	float	varying	Change in surface parameters
s,t	float	varying	Surface texture coordinates
L	vector	varying	Incoming light ray direction*
Cl	color	varying	Incoming light ray color *
Ol	color	varying	Incoming light ray opacity*
E	point	uniform	Position of the eye or camera
I	vector	varying	Incident ray direction. Direction vector going from the camera to the current shading point
ncomps	float	uniform	Number of color components
time	float	uniform	Current shutter time
dtime	float	uniform	Amount of time covered by this shading sample
dPdtime	vector	varying	How the surface position P is changing per unit time, as described by motion blur in the scene.
Ci	color	varying	Shader output color
Oi	color	varying	Shader output opacity

* Available only in illuminance statements

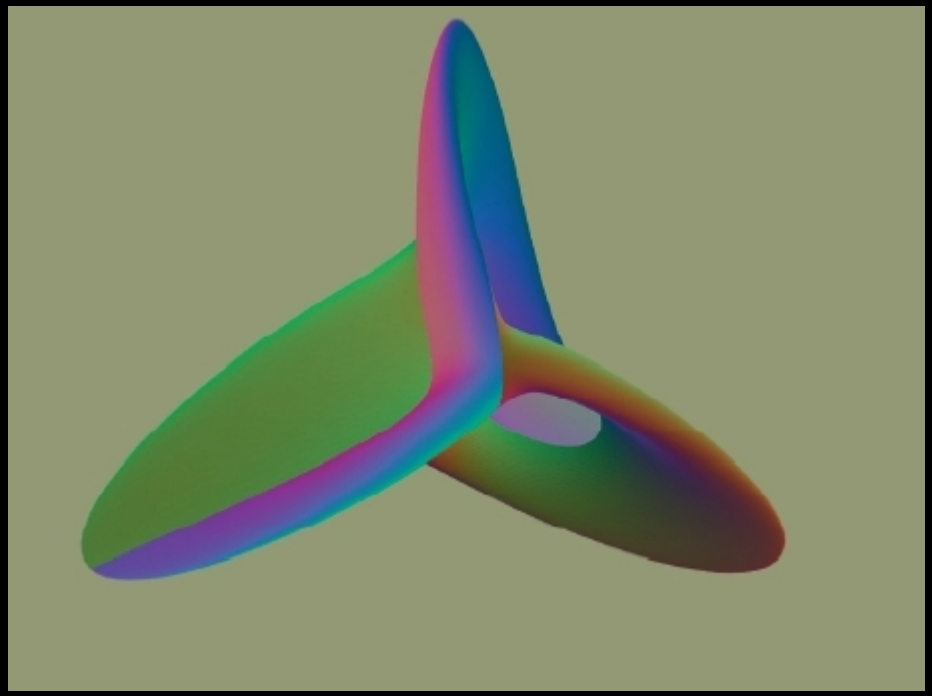
RSL


Three sources of shader data
internal variables
shader
global data from RenderMan
user-defined parameter values via RIB file

RSL


```
surface showN()
{
 normal NN = normalize(N);
 vector posNorm = 0.5*(vector(1,1,1)+NN);

 Oi = Os;
 Ci = Oi*color(comp(posNorm,0), comp(posNorm,1),
 comp(posNorm,2));
}// showN()
```

