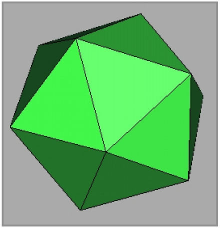
Structuring 3D Geometry based on Symmetry and Instancing Information

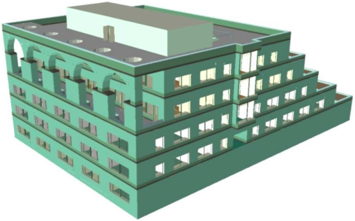
Aurelien MARTINET

May 14, 2007

3D Geometry

Representation



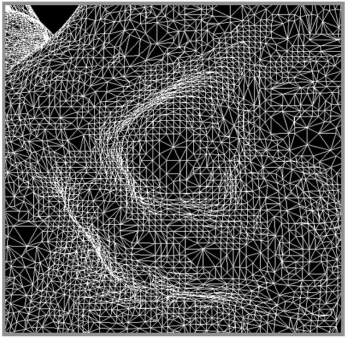


Question

How can these objects be represented in a computer ?

3D Geometry
Representation

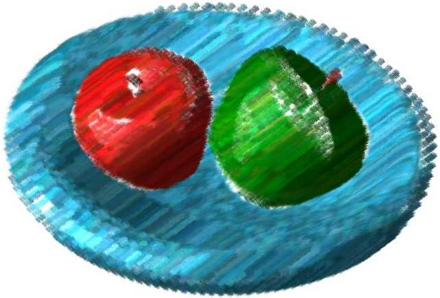
- 3D Geometry represented as a collection of polygons



3D Geometry

Treatements

- Rendering
- Animation
- Editing

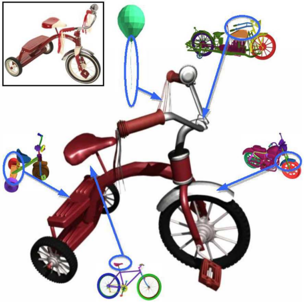


3D Geometry
Treatements

- Rendering
- Animation
- Editing

3D Geometry
Treatements

- Rendering
- Animation
- Editing

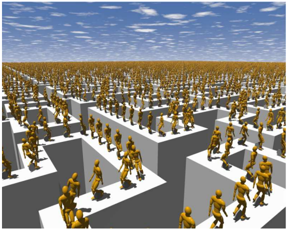


Motivations
Observations

Fact

Structure of Geometry is a key to Efficiency

- *Improve rendering speed*
- *Reduce memory usage*
- ...



Motivations

Observations

Fact

3D Geometry is often unstructured

$\Updownarrow$

Structural Information is not accessible

Raises two important questions:

1. What is Structural Information ?
2. Why is it not accessible ?

Motivations
Observations

Fact

3D Geometry is often unstructured

$\Updownarrow$

Structural Information is not accessible

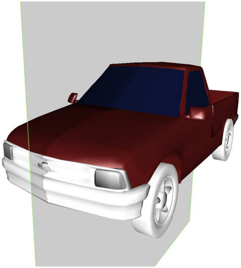
Raises two important questions:

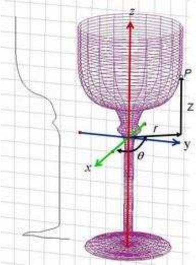
1. What is Structural Information ?
2. Why is it not accessible ?

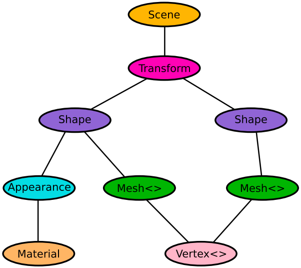
Structural Information

Generalities

- In Computer Graphics:
  - Symmetry Group of a Shape
  - Parameters of a Revolution Surface
  - Scene-Graph
  - ...







Accessibility of Structural Information

Fact

Structural Information is not accessible

Sources of Problems

- Asset Exchange
- Non-Interactive Modeling Techniques

Accessibility of Structural Information

Asset Exchange

Generation of 3D Geometry → Asset Exchange → Utilisation of 3D Geometry: Rendering, Editing, ...

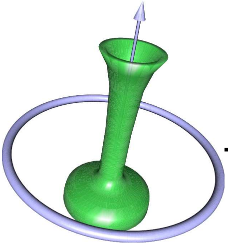
- Exchanging assets is a major problem:
    - Constraints due to multiple platforms, software and file formats
    - Must ideally preserve *the structure* of the geometry

Accessibility of Structural Information

Asset Exchange

Generation of 3D Geometry → Asset Exchange → Utilisation of 3D Geometry: Rendering, Editing, ...

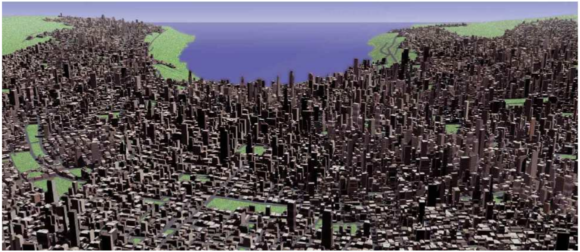
- Exchanging assets is a major problem:
  - Constraints due to multiple platforms, software and file formats
  - Must ideally preserve *the structure* of the geometry



→ Asset Exchange →

Accessibility of Structural Information

Non-Interactive Modeling Techniques



Pros

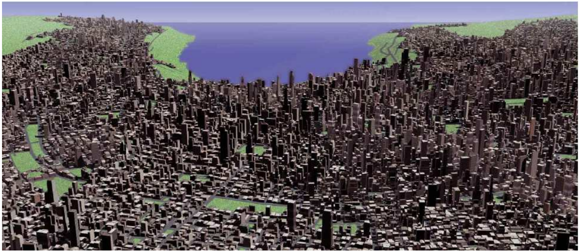
- Reach High-Complexity

Cons

- Unstructured Output

Accessibility of Structural Information

Non-Interactive Modeling Techniques



Pros

- Reach High-Complexity

Cons

- Unstructured Output

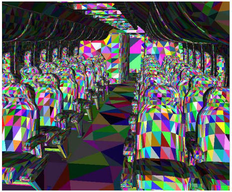
Structural Information
Scene and Objects

Structural Information as a two-scale notion:

- Object Level
- Scene Level

Approach
Pipeline and Outline

- A three-stage pipeline:



