Space Deformation

Alexis Angelidis

Graphics & Vision Lab - Otago University

2004

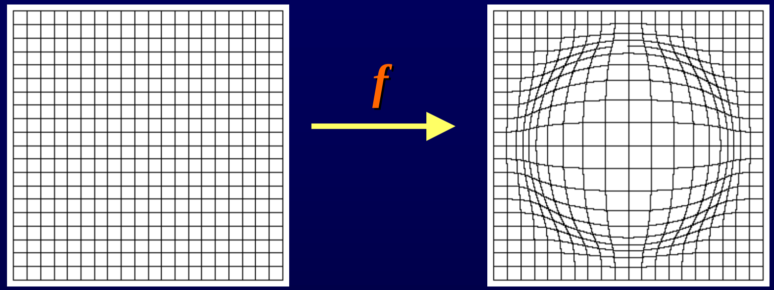
Inspired by the fantastic work of many people

Introduction

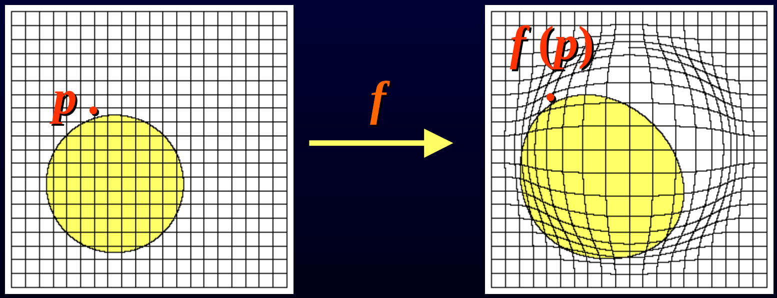
- Earliest reference in Graphics: 1984 [A. Barr'84]
- Space deformation is also called free-form modeling or warping
- Space deformation is a class of techniques
- Some are popular for modeling and animation:
  Maya® & 3D Studio®

What is a space deformation?

- A <u>mapping</u> from IR^n to IR^n
  2D example:



- How is it <u>usefull</u>?
  - Modelling/animation:
    from IR^3 to IR^3
    object made of points
  - Morphing:
    from IR^2 to IR^2



Willow (1988), From Morf to Morphing, Fox

- How do people <u>define</u> *f* ?

The most simple space deformations (1/3)

- **Affine transformations:**
  - **Scale, Translation, Rotation …**
- **Scale**

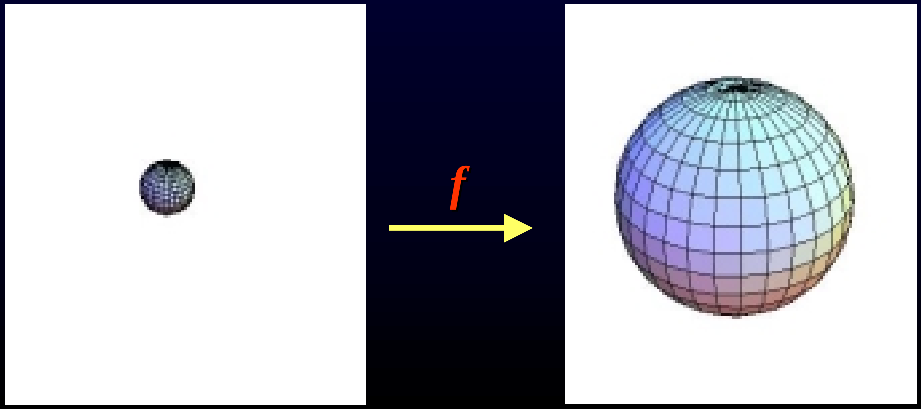
Short :

$$f(p) = sp$$

Matrix Form :

$$\begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} sp_x \\ sp_y \\ sp_z \\ 1 \end{pmatrix}$$

f **applies to all the point in** $\mathbb{R}^3$



The most simple space deformations (2/3)

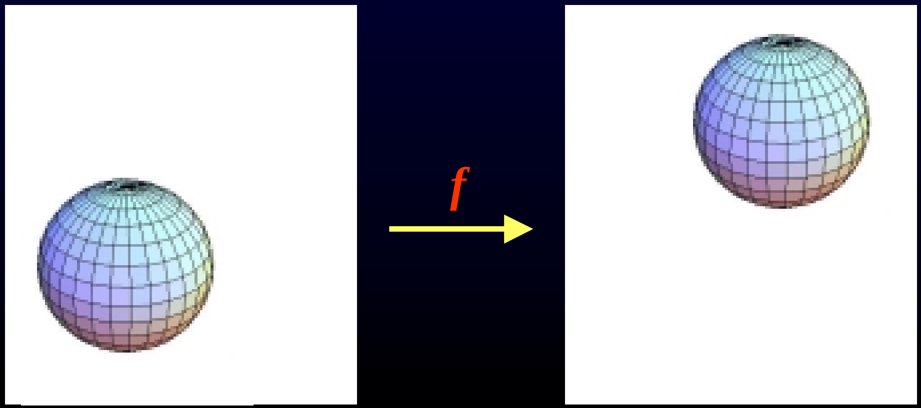
- **Translation**

Short :

$$f(p) = p + \vec{t}$$

Matrix Form :

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{pmatrix}$$



The most simple space deformations (3/3)

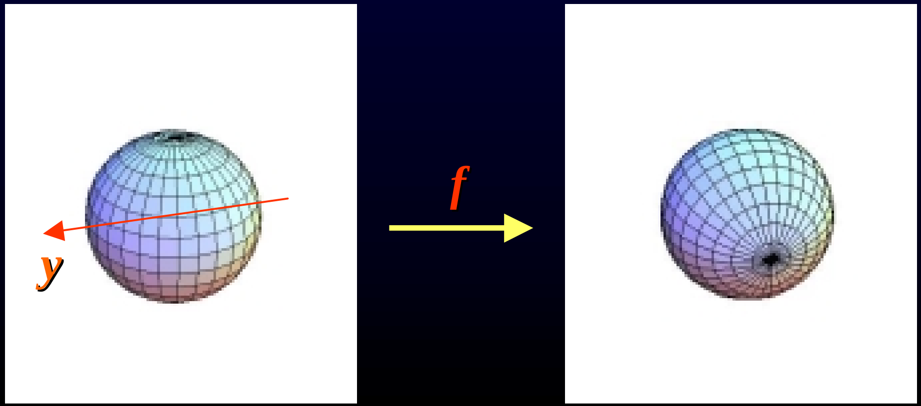
- **Rotation**

Short :

$$
\begin{aligned}
f(p) &= (1-\cos\theta)(p\vec{y})\vec{y} \\
&+ p\cos\theta \\
&+ p\times\vec{y}\sin\theta
\end{aligned}
$$

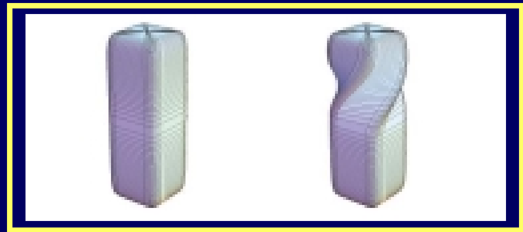
Matrix Form :

$$
\begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta p_x + \sin\theta p_y \\ -\sin\theta p_x + \cos\theta p_y \\ p_z \\ 1 \end{pmatrix}
$$

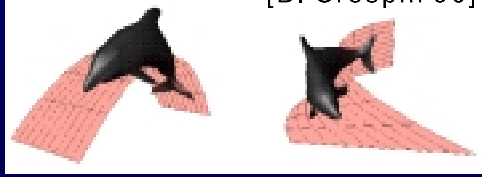


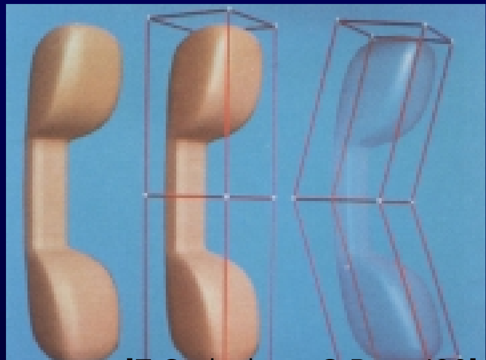
How does it get useful?

