Weta Digital, December 18th, 2013

Reproducing the Natural Complexity

Fabrice NEYRET

CNRS / INRIA / Grenoble University, France

Reproducing the Natural Complexity

ultra-detailed + ultra large

Fabrice NEYRET

CNRS / INRIA / Grenoble University, France

Fabrice NEYRET

CNRS / INRIA / Grenoble University, France

Reproducing the Natural Complexity

ultra-detailed + ultra large

shape + animation + rendering

seamless + realistical

Fabrice NEYRET

CNRS / INRIA / Grenoble University, France

Reproducing the Natural Complexity

ultra-detailed + ultra large

shape + animation + rendering

realistical

in real-time

Fabrice NEYRET

CNRS / INRIA / Grenoble University, France

Reproducing the Natural Complexity

ultra-detailed + ultra large

shape + animation + rendering

realistical

in real-time

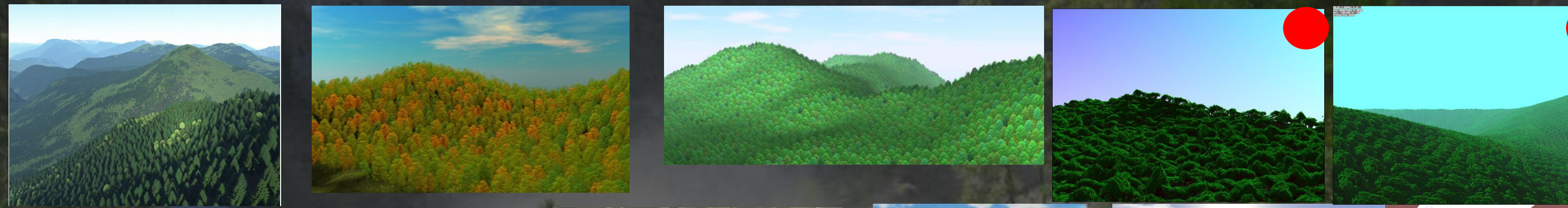
controlable

Fabrice NEYRET

CNRS / INRIA / Grenoble University, France

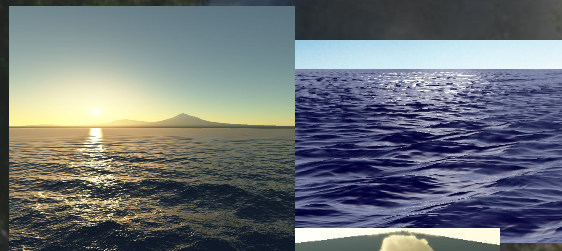
Pheno:

- Forests:



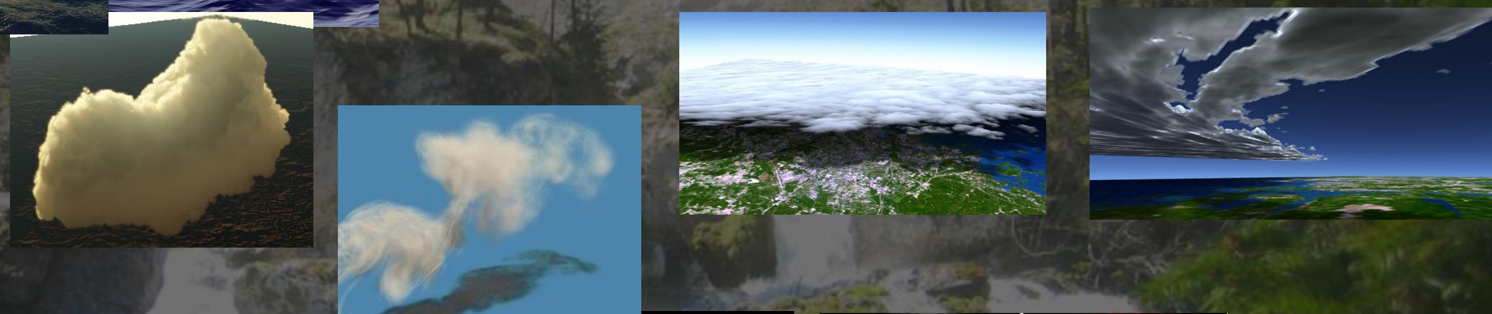
- Rivers:

- Ocean:

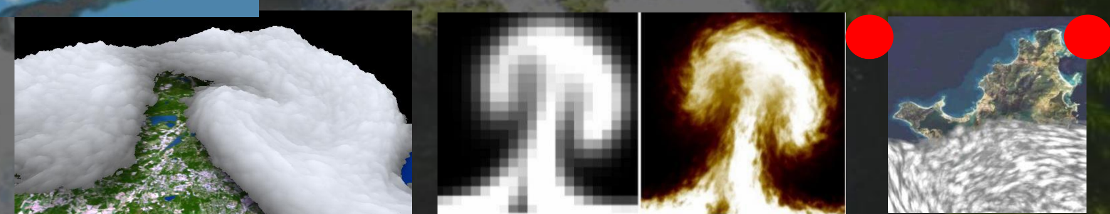


- Clouds:

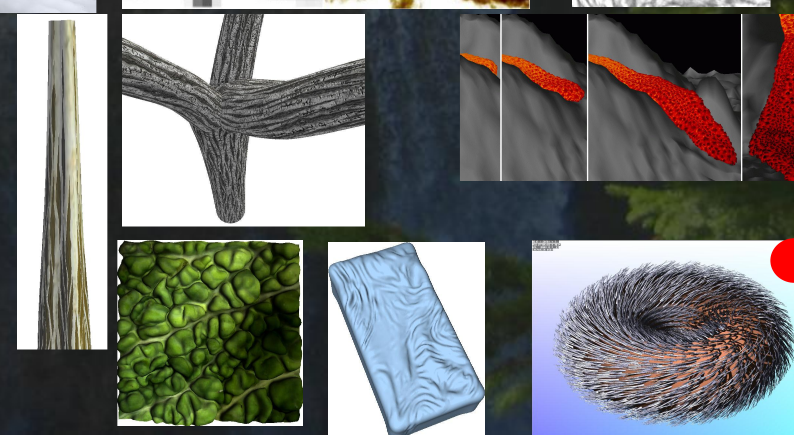
- Smoke:



- Advected textures, Flow Noise:



- Bark:
- Folds, hairs, morphogenesis:

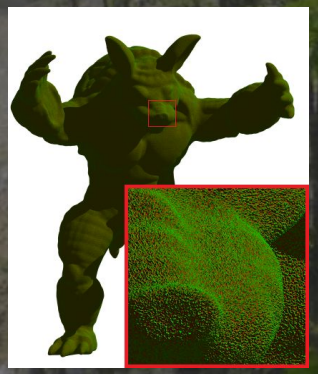


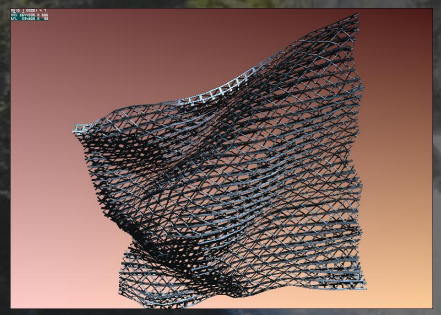
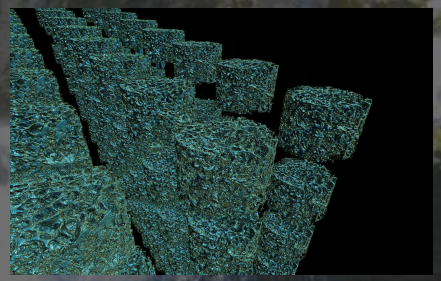
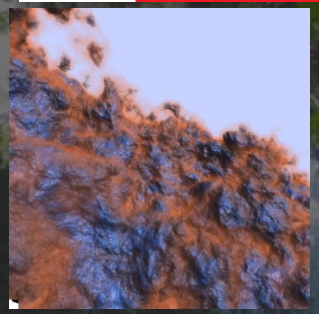
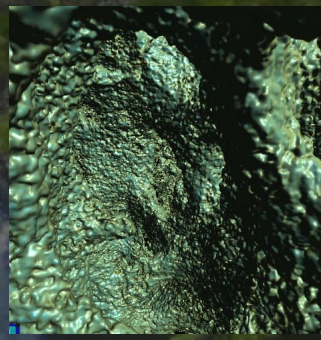
Representations:

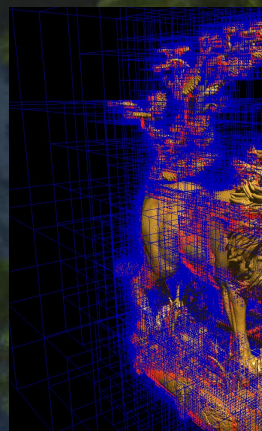
- Textures:
- Appearance filtering:
- SVO:
- alt repr:
-





Organisation of my Talk

1. *Copy-Paste of 256 powerpoint presentations*
2. *Brain dump of my understandings*

Organisation of my Talk

1. ***Tour through various pieces of work.***
   - *From Volumetric Textures to Gigavoxels, Proland, oceans & galaxies*
   - *From textures to fluids*
   - *More forestry*

   Purpose: *recipes for efficient modelization of complexity*

2. ***Abstracting some tools and principles***

context: **Why I'm doing all that ?**

- **Come from SFX** (TDI, AW)
  → End-users. Domain issues. Usability in prod.

- **Science deep lover** (understanding. popularization)
  → Catching phenos. Testing models. Perception.
  NB: Building representations *is* doing physics

- **Geek + Maths** (DESS/ENST, EDF)
  → Tools to be God

context: **Why I'm doing all that ?**

- **Come from SFX** (TDI, AW)
  → End-users. Domain issues. Usability in prod.

- **Science deep lover** (understanding. popularization)
  → Catching phenos. Testing models. Perception.
  NB: Building representations *is* doing physics

- **Geek + Maths** (DESS/ENST, EDF)
  → Tools to be God
  (But just tools. Don't let them be your master).

context: **Why I'm doing all that ?**

- **Come from SFX** (TDI, AW)
  → End-users. Domain issues. Usability in prod.

- **Science deep lover** (understanding. popularization)
  → Catching phenos. Testing models. Perception.
  NB: Building representations *is* doing physics

- **Geek + Maths** (DESS/ENST, EDF)
  → Tools to be God
  (But just tools. Don't let them be your master).
  (The real God is the user).

context: **Why I'm doing all that ?**

- **Come from SFX** (TDI, AW)
  → End-users. Domain issues. Usability in prod.

- **Science deep lover** (understanding. popularization)
  → Catching phenos. Testing models. Perception.
  NB: Building representations *is* doing physics

- **Geek + Maths** (DESS/ENST, EDF)
  → Tools to be God
  (But just tools. Don't let them be your master).
  (The real God is the user).
  (BTW, the real master is your computer).

1. A tour through various work pieces

1. A tour through various work pieces

- **Natural scenes** *(landscapes, forest, rivers & ocean, clouds…)*
  - ➔ *large + detailed + continuous front to back*
  - ➔ *complex ! (massive data, store, compute, pheno to simul…)*
- **representation** *of data & phenomena*
  - ➔ *shape, motion, material & shading, textures, light transport, GPU*
- **criterions**
  - ➔ *realism / plausible + real-time + controlable*
  - ➔ *minimalism, best representation*
  - ➔ *use a priori knowledge. bridges with “true” physics*
  - ➔ *don’t forget application & users*

What I did

From prehistory (namely: pre-internet) to now

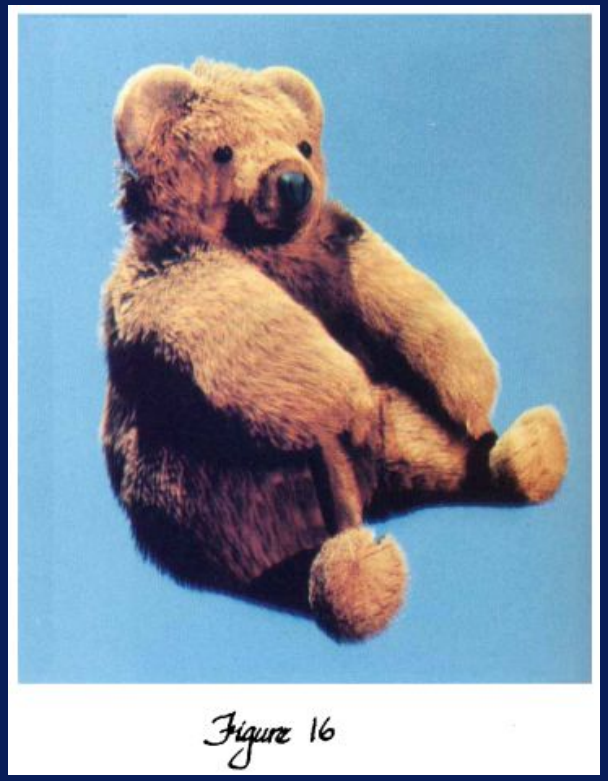
90-92:

1984: fonded
1987: ***Explore*** image synthesis software
1989: split software / production (→ ExMachina)
1993: SGI > Alias > wavefront > TDI (MS > softimage)

EXPLORE:
NURBS, process trees, IPR, …
→ Maya

wiki

Volumetric Textures



Rendering fur with three dimensional textures

Kajiya & Kay Siggraph'**89**

pdf

Volumetric Textures

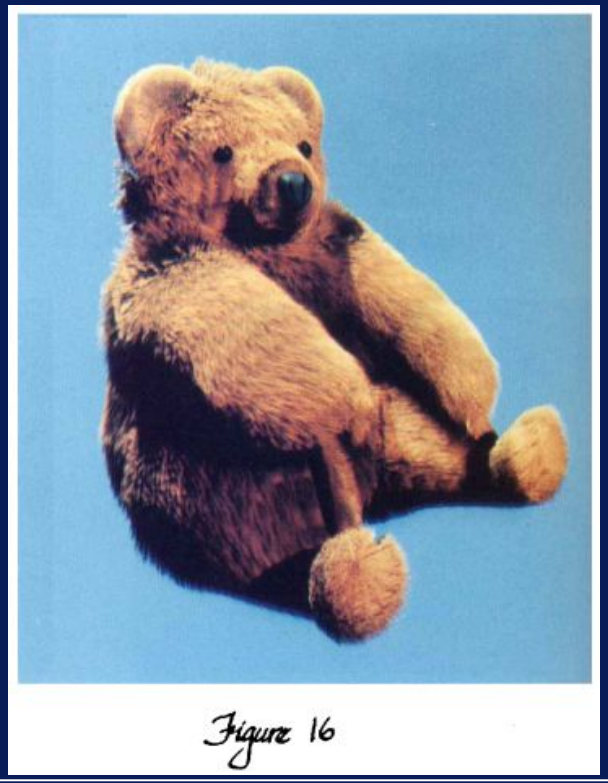


Figure 16

- volume = impressionism illusion
- hierarchy of models [Kaj86]
  geom → texture → shading LOD
- mapping shapes onto shapes
  (shape as a 3D material)

Rendering fur with three dimensional textures

Kajiya & Kay Siggraph'**89**


Figure 9

Figure 10

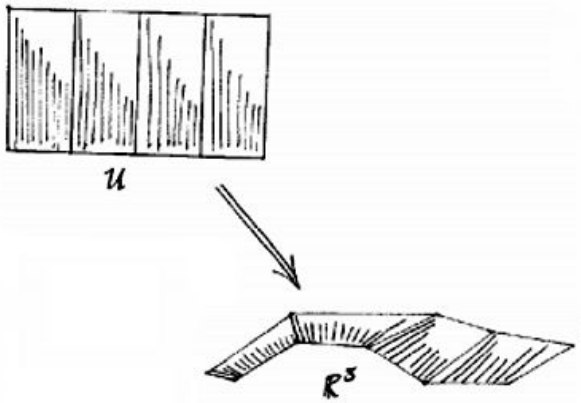


Figure 5

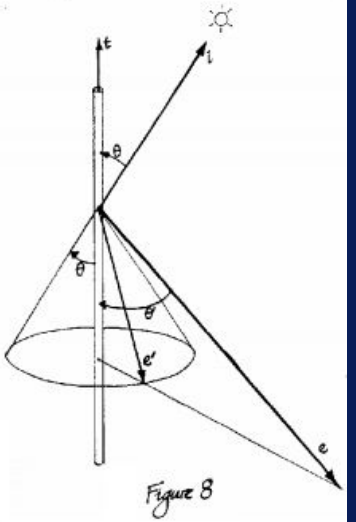
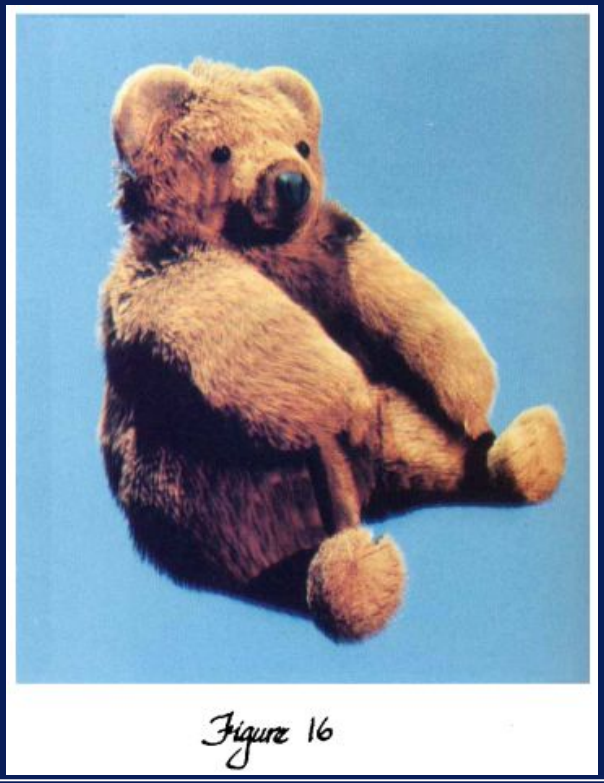


