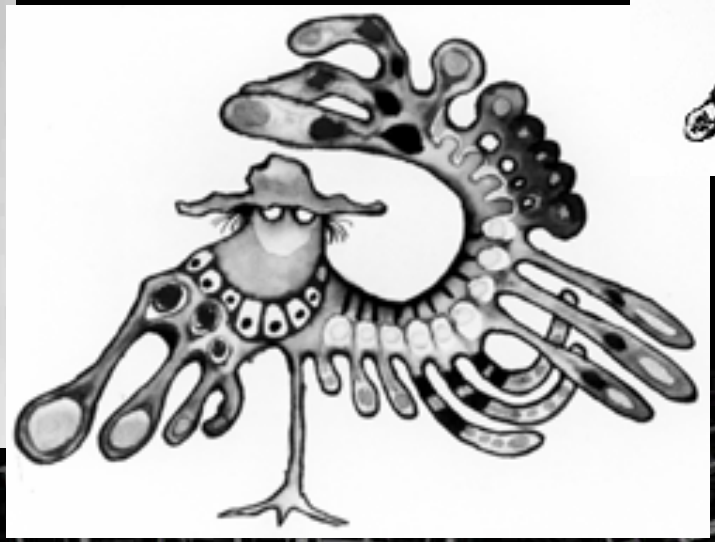
Psychorealism: Artist driven interactive graphics

Karan Singh
Dynamic Graphics Project
Computer Science University of Toronto

Ryan Larkin



*

Ryan Larkin

*

Psychorealism

The realism of the human psyche, expressed through the medium of art and animation.

Guernica. Pablo Picasso, 1937

Isabel Rawsthorne. Francis Bacon, 1965

Self. Ivan Albright, 1934

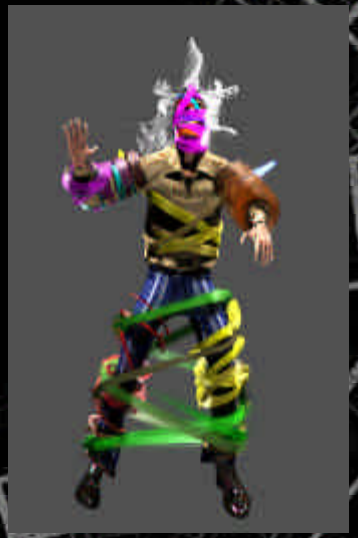
Psychorealism and *Ryan*

*

Overview

- Nonlinear projection

- Cords: Geometry with physical attributes



- *Ryan*

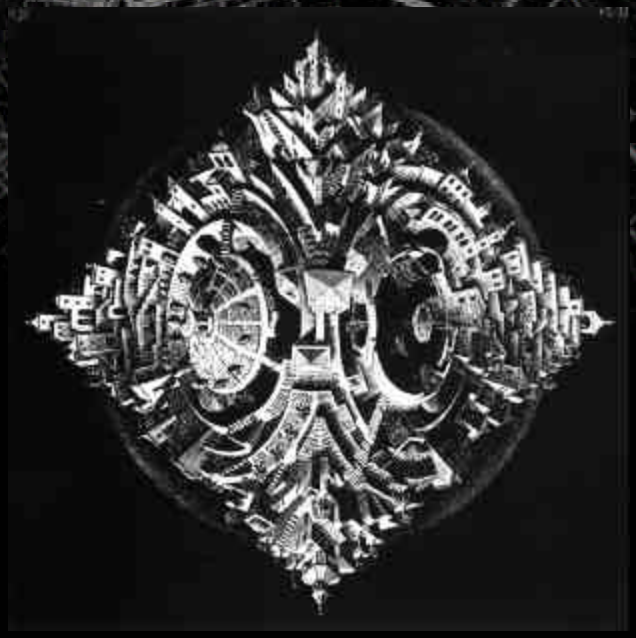
Projection

Linear perspective

- Good approximation of human visual system
- Conceptually simple and predictable
- Aids depth perception
- Efficient graphics pipeline

Motivation

Femme nue accroupie Pablo Picasso,1959



Tetrahedral Planetoid, M. C. Escher

Pearblossom Hwy. No. 2, David Hockney, 1986

*

Nonlinear projection

- Extend visual range
- Avoid disjoint images for complex scenes
- Artistic expression

Nonlinear projection model

- *C,M,V* are the eye-space and perspective, viewport matrices for a linear perspective camera.

- A point in the scene *P* thus linearly projects under the camera to <*x,y*> in the image at depth *z* where, <*x,y,z*>=*PCMV.*
- Lets call *CMV, E.*