- **Matrix = constant, everywhere…**
- **Deforming is all about control**
  - ➡ **Axial Space Deformations**
  - – **Surface Space Deformations**
  - – **Lattice Space Deformations**
  - – **Specialized Space Deformations**

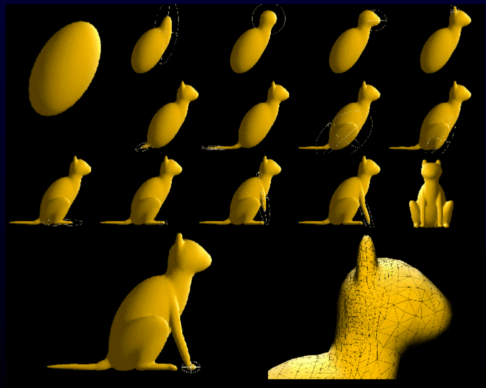


[B. Crespin'96]





[T.Sederberg,S.Parry'86]



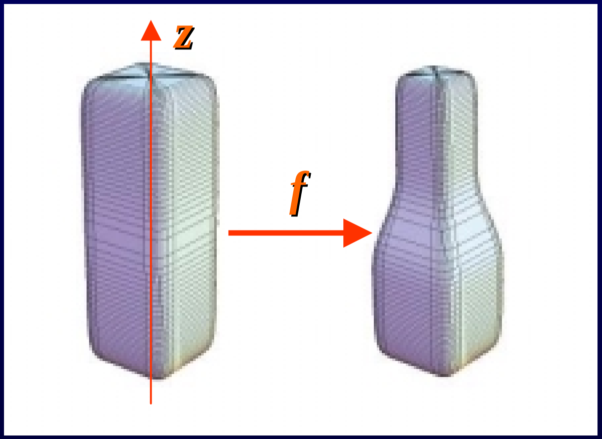
[P.Decaudin'96]

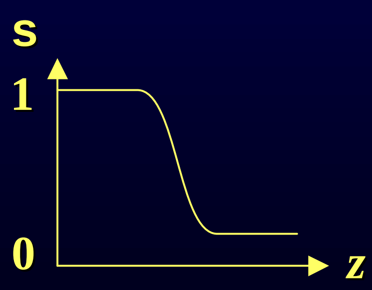
Global Deformation (1/3)

[A. Barr'84]

- **scale → taper**

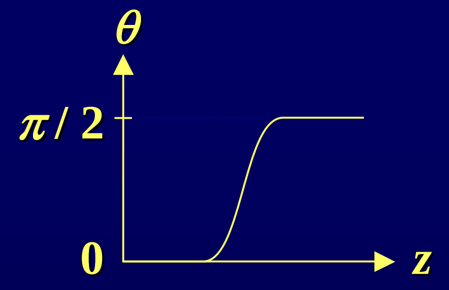
$$\begin{pmatrix} s(z) & 0 & 0 & 0 \\ 0 & s(z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} s(z)p_x \\ s(z)p_y \\ p_z \\ 1 \end{pmatrix}$$





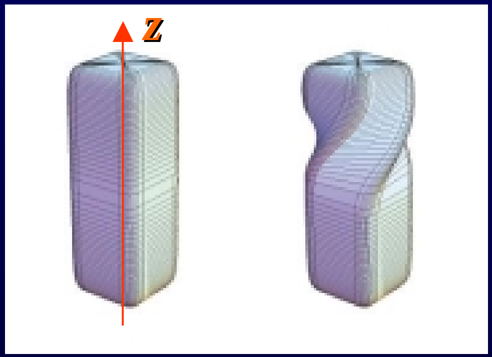
Global Deformation (2/3)

[A. Barr'84]



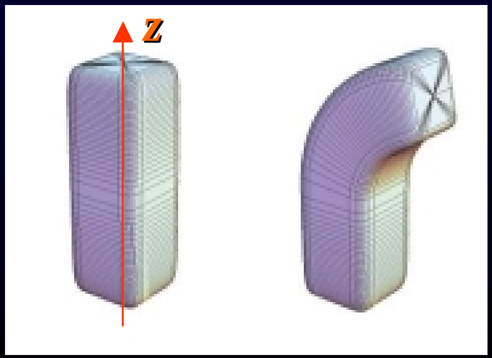
- **rotation in z → twist**

$$\begin{pmatrix} \cos\theta(z) & \sin\theta(z) & 0 & 0 \\ -\sin\theta(z) & \cos\theta(z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$



- **rotation in y → bend**

$$\begin{pmatrix} \cos\theta(z) & 0 & -\sin\theta(z) & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta(z) & 0 & \cos\theta(z) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$



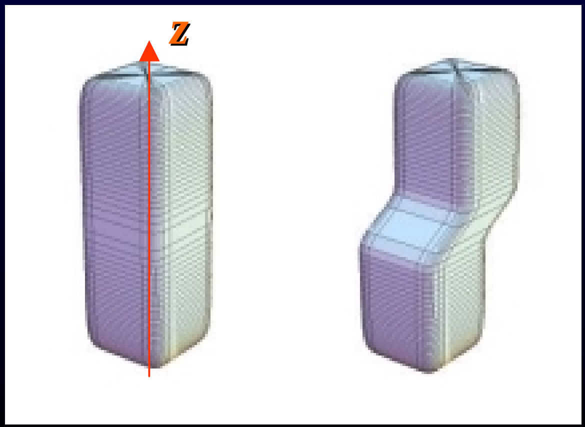
(a bit more complex in [A.Barr'84])

Extended Global Deformation (3/3)

[C.Blanc'95]

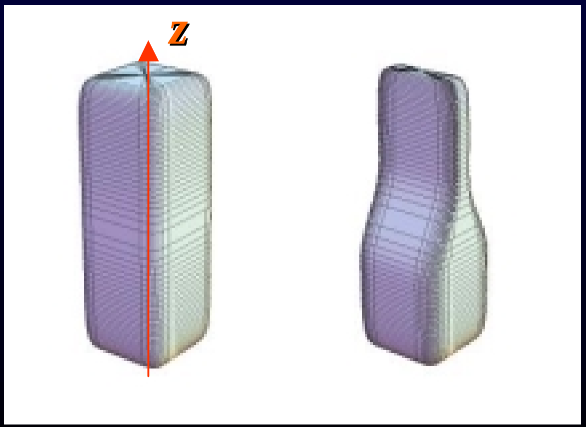
- **translation in $x \rightarrow$ shear**

$$\begin{pmatrix} 1 & 0 & 0 & t(z) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



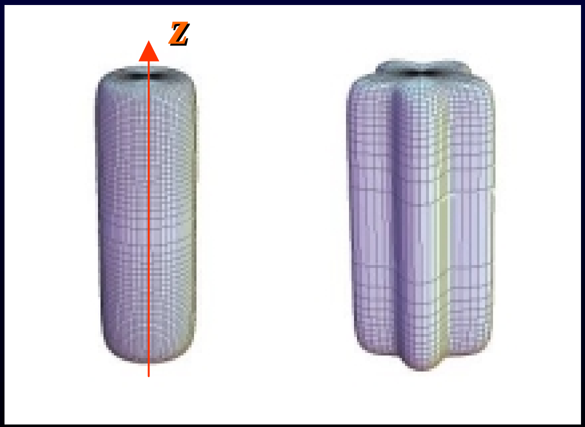
- **scale in $x \rightarrow$ pinch**

$$\begin{pmatrix} s(z) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



- **scale in $\theta \rightarrow$ mould**

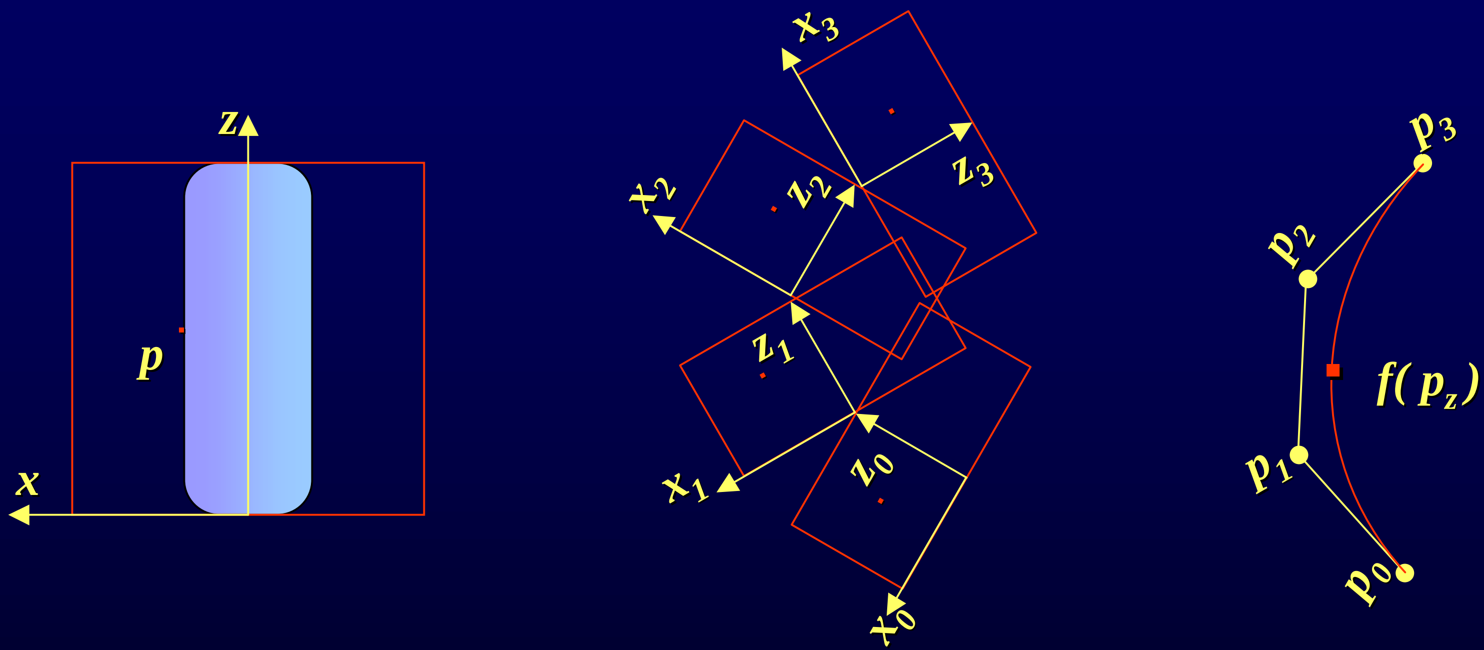
$$\begin{pmatrix} s(\arctan(x,y)) & 0 & 0 & 0 \\ 0 & s(\arctan(x,y)) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