Figure 8

pdf

Volumetric Textures



Rendering fur with three dimensional textures

Kajiya & Kay — Siggraph'**89**

- volume = impressionism illusion
- hierarchy of models [Kaj86]
  geom → texture → shading LOD
- mapping shapes onto shapes
  (shape as a 3D material)

Limitations:

- hairs only
- no volume stored
- not filterable
- still costly

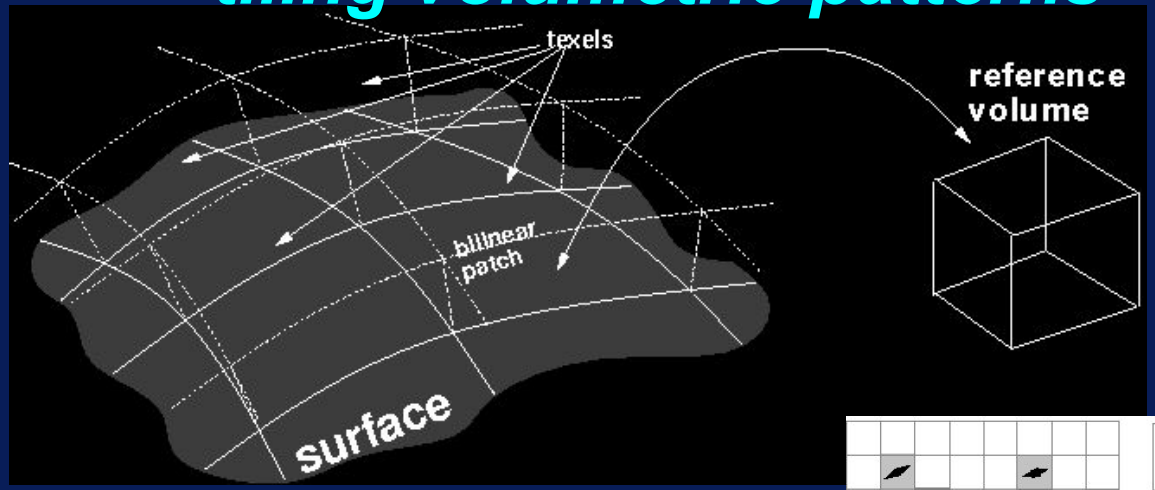
→ PhD topic found ! :-)

pdf

Volumetric Textures

94→98: *PhD. ray-tracing* 98→04: *on GPU*
with A Meyer, Ph Decaudin, ...

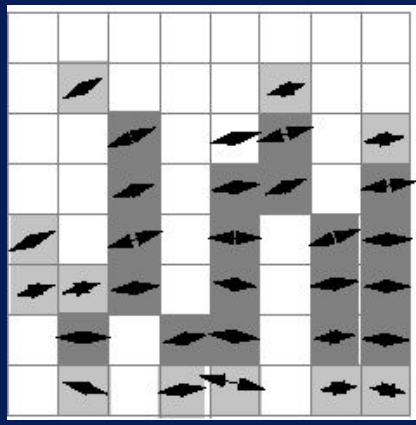
tiling volumetric patterns

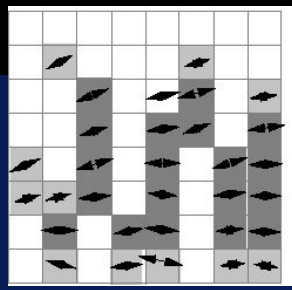


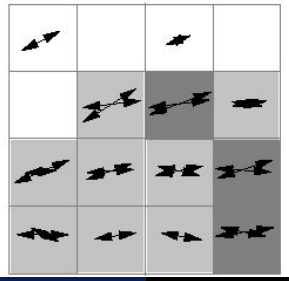
Volume:
- *SVO*
- *multiscale*

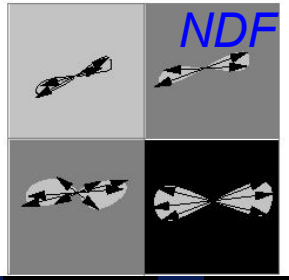
→ *octree of voxels*

Voxel data:
- *viewdep density*
- *reflectance*

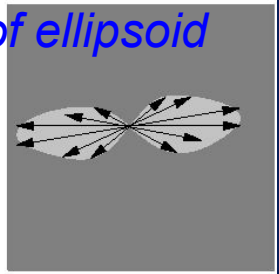


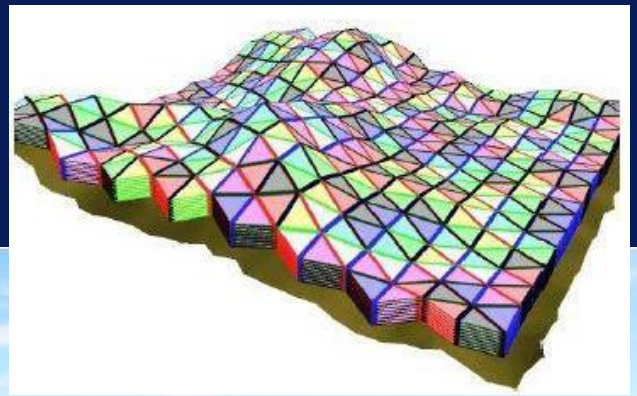






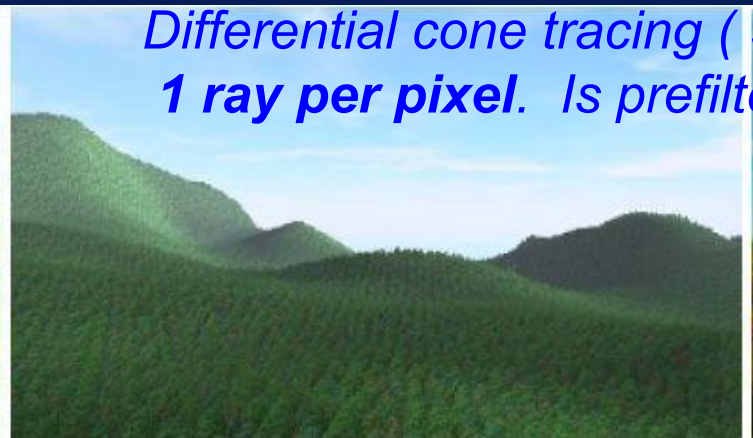
NDF of ellipsoid

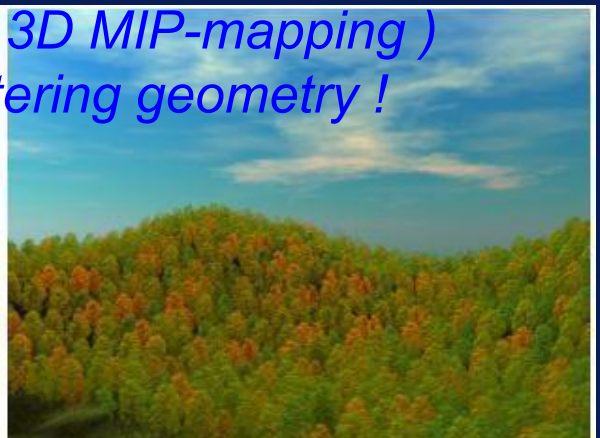




video torus | video forest | video flag | video lawn

Differential cone tracing (3D MIP-mapping)
***1 ray per pixel**. Is prefiltering geometry !*





SCA'02,EG'10: **Real-time ocean**, *with D. Hinsinger*

simulate a few wave trains ...only at useful pos & resol: dice'n displace

waves eqn (Σ trochoïds)

$$\begin{cases} x - x_0 = Ae^{kz_0}\sin(\omega t - kx_0) \\ z - z_0 = Ae^{kz_0}\cos(\omega t - kx_0) \end{cases}$$

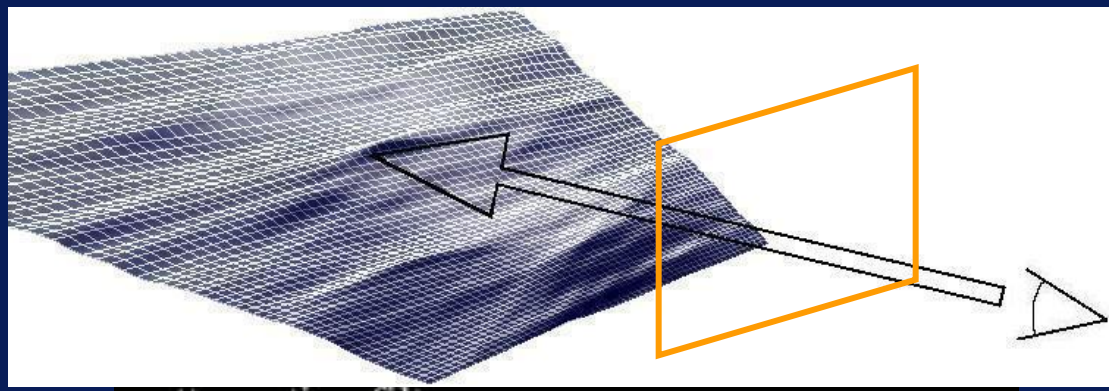
dispersion eqn

$$c = \frac{\omega}{k} = \sqrt{\frac{g}{k}\tanh(kH)}$$

amplitude spectrum (PM)

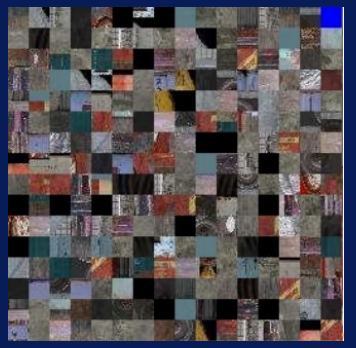
$$F_{PM}(f) = \frac{ag^2}{(2\pi)^4 f^5} e^{\frac{5}{4}(\frac{f_m}{f})^4}$$

$$F(f,\alpha) = F_{PM}(f)D(f,\alpha)$$



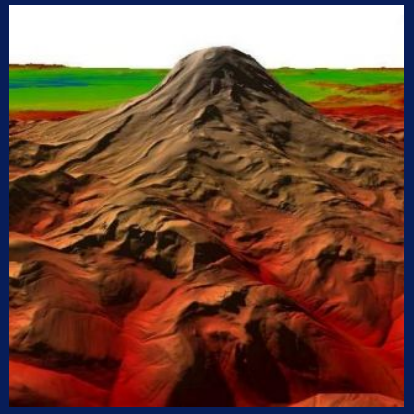
Textures: 03-05, with Sylvain Lefebvre

- Distortion-free pattern mapping [
  uv-mapping ill-posed → get rid of glo

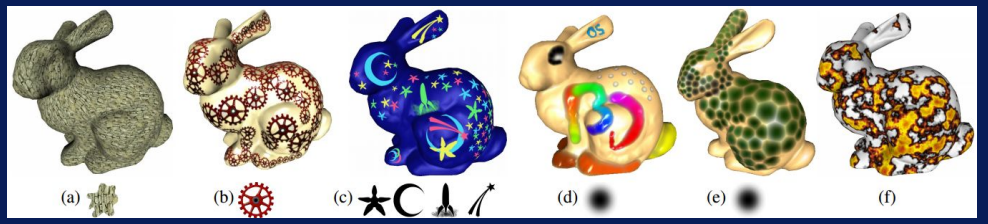


- Texture memory management

- huge detailled maps
- load in (GPU) memory only what's needed
  (view frustum, visibility, LOD)
  → out of core hierarchical tile cache. Load on demand



- stores only what is needed
  → octree textures ; bounding vol. projectors

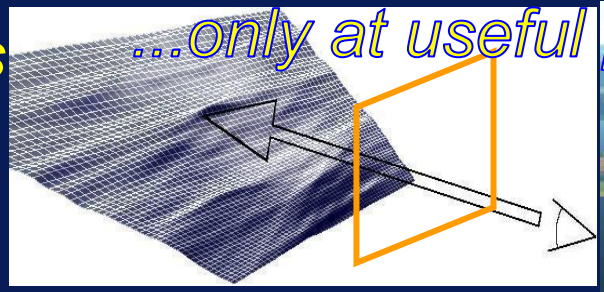


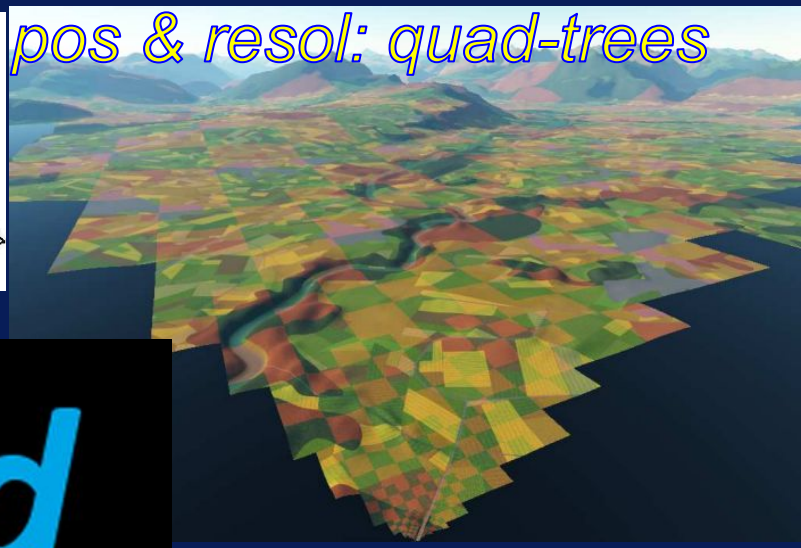
- Texture-space animation

- Mecanical textures: folds and cracks

08→12: PROLAND, with Eric Bruneton

- whole Earth, all scales
- out of core





Proland

http://proland.inrialpes.fr www

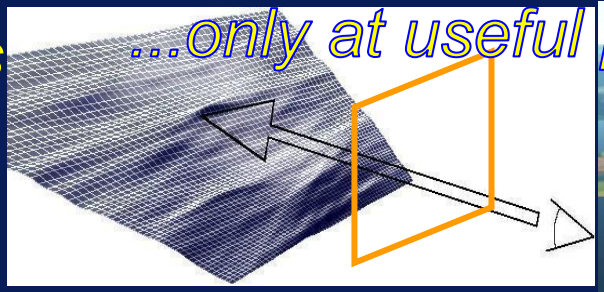
A C++/OpenGL library for the real-time realistic rendering of very large and detailed 3D natural scenes on GPU

video demo

Youtube E.Bruneton

08→12: **PROLAND**, *with Eric Bruneton*

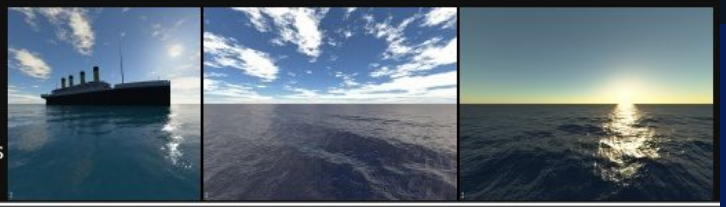
- *whole Earth, all scales*
- *out of core*



...only at useful pos & resol: quad-trees

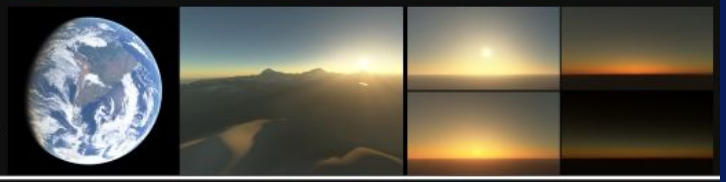
Real-time Realistic Rendering and Lighting of Forests
Bruneton Éric, Neyret Fabrice
Comput. Graph. Forum, **29** (2), ???-???, 2012.

Real-time Realistic Ocean Lighting using Seamless Transitions from Geometry to BRDF
Bruneton Éric, Neyret Fabrice, Holzschuch Nicolas
Comput. Graph. Forum, **29** (2), 487-496, 2010.



Scalable Real-Time Animation of Rivers
Yu Qizhi, Neyret Fabrice, Bruneton Éric, Holzschuch Nicolas
Comput. Graph. Forum, **28** (2), 239-248, 2009.

Precomputed Atmospheric Scattering
Bruneton Éric, Neyret Fabrice
Comput. Graph. Forum, **27** (4), 1079-1086, 2008.



Real-time rendering and editing of vector-based terrains
Bruneton Éric, Neyret Fabrice
Comput. Graph. Forum, **27** (2), 311-320, 2008.