Unstructured 3D Geometry

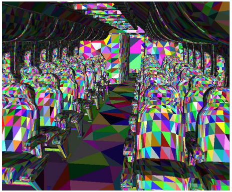
1

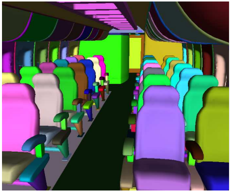
2

3

Approach
Pipeline and Outline

- A three-stage pipeline:





2

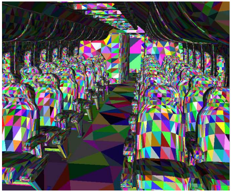
3

Unstructured 3D Geometry

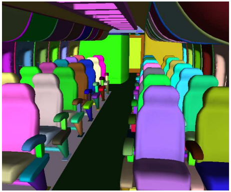
Pre-Processing 3D Geometry

Approach
Pipeline and Outline

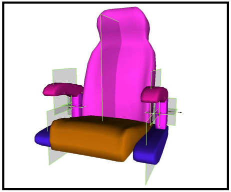
- A three-stage pipeline:



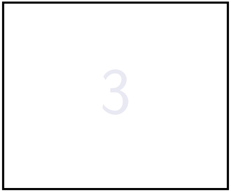
Unstructured 3D Geometry



Pre-Processing 3D Geometry



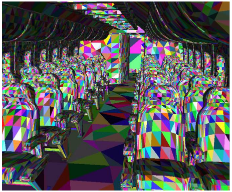
Object Level: Symmetry



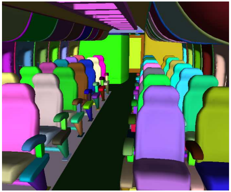
Approach

Pipeline and Outline

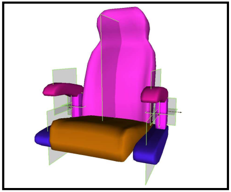
- A three-stage pipeline:



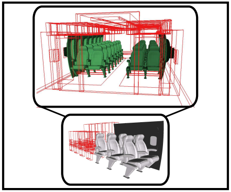
Unstructured 3D Geometry



Pre-Processing 3D Geometry



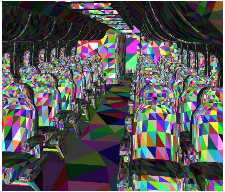
Object Level: Symmetry



Scene Level: Instancing

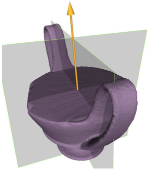
Contributions

1. A new way of partitioning unstructured geometry
2. Original methods to compute Symmetries of 3D Shapes
    - Algorithm for single shapes
    - Algorithm for composite shapes
3. A new shape congruency descriptor
4. Original method to represent 3D geometry as a hierarchy of instances

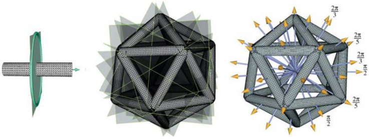
 

Contributions

1. A new way of partitioning unstructured geometry
2. Original methods to compute Symmetries of 3D Shapes
    - Algorithm for single shapes
    - Algorithm for composite shapes
3. A new shape congruency descriptor
4. Original method to represent 3D geometry as a hierarchy of instances



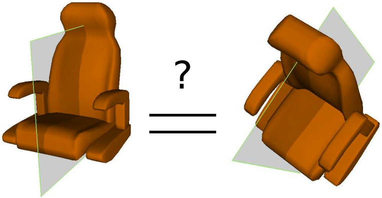
Single Shape



Composite Shape

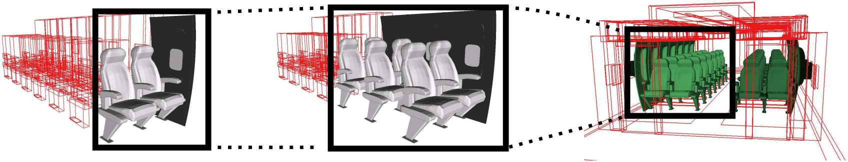
Contributions

1. A new way of partitioning unstructured geometry
2. Original methods to compute Symmetries of 3D Shapes
   - Algorithm for single shapes
   - Algorithm for composite shapes
3. A new shape congruency descriptor
4. Original method to represent 3D geometry as a hierarchy of instances



Contributions

1. A new way of partitioning unstructured geometry
2. Original methods to compute Symmetries of 3D Shapes
   - Algorithm for single shapes
   - Algorithm for composite shapes
3. A new shape congruency descriptor
4. Original method to represent 3D geometry as a hierarchy of instances



Objective

Question

What is an Object ?

Generalities

- Ill-Defined Notion
- Large Number of Possibilities

In this thesis, we define Objects as Tiles

Objective

Question

What is an Object ?

Generalities

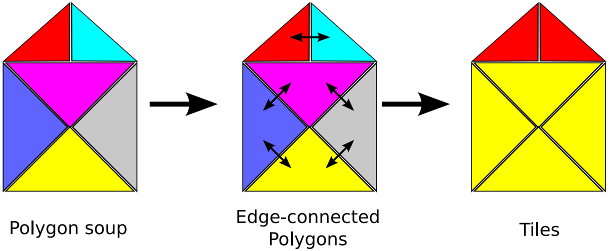
- Ill-Defined Notion
- Large Number of Possibilities

In this thesis, we define Objects as Tiles

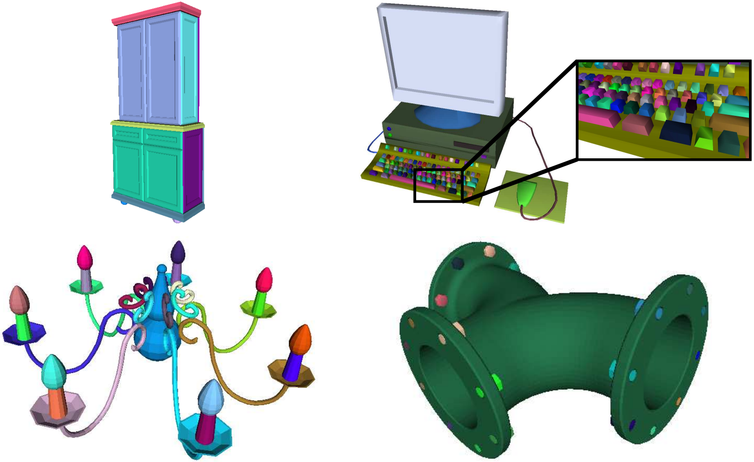
Tiles: Definition and Construction

Definition

A tile is a maximal set of edge-connected polygons

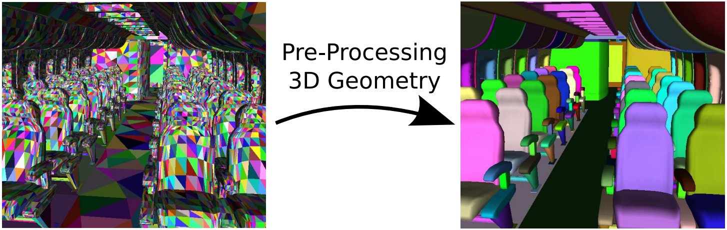


Examples of Tiles Decomposition



Computational-Friendly: Less than 2 minutes for 13M Polygons.

