Spatial Deformations: FFDs, wires, wraps...

Karan Singh

Dynamic Graphics Project
University of Toronto
www.dgp.toronto.edu

What are spatial deformations?

Functional mapping from $\Re^n \rightarrow \Re^n$

Affine Transformations

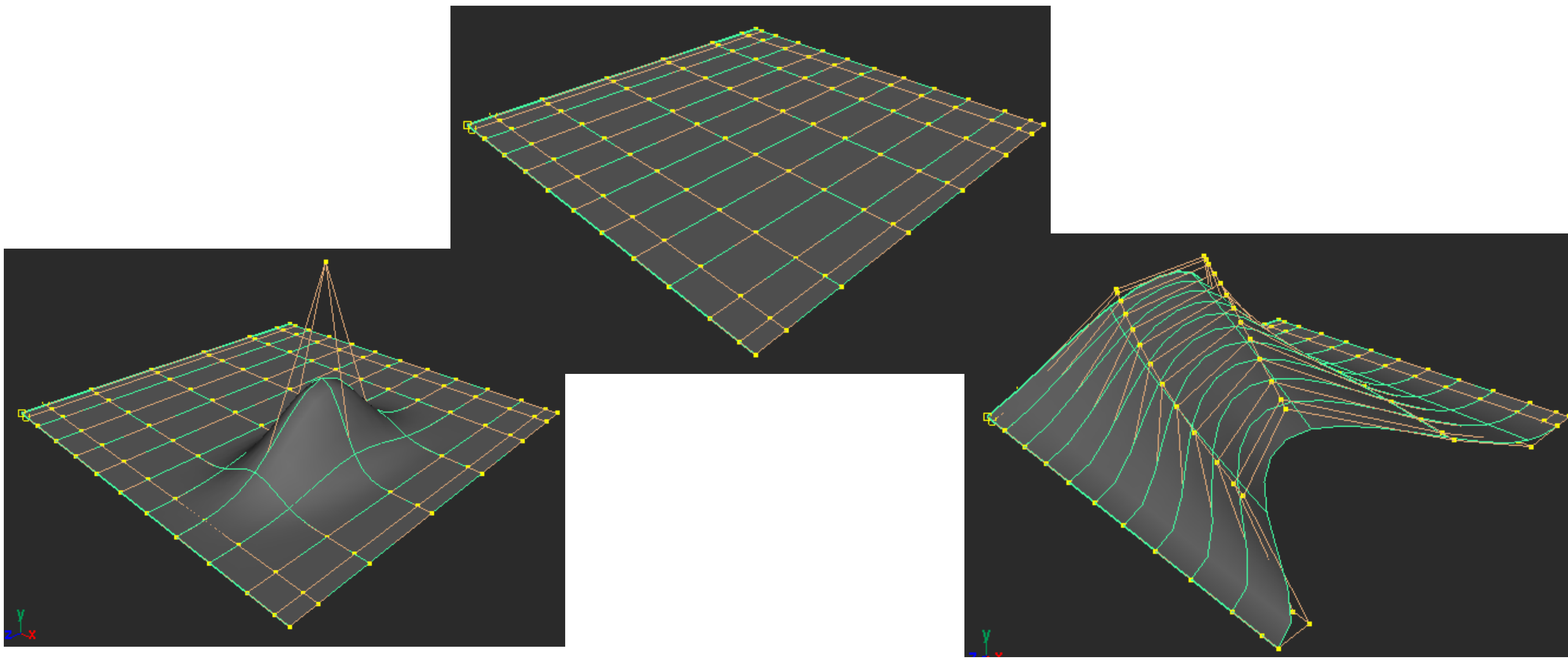
Nonlinear Deformations | Axial Deformations

Free-form Deformations | Wires

↓ Increasingly complex

Why spatial deformations?

Surface deformation may require the coordinated control of many points defining the surface.

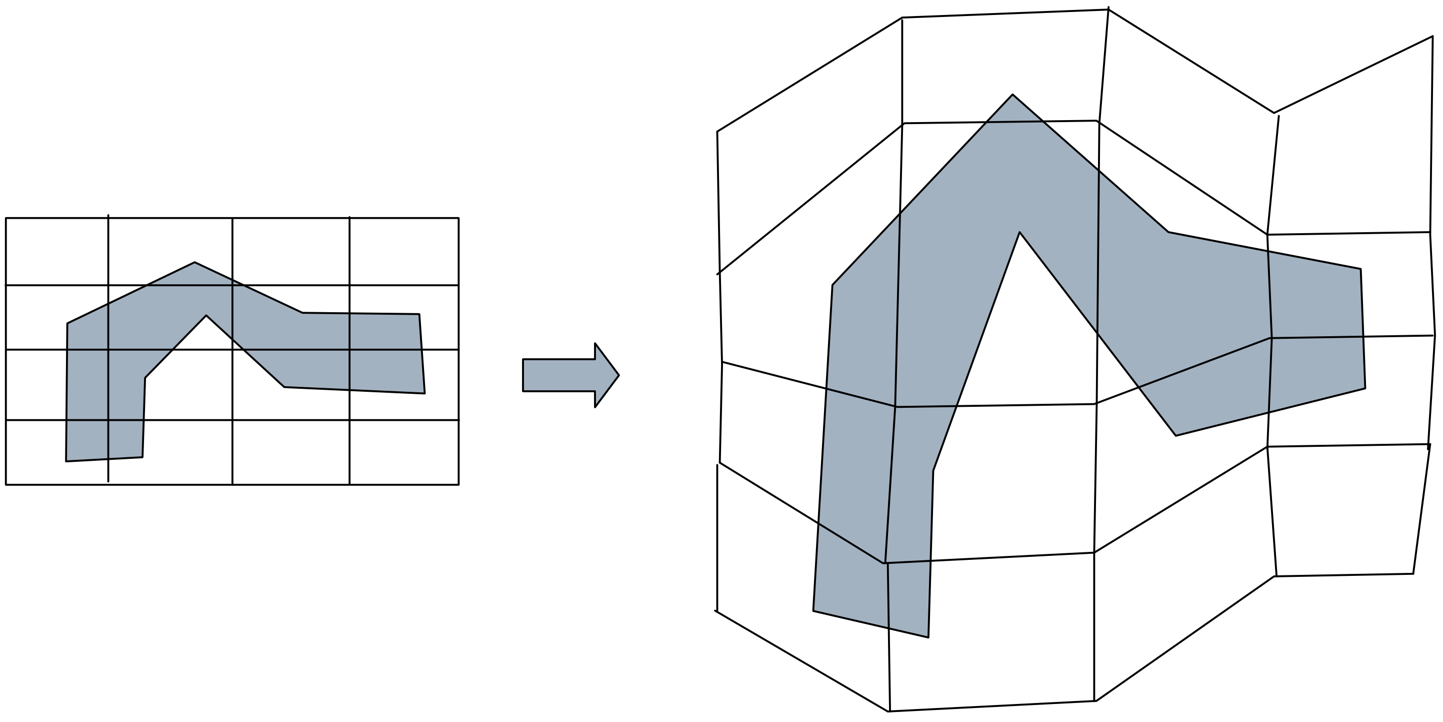


easy

difficult

Grid Deformation

Deformation used for film *Faim (NFB 1974)* using bilinear interpolation within grid cells.

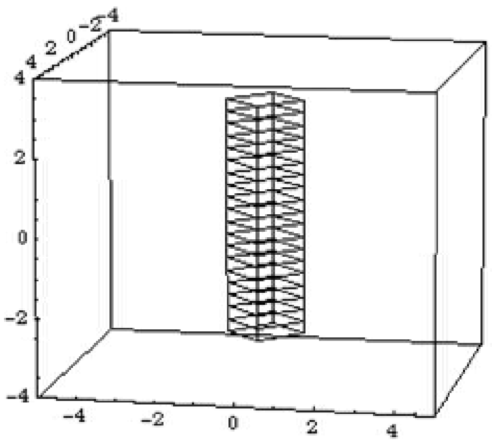


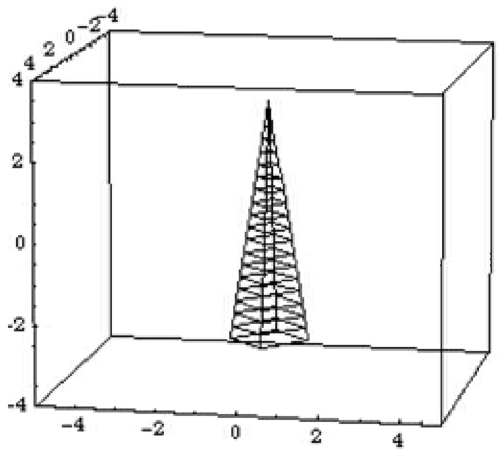
Notion of spatial deformation

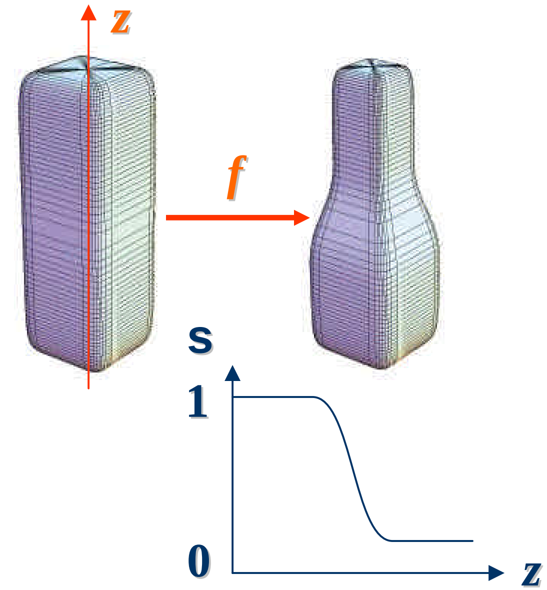
Bezier introduced the idea of deforming shape through a mapping implemented as a free-form (tensor product) spline.

Nonlinear deformations

Barr 1984: Apply an affine transformation whose parameters vary spatially.



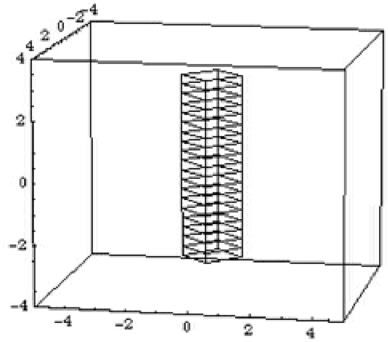


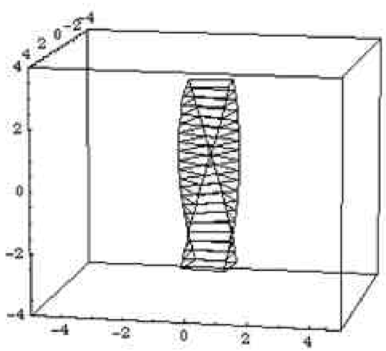


Taper: $\mathbf{p'=S_{xy}p}$, where the scale value **s** in the **xy** plane $\mathbf{S_{xy}}$, is a function of **p.z**, $\mathbf{s(z) = \frac{(maxz - z)}{(maxz - minz)}}$.

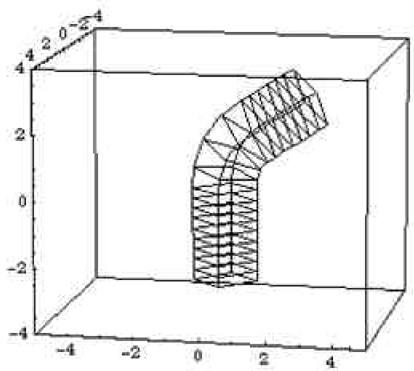
Nonlinear deformations

Barr 1984: Apply an affine transformation whose parameters vary spatially.