A C++/OpenGL library for the r

video
demo

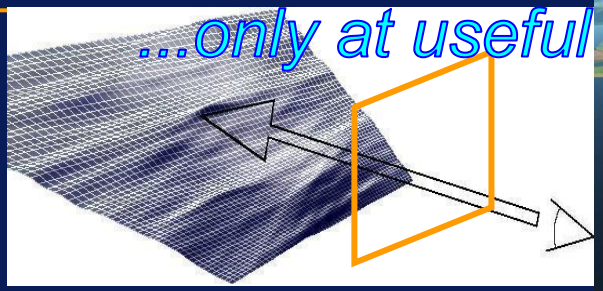
video
torus

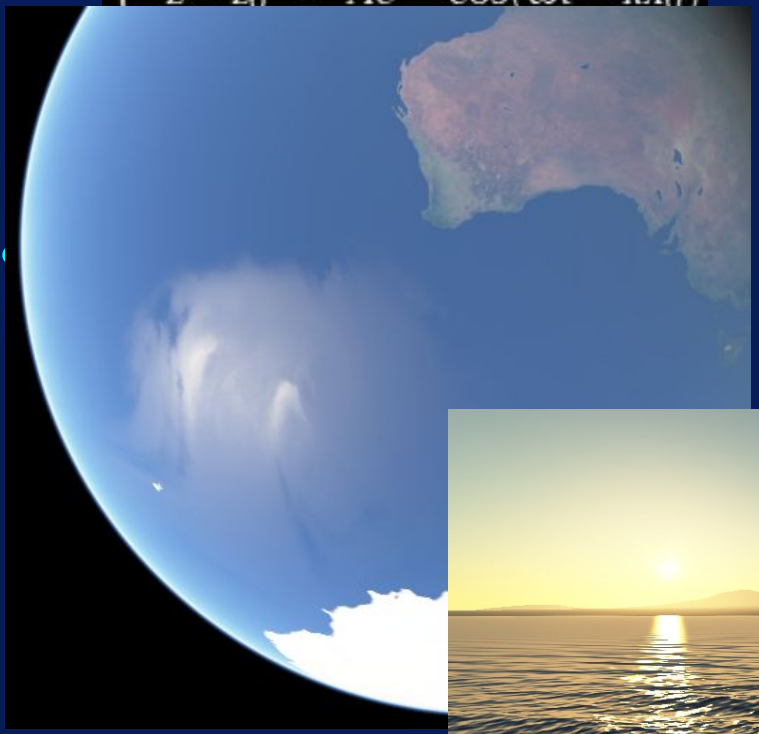
SCA'02, EG'10: **Real-time ocean**, *with Eric Bruneton*

simulate all waves (FFT)

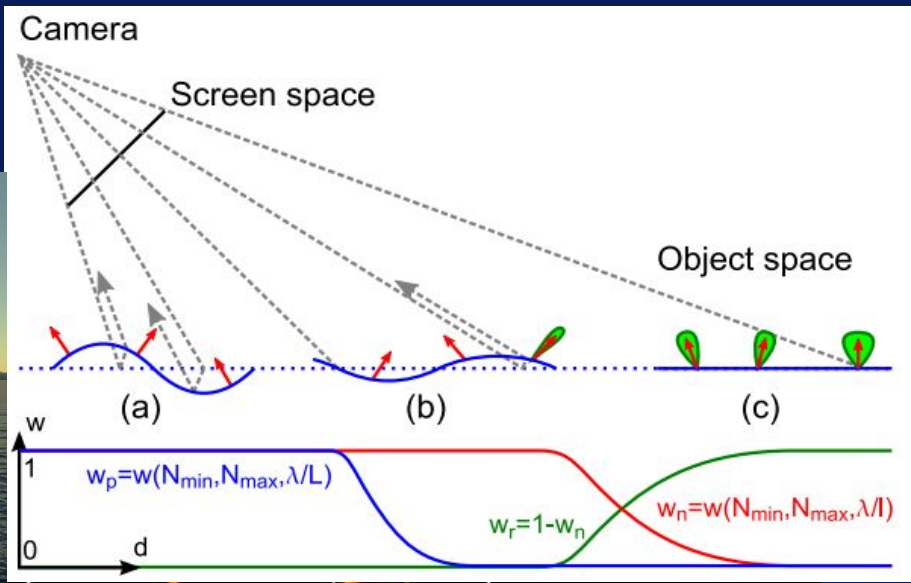
waves eqn (Σ trochoïds)

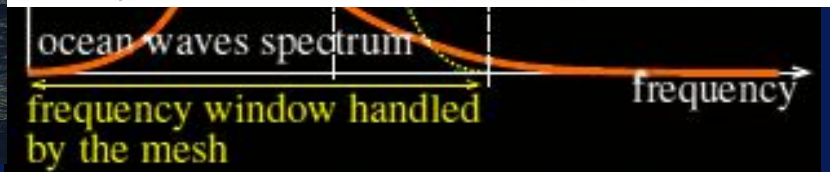
$$\begin{cases} x - x_0 = Ae^{kz_0}\sin(\omega t - kx_0) \\ z - z_0 = Ae^{kz_0}\cos(\omega t - kx_0) \end{cases}$$





Appearance filtering: *shape→N→BRDF*



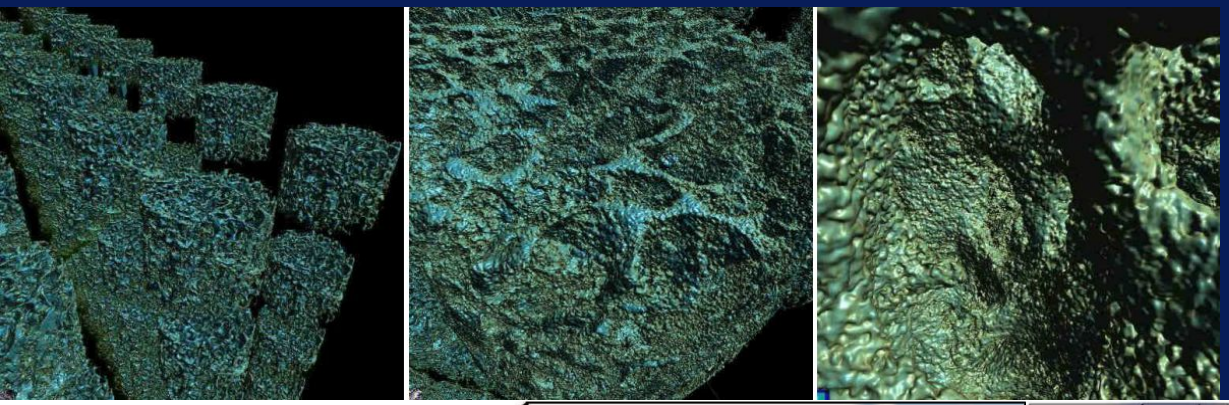


video ocean

video ocean

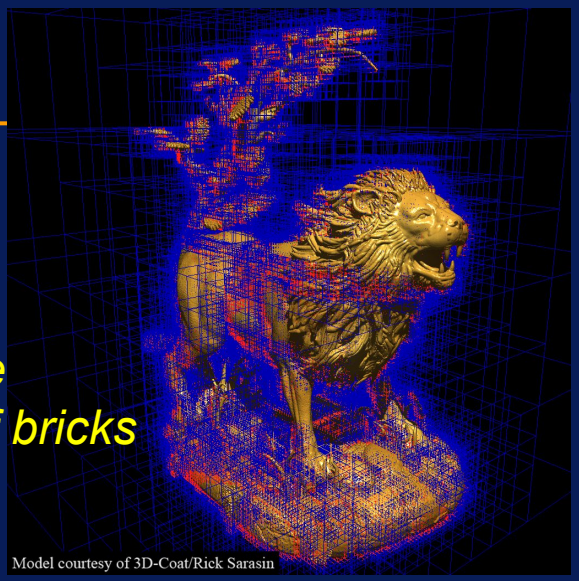
GigaVoxels *I3D'09, EG'11→ , with Cyril Crassin*

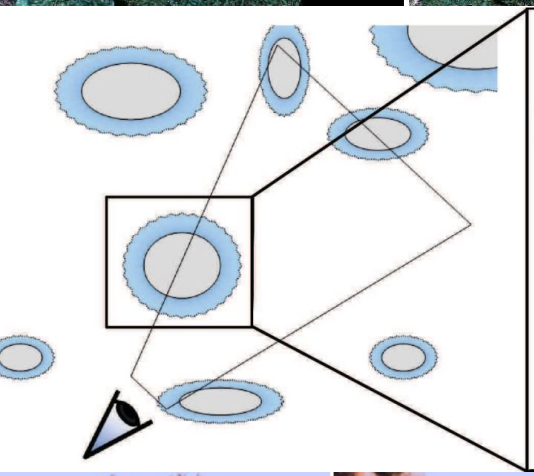
full volumetric scene, real-time

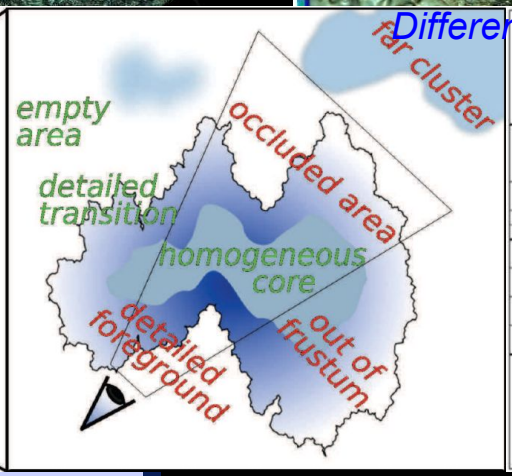


Volume:
- *SVO*
- *multiscale*
- *out of core*

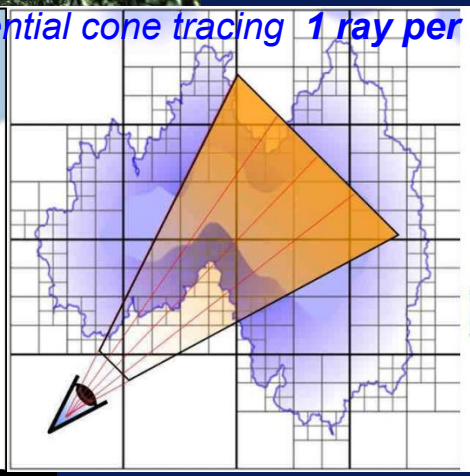
→ octree of bricks

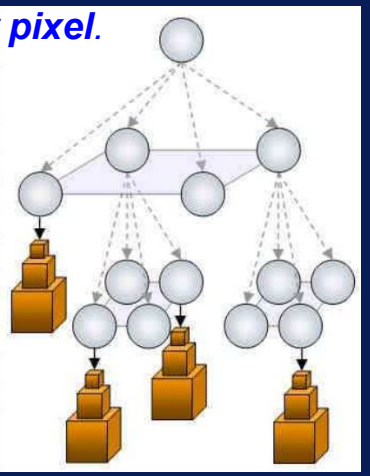


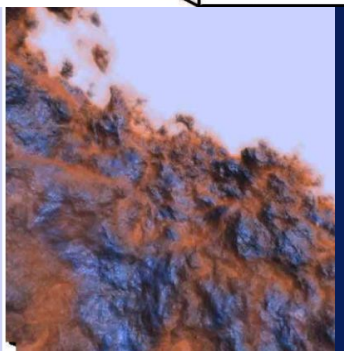


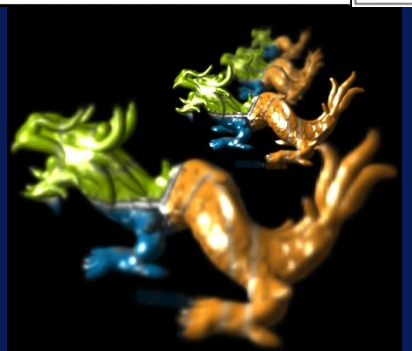


Differential cone tracing ***1 ray per pixel****.*









sort of geometry prefiltering

*→ ~**true shape LOD***

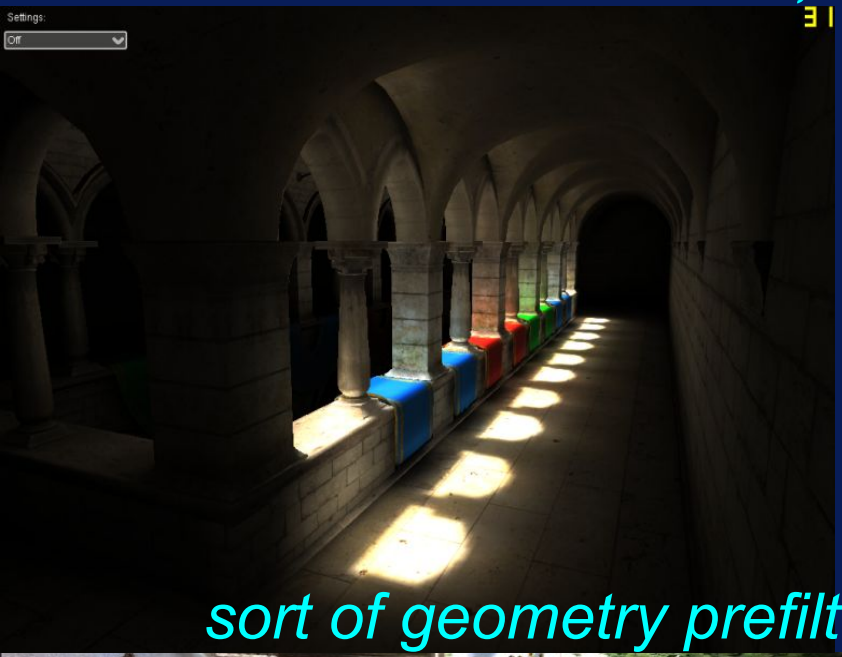
www

video
torus

Youtube
C.Crassin

GigaVoxels *I3D'09, EG'11→ , with Cyril Crassin*

full volumetric scene, real-time



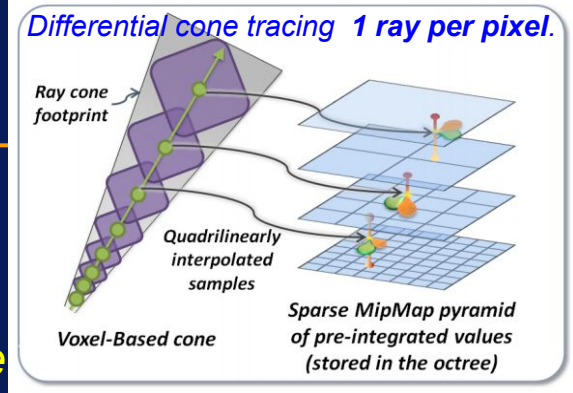
sort of geometry prefiltering

→ ~true shape LOD

Volume:

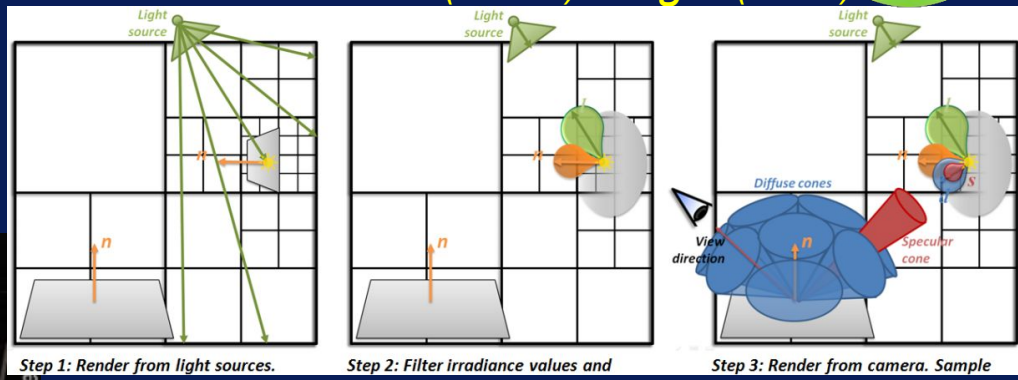
- *SVO*
- *multiscale*
- *out of core*

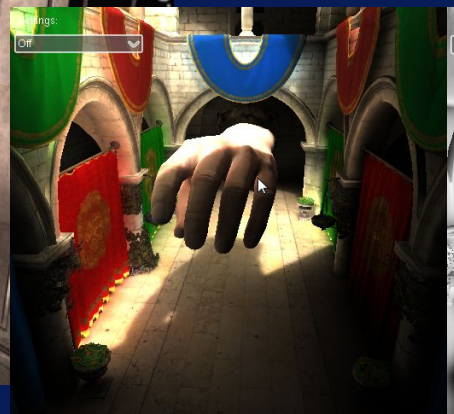
→ octree of bricks

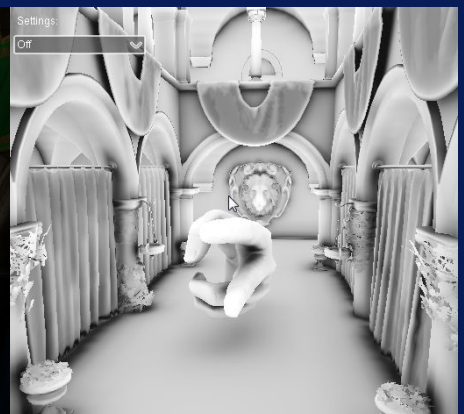


Voxel data:

- *viewdep density (6 dir)*
- *reflectance (lobes) & light (6 dir)*







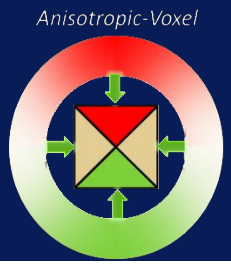
pdf

video
GI voxels

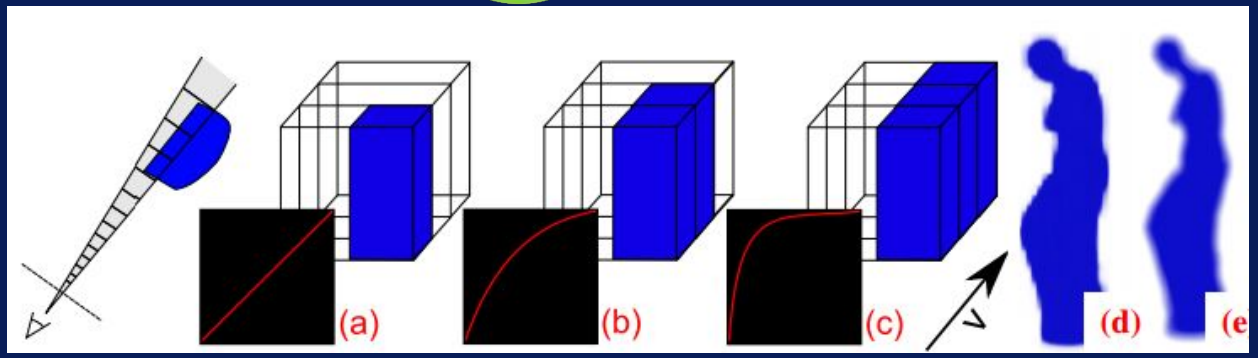
Appearance Filtering

~~*sort of* geometry prefiltering~~ → ***true*** *local* ***shape LOD***

issues with volumes of density:



*- **no viewdep** → 6 dir GIvoxels*
- fixed dir → only ok for (some) buildings



*- **no correlation**:*

- fat silhouette
- some light leaking

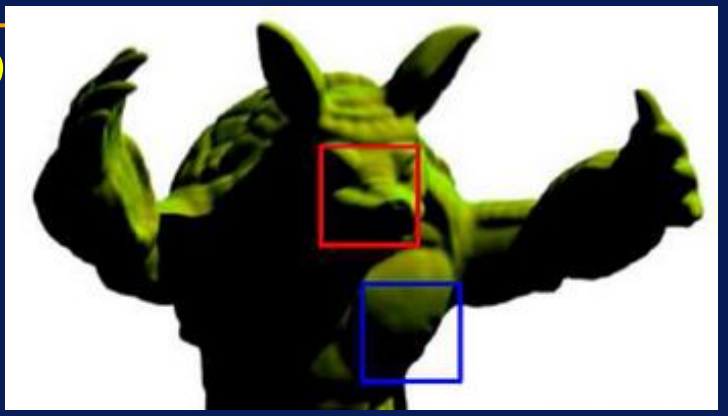
*→ density /extinction = **bad occupancy estimator for rendering purpose.***
opacity/transparency also.

- not view-dep → distrib
- not interpolate right either for xy or z: vis(mean(.)) # mean(vis(.))
- should interpolate differently in xy vs z !

→ need volumes of something else.

HPG'12,I3D'13, SIGA'13 **Appearance Filtering**, *with Eric Heitz*

~~*sort of*~~ *geometry prefiltering* → ***true*** $_{local}$ ***shape LOD***



- Small scale relief + visibility

→ ***all is view-dep and light-dep !***

- Correlations every where !

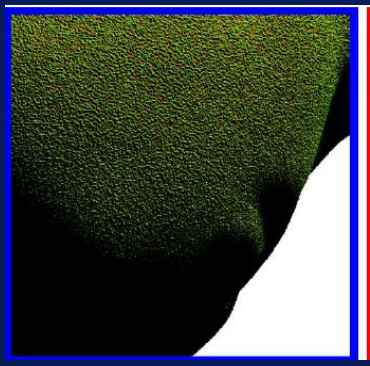
→ *light and colors* → *+ content correlation*

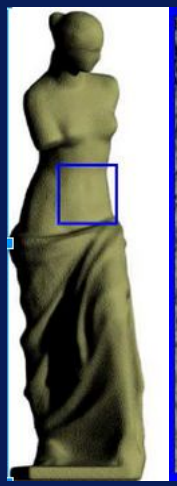
→ *normals* → *missing in all bumps*

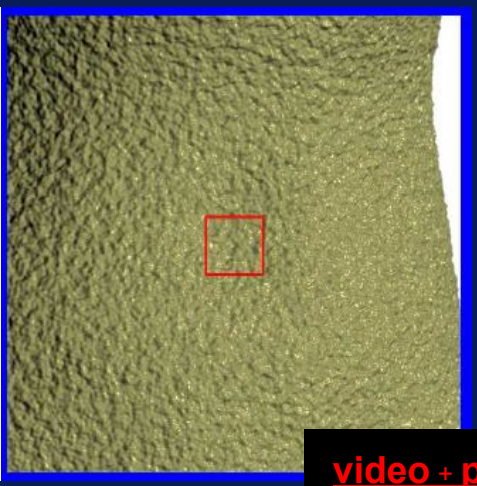
→ *visibility* → *microfacet models*

→ *occlusion*

...





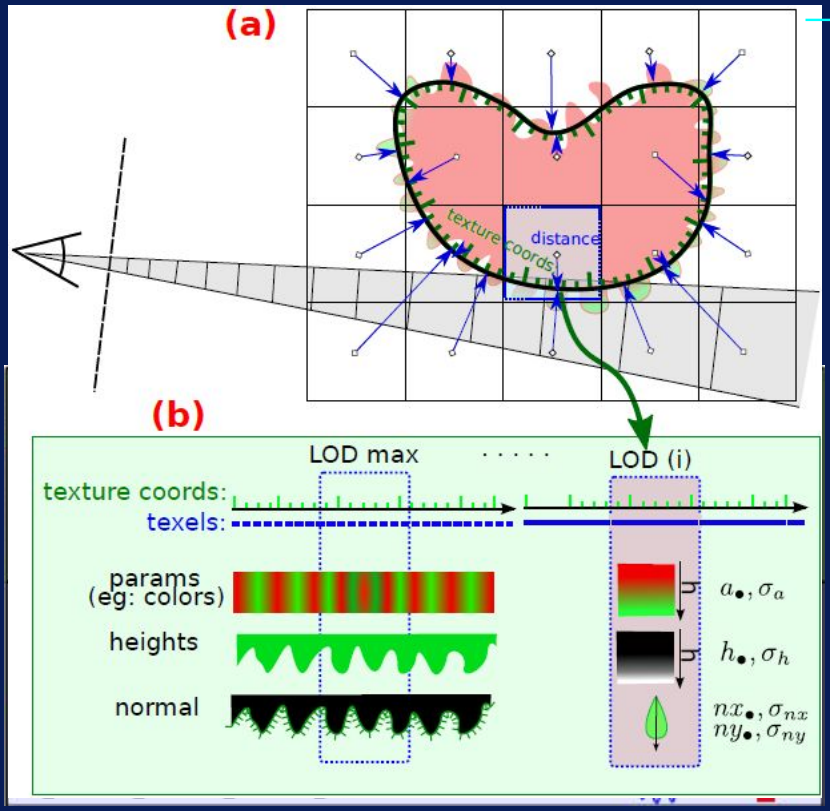


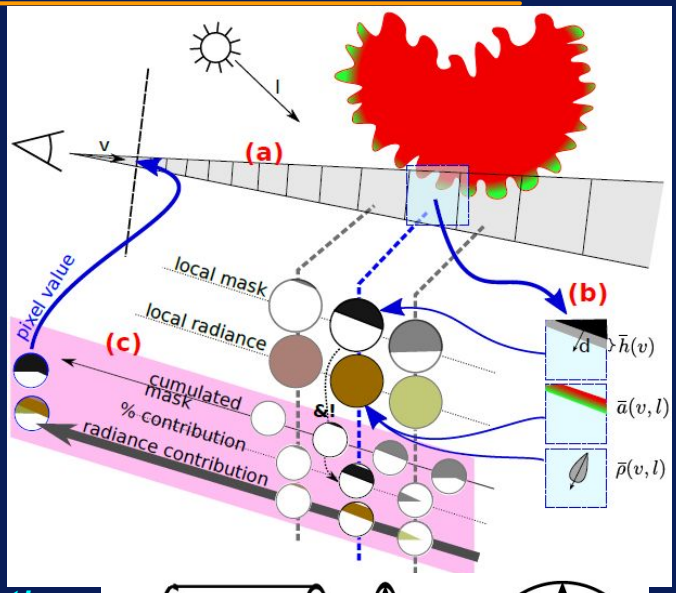
video + pdf

HPG'12,I3D'13, SIGA'13 **Appearance Filtering**, *with Eric Heitz*

1: *differential cone tracing*

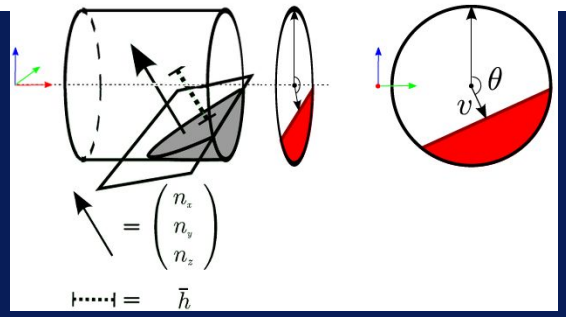
→ *select continuous LOD*





2: *3D A-buffer*

→ *solve z-correlation*



3: *Voxel data = distribs*

- *distrib h*
- *distrib N*
- *distrib param (col, …)*

distribs = N(m,σ^2) → *solve light- and view- dep*

→ *+ some content correlation*

macrosurface = SDF

assumed locally flat

details = height field

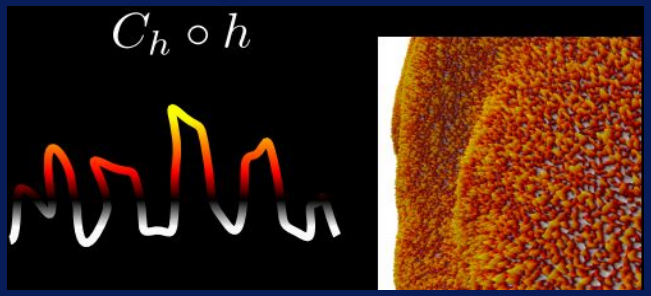
→ *Cook-Torrance shading*

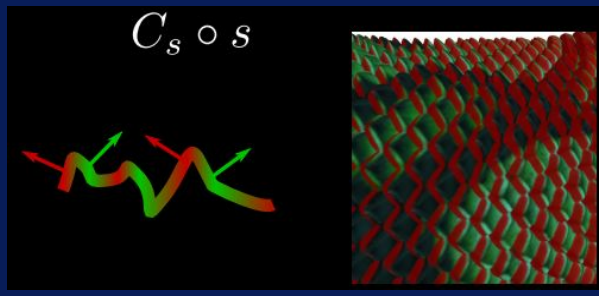
pdf

HPG'12,I3D'13, SIGA'13 **Appearance Filtering**, *with Eric Heitz*

content correlation

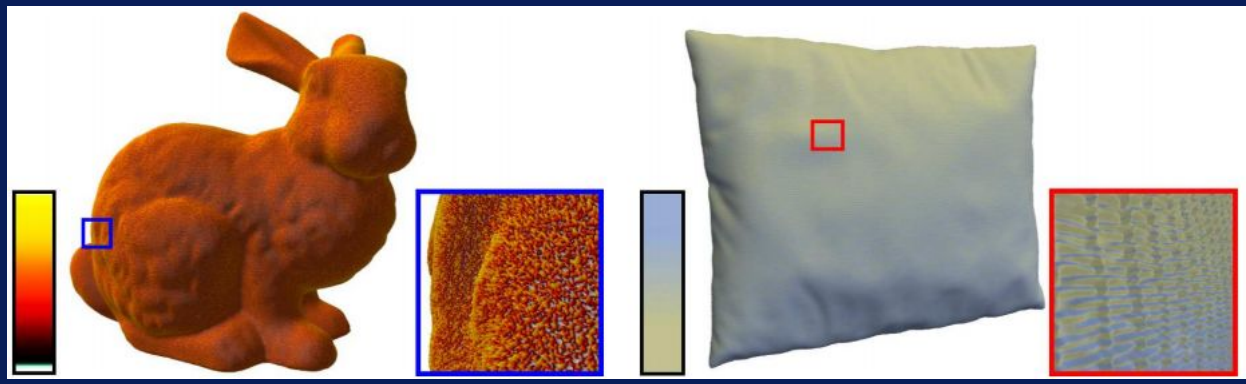
ex: color-height,
color-orientation

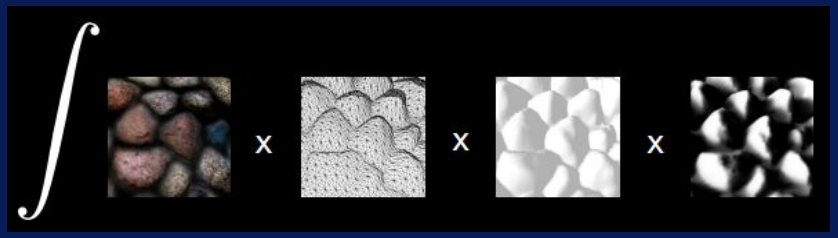


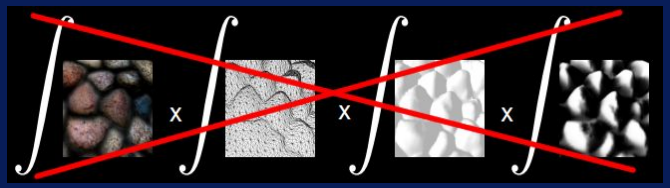


⇒ pixel integral not separable

$$I = \frac{\int_{\mathcal{P}} L_i(x,\omega_i)\, C(x)\, \rho(n_x,\omega_o,\omega_i)\, V_o(x)\, V_i(x)\, w_P(x)\, dx}{\int_{\mathcal{P}} V_o(x)\, w_P(x)\, dx}$$







video
forest

video+pdf

HPG'12,I3D'13, SIGA'13 **Appearance Filtering**, *with Eric Heitz*

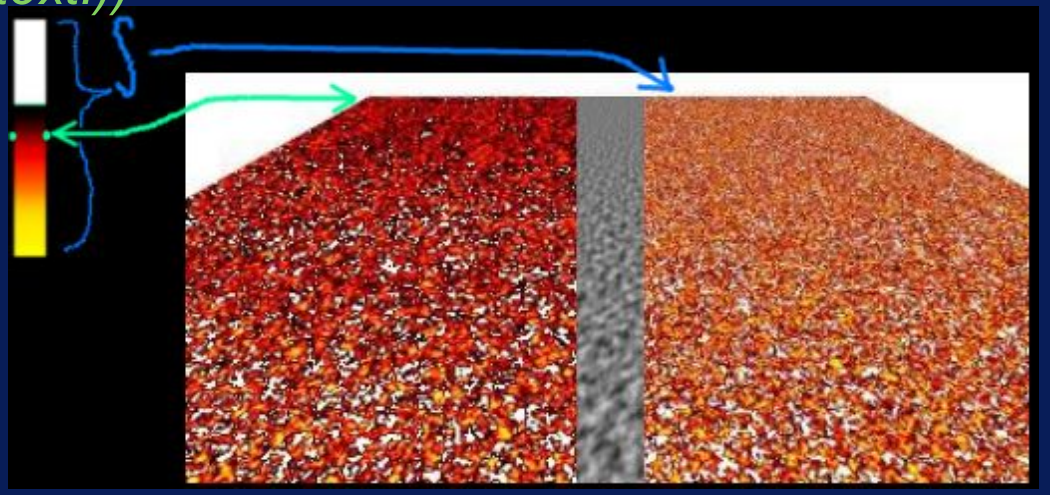
BTW, even color(texture) *is an issue*

→ also all f(noise): LUT, clamp, abs...

since

average(LUT(text.)) # *LUT(average(text.))*

$$\cancel{C_0(\int f)} \qquad \int C_0(f)$$



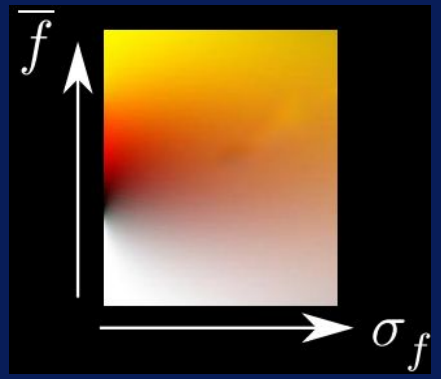
Idea: *use color distrib*

1: *average = LUT * histogram*

2: *histogram ~ gaussian*

$$\overline{C_0} = < \overset{v}{\underset{D_f}{\int}} , \; >$$

3: *simply precompute* iLUT(v,σ)



NB: applies to any distrib
e.g., heights ...

