Procedural Modeling

Thomas Funkhouser
Princeton University

Procedural Modeling

Goal:

- Describe 3D models algorithmically

Best for models resulting from ...

- Repeating processes
- Self-similar processes
- Random processes

Advantages:

- Automatic generation
- Concise representation
- Parameterized classes of models

Procedural Modeling

Sweeps

Fractals

Grammars

Probabilistic models

Probabilistic grammars

Procedural Modeling

Sweeps ←

Fractals

Grammars

Probabilistic models

Probabilistic grammars

Example: Seashells

Create 3D polygonal surface models of seashells

"Modeling Seashells,"
Deborah Fowler, Hans Meinhardt,
and Przemyslaw Prusinkiewicz,
Computer Graphics (SIGGRAPH 92),
Chicago, Illinois, July, 1992, p 379-387.

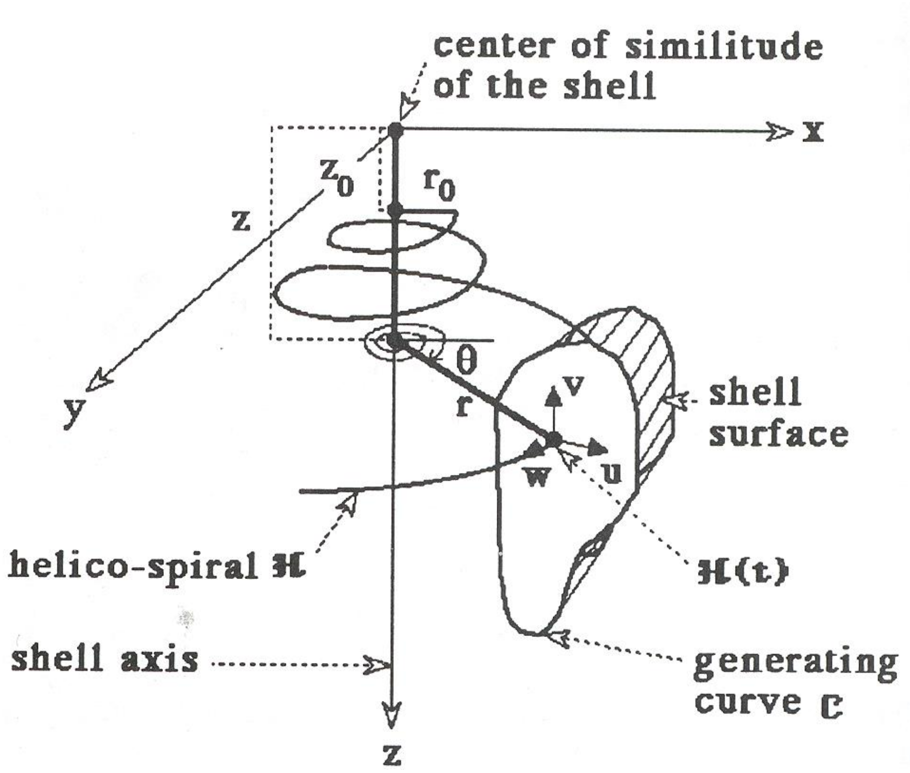
Fowler et al. Figure 7

Example: Seashells

Sweep generating curve around helico-spiral axis

Helico-spiral definition:

$$\Theta_{i+1} = \Theta_i + \Delta\Theta$$
$$r_{i+1} = r_i \lambda_r$$
$$z_{i+1} = z_i \lambda_z$$



Fowler et al. Figure 1

Example: Seashells

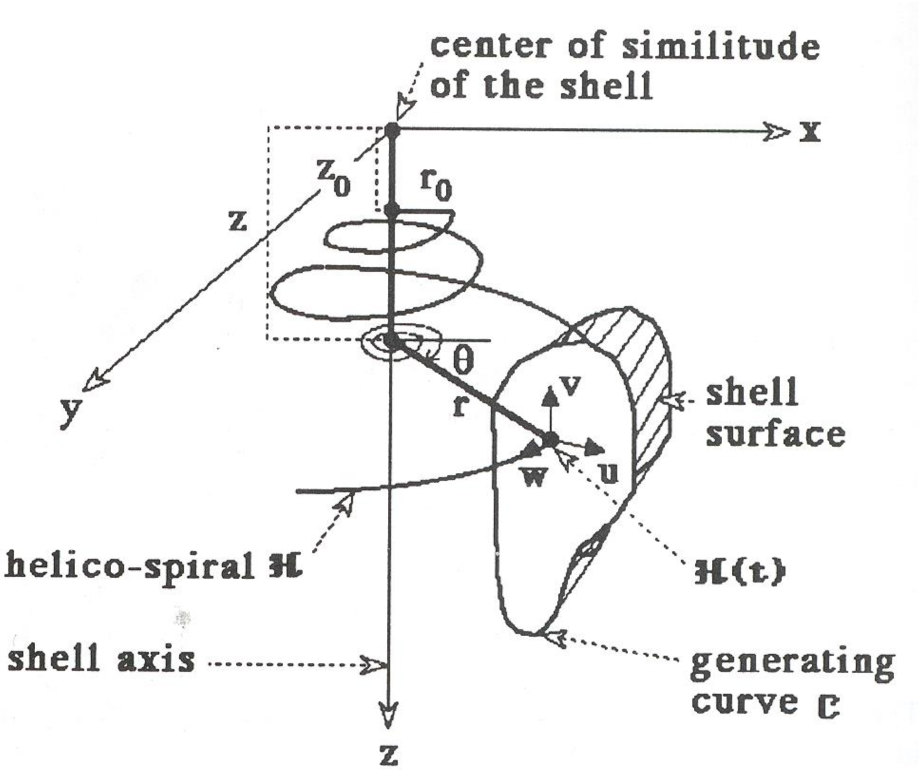
Connect adjacent points to form polygonal mesh

Fowler et al. Figure 6

Example: Seashells

Model is parameterized:

- Helico-spiral: z_0, λ_z, r_0, λ_r, N_θ, $\Delta\theta$
- Generating curve: shape, N_c, λ_c



Fowler et al. Figure 1

Example: Seashells

Generate different shells by varying parameters

Different helico-spirals

Fowler et al. Figure 2

Example: Seashells

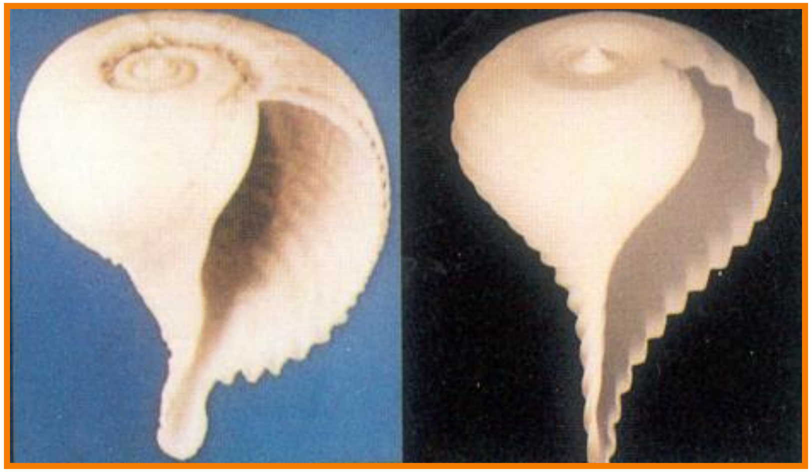
Generate different shells by varying parameters

Different generating curves

Fowler et al. Figure 3

Example: Seashells

Generate many interesting shells with a simple procedural model!



Fowler et al. Figures 4,5,7

Procedural Modeling

Sweeps

Fractals

Grammars

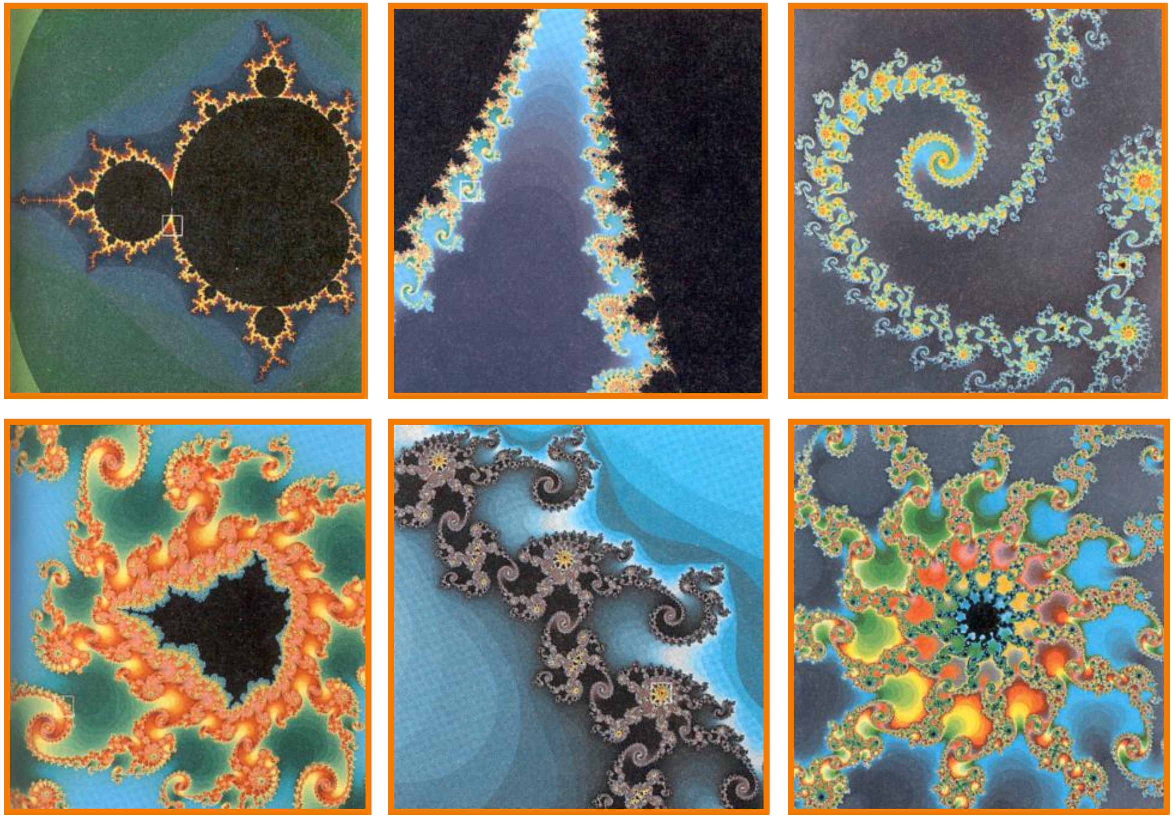
Probabilistic models

Probabilistic grammars

Fractals

Defining property:

- Self-similar with infinite resolution



Mandelbrot Set

H&B Figure 10.100

Fractals

Useful for describing natural 3D phenomenon

- Terrain
- Plants
- Clouds
- Water
- Feathers
- Fur
- etc.

