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
Reproducing the *Natural Complexity*

*ultra-detailed + ultra large shape + animation + rendering*

**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
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**
Why I’m doing all that?

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

- **Science deep lover** (understanding. popularization)
  **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)
  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.)
Why I’m doing all that?

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

- **Science deep lover** (understanding, popularization)
  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 )
  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 + controllable
  - 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: founded
1987: Explore image synthesis software
1989: split software / production (→ ExMachina)
1993: SGI > Alias > wavefront > TDI (MS > softimage)

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

Rendering fur with three dimensional textures
Kajiya & Kay
Siggraph’89
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)
Volumetric Textures

Rendering fur with three dimensional textures
Kajiya & Kay

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

→ PhD topic found! :-)

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

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

Volume:
- SVO
- multiscale
  → octree of voxels

Voxel data:
- viewdep density
- reflectance

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 ($\Sigma$ trochoïds)

\[
\begin{align*}
    x - x_0 &= Ae^{kz_0} \sin(\omega t - kx_0) \\
    z - z_0 &= Ae^{kz_0} \cos(\omega t - kx_0)
\end{align*}
\]

dispersion eqn

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

amplitude spectrum (PM)

\[
F_{PM}(f) = \frac{ag^2}{(2\pi)^4 F_0 e^{\frac{5}{4} \left(\frac{f_m}{f}\right)^4}}
\]

\[
F(f, \alpha) = F_{PM}(f) D(f, \alpha)
\]
Textures: 03-05, with Sylvain Lefebvre

- Distortion-free pattern mapping
  - uv-mapping ill-posed → get rid of global param

- Texture memory management
  - huge detailed 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

A C++/OpenGL library for the real-time realistic rendering of very large and detailed 3D natural scenes on GPU
08→12: PROLAND, with Eric Bruneton

- whole Earth, all scales
- out of core

Real-time Realistic Rendering and Lighting of Forests
Bruneton Eric, Neyret Fabrice

Real-time Realistic Ocean Lighting using Seamless Transitions from Geometry to BRDF
Bruneton Eric, Neyret Fabrice, Holzschuch Nicolas

Scalable Real-Time Animation of Rivers
Yu Olzhi, Neyret Fabrice, Bruneton Eric, Holzschuch Nicolas

Precomputed Atmospheric Scattering
Bruneton Eric, Neyret Fabrice

Real-time rendering and editing of vector-based terrains
Bruneton Eric, Neyret Fabrice
SCA’02, EG’10: **Real-time ocean**, with Eric Bruneton

simulate all waves (FFT)

waves eqn \((\Sigma \text{trochoïds})\)

\[
\begin{align*}
x - x_0 &= A e^{kz_0} \sin(\omega t - kx_0) \\
z - z_0 &= A e^{kz_0} \cos(\omega t - kx_0)
\end{align*}
\]

Appearance filtering: shape→\(N\)→BRDF

Camera

Screen space

Object space

\(w_p = w(N_{\min}, N_{\max}, \lambda/L)\)

\(w_n = w(N_{\min}, N_{\max}, \lambda/L)\)

\(w_r = 1 - w_n\)

frequency waves spectrum

frequency

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

Model courtesy of 3D-Coast/Rick Sansia

Youtube C.Crassin

www
GigaVoxels I3D’09, EG’11 → , with Cyril Crassin

**full volumetric scene, real-time**

- SVO
- multiscale
- out of core
→ octree of bricks

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

**Voxel data:**

- sort of geometry prefiltering
→ ~true shape LOD
Appearance Filtering

sort of geometry prefiltering $\rightarrow$ true local shape LOD

issues with volumes of density:

- no viewdep $\rightarrow$ 6 dir Glvoxels
- fixed dir $\rightarrow$ only ok for (some) buildings

- no correlation:
  - fat silhouette
  - some light leaking

$\rightarrow$ density/extinction = bad occupancy estimator for rendering purpose.
opacity/transparency also.

- not view-dep $\rightarrow$ distrib
- not interpolate right either for xy or z: \( \text{vis(mean(.))} \neq \text{mean(vis(.))} \)
- should interpolate differently in xy vs z!

$\rightarrow$ need volumes of something else.
Appearance Filtering, with Eric Heitz

- Small scale relief + visibility
  → all is view-dep and light-dep!

- Correlations everywhere!
  → light and colors
  → normals
  → visibility
  → occlusion

  → + content correlation
  → missing in all bumps
  → microfacet models
1: differential cone tracing

→ select continuous LOD

2: 3D A-buffer

→ solve z-correlation

3: Voxel data = distrib

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

\[
\text{distrib} = N(m, \sigma^2) \rightarrow \text{solve light- and view- dep}
\]

→ + some content correlation

→ Cook-Torrance shading

macrosurface = SDF
assumed locally flat
details = height field
content correlation

ex: color-height, color-orientation

⇒ pixel integral not separable
BTW, even color(texture) is an issue
→ also all f(noise): LUT, clamp, abs...

since
average(LUT(text.)) # LUT(average(text.))