Master and lackey cameras

For *P'* to appear in master camera *b,* as *P* appears in lackey camera *i :*

$?E_b = P\,E_i$

$$P' = PC_iM_iV_i(C_bM_bV_b)^{-1}.$$

$$A_i = C_iM_iV_i(C_bM_bV_b)^{-1}$$

Master and lackey cameras

Given weight w_{iP} for lackey camera *i* point *P* deforms to *P' :*

$$P' = P + P(w_{iP}(A_i - I))$$

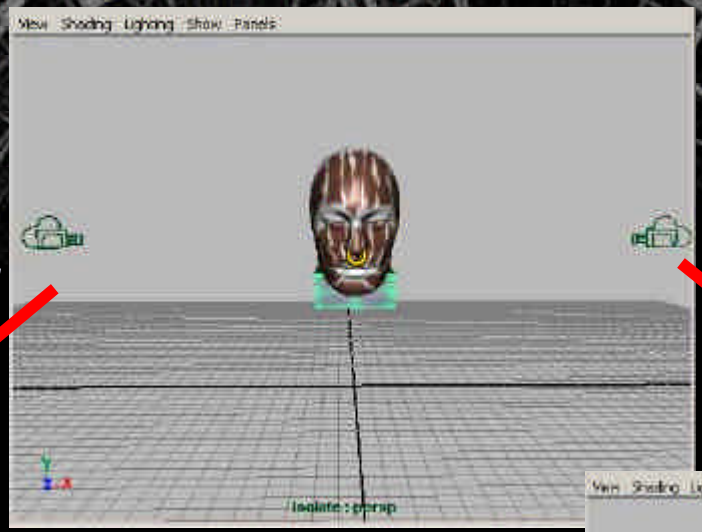
…and for many lackey cameras

$$P' = P + \sum_{i=1}^{n} P(w_{iP}(A_i - I)).$$

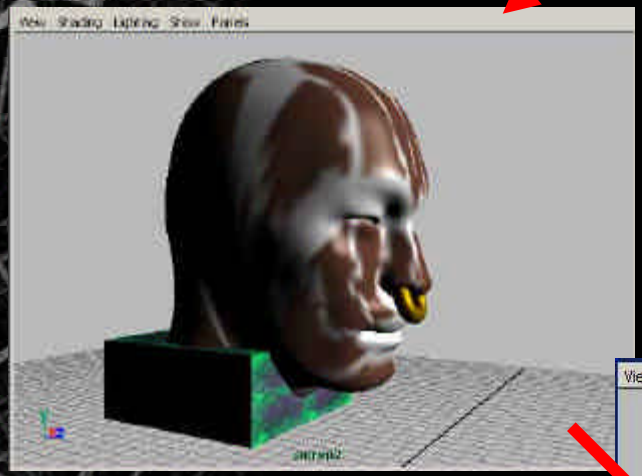
Interface

master

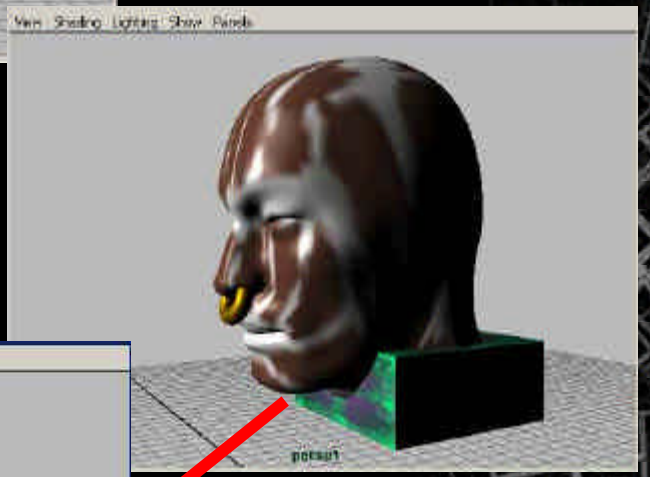
lackey



Exploratory View



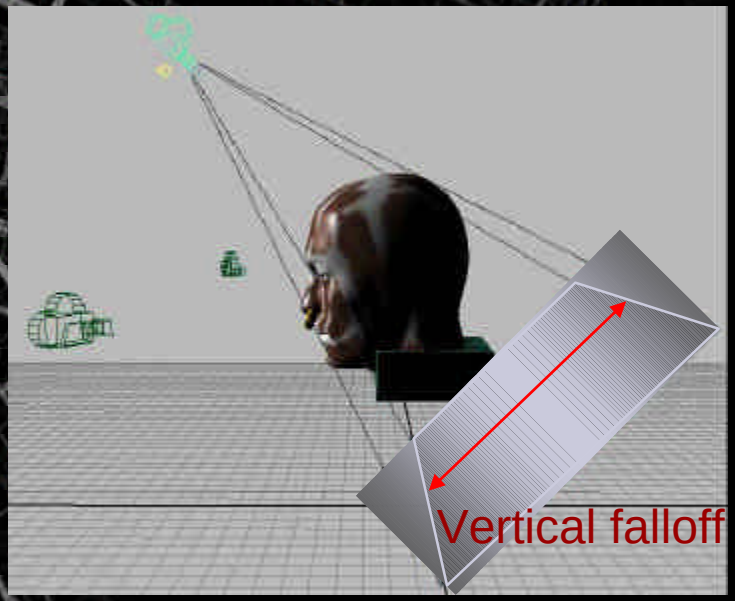
master **Linear Perspective**



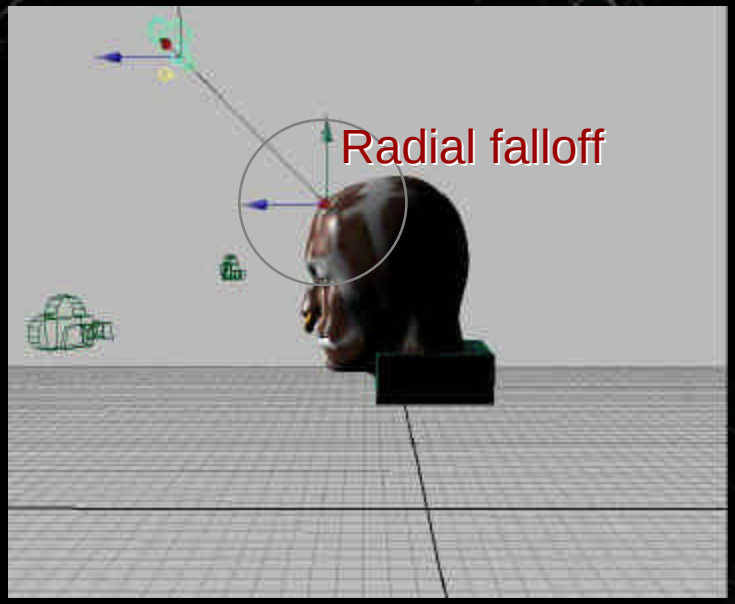
lackey **Linear Perspective**

Nonlinear Projection

Defining projection weights

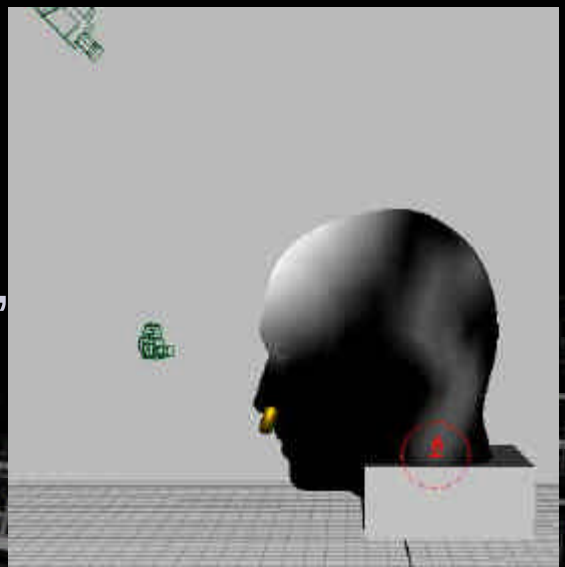


Directional



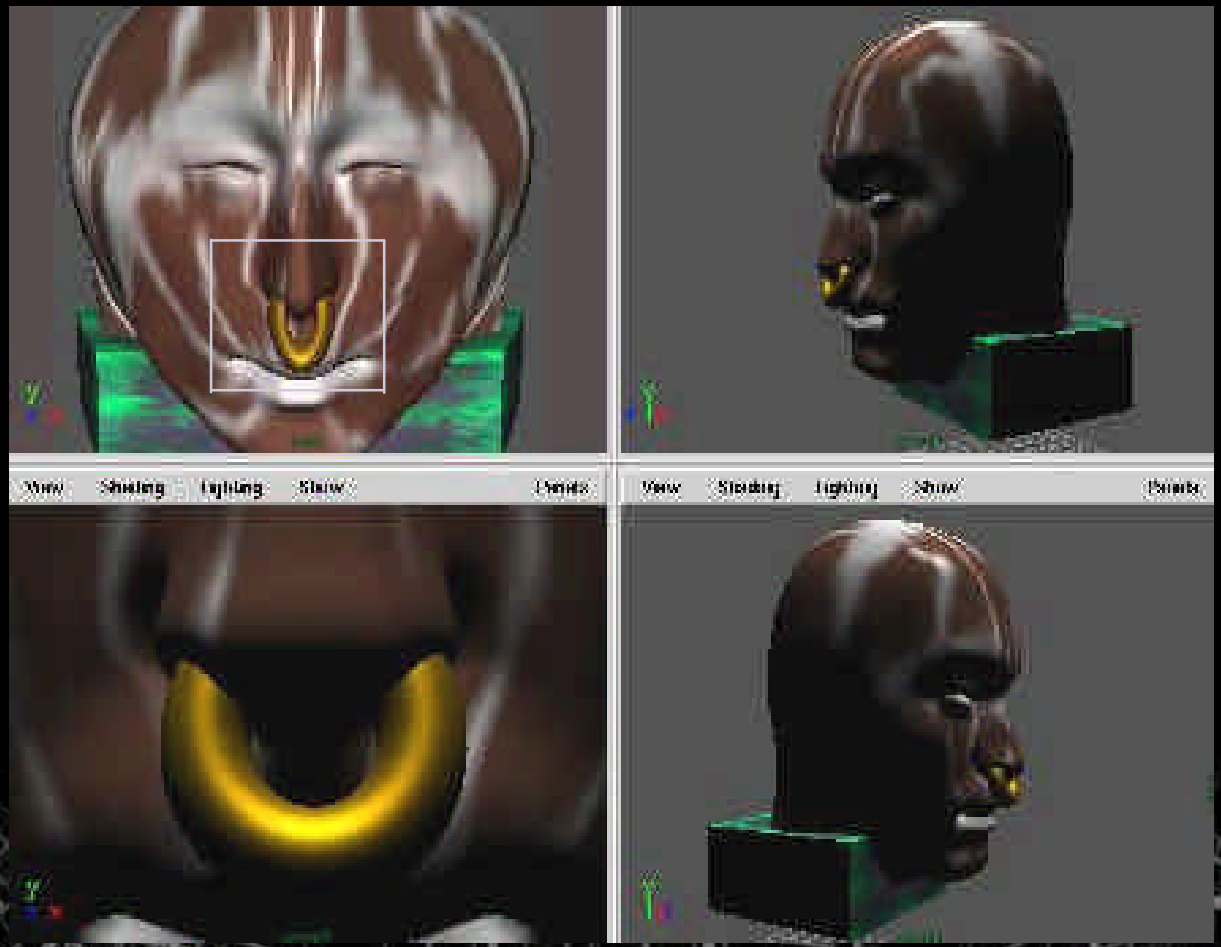
Positional

Feature based,
User Painted



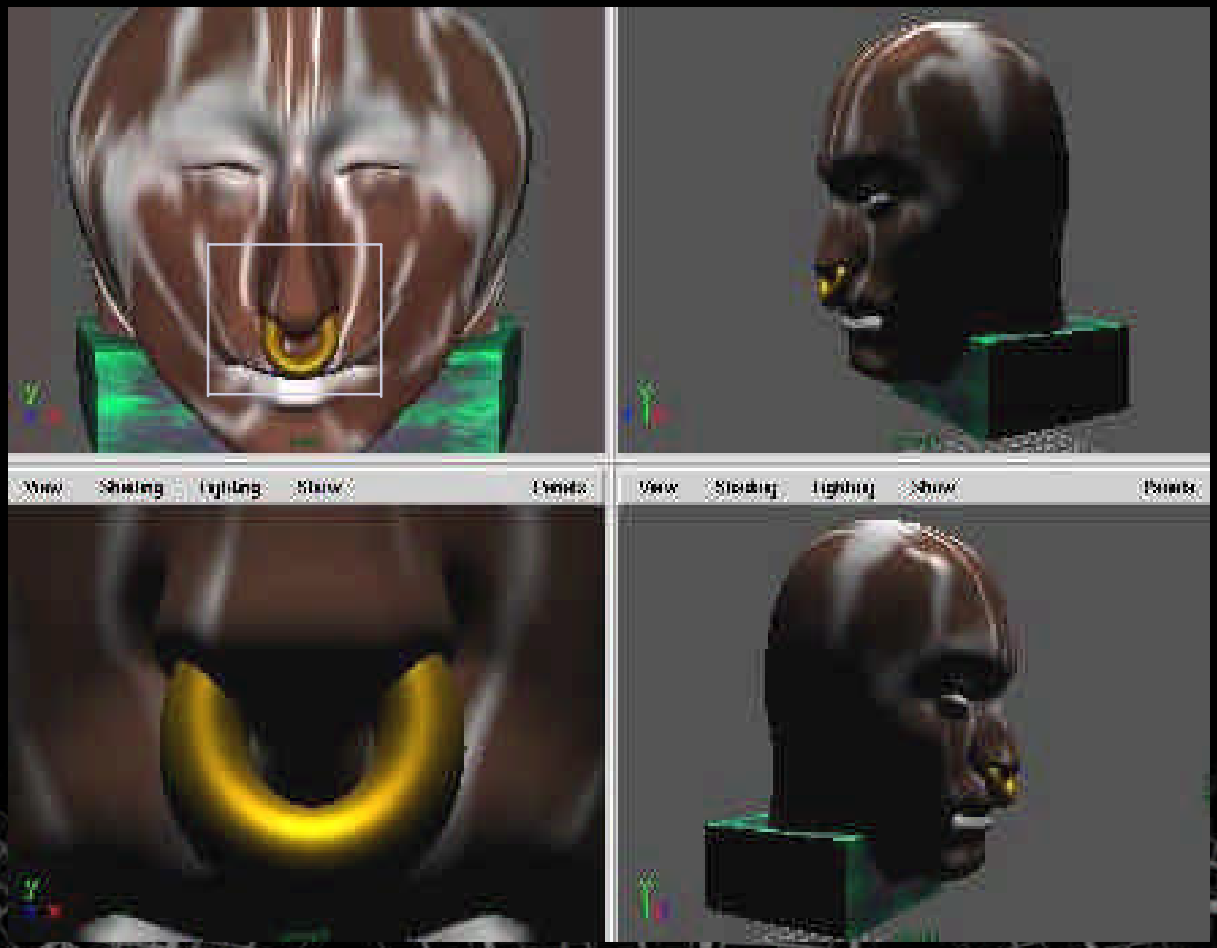
Examples

Positional Control



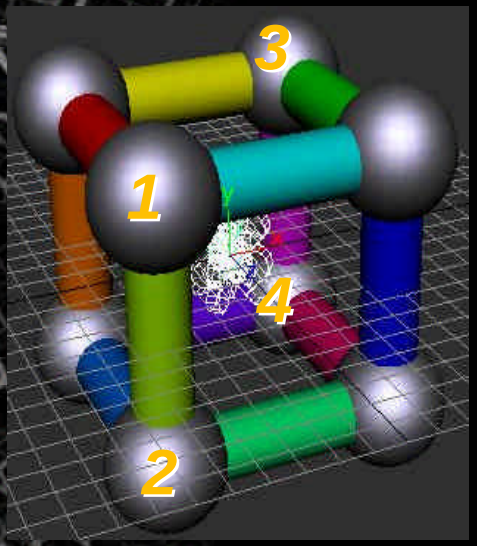
Examples

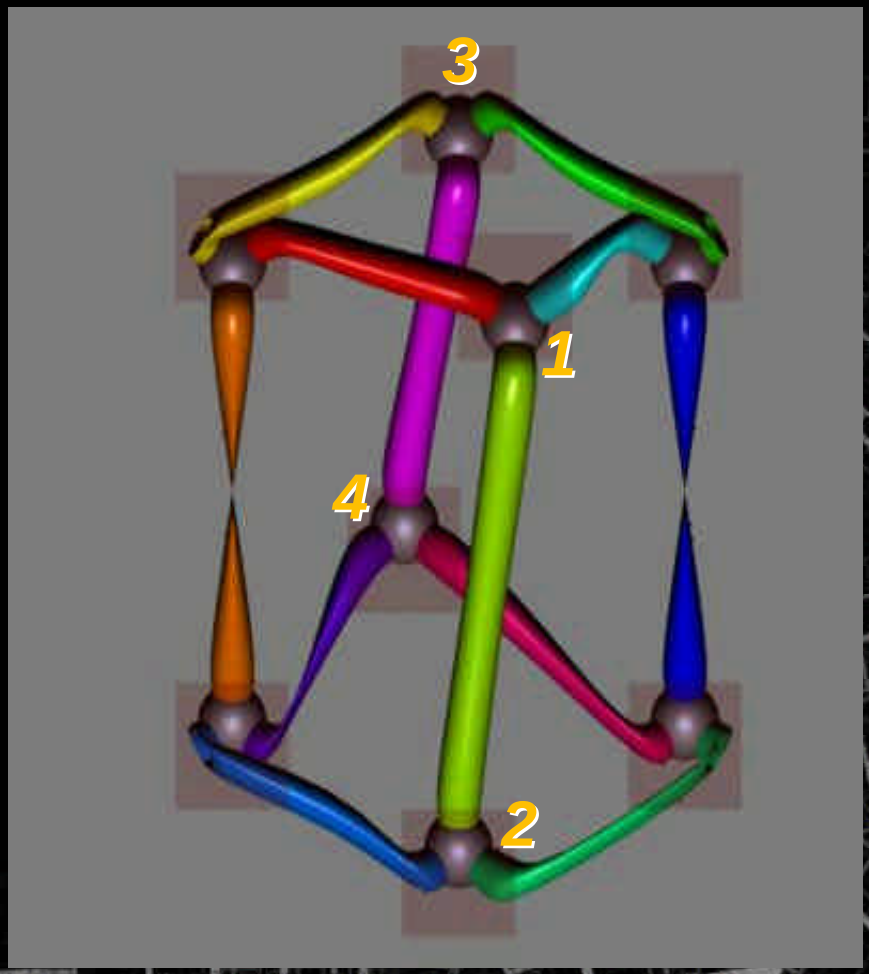
Positional Control



Examples

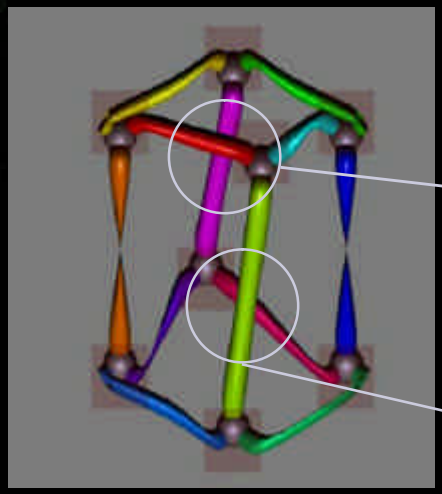
Depth Control

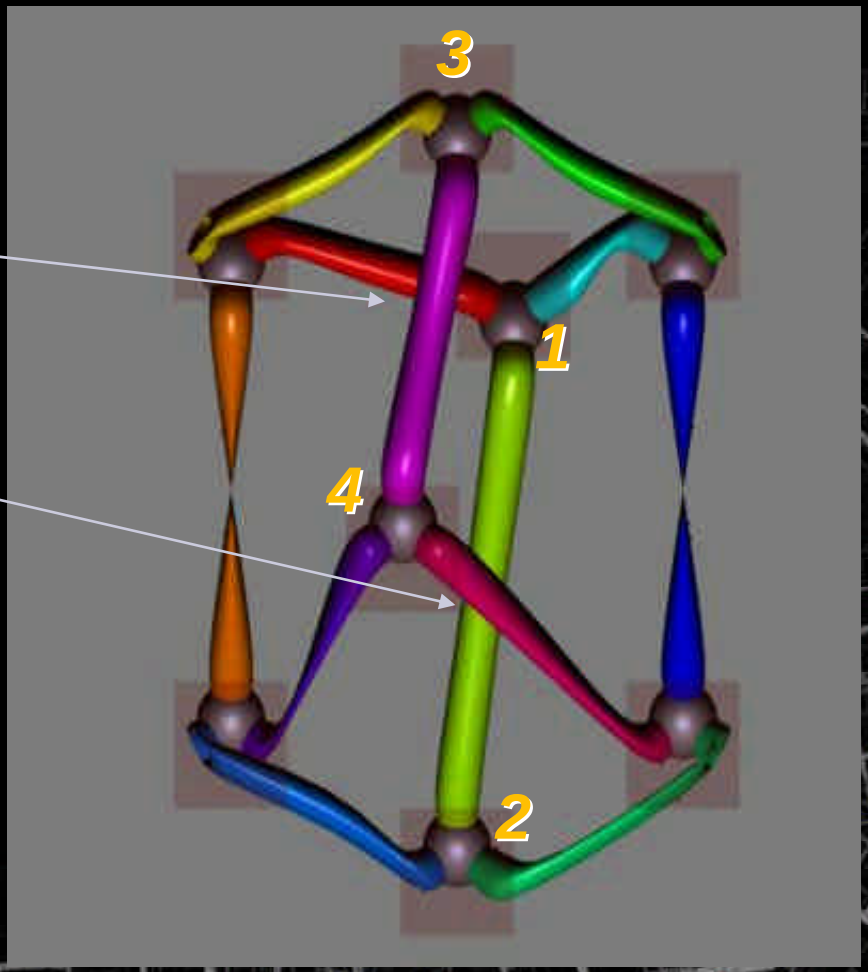




Examples

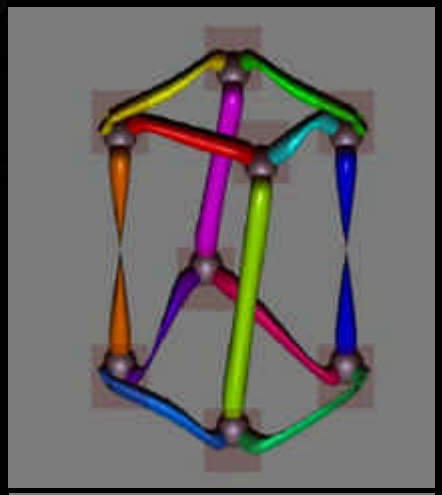
Depth Control

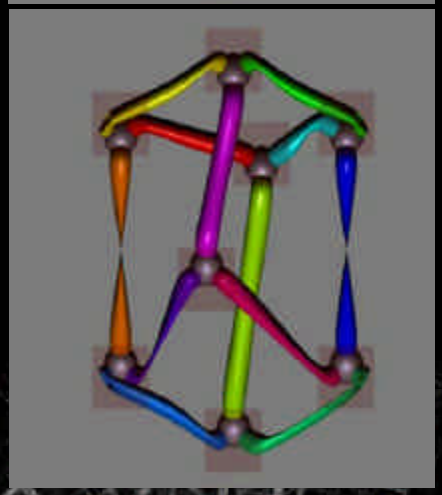


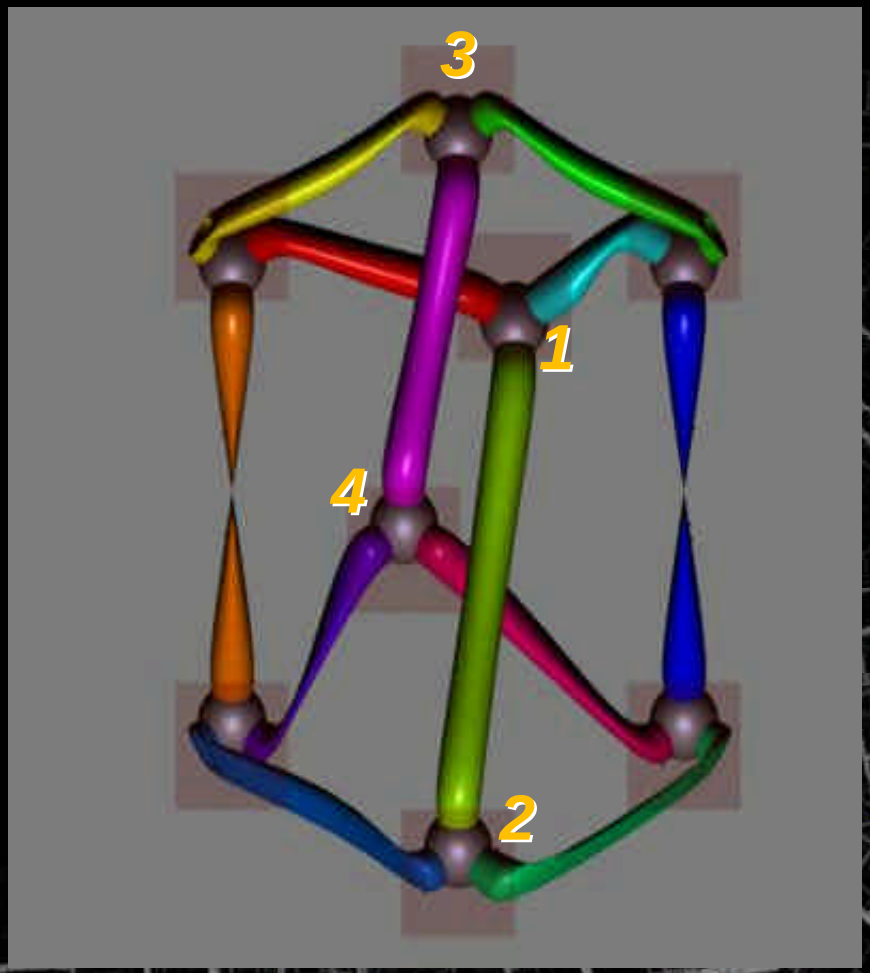


Examples

