A Multiple-Scale Stochastic Modelling Primitive

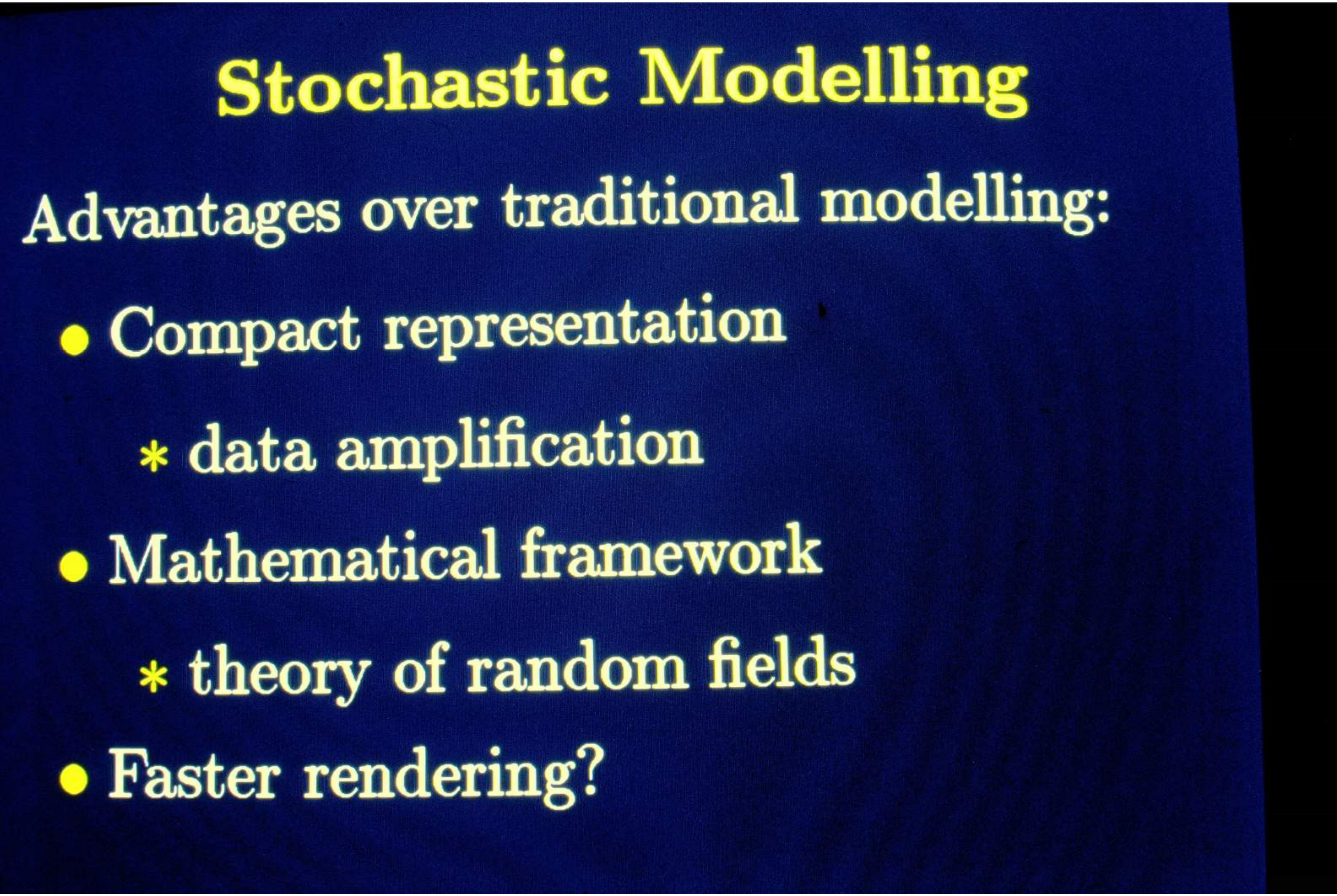
Jos Stam
Eugene Fiume

Department of Computer Science
University of Toronto
Toronto, Canada

Modelling Natural Phenomena

We want:

- Photo-realism for
  - Visual simulation
- User control over shape for
  - Design
  - Animation



Stochastic Modelling

Advantages over traditional modelling:

- Compact representation
  - data amplification
- Mathematical framework
  - theory of random fields
- Faster rendering?