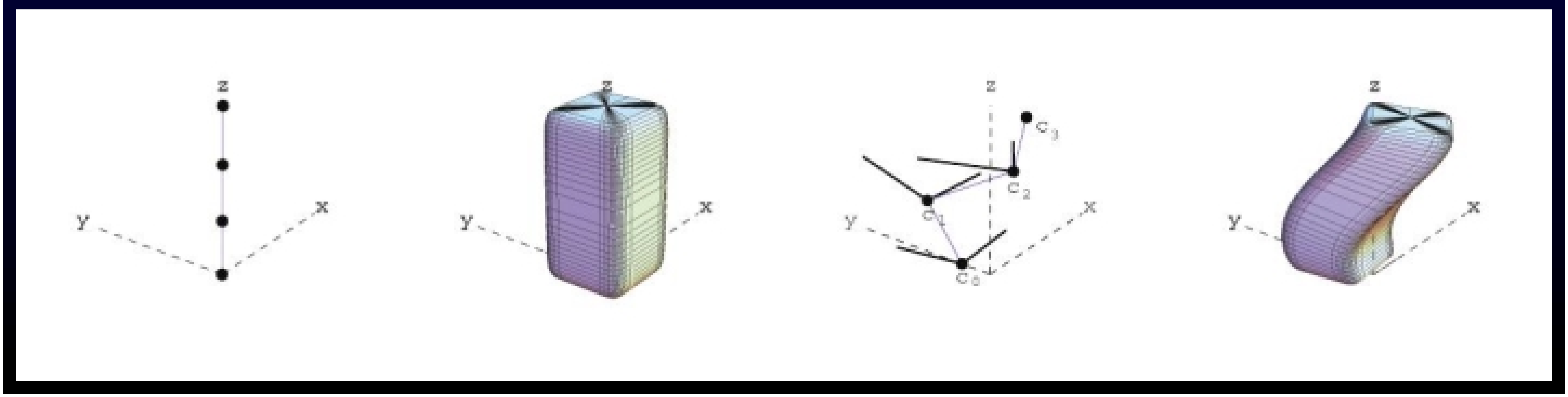


More control (1/2)

[Y.K.Chang and A.P.Rockwood'94]

- **One variable, *z*, is not enough control**





More control (2/2)

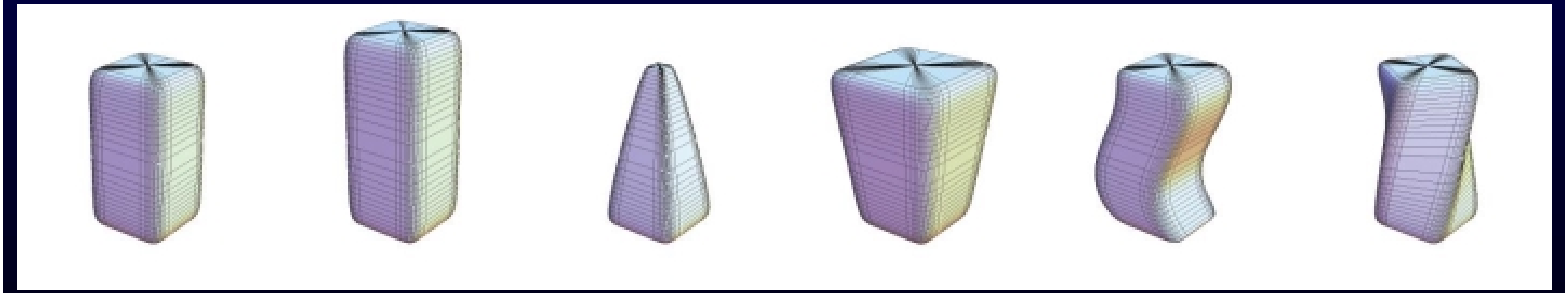
[Y.K.Chang and A.P.Rockwood'94]

- **Bézier curve**

$$f_i^{\,j}(p_z) = (1 - p_z)\, f_i^{\,j-1}(p_z) + p_z\, f_{i+1}^{\,j-1}(p_z)$$

where $f_i^{\,0}(p_z) = p_i$

- **Examples**



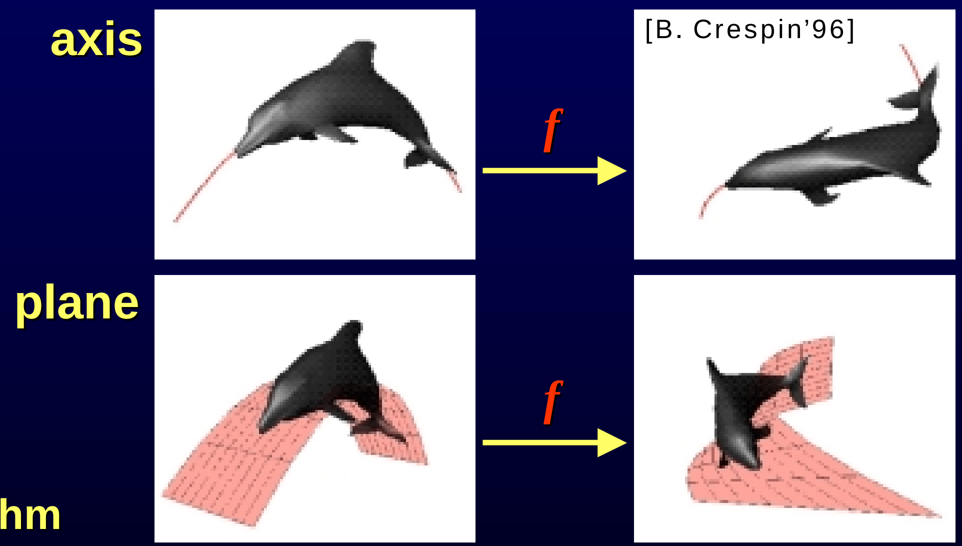
Initial shape **stretch** **taper** **swell** **bend** **twist**

and more…

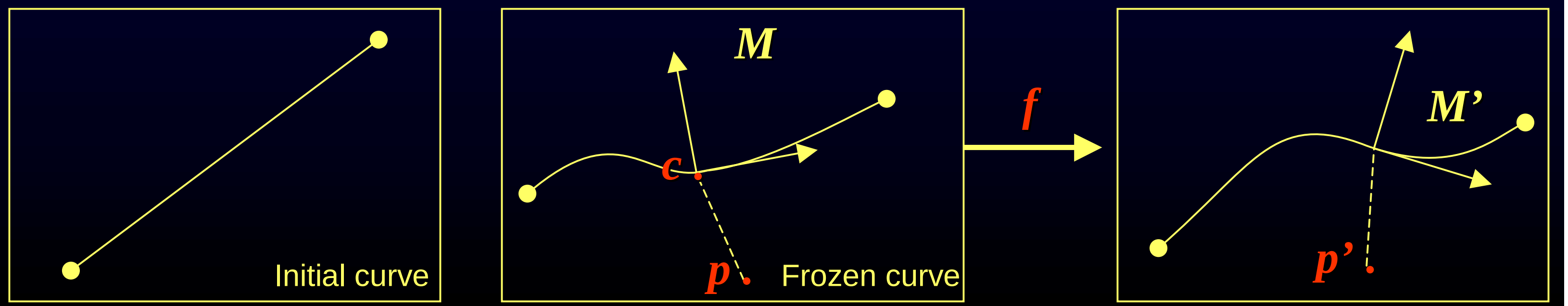
Generalized Axis

[B. Crespin'96]

- **and if initial shape is bent?**
- **Limitation of previous methods: control along straight axis**
- **[B. Crespin'96] proposes initially bent axis (and surface)**
- **3 steps for the artist**
  - **Input initial parameters**
  - **Freeze parameters**
  - **Input deformation parameters**



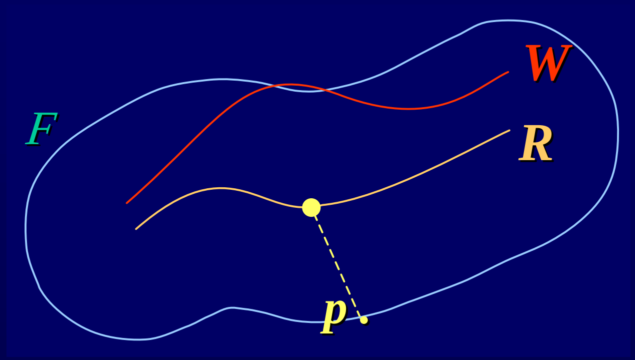
Fast hierarchical closet-point algorithm