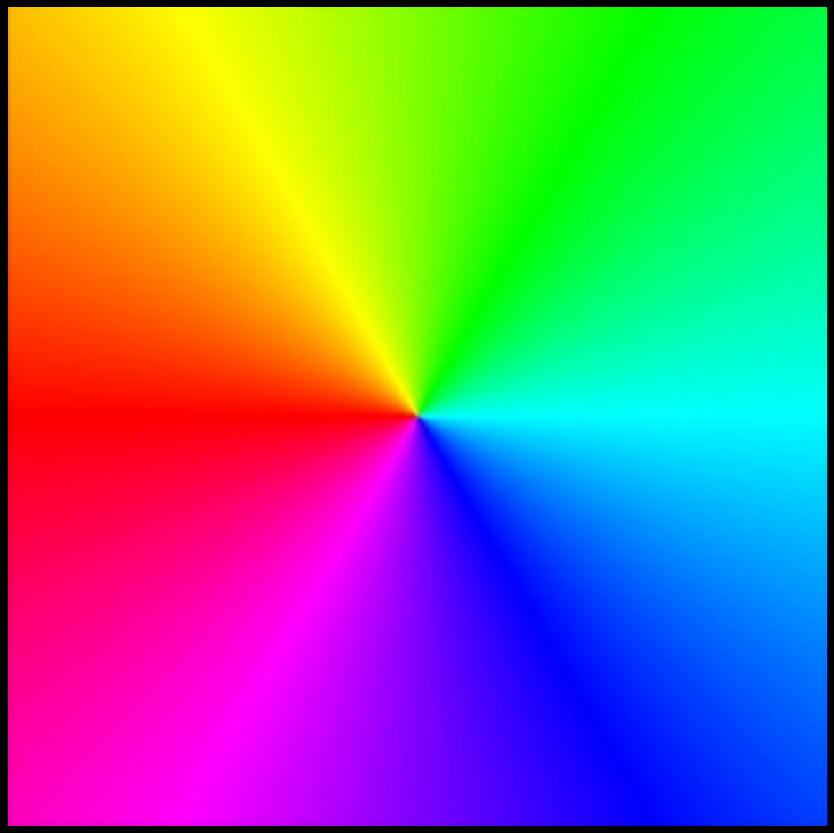
RSL

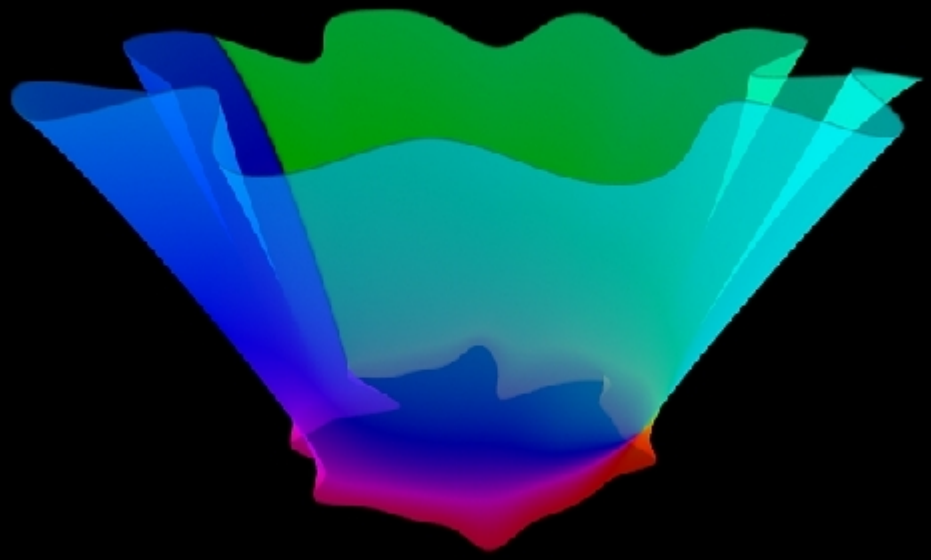


RSL


```
surface tex(string tmap="generic.tex";)
{
  float alpha;

  /* get base color from map */
  if(tmap!="")
  {
    color Ct = color texture(tmap,s,t);
    alpha = texture(tmap[3],s,t);
    Oi = alpha;
    Ci = Oi*Ct;

  }

}//tex()
```