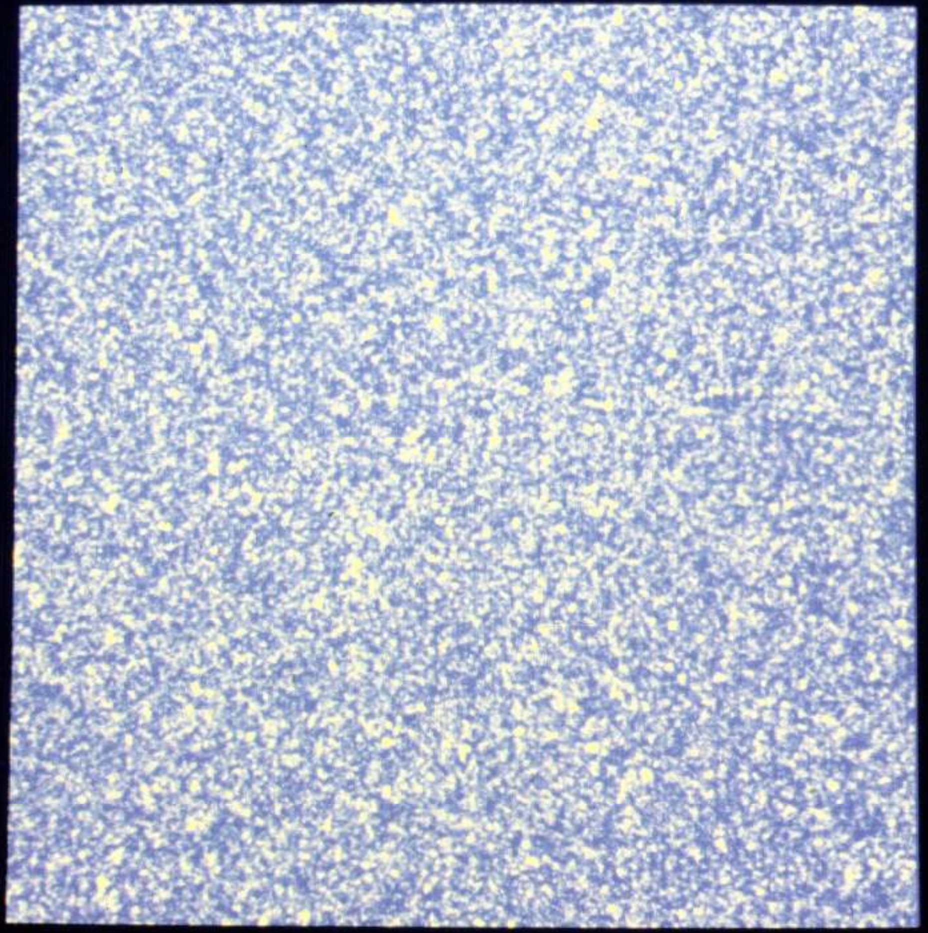
Random Fields

Model phenomenon as a function $R(\mathbf{t})$
R takes a random value at each location $\mathbf{t}$

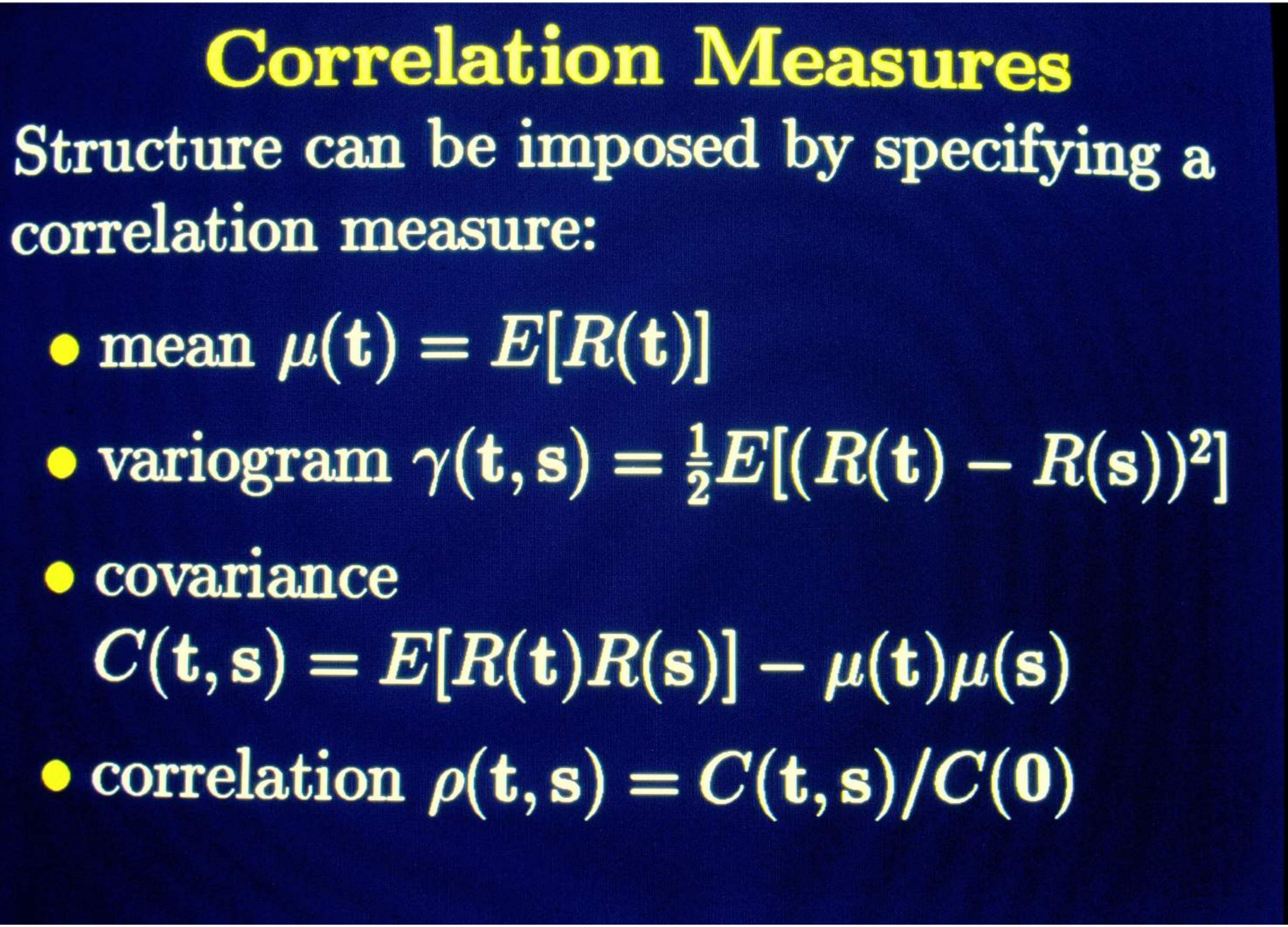
For clouds: $R(\mathbf{t})$ is the density, with