Summary

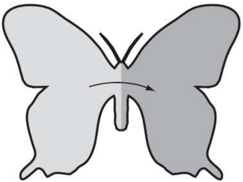


- Polygon Soup is now a Set of Tiles
- Next step is to compute the Symmetries of each Tile.

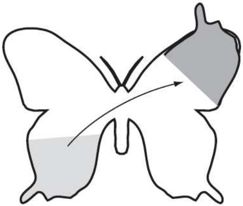
Problem Statement

Objectives

- Detect global symmetries of a 3D Shape
- Independent of Shape Tesselation



Global Symmetry

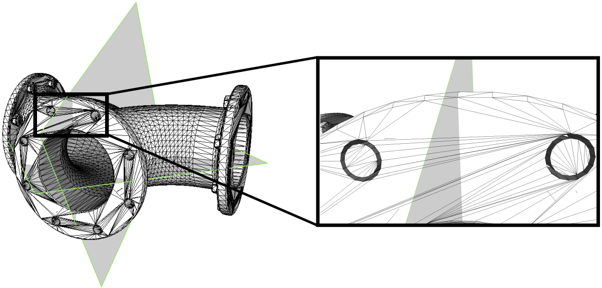


Local Symmetry

Problem Statement

Objectives

- Detect global symmetries of a 3D Shape
- Independent of Shape Tesselation

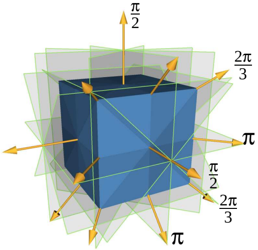


Problem Statement

Definition

Finding a symmetry of a shape $\mathcal{S}$ is equivalent to find an isometry $A = (\mathbf{X}, \alpha)$, such that:

$$A\mathcal{S} = \mathcal{S}$$



Problem Statement

Question

How efficiently found parameters of symmetries of a 3D Shape ?

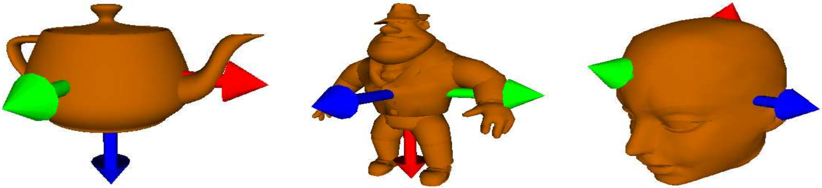
Approach

Our approach *extends* PCA-Based Approach:

1. What is PCA-Based Approach ?
2. What are its limitations ?
3. Our approach: The Generalized Moment Functions

PCA-Based Approach

- Principal Component Analysis (PCA)
  - Used to affect a local frame to a 3D Shape,
  - called Principal Axes



PCA-Based Approach

Fundamental Idea [Minovic, 1993]

ω is a Symmetry Axis of $\mathcal{S}$

$\Downarrow \quad \not\Uparrow$

ω is a Principal Axis of $\mathcal{S}$

The Method

1. Compute principal axes of the shape
2. Check each axis for a symmetry

PCA-Based Approach

Problems

What happends if principal axes are not uniquely defined ?

Properties of Principal Axes

- Along direction of maximum variance
- Unicity only if extrema are strict

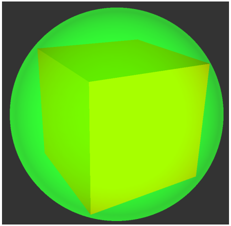
PCA-Based Approach

Problems

What happends if principal axes are not uniquely defined ?

Properties of Principal Axes

- Along direction of maximum variance
- Unicity only if extrema are strict



The Generalized Moment Functions

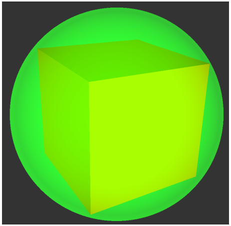
Variance Function *a.k.a.*
Moment of Order 2

$$\mathcal{M}^2(\omega) = \int_{s \in \mathcal{S}} \|\mathbf{s} \times \omega\|^2 \, \mathrm{d}\mathbf{s}$$

$$\Downarrow$$

Generalized Moment of Order k

$$\mathcal{M}^k(\omega) = \int_{s \in \mathcal{S}} \|\mathbf{s} \times \omega\|^k \, \mathrm{d}\mathbf{s}$$



$\mathcal{M}^2$

The Generalized Moment Functions

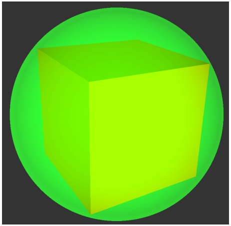
Variance Function *a.k.a.*
Moment of Order 2

$$\mathcal{M}^2(\omega) = \int_{\mathbf{s}\in\mathcal{S}} \|\mathbf{s}\times\omega\|^2 \,\mathrm{d}\mathbf{s}$$

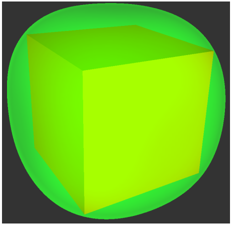
$$\Downarrow$$

Generalized Moment of Order k

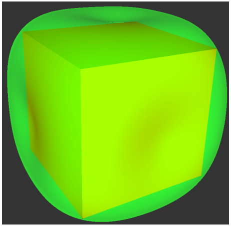
$$\mathcal{M}^k(\omega) = \int_{\mathbf{s}\in\mathcal{S}} \|\mathbf{s}\times\omega\|^k \,\mathrm{d}\mathbf{s}$$