RSL

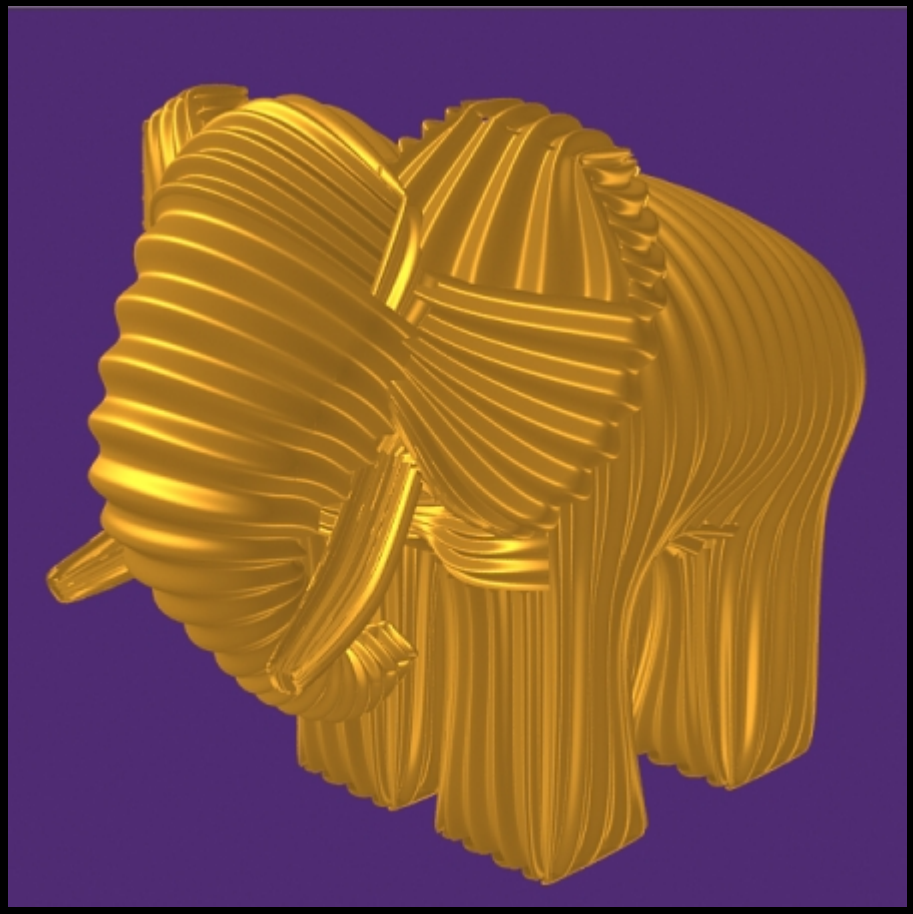




RSL


```
displacement sinewaves(float freq=1.0, ampl=1.0,
sphase=0, tphase=0, paramdir=0)
{
  // displace along normal, using sin(s),sin(t) or both
  if(0==paramdir)
  {
    P += ampl*sin(sphase+s*freq*2*PI)*normalize(N);
  }
  else if (1==paramdir)
  {
    P += ampl*sin(tphase+t*freq*2*PI)*normalize(N);
  }
  else
  {
    P += ampl*sin(sphase+s*freq*2*PI)*
    sin(tphase+t*freq*2*PI)*normalize(N);
  }
  N = calculatenormal(P);
}// sinewaves
```