- $R(\mathbf{t}) = 1$ meaning total opacity
- $0 < R(\mathbf{t}) < 1$ "interesting" region
- $R(\mathbf{t}) = 0$ meaning total translucency

White Noise



Clearly too unstructured!



Correlation Measures

Structure can be imposed by specifying a correlation measure:

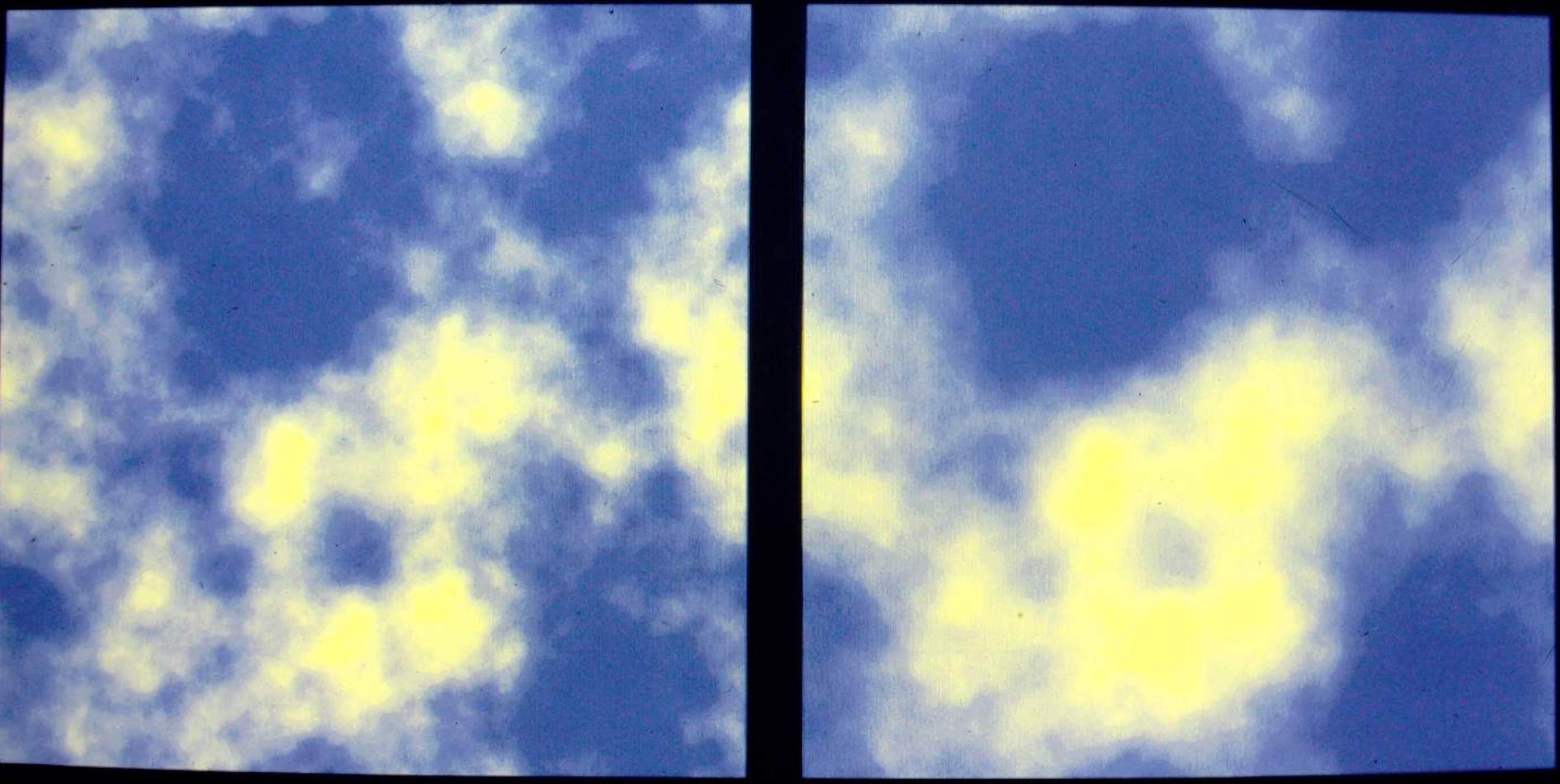
- mean $\mu(\mathbf{t}) = E[R(\mathbf{t})]$
- variogram $\gamma(\mathbf{t},\mathbf{s}) = \frac{1}{2}E[(R(\mathbf{t}) - R(\mathbf{s}))^2]$
- covariance
  $C(\mathbf{t},\mathbf{s}) = E[R(\mathbf{t})R(\mathbf{s})] - \mu(\mathbf{t})\mu(\mathbf{s})$
- correlation $\rho(\mathbf{t},\mathbf{s}) = C(\mathbf{t},\mathbf{s})/C(\mathbf{0})$