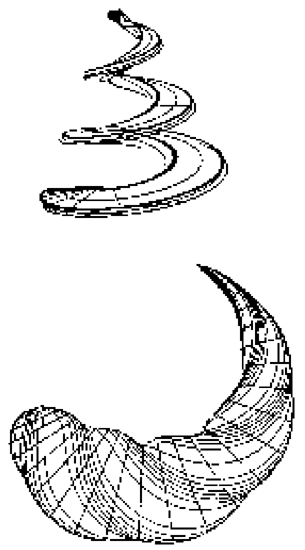




Twist



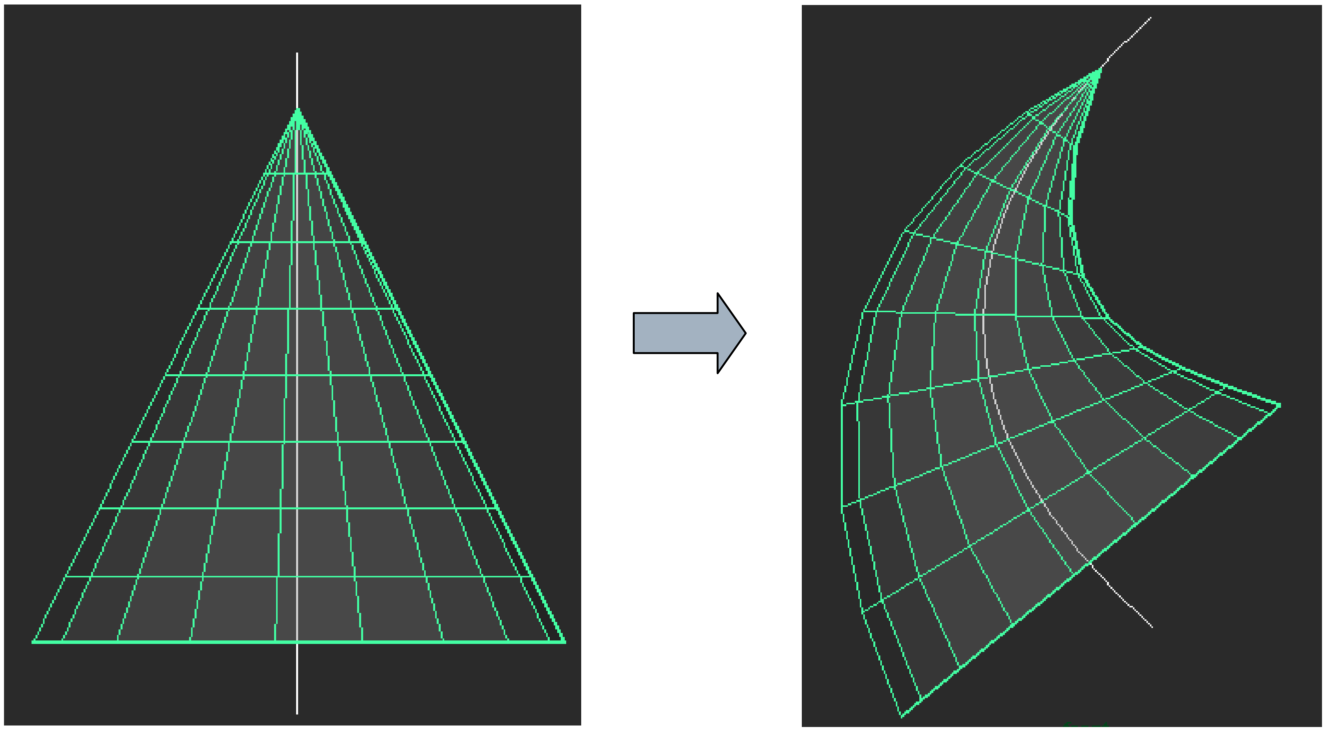
Bend



Compound

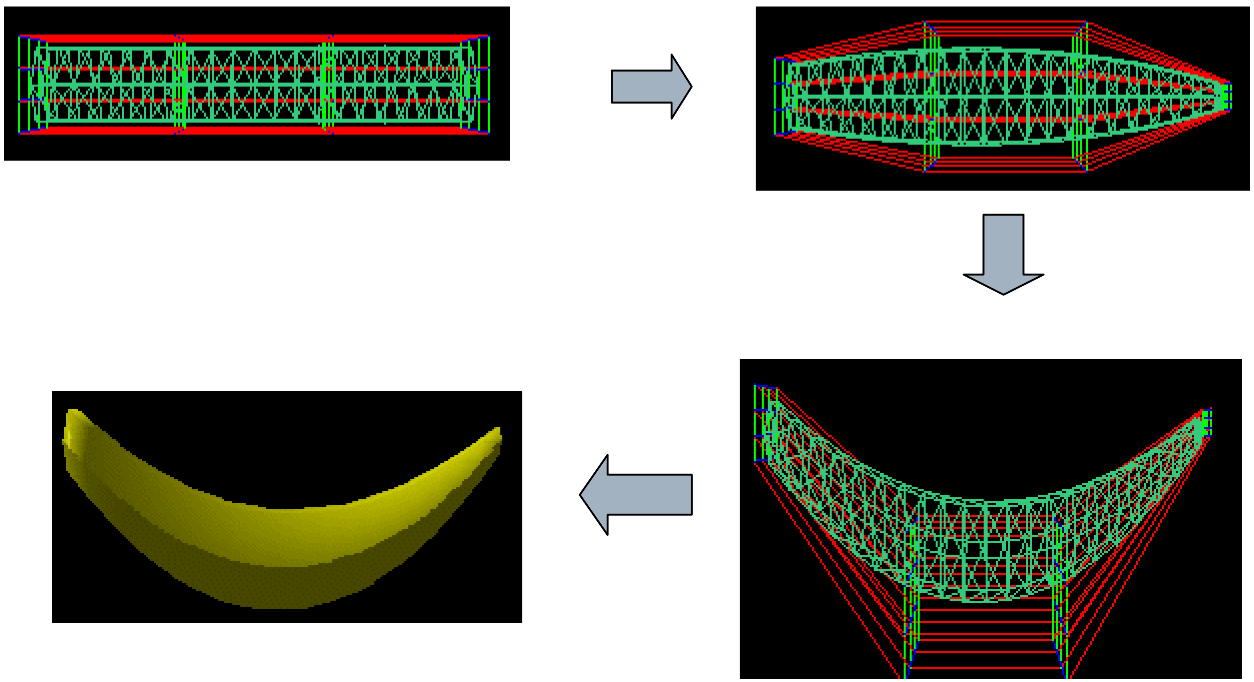
Axial Deformations

Lazarus et al 1994: Apply the transformation of a proximal reference frame along a curve.



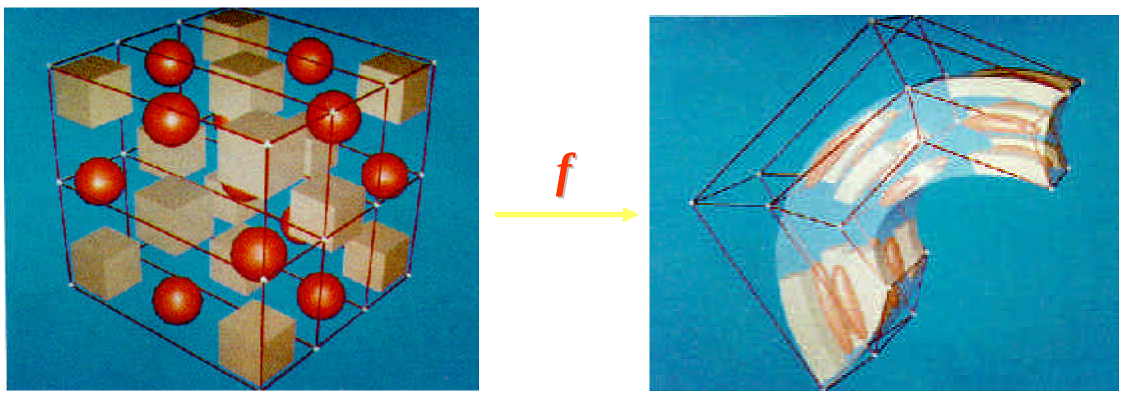
Free-form Deformations

Sederberg and Parry 1986: Grid deformation in 3D using tricubic Bezier interpolation.



FFD Algorithm

- Define a local coordinate frame on a parallelopiped.
- Compute the **(s,t,u)** coordinates of any point in the parallelopiped.
- Impose an **(l,m,n)** grid of control points on the parallelepiped.
- Move the control points around.
- Evaluate new position of model point based on trivariate Bersntein polynomials or other type of volumes.



[T.Sederberg & S.Parry'86]

FFD Algorithm

- A point in the STU coordinate system:

$$\mathbf{X} = \mathbf{X}_0 + s\mathbf{S} + t\mathbf{T} + u\mathbf{U}.$$

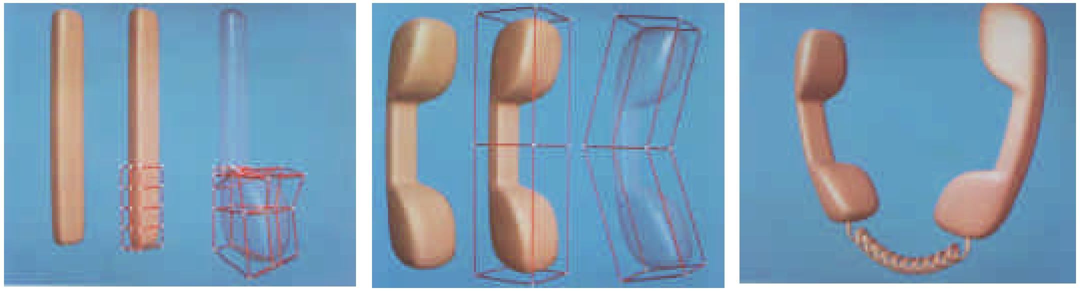
$$s = \frac{\mathbf{T}\times\mathbf{U}\cdot(\mathbf{X} - \mathbf{X}_0)}{\mathbf{T}\times\mathbf{U}\cdot\mathbf{S}},\quad t = \frac{\mathbf{S}\times\mathbf{U}\cdot(\mathbf{X} - \mathbf{X}_0)}{\mathbf{S}\times\mathbf{U}\cdot\mathbf{T}},\quad u = \frac{\mathbf{S}\times\mathbf{T}\cdot(\mathbf{X} - \mathbf{X}_0)}{\mathbf{S}\times\mathbf{T}\cdot\mathbf{U}}$$

$$0 < s < 1,\ 0 < t < 1 \text{ and } 0 < u < 1.$$

- A deformed point:

$$\mathbf{X}_{ffd} = \sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^i \left[\sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j \left[\sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k \mathbf{P}_{ijk} \right]\right]$$

FFD Shortcomings



[T.Sederberg & S.Parry'86]

- Parallelopiped lattice shape limits deformation shape.
- No control over spatial distribution of control points.
- Bezier basis functions have a global influence.
- Discontinuity in deformed space at lattice boundary. How do overlapping lattices deform space?

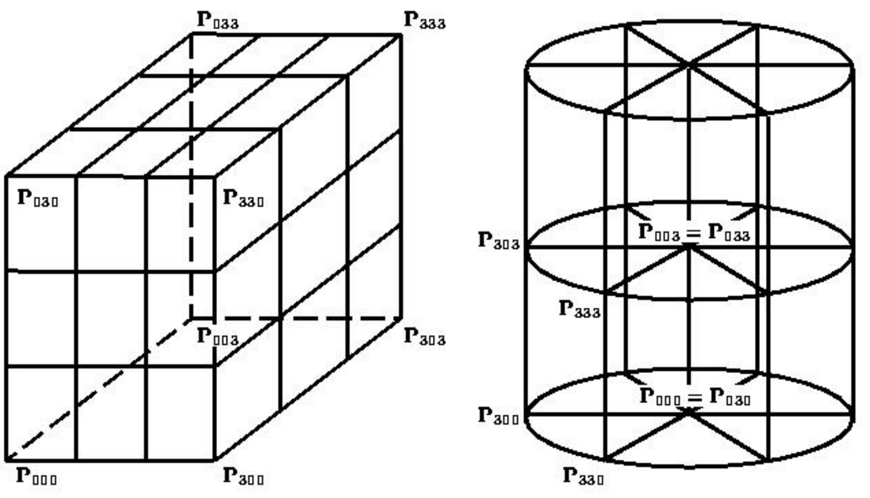
FFD Shortcomings

- B-spline basis FFD (*Purgathofer*).
- Extended FFD (*Coquillart*)
- deCasteljau FFD (*Chang & Rockwood*)
- Catmull-Clark FFD (*McCracken & Joy*)
- Dirichlet FFD (*Moccozet & Thalmann*)

Extended FFD

- Edit the lattice before associating the model with it.

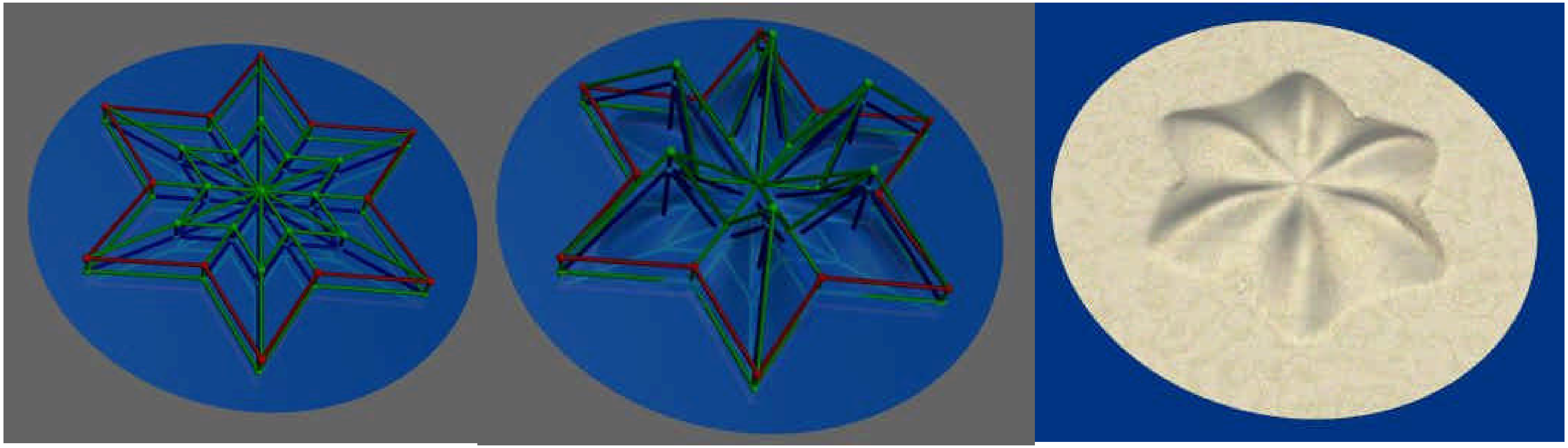
- Arbitrary lattices:
  - Prismatic lattices
  - Tetrahedral
  - ...



