Computer Graphics and Constraint Solving: An Application to Virtual Camera Control

Marc Christie

LINA - Laboratoire d'Informatique de Nantes Atlantique
FRE CNRS 2729
2, rue de la Houssinière,
F-44322 Nantes Cedex 3 - France
marc.christie@lina.univ-nantes.fr

FJCP 2005

Outline

Computer Graphics & Constraint Solving

- For the best
- And for the worst ...

Semantic Space Partitioning for Virtual Camera Control

- A Declarative Approach
- Computing Semantic Volumes
- Computing *best* representatives

Discussion

- Future Work
- Conclusion

Motivations

Computer Graphics

- generally manage complex geometric shapes
- with complex relations between the shapes

Naturally one wants to easily set and maintain the relations between the shapes

Constraint Solving

- offers a natural declarative modelling framework
- offers a broad set of solving methods

Is this a happy marriage?

Yes

- provides a high-level and natural interaction process
- is extensible through the flexibility of the modelling framework

... well, to some extend...

- what happens when the system is inconsistent (best approximation?)
- how is the user informed of the satisfied properties (explanations?)
- what if the system has multiple solutions/classes of solutions?
- semantics is generally lost in the solving and interaction process

Help? The answer is in communication!

- why not maintain semantics in modelling, solving and interaction ?

What is missing?

Maintaining semantics in the different steps:

- Modelling
- Solving
- Interaction

We present an application in Virtual Camera Control

- Identifying and computing possible classes of solutions
- Interacting and reasonning with these classes of solutions

What is the problem?

Motivations

Positioning a camera in a virtual 3D world is not an intuitive task:

- only a 2 *d.o.f* input device to control a 7 *d.o.f* camera;
- mental inversion process to derive the coordinates of the camera from some desired properties on the screen;
- tedious and time consuming process (position the camera, check the result)
- lack of tools that provide high level approaches to manipulate the camera.

Cinematography

Cinematography provides a large consensus on a grammar of the film language [Arijon]

- on screen layout (relative locations of objects on the screen)
- scale shots (close-up, long shot)
- vantage points (high or low angle)

A Semantic Space Partitioning Approach

Proposition

Offer a tool to explore all the cinematographic possibilites of a scene:

- maintain semantics in all steps of the process (description, solving, interaction);
- compute and propose different classes of solutions to the user.

How?

- Step 1: split the world of possible camera locations into areas that share similar cinematographic caracteristics (can be related to [Bares, Pickering]);
- Step 2: intersect the areas corresponding to the user's description;
- Step 3: compute a *(good)* camera configuration representative of each possible area.

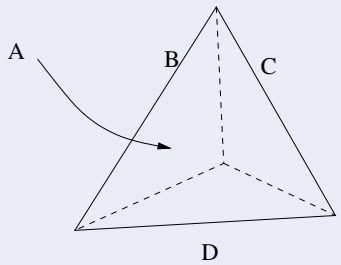
Step 1: Semantic Space Partitioning

The semantic partitioning is based on an extension of visual aspects:

Visual Aspects

Introduced by J. Koenderink and J. van Doorn, they regroup viewpoints that :

- share the same topological properties,
- *w.r.t.* a single object.