Simplifications

- Homogeneity : $\mu(\mathbf{t}) = \mu_0$ and $C(\mathbf{t}, \mathbf{s}) = C(\mathbf{t} - \mathbf{s}) = C(\mathbf{h})$
- Isotropy : $C(\mathbf{h}) = C(\|\mathbf{h}\|) = C(\tau)$
- Quasi-isotropy : $C(\mathbf{h}) = C(\mathbf{hQh}^t)$ where $\mathbf{Q}$ defines an ellipsoid

Gaussian Correlation

$$\rho(\mathbf{t}, \mathbf{s}) = \exp(-\alpha \|\mathbf{t} - \mathbf{s}\|^2)$$



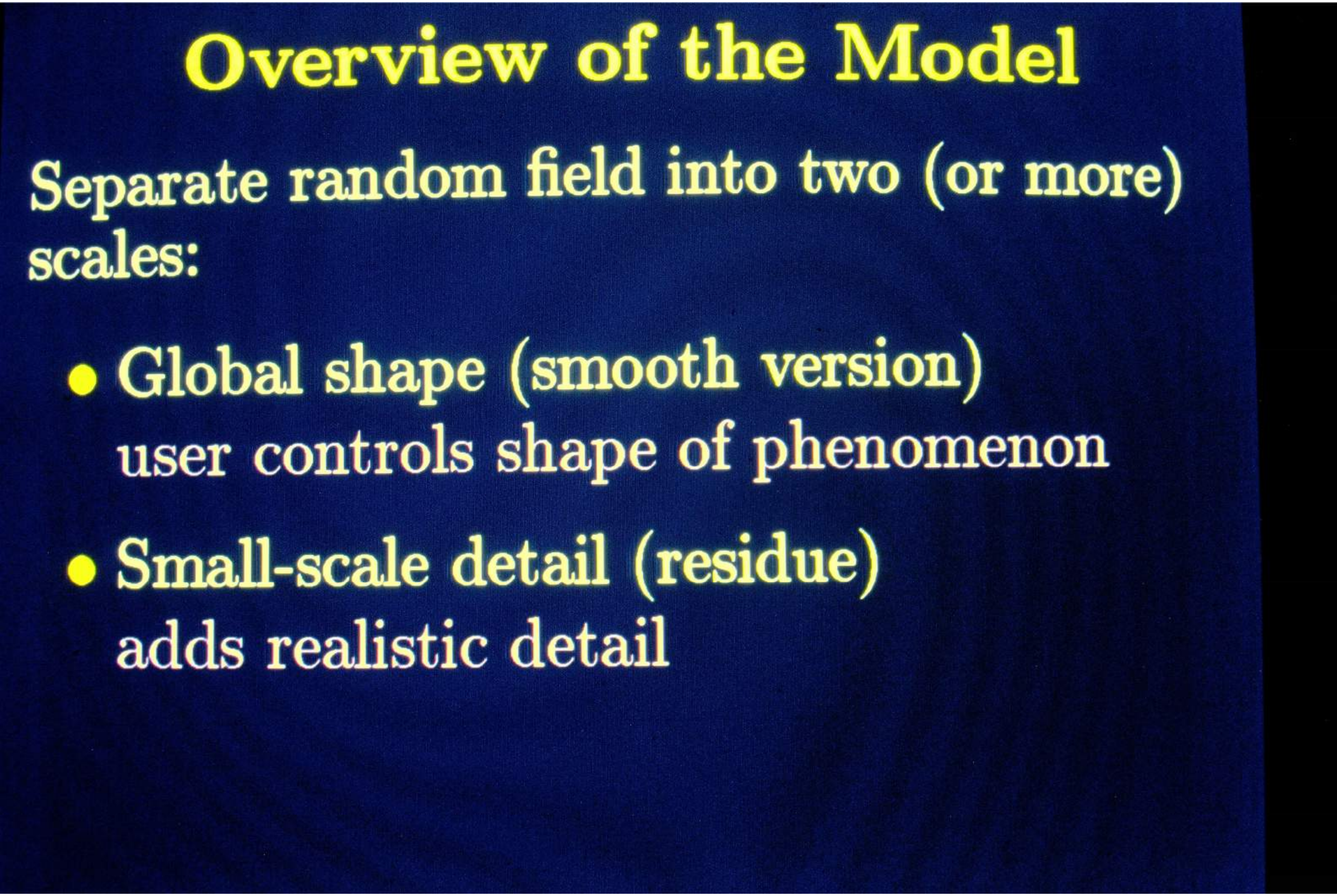
$\alpha = 2.0$ $\alpha = 0.5$