$\mathcal{M}^2$



$\mathcal{M}^4$



$\mathcal{M}^6$

Definition

Property 1 of $\mathcal{M}^k$

Isometry A is a symmetry of $\mathcal{S}$

$$\Downarrow \quad \not\Uparrow$$

Isometry A is a symmetry of $\mathcal{M}^k$, for all k

Strategy

1. Candidates Symmetries are Symmetries of $\mathcal{M}^k$, for all k
2. Check candidates on the shape $\mathcal{S}$

Definition

Property 1 of $\mathcal{M}^k$

Isometry A is a symmetry of $\mathcal{S}$

$$\Downarrow \qquad \not\Uparrow$$

Isometry A is a symmetry of $\mathcal{M}^k$, for all k

Strategy

1. Candidates Symmetries are Symmetries of $\mathcal{M}^k$, for all k
2. Check candidates on the shape $\mathcal{S}$

Determination of the Axis of Symmetry

Property 2

ω is a symmetry axis of $\mathcal{M}^k$

$$\Downarrow \quad \not\Uparrow$$

$$\|\nabla \mathcal{M}^k(\omega)\|^2 = 0$$

Potential Symmetry Axis ω of the Shape by solving:

$$\forall k \quad (\nabla \mathcal{M}^k)(\omega) = \mathbf{0}$$

Determination of the Axis of Symmetry
Efficient Computation

- Closed-form expression of $\mathcal{M}^k$ for k even
- Using Spherical Harmonic (SH) Basis

$$\begin{aligned}\mathcal{M}^{2p}(\omega) &= \int_{\mathbf{s}\in\mathcal{S}} \|\mathbf{s}\times\omega\|^{2p}\,\mathrm{d}\mathbf{s} \\ &= \sum_{l=0}^{p}\sum_{m=-2l}^{2l} C_l^m Y_{2l}^m(\omega)\end{aligned}$$

Determination of the Axis of Symmetry

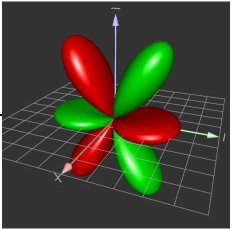
Efficient Computation

- Closed-form expression of $\mathcal{M}^k$ for k even
- Using Spherical Harmonic (SH) Basis

SH coefficient

$$\mathcal{M}^{2p}(\omega) = \int_{s \in \mathcal{S}} \|\mathbf{s} \times \omega\|^{2p} \, d\mathbf{s}$$

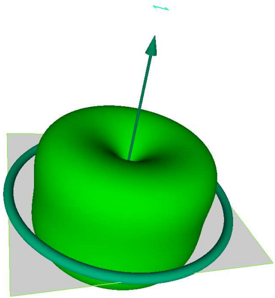
$$= \sum_{l=0}^{p} \sum_{m=-2l}^{2l} C_l^m Y_{2l}^m(\omega)$$



Determination of Symmetry Parameters

Property 3

Symmetries of $\mathcal{M}^{2p}$ are obtained by *testing* SH coefficients.



Determination of Symmetry Parameters

Example: Testing Revolution-Symmetry

Question

Has $\mathcal{M}^{2p}$ a revolution-symmetry around axis $\mathbf{n}$?

We use the following powerful property:

Property

A Moment Function has a revolution-symmetry around $\mathbf{z}-$axis if:

$$\forall l \quad \forall m \quad m \neq 0 \Rightarrow C_l^m = 0$$

Determination of Symmetry Parameters

Example: Testing Revolution-Symmetry

Question

Has $\mathcal{M}^{2p}$ a revolution-symmetry around axis $\mathbf{n}$?

We use the following powerful property:

Property

A Moment Function has a revolution-symmetry around $\mathbf{z}-$axis if:

$$\forall l \quad \forall m \quad m \neq 0 \Rightarrow C_l^m = 0$$

Determination of Symmetry Parameters

Example: Testing Revolution-Symmetry

Lead to a simple 2-step method:

1. Rotate $\mathcal{M}^{2p}$ to align axis $\mathbf{n}$ on $\mathbf{z}$
2. Test the nullity of the "new" coefficients C_l^m (up to a threshold)

Determination of Symmetry Parameters

- Equivalent properties exist for:
  - Planar symmetries,
  - fixed-angle rotational symmetries

Last step is to check candidates on the 3D shape S

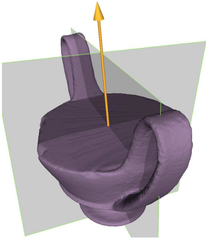
Determination of Symmetry Parameters

- Equivalent properties exist for:
  - Planar symmetries,
  - fixed-angle rotational symmetries

Last step is to check candidates on the 3D shape S

Testing a symmetry on a 3D Shape

- Define a Symmetry Measure
- Symmetries defined up to a threshold
- Allow approximate symmetries.

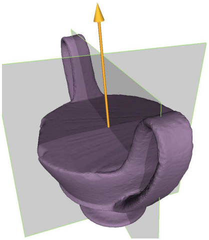


Fact

- *Testing a symmetry on S is costly.*
- *Symmetry Measure only computed for few candidates*

Testing a symmetry on a 3D Shape

- Define a Symmetry Measure
- Symmetries defined up to a threshold
- Allow approximate symmetries.

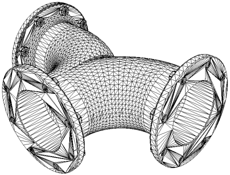


Fact

- *Testing a symmetry on $\mathcal{S}$ is costly.*
- *Symmetry Measure only computed for few candidates*