(a) (b)

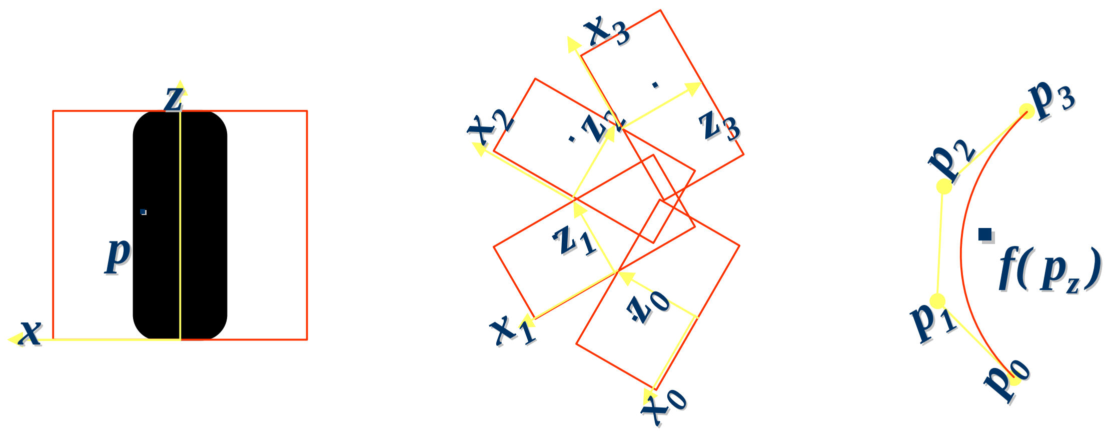
Computing (s, t, u) coordinates

- Subdivision.
- Newton iteration.
- Projection (limited, but fast).



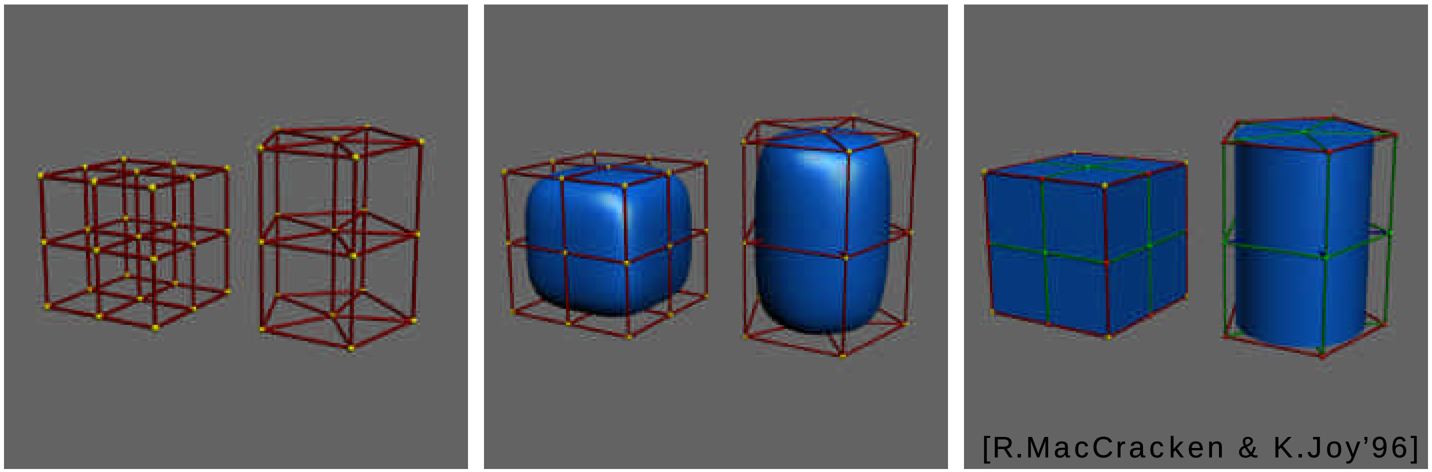
De Casteljau FFD

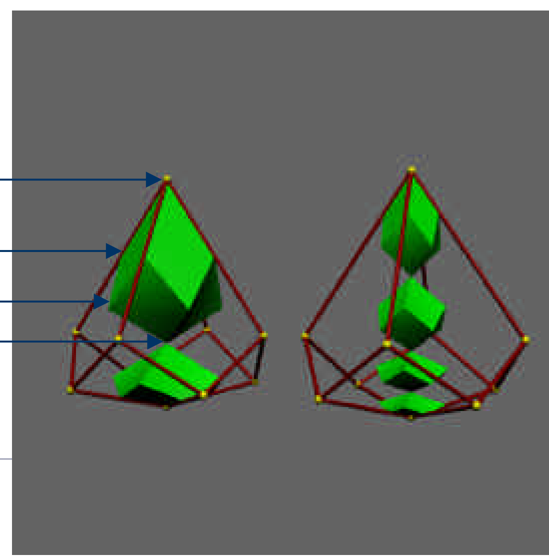
- Sequence of affine transformations



Catmull-Clark FFD

- Lattice = subdivision volume.



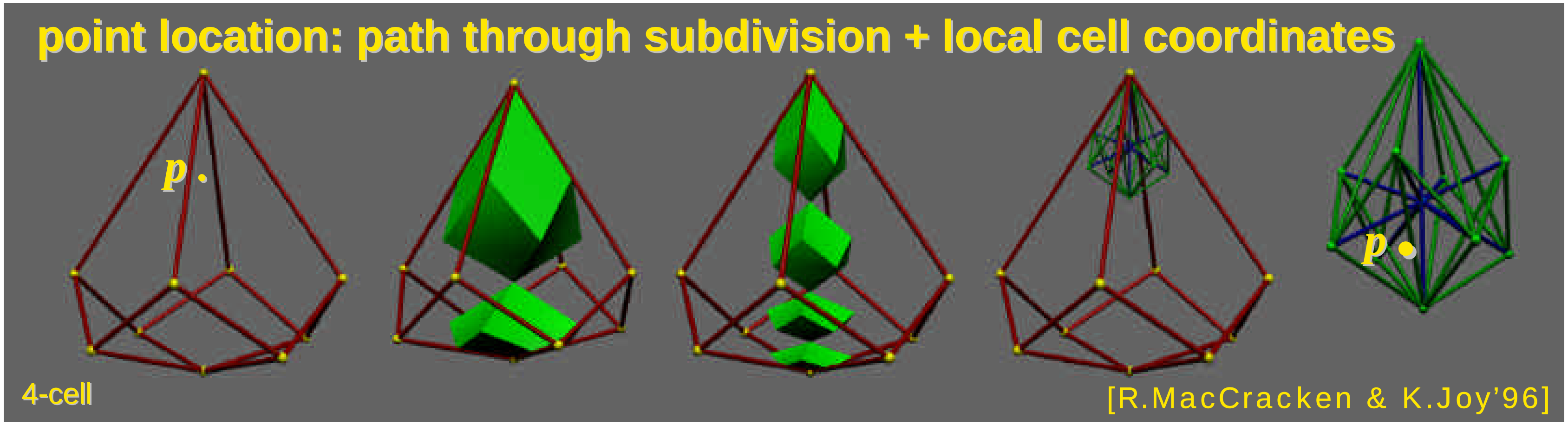


How do we parameterize a point in the lattice?

Dynamic Graphics Project
University of Toronto
www.dgp.toronto.edu

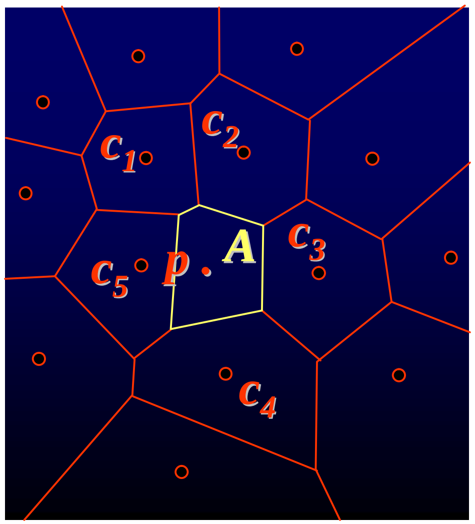
Catmull-Clark FFD

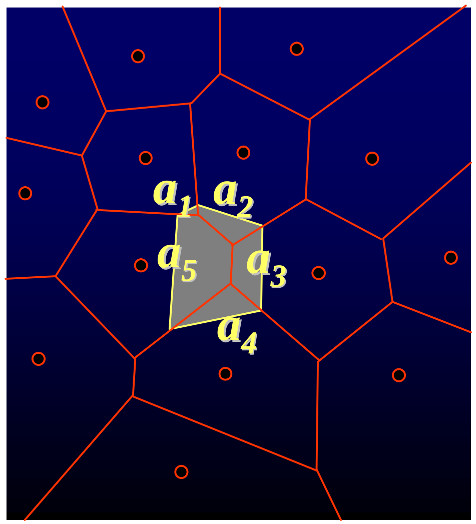
- Find position of ***p*** in lattice and hold local coordinates.
- Move lattice.
- Trace new position of ***p*** in new lattice using local coordinates.

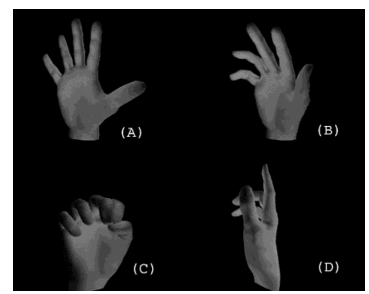


Dirichlet FFD

- Lattice implied by Voronoi space.
- Sibson coordinates used as a linear interpolant.
- Smoothed by multivariate Bernstein polynomials.

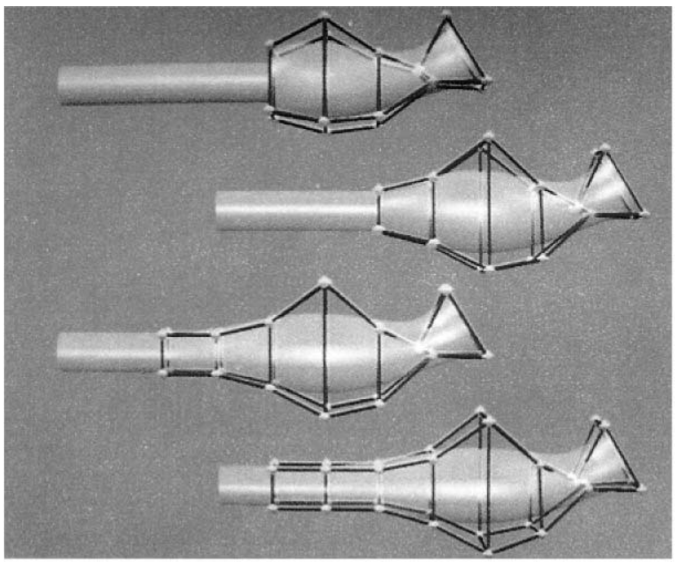


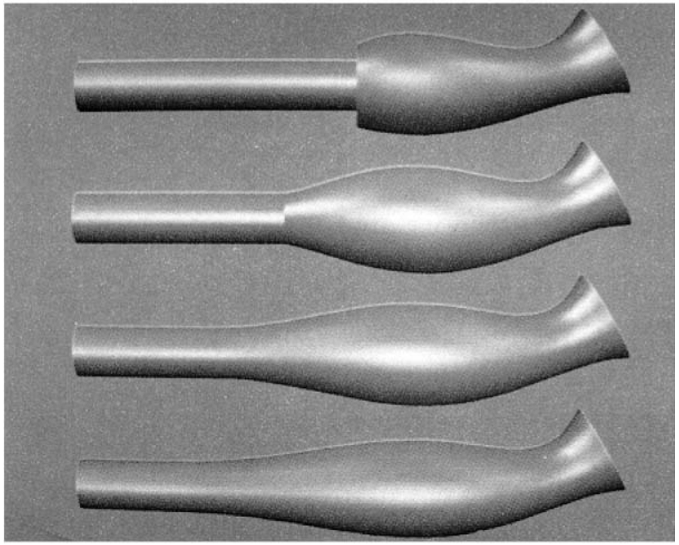




[L.Moccozet & N.Magnenat-Thalman'97

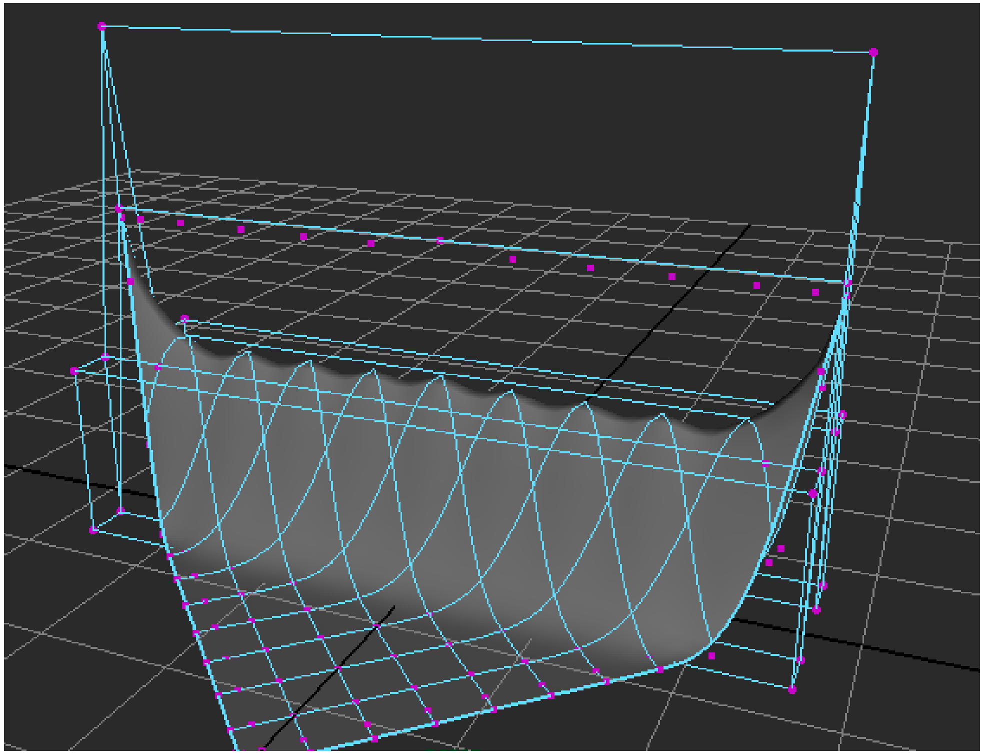
Continuity, overlap at lattice boundaries





Space deformations and surface resolution

- Surface resolution and topology is still a problem!