Previous Work

For 3D phenomena previous approaches have following drawbacks:

- high storage costs
- little control over shape
- high rendering computation costs

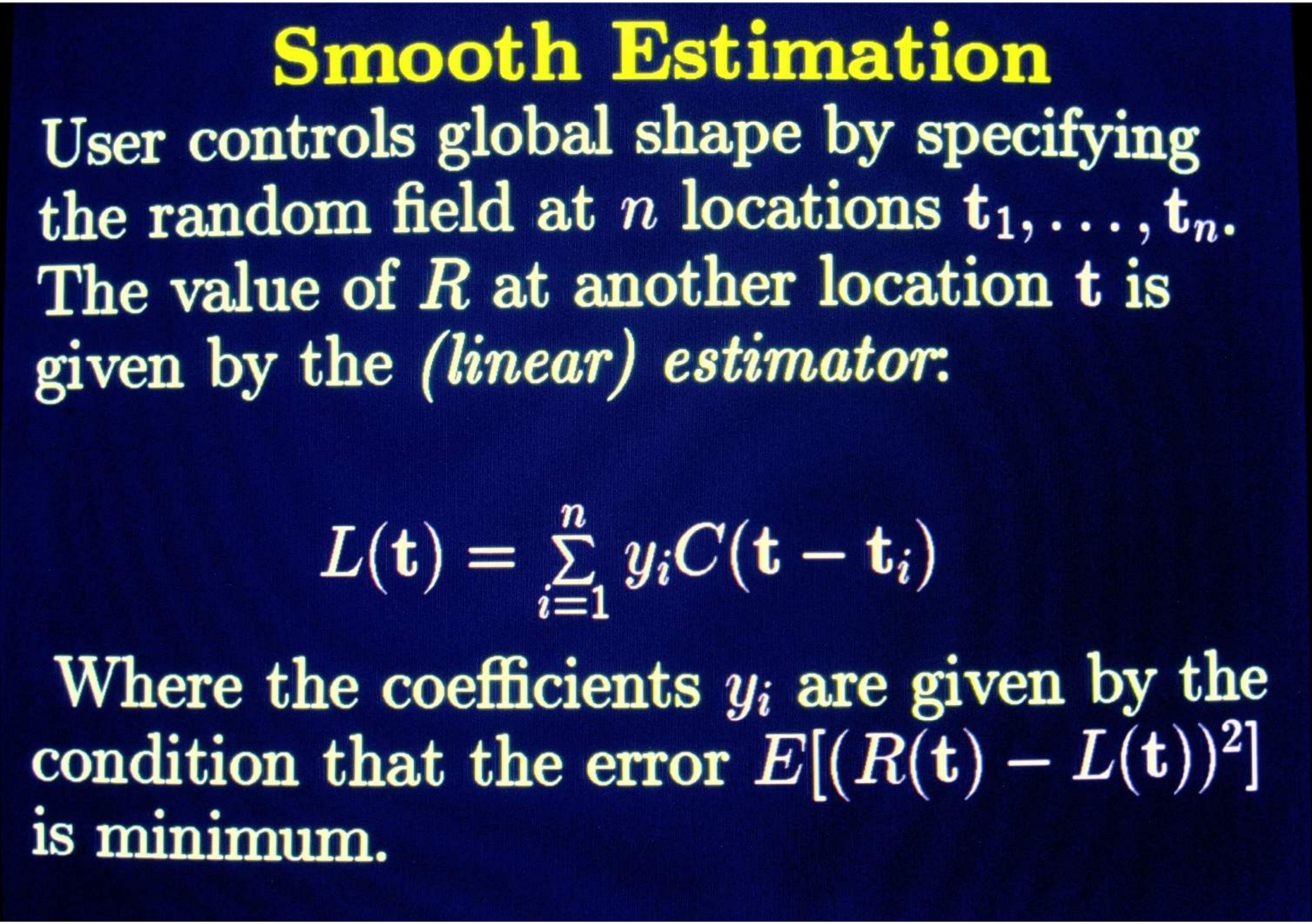
- Spectral synthesis (Voss 85)
- Stochastic displacement (Fournier et al. 82)
- Constrained fractals (Szeliski et al. 89)
- Generalized stochastic subdivision (Lewis 87)
- Textured ellipsoids (Gardner 85)
- Thick textures (Kajiya and Perlin et al. 89)