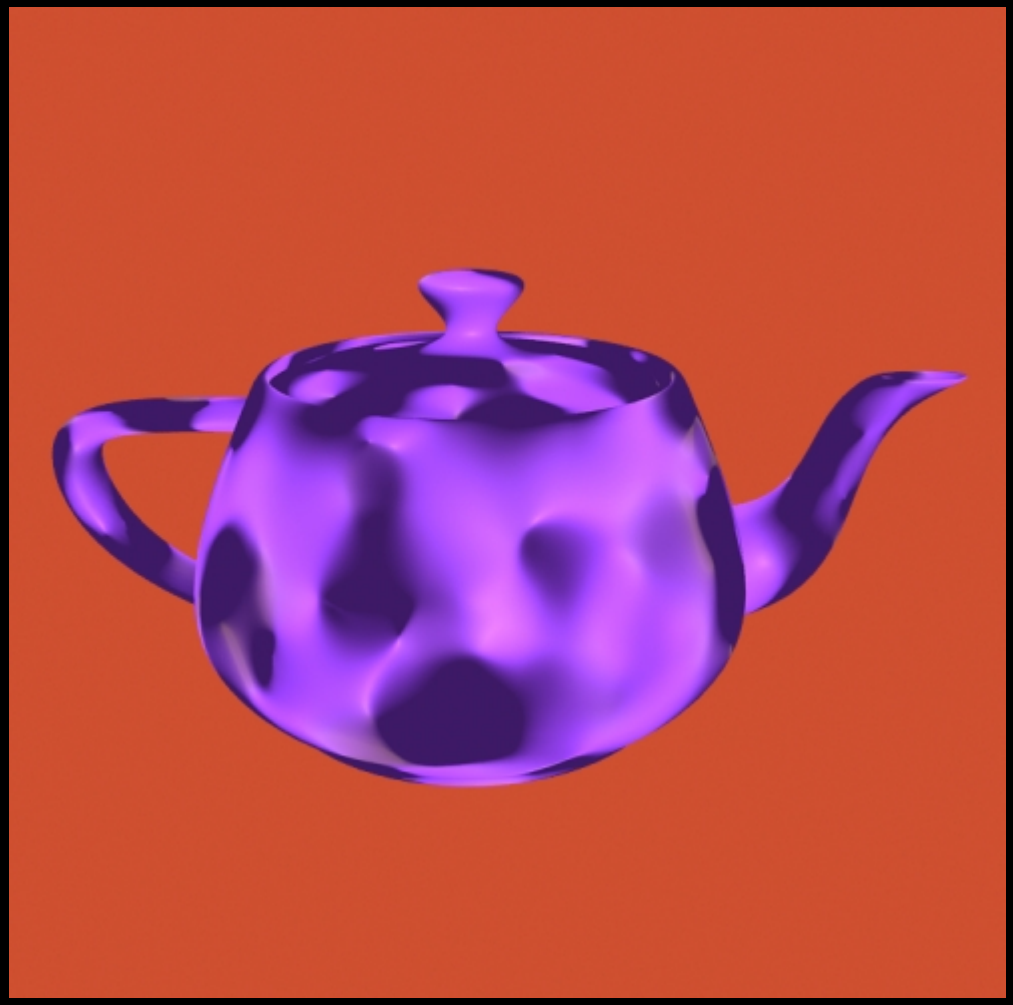
RSL



RSL


```
light
Kessonlt(
    float  intensity=1 ;
    color  lightcolor=1 ;
    float freq=1.0, coneangle=PI/2;
)
{
  point Pt = freq*transform("shader",Ps);
  vector ldir = 2*noise(freq*Pt) - 1;
  solar(ldir,coneangle)
  {
    Cl = intensity * lightcolor;
  }
}// Kessonlt()
```

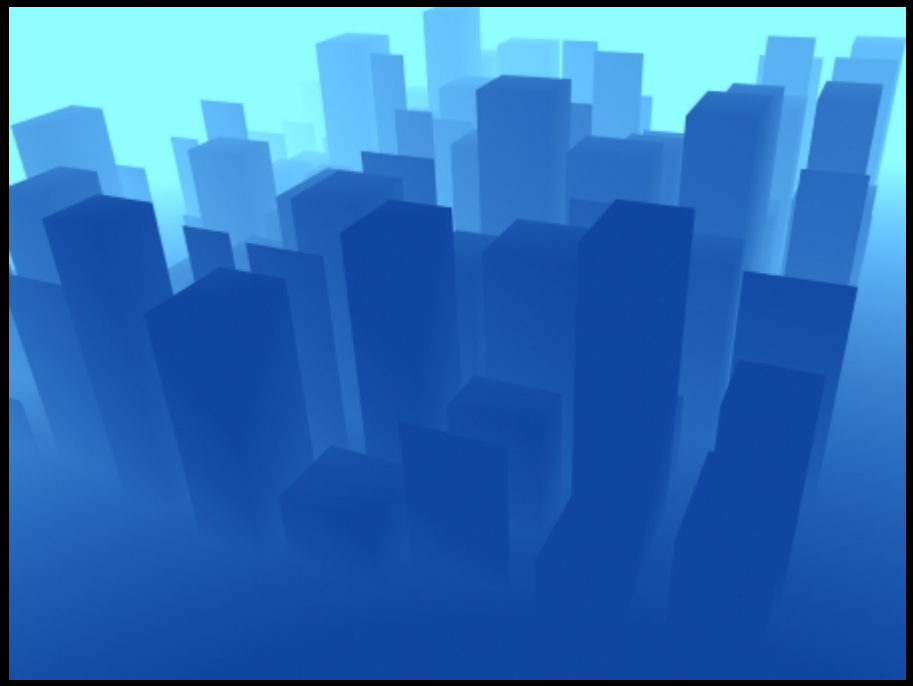
RSL



RSL


```
volume underwater(
    float mindist=0, maxdist= 1;
    color fg=1, bg=1;
    float inten=1, gam=1, mixf=0.5;
    )
{
    color c; float d;
    d = length(I);
    if(d<=mindist)c = fg;
    else if(d>=maxdist)c = bg;
    else
    {
     d = (d-mindist)/(maxdist-mindist);
     d = pow(d,gam);
     c = mix(fg,bg,d);
    }
    Ci = inten*mix(Ci,c,mixf);
    Oi = mix( Oi, color (1,1,1), d );
}// underwater()
```