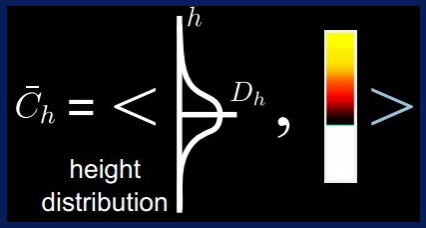
$$\bar{C}_h = < \overset{h}{\underset{D_h}{\int}} , \; >$$

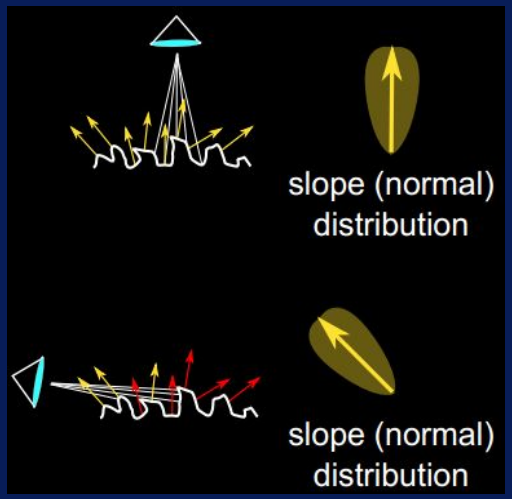
height distribution

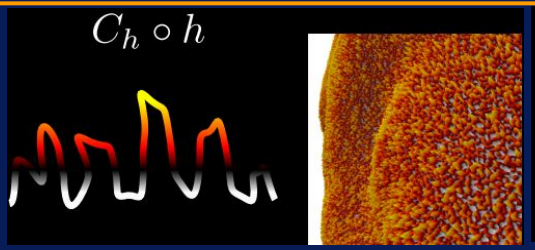
video
forest

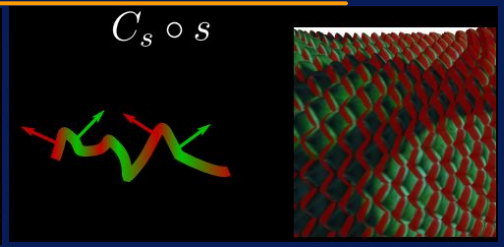
HPG'12,I3D'13, SIGA'13 **Appearance Filtering**, *with Eric Heitz*

...PB: screen-wise heights distribution is view-dep and light-dep



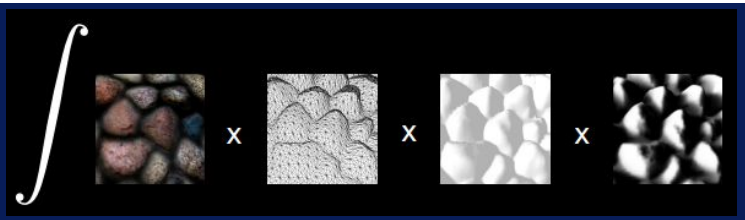






color-shape correlation ⇒ view+light correlation

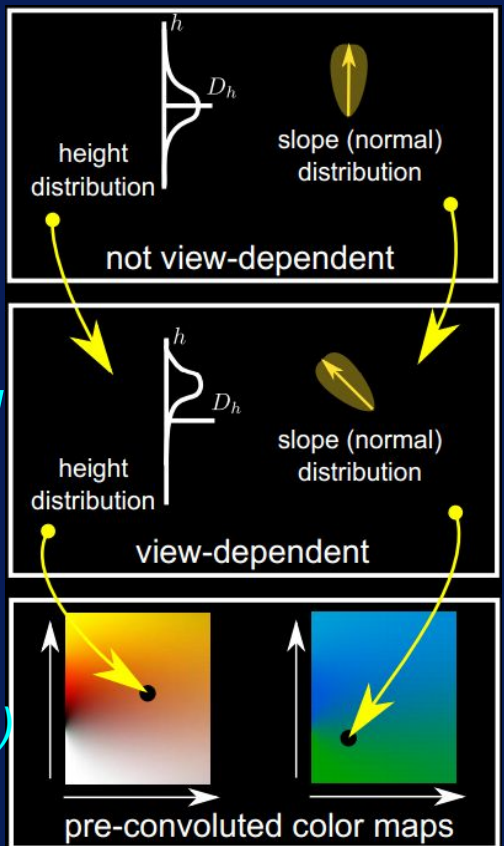
$$I = \frac{\int_{\mathcal{P}} L_i(x,\omega_i)\, C(x)\, \rho(n_x,\omega_o,\omega_i)\, V_o(x)\, V_i(x)\, w_P(x)\, dx}{\int_{\mathcal{P}} V_o(x)\, w_P(x)\, dx}$$



...but:

***4:** effect = lobe tilting*

→ easy !

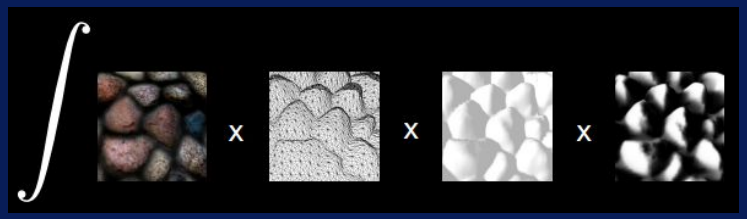


***5:** NB: for diffuse surface,*
effect of envmap = irradiance_map(N)
→ cf colormap(slope)

video
forest

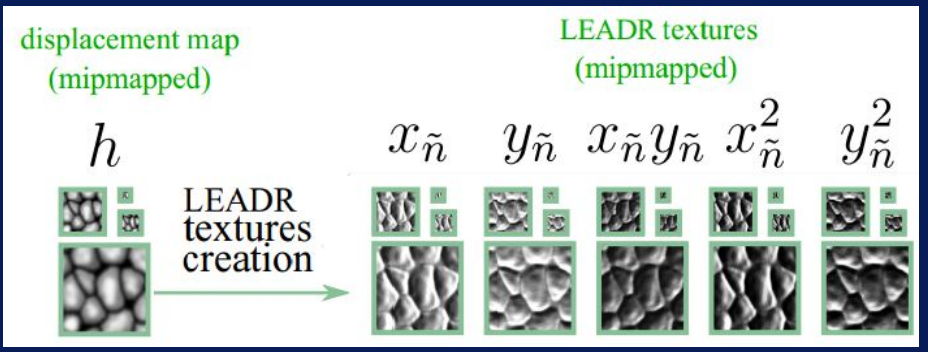
HPG'12,I3D'13, SIGA'13 **Appearance Filtering**, *with Eric Heitz*

$$I = \frac{\int_{\mathcal{P}} L_i(x,\omega_i)\, C(x)\, \rho(n_x,\omega_o,\omega_i)\, V_o(x)\, V_i(x)\, w_P(x)\, dx}{\int_{\mathcal{P}} V_o(x)\, w_P(x)\, dx}$$

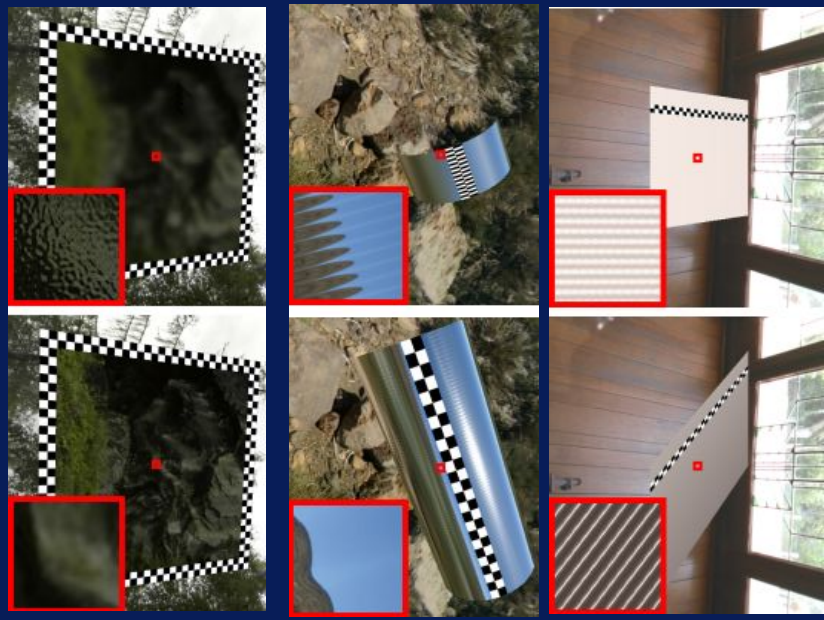


***LEAD-R**: displacement -> microfacets , with:*

- tilted lobe
- anisotropic
- true masking is Smith, not Cook-Torrance
- point light + IBL



Effect of geometric distortion on appearance:



video
T-rex

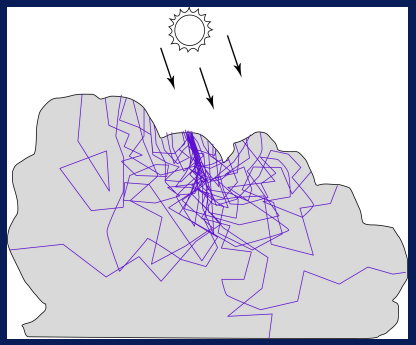
video
prop

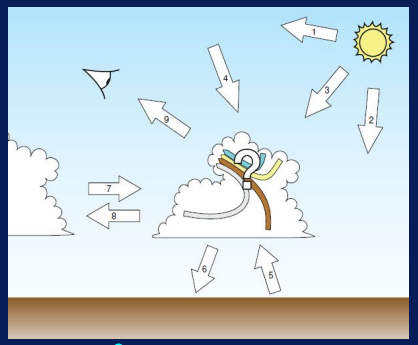
pdfs

*geometry prefiltering → ~**true** local **shape LOD***

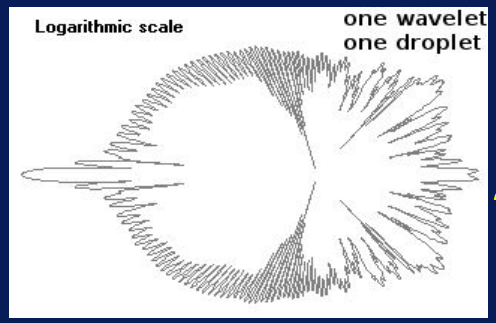
EGNP'06,I3D'08: **Realistic clouds in real time,** *with Antoine Bouthors*

simulating all light paths:
hard pb.
→ real time !



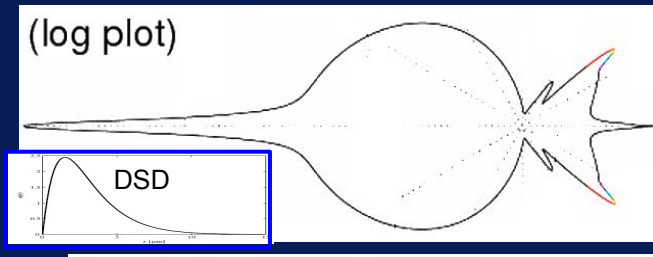


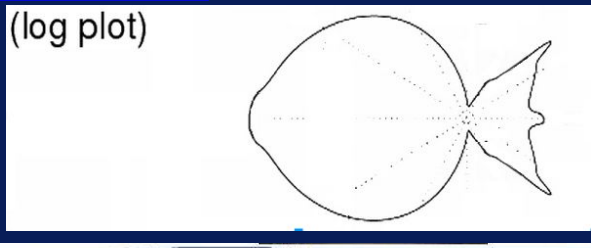
reflectance:
Mie :-s
absorption: 0



1: $\int$ *Droplet Size Distrib*
→ cancels Bessel oscillations

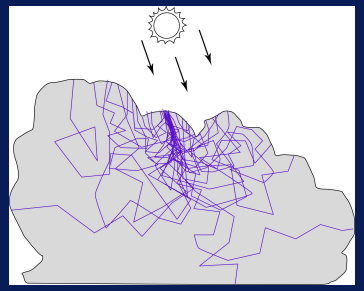
2 : $n_{scatter} > 1$
→ - peak (50% E) ~= no hit
- high freq useless
- no colored back-scatter

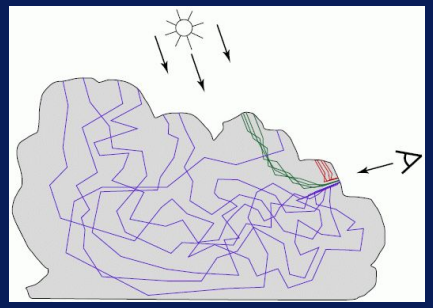




EGNP'06,I3D'08: **Realistic clouds in real time,** *with Antoine Bouthors*

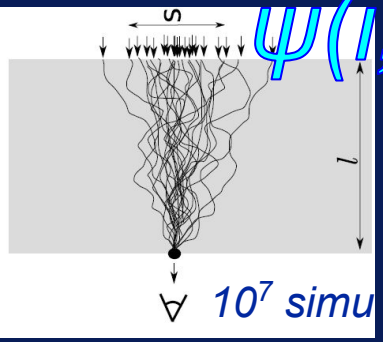
simulating all light paths:
hard pb.
→ real time !

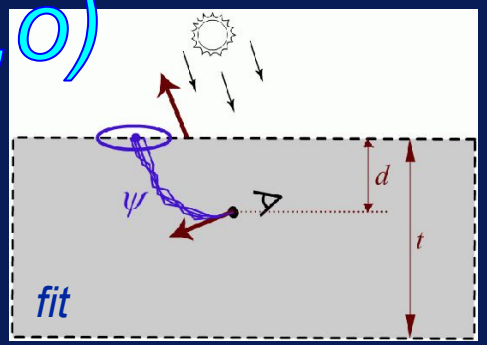


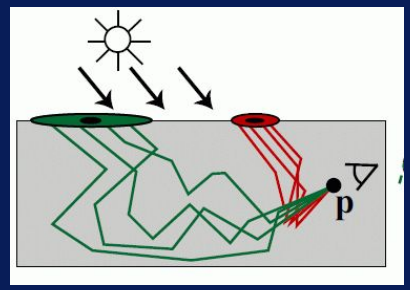


3: *separate scattering orders*
→ shift Most Probable Path and scale of transport

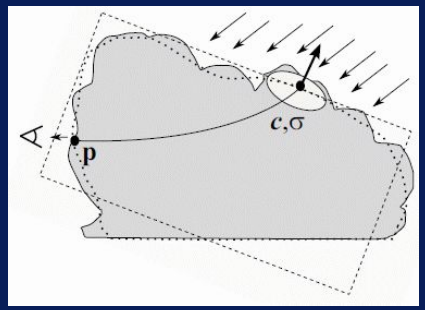
4: *macro-material* $\psi(i,o)$ $(L,V,Vpos,thick.)_{5D}$ ➔ $collector(pos,\sigma)_{3D}$

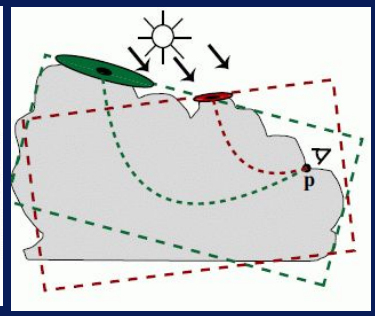




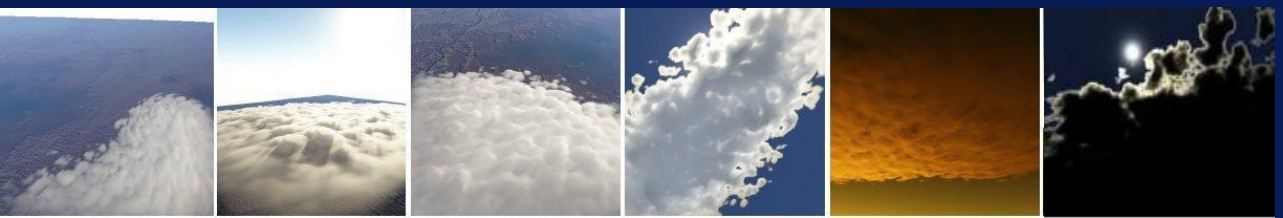


5: *solve i=collect(o) for cloud shape*





5: *ground ⟺cloud radiosity, sky illu*

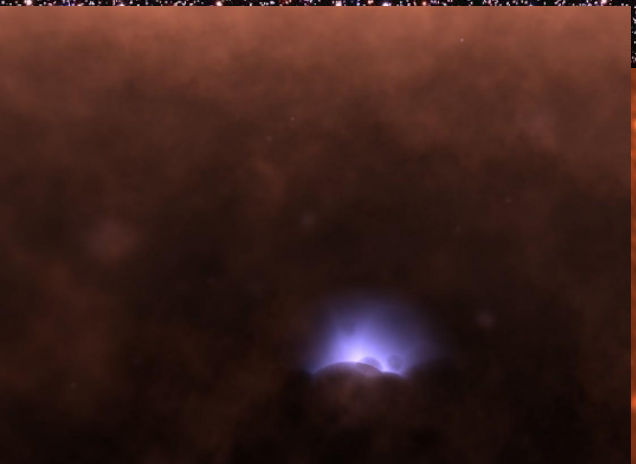


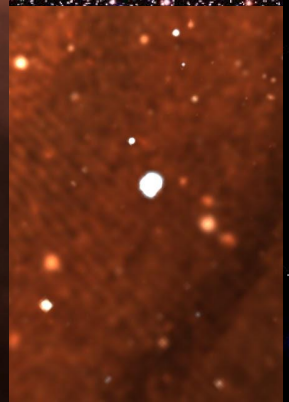
11→ ANR veRTIGE: **Galaxy Project**, with RSA Cosmos & Paris-Meudon Observatory

- Real time walk-through
- ~ Hubble quality
- far / close / Earth sky

GigaVoxels

- Huge scale span, all transp
- spectral
- non uniform spreadings
- correlated stars/clouds





a few other trampings

a continuum from texture to smoke

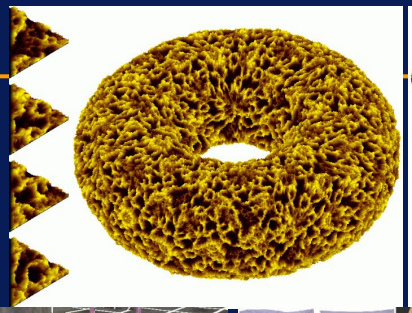
Textures: 99-05: with MP Cani, S Lefebvre, ...