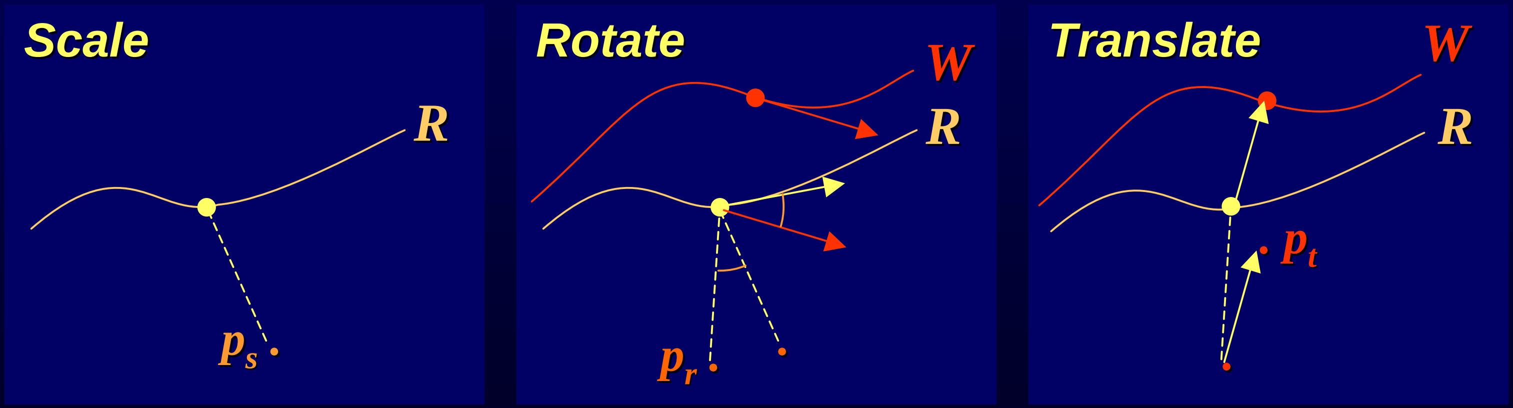
Wires (1/3)

[K.Singh & E.Fiume'98]

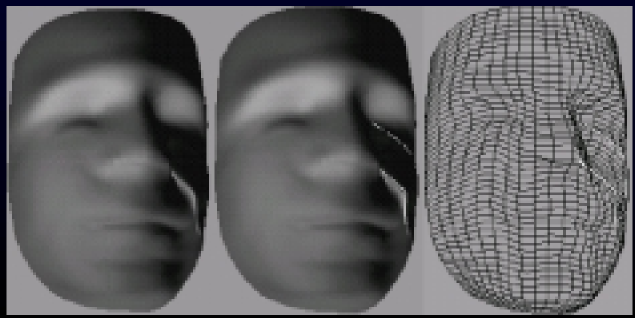
- wire ≡ armature used by sculptors
  - *R* reference curve
  - *W* wire curve
  - *F* density function



- Deformation = three steps



- The density function modulates each step
  - – scale factor *s F(p)*
  - – rotation angle *q F(p)*
  - – translation vector *t F(p)*

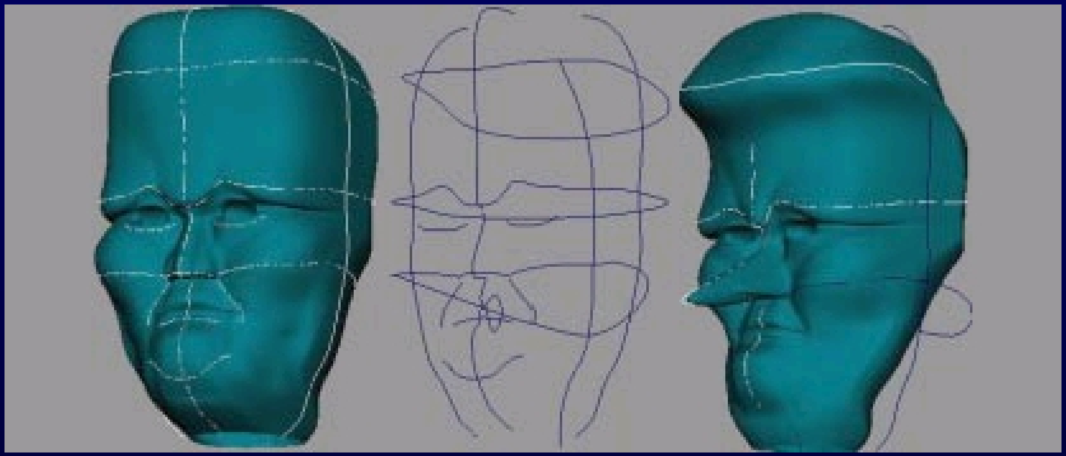


[K.Singh & E.Fiume'98]

Wires (2/3)

[K.Singh & E.Fiume'98]

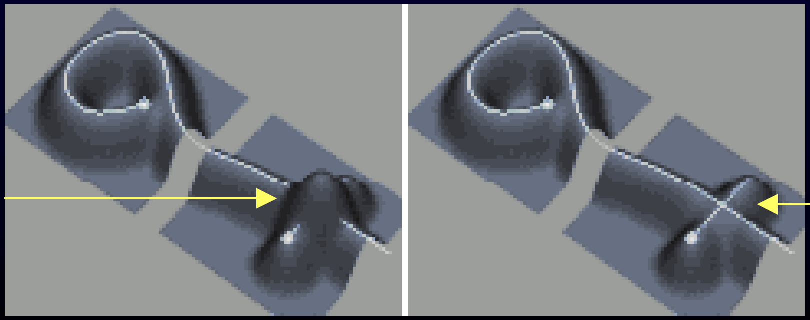
- **Multiple wires**



[K.Singh & E.Fiume'98]

- **How do they blend?**

Double translation



Normalized translation (with neglected artefacts)

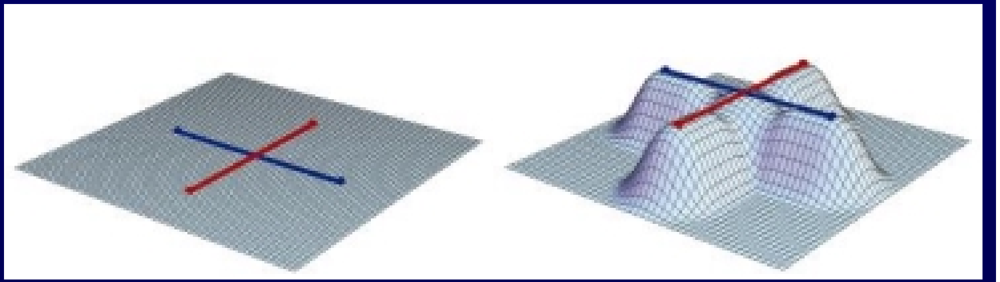
[K.Singh & E.Fiume'98]

Wires (3/3)

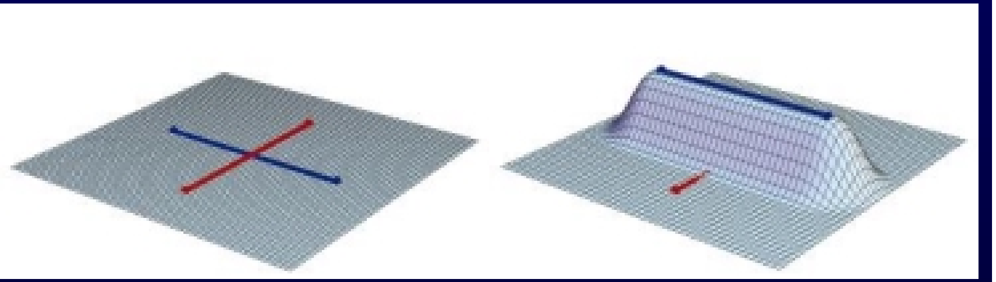
[K.Singh & E.Fiume'98]

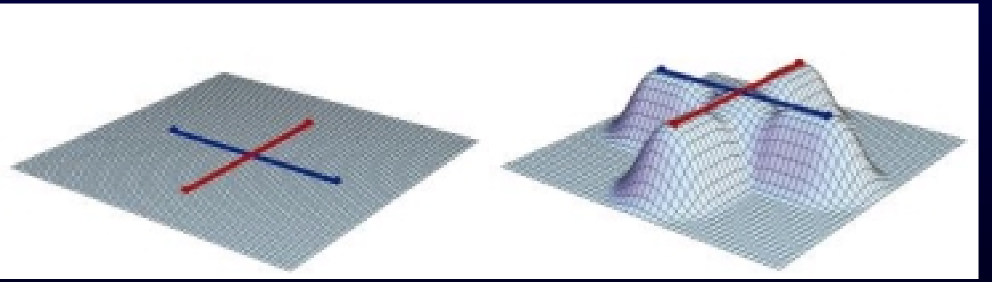
- **Blending multiple wires**

$$\Delta p_i = f_i(p) - p$$

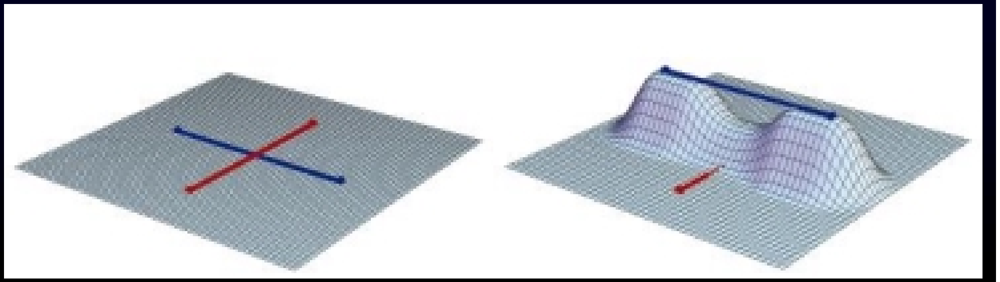


$$f(p) = p + \frac{\sum_i \Delta p_i \|\Delta p_i\|}{\sum_i \|\Delta p_i\|}$$



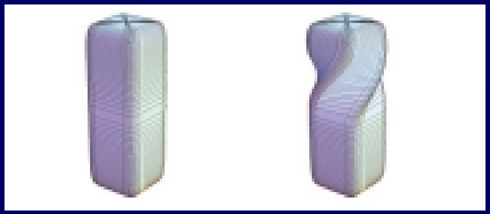


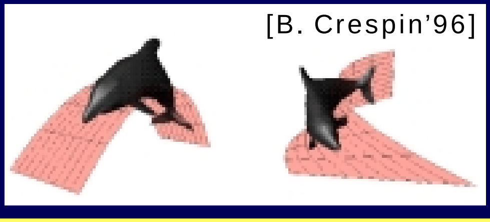
$$f(p) = p + \frac{\sum_i \Delta p_i F_i(p)}{\sum_i F_i(p)}$$