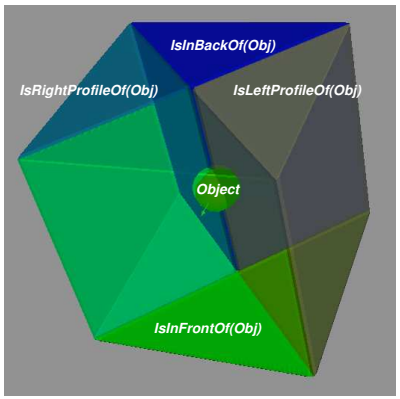


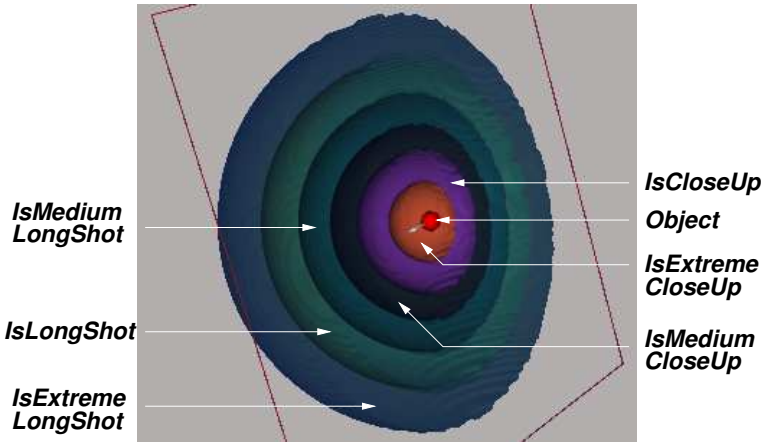
The Semantic Volumes: An extension of Visual Aspects to Virtual Camera Control

- transpose the topological information into cinematographic properties;
- extend the space characterization to multiple objects.

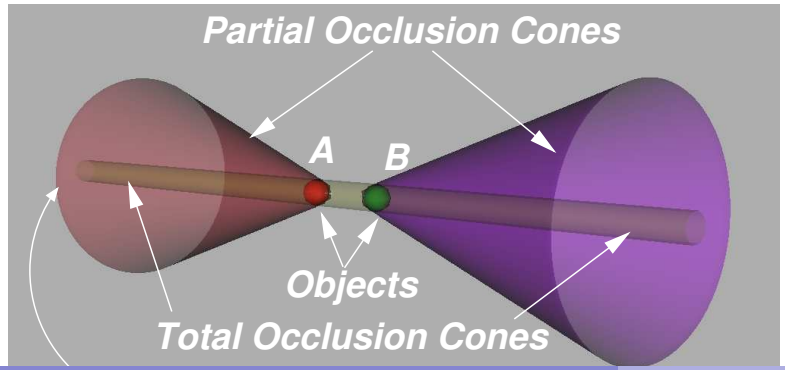
The Semantic Volumes

- we introduce the notion of *Semantic Volumes* as a volume of possible camera locations (*i.e.* in $\mathbb{R}^3$) that gives rise to qualitatively equivalent shots;
- each *semantic volume* is characterized by a set of semantic tags from the *"Grammar of the Film Language"*[Arijon]:
  - a single object (*e.g.* viewing angles, scale shots distances, . . .),





  - two or more objects (occlusions, relative positions of objects in the image).



How to compute the semantic volumes ?

Geometric Filtering Operator G_f:

G_f computes the semantic volume s_v attached to a property p

- G_f is complete in that it does not lose any correct camera placement;
- every camera location inside s_v possibly satisfies p;
- every camera location outside s_v certainly violates p.

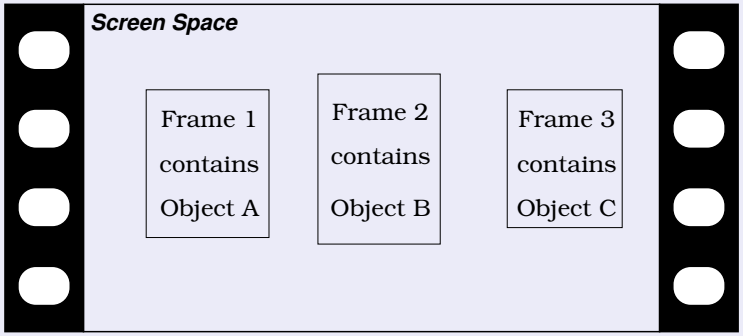
Properties

- scale shot property: *ExtremeCloseUp*(A) to *ExtremeLongShot*(A);
- orientation property: *IsLeftProfileOf*(A), *IsRightProfileOf*(A) up to *IsThreeQuaterBackRight*(A);
- occlusion property: *Occlusion*$(A, B, Total)$, *Occlusion*$(A, B, Partial)$, *NoOcclusion*(A) ;
- relative screen locations: *IsLeftOf*(A, B), *IsRightOf*(A, B), *IsAboveOf*(A, B), *IsUnderOf*(A, B);
- framing property: *Frame*$(A, x1, x2, y1, y2)$
  - requires a specific study (see next slide)