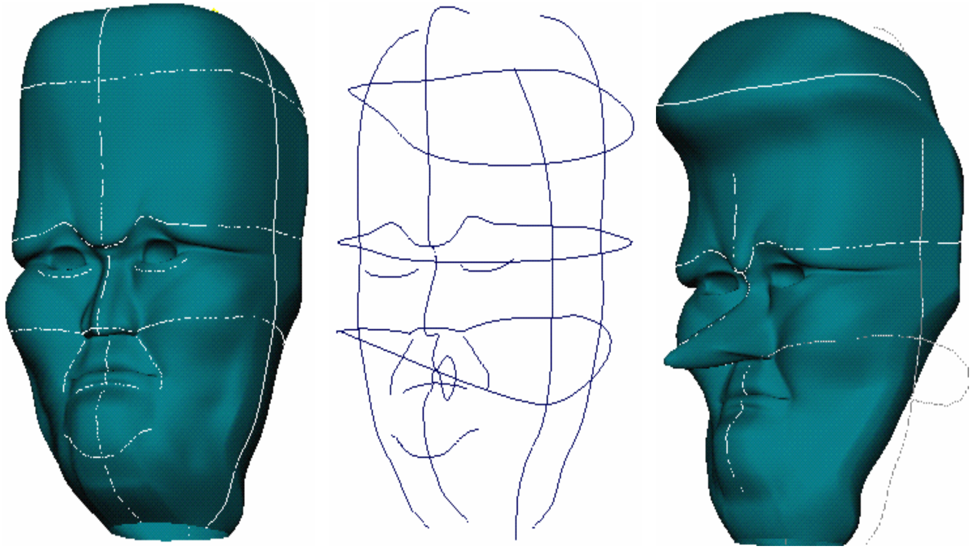
Coffeeeeeeeeeeeeee...

Motivation for Wires and wraps

- To provide a direct object deformation tool that is independent of the object's underlying geometric representation.
- To provide a minimalist visual model of the object that highlights the important deformable features of the object.

Inspiration

- Wires define and control an object's features like a sculptor's armature.
- Projections of wire curves directly map to sketches or line drawings of the object.



Contributions

- Effective deformation technique employing space curves and implicit functions (wire).
- An implicit function primitive defined by space curves (wire).
- Effective deformation technique using polymeshes (wrap).
- Multiresolution modeling appeal of subdivision surfaces made applicable to arbitrary surface representations (wrap).
- Construction of the deformer from underlying geometry can be automated (wire,wrap).
- An efficient and controlled approach to the aggregation of multiple deformations (wire,wrap).

Wire Overview

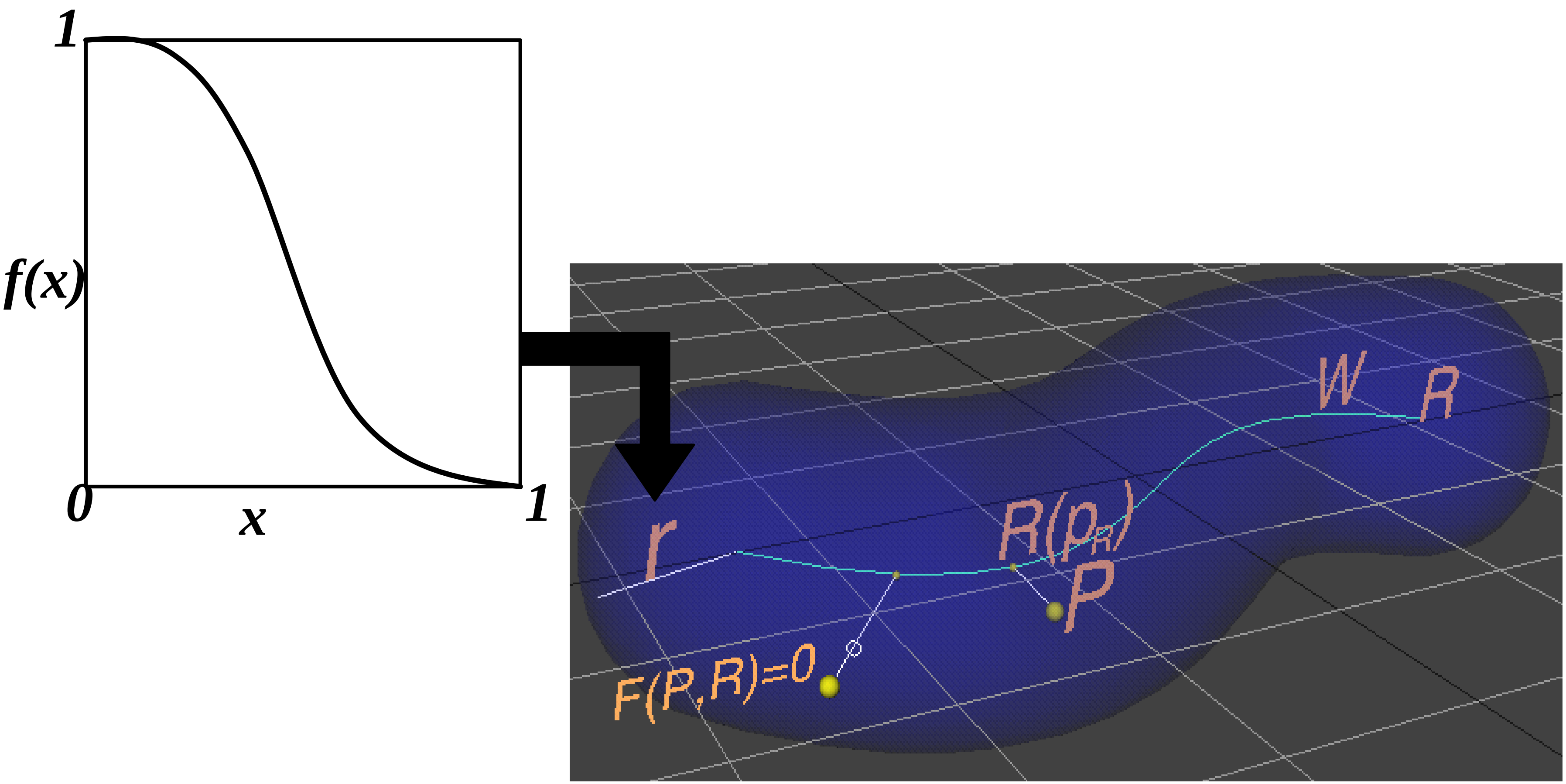
- Wire definition and algorithm.
- Spatial control of wire parameters.
- Multiple wire interaction.

Wire definition

- A free-form curve whose manipulation deforms an associated object (or space).

  - ***W*** :*The wire curve*. A free-form curve representing the wire**.**
  - ***R*** : *The reference curve*. A copy of ***W*** is made when objects are bound to the wire.
  - ***r*** : *Radius of influence* around the wire**.**
  - ***s*** : Radial *scaling factor* around the wire.
  - ***f*** : Scalar sigmoid function $\mathbb{R}^{+}$ -> **[0,1]** *(density function).*

Wire definition



Wire algorithm *(Binding)*

- Binding an object to a wire **< *W,R,r,s,f* >**

- For every point ***P*** representing the object:

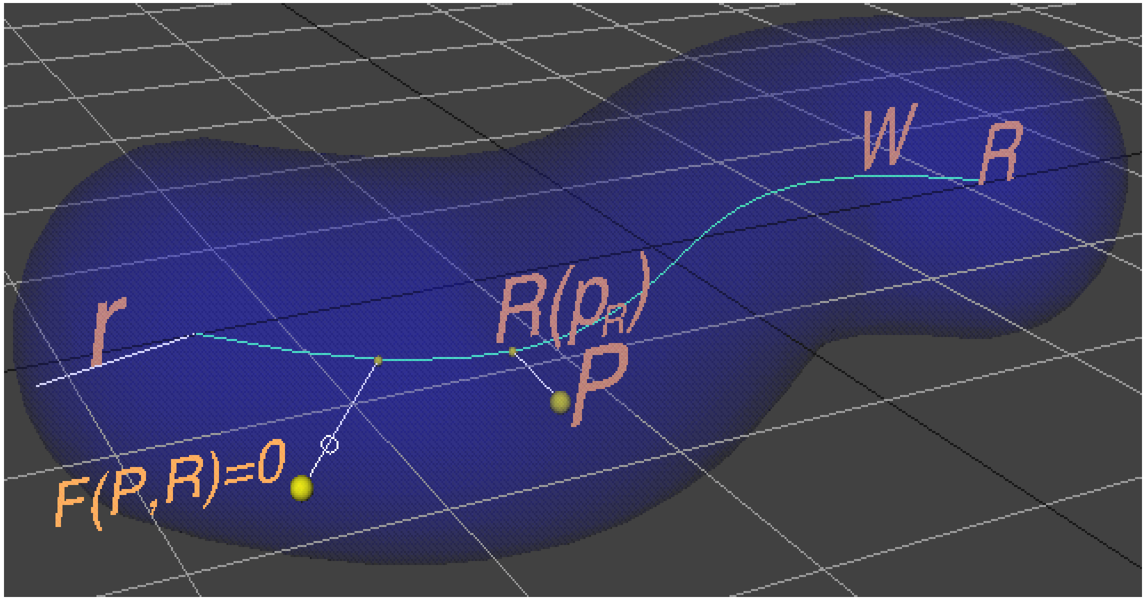
  - Calculate p_R, the parameter value corresponding to the Euclidean closest point to ***P*** on curve ***R***, $R(p_R)$.

  - Calculate ***F(P,R)***, the influence function for curve ***R*** at ***P.***

Wire algorithm *(Binding)*

Influence function ***F(P,R)*** of a wire ***< W,R,r,s,f >*** at a point ***P*** :

$$F(P,R) = f\left(\frac{\|P-R(p_R)\|}{r}\right).$$



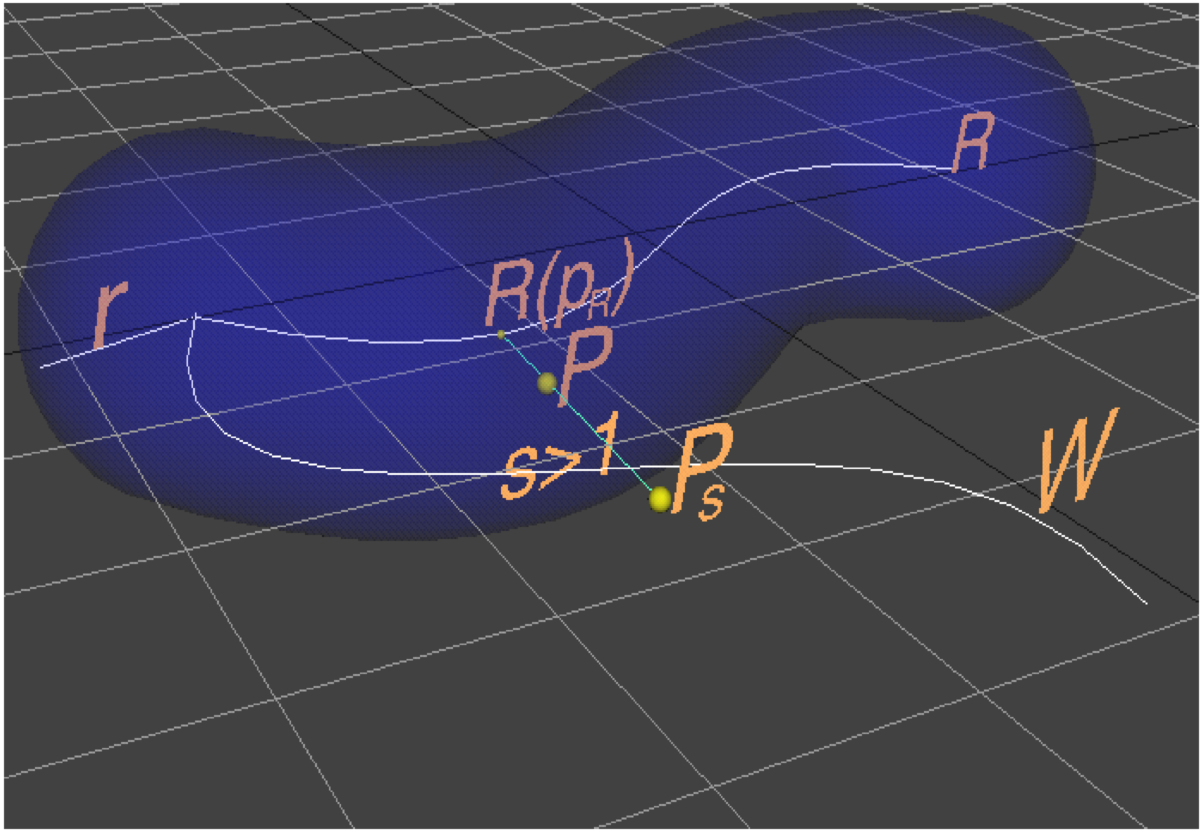
Wire algorithm *(Deformation)*

- Deforming an object by the wire **$< W,R,r,s,f >$**

- For every point ***P*** representing the object:
  - Scale ***P*** by a factor of ***s*** radially around ***R***.
  - Rotate the result around $R(p_R)$ by the angle between the tangents to curves ***W*** and ***R*** at parameter value p_R.
  - Translate the resulting point by the relative translation of points on curve ***W*** and ***R*** at parameter value p_R.