Idea: use color distrib

1: average = LUT * histogram

2: histogram ~ gaussian

3: simply precompute iLUT(v,σ)

NB: applies to any distrib e.g., heights ...
Appearance Filtering, with Eric Heitz

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

\[
\tilde{C}_h = \left< D_h, h \right>
\]

slope (normal) distribution

\[
\int_P L_i(x, \omega_i)C(x)\rho(n_x, \omega_o, \omega_i)V_o(x)V_i(x)w_p(x) \, dx
\]

\[
\int_P V_o(x)w_p(x) \, dx
\]

... but:

4: effect = lobe tilting

\[\rightarrow \text{easy!}\]

5: NB: for diffuse surface, effect of envmap = irradiance_map(N)

\[\rightarrow \text{cf colormap(slope)}\]
**Appearance Filtering**, with Eric Heitz

\[ I = \int_P \! L_i(x, \omega_i) C(x, \rho(n_0, \omega_0, \omega_i)) V_o(x) V_i(x) w_P(x) \, dx \]
\[ \int_P V_o(x) w_P(x) \, dx \]

**LEAD-R**: displacement -> microfacets, with:
- tilted lobe
- anisotropic
- true masking is Smith, not Cook-Torrance
- point light + IBL

**Effect of geometric distortion on appearance:**

**displacement map** (mipmapped)  
**LEADR textures** (mipmapped)

**geometry prefiltering** $\rightarrow$ ~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: \[ \int \text{Droplet Size Distrib} \]

→ cancels Bessel oscillations

2: \[ n_{\text{scatter}} > 1 \]

→ peak (50% E) ~ no hit
- high freq useless
- no colored back-scatter

Youtbe
Realistic clouds in real time, with Antoine Bouthors

1. simulating all light paths: hard pb.

2. $\rightarrow$ real time!

3. separate scattering orders $\rightarrow$ shift Most Probable Path and scale of transport

4. macro-material $\psi(i,o)$ $\rightarrow$ collector(pos,\(\sigma\))

5. ground $\leftrightarrow$ cloud radiosity, sky illu

$\rightarrow$ solve $i=\text{collect}(o)$ for cloud shape

$10^7$ simu. fit

$L,V,V_{\text{pos}},\text{thick.}_{5D}$

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

- Mechanical textures: folds and cracks
**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)
Ad vect ed 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
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, asynchronous, unbounded)

- Poisson-disk particles
- deformable sprites
- renew too deformed particles
- boundary conditions
- 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, controllable…
- closer to std CG workflow

Local space rotation operator

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

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

\[ \Gamma = \int_L \mathbf{v} \cdot d\mathbf{l} = \iint_S \mathbf{\omega} \cdot d\mathbf{S} \]

+ vtx noise
+ ellipsoid particles
a few other trampings

more forestry
(and fancy representations)
1: (small) $BTF(L,V)$ x 3 LOD

2: Hierarchical visibility (small) cubemaps
in Proland → all scales, real-time, seamless LOD

realistical: sun+sky, silverlinings & transparencies, all-scales correlations (hot spot) + shadowing (ambient occlusion)
EG’12: **Endless forest**, with Eric Bruneton

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

Near: ~Z-buffer IBR

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

3 representations: near, mid, far

Near: ~Z-buffer IBR

Mid & far: masks*shaders (~ Fake Fur Rendering)
EG’12: *Endless forest*, with Eric Bruneton

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

- 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 + transcend. 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 improvement?
  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 → weighting, 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.**  
$dS$, $dl$, $d\omega$, cones…  = local integral

differential domain $\Rightarrow$ 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 \(\approx\) 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** → \( \text{average}(f(x)) \text{ is not } f(\text{average}(x)) \). same for interpolation

- **correlations, non-separability** → \( \int fg \text{ is not } \int f \int g \)

- a cascade of wrongness & clandestine hypothesis: MIPmapping

→ Reformulate:

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

<table>
<thead>
<tr>
<th>geometry/material/brdf</th>
<th>texture</th>
<th>render</th>
<th>geom</th>
<th>anim</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample span</td>
<td>fetch</td>
<td>pixel/intersect</td>
<td>vertex/polygon</td>
<td>vertex/voxel</td>
</tr>
<tr>
<td>multiscaling subgrid</td>
<td>footprint/kernel</td>
<td>kernel(Srate,DoF)</td>
<td>surf(kernel)mesh/vol(kernel), dt</td>
<td></td>
</tr>
<tr>
<td>interpolation mag,min</td>
<td>LODfetch(aniso)</td>
<td>fragment(Abuff)</td>
<td>brdf</td>
<td></td>
</tr>
<tr>
<td>aliasing/oversmoothing Moire,noise</td>
<td>mag,min</td>
<td>mag,LOD</td>
<td>subdiv,decim</td>
<td>mag, more blur</td>
</tr>
<tr>
<td>poor: (beside aliasing)</td>
<td>color change</td>
<td>shading change</td>
<td>peak,jaggies</td>
<td>flick, pop, backturn</td>
</tr>
<tr>
<td>filtering (pre-integration)</td>
<td>lod+aniso</td>
<td>multisampling</td>
<td>micropolygon</td>
<td>sampled blur</td>
</tr>
<tr>
<td>‘filtering’ means:</td>
<td>no/poor filter</td>
<td>sampling anything</td>
<td>dis/pl(model/rend)</td>
<td>t-sampling anything</td>
</tr>
<tr>
<td>Shannon-Nyquist obeing:</td>
<td>op after filter</td>
<td>having screen hifreq</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
  Artificial symmetries, 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 models [Kaj85]
  – **Repr. of shapes:** meshes, surfels, voxels…
    Properties ≠: structuration, cost, filtering…
  – **Decoupling** (geom / texture space / light space / …)
Phenomenological simulation

• Large & detailed: physical simulation of reach. + [PDI-LF02]

• Some a priori knowledge usually exists!
  – values ranges, modes, dominant phenomena...
  – at least: what the purpose is, what the scene is

• **Emerging effects**: instability, waves, folds, equilibrium...
  – Equations: indirect, phys++. While predictable
  – Closer to meaning, macroscopic, intuition, user language

• **Direct repr of emerging phenomena**

  Macroscopic phys (phenomenological / empirical / analytical)
  • Available models / analytical / direct obs. / obs. ref simu

  Macroscopic primitive
  – XVIIIth - XXth treasures
  – revisit, make yours, invent, generalize...
  – uneasy, sparsely explored...but results might pay.
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-geometry ! 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]