H&B Figure 10.80

Fractal Generation

Deterministically self-similar fractals

- Parts are scaled copies of original

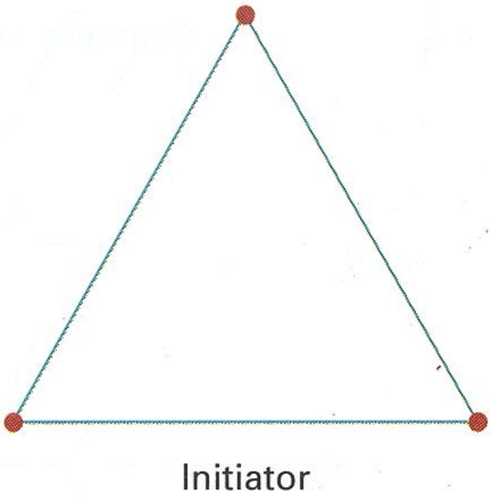
Statistically self-similar fractals

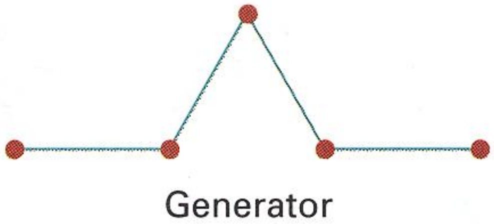
- Parts have same statistical properties as original

Deterministic Fractal Generation

General procedure:

- Initiator: start with a shape
- Generator: replace subparts with scaled copy of original

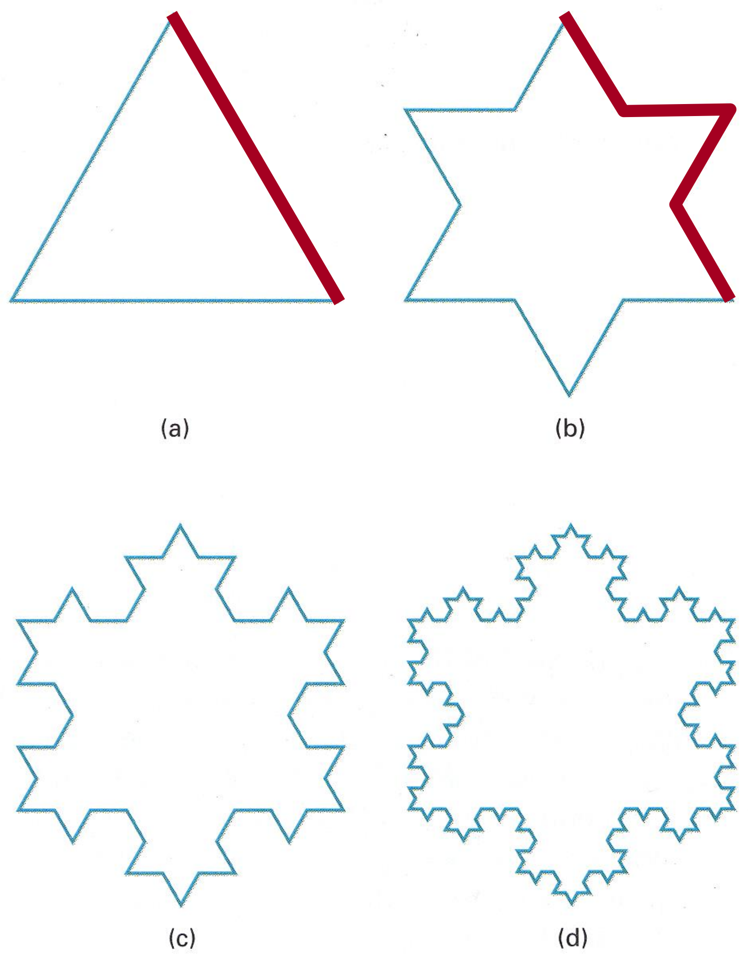




H&B Figure 10.68

Deterministic Fractal Generation

Apply generator repeatedly

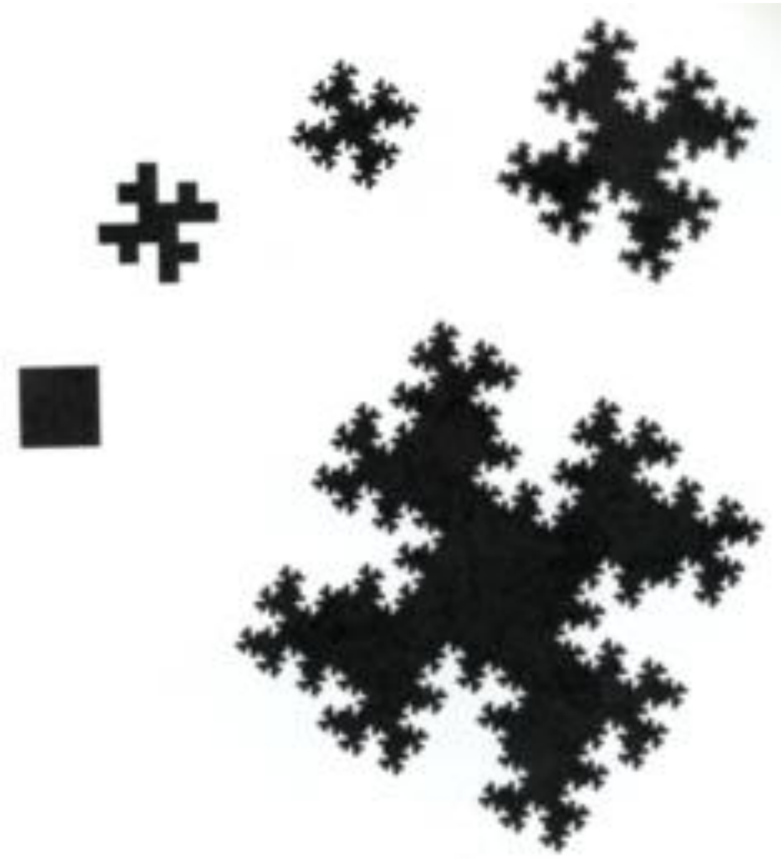


Koch Curve

H&B Figure 10.69

Deterministic Fractal Generation

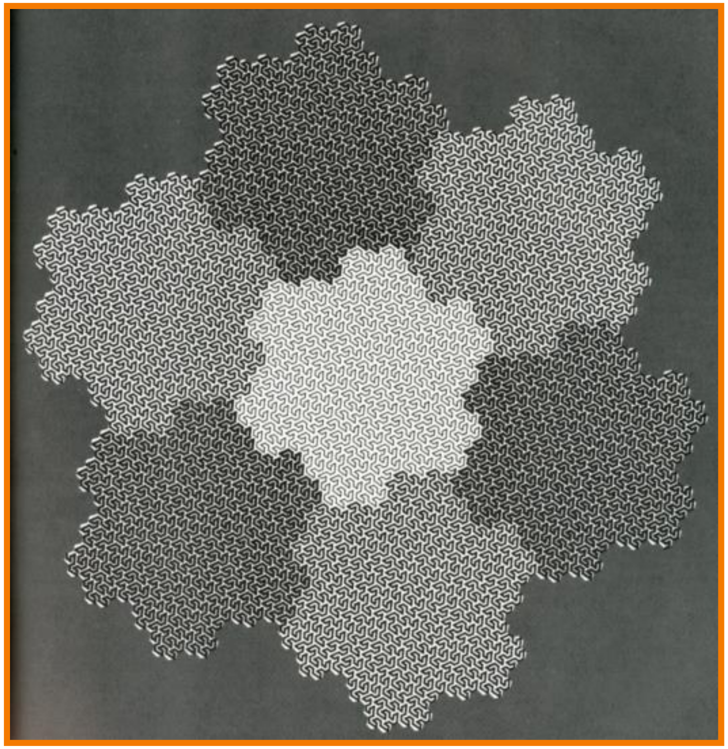
Useful for creating interesting shapes!



Mandelbrot Figure X

Deterministic Fractal Generation

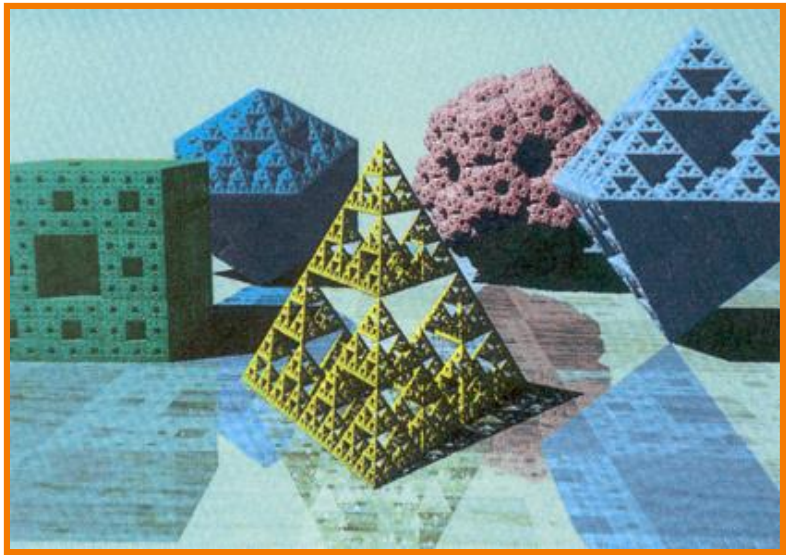
Useful for creating interesting shapes!



Mandelbrot Figure 46

Deterministic Fractal Generation

Useful for creating interesting shapes!