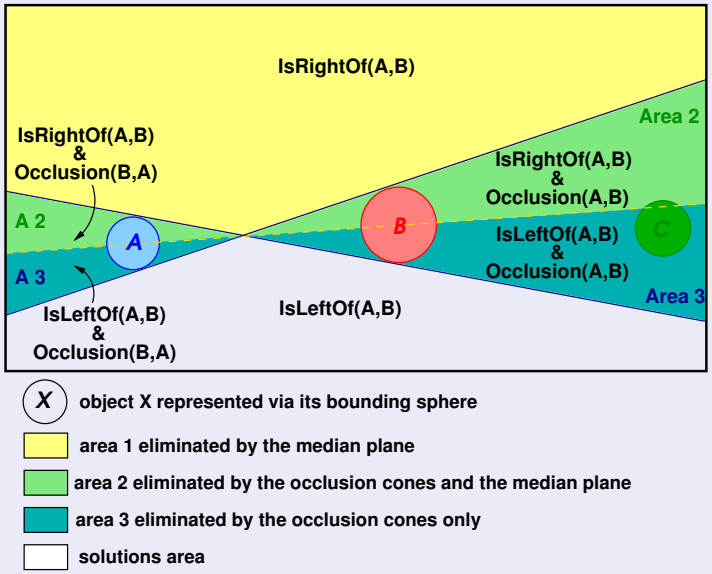
Framing Property: Case #1

Non overlapping Framing Configuration

- User input:



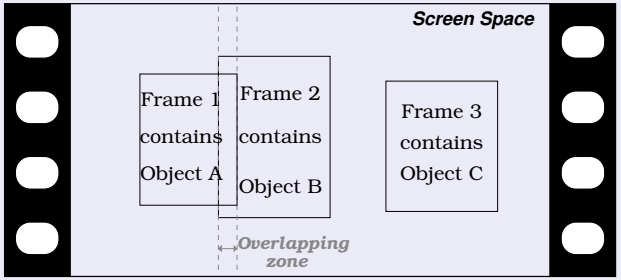
- Semantic volumes (2D overview representation):



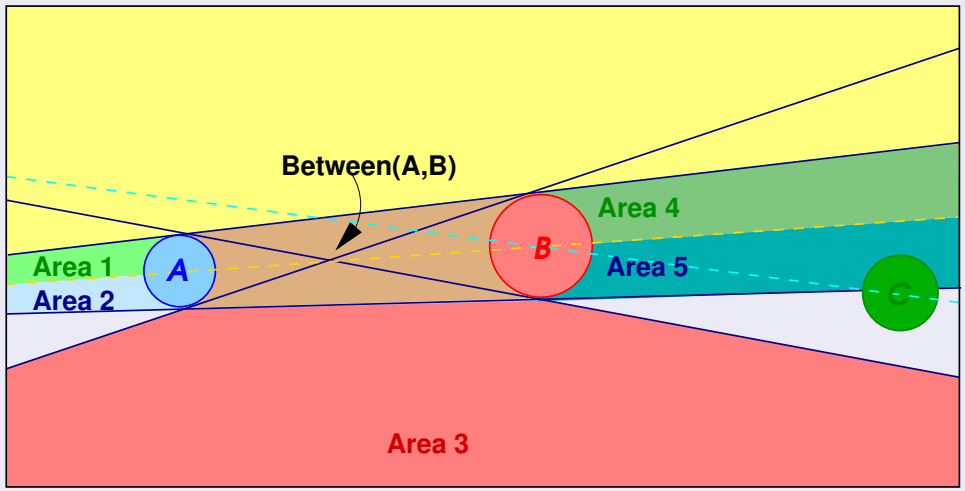
Framing Property: Case #2

Overlapping Framing Configuration

- User input:



- Semantic volumes (2D overview representation):



area 1 : IsRightOf(A,B) & TotalOcclusion(A,B)

area 2 : IsLeftOf(A,B) & TotalOcclusion(A,B)

area 3 : IsLeftOf(A,B) & NoOcclusion(A,B)

area 4 : IsRightOf(A,B) & TotalOcclusion(B,A)

area 5 : IsLeftOf(A,B) & TotalOcclusion(A,B)

solutions area

Step 2: Intersection of Semantic Volumes with implicit functions

Definition

An implicit surface is a function defined given a potential field: $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, as the set of 3D points $P(x, y, z)$ such that $P(x, y, z) = c$, where c is a constant (iso-value). By convention, we call interiors the points P such that : $f(P) < c$.