Depth Control







Constraints

No Constraints

With Constraints

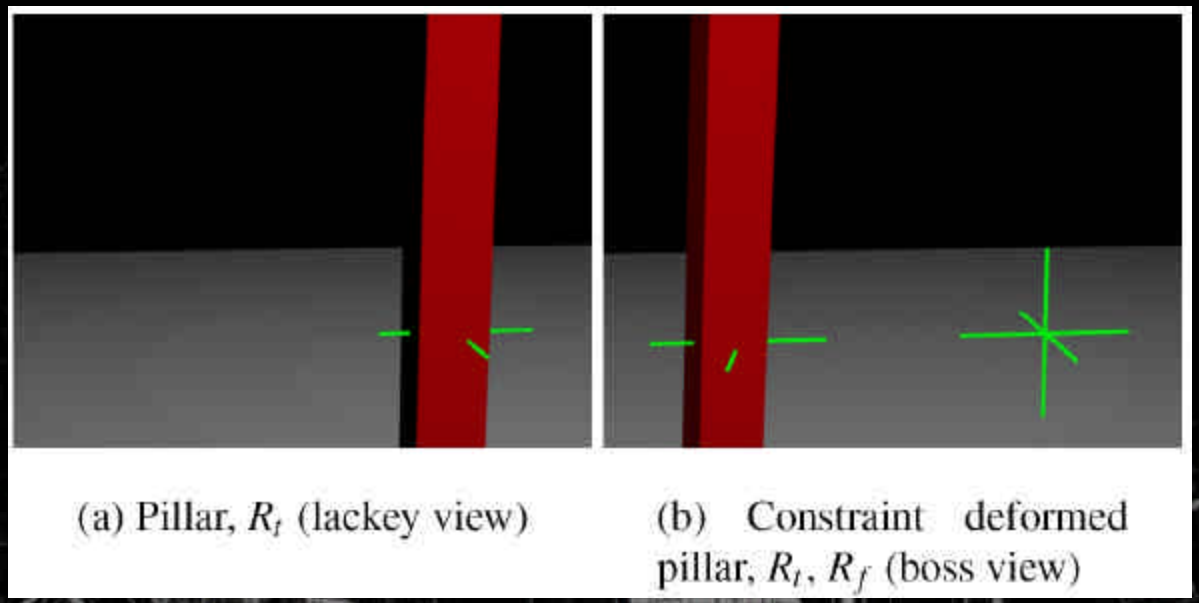
Constraints

To see constraint frame R_f in lackey as R_t in master camera *:*

E_i ? E_b^{-1}

$$A_i = C_i M_i V_i (Con)(C_b M_b V_b)^{-1}.$$

…where *Con* is a constraint matrix that captures the affine transformation that maps $R_f E_i$ to $R_t E_b$.



(a) Pillar, R_t (lackey view)

(b) Constraint deformed pillar, R_t, R_f (boss view)

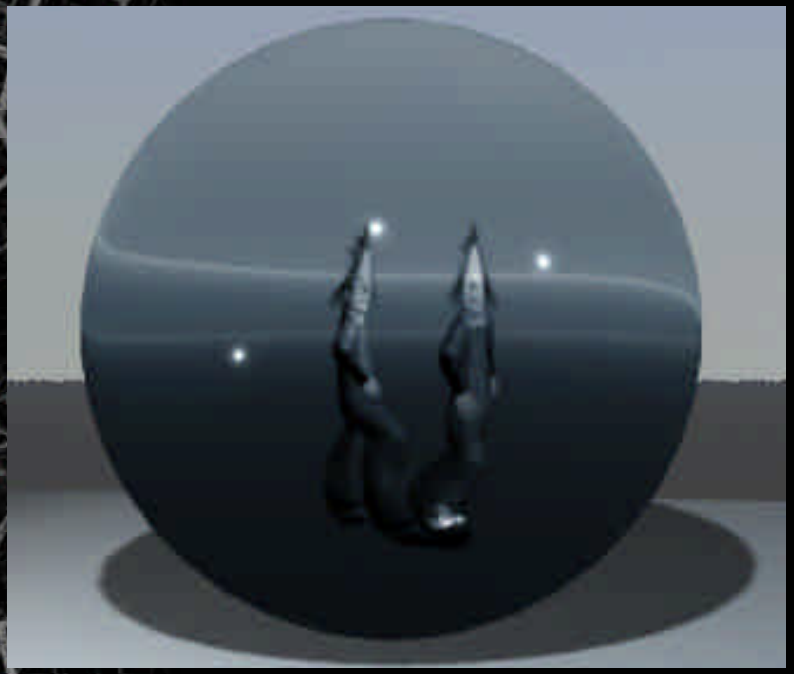
Constraints

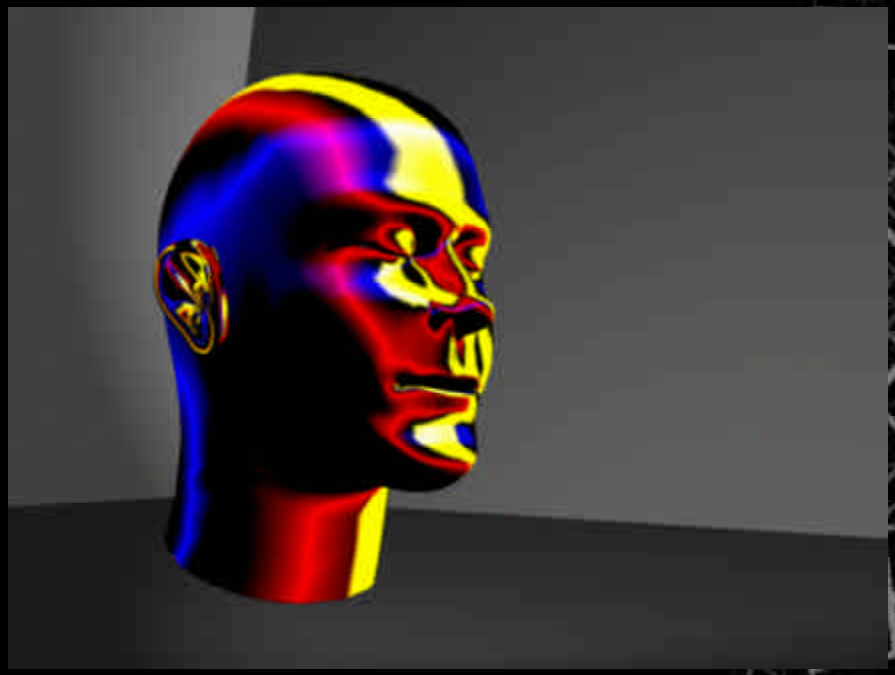
To see constraint frame R_f in lackey as R_t in master camera *:*

$$Con = (Cartesianize(R_f C_i M_i V_i))^{-1} Cartesianize(R_t C_b M_b V_b)$$

…in general *Con* is defined as an RBF interpolation of multiple constraints per scene object, per camera.

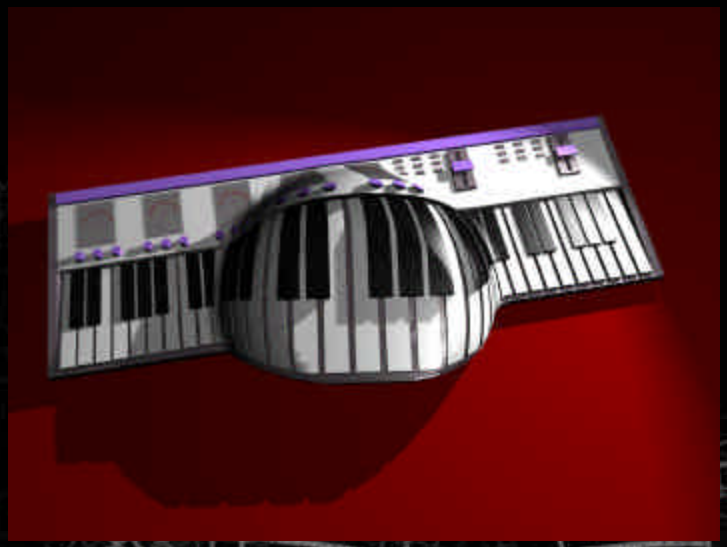
Stylized multiview shading





Stylized multiview shading

Shadows



Wrong shadows

Corrected shadows

Ryan

http://www.dgp.toronto.edu/~karan/perspective.htm

Cords: a physical 3D curve

- Cord: A geometric curve primitive that bends and wraps around scene geometry with physical attributes of stiffness and elasticity.

- Desired results are rooted in physics, under precise animator control.

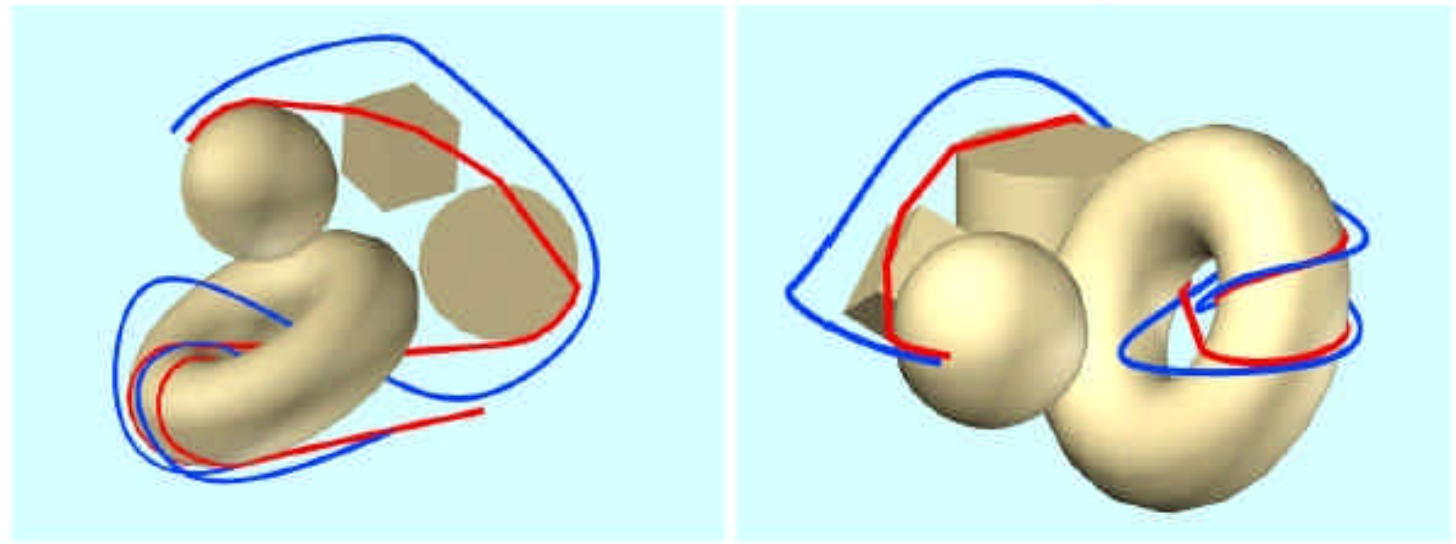
Motivation

Appealing sparse geometric representations are artistically more challenging than real geometry.