Overview of the Model

Separate random field into two (or more) scales:

- Global shape (smooth version)
  user controls shape of phenomenon
- Small-scale detail (residue)
  adds realistic detail

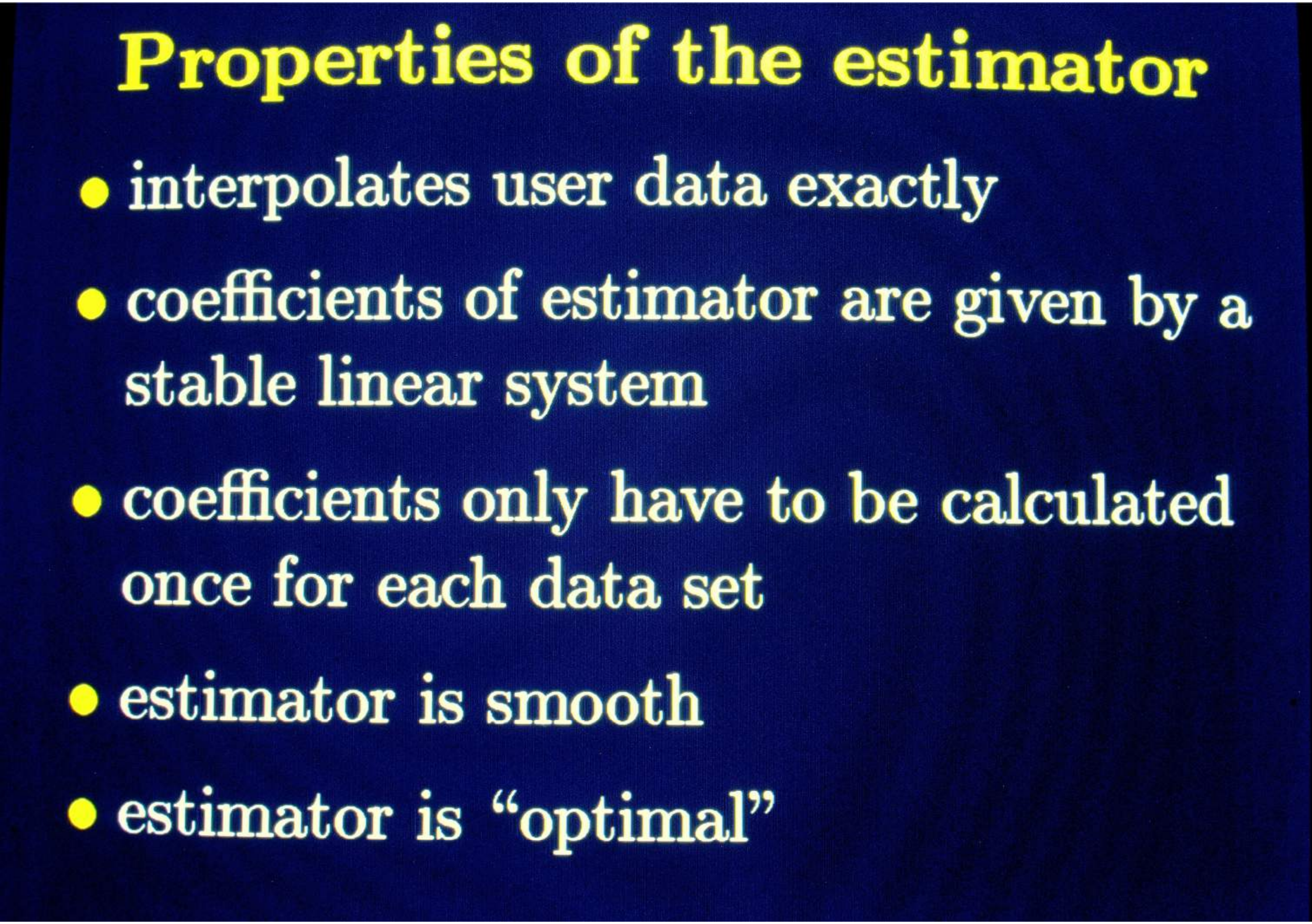


Smooth Estimation

User controls global shape by specifying the random field at n locations $\mathbf{t}_1, \ldots, \mathbf{t}_n$. The value of R at another location $\mathbf{t}$ is given by the *(linear) estimator*:

$$L(\mathbf{t}) = \sum_{i=1}^{n} y_i C(\mathbf{t} - \mathbf{t}_i)$$

Where the coefficients y_i are given by the condition that the error $E[(R(\mathbf{t}) - L(\mathbf{t}))^2]$ is minimum.