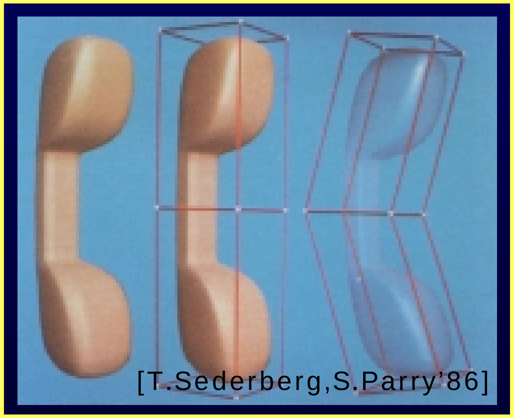
Outline

- **More control…**
  - Axial Space Deformations
  - Surface Space Deformations
  - ➡ **Lattice Space Deformations**
  - Specialized Space Deformations

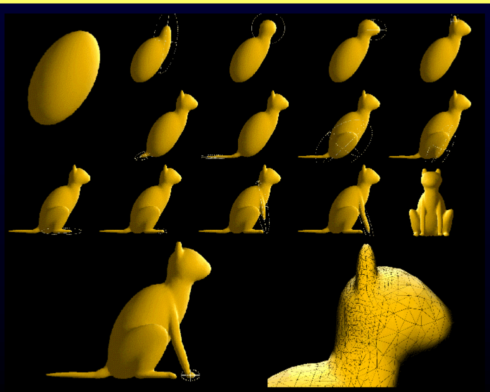




[B. Crespin'96]



[T.Sederberg,S.Parry'86]



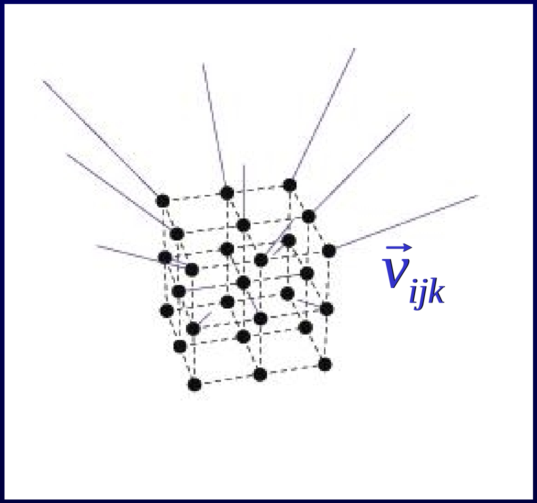
[P.Decaudin'96]

Lattice Space Deformation

- **2 interpretations**
  - **A blured grid of displacement vectors**

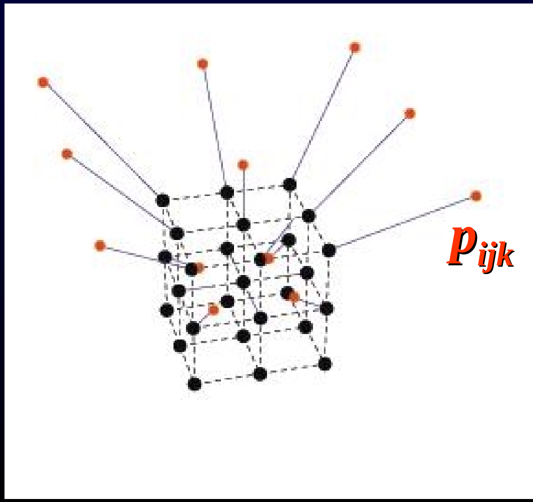
$$p_{def} = p + f_{\vec{v}}(\vec{v}_{ijk}, p)$$

    - **physical analogy:** **flux of a fluid** / **electromagnetic field**
    - **more insight:** **straight path from source to destination**



  - **A blurred grid of new positions**

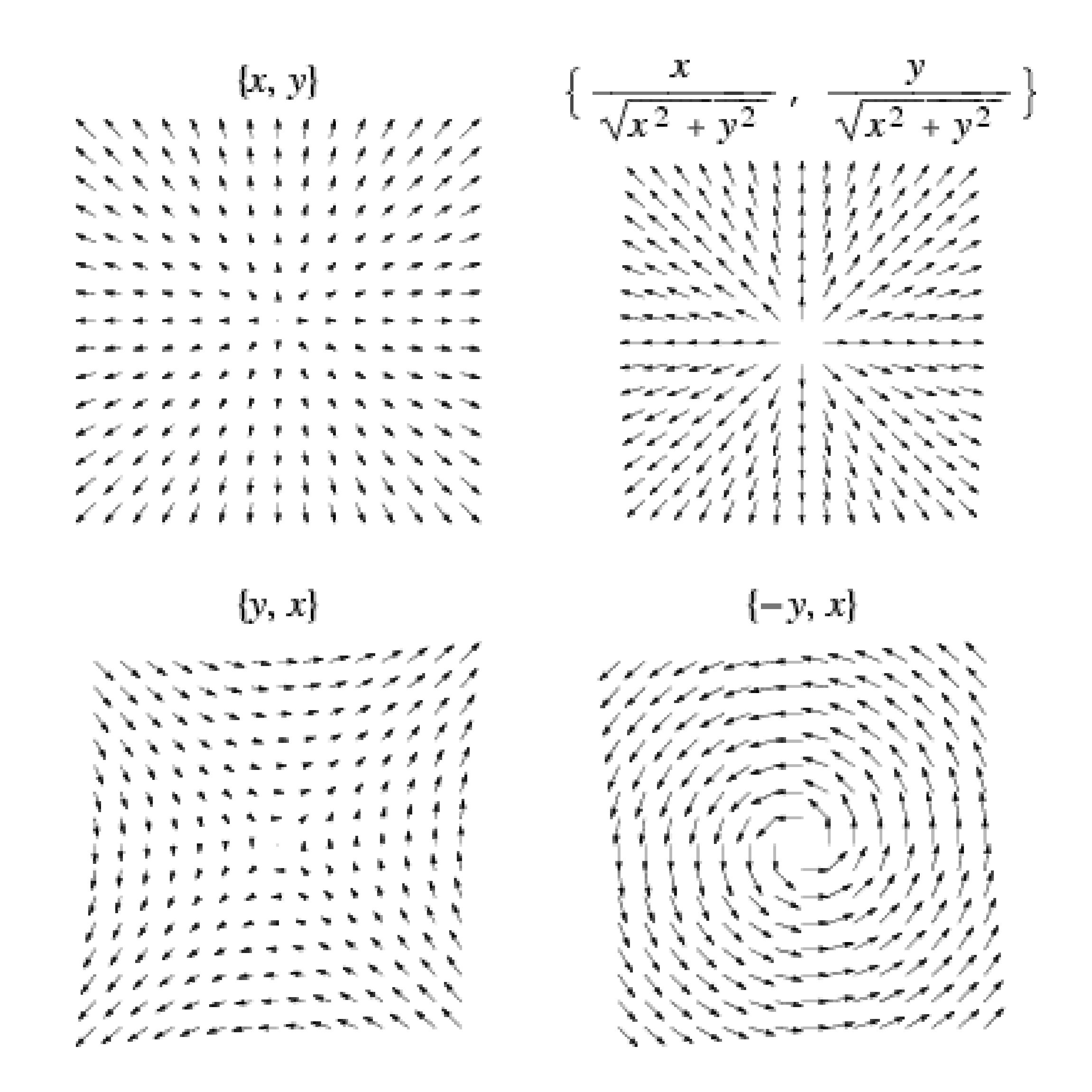
$$p_{def} = f_p(p_{ijk}, p)$$



$$\vec{v}(p) = \sum_k \left(B_k^{\,n}(z) \sum_j \left(B_j^{\,m}(y) \sum_i B_i^{\,l}(x) \vec{v}_{ijk} \right) \right)$$