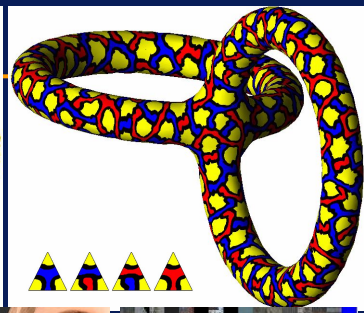
- Distortion-free pattern mapping [Sig'99]
  - uv-mapping ill-posed → get rid of global param

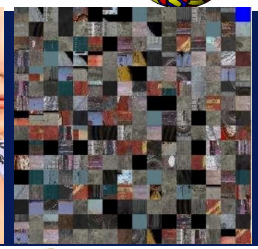
- Texture memory management
  - out of core hierarchical tile cache - on demand
  - octree textures. B.V. proj

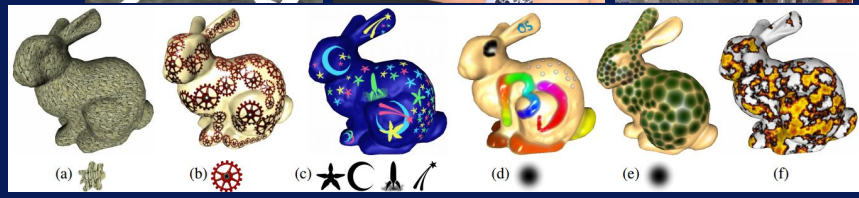
- Texture-space animation

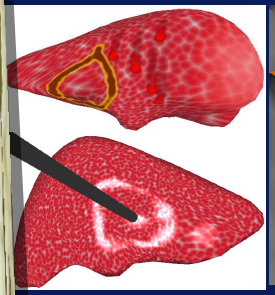
- Mecanical textures: folds and cracks

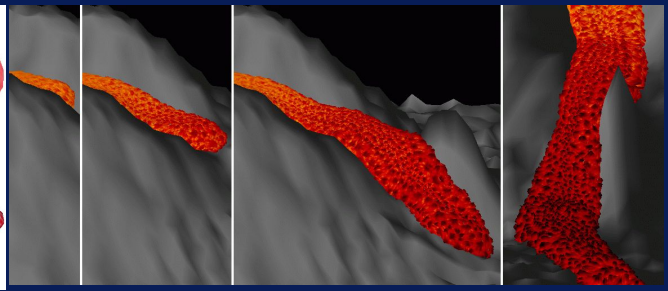


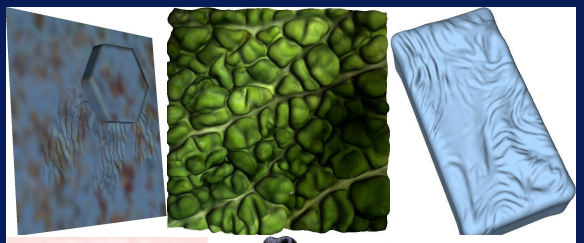


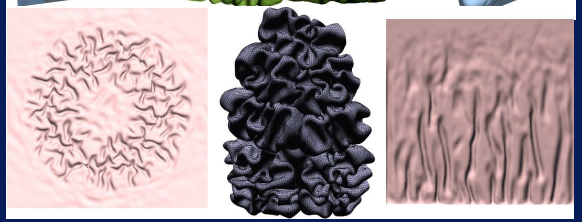


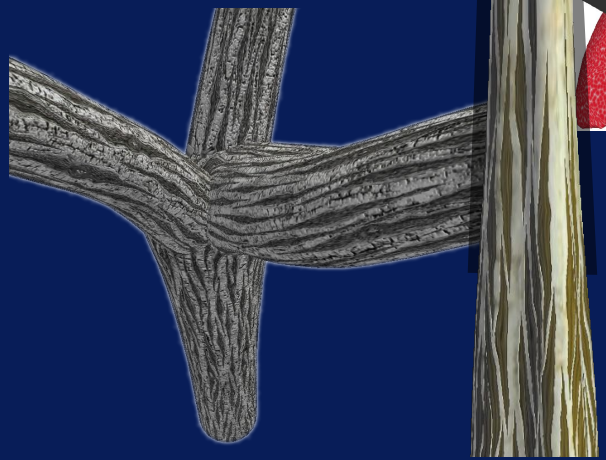










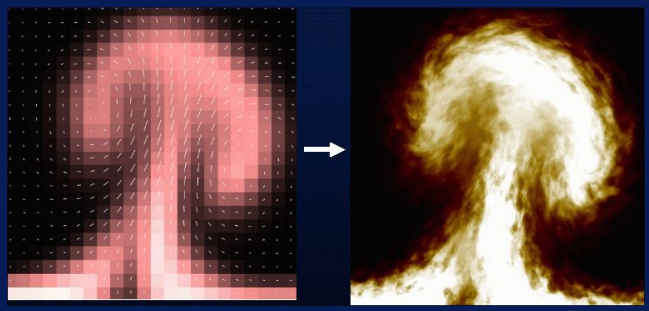


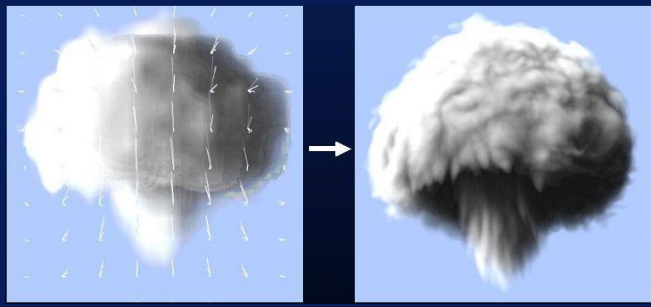
SigSketch'01,SCA'03, Sig'07,TVCG'11: Advected textures

Amplify fluid simu with textures

paradox: follow fluid + keep aspect (spectrum)

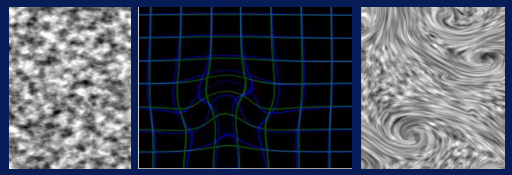
*idea: **1**:advect'n renew **2**:sub-anim **3**:couple scales*



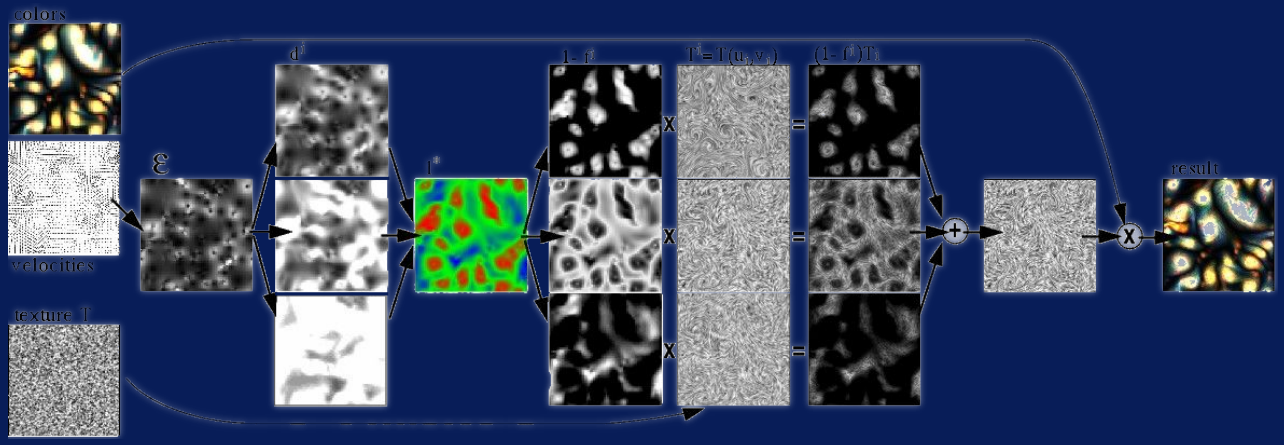


***1a**: base illusion:*

3 channels of dephased uv-advection



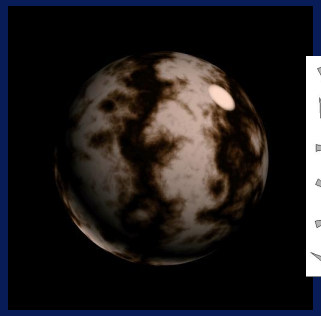
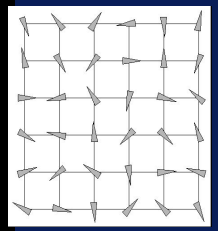
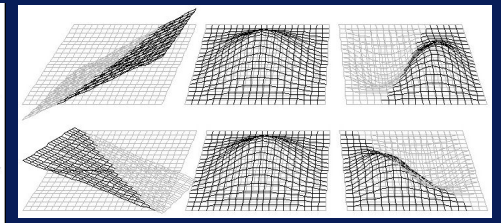
***1b**: N layers, cycling ~distortion rate → MIPmap(lod=E disto rate)*



SigSketch'01,SCA'03, Sig'07,TVCG'11: Advected textures

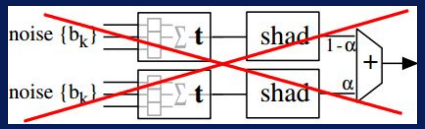
Amplify fluid simu with textures

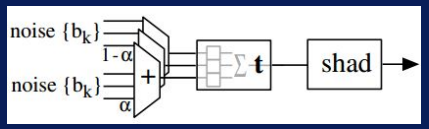
2a: flownoise [PN01]

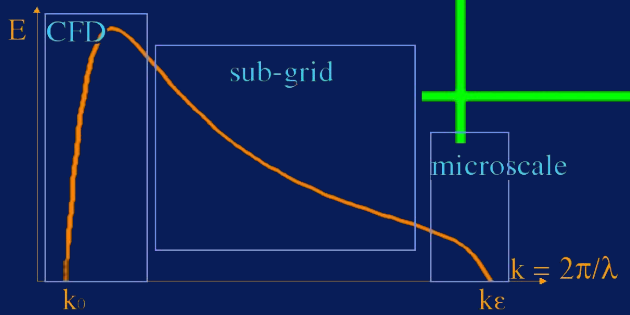

3: sub-grid animation: turbulence scaling law (Kolmogorov)

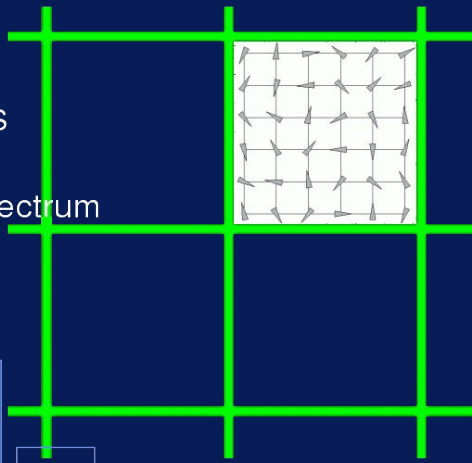
2a: procedural: don't blend (→ ghosting), morph (→ blend params)





- Flownoise for sub-scales
  - → rotations ≡ vorticity spectrum
  - → Kolmogorov cascade

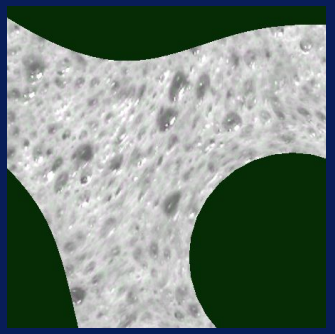




video + pdf

SigSketch'01,SCA'03, Sig'07,TVCG'11: Advected textures

Amplify fluid simu with textures

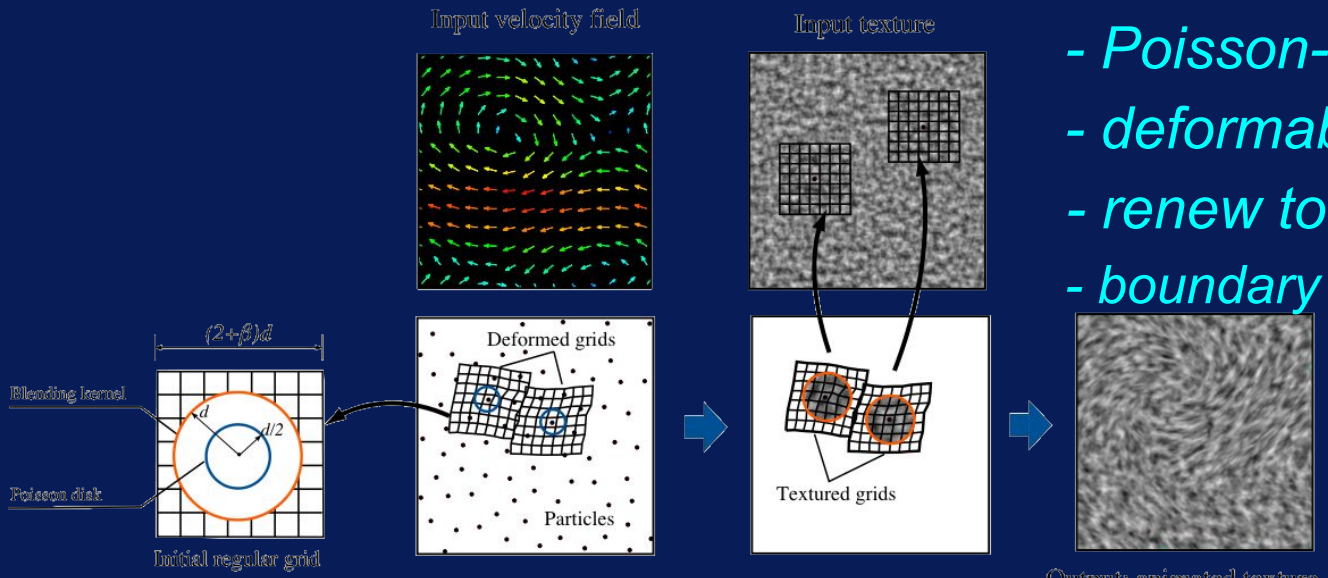


grid-based (Eulerian):

- too global scale
- too synchronous renew
- finite domain

→

1': Lagrangian texture advection *(local, asynchrone, unbounded)*



- Poisson-disk particles
- deformable sprites
- renew too deformed partics
- boundary conditions

video steps

pdf

video result

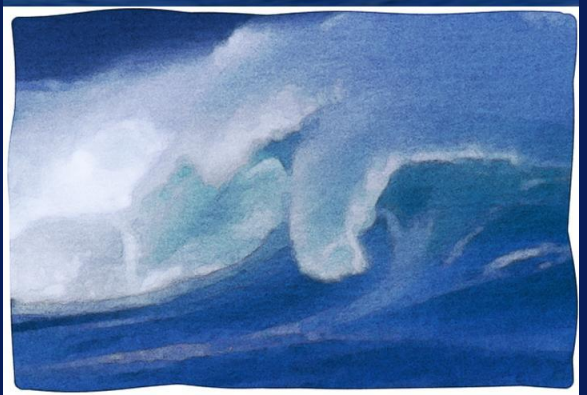
- *K. Perlin*
- *D. Salesin*

- *Digital Domain (Pirates of the Caribbean 3)*
- [PN01]: *Pacific Data Images (shrek)*
- *Adobe*

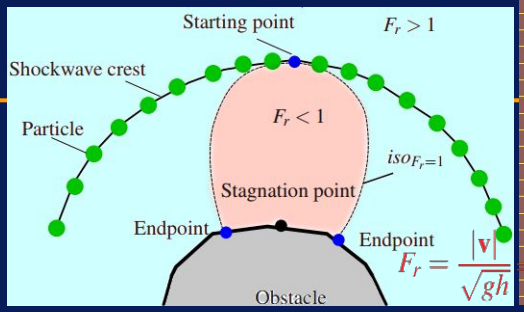
[Sig'07]:
animated paintings

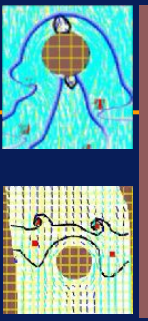
V field =optical flow

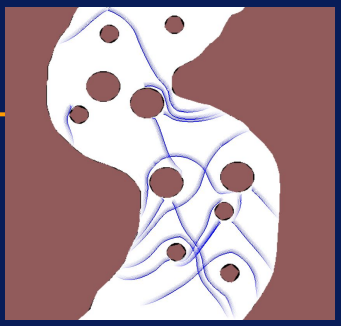
inward+backward adv.