Wire algorithm *(Deformation)*

- Deformation of a point ***P*** by a wire ***<W,R,r,s,f>***

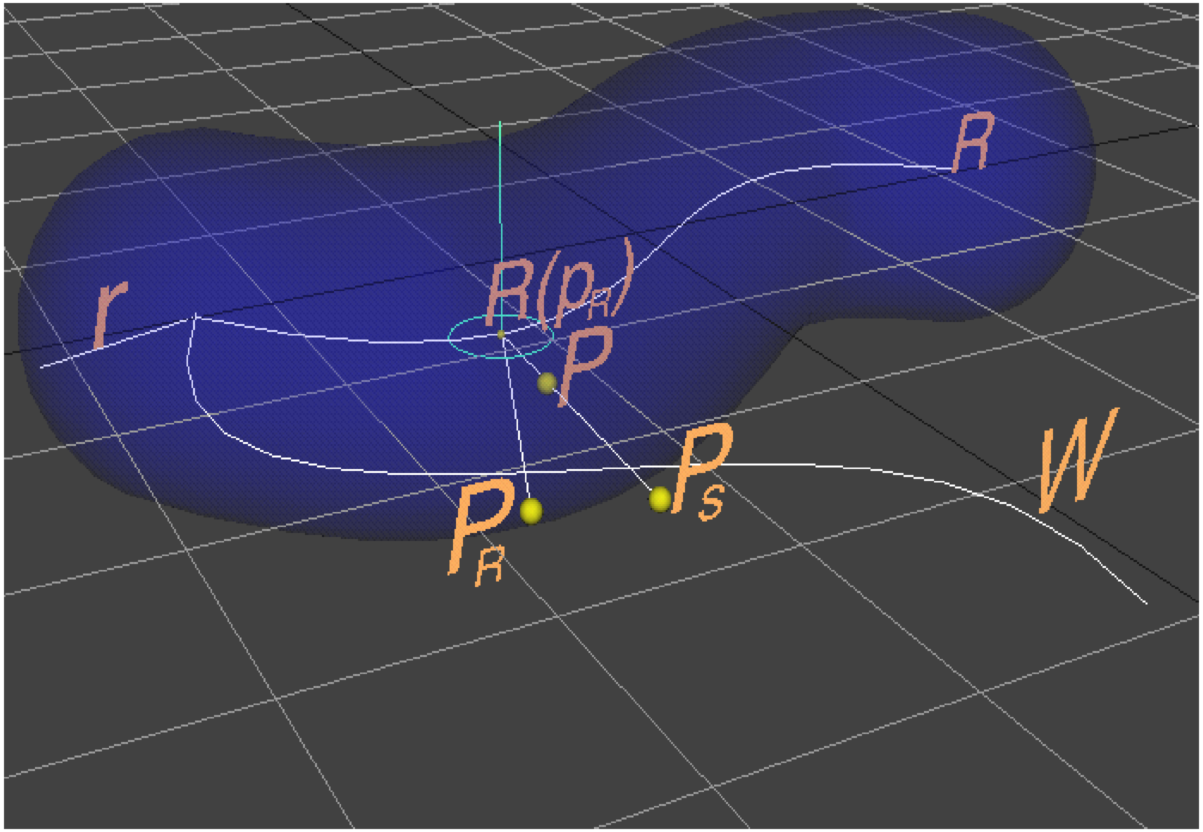


Scale

dgp Dynamic Graphics Project
University of Toronto
www.dgp.toronto.edu

Wire algorithm *(Deformation)*

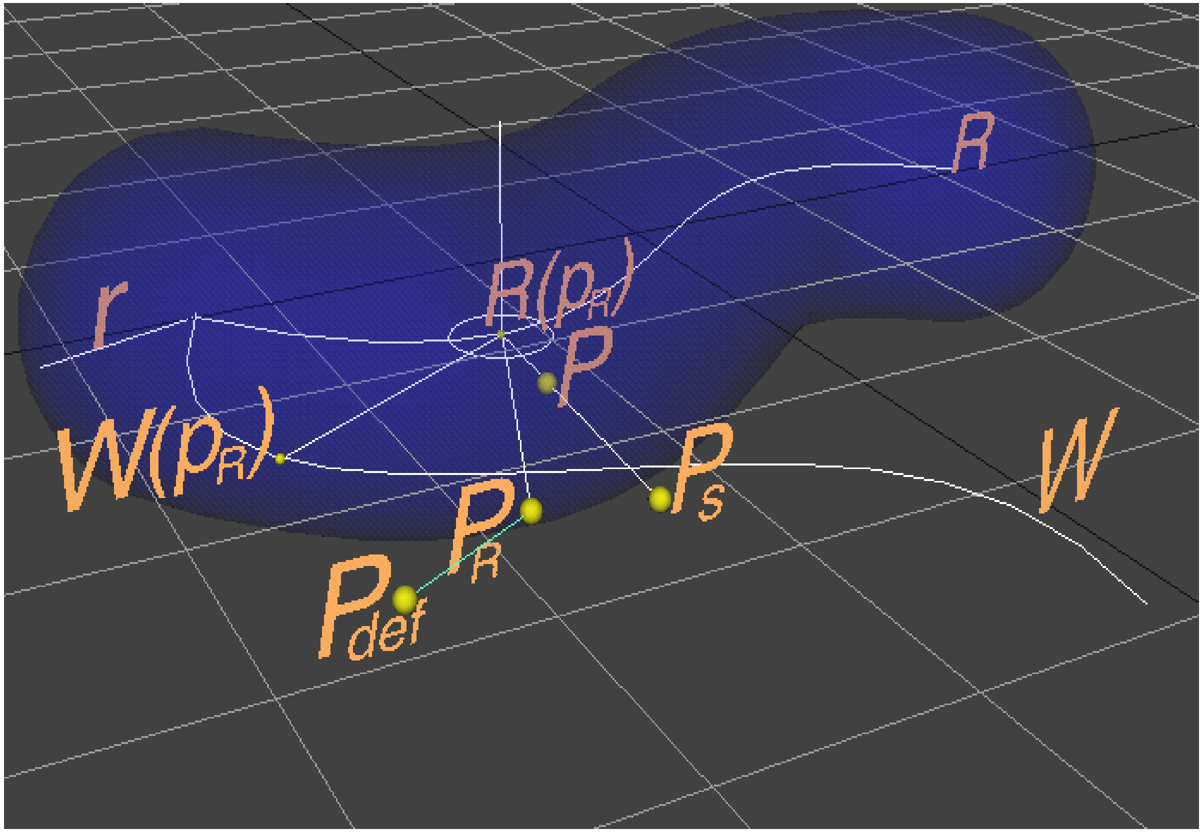
- Deformation of a point ***P*** by a wire ***<W,R,r,s,f>***



Rotate

Wire algorithm *(Deformation)*

- Deformation of a point ***P*** by a wire ***<W,R,r,s,f>***



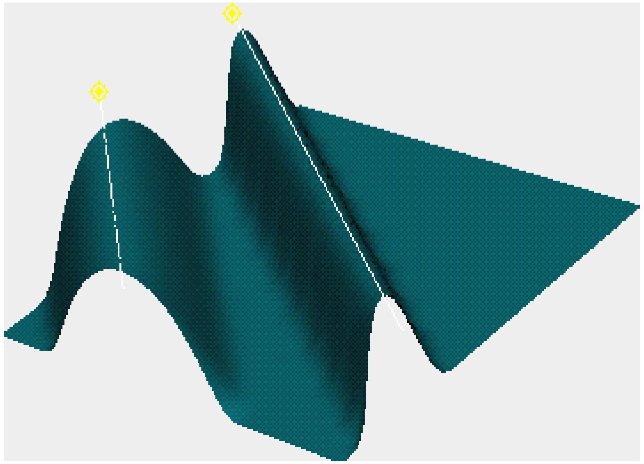
Translate

Wire algorithm *(Deformation)*

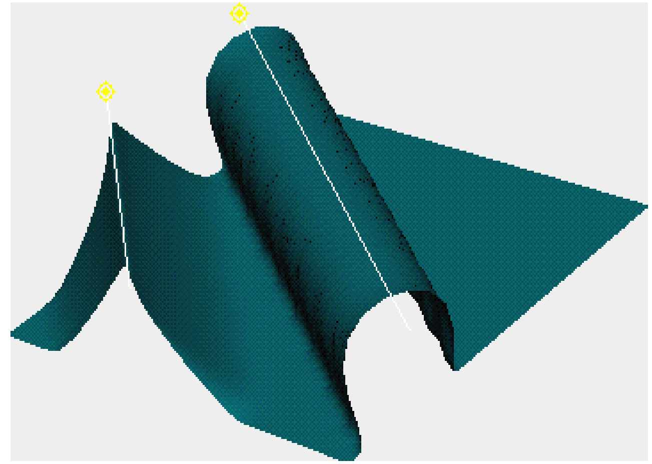
- Deforming an object by the wire **< *W,R,r,s,f* >**

- For every point ***P*** representing the object:
  - Scale ***P*** by a factor of ***s*** radially around ***R.***
  - Rotate the result around $R(p_R)$ by the angle between the tangents to curves ***W*** and ***R*** at parameter value p_R.
  - Translate the resulting point by the relative translation of points on curve ***W*** and ***R*** at parameter value p_R.

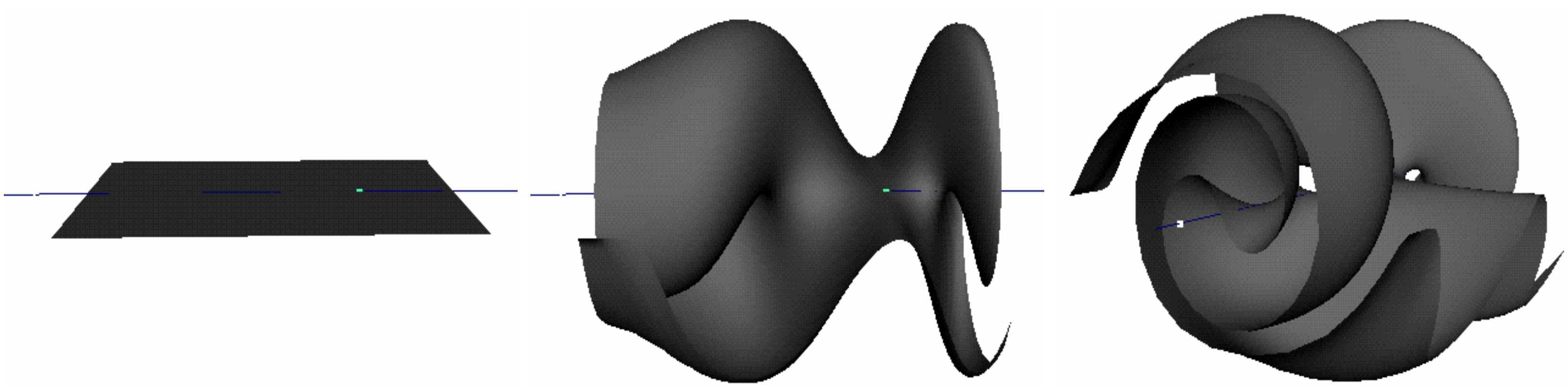
Wire algorithm parameters



Varying *r*



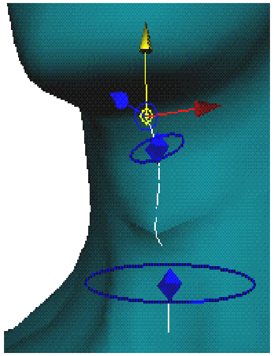
Varying *s*



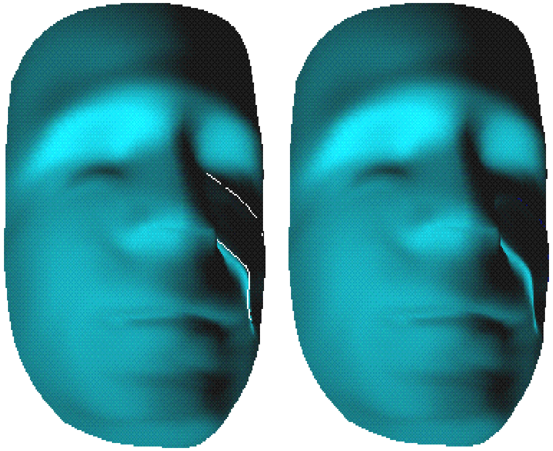
Tangency and Twist

Spatial control of wire parameters

- **Locators** : define wire parameter values by interpolating along the wire curve.