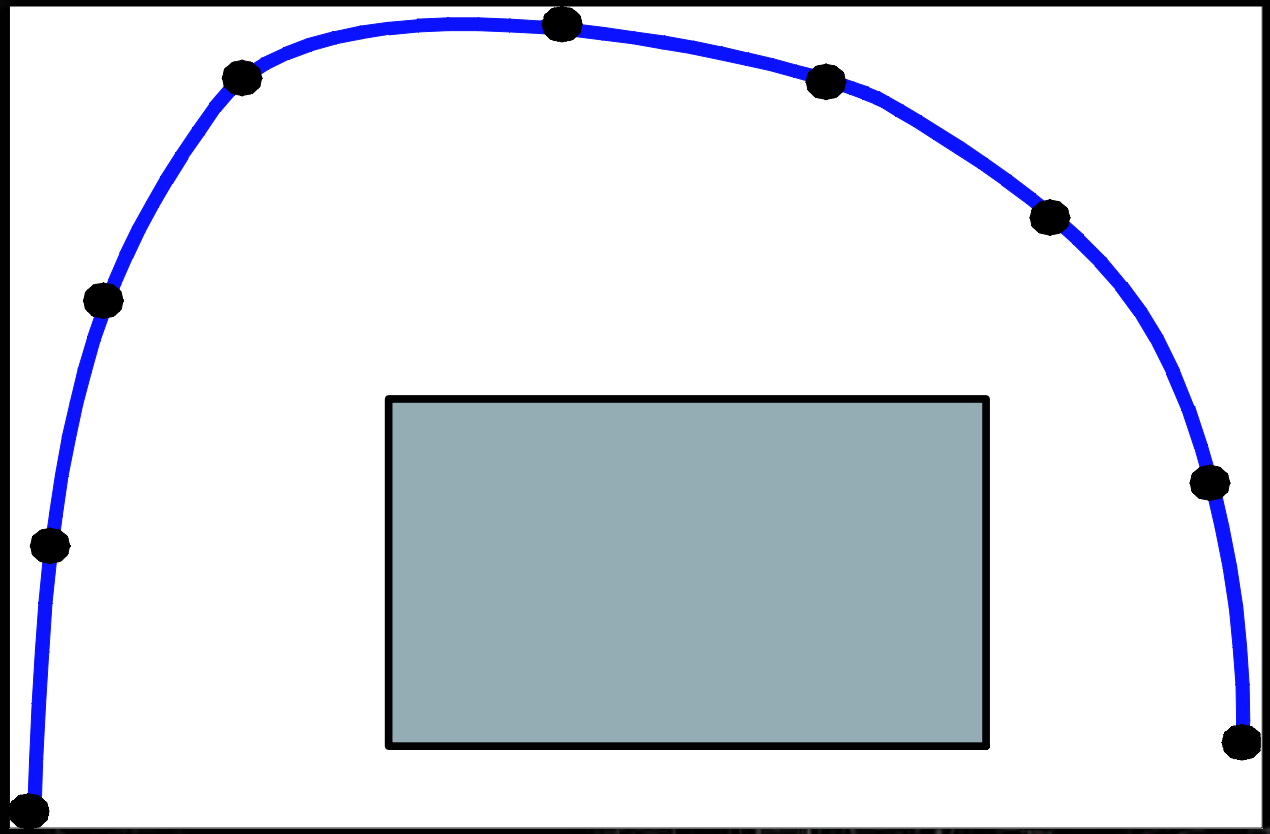
Cords

Cord Definition

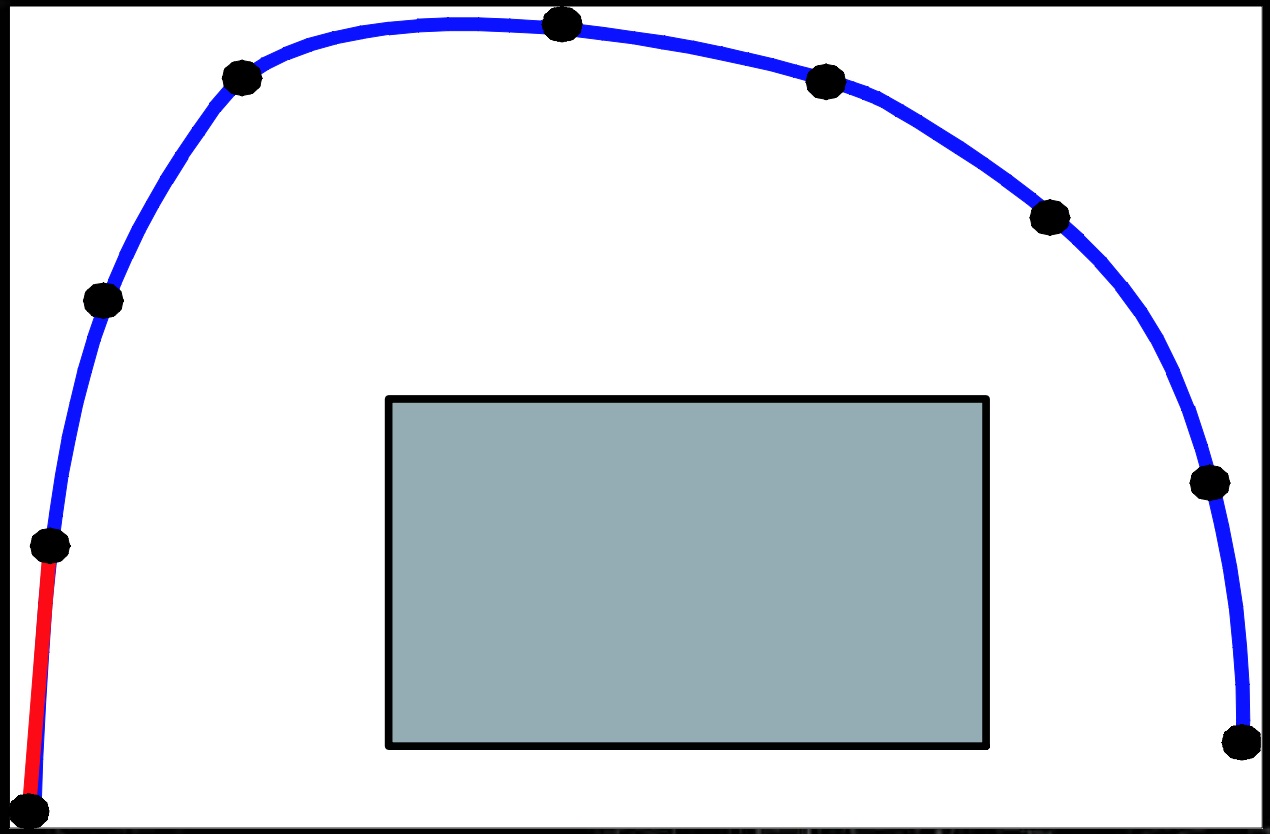
- Defined by a guide curve f(t), stiffness, elasticity, length and 3D scene geometry.
- Cord must interactively bend and wrap around geometry in response to animator control of Cord parameters.