Properties of the estimator

- interpolates user data exactly
- coefficients of estimator are given by a stable linear system
- coefficients only have to be calculated once for each data set
- estimator is smooth
- estimator is “optimal”

Small-scale Detail

Use “simple” random functions to model small-scale detail.

- given by a small number of coefficients
- evaluations of the function are independent
- control over structure

Examples of Functions

- Perlin's noise: "fractal-like"
- Weierstrass-Mandelbrot function: control over fractal dimension
- Sparse Convolution (Lewis): control over spectrum

Rendering of Clouds

We want:

- to use geometry of global shape which gives semi-global illumination effects
- to avoid volume-rendering techniques because they are too expensive
- easy incorporation into a standard rendering package

Ray Tracing Algorithm

For each ray:

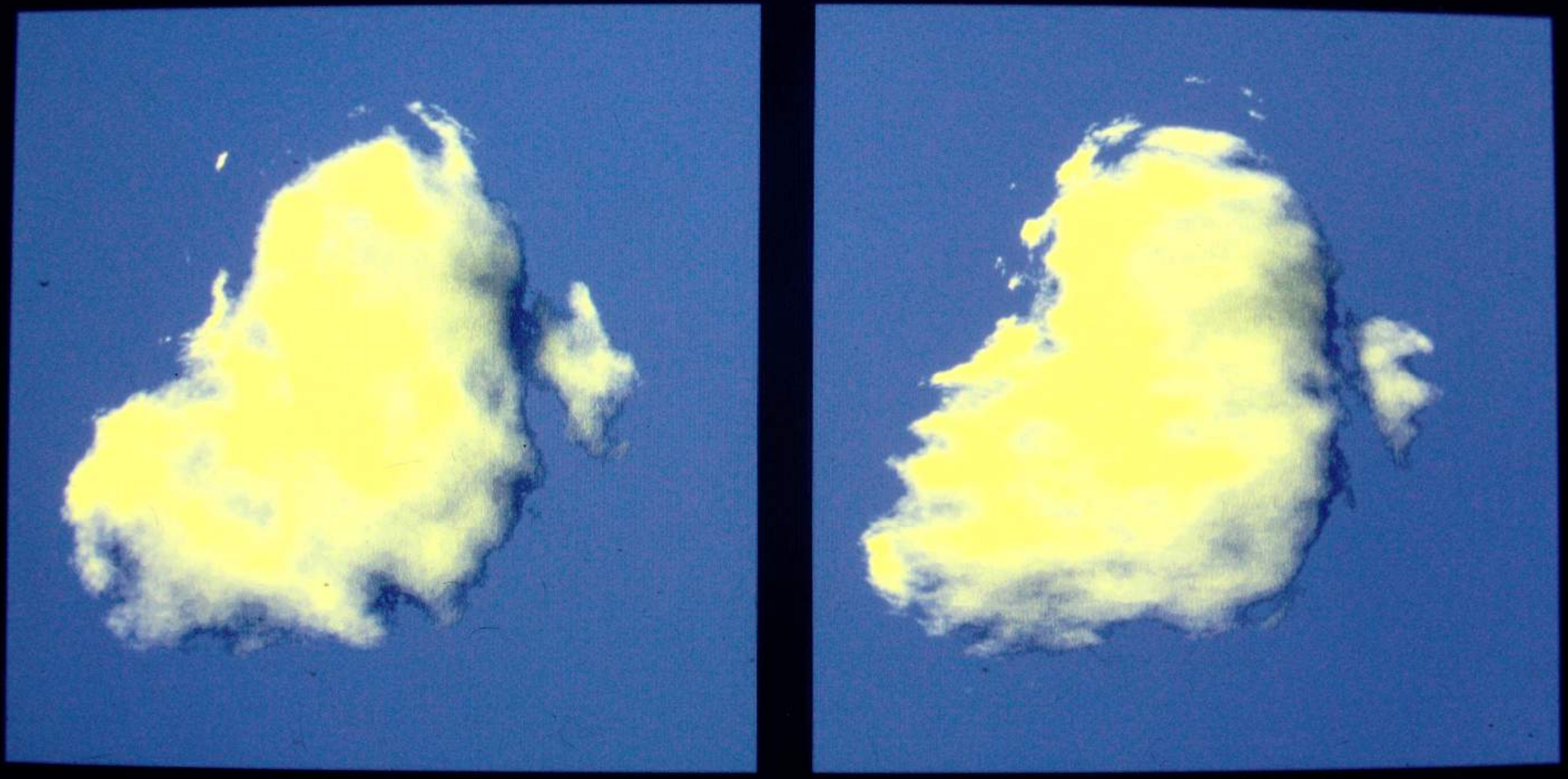
- find intersection(s) with global shape
- if none, next ray
- calculate brightness and self-shadowing using geometry of global shape
- use small-scale noise to calculate translucency and perturbe brightness
- if translucency < 1, continue to trace