Implementation

- each semantic volume is represented by an implicit function
- intersection:

$$V_1 \cap V_2 = f_{V_1 \cap V_2}(x, y, z) = max(f_1(x, y, z), f_2(x, y, z))$$

Tesselation

- only necessary to count the number of non-connected volumes
- each non-connected volume represents a new class of solutions
- if the volume is empty, the problem has no solutions

Step 3: Computing the best representative

Objectives

For each distinct volume, we want to solve:

$$\begin{cases} \min \sum_i \text{cost}_{p_i}(c) \\ \forall j, f_j(c) \text{ is satisfied} \\ c \in \mathcal{V} \end{cases}$$

where $\text{cost}_{p_i}(c)$ stands for the function cost associated to property p_i, and where f_j is the j-th framing property given by the user.

Approach

We propose a stochastic interval-based Local Search method looking for a camera configuration:

- best satisfies the constraints;
- minimizes the cost functions;
- belongs to the semantic volume.

Local Search Framework

Principle

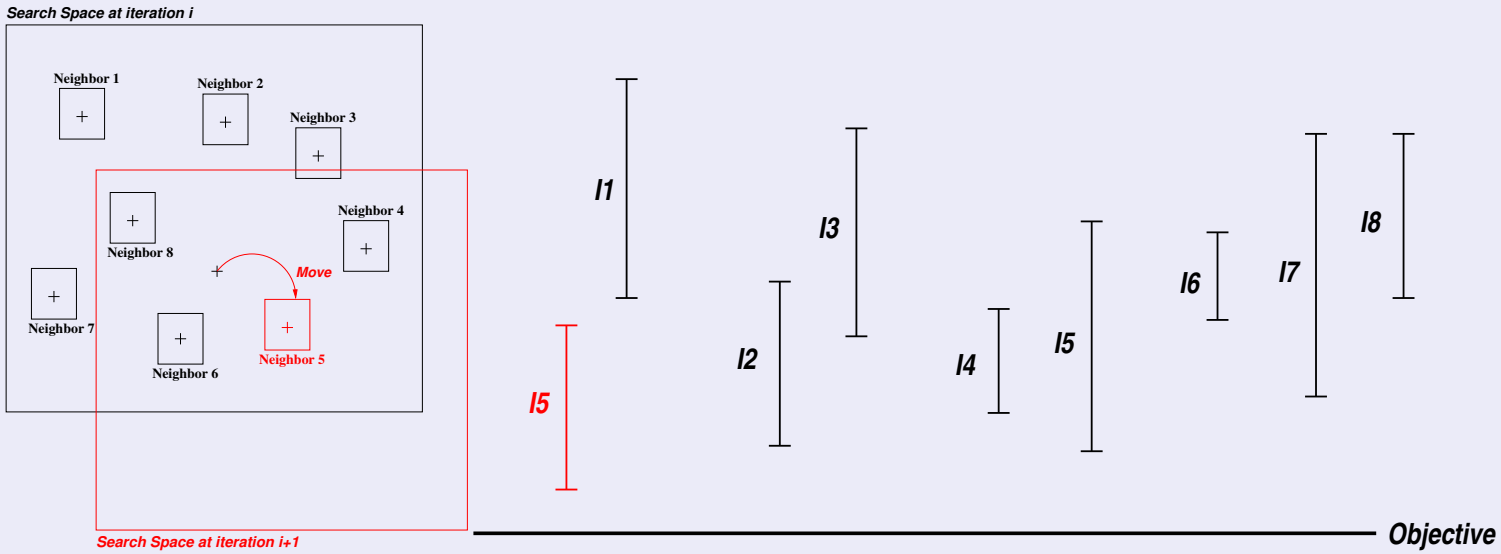
1. choose an initial configuration
2. generate a set of *neighbors*, evaluate them and choose the best one
3. move to the best neighbor
4. start back in 2 // *intensification dives towards a solution*
5. start back in 1 // *diversification jumps out local minima*

An interval-based extension

- a configuration is an Cartesian Product ($\mathbb{R}^7$) of floating-point intervals (namely a box)
- generate possible boxes as neighbors of the current configuration,
- evaluate each box and choose the best one in terms of interval comparison,
- set this best neighbor as the current configuration.