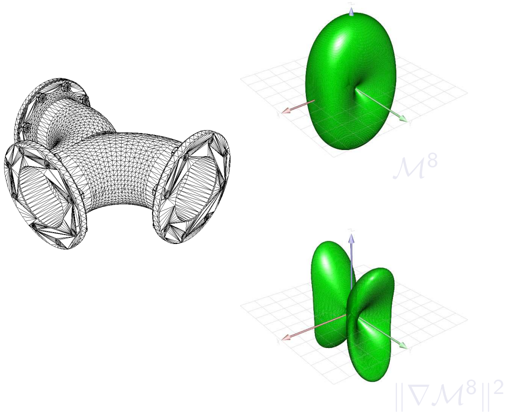
Results

Complete Example



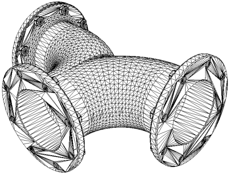
Results

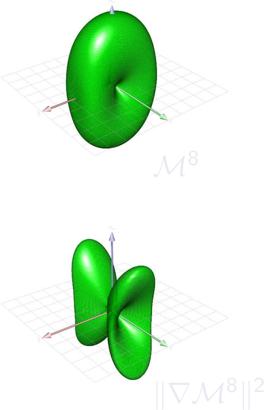
Complete Example

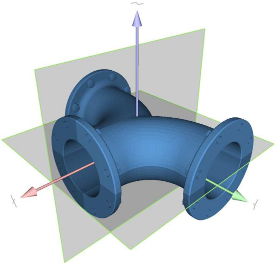


Results

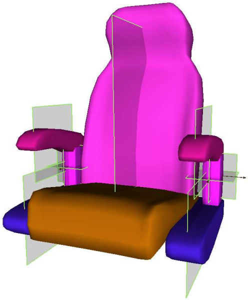
Complete Example







Summary



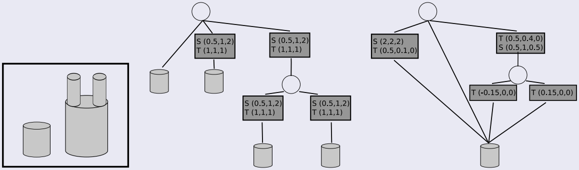
- Each tile (object) is structured using symmetry information
- Last step is to compute a representation of the geometry as a Hierarchy of Instances.

Hierarchical Instantiation

Definition

Instantiation

Factorize repeated geometry



Without Instancing

With Instancing

Hierarchical Instantiation

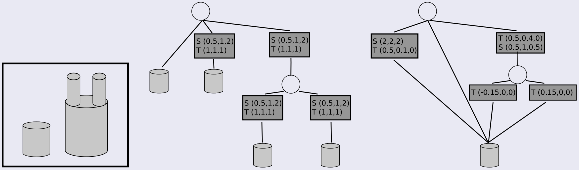
Extend Instantiation at multiple scales

Hierarchical Instantiation

Definition

Instantiation

Factorize repeated geometry



Without Instancing

With Instancing

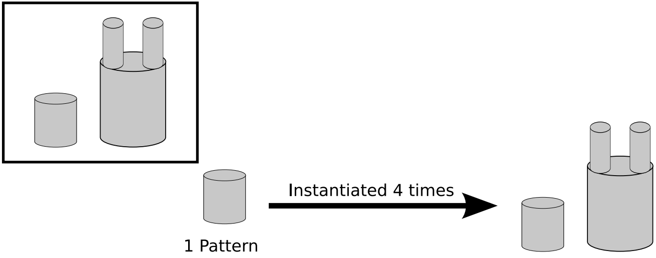
Hierarchical Instantiation

Extend Instantiation at multiple scales

Patterns and Instances

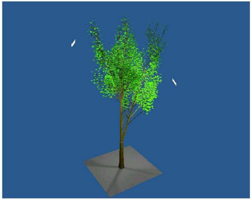
Definition

- *A pattern* is a generic set of objects,
- represented in the scene by *its instances*.
- Per instance attributes: Transformation matrix

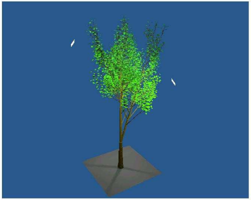


Utility of Instancing Information

- Rendering
- Geometry Editing



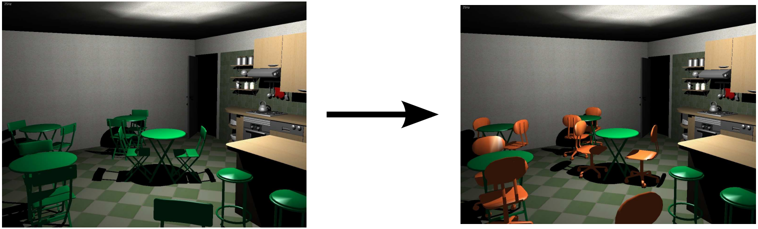
H.Radiosity (**119**mn/**123**MB)



H.Instantiation (**14**mn/**8**MB)

Utility of Instancing Information

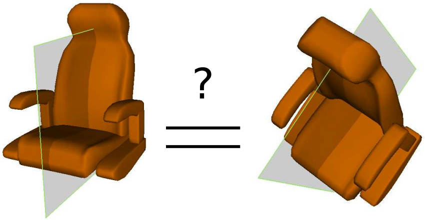
- Rendering
- Geometry Editing

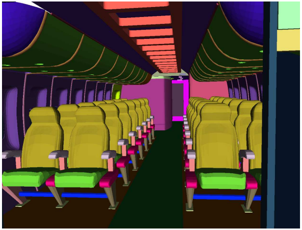


Overview

Input data

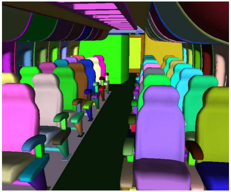
- Set of Tiles
- Class of Congruency
    - Congruent Descriptor (see manuscript)
    - Derived from $\mathcal{M}^{2p}$



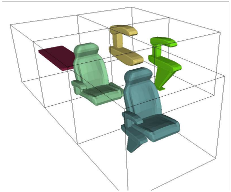


Overview

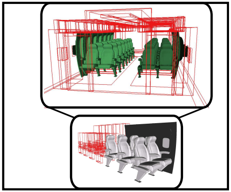
A two-step approach:



Structured 3D Geometry
Object Level



Frequent Patterns
Discovery



Organizing
Frequent Patterns

Problem Definition

Definition

The *frequency* of a pattern is equal to the number of its (possibly overlaping) instances

$\Downarrow$

Objective

Given a threshold f, identify all patterns P which frequency is greater or equal to f.

Problem Definition

Definition

The *frequency* of a pattern is equal to the number of its (possibly overlaping) instances

$\Downarrow$

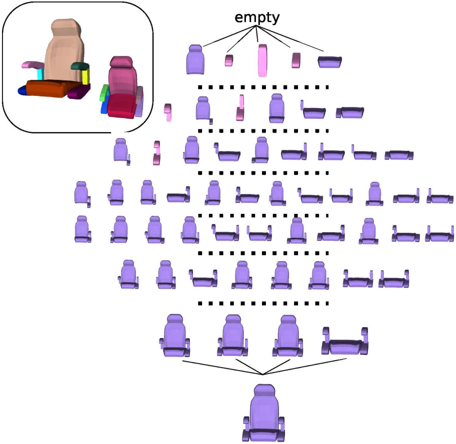
Objective

Given a threshold f, identify all patterns P which frequency is greater or equal to f.