Vector fields

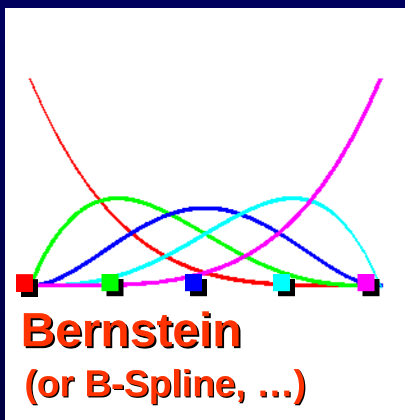
- **Examples in 2D**

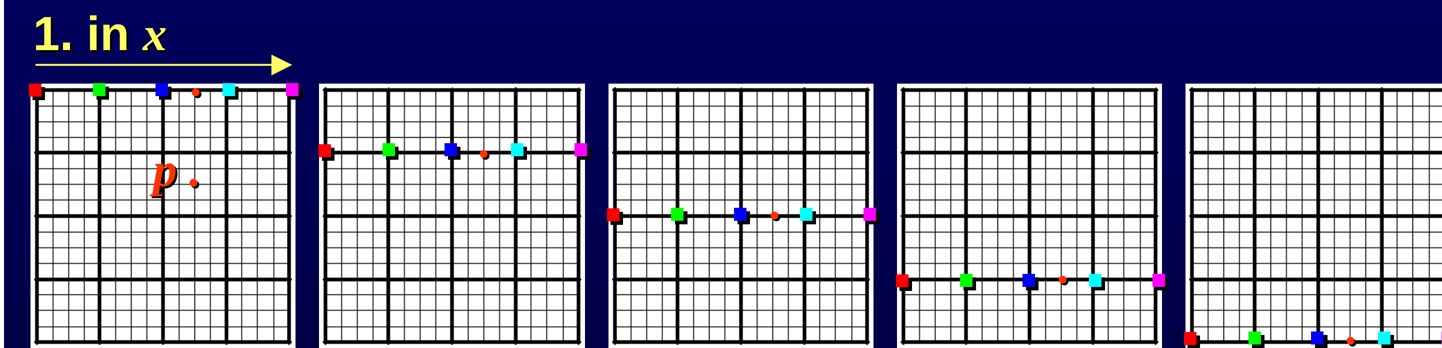


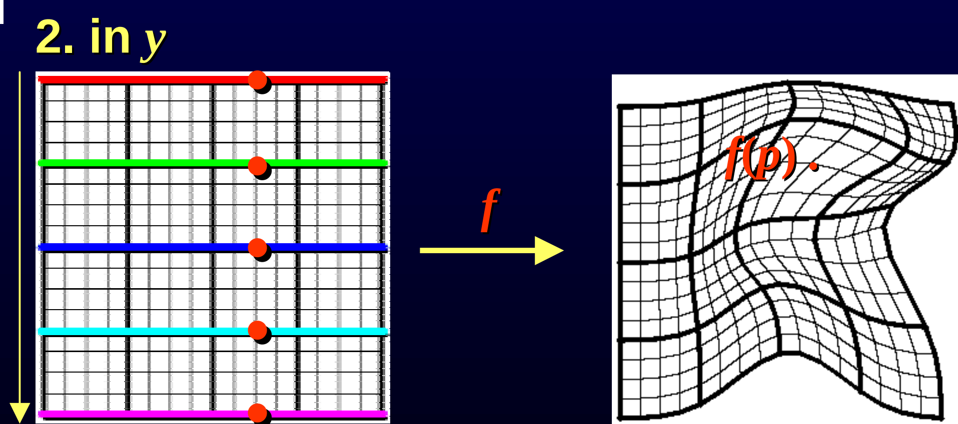
Blurring a vector field

[T.Sederberg & S.Parry'86]

- **Influence curves, in each dimension. 5x5 lattice in 2D :**







$$\sum_j \left(B_j^m(j,y) \sum_i B_i^l(x) \vec{v}_{ij} \right)$$

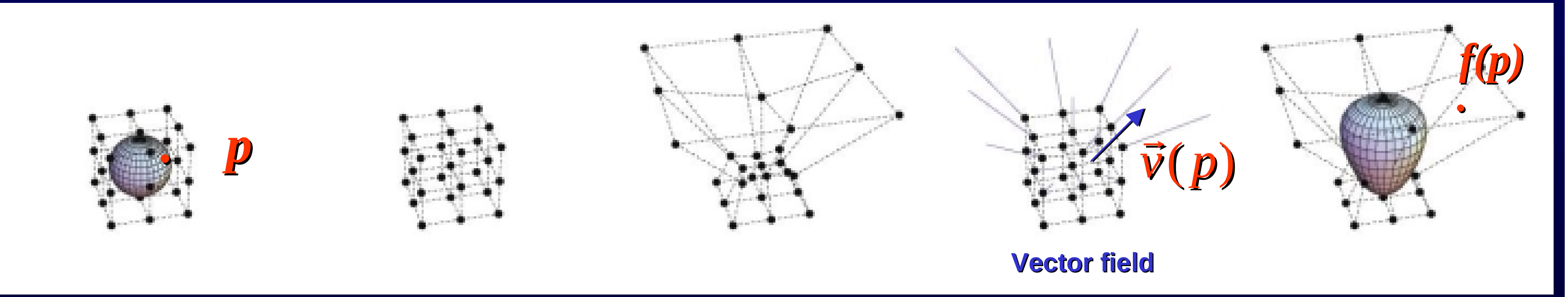
- **Easy to find *p*'s coordinates in the lattice**
- **Limitation: the lattice is regular**

Basic FFD (1/2)

[T.Sederberg & S.Parry'86]

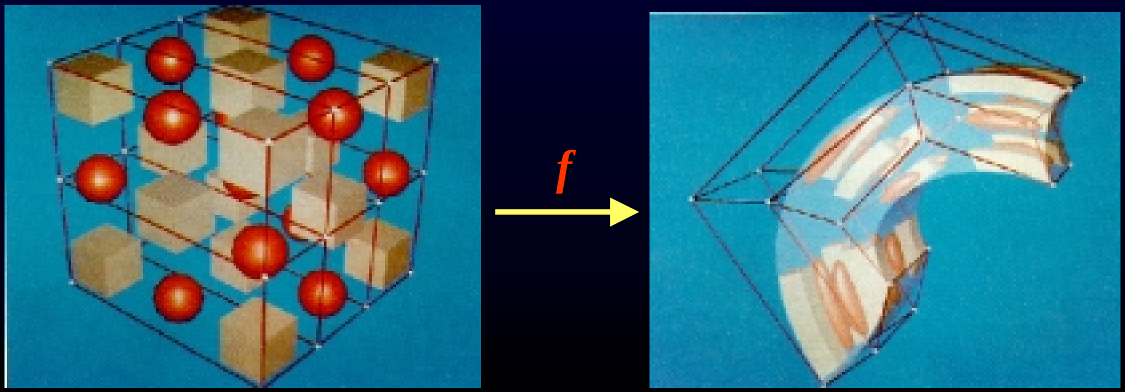
- [A. Barr'84] [B. Crespin'96] deformation constant along a straight axis/plane or a bent axis/plane

- More control: vectors specified at points, and smooth them out



Steps

- Place lattice
- Move control points
- Apply the field to the points

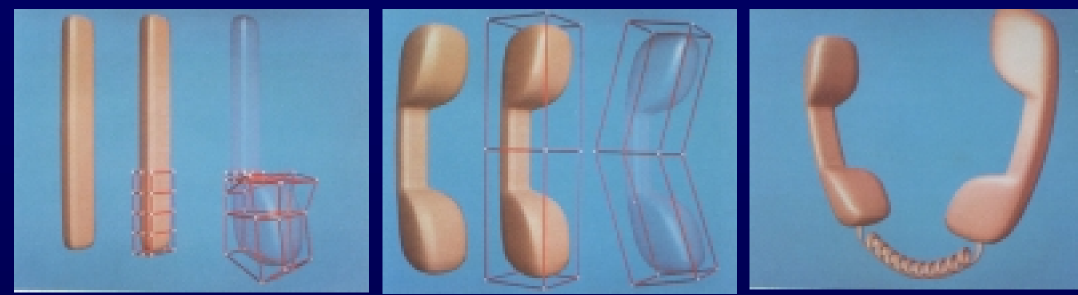


[T.Sederberg & S.Parry'86]

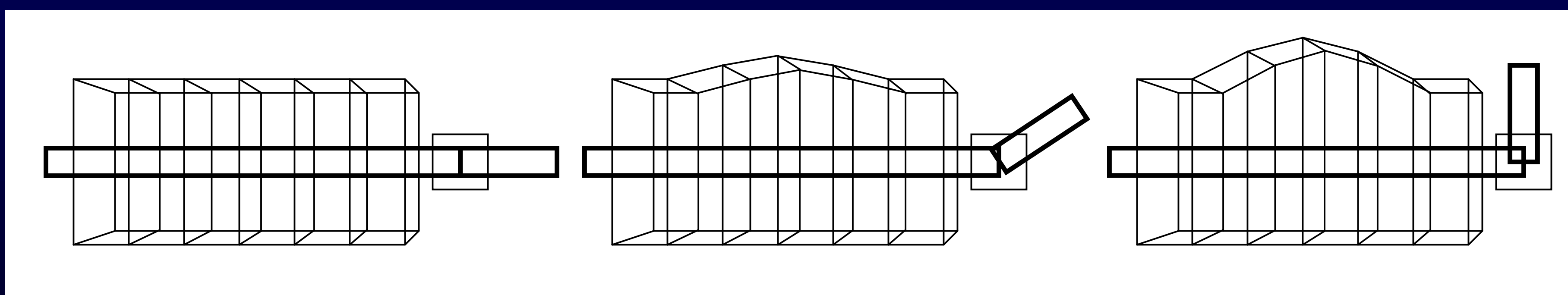
Basic FFD (2/2)