H&B Figures 75 & 109

Fractal Generation

Deterministically self-similar fractals

- Parts are scaled copies of original

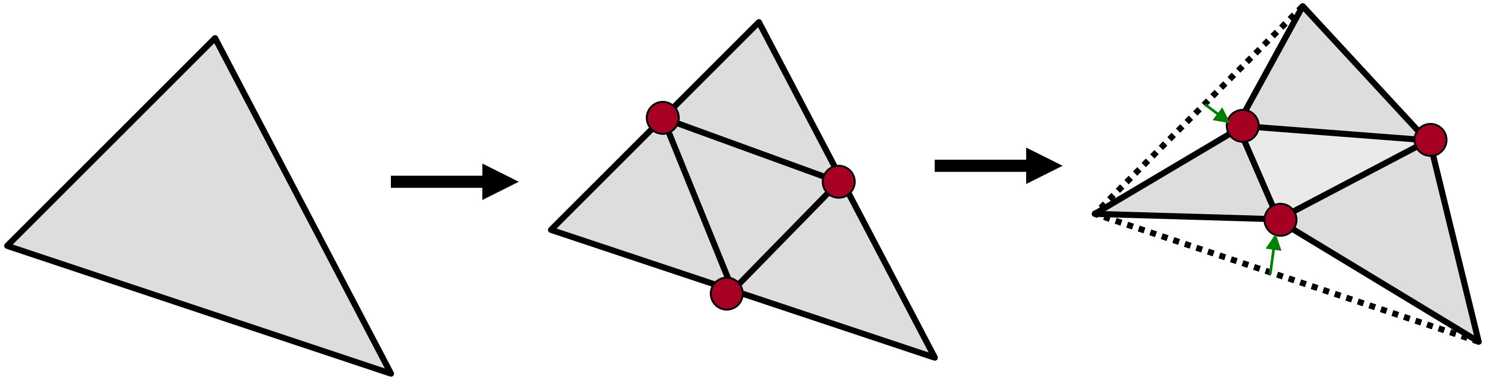
Statistically self-similar fractals

- Parts have same statistical properties as original

Statistical Fractal Generation

General procedure:

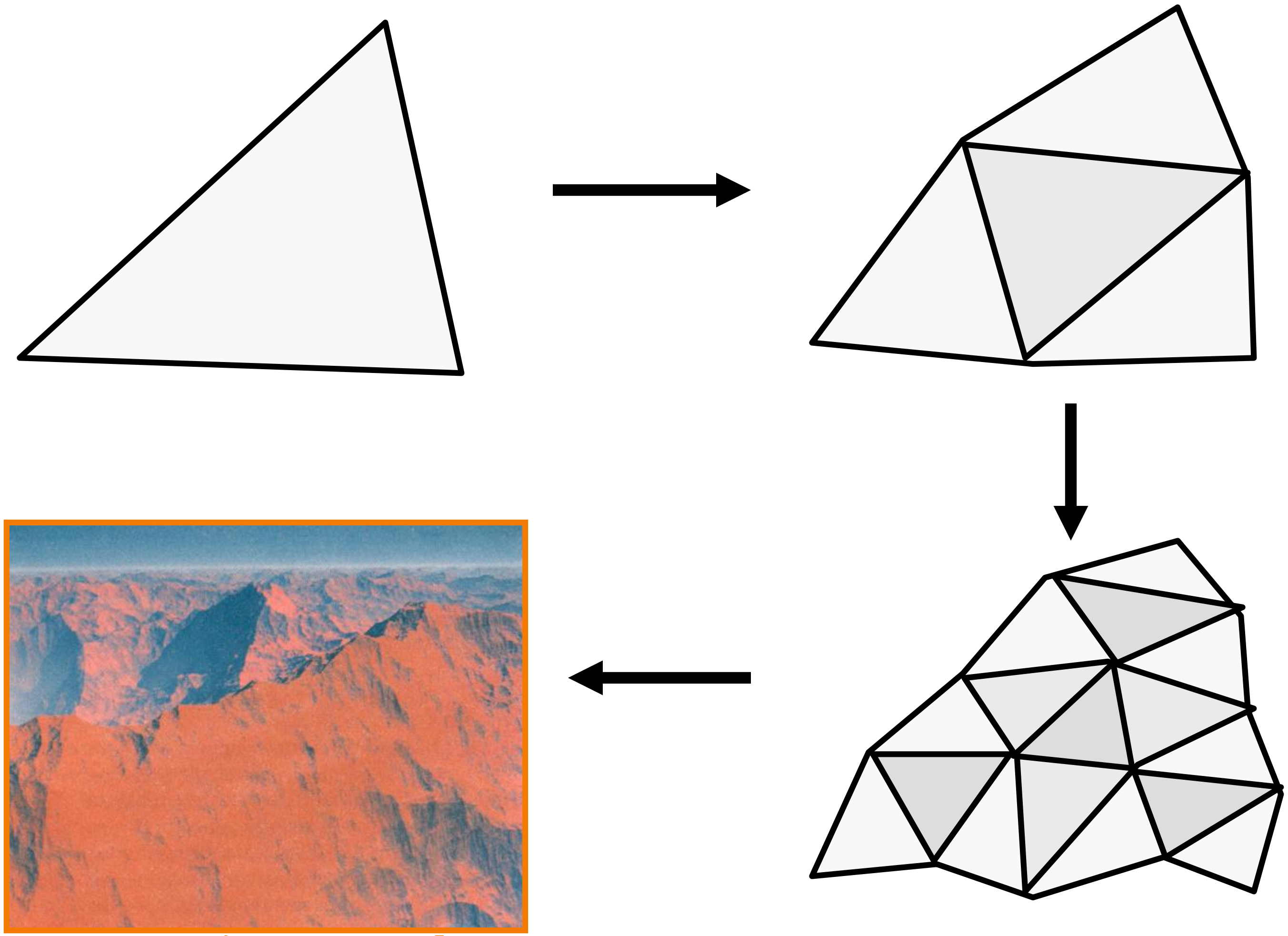
- Initiator: start with a shape
- Generator: replace subparts with a self-similar random pattern



Random Midpoint Displacement

Statistical Fractal Generation

Example: terrain



H&B Figure 10.83b

Statistical Fractal Generation

Useful for creating mountains

H&B Figure 10.83a

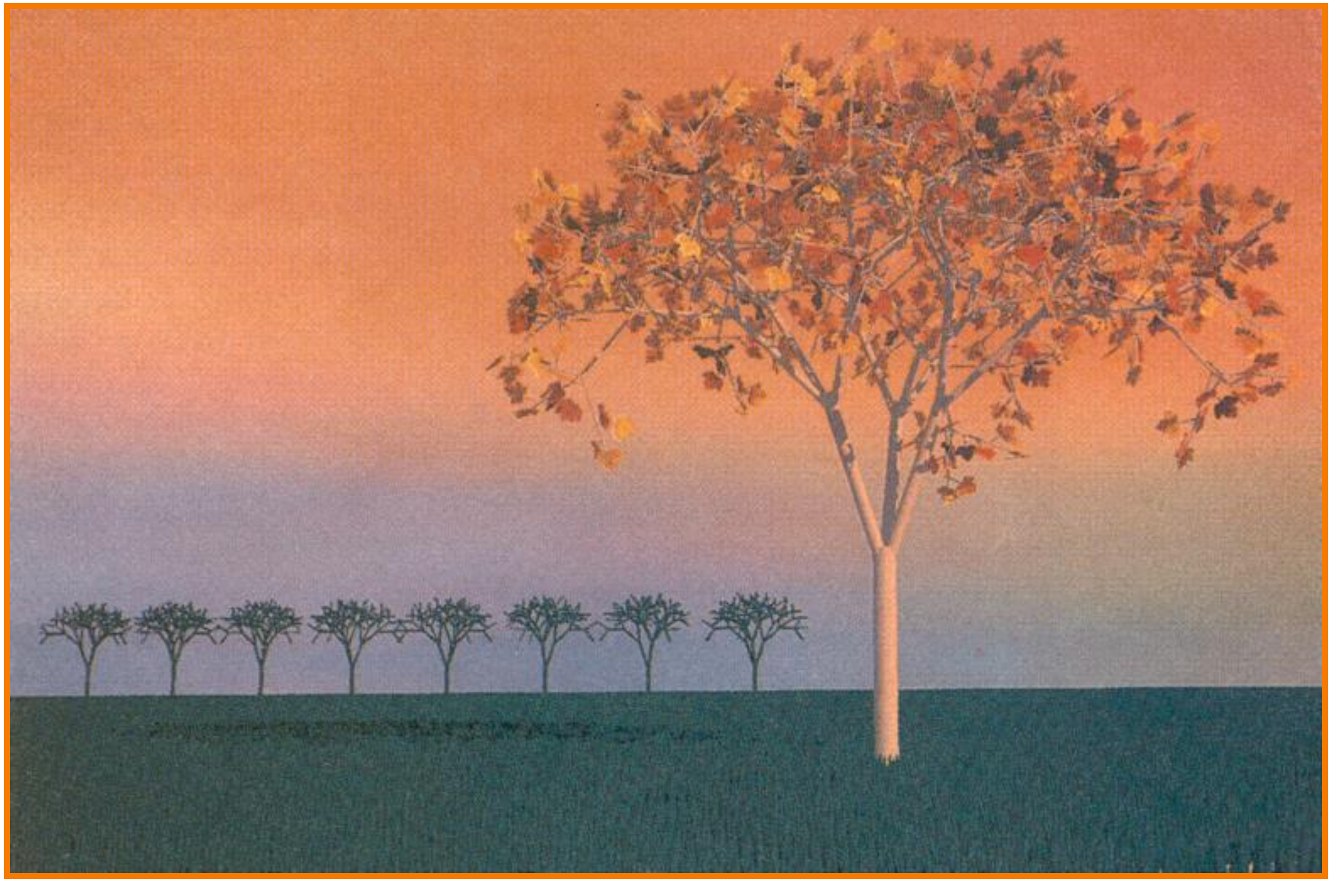
Statistical Fractal Generation

Useful for creating 3D plants

H&B Figure 10.82

Statistical Fractal Generation

Useful for creating 3D plants



H&B Figure 10.79

Procedural Modeling

Sweeps

Fractals

Grammars

Probabilistic models

Probabilistic grammars

Grammars

Generate description of geometric model
by applying production rules

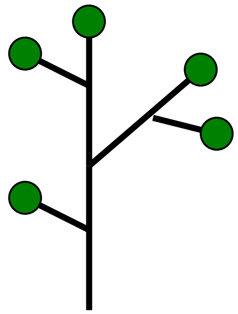
$$
\begin{aligned}
S &\longrightarrow AB \\
A &\longrightarrow Ba \mid a \\
B &\longrightarrow Ab \mid b
\end{aligned}
$$

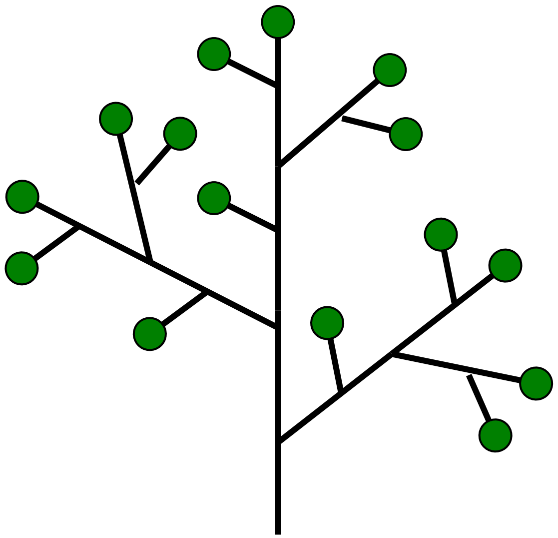
ab
bab
baab
abaab
.
.
.