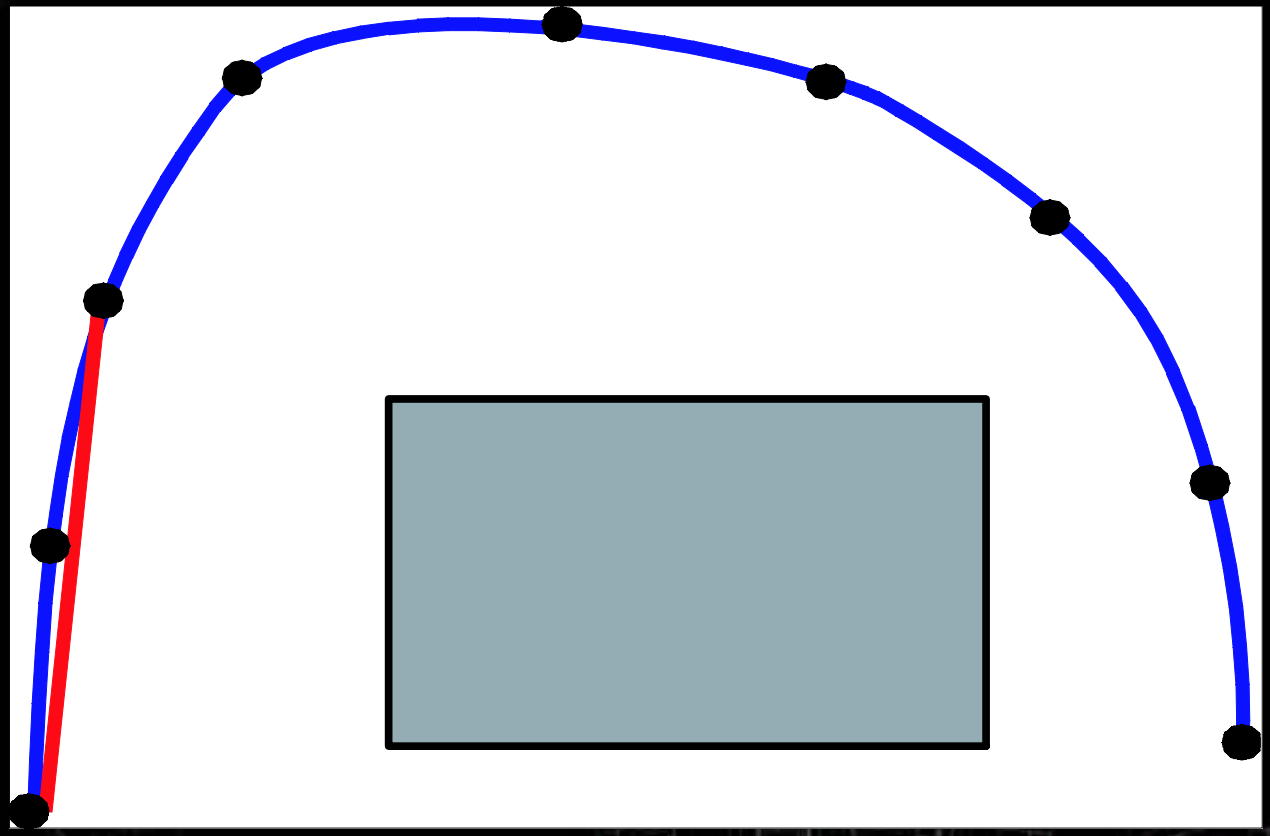
Cords Algorithm



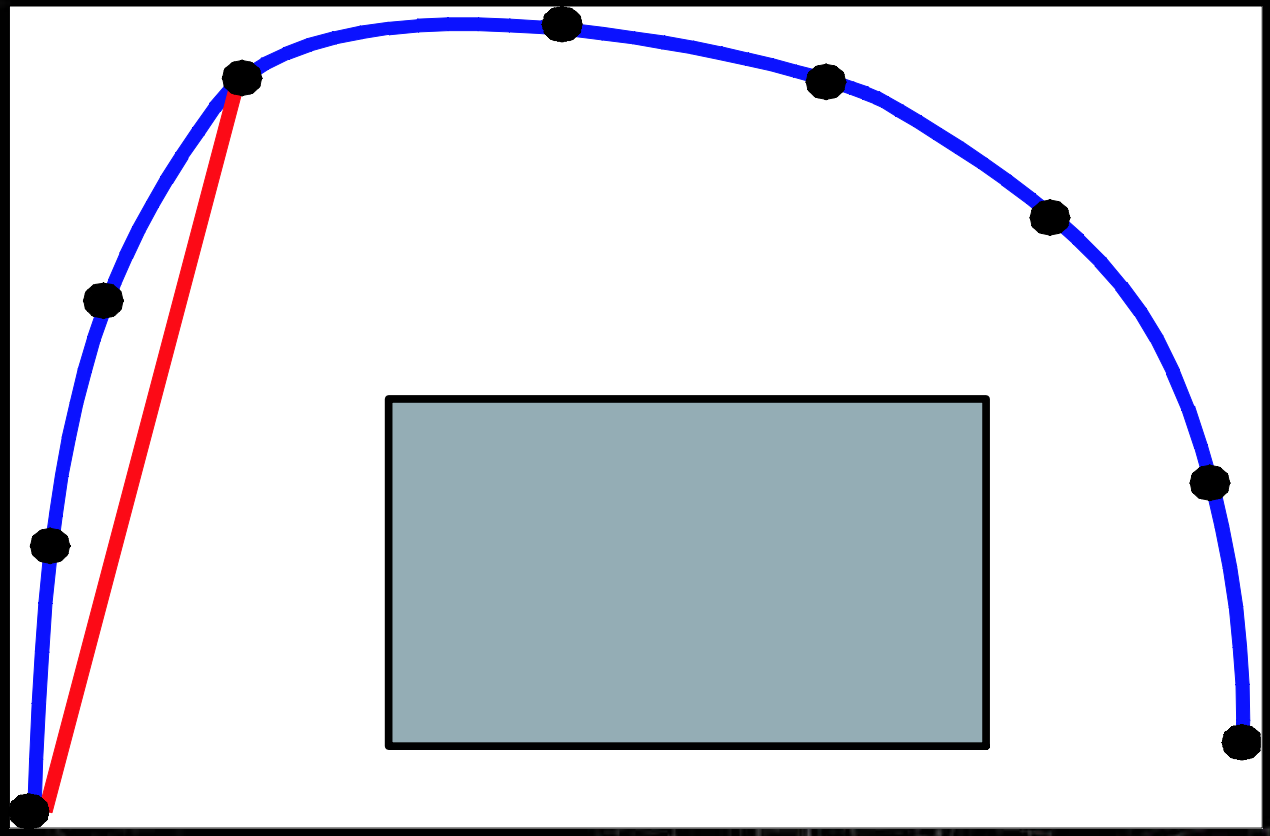
Cords Algorithm



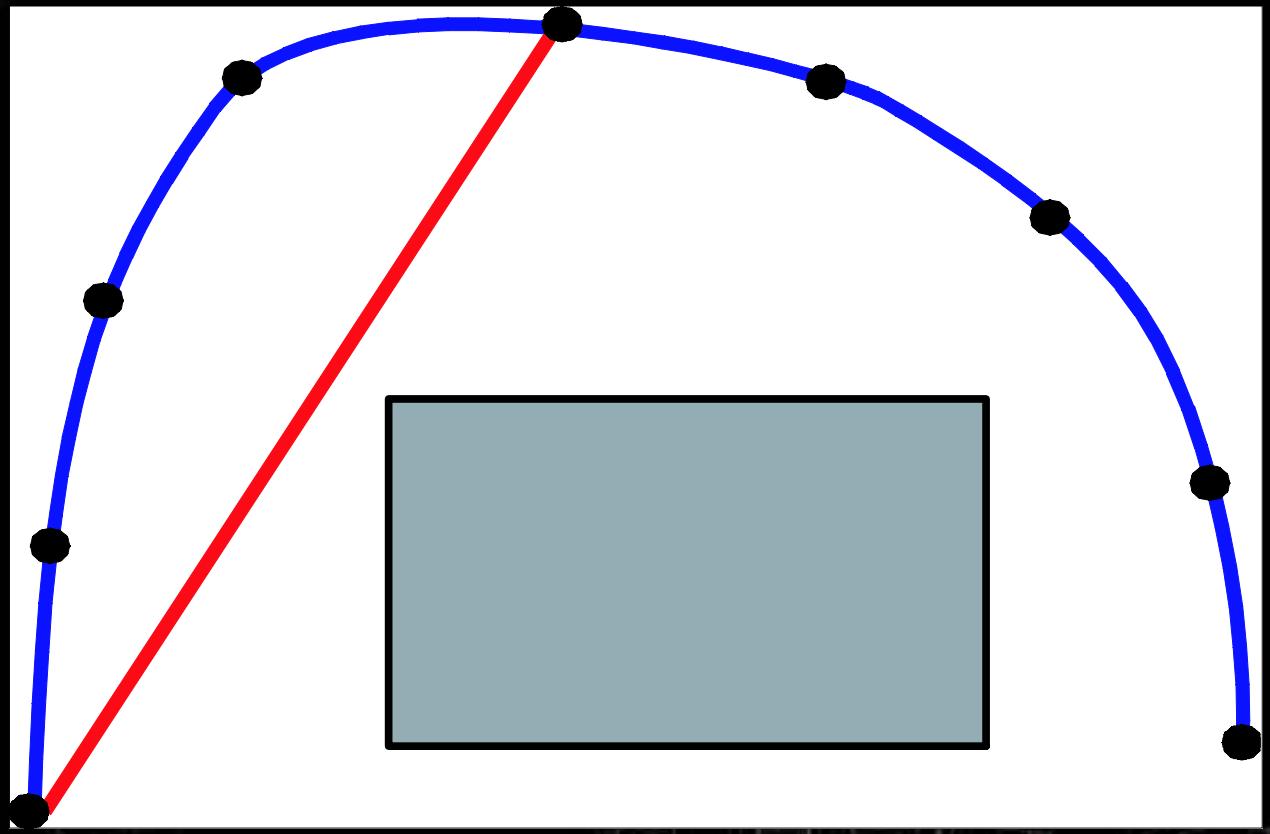
Cords Algorithm



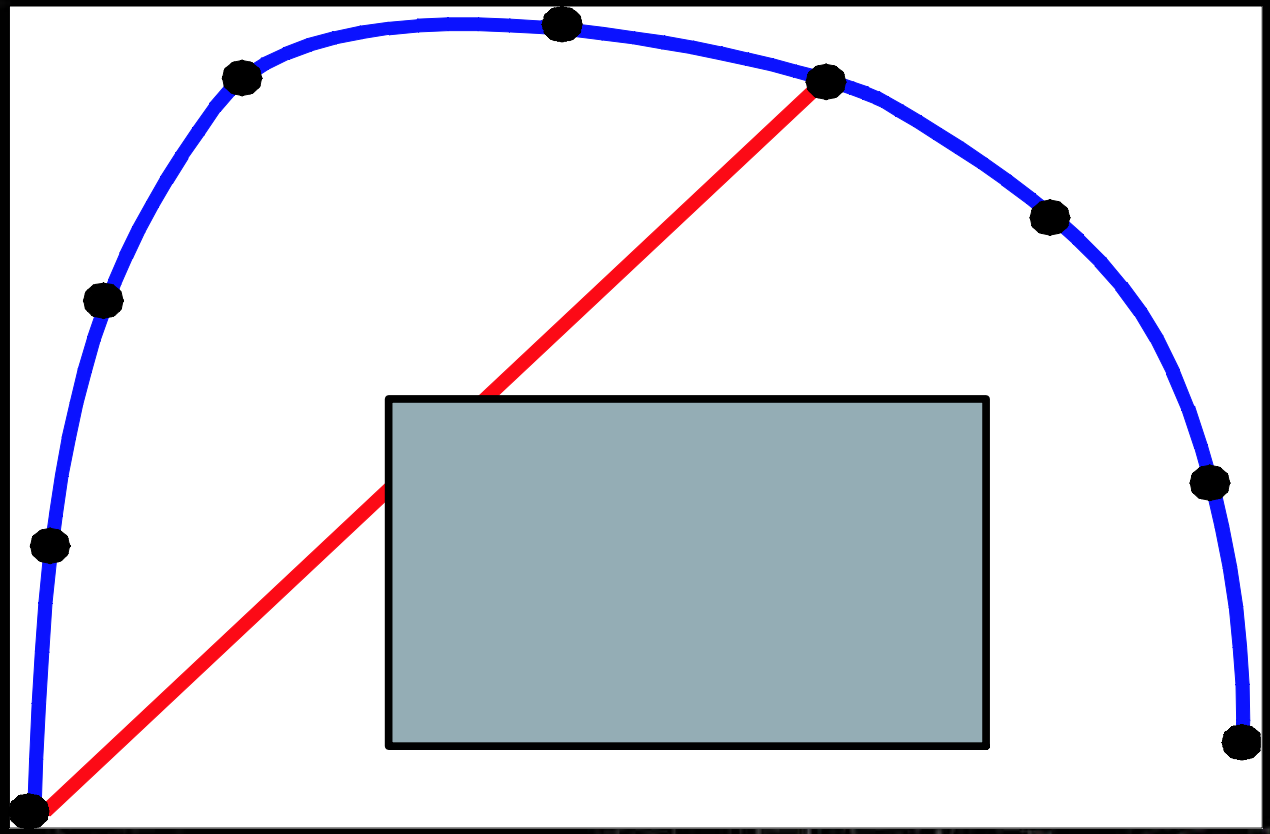
Cords Algorithm



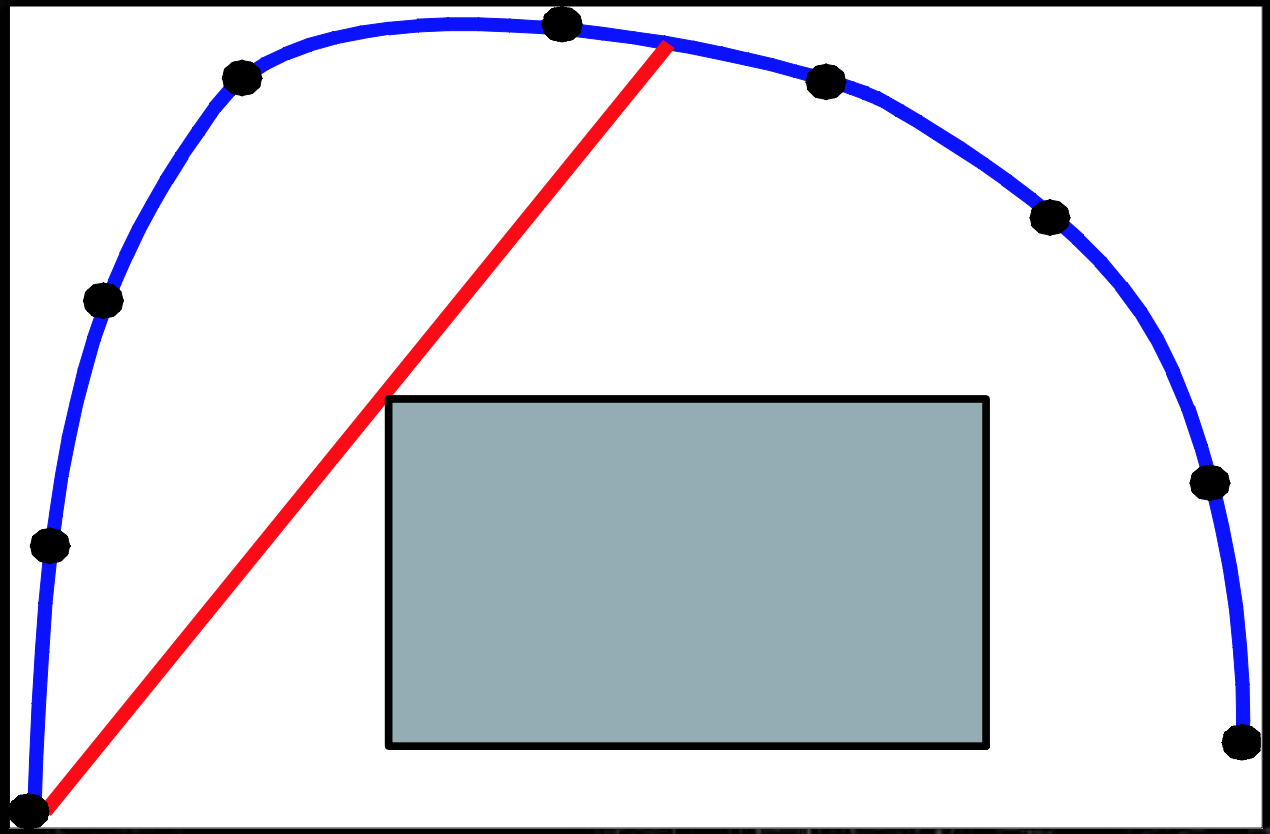
Cords Algorithm



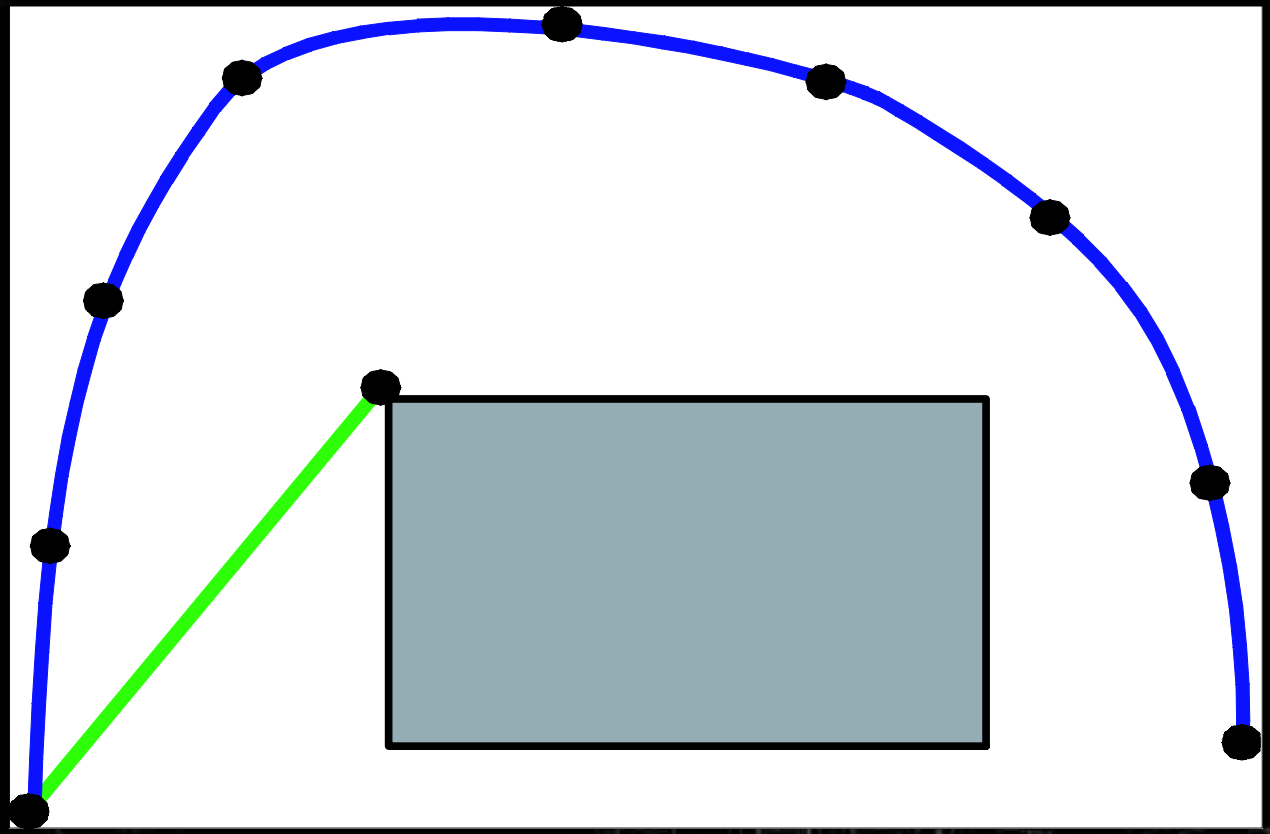
Cords Algorithm



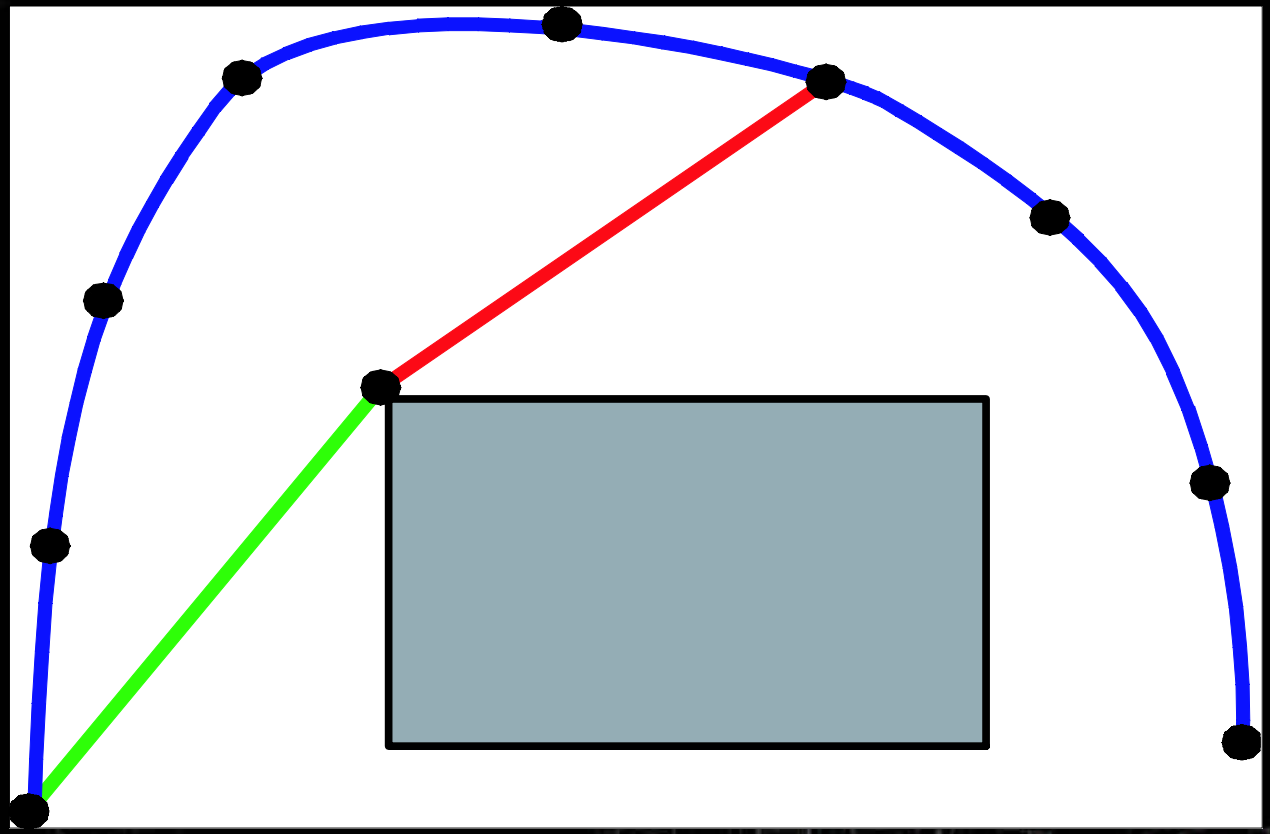
Cords Algorithm



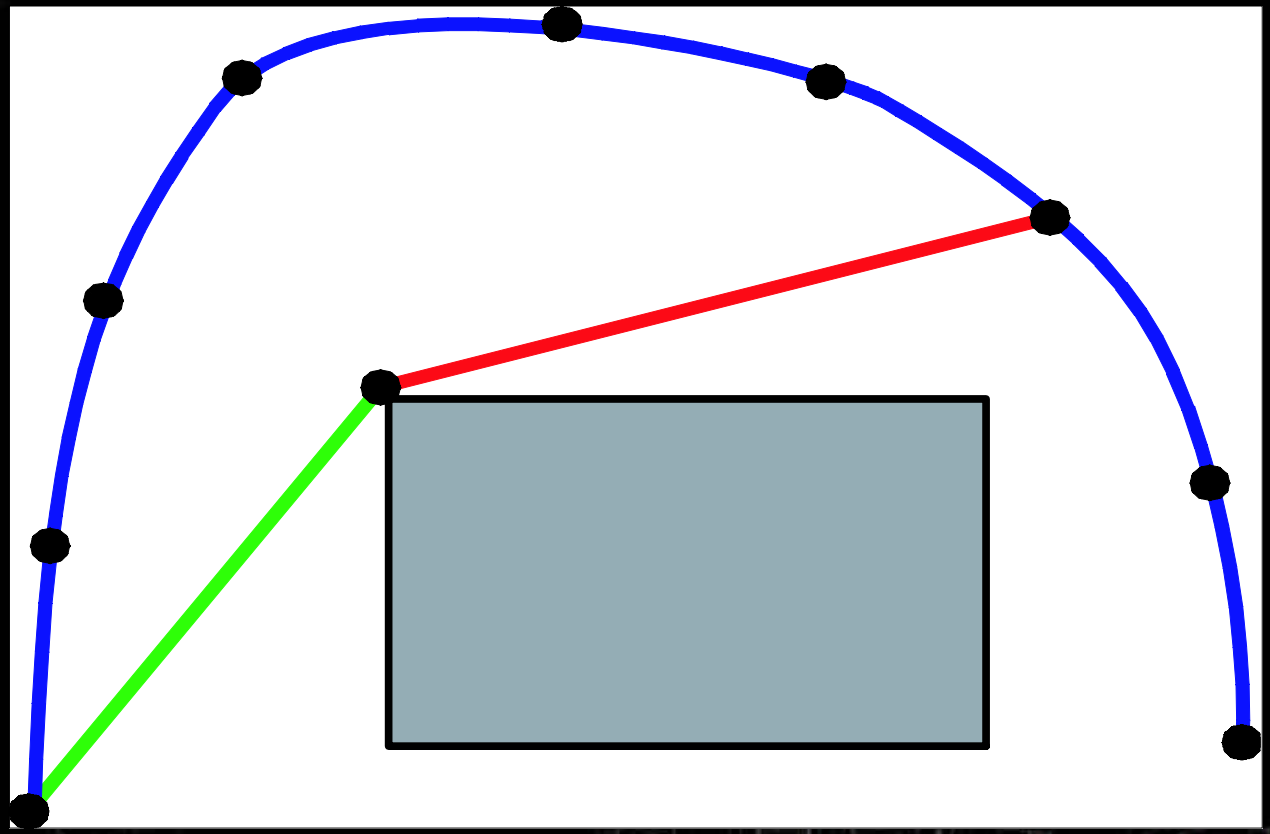
Cords Algorithm



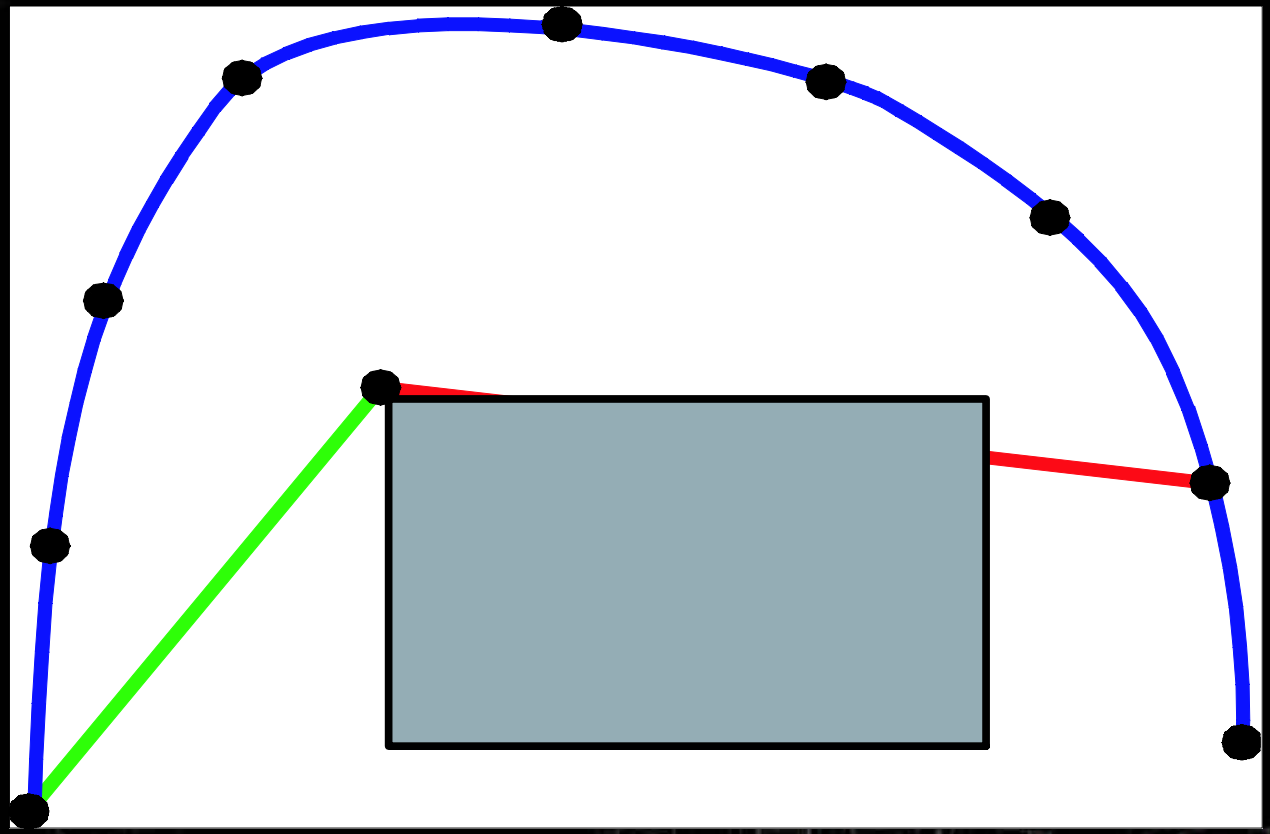
Cords Algorithm



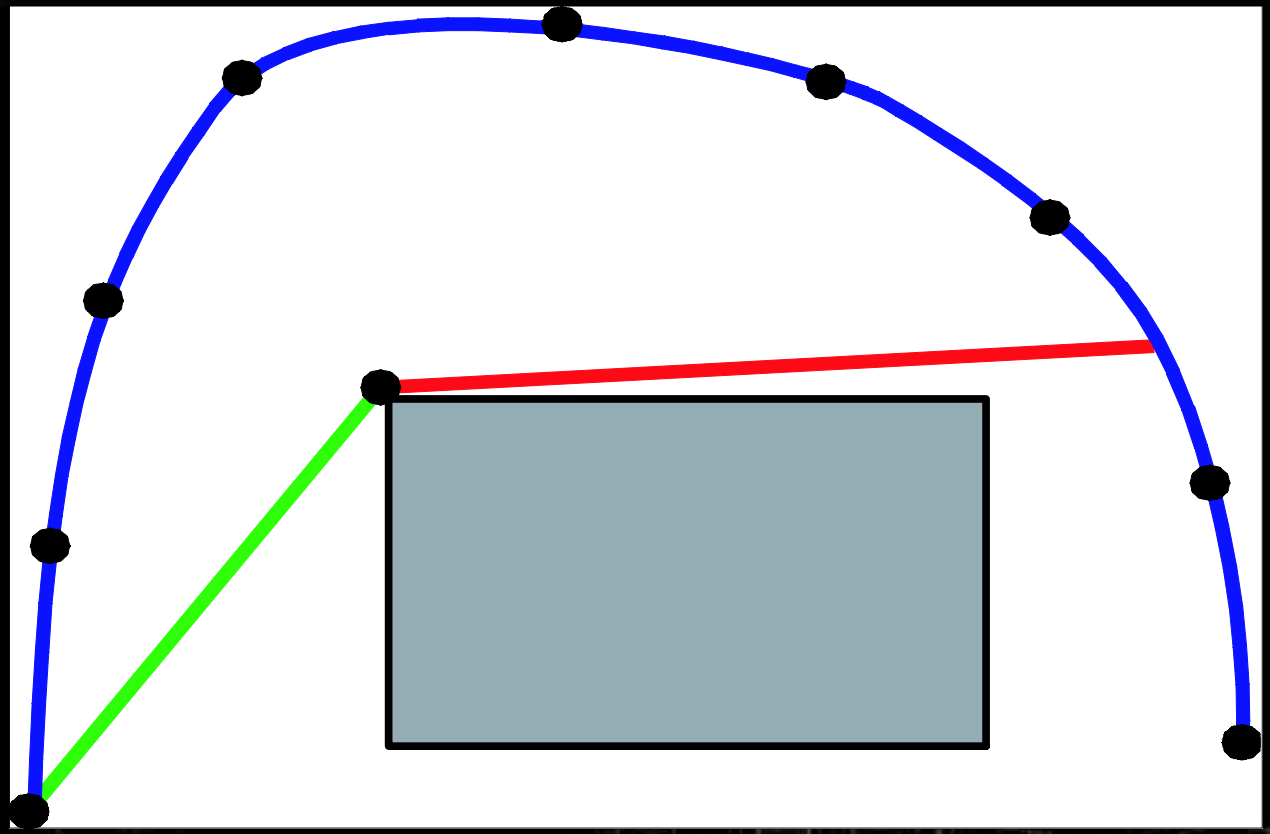
Cords Algorithm



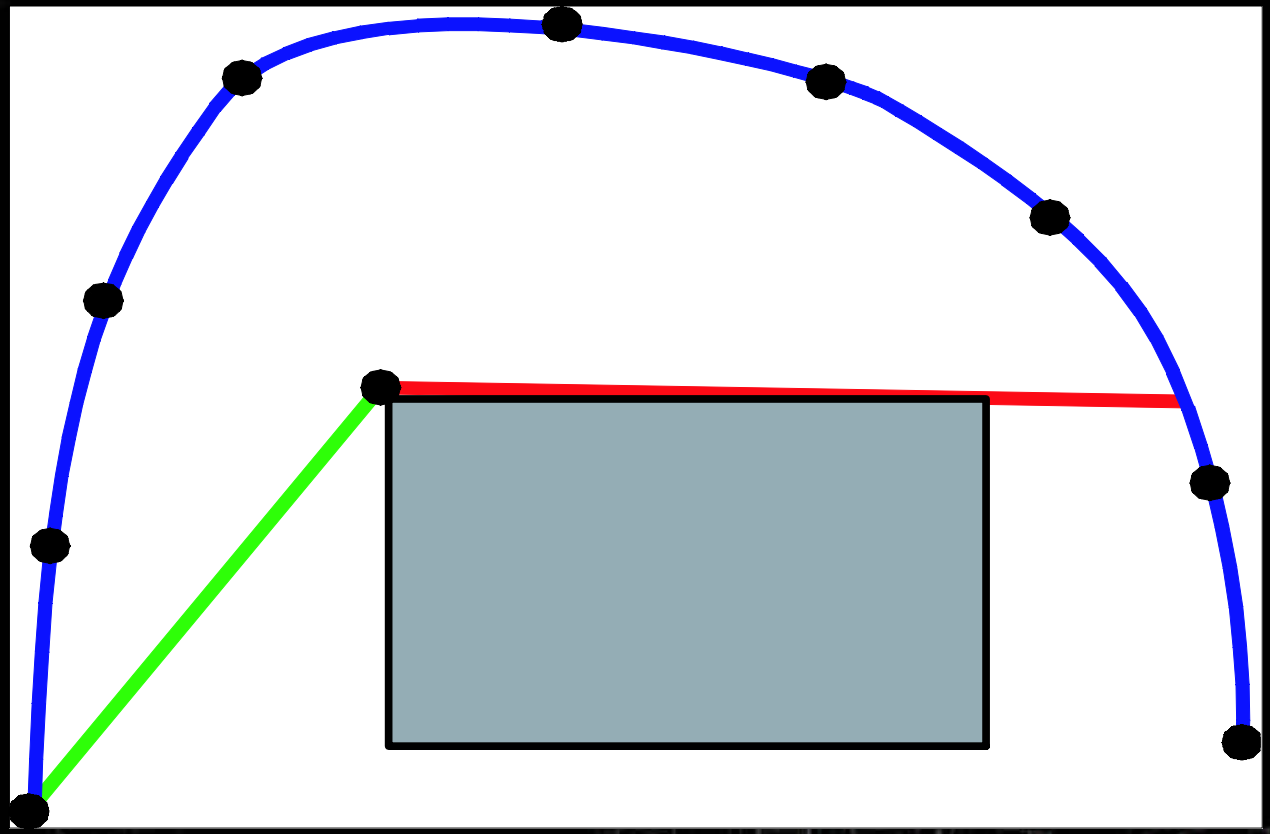