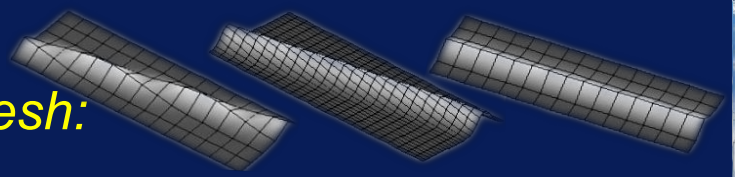
01-11: Rivers, with NP,QY,...



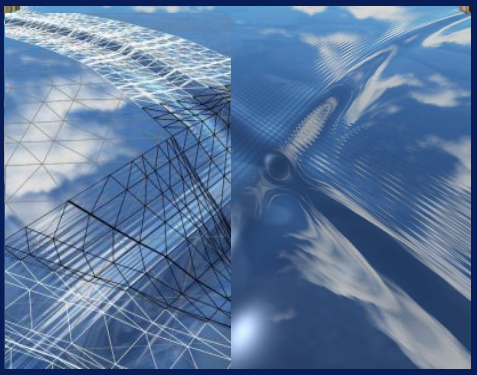




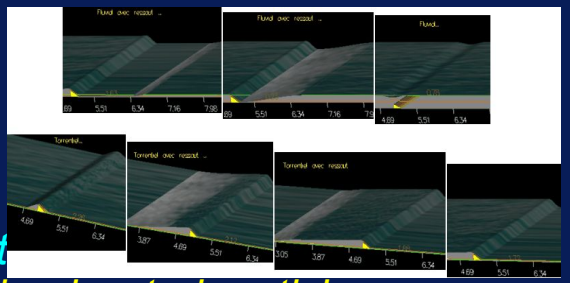
- direct simu of surface features:
  - vector shockwaves & streakwaves
  - wave propag in quasi-stationary flow
  - advected perturbations



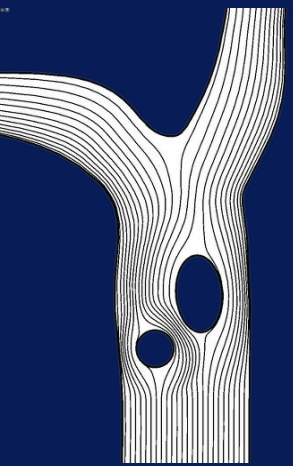
- capillary waves (~1mm)
  with light & aliasing-free mesh:
  - align to features !

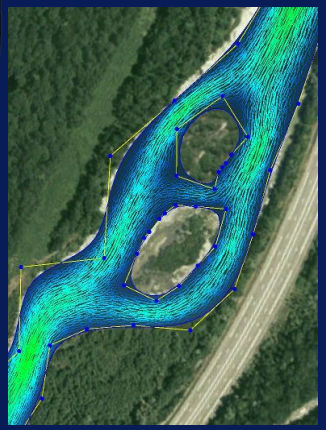


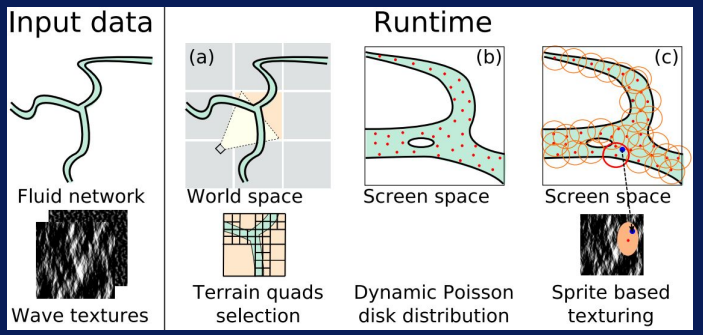
- direct simu of surface features:
  - hydraulic jumps. fluvial / torrential

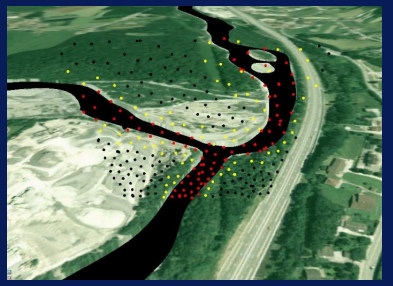


- scalable editable river (in Proland)
  - analytical flow (real-time generation & edit
  - lagrangian dicing: screenwise Poisson-disk advected particles



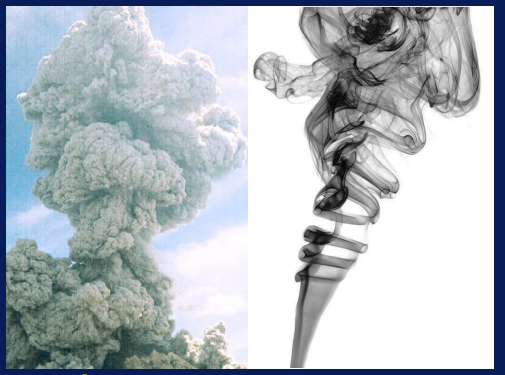


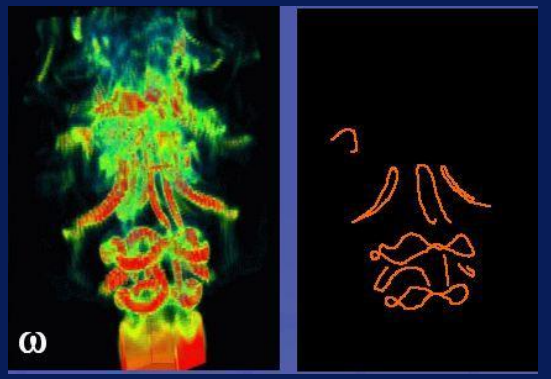


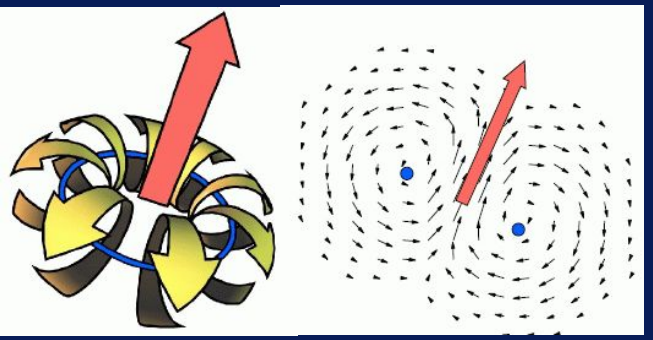


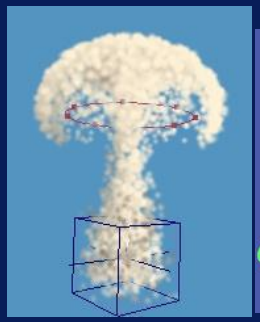
SCA'05,06: **Fluids as vortex filaments**, *with Alexis Angelidis*

- "soul" of fluid motion
- compact, highres, controlable…
- closer to std CG workflow









Lagrangian curves

moving quantity / representation	Velocity **v**	Vorticity **ω**
Eulerian	popular	
Lagrangian		our method

$$\mathbf{w} = \nabla \times \mathbf{v}$$

$$\mathbf{v} = \iiint_x \frac{(\mathbf{p}-\mathbf{x}) \times \mathbf{w}}{4\pi \|\mathbf{p}-\mathbf{x}\|^3} \, d\mathbf{x}$$

$$\mathbf{v} = \int \frac{\Gamma \boldsymbol{\omega} \times (p-x)}{4\pi \|p-x\|^3} dl$$

local space rotation operator

$$\frac{d\mathbf{w}}{dt} = \mathbf{w} \cdot \nabla \mathbf{v}$$

$$\Gamma = \int_L \mathbf{v} \cdot d\mathbf{l} = \iint_S \boldsymbol{\omega} \cdot d\mathbf{S}$$

+ vtx noise
+ ellipsoid particles

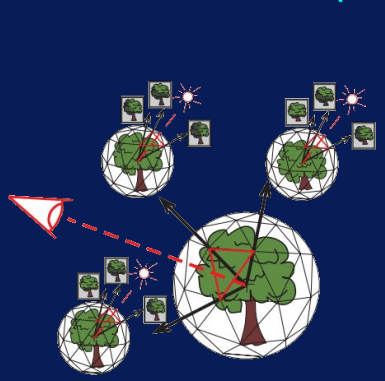
a few other trampings

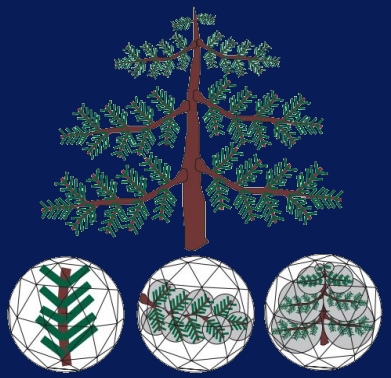
more forestry

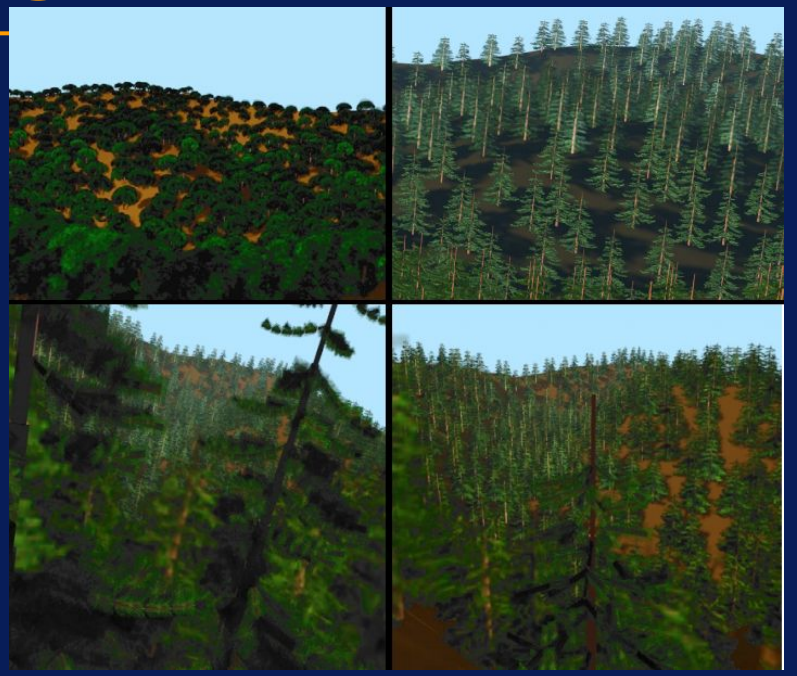
(and fancy representations)

EGSR'01: Multiscale shadowing

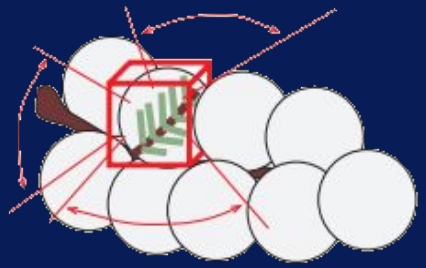
1: *(small) BTF(L,V) x 3 LOD*

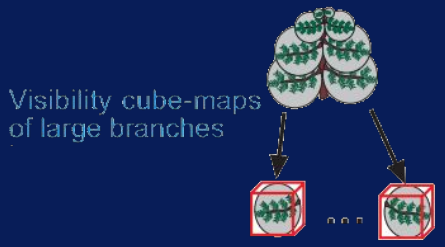


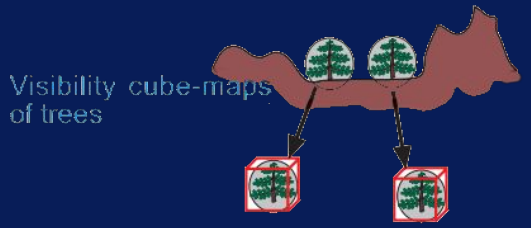


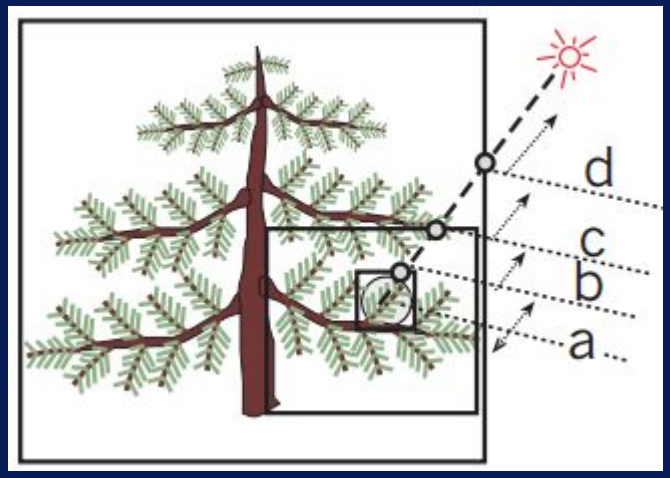


2: *Hierarchical visibility (small) cubemaps*









EG'12: **Endless forest**, with Eric Bruneton

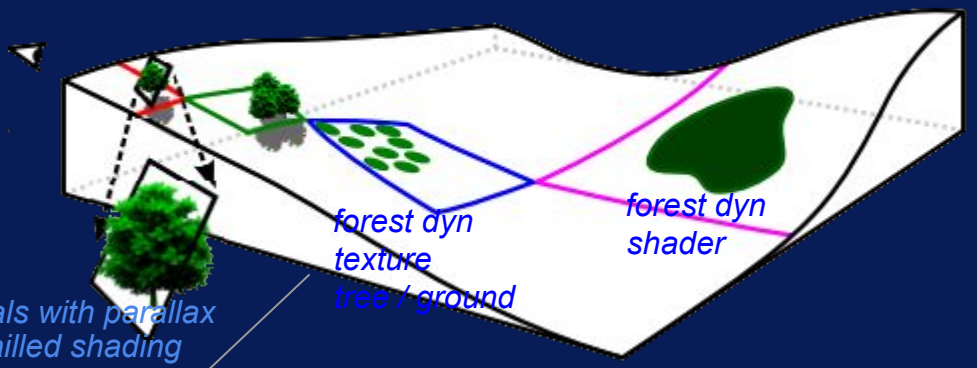
in Proland → *all scales, real-time, seamless LOD*

realistical: sun+sky, silverlinings & transparencies,
all-scales correlations (hot spot) + shadowing (ambiant occlusion)

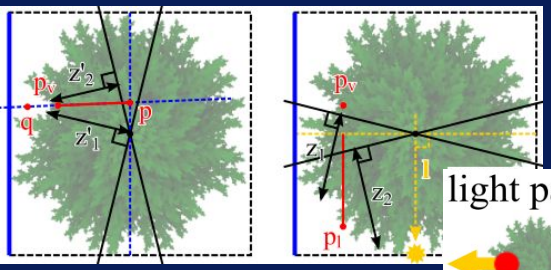
EG'12: **Endless forest**, with Eric Bruneton

- several tree species
- Poisson-disk distribs
- gaussians params
- large scale: param maps