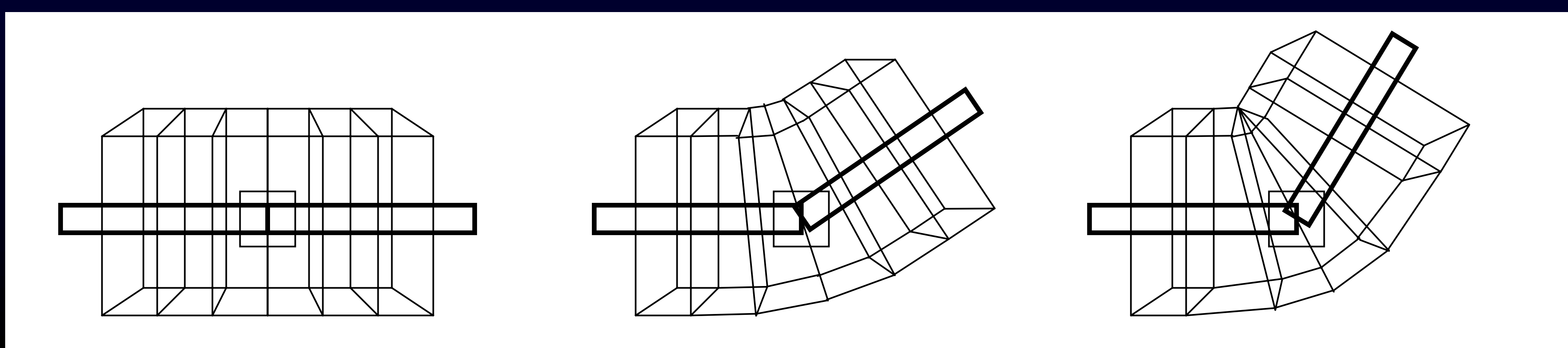
[T.Sederberg & S.Parry'86]

- **Examples of application**
  - **Modeling**
  - **Animation**



[T.Sederberg & S.Parry'86]



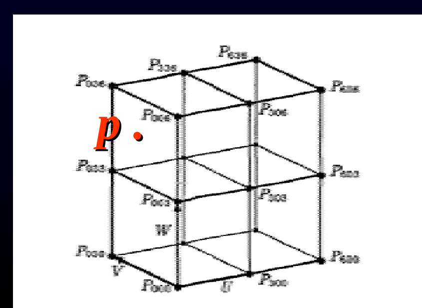


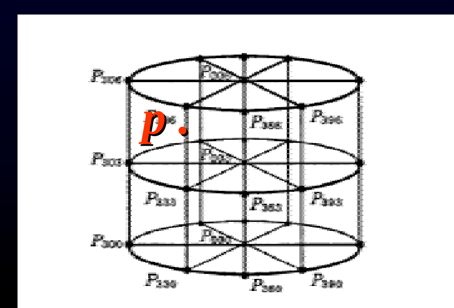
Extended FFD - Structured Lattice

[S.Coquillart'90]

- **Initially deformed lattice**
  - **cells are not cubes. They are small 4x4x4 FFD**

- **Drawbacks:**
  numerical computation of local coordinates in a deformed lattice
  tedious connection of cells

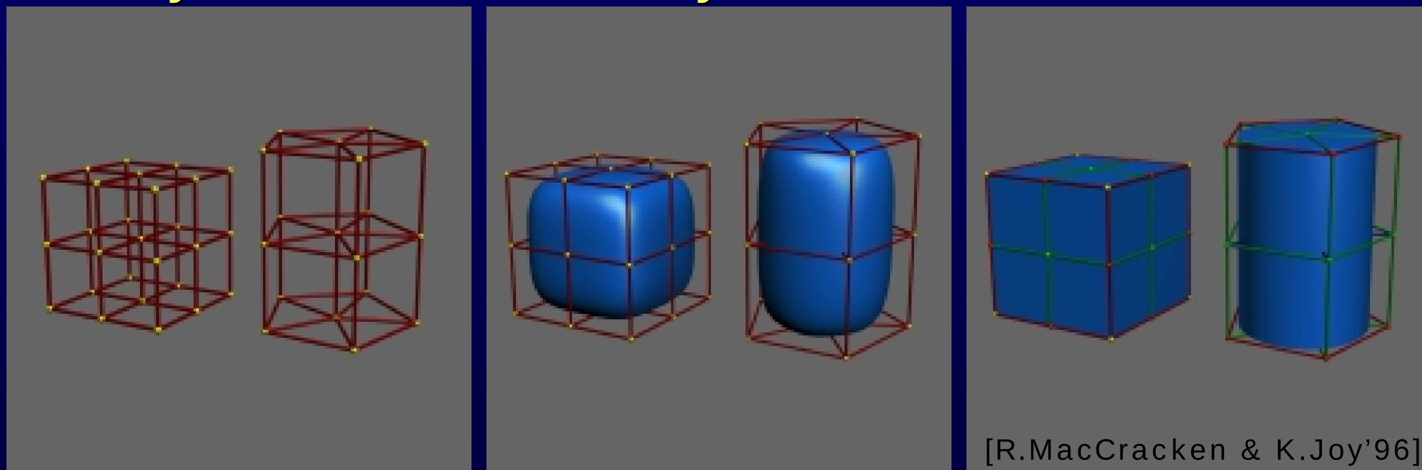




SFFD (1/2)

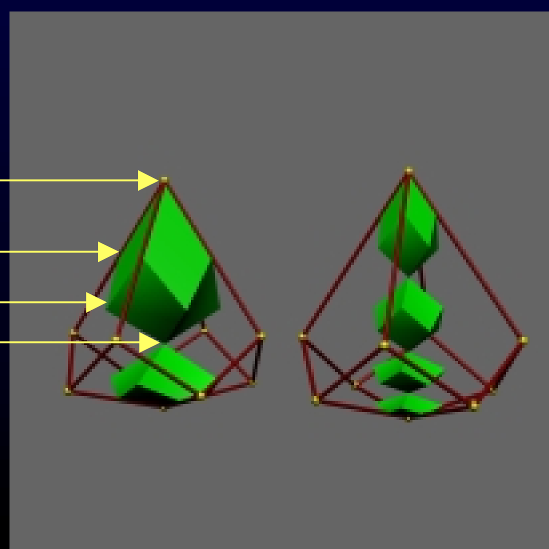
[R.MacCracken & K.Joy'96]

- **Lattice ≡ subdivision volume**
  rich variety of lattices & continuity across faces



- **Subdivision rules for**

vertex point
edge point
face point
cell point

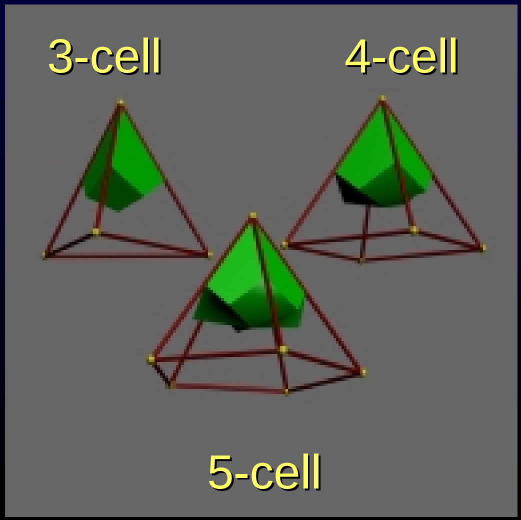


There is no parameterization for subdivision surfaces/lattices

How is a point assigned lattice coordinates?

SFFD (1/2)

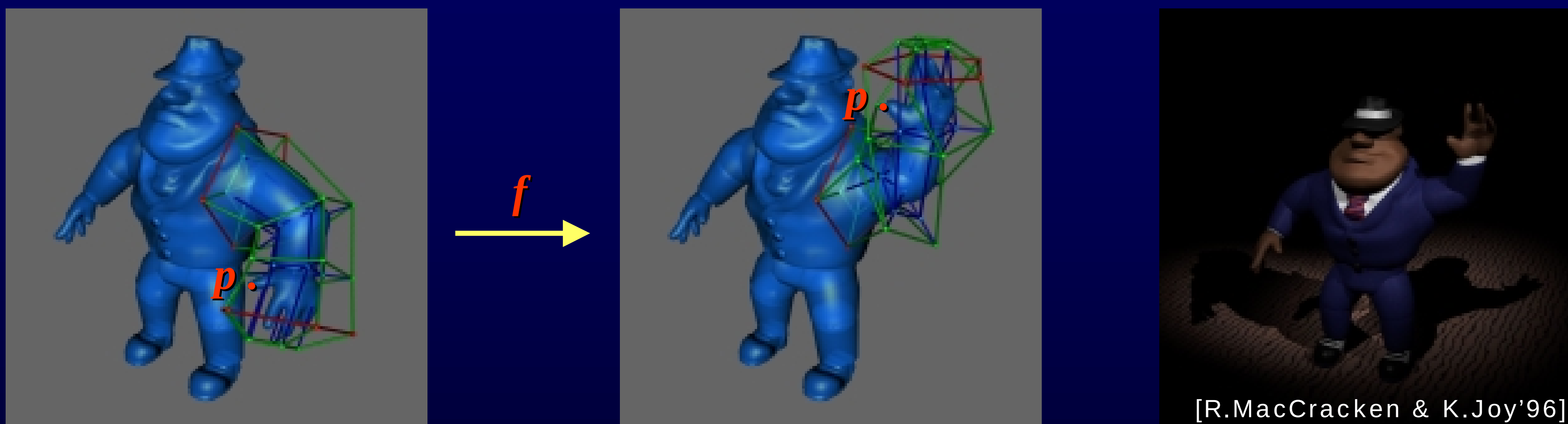
[R.MacCracken & K.Joy'96]