An Interval-based Local Search Algorithm

Illustration



Why rely on interval arithmetics?

- exploration of the search space with subregions (boxes) is more efficient than a floating-point exploration
- interval arithmetic containment property [Moore66] guarantees that the evaluation of a box contains the evaluation of every point in the box:
  - each classical operator $(+, -, \times, \sin, \cos, \ldots)$ is replaced by its interval representation
    $[a, b] + [c, d] = [a + c, b + d]$
    $[a, b] \times [c, d] = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)]$
  - containment property: $F(X_1, \ldots, X_n) \supseteq f(x_1, \ldots, x_n)$

Results

The over-the-shoulder Shot

```
Frame $objA -0.8 -0.3 -1 1 #Frame A on the left
Frame $objB 0.3 0.8 -1 1 #Frame B on the right
Orientation $objA BackRightThreeQuarter
Orientation $objB Front
Projection $objA LongShot
Projection $objB ExtremeLongShot
```

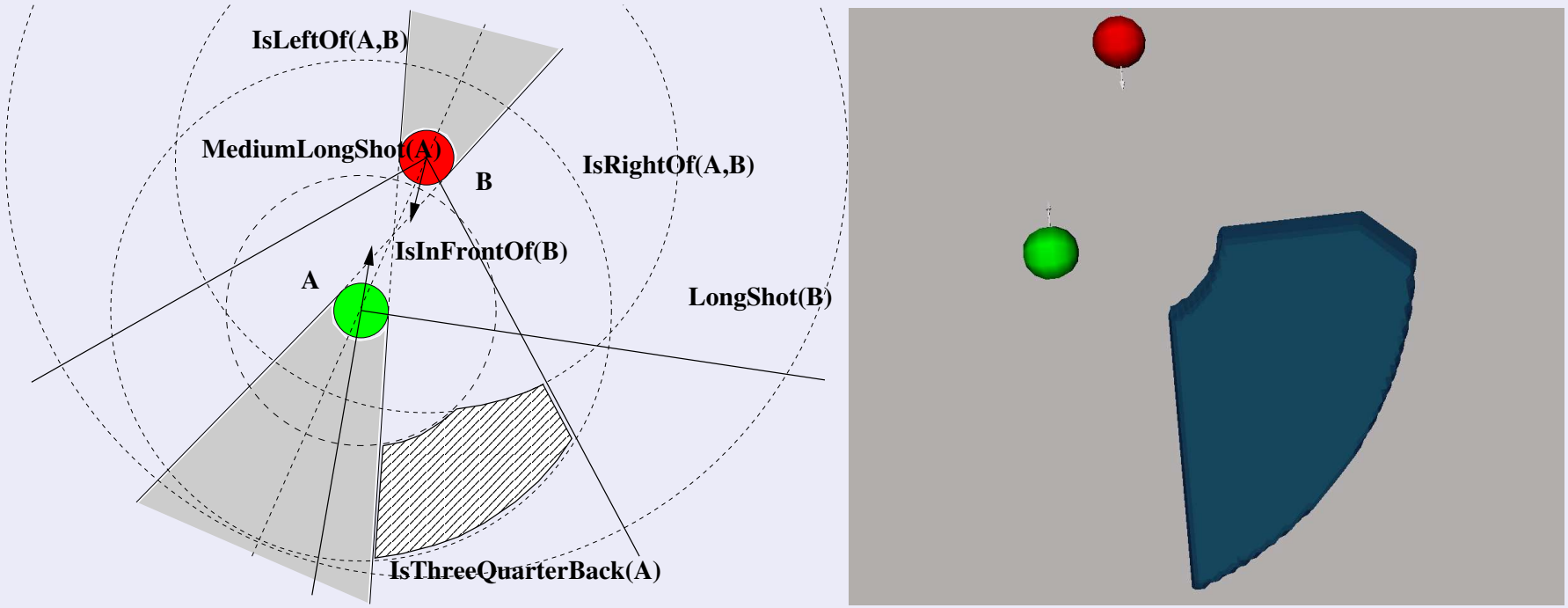


Figure: Top view of the search space (left) and tessellation of the implicit volume (right).