Cords Algorithm



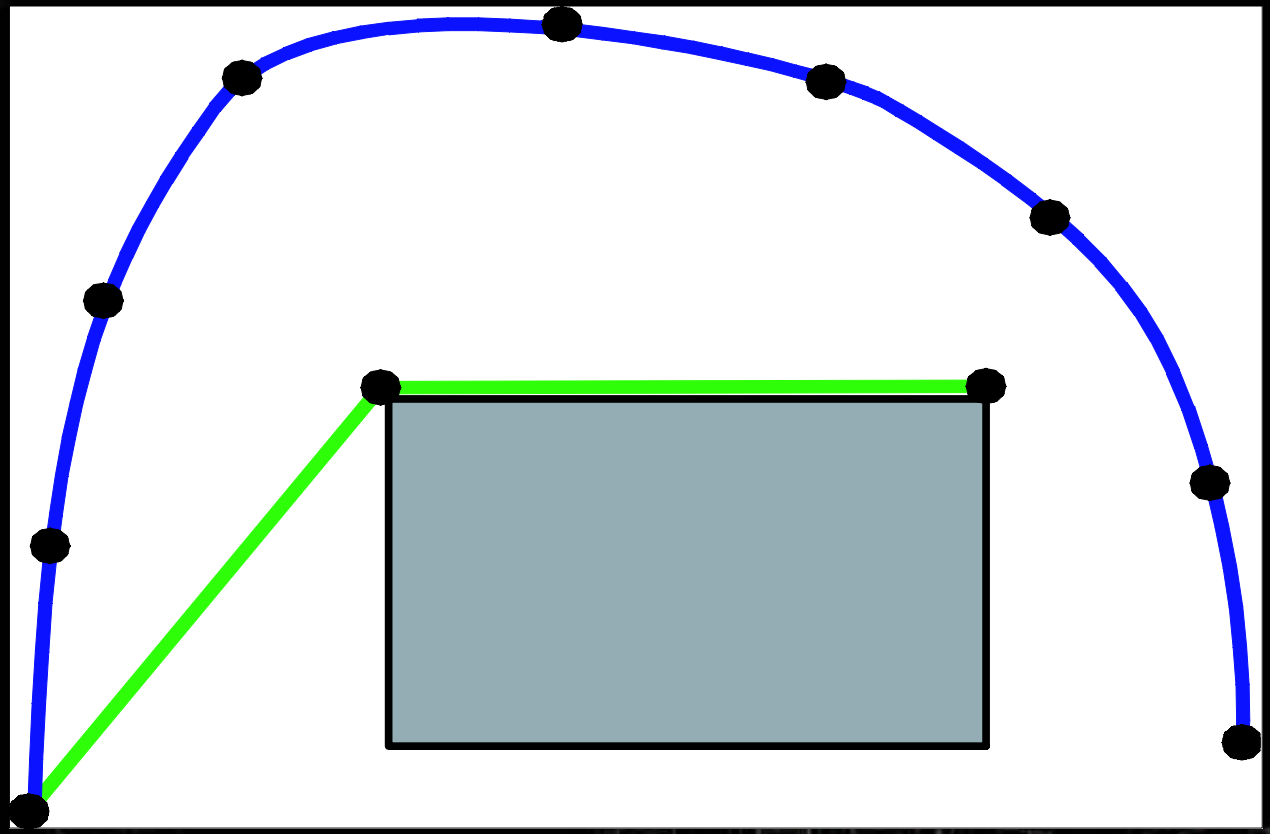
Cords Algorithm



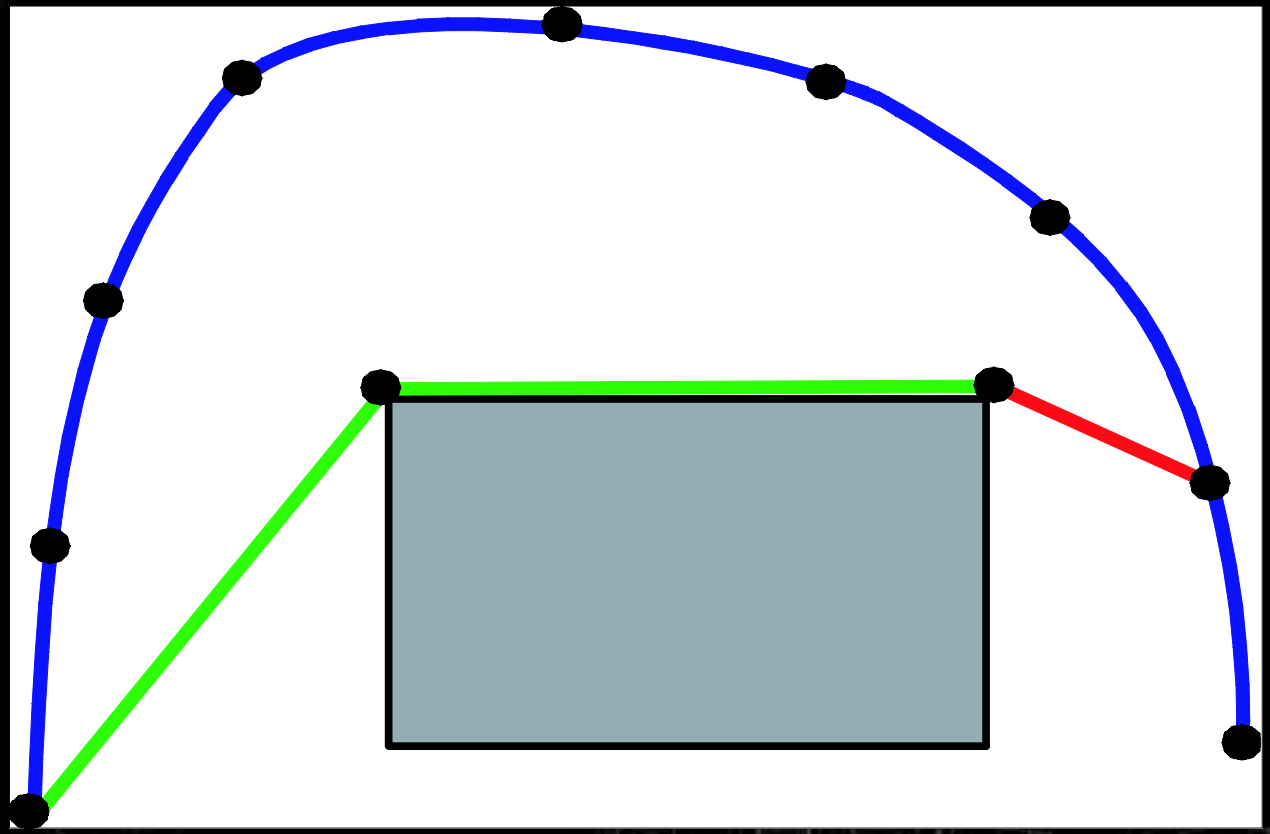
Cords Algorithm



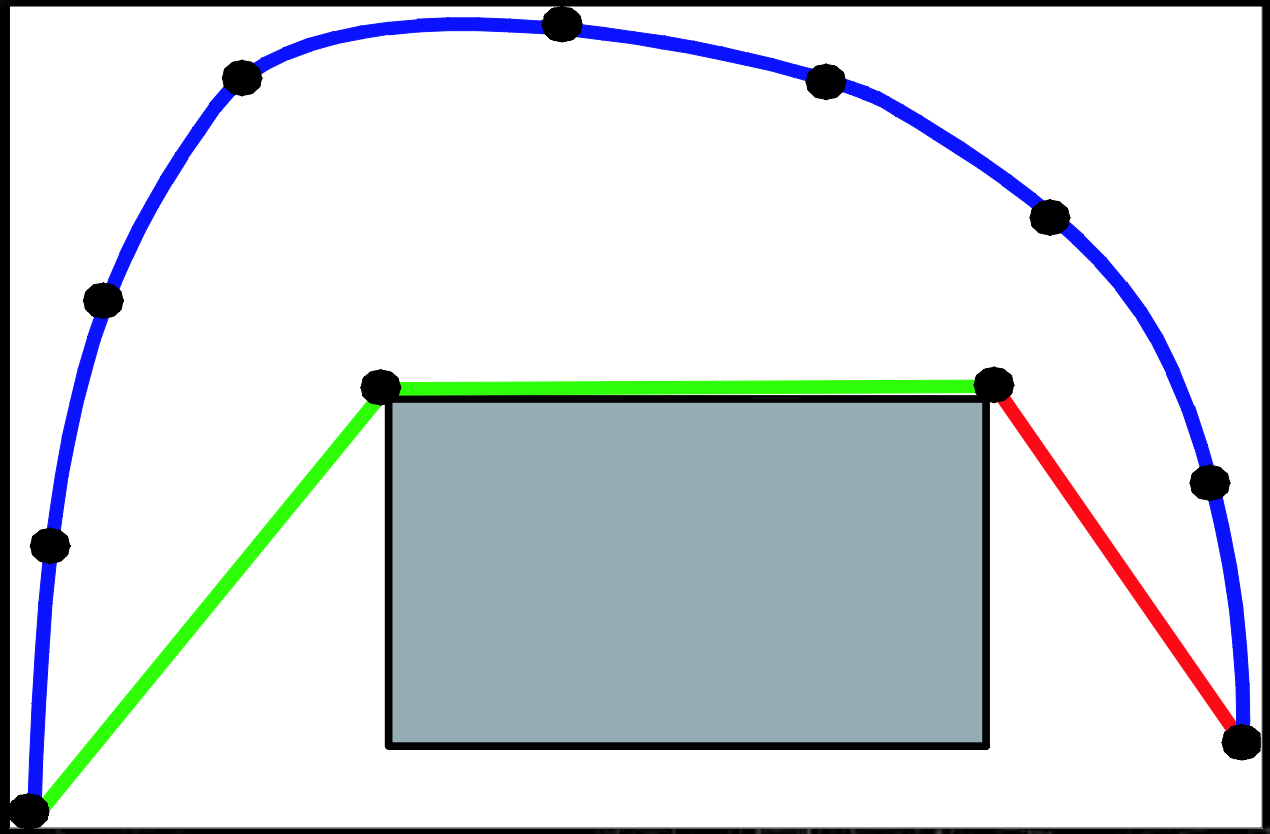
Cords Algorithm



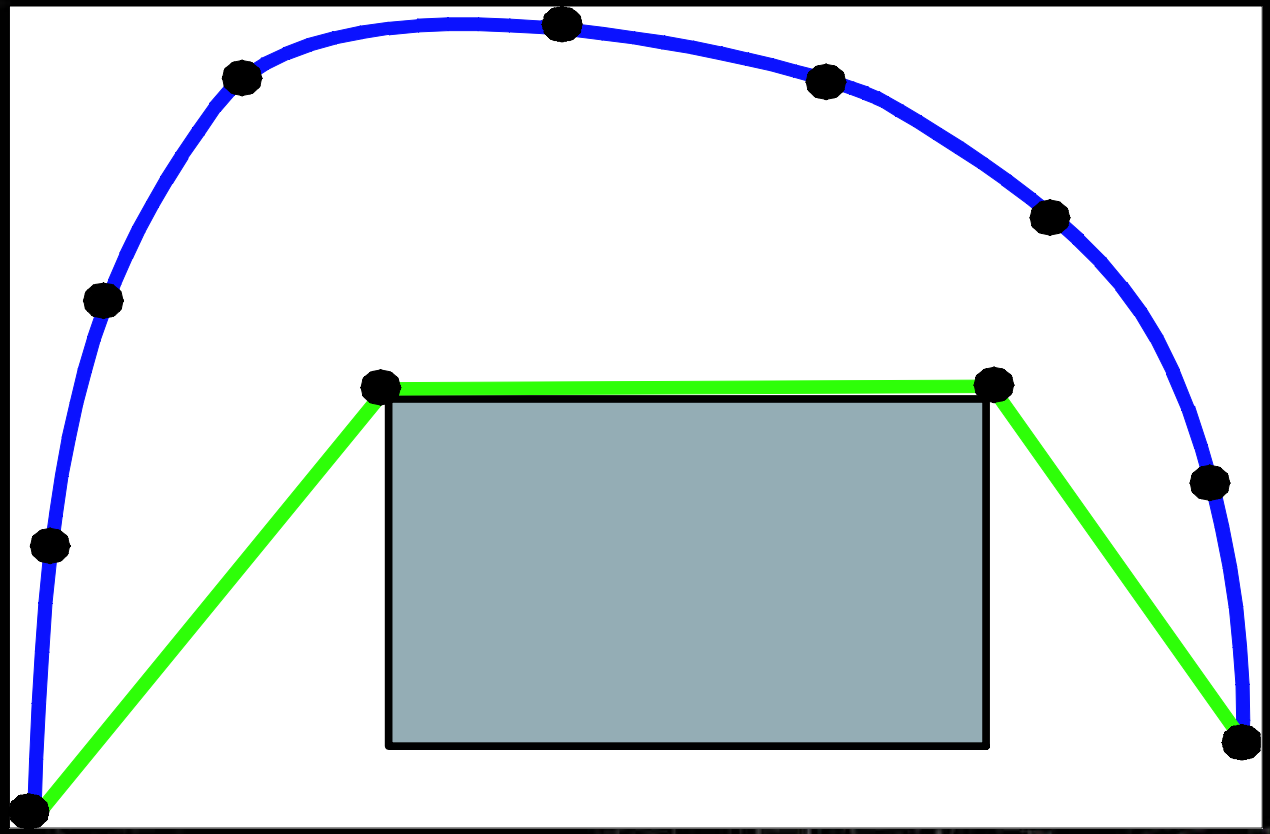
Cords Algorithm



Cords Algorithm



Cords Algorithm

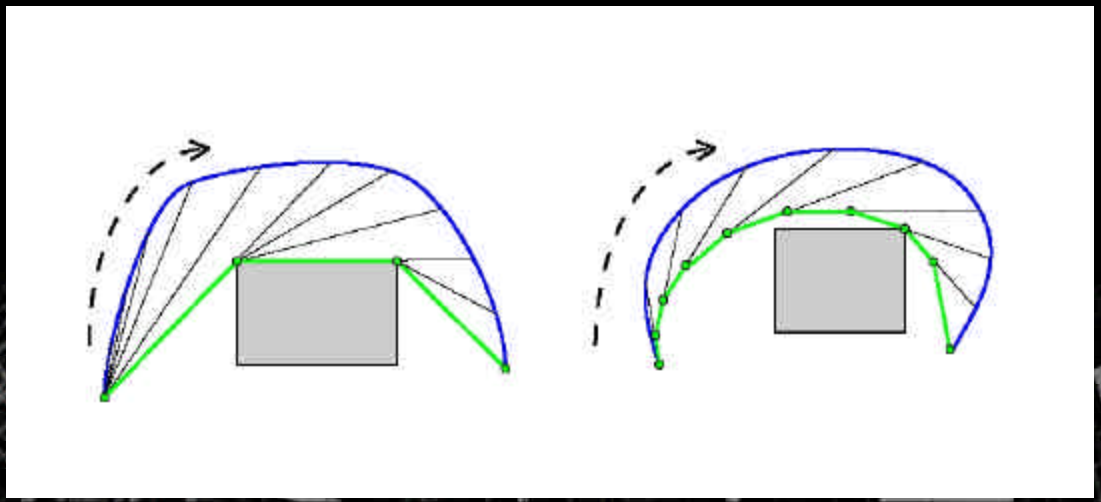


Cord Algorithm

- Initialize Cord to f(0).
- Grow the Cord by stepping along f.

```
if (ray from Cord to f intersects geometry)
  grow cord to intersection
else
  grow cord by a stiffness factor along the ray
```

- Adjust Cord to given length, elasticity.



Cord Analysis