Approach

Two potential approaches

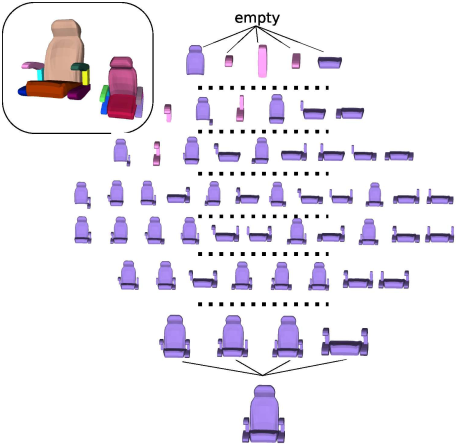
1. Agglomerative Approach
   - Progressively grow up a pattern
   - Efficient traversal of the search space
   - Exponential Complexity
2. Symmetry-Based Approach



Approach

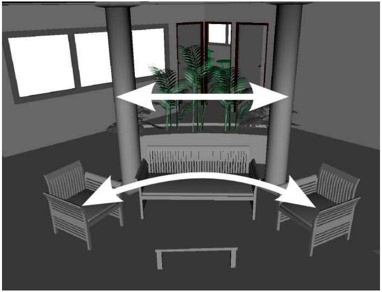
Two potential approaches

1. Agglomerative Approach
   - Progressively grow up a pattern
   - Efficient traversal of the search space
   - Exponential Complexity
2. Symmetry-Based Approach



Symmetry-Based Approach

Basic assumption



Fact

Two instances of a pattern form a **local symmetry**.

Symmetry-based approach

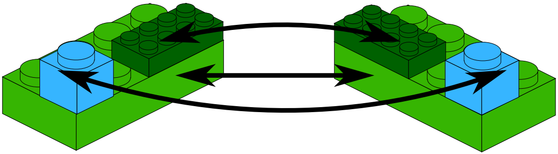
Strategy

Set of Frequent Patterns

$\Updownarrow$

Local Symmetries of the Scene

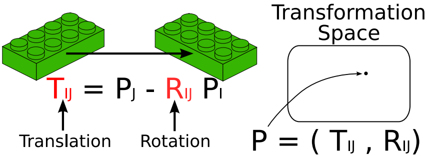
Symmetry-Based Approach



Overview

- Consider each couple of congruent tiles
- Compute the transformation that map one tile to the other
- Add the corresponding point in the Transformation Space

Symmetry-Based Approach



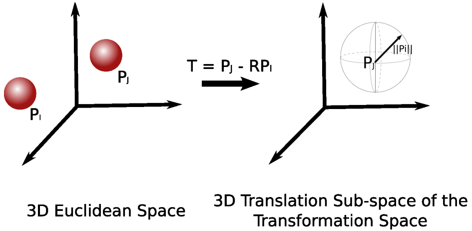
Overview

- Consider each couple of congruent tiles
- Compute the transformation that map one tile to the other
- Add the corresponding point in the Transformation Space

Symmetry-Based Approach

The Transformation Space

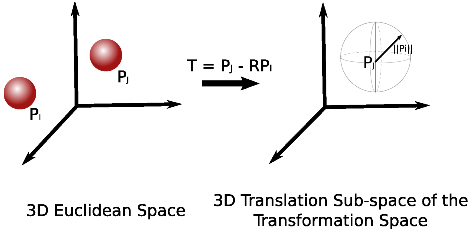
- For a couple of tiles, transformation is not unique:
  - Discrete symmetries *i.e.* rotation or planar symmetries.
  - continuous symmetries:
    - spherical symmetries,
    - cylindrical symmetries.



Symmetry-Based Approach

The Transformation Space

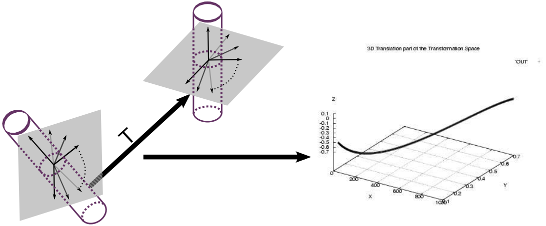
- For a couple of tiles, transformation is not unique:
  - Discrete symmetries *i.e.* rotation or planar symmetries.
  - continuous symmetries:
    - spherical symmetries,
    - cylindrical symmetries.



Symmetry-Based Approach

The Transformation Space

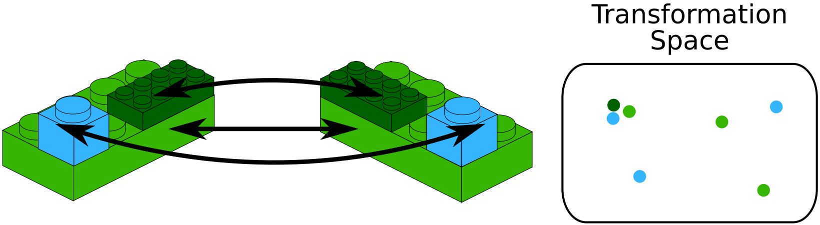
- For a couple of tiles, transformation is not unique:
  - Discrete symmetries *i.e.* rotation or planar symmetries.
  - continuous symmetries:
    - spherical symmetries,
    - cylindrical symmetries.



Symmetry-Based Approach

Forming patterns

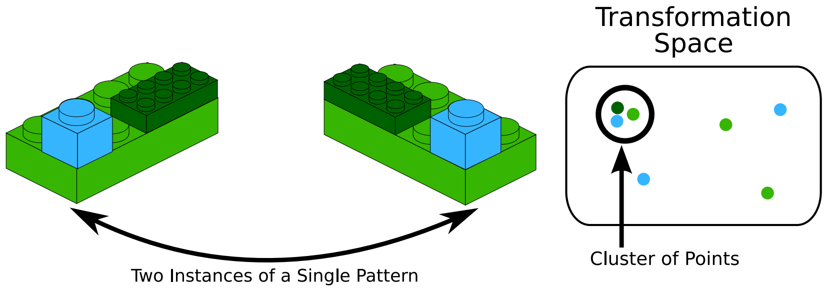
- Each point of the Transformation Space contains an information of mapping between two tiles,
- Local symmetries are obtained by *clustering points*