Results

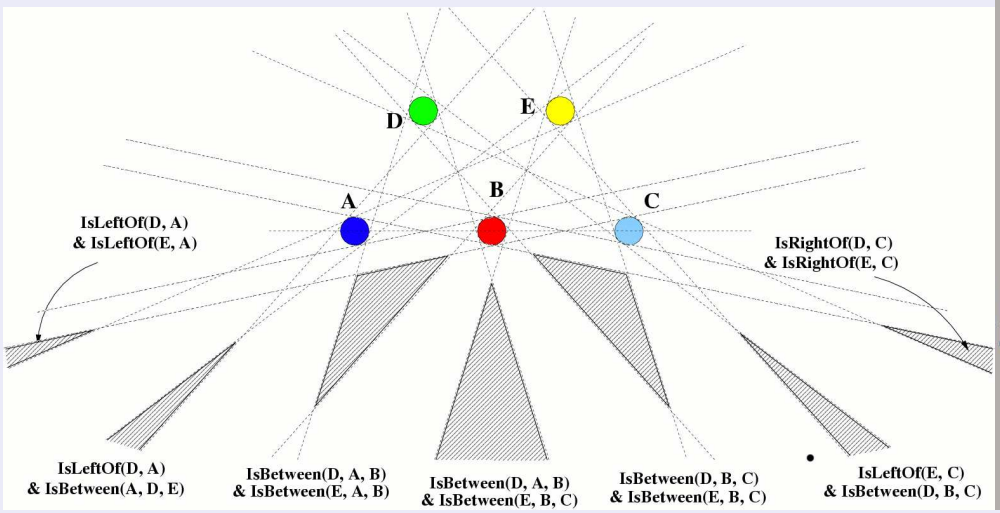
The over-the-shoulder Shot

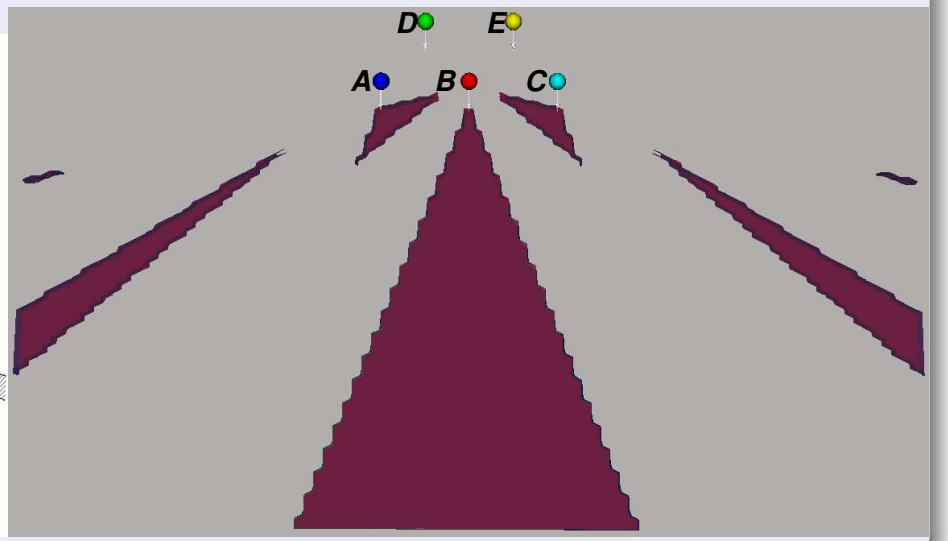
Computation of distinct components	0.21s
Nb Marching cubes	25000
Nb Polygons	240
Nb components	1
Local search time	0.60s
Nb Steps	25
Nb Tries	25