Grammars

Useful for creating plants

Tree $\longrightarrow$ Branch Tree | Leaf
Branch $\longrightarrow$ Cylinder | [Tree]



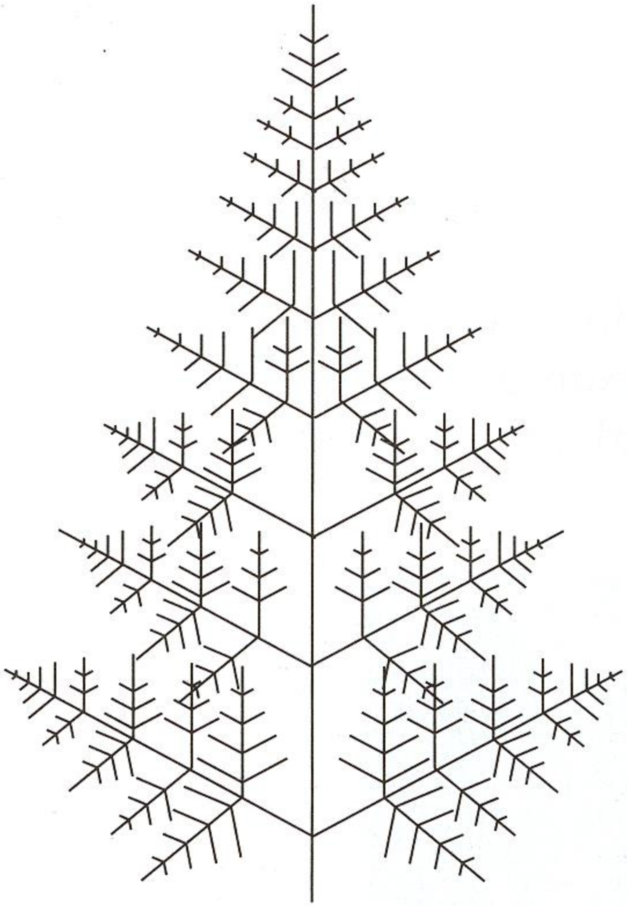


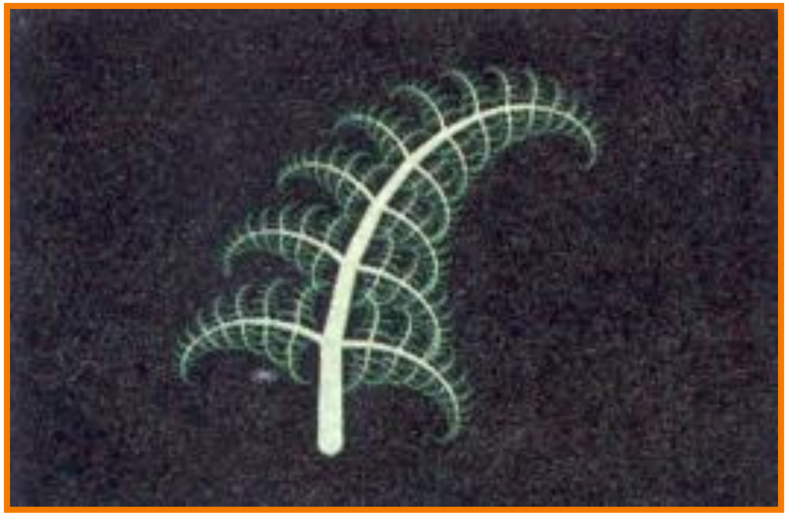
C[CL]C[C[CL][CL]]C[[CL][CL]]

C[*]C[*][*]

Grammars

Useful for creating plants





H&B Figure 10.77

Procedural Modeling

Sweeps

Fractals

Grammars

Probabilistic models

Probabilistic grammars

Probabilistic Models

Exemplar scenes

+

Database of Scenes

Probabilistic Model of Shape

Synthesized novel scenes

Probabilistic Models

Exemplar scenes

+

Database of Scenes

Probabilistic Model of Shape

Synthesized novel scenes

Challenge
Need to learn a model with great generality from few examples

Probabilistic Model of Scenes

Represent the probability of a scene S by a generative model based on category cardinalities (c), support hierarchy topology relationships (t), and spatial arrangement relationships (a)

$$P(S) = P(c,t,a) = P(a|t,c)\, P(t|c)\, P(c)$$

Exemplar scenes

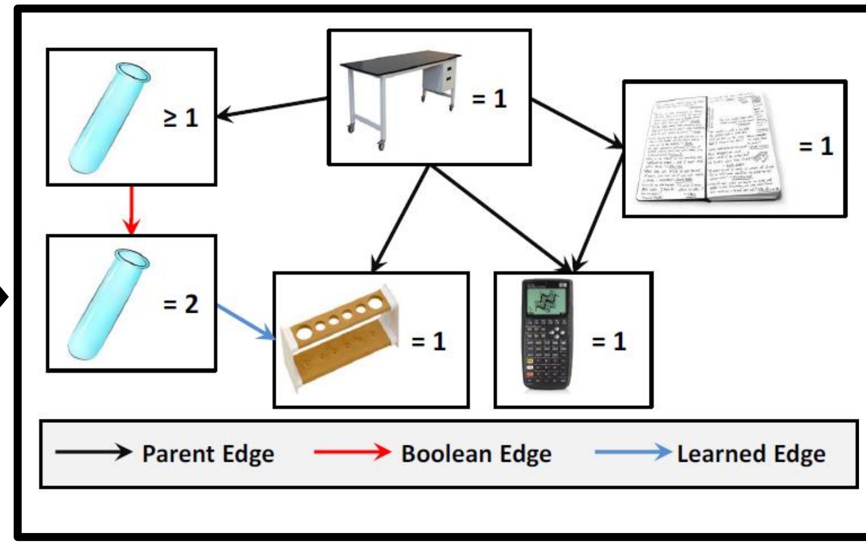
Probabilistic Model of Scenes

Category cardinalities: $P(c)$

- Represent with Bayesian network
- Boolean random variables (# desks > 1?)
- Add support surface constraints

Lab table	Test tube rack	Test tube	Notebook	Calculator
1	1	2	0	0
1	0	1	0	1
1	0	0	1	1

Object frequencies in target scenes
+ support constraints



Bayesian network

Probabilistic Model of Scenes

Support relationships: $P(t|c)$

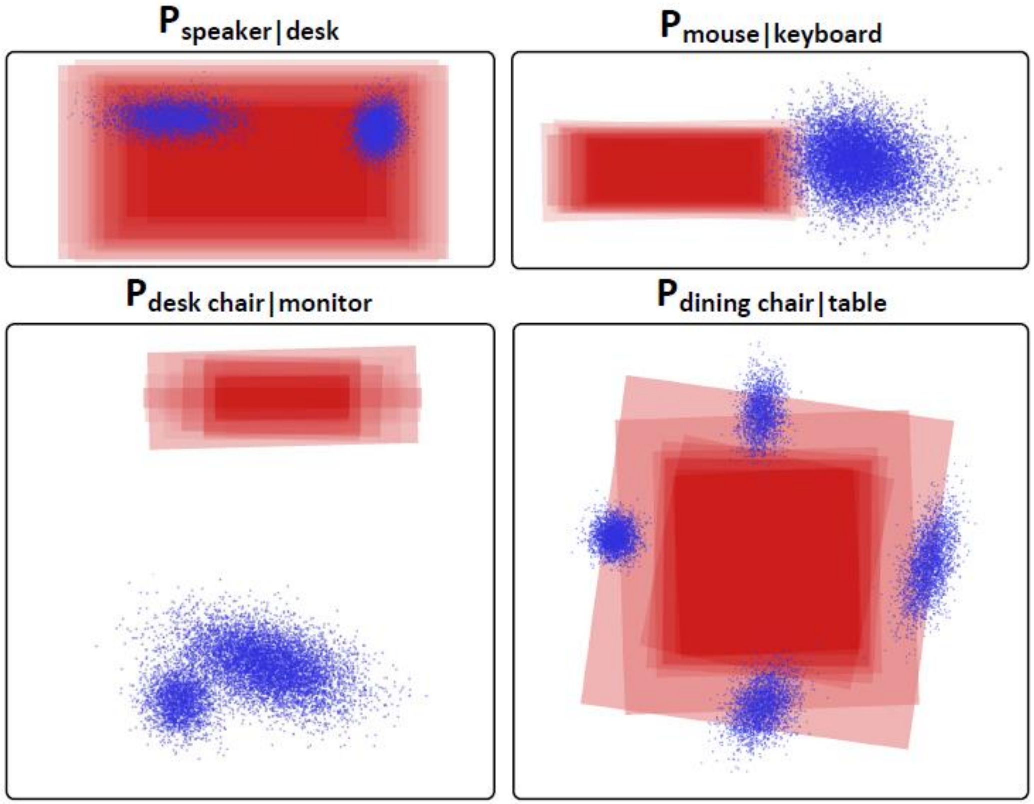
- Boolean random variables (desk supports keyboard?)
- Learn frequencies for pairs of categories
- Total probability is product over all objects in scene

$$P(t|c) = \prod_{o} P(C(o), C(support(o)))$$

Probabilistic Model of Scenes

Spatial arrangements: $P(a|t,c)=R(a,t,c)S(a,t,c)$

- Random variables for relative positions and orientations
- Pairwise distributions of spatial relationships

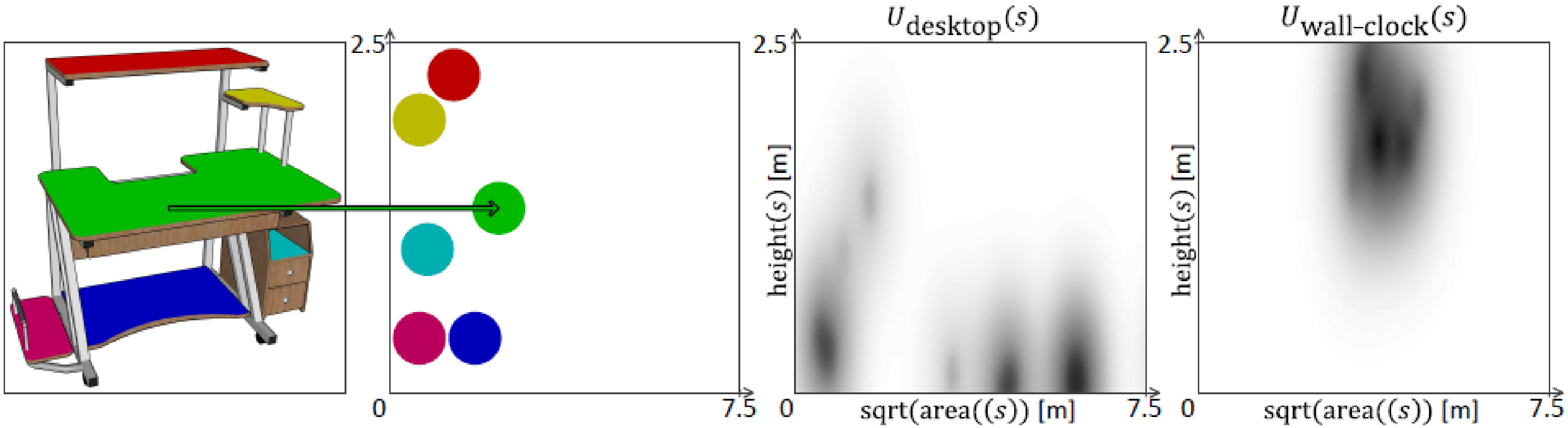


Distributions of spatial relationships for pairs of object categories

Probabilistic Model of Scenes

Spatial arrangements: $P(a|t,c)=R(a,t,c)S(a,t,c)$

- Random variables for relative positions and orientations
- Pairwise distributions of spatial relationships
- Feature distributions for positions on support surfaces



Distributions of geometric features of support surfaces

Side Note on Object Categories

Define categories of objects based on their contexts in a scene rather than basic functions

- Learned from examples by clustering of objects with similar spatial neighborhoods