3 representations: near, mid, far

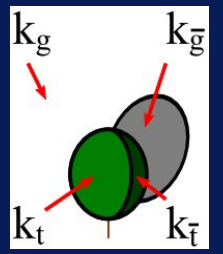


Near: ~Z-buffer IBR

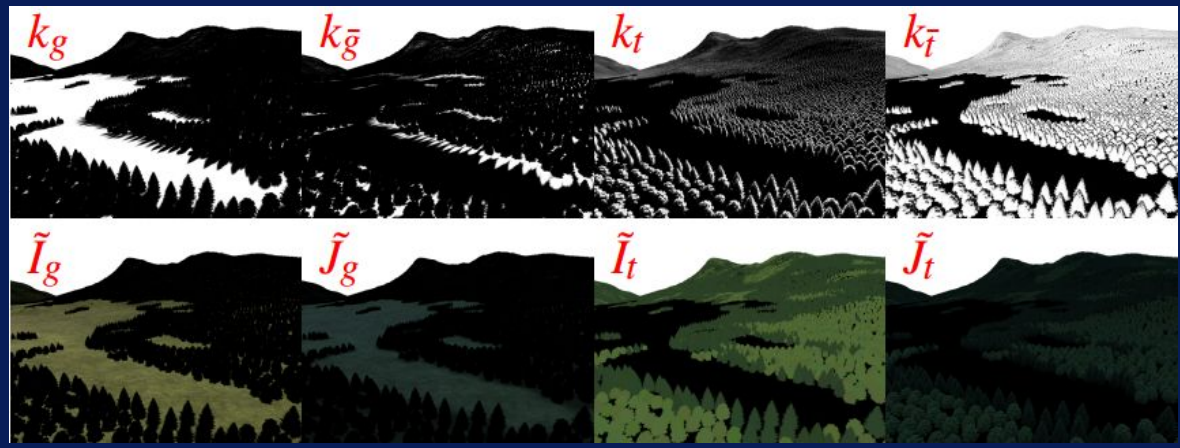


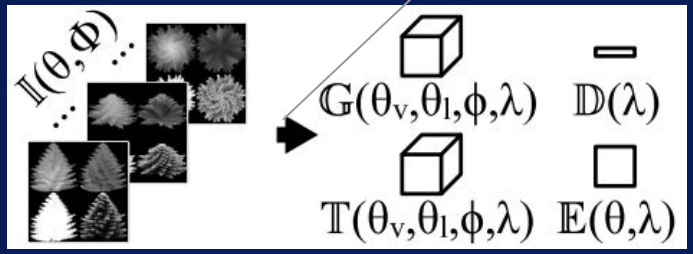
Zv,Zl → extinction

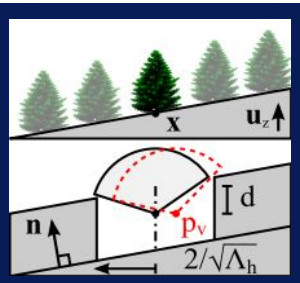
individuals with parallax
and detailled shading



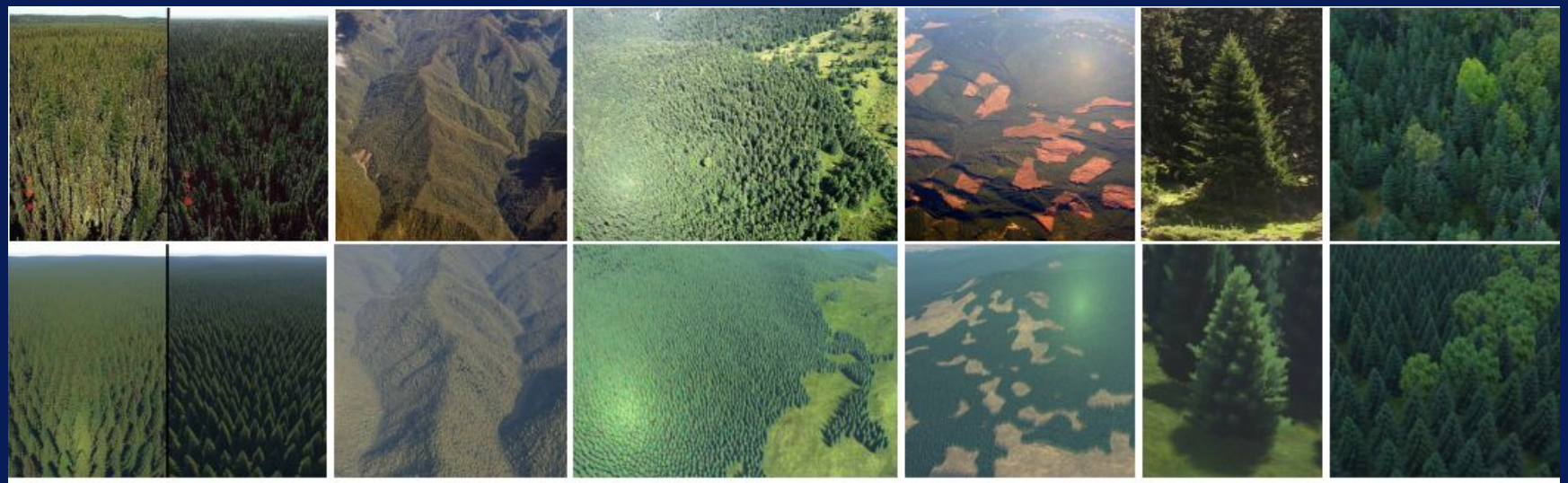
Mid & far: masks*shaders (~ Fake Fur Rendering)







EG'12: **Endless forest**, with Eric Bruneton



Comparisons between photos (top) *and our results* (bottom)

video
demo

video
full

2. A few things I learned

Representations

Many tools on store !

raster (e.g., Photoshop) = grid
vs vector (e.g. Illustrator) = shape

grids: image texture. Voxels. Eulerian simu. BRDF table. SH.
vectors: GL,ps,laser. Mesh. Lagrangian simu, filaments. Lobes.

- indeed, more continuous: amount of info:
compressed data, base decomp., compr.sensing, fit, procedural, analytic
size matter:
- 4D table is cheap if interpolated low-res
- fitting or SH is not cheap if 798 coefs + trancend. math op

- opposed pro- and con- :
- no universal one: choose the appropriate
→ can be mixed :
- can change with scale or interaction length (local / long dist)
- each box can use different one:
shape, colors, shadowing & light transport, anim (space def)

Representations

Ones from Physics & maths:

Eulerian vs Lagrangian

Space vs Fourier

Velocity vs vorticity

Point-mechanics vs Finite elements / SPH

Color spaces

Point mechanics / statistic mechanics / fluids / waves / spectrums
energy lines

Photons / waves / rays / energy

(don't forget validity domain & hypothesis)

Representations

Where to start:

- where is largest potential for improvment ?

ie, what worse part in the look / workflow ?

- best improvement reachable for each bit of extra budget ?

think "differentials everywhere" : pixel=circle, ocluder=slab, ray=spline.

= 1st order Taylor approx

better = F(P)+PX.grad(P), X in neighborhood. → integrate(f(Fb(P,X),X))

- what constraints ? preferences ?

time budget ? storage budget ? precision budget ? hard or sloppy ?

Have quality estim

→ faith → weigthing, transition to backup to canonical approach

Reminder: quality = worst box, not best

so long "perfect equation" if no accurate parameter available

→ forgot nothing ? Shannon-Nyquist ok ? Large Numbers ok ?

Differential everywhere !

= continuous integral everywhere

Points are not physical objects
differentials are. *dS, dl, dω, cones…* = *local integral*

differential domain ⇒ value=distrib.

→ *Distributions everywhere !*

Any scalar → distribution *(colors, mask …)*

Any vector → distribution *(velocity, pos, ...)*

*- minimal is **a lot** better than nothing*

- can be cheap to have & store: Gaussian stddev, lobe width

- can be cheap to use

make well-posed many ill-posed problems

e.g., aliasing and filtering issues

is a kind of LOD (subgrid model)

LOD everywhere !

Reminder: ***metrics = pixel color***

→ LOD is not "anything simpler"

LOD ~= pre-integration over the pixel

i.e., preparation of the colorfield pixel integral giving

→ compact magic atom renderable with 1 sample

Some LOD examples:

- CG: roughness. brdf, glossyness. surface.

NDF,MIPmap,texture. impostors,particles.

Physics:

- pseudoforces: buoyancy, coupling, ….
- pseudo objects: rays & optic geometry. Surfaces & solids
- emerging numbers: Temp, Pressure…. even Velocity…
(probably even space & time)

LOD everywhere !

LOD ~= pre-integration over the pixel

i.e., preparation of the colorfield pixel integral giving

Not so easy:

- *non-linearity* → *average(f(x))* ***is not*** *f(average(x)).* *same for interpolation*
- *correlations, non-separability* → $\int fg$ ***is not*** $\int f \int g$
- *a cascade of wrongness & clandestine hypothesis: MIPmapping*

→ *Reformulate:*

$$I = \frac{\int_{\mathcal{P}} L_i(x,\omega_i)\, C(x)\, \rho(n_x,\omega_o,\omega_i)\, V_o(x)\, V_i(x)\, w_P(x)\, dx}{\int_{\mathcal{P}} V_o(x)\, w_P(x)\, dx}$$

- *other physics or math handle*
- *distributions. Stat momentums.*
- *reparameterize: log, sqrt, ^2, 1/x, equivalent set (e.e., polar)*
- *change space: uv → uvw, or no uv*

LOD everywhere !

hierarchical:

- *scalewise divide and conquer*
- *don't forget upstream and downstream:*
  - *frequencies in data ? frequencies once rendered ?*

different scales might be totally different problems:

- *different purpose (scenario)*
- *different perception (river-way / flow / details)*
- *different knowledge*

→ *different controls*

→ *Choose best representation*

LOD everywhere !

hierarchical:

- *scalewise divide and conquer*
- *don't forget upstream and downstream:*
  - *frequencies in data ? frequencies once rendered ?*

different scales might be totally different problems:

- *different purpose (scenario)*
- *different perception (river-way / flow / details)*
- *different knowledge*

→ *different controls*

→ *Choose best representation*

Reproducing the Natural Complexity

Quality real-time rendering / animation is sometime reachable

- Choose the right representation

- Be smart rather than brute force

- Don’t get blinded by what you know
  → look through the window, Nature is right there ! :-)


Reproducing the Natural Complexity

heap

(extra discussion material)

Announcement

Wednesday,18 Decembrer 2013 - Weta Digital Record Press Conf, Wellington, Miramar

Creating, simulating, exploring, rendering the tremendous huge and detailed complexity of natural scenes

A quick tour through scalable modelling, smoke animation, cloud rendering, appearance LOD, animated textures, landscape-size rivers, forests, and ocean, even all-scales galaxy, as models aiming at complexity and realism in real-time I worked on with my PhD students and collaborators (which includes Sylvain Lefebvre, Antoine Bouthors, Alexis Angelidis, Cyril Crassin, Eric Heitz, Erc Bruneton you might know ;-)).

*In a second part, I will try to re-settle a few things through **Big Questions of the CG Universe** like "*what is a 'surface' ? a volume ? a LOD ? a sample ?*" as a way to fight the dragon of scenes complexity with better swords.*

Fabrice NEYRET, senior researcher at CNRS/INRIA/Grenoble University, FRANCE
http://evasion.imag.fr/~Fabrice.Neyret/

Date**: WEDNESDAY 18 DECEMBER 12 NOON**
Where: **Record Press Conf**

Title: **"Creating, simulating, exploring, rendering the tremendous huge and detailed complexity of natural scenes"**
Speaker: Fabrice Neyret, senior researcher at CNRS/INRIA/Grenoble University, France
This chalk talk will be recorded

Chalk Talks are informal weekly presentations of topics throughout the visual effects production pipeline.
Experts from all departments on any and all subjects are invited to give a talk.
If you are interested please email: jgillespie@wetafx.co.nz

All the notes from our previous Chalk Talks are on our twiki,
http://twiki.wetafx.co.nz/RnD/ChalkTalks

Undeveloped (so many slides, so little time...)

- ***Philosophical key questions***

  - What is an LOD ? (metrics: screen, pixels)
  - What is a volume ? a surface ?
  - What is a normal ? a transparency ?
  - What is a sample ? a texture ?

- ***Sampled scales along graphics pipeline → aliasing & bias***

  maths (integration calculus, signal processing)

	texture	***render***	***geom***	***anim***
geometry/material/brdf	*fetch*	*pixel/intersect*	*vertex/polygon*	*vertex/voxel*
sample span	*footprint/kernel*	*kernel(Srate,DoF)*	*surf(kernel)*	*mesh/vol(kernel), dt*
multiscaling subgrid	*LODfetch(aniso)*	*fragment(Abuff)*	*brdf*	*subgrid, motion blur*
interpolation mag,min	*mag,min*	*mag,LOD*	*subdiv,decim*	*mag, more blur*
aliasing/oversmoothing	*Moire,noise*	*jaggies,noise*	*peak,jaggies*	*flick,pop,backturn*
		(col,spec,shadow)	*(shape,sillh,shadows,+render)*	
poor: (beside aliasing)	*color change*	*shading change*	*silh,small feat.*	*ghosting,polymove*
filtering (pre-integration)				
'filtering' means:	*lod+aniso*	*mutisampling*	*micropolygon*	*sampled blur*
Shannon-Nyquist obeing:	*no/poor filter*	*sampling anything*	*displ(mdl/rend)*	*t-sampling anything*
	op after filter	*having screen hifreq*		

About “physical models” (in CG tongue)

« ‘physical approach’, ‘exact’, ‘rigorous’»

- There is no such thing like «exact» in physics
- «Physical» ≠ local (equa-diff)
- Local eqn vs macroscopic, «rigorous vs empirical»: subjective !
  - mecaQ → molecules → stat phys → thermodyn → → NS → hydraulics/waves/atmo(oceano)sc
  - mecaQ → EM field → Huygens → geom optic → → RT/radios/visibility
- Hypothesis, conditions, limits of validity

  ex, continuous fluids: notion of P,T, V, parcel (emergence)
- Border conditions, parameters

  one half of the problem is not or poorly known !
- continuous eqn → numerical engineering: resol issues
  - subgrid models: on-going research
  - sous-res → erreurs qualitatives et quantitatives [SAA00]
- **Outil, inspiration. Mais pas sacraliser. contextualiser!**

What does users want ?

- **Graphist:**
  - Super-spectator
  - Scenario
  - Expressive tools: not black box !
    - Usable
    - Controlable
    - Intuitive & predictables parameteres
    - Generative space rich / useful enough
    - Feedback (→ fast is useful even for SFX)
    - For on scene, on shot.
      → All tools are on shell + full manual

Studying real world

Physics eqn vs the real Nature

- Structured vs ‘blurry’, known vs dirt & fluctuations
  Artificiql symetries,regularities,rigidities change the phenomenon (buckling, natural convection, silhouette brdf)
- Clandestine hypothesis (Evil !)
- LC: borders, such a mysterious thing !
  (meso-shape, param value) e.g. “river bed”, “bark”
- Useless details vs uncontrolled emerging phenomena
- Simu: résultat change avec résol [PDI-LF02]

A.Fournier: ***start from real images, end with real images***
(inspiration, validation)

- Observe. picture. film. touch. draw. Repeat.
- Learn how to see. Find the ‘meaning’ (the ‘structure’. of things & eye)
- Pb of subjective validation

Alternate representations

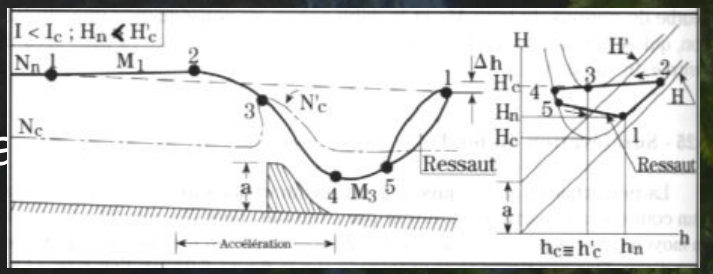
- Scales: (≠ meaning, perception, goal, data, simu)
  → coupling different models
- Formes, surfaces: subjectives notions !
- **How to representer the world ?**
  - What we know / what we see (shape, relief…)
  - Minimalist, impressionnist approaches
    separate shape/relief, normals, shading
    Adaptive: hierarchy of modeles [Kaj85]
  - Repr. of shapes: meshs, surfels, voxels…
    Properties ≠ : structuration, cost, filtering…
  - Decoupling (geom / texture space / light space / ...)

Phenomenological simulation

- Large & detailed: physical simuout of reach. + [PDI-LF02]
- Some a priori knowledge usually exists !
  - values ranges, modes, dominant pheno…
  - at least: what the purpose is, what the scene is
- **Emerging effects**: instabil., waves, folds, equilibrium…
  - Equations: indirect, phys++. While predictable
  - Closer too meaning, macroscopic, intuition, user langage
- **Direct repr of emerging phenomenons**

  Macroscopic phys (phenomenological / empirical / analytical)
    - Availablemodels / analytical / direct obs. / obs. ref simu

  Macroscopic primitive
  - XVIIIth - XXth treasures
  - revisit, make yours, invent, generalize…
  - uneasy, sparsly explored…but results might pa



Settling a problem

- Purpose
  (what are we aiming at ? why ?)
  Idem que but finaliste (appli) ou constructif (outils fondam.)
- Formalize data/knowledge
- Formalize hypothesis (raisonnées),
  Objectifs (list of requirements),
  Criterions
- Proposal
  - What already exist ? what to draw on, what's inadapted and why ?
  - Your way (explicit and justified choices)
    goals →sub-goals →details (c/ code review!)
  - Validation, **+** & **-**, perfs, limitations, comparaisons

Texture filtering (interp & MIP-map)

- **Clandestines hypothesis:**
  - Linearity 1: N, courb., visibility, shadows, const params.
  - Linearity 2: fragment = lin(texture) , i.e.: text = RGBA
  - Continuity: neglect borders, holes, atlases, tiling

Texture filtering (interp &

- **Clandestines hypothesis:**
  - Linearity 1: N, courb., visibility, shadows, const params.
    ➢ pb: micro-geomerty ! Ultimate filtering!
  - Linearity 2: fragment = lin(texture) , i.e.: text = RGBA
    ➢ pb: textures for anything (Z,N,…) !
  - Continuity: neglect borders, holes, atlases, tiling
    ➢ pb: indirections !

- **Geometry filtering:**
  - Polygons not antialiased
  - Get smaller and smaller
  - Not pre-filterable

  → repr alt, model transition [Kaj85]