Results

The Five Framing Shot

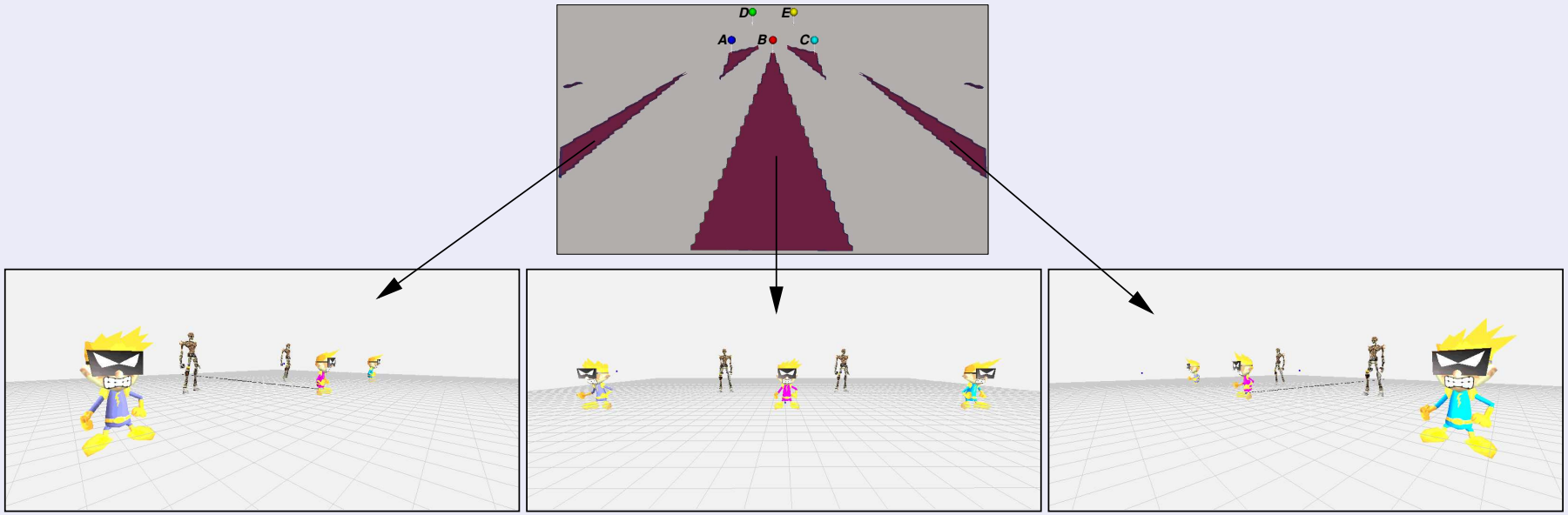
```
Frame $objA -1.0 -0.33 -1.0 1.0 #Frame A on the left third
Frame $objB -0.33 0.33 -1.0 1.0 #Frame B on the middle third
Frame $objC 0.33 1.0 -1.0 1.0 #Frame C on the third right
Frame $objD -1.0 1.0 -1.0 1.0 #Frame D on the screen
Frame $objE -1.0 1.0 -1.0 1.0 #Frame E on the screen
NoOcclusion $objD
NoOcclusion $objE
```





Results

The Five Framing Shot



Computation of distinct components	0.53s
Nb Marching cubes	25000
Nb Polygons	4036
Nb components	7
Local search time (per shot)	0.67s
Nb Steps	25
Nb Tries	25

Exploitation of the Semantic Volumes

The exploitation of the *semantic volumes* offers different possible interactions:

- making requests on the *semantic volumes*: the user can ask for the differences between two or more *semantic volumes* in term of properties he has not yet considered.
- making requests on a single *semantic volume*: the user can ask for a more precise characterization of a *semantic volume*, by requesting additional information: *e.g. " Which are the possible scale shots for this class of solution?"*

Future Work

Perspectives

- improve the occlusion model
  - integrate shadow volumes?
  - integrate hardware rendering techniques?
  - enhance expressivity (*e.g.* see a character through the leaves of a tree).
- provide an intuitive interface for interacting with the semantic volumes:
  - interaction through a sketch-based interface with high-level manipulation;
  - interaction with the tree of possible semantic volumes.
- extend the expressivity (*e.g.* perspective lines, construction shapes, lighting).
- extend the approach to camera animation.

Conclusion

Semantic Space Partionning:

- Identifies and characterizes possible classes of solutions
- Computes a *nice* representative of each class
- Maintains the semantics in the whole process (modelling, solving, interaction)
  - Offers a high-level interaction framework
  - Brings the interaction process in the semantics field
  - Provides a good undestanding of the possibilities in a scene

Semantics

- can bring a solution to a number of limitations in current applications
  - high-level interaction
  - solution characterization