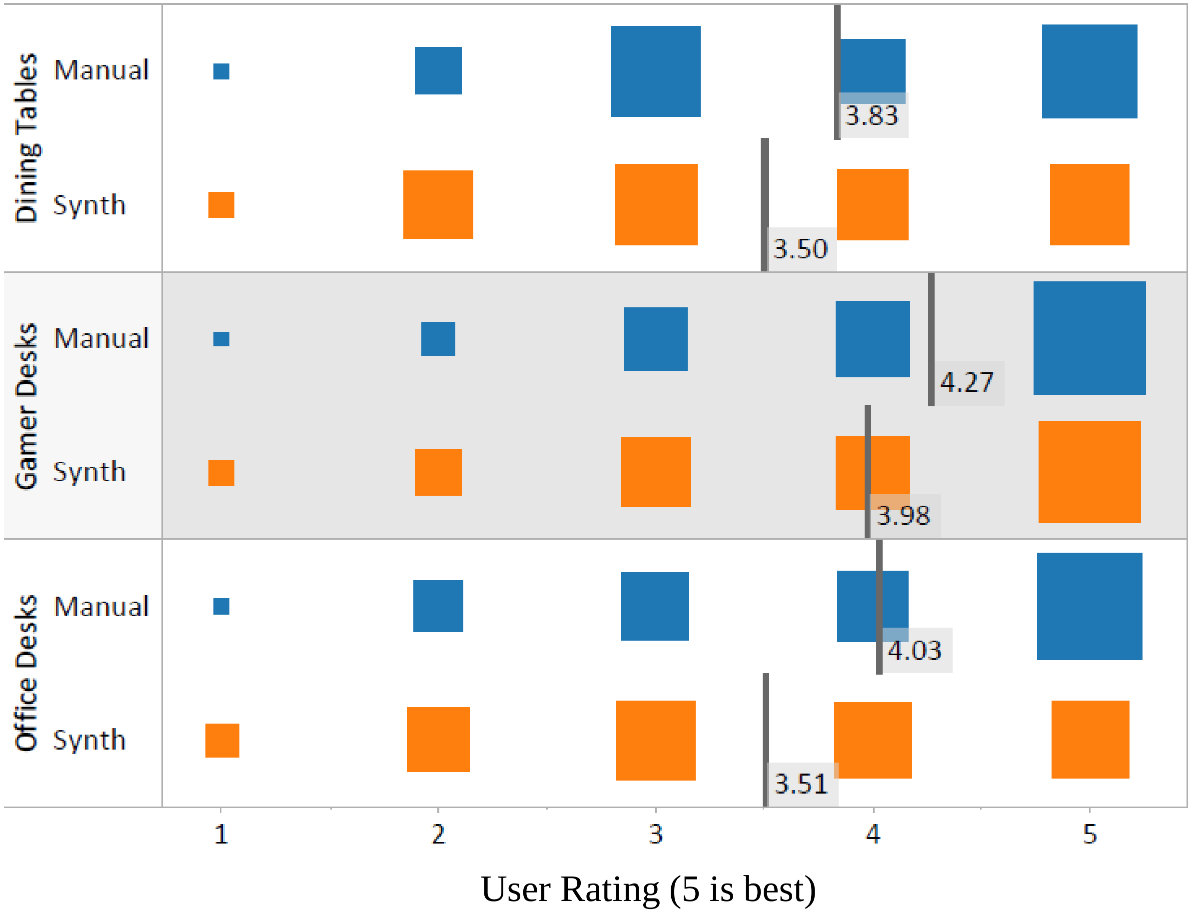
Some Contextual Object Categories

Scene Synthesis Results

Synthesized novel scenes

Scene Synthesis Results

User study suggests that people find our synthesized scenes almost as good as manually created ones