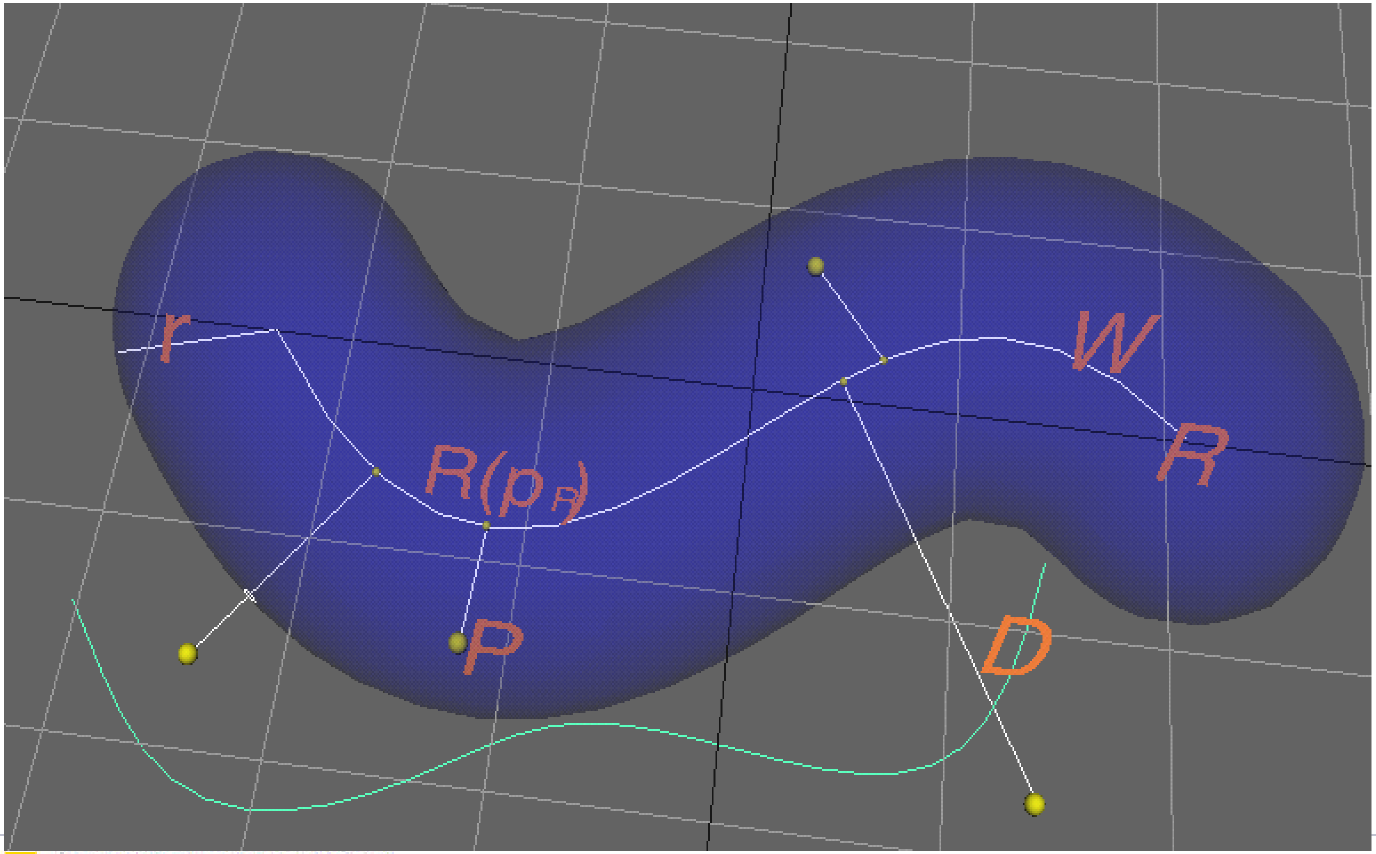


- **Domain Curves** : spatially define wire parameter values using free-form curves.



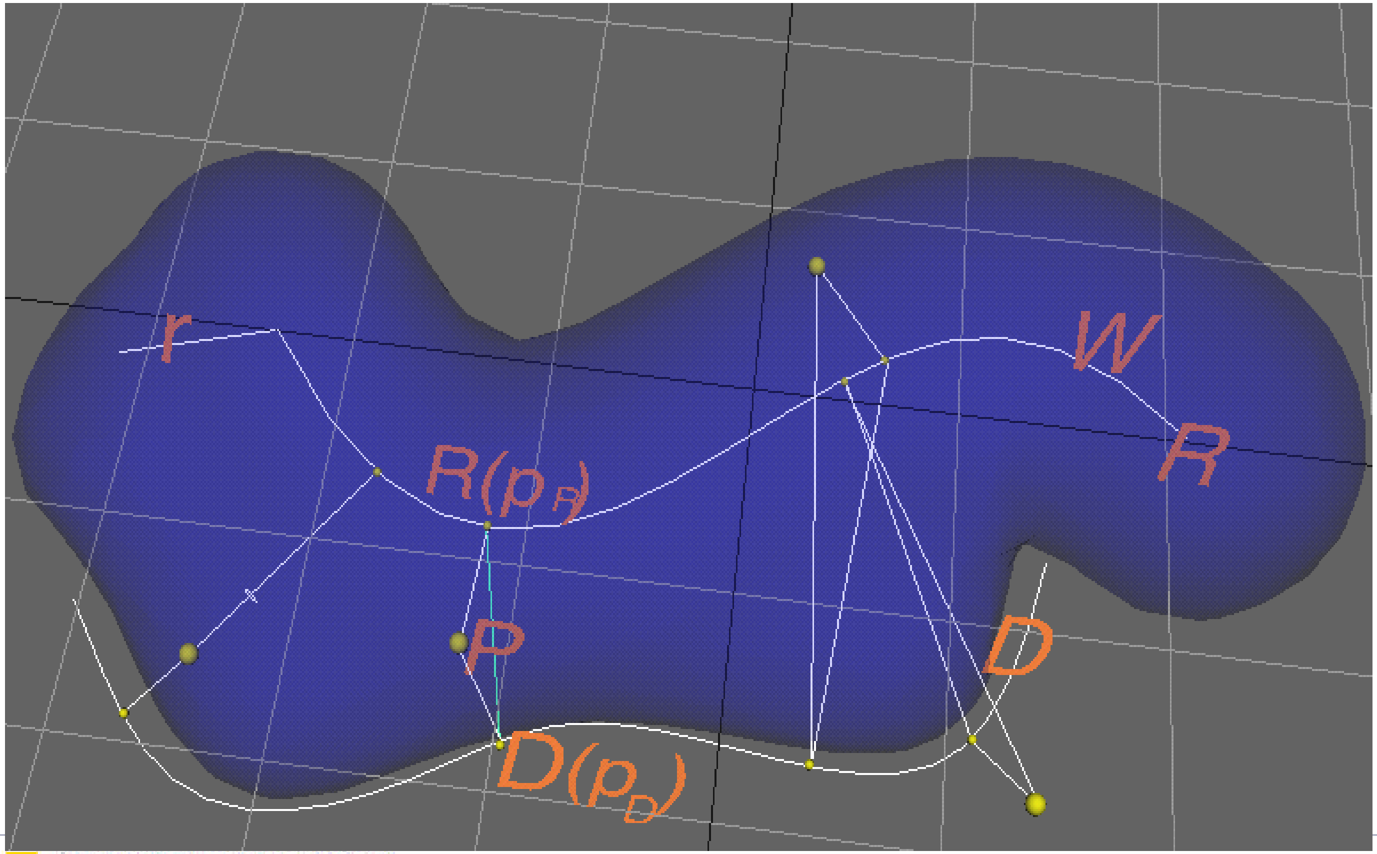
Domain curve algorithm

•



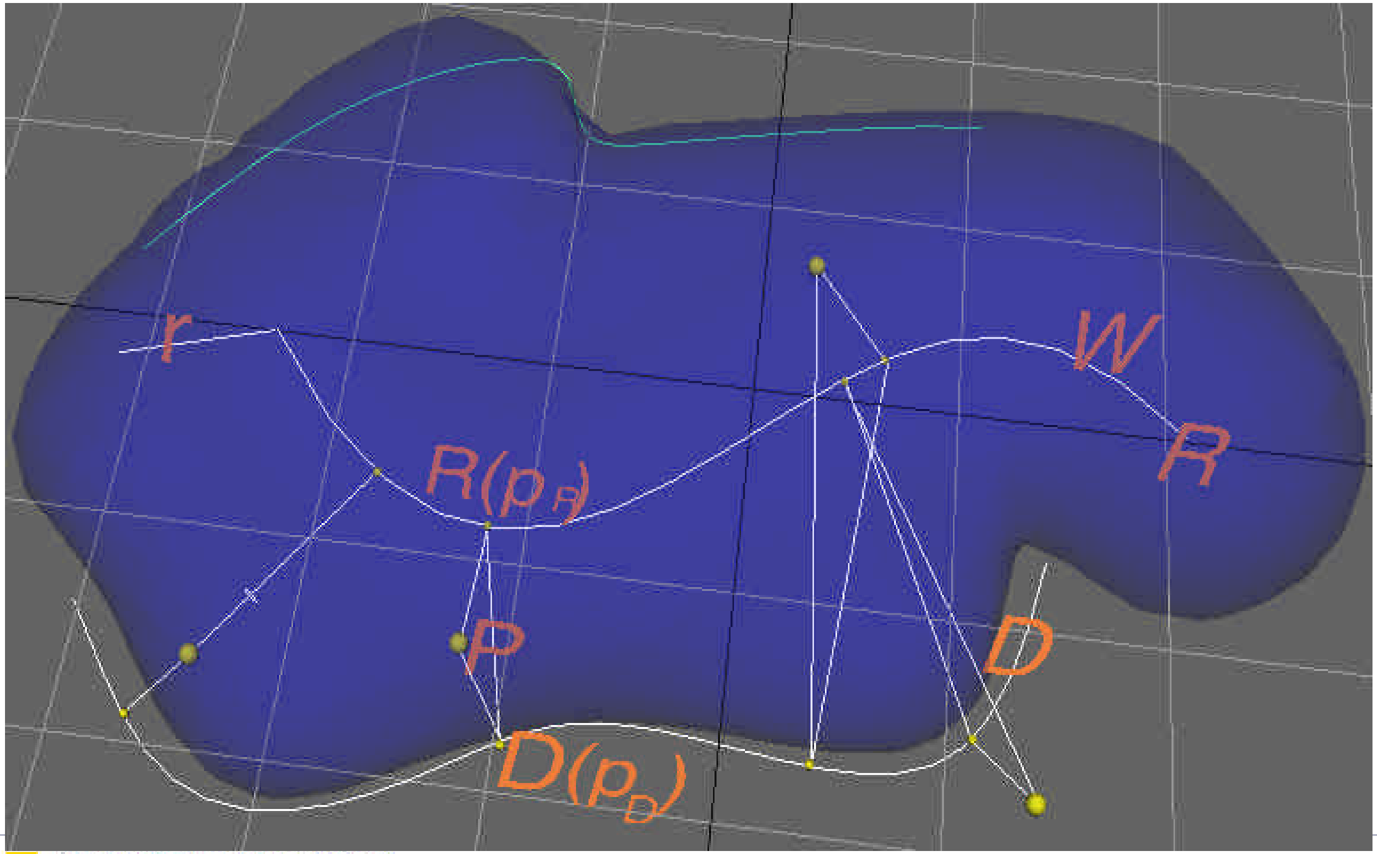
Domain curve algorithm

•



Domain curve algorithm

•



Multiple Wires

• A smooth union of individual wire deformations defines overall shape **(Sculptor's armature metaphor)** :

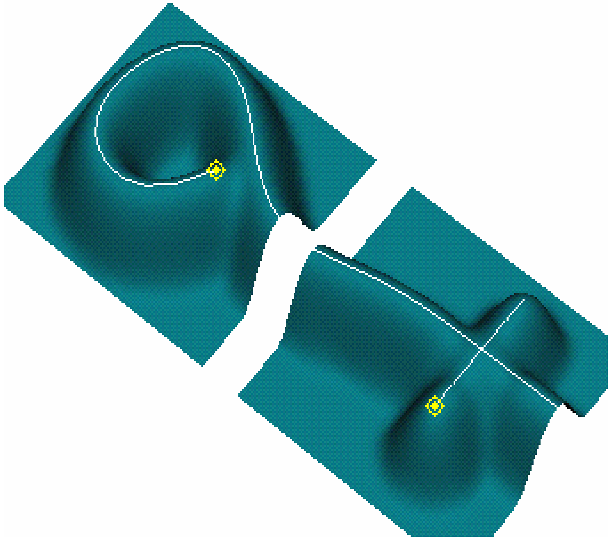
Let the i^{th} wire displace a point $\mathbf{P}$ by $D\mathbf{P}_i$ **:**

$$\mathbf{P}_{def} = \mathbf{P} + \frac{\underset{i=1}{\overset{n}{S}} (\|D\mathbf{P}_i\|^{m} D\mathbf{P}_i)}{\underset{i=1}{\overset{n}{S}} (\|D\mathbf{P}_i\|^{m})}$$

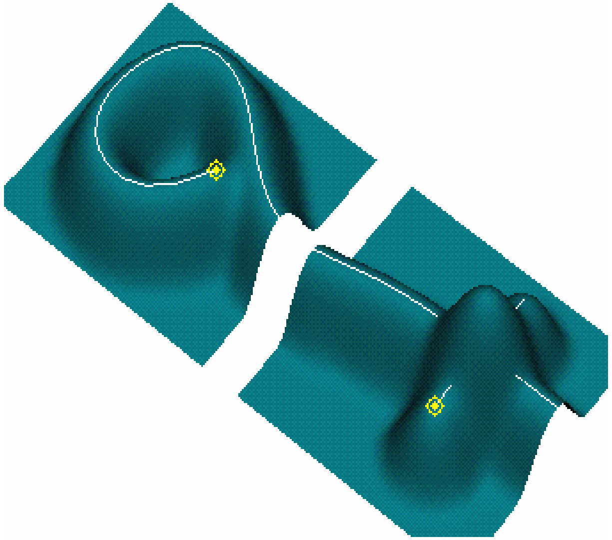
Behavior varies from an average at m=0 to max($D\mathbf{P}_i$) for large m.