Implementation

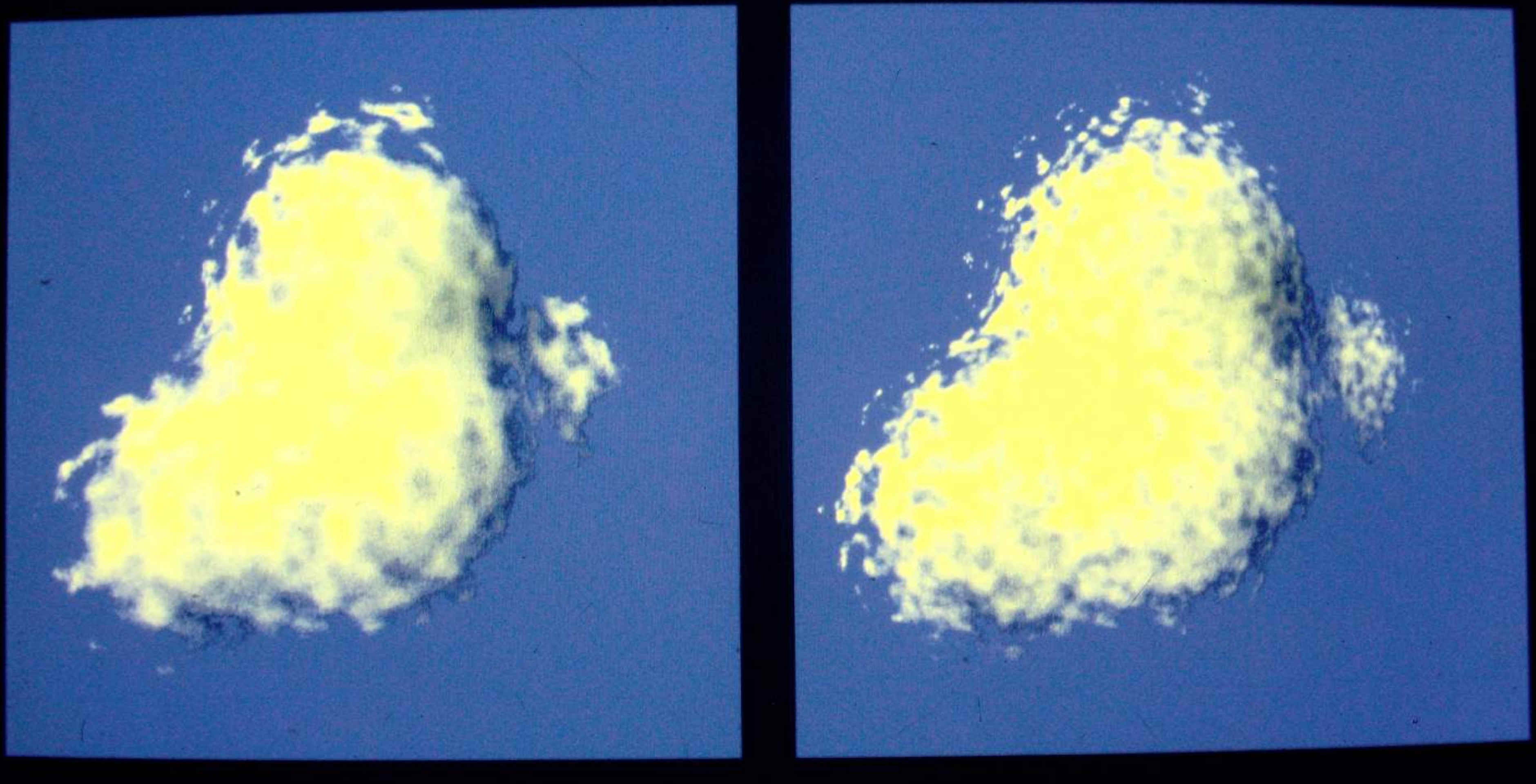
- Intersection with global shape calculated using interval arithmetic (Mitchell 90)
- Small-scale detail is added in a manner similar to (Gardner 85)

Isotropic Global Shape

Addition of Small-scale



Addition of Small-scale



Quasi-Isotropic Global Shape

Future Work

- Improve rendering: try more physically based approaches
- Apply model to other 3D phenomena
- Apply model to 4D phenomena, where temporal correlation gives frame to frame coherence