Procedural Modeling

Sweeps

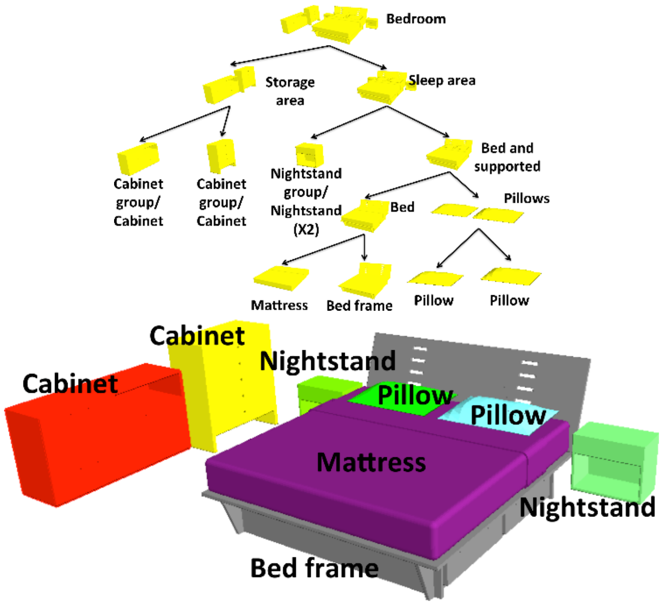
Fractals

Grammars

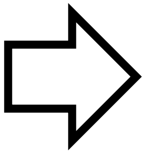
Probabilistic models

Probabilistic grammars

Probabilistic Grammars

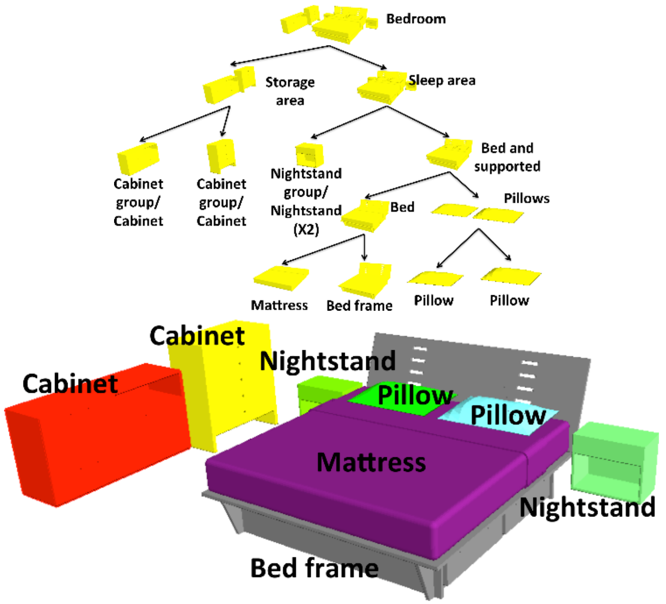


Training set of labeled scene graphs



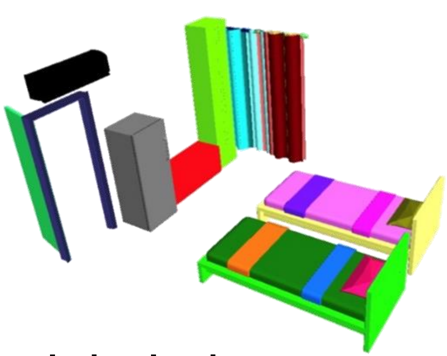
Probabilistic Model of Shape

Probabilistic Grammars



Training set of labeled scene graphs

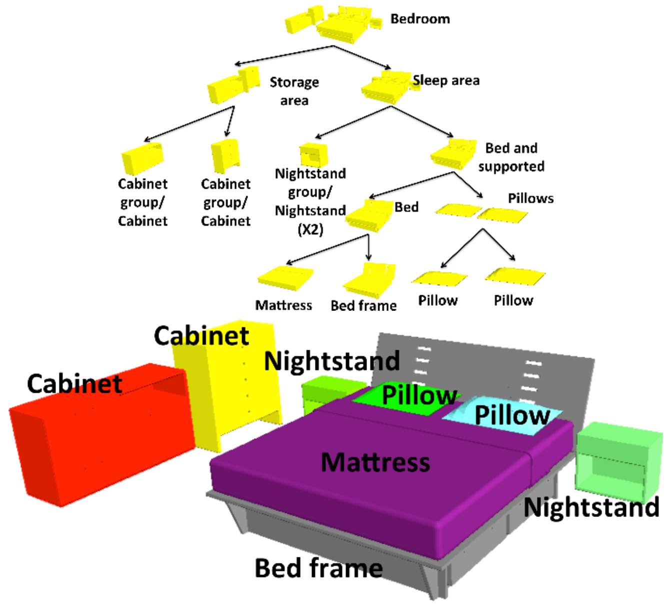
+



Unlabeled test scene

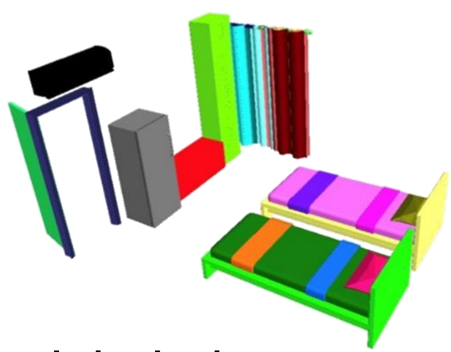
Probabilistic Model of Shape

Probabilistic Grammars



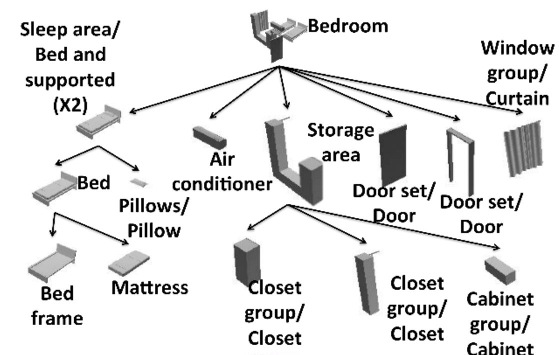
Training set of labeled scene graphs

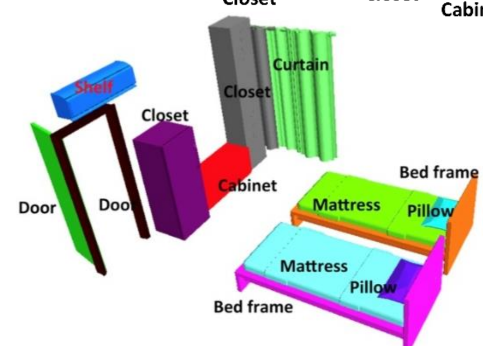
+



Unlabeled test scene

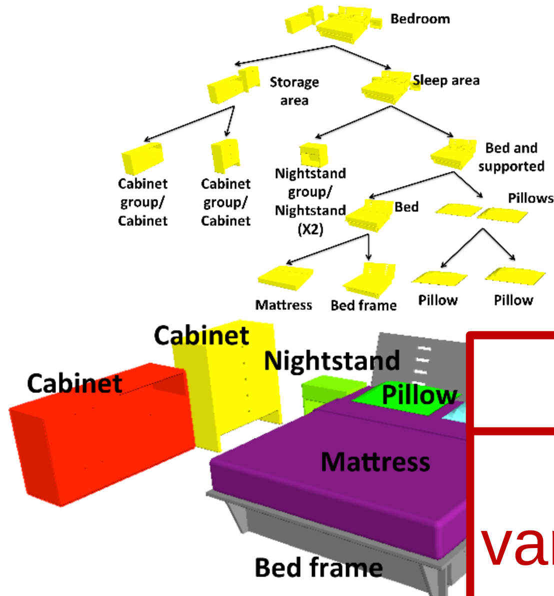
Probabilistic Model of Shape





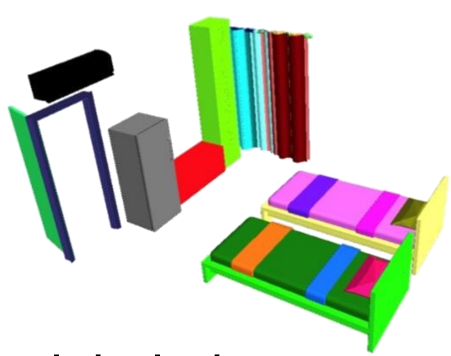
Labeled test scene graph

Probabilistic Grammars



Training set of labeled

+

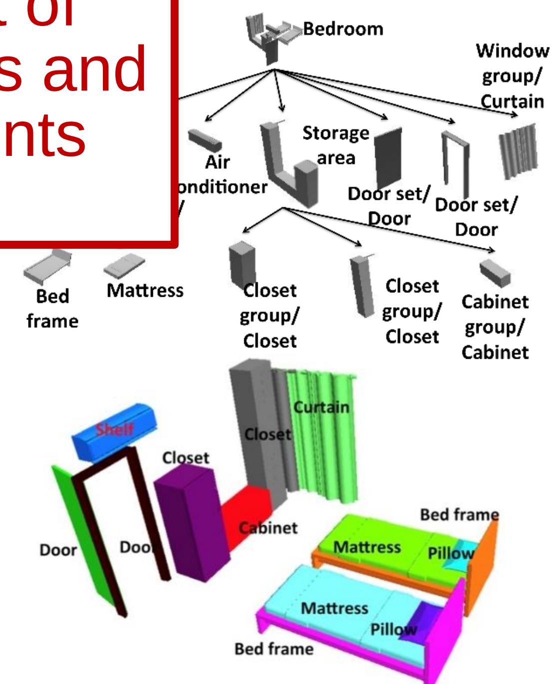


Unlabeled test scene

Probabilistic Model of Shape

Challenge

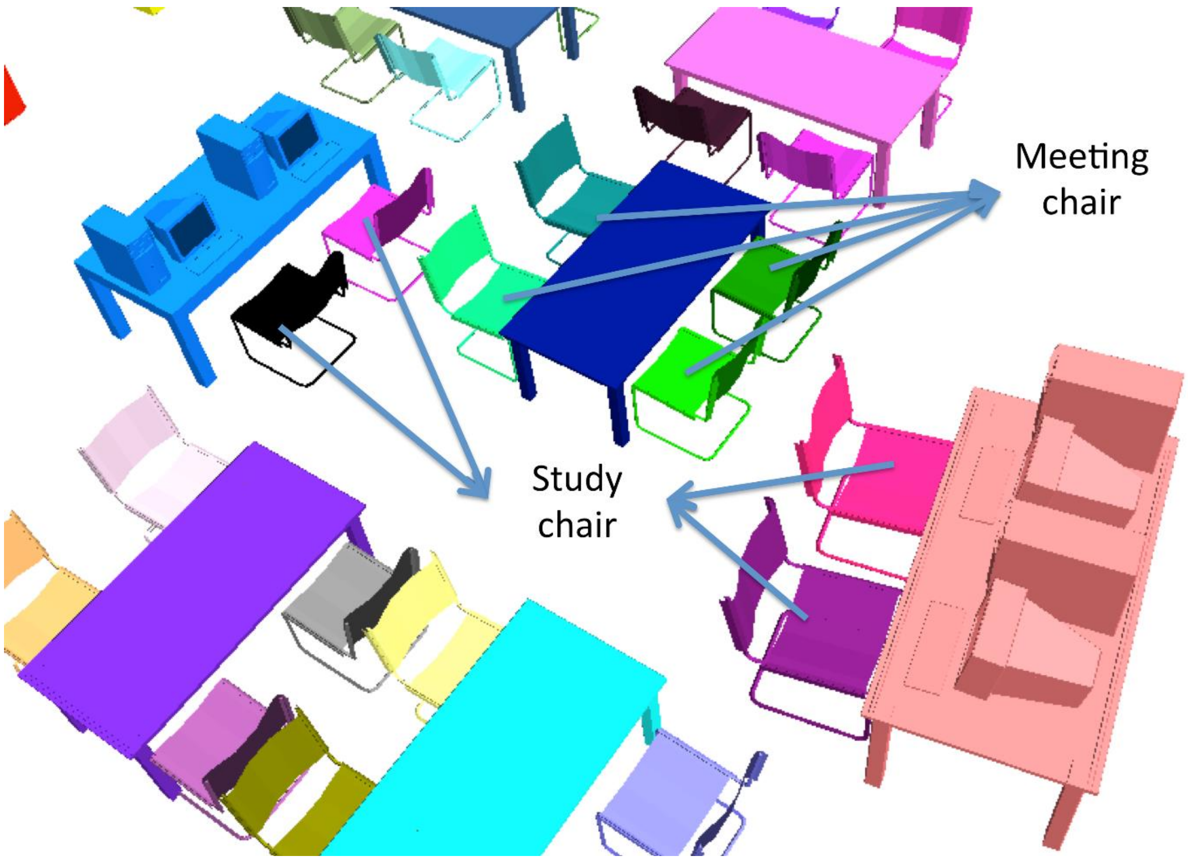
Scenes have a lot of variability in the types and spatial arrangements of objects

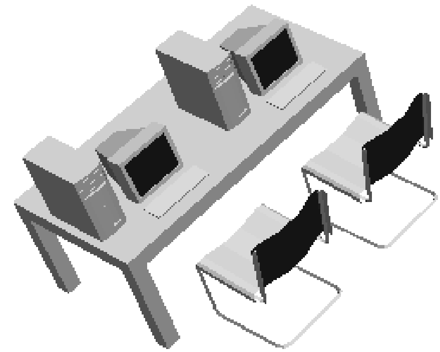


Labeled test scene graph

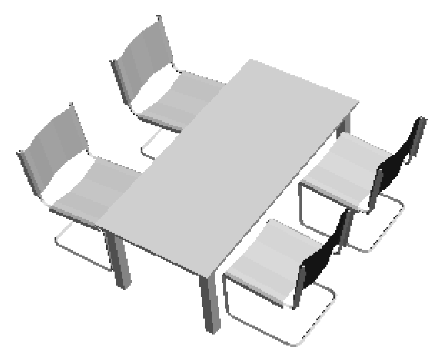
Probabilistic Grammars

Semantic and functional relationships are often more prominent within hierarchical contexts





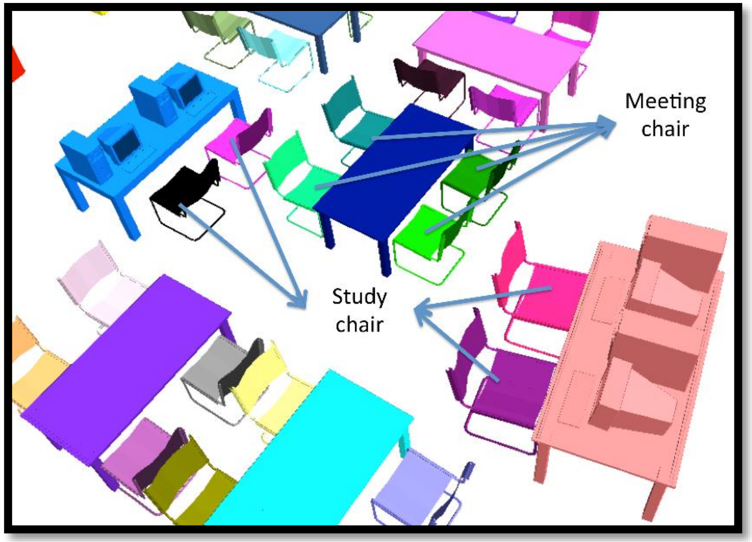
Study area

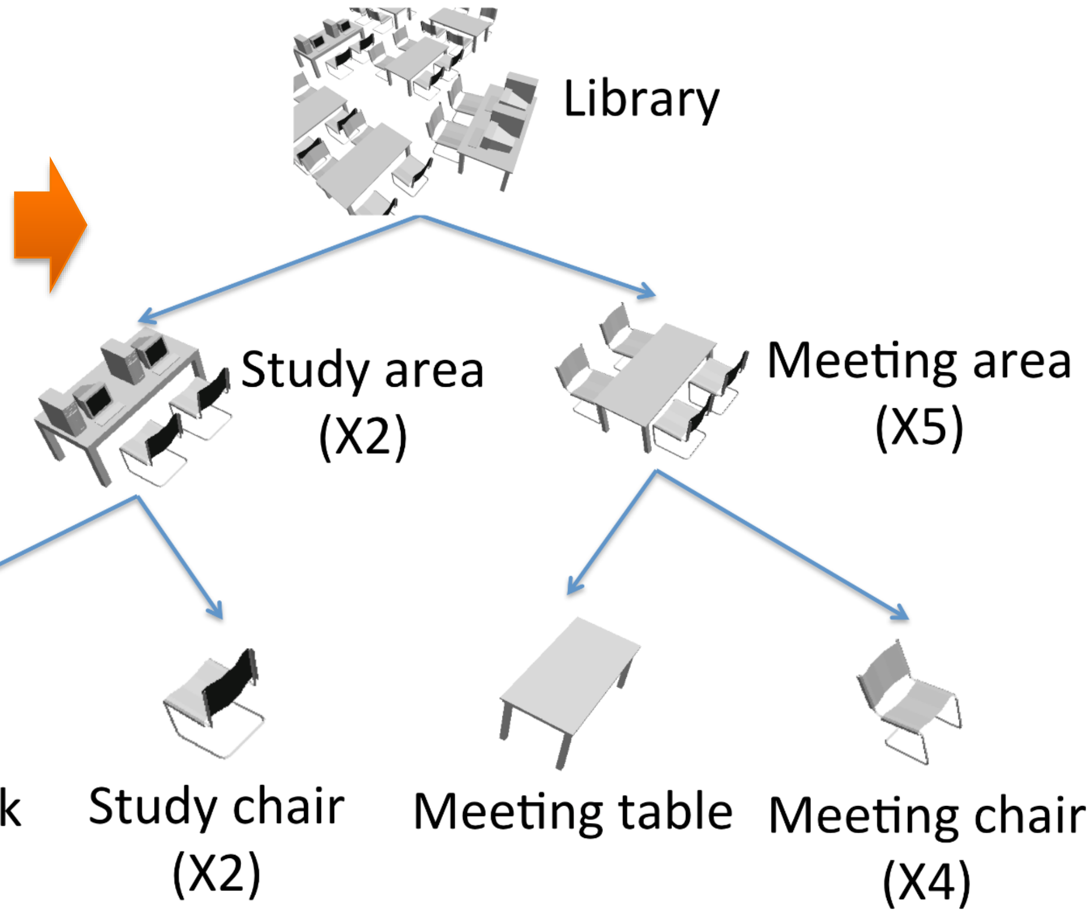


Meeting area

Hierarchical Grammar

We learn a probabilistic grammar from examples, and then use it to parse new test scenes