- Cord is represented by polyline $p_0, p_1, p_2 \ldots$ where

$$p_i = p_{i-1} + stiffness * s * (f(i * s) - p_{i-1}) \text{ and } p_0 = f(0).$$

- We wish to prove that the Cord is a continuous limit curve *g(t)*, (*a=stiffness, m=t/s*) as step-size *s→0* or *m→a*.

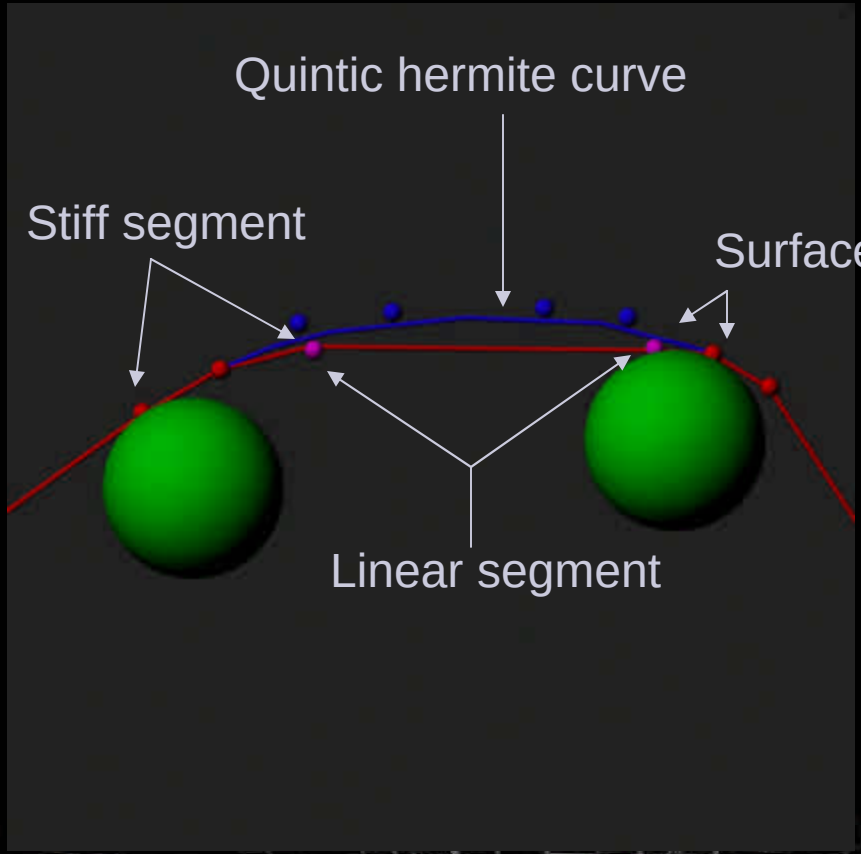
$$p_m = f(0) * (1 - a * t/m)^m + (a * t/m) * (\sum_{i=1}^{m} f(i * t/m) * (1 - a * t/m)^{m-i}).$$

- We show using Riemann sums that:

$$g(t) = f(0) * e^{-a*t} + a * e^{-a*t} * (\int_0^t f(x) e^{ax} dx)$$

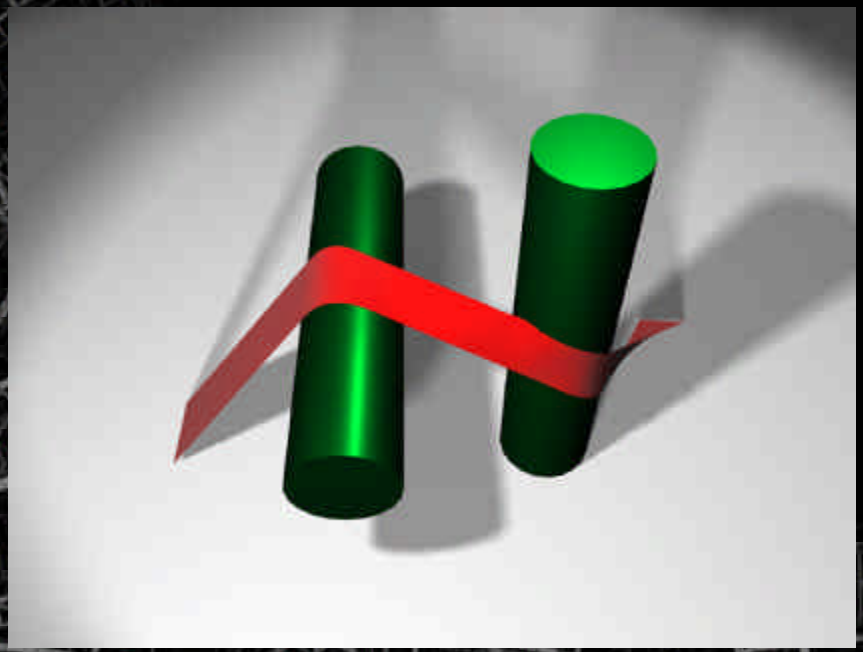
- Cords can thus be analytically defined for parametric polynomial guide curves.