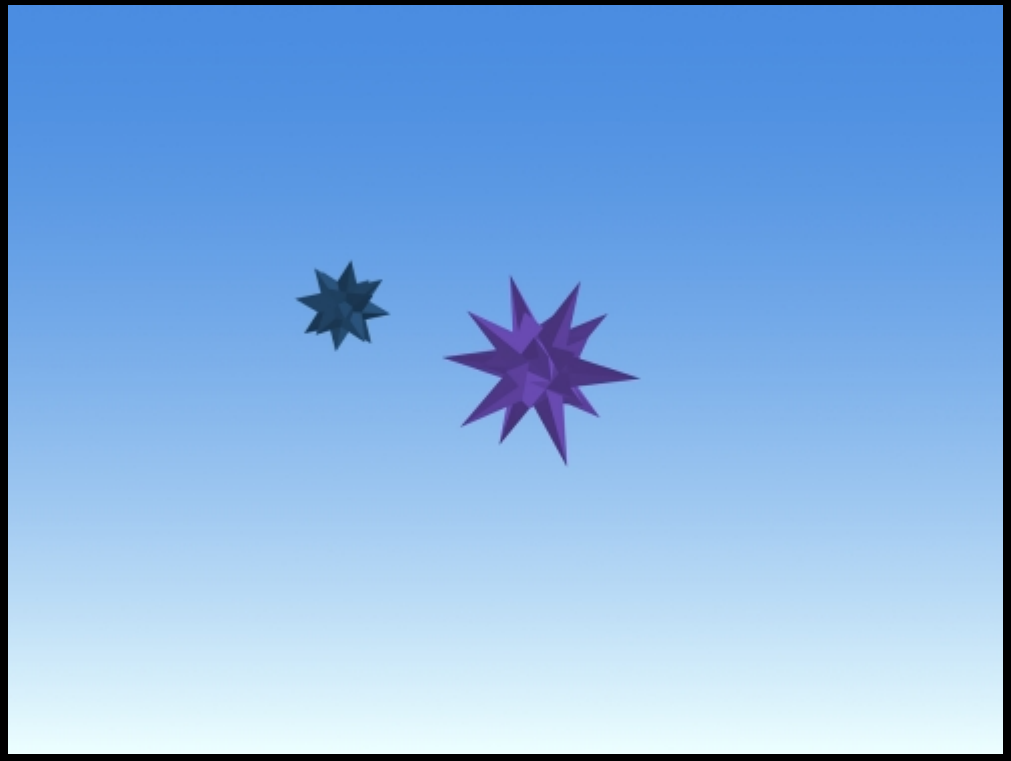
RSL



RSL


```
imager Imager_ramp
(
 color ctop = color(1,1,1);
 color cbot = color(0,0,0);
 float gam=1.0;
 )
{
 float curr_y;
 float rez[3];
 color rampcol;
 float mixf;

 option("Format",rez);
 curr_y = ycomp(P)/ rez[1]; // 0 to 1, top to bottom
 curr_y = pow(curr_y,gam);

 rampcol = mix(ctop,cbot,curr_y);
 Ci += (1-Oi)*rampcol;
 Oi = 1.0;
}// Imager_ramp()
```