Hierarchical Grammar

Labels: object group, object category, object part

sleep area, bed, curtain piece

Rules: derivation from a label to a list of labels

bed → bed frame mattress

Hierarchical Grammar

Probabilities:

Derivation: $P_{nt}(rule \mid lhs)$

$$\text{bed} \xrightarrow[P\,=\,0.8]{} \text{frame} \quad \text{mattress}$$

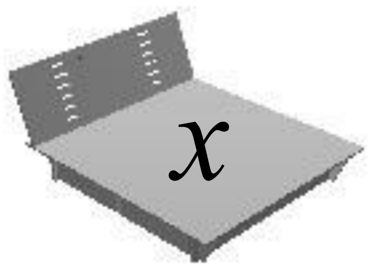
Cardinality distribution: $P_{Card}(\#, rhs \mid lhs)$

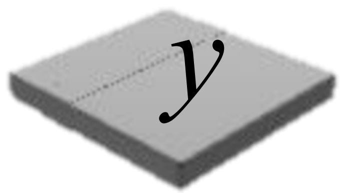
$$\text{sleep area} \rightarrow \text{bed} \quad \text{nightstand} \quad \text{rug} \quad \ldots$$

$P_{card}(* \mid sleeparea)$	**0**	**1**	**2**	**3**	**4+**
bed	…	…	…	…	…
nightstand	0.3	0.3	0.4	0	0
rug	…	…	…	…	…

Hierarchical Grammar

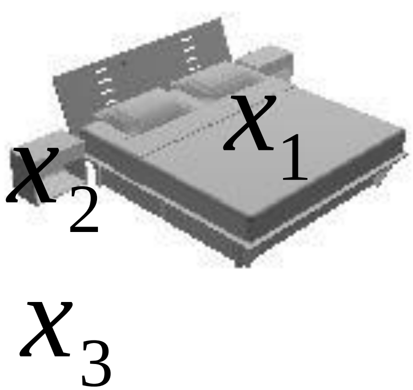
Shape descriptor probability: $P_g(x \mid label)$





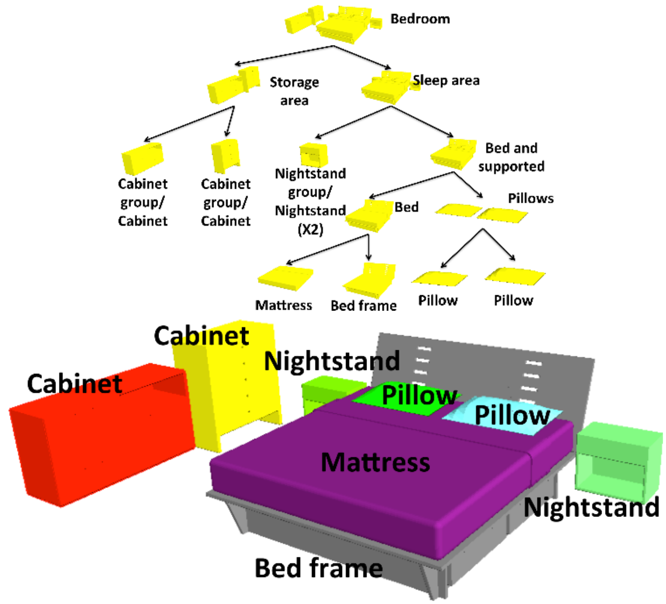
$$P_g(x \mid bedframe) > P_g(y \mid bedframe)$$

Spatial relationships: $P_g(v \mid lhs, rhs1, rhs2)$



$$P_s(x_1, x_2 \mid sleeparea, bed, nightstand) >$$
$$P_s(x_1, x_3 \mid sleeparea, bed, nightstand)$$