Multiple Wires

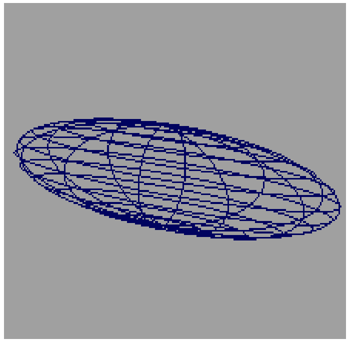
•



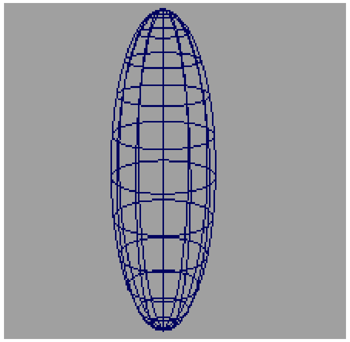
Integrated deformation



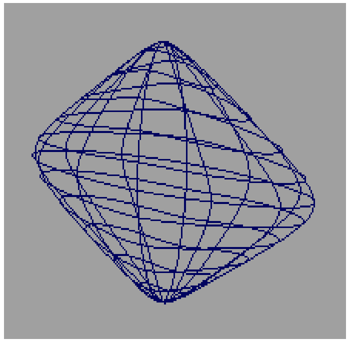
Additive deformation



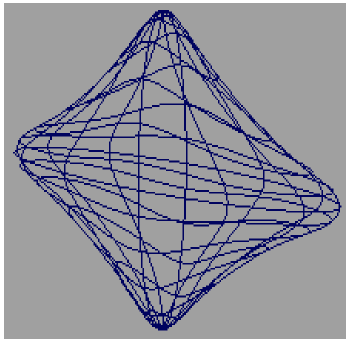
deformation 1



deformation 2



m=1



m=5

Multiple Wires *(local control)*

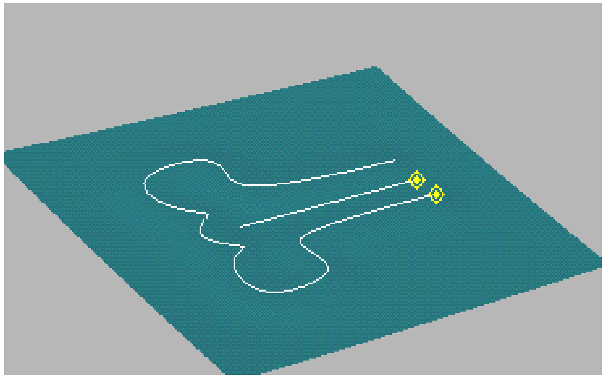
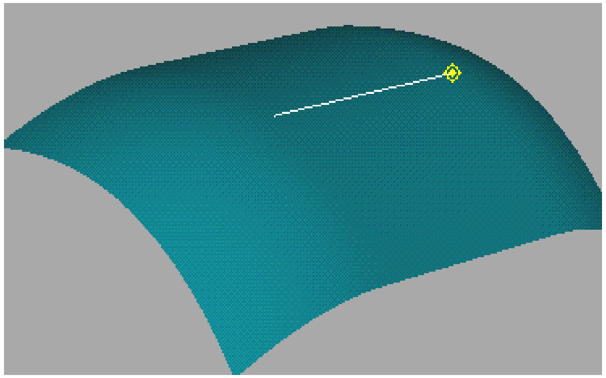
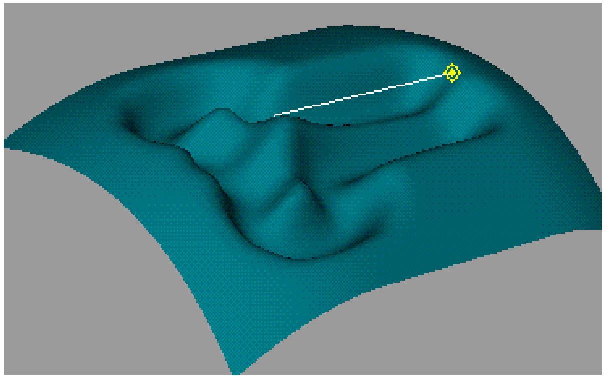
•Control for wire curves in the proximity of the curve to limit global deformation in a region :

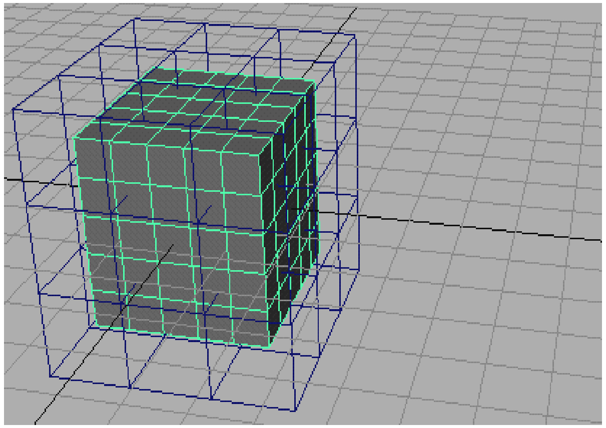
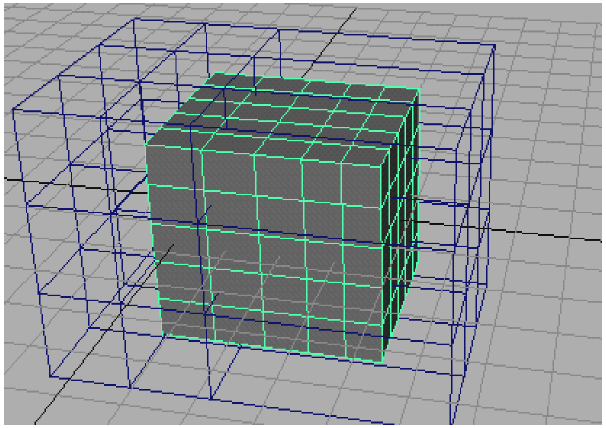
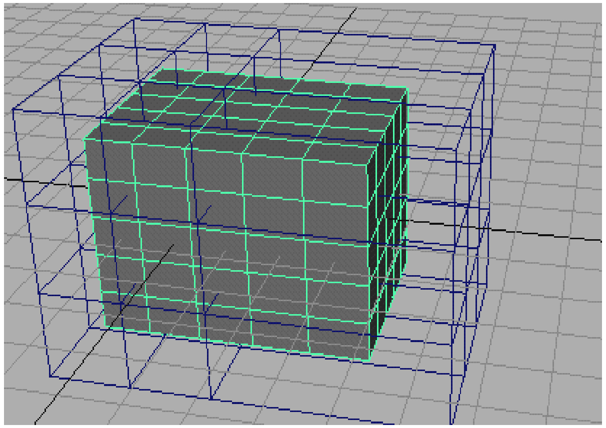
Let the i^{th} wire displace a point $\mathbf{P}$ by $\Delta\mathbf{P}_i$:

$$\mathbf{P}_{def} = \mathbf{P} + \frac{\sum_{i=1}^{n} (F(\mathbf{P},\mathbf{R}_i)^k \, \Delta\mathbf{P}_i)}{\sum_{i=1}^{n} F(\mathbf{P},\mathbf{R}_i)^k}$$

Behavior in a region of the object is strongly influenced by proximal wires for large k.

Multiple Wires *(local control)*

Global control **Local control**

Wrap definition

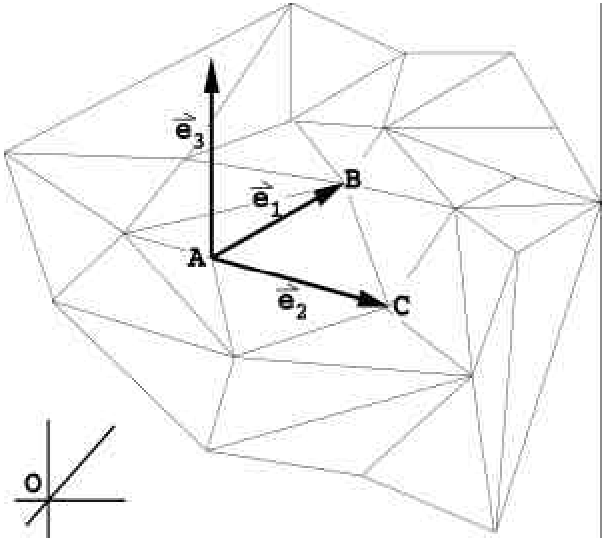
- A polymesh whose manipulation deforms an associated object (or space).
  - ***D***:*The wrap deformer mesh*.
  - ***R*** : *The reference mesh*. A copy of ***D*** is made when objects are bound to the wrap.
  - ***r*** : *Radius of influence* around the mesh***.***
  - ***local*** : Scalar to control locality of deformations.
  - ***f*** : Scalar sigmoid function $\mathbb{R}^{+}$->**[0,1]** *(density function).*

Wrap algorithm *(Binding)*

- Every face of the mesh ***R*** is a control element ***k***.
  - Influence function ***F(P,k)*** of a wrap at a point ***P*** :

    $$F(P,k) = f(\,dist(P,k)).^{local}$$

  - ***P*** is computed in a coordinate system local to control element ***k.***



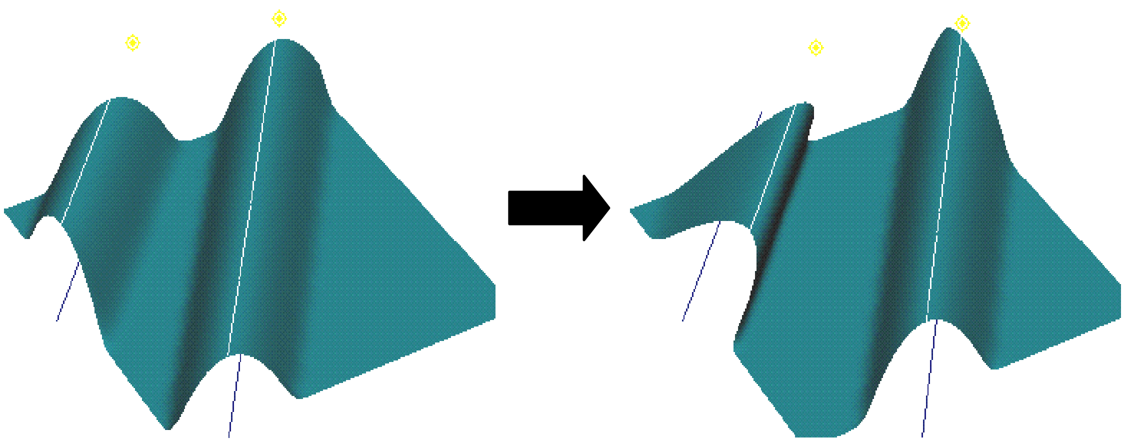
Wrap algorithm *(Deformation)*

- Every face of the mesh ***D*** is a control element ***k***.
  - ***P*** is deformed to preserve the local coordinates computed when bound to the control element ***k.***
  - The overall deformation to point ***P*** is a weighted - average of the deformation from all control elements based on their influence function.