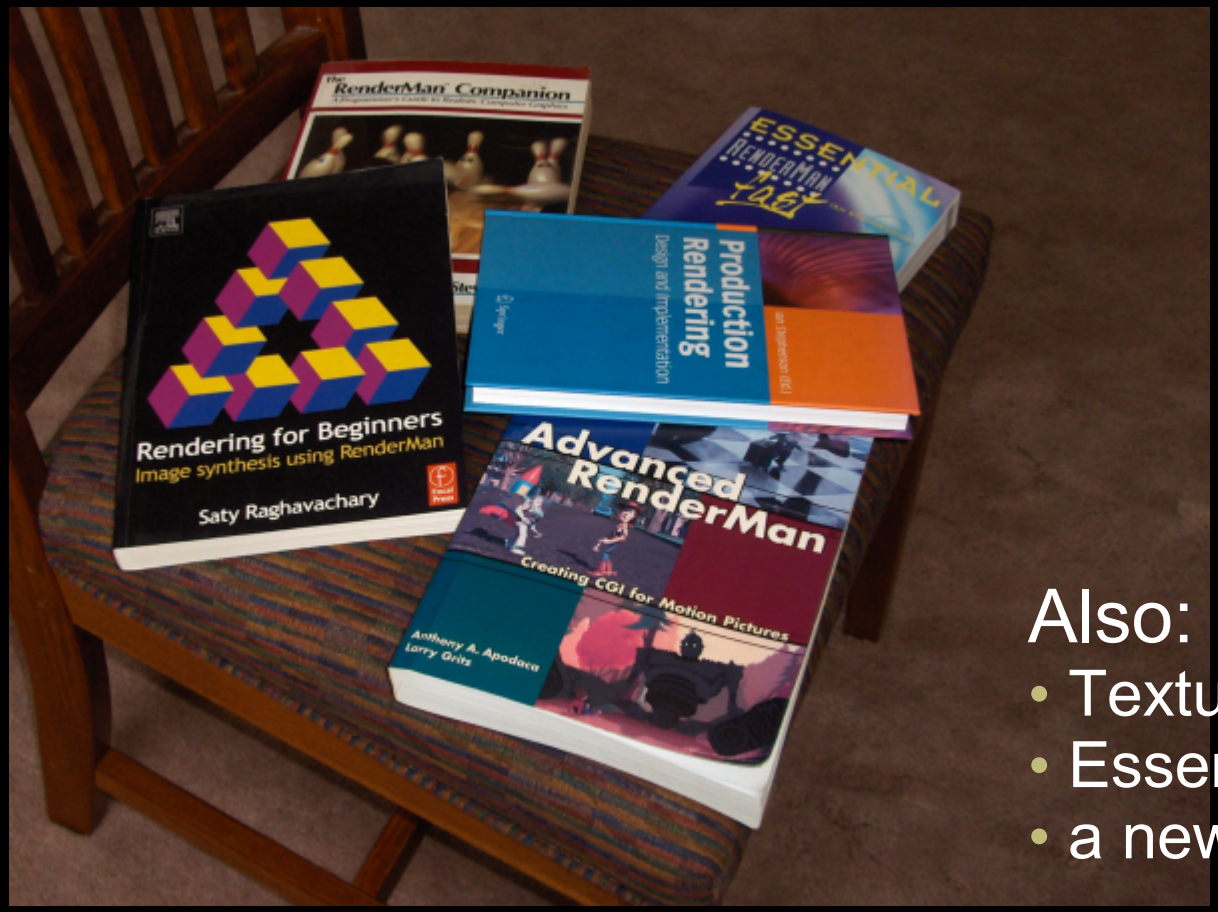
RSL



Resources

SIGGRAPH2006

Books



Also:

- Texturing and Modeling
- Essential RenderMan
- a new RSL book

Resources

Notes, tutorials

- SIGGRAPH course notes
  '92, '95, '98, '99, '00, '01, '02, '03, '06
- Prof. Malcolm Kesson – 'fundza'
- Zhang Jian - “ZJ”
- Steve May - RManNotes
- Katsuaki Hiramitsu - 'Katsu's RooM'

Resources

Portals, forums

- RenderMan Repository – RMR – Tal Lancaster
- RenderMan Academy – Rudy Cortes
- RenderMania – Simon Bunker
- highend3d.com
- deathfall.com
- comp.graphics.rendering.renderman

Resources

Implementors' sites

- Pixar - www.pixar.com
- RenderDotC - www.dotcsw.com
- Aqsis - www.aqsis.org
- AIR - www.sitexgraphics.com
- Pixie - sourceforge.net/projects/pixie
- 3Delight - www.3delight.com
- ART - www.art-render.com/

Summary

- origins
- RI Spec
- workflow
- RIB files
- shaders (RSL)
- resources

HAVE FUN!!