Grammar Learning and Parsing



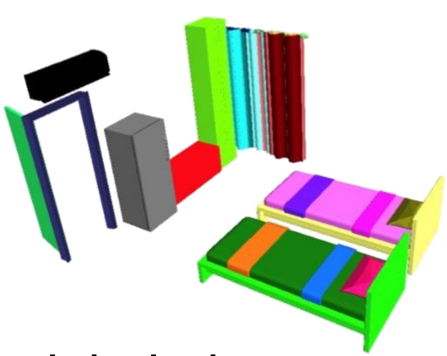
Training set of labeled scene graphs

Learn

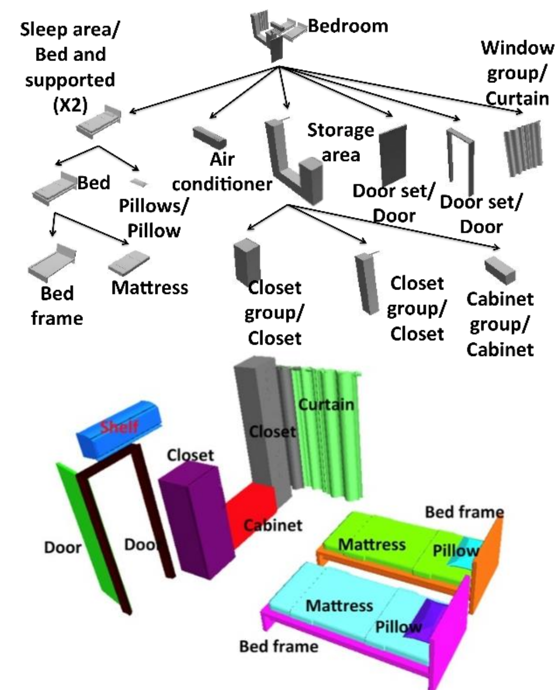
Probabilistic Hierarchical Grammar

Parse

+



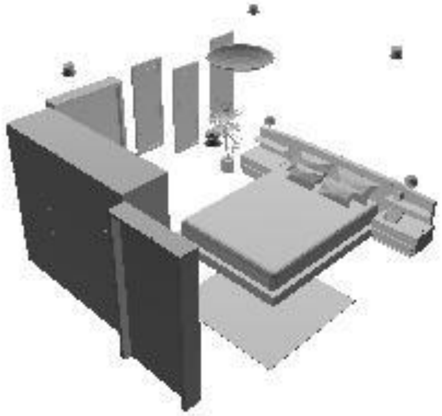
Unlabeled test scene



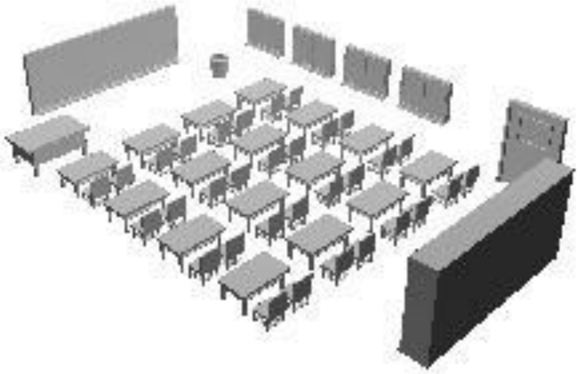
Labeled test scene graph

Hierarchical Grammar Results

Learned hierarchical probabilistic grammars from scenes in Trimble 3D Warehouse

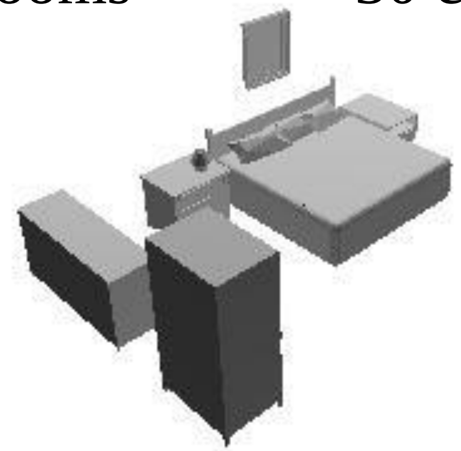


77 bedrooms

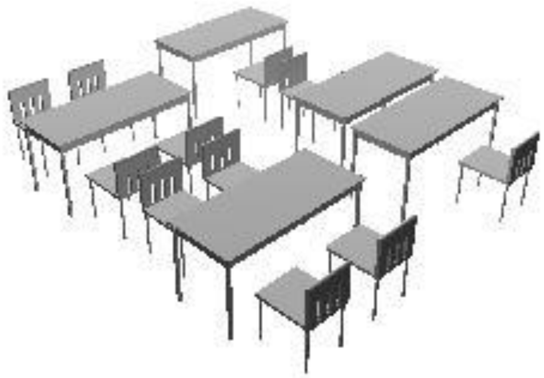


30 classrooms

8 libraries



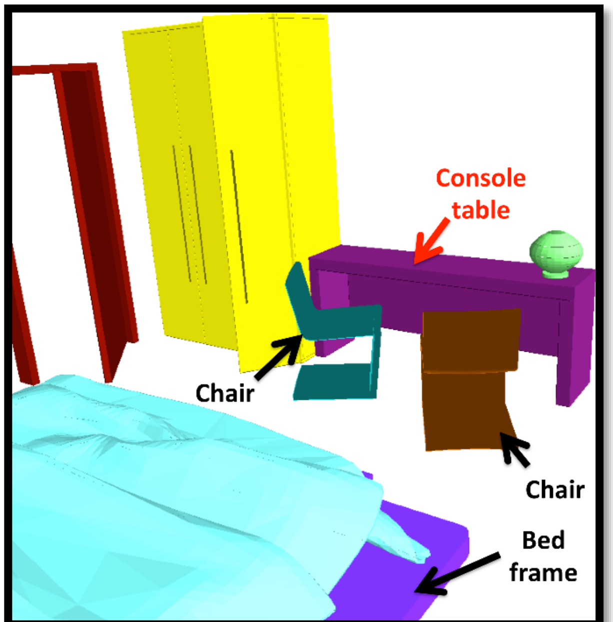
17 small bedrooms



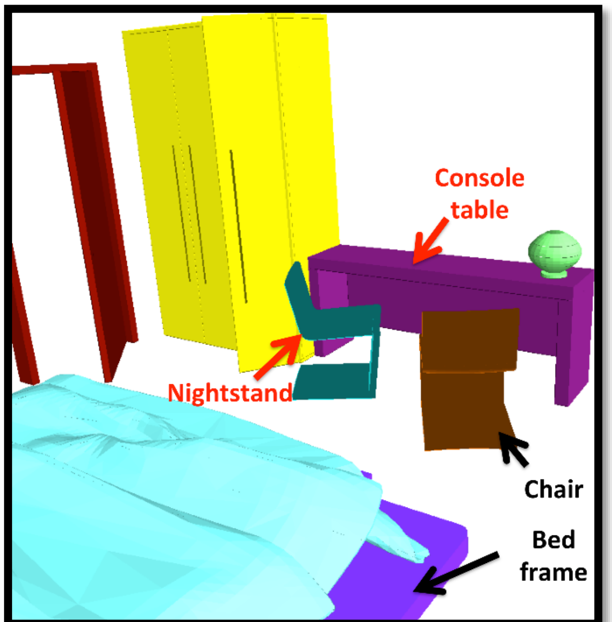
8 small libraries

Hierarchical Grammar Results

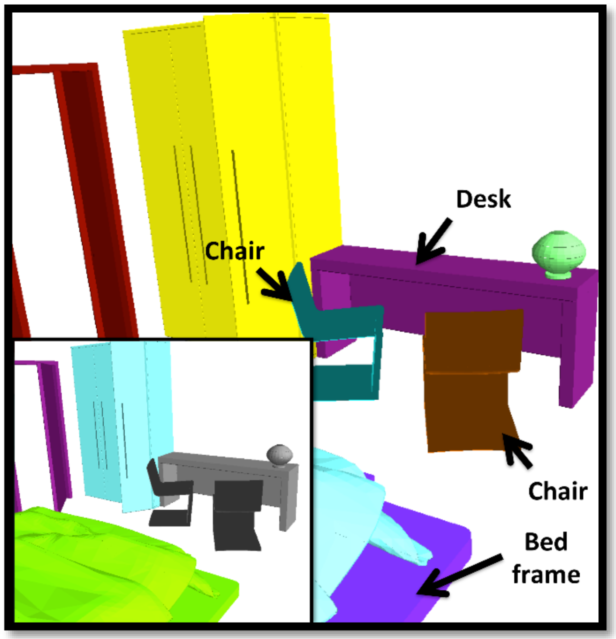
Parsed left-out scenes with learned grammar



Shape Only



Flat Grammar

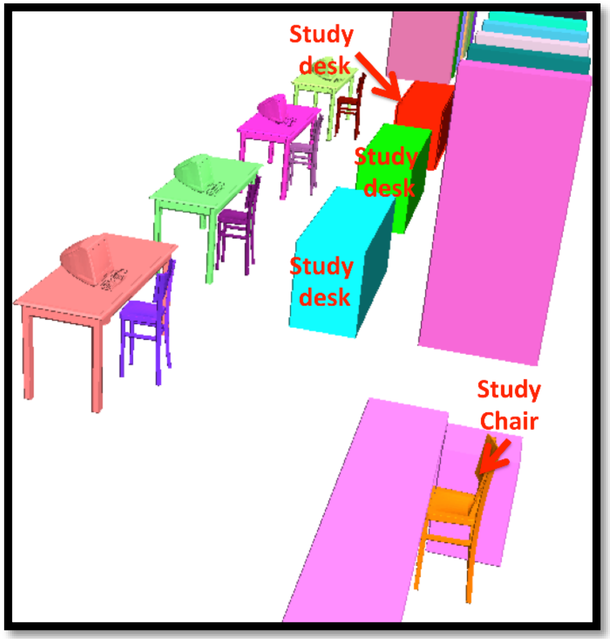


Our Hierarchical Grammar

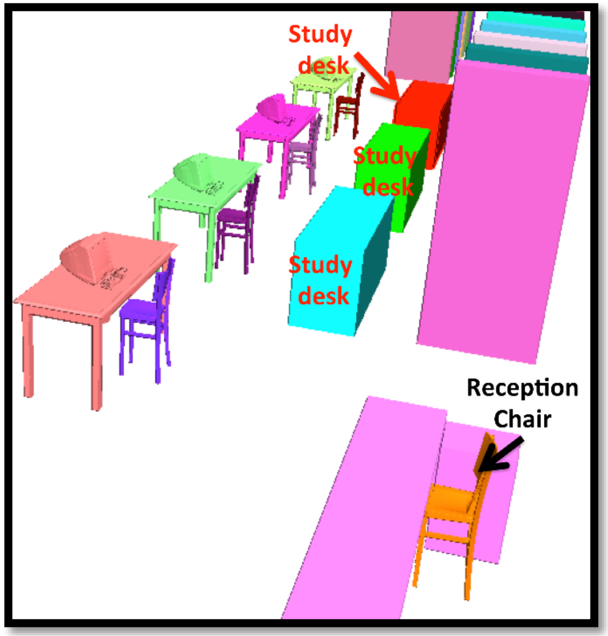
Comparison of our parsing results to other methods

Hierarchical Grammar Results

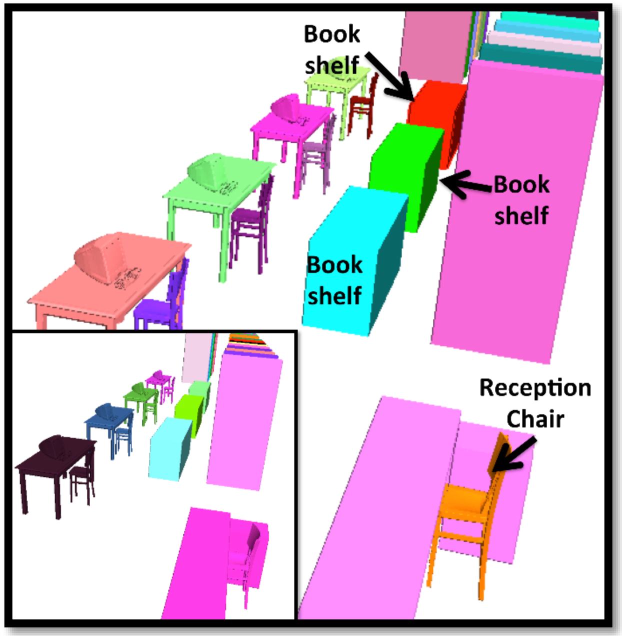
Parsed left-out scenes with learned grammar



Shape Only



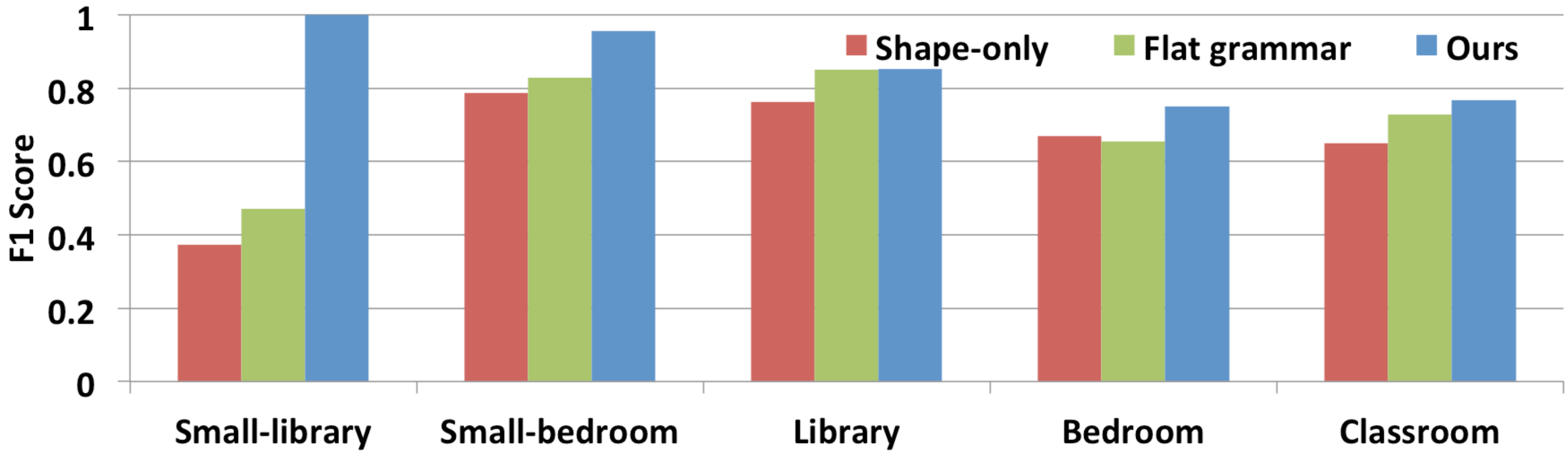
Flat Grammar



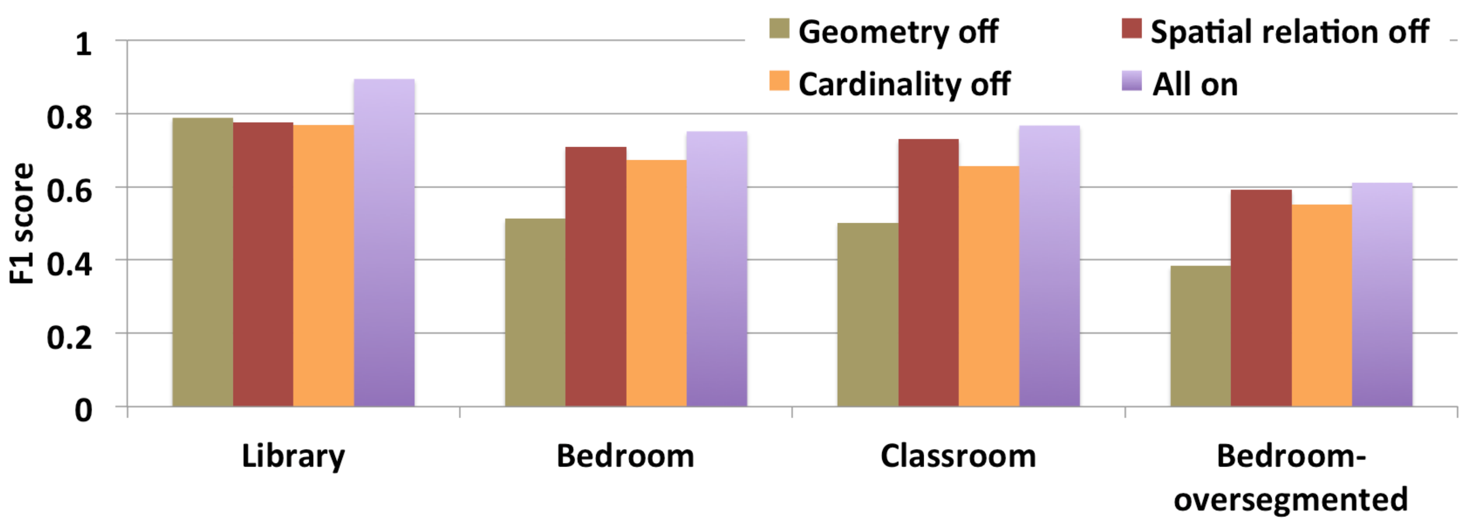
Our Hierarchical Grammar

Comparison of our parsing results to other methods

Hierarchical Grammar Results



Comparison of object classification



Impact of Individual Energy Terms

Procedural Modeling

Sweeps

Fractals

Grammars

Probabilistic models

Probabilistic grammars

Probabilistic programs ???

Procedural Modeling Summary

Motivation:

- Describe 3D models algorithmically

Methods:

- Sweeps, fractals, grammars
- Probabilistic models, grammars, and programs (?)

Advantages:

- Automatic generation
- Concise representation
- Parameterized classes of models
- Learned from examples