Cord Continuity



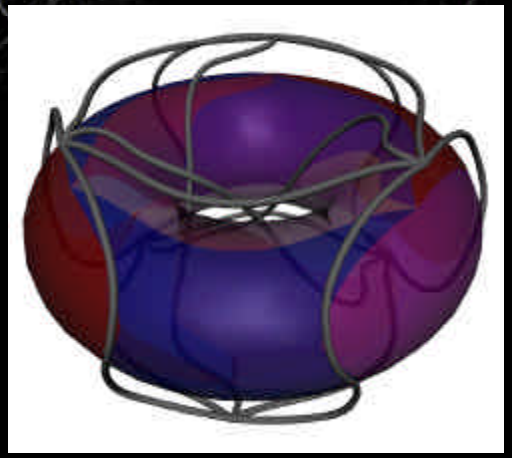
Wide and thick Cords

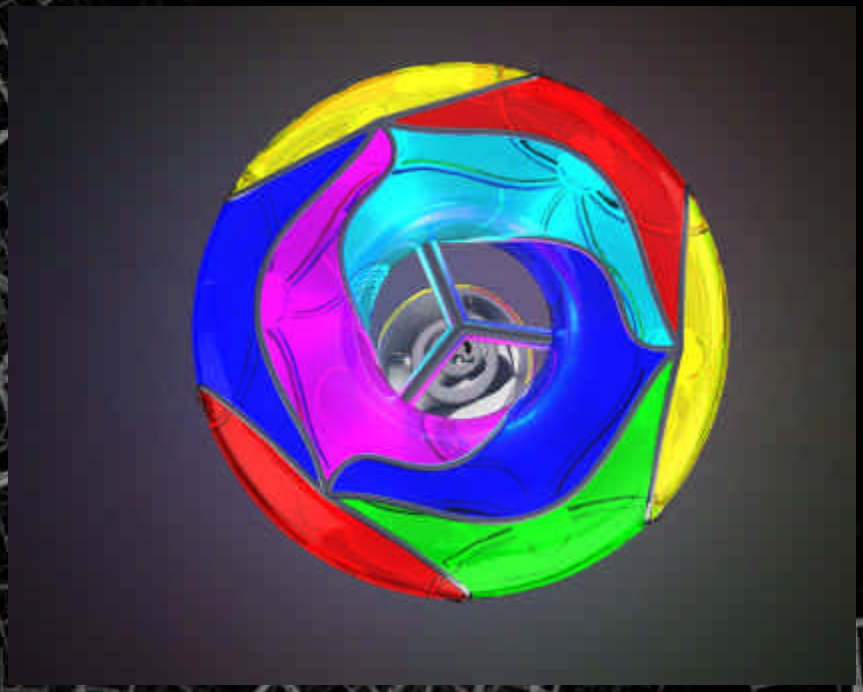
- Parameterized sparse geometric representation.
- Extension to higher dimensional primitives.



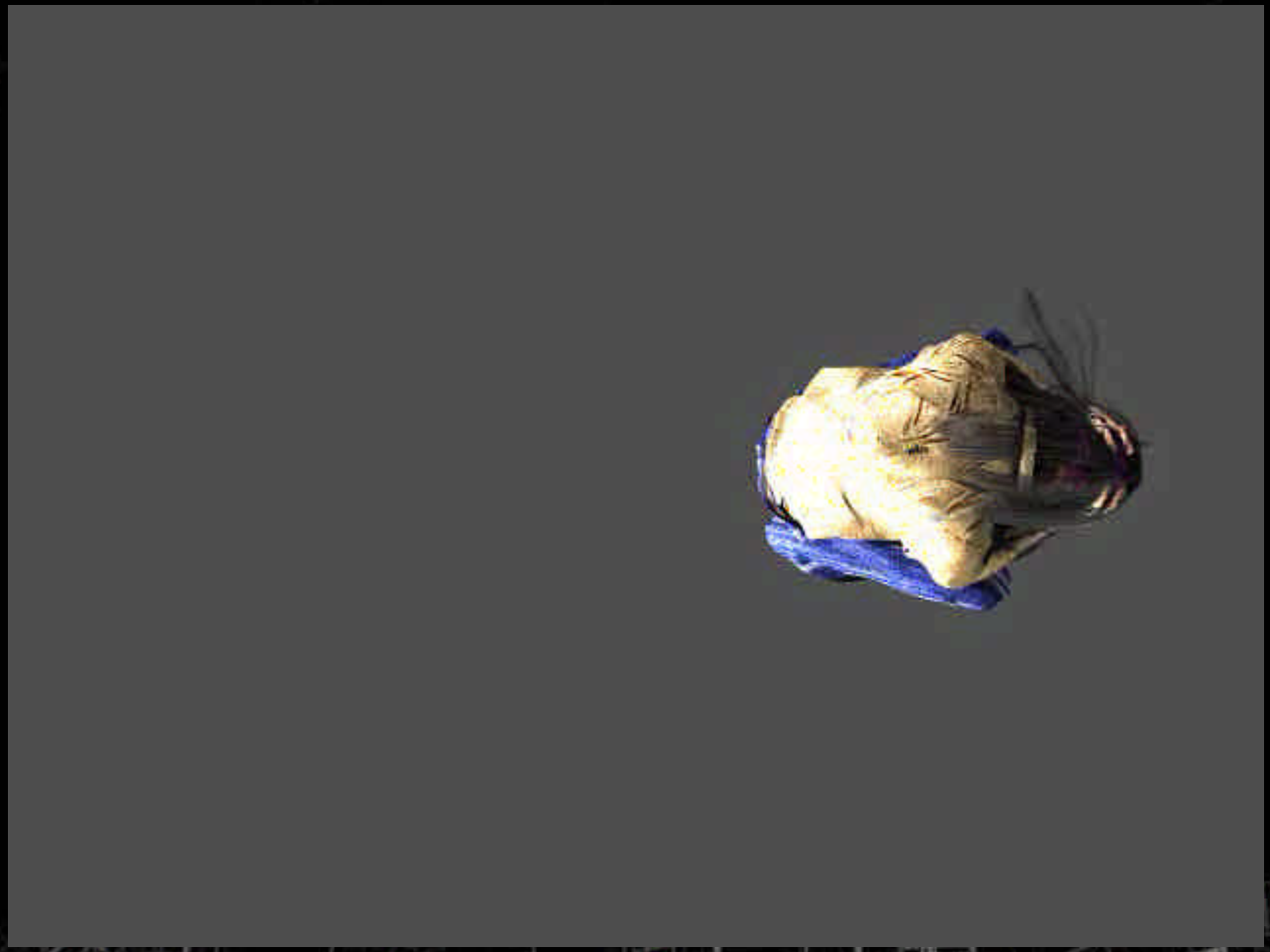
Controlling curve on Geometry

K6 embedded on a torus.





Tying up the Animators



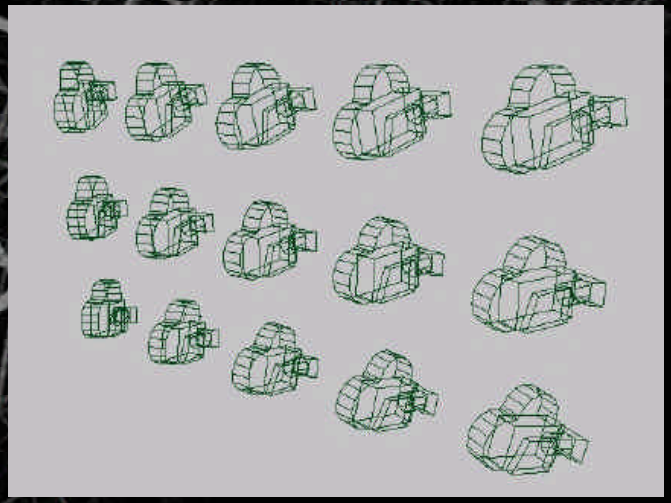
Acknowledgements

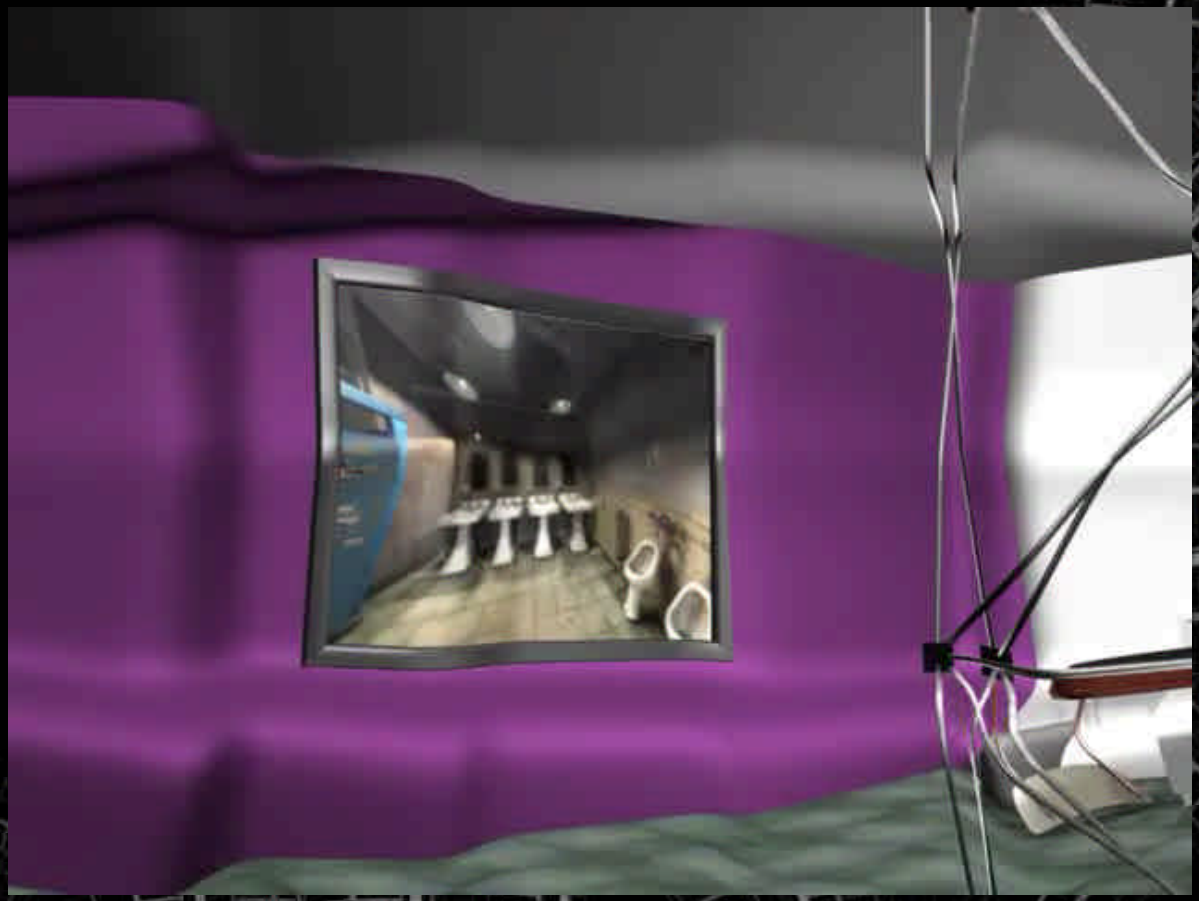
Chris Landreth, Dave Baas.
Ryan programming and animation crew.

Support: MITACS, NSERC, CCA, NFB Canada, Seneca College, Alias, Pixar.

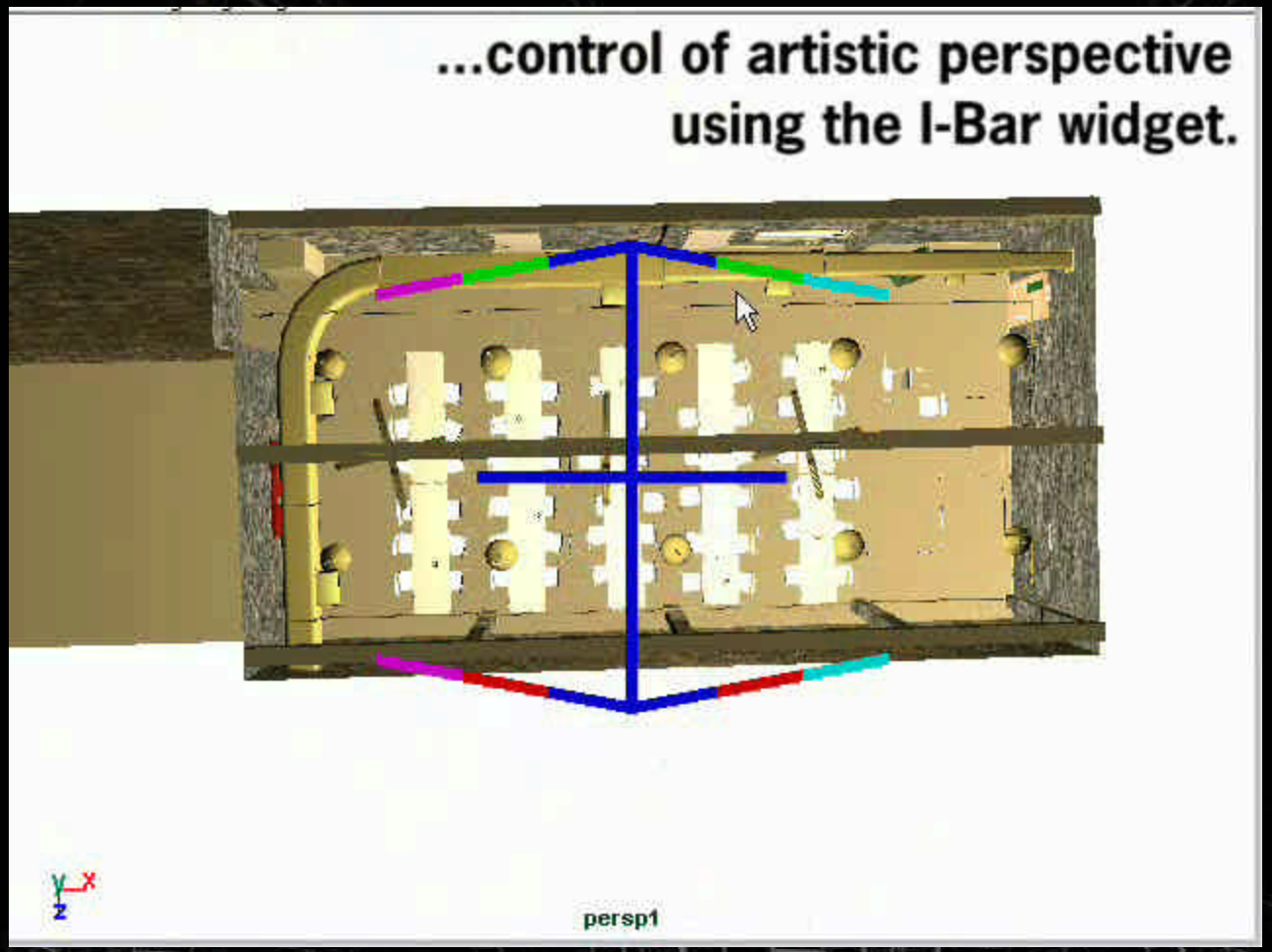
http://www.dgp.toronto.edu/~patrick/ryanTech

Hierarchical nonlinear projections





Projection Widgets



Datamining cameras for projection

Nonlinear projection goals

- Local linear perspective
- Continuous nonlinear projections
- Artistic control of composition, projection
- Coherent shading, shadows, lighting
- Interactive and incremental
- Handle complex scenes

Nonlinear projection

Cords: tying up animators