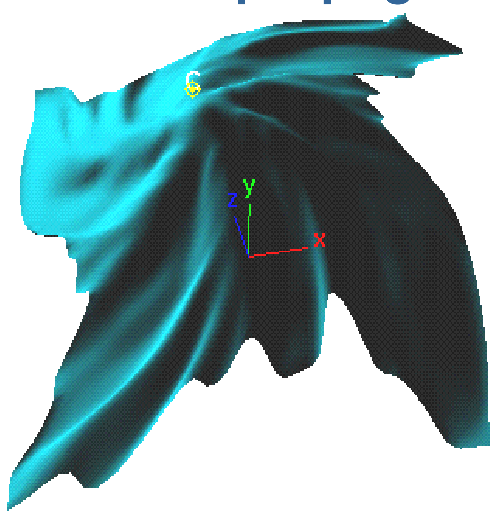
Example Applications

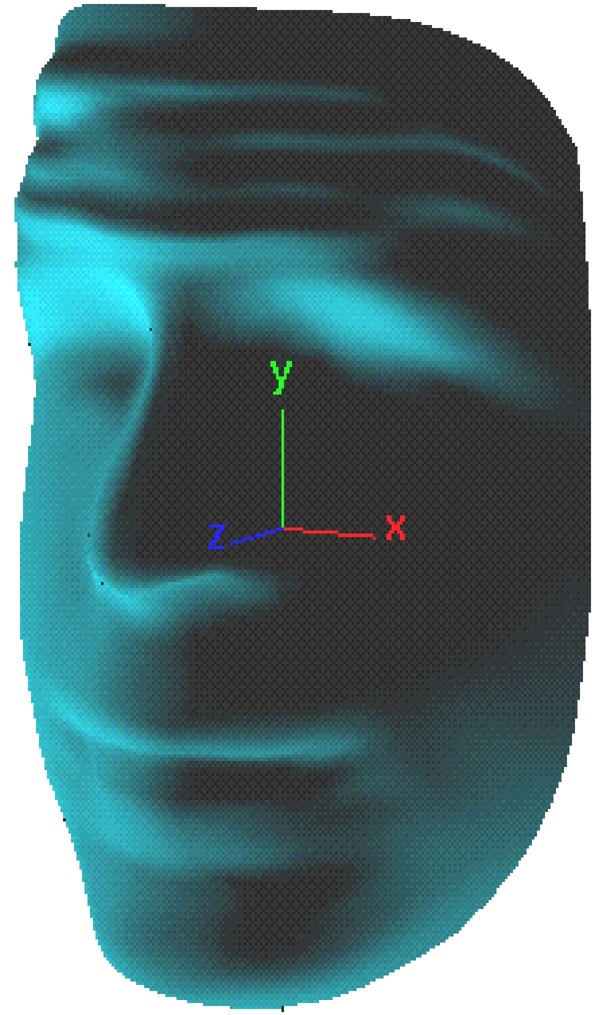
- Facial animation.
- Wrinkles.
- Kinematics for flexible skeletons.
- Character skinning workflow.

Wrinkles

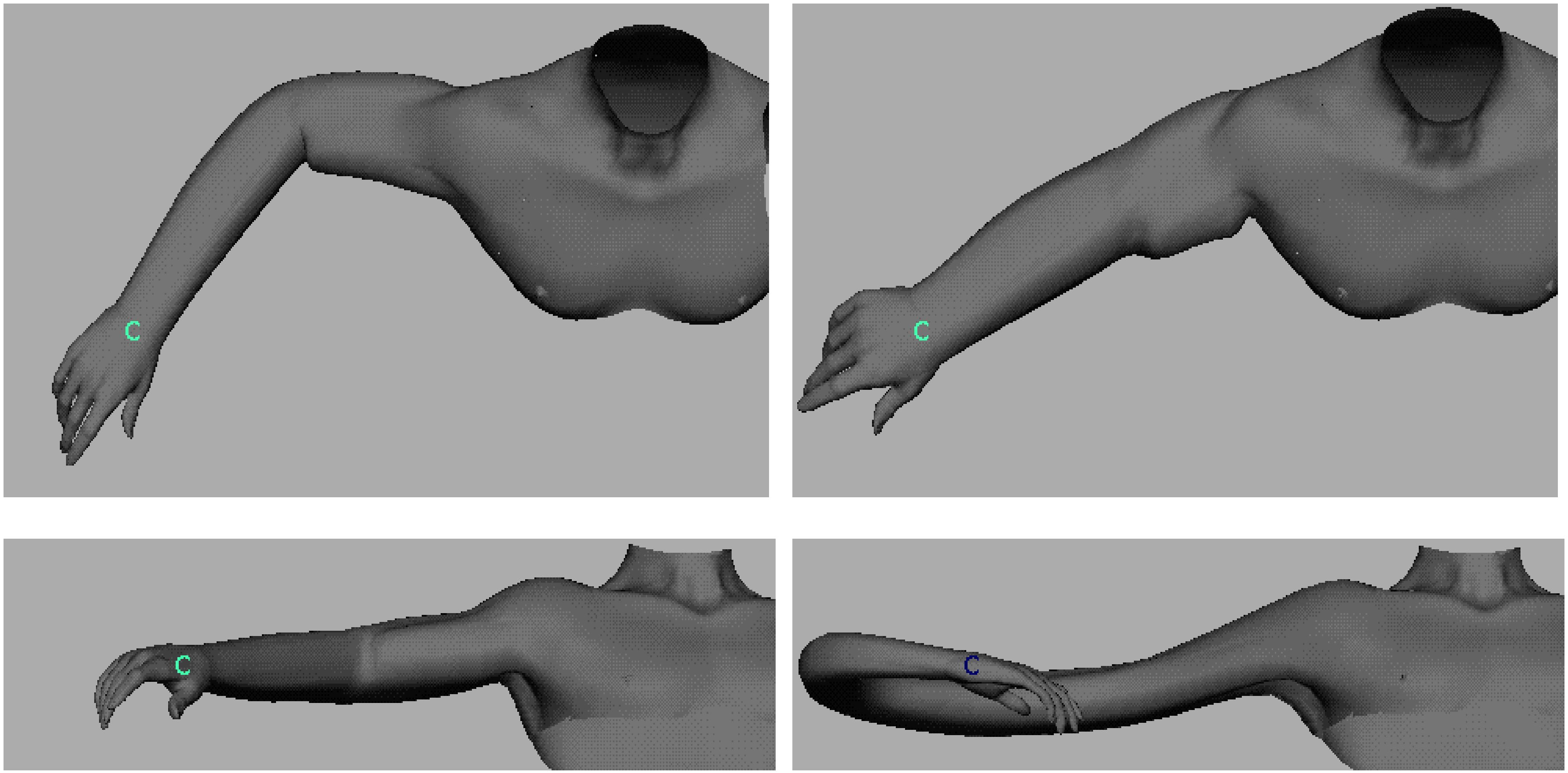


Wrinkle propagation





Kinematics for flexible skeletons



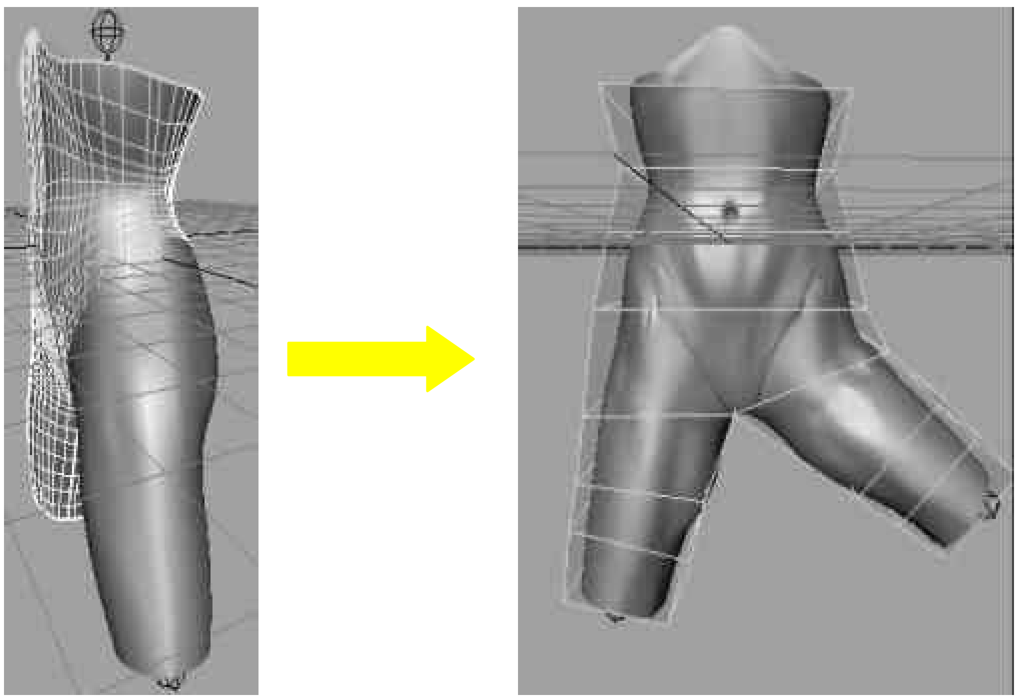
No flexibility **Highly flexible**

Automated Character Skinning

- Create wrap meshes from skin geometry.
- Stitch to a single skin mesh.
- Bind wrap mesh to the skeleton
- Bind skin to wrap mesh.
- Customize : Add resolution, skeletal control to the wrap mesh. Edit the mapping from skin to control elements of the wrap mesh.

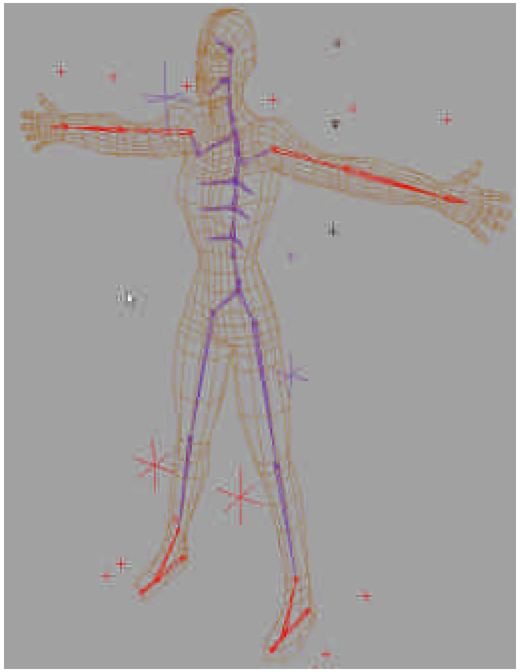
Automated Character Skinning

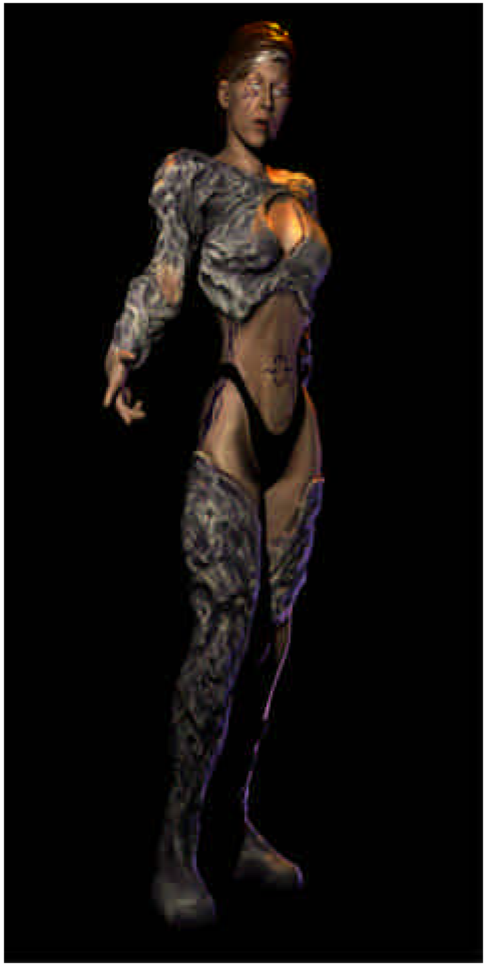
- Create wrap meshes from skin geometry.
- Stitch to a single skin mesh.



Automated Character Skinning

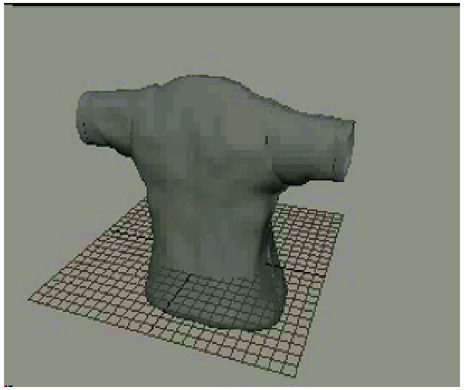
- Bind wrap mesh to the skeleton
- Bind skin to wrap mesh.

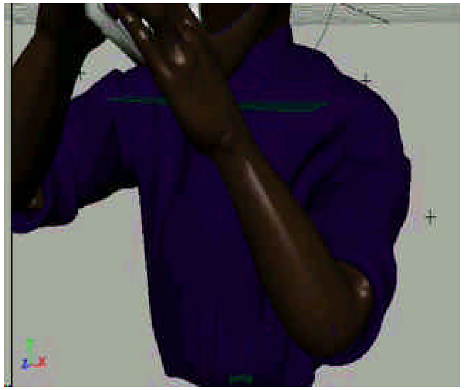




Automated Character Skinning

- Examples





Summary

- An effective deformation technique employing space curves and implicit functions.
- An efficient and controlled approach to the aggregation of the results of multiple deformations...
- An implicit function primitive defined by space curves.
- Applications that illustrate the power and utility of the described techniques.

Acknowledgements

Alias Inc.
Eugene Fiume
Alexis Angelidis