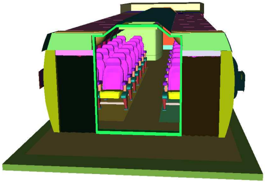
Symmetry-Based Approach

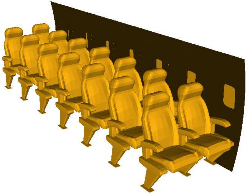
Forming patterns

- Each point of the Transformation Space contains an information of mapping between two tiles,
- Local symmetries are obtained by *clustering points*



Frequent Patterns: Results





# tiles = 154
Freq = 2

# tiles = 66
Freq = 4

# Tiles	480
Frequency Threshold	2
Runtime (secs)	1.4
# Frequent Patterns	65

Summary of Contributions

1. Identify the "hardness" of the problem
2. A new approach to generate frequent patterns
3. Not presented in this talk (see manuscript):
   - Analytic expression of curve equation for continuous symmetry
   - Method to reduce the number of mappings

Next step is to represent the scene as a Hierarchy of Instances

Summary of Contributions

1. Identify the "hardness" of the problem
2. A new approach to generate frequent patterns
3. Not presented in this talk (see manuscript):
   - Analytic expression of curve equation for continuous symmetry
   - Method to reduce the number of mappings

Next step is to represent the scene as a Hierarchy of Instances

Summary of Contributions

1. Identify the “hardness” of the problem
2. A new approach to generate frequent patterns
3. Not presented in this talk (see manuscript):
   - Analytic expression of curve equation for continuous symmetry
   - Method to reduce the number of mappings

Next step is to represent the scene as a Hierarchy of Instances

Summary of Contributions

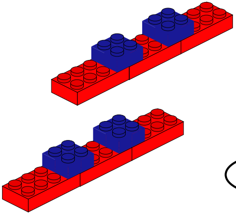
1. Identify the “hardness” of the problem
2. A new approach to generate frequent patterns
3. Not presented in this talk (see manuscript):
    - Analytic expression of curve equation for continuous symmetry
    - Method to reduce the number of mappings

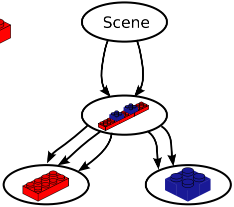
Next step is to represent the scene as a Hierarchy of Instances

Problem Statement

Goal

- Obtain a hierarchy of instances
- Represented as a **H**ierarchy **A**ssembly **G**raph **(HAG)**



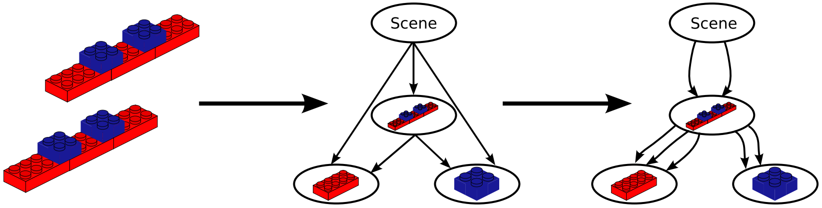


- Directed Acyclic Graph
- Each node is a pattern
- Each edge carries geometric transformation

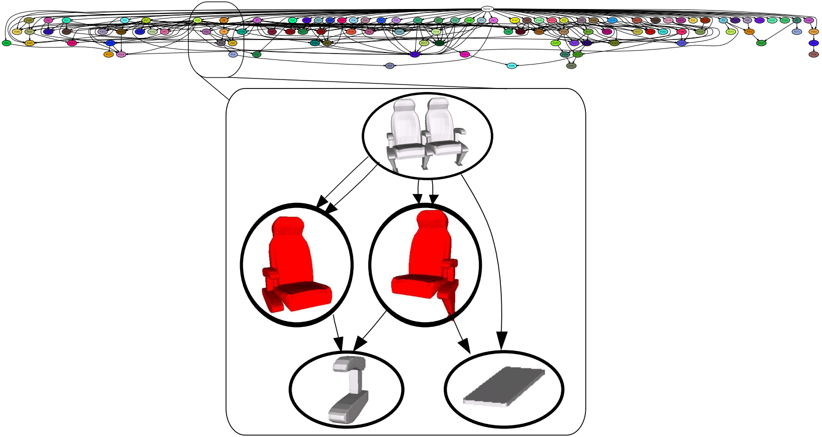
HAG Construction

HAG Construction

- Represent inclusion between frequent patterns
- Pick a *reference instance* for each pattern
- Compute the appropriate transform by iterating through edges



HAG Construction
Example: The Plane Model



HAG Construction

Observation

A HAG with overlap nodes is hardly usable

Overlapping Problems

1. Multiple rendering of overlaped parts
2. Inefficient for memory reduction
3. Geometry editing is much more difficult

HAG Construction

Observation

A HAG with overlap nodes is hardly usable

Overlapping Problems

1. Multiple rendering of overlaped parts
2. Inefficient for memory reduction
3. Geometry editing is much more difficult

Deriving a usable Hierarchy of Instances

- Usable Hierarchy : Hierarchy with no-overlap
- Some choices must be made
- This process is Application-Dependent

Example

- Hierarchy of Instances optimized for Ray-Tracing

Deriving a usable Hierarchy of Instances

- Usable Hierarchy : Hierarchy with no-overlap
- Some choices must be made
- This process is Application-Dependent

Example

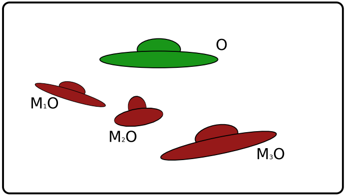
- Hierarchy of Instances optimized for Ray-Tracing

Deriving a usable Hierarchy of Instances

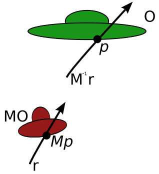
Ray-Tracing

Generalities

- Ray-Tracing naturally allows instancing
- Load a single pattern per instance
- Reduce part of the geometry loaded in memory



Simple Scene with instances



Ray-object intersection

Deriving a usable Hierarchy of Instances
Ray-Tracing

If considering whole scene itself as a Pattern of frequency 1:

Strategy

Hierarchy of Instances optimized for Ray-Tracing

$\Updownarrow$

Reduce storage cost $C(P)$ of each pattern P

Deriving a usable Hierarchy of Instances

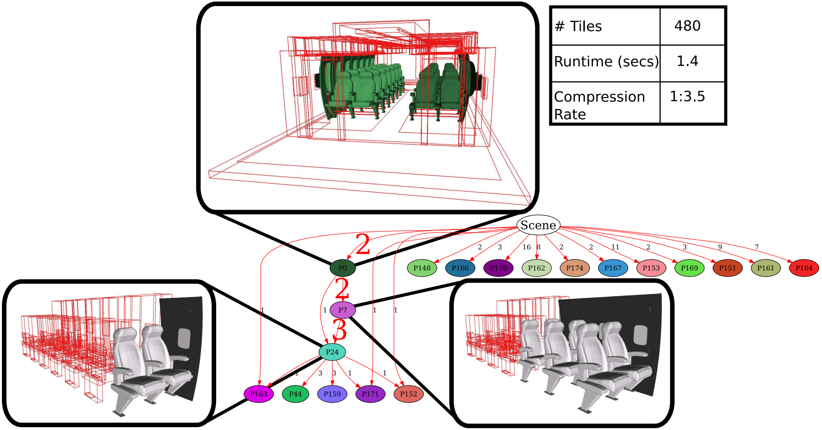
Ray-Tracing

Strategy

- Bottom-Up Approach
- For each pattern P:
  - $\min \mathrm{C}(P)$ constrained by non-overlap
- Such problem is $NP-$complete
- Need an approximation algorithm: greedy approach

Deriving a usable Hierarchy of Instances

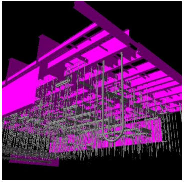
Example: Plane Model

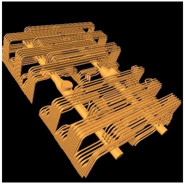


# Tiles	480
Runtime (secs)	1.4
Compression Rate	1:3.5

Deriving a usable Hierarchy of Instances

Example: Powerplant Model

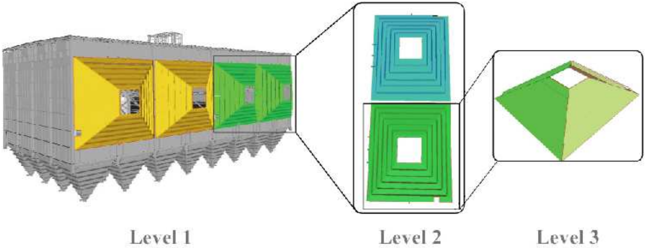




# Polygons	Pre-processing Geometry	# Tiles	Structuring Object Level (Parallel process)	Frequent Patterns	
				Runtime	# Patterns
12,748,510	2 minutes	155,348	Approx. 16 hours	1h45	87,100

Deriving a usable Hierarchy of Instances

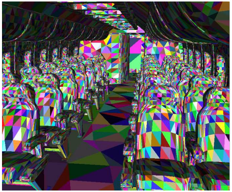
Example: Powerplant Model



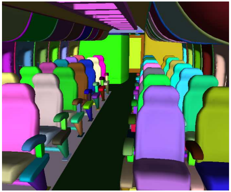
Runtime	Compression Rate
51 seconds	1:5.2

Summary

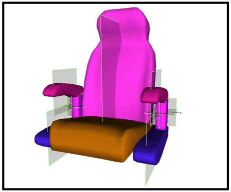
- A whole pipeline for structuring 3D Geometry:



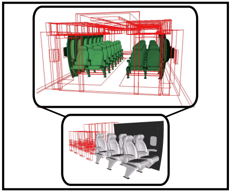
Unstructured 3D Geometry



Pre-Processing 3D Geometry



Object Level: Symmetry



Scene Level: Instancing

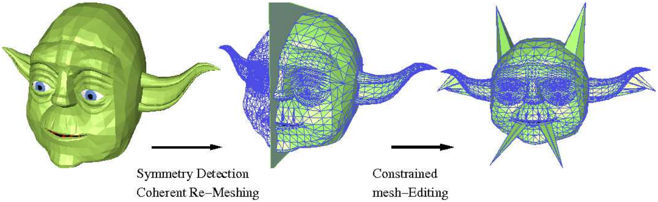
- Potential Applications:
  - Rendering
  - Geometry Editing

Summary of Contributions

Two contributions for Structuring 3D Geometry:

Detection of Symmetries in 3D Shapes

- The Generalized Moment Functions
- Algorithms for Single and Composite Shapes
- Potential Applications: Compression, Geometry Editing, . . .

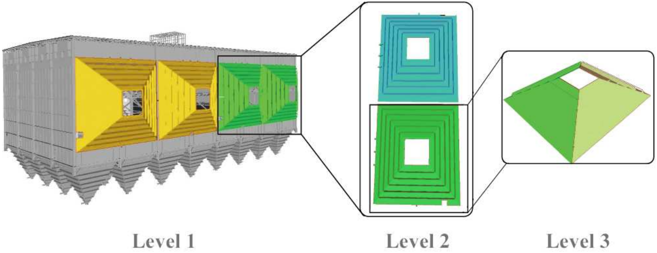


Summary of Contributions

Two contributions for Structuring 3D Geometry:

Hierarchical Instancing of Geometry

- A way of representing geometry as a Hierarchy of Instances
- Potential Applications: Geometry Editing, Compression, rendering, . . .

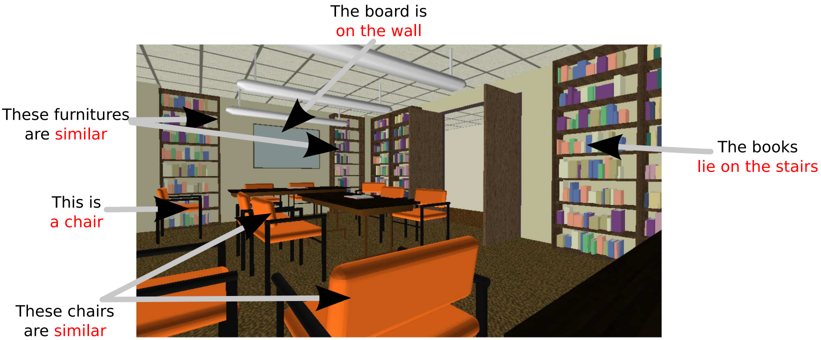


Future Work

- Most Promising Work: Structuring at the Semantic Level
- Adaptive Display Algorithm [Funkhouser et al. 93]
  - Adapt geometry to render it at interactive frame rates

Future Work

- Most Promising Work: Structuring at the Semantic Level
- Adaptive Display Algorithm [Funkhouser et al. 93]
  - Adapt geometry to render it at interactive frame rates



Thank you for your attention