- **rich variety of lattice shapes and topology**
  - **after 1st subdivision: all cells are n-cells**

SFFD (2/2)

[R.MacCracken & K.Joy'96]

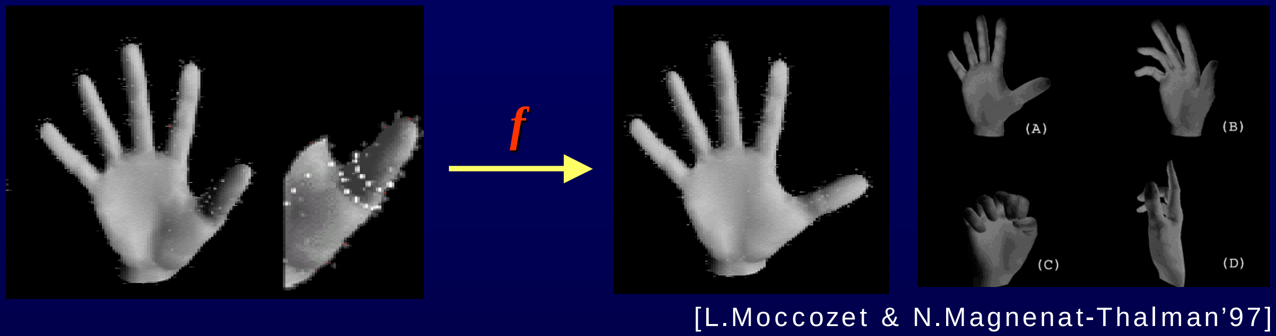


- **find position of p in lattice and hold local coordinates**
- **move lattice**
- **Trace new position of p in new lattice using local coordinates**

FFD - Avoiding Lattice (3/3)

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

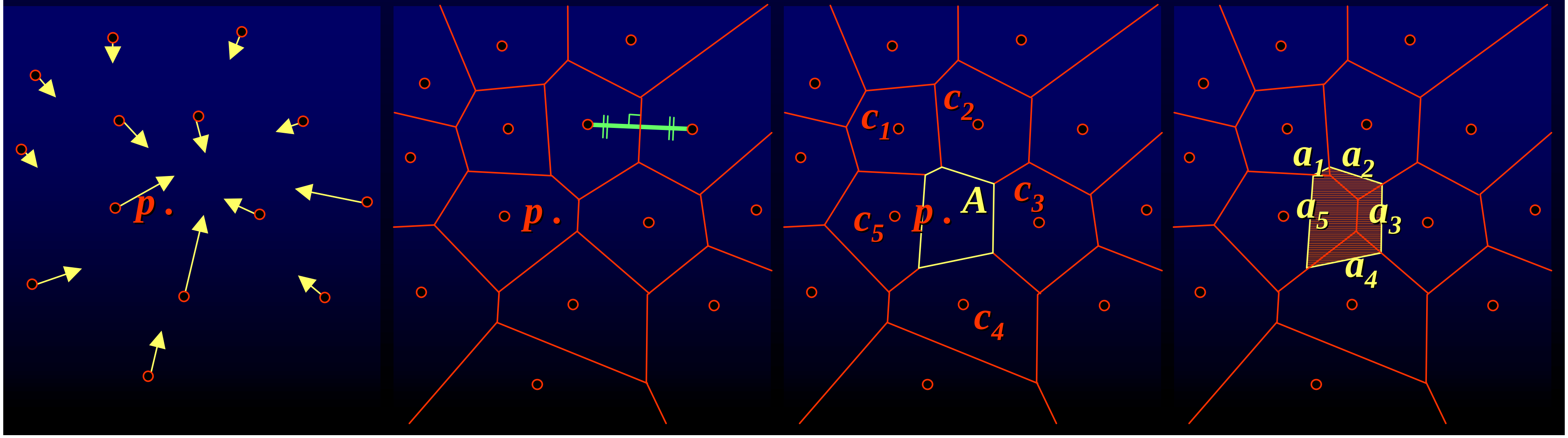
- **Lattice defined over arbitrary points:**



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

- **Voronoï cells**
- **Sibson coordinates ≡ linear interpolant,**
  (smoothed out with multivariate Bernstein polynomials)

$$\vec{v}(p) = \frac{\sum_i a_i \vec{v}_i}{A}$$



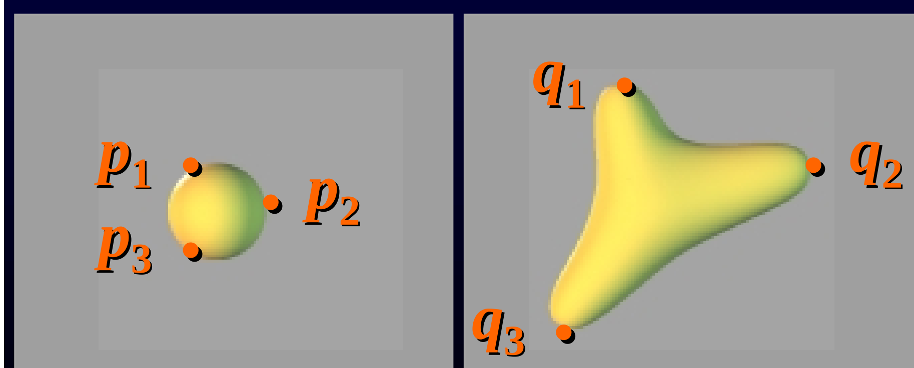
About explicit lattices

- In FFD, EFFD, SFFD… : too many control points ⇒ self-occlusion
- Control-points = preset handles
- A Single Control point cannot “grab” the surface. The surface slips.

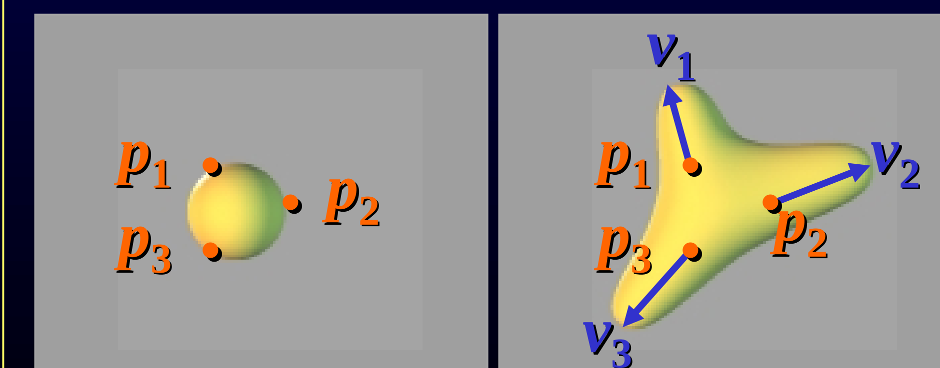
← no control points

- Instead of a lattice, pairs of…

point and image s (p_i, q_i)



point and displacement points (p_i, v_i)

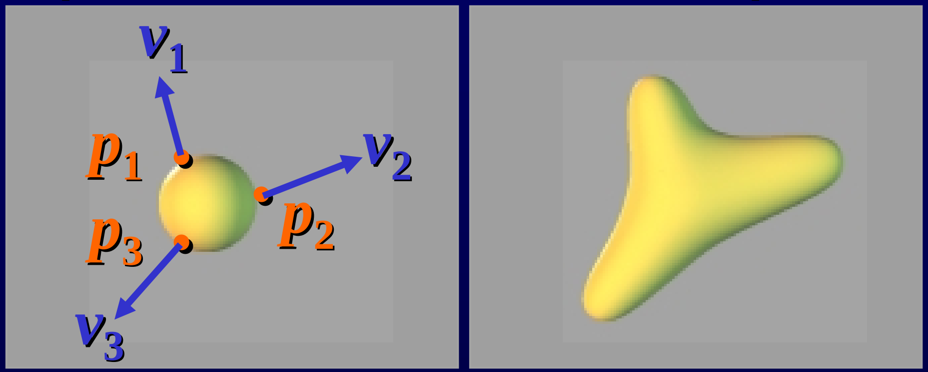


3 techniques with no explicit lattice definition…

Circumventing the Lattice with FFD (1/2)

[W.Hsu J.Hughes & H.Kaufman'92]

- **Deduce control points from constraints on points**



$$\vec{v}(p) = \sum_k \left(B_k^{\,n}(z) \sum_j \left(B_j^{\,m}(y) \sum_i B_i^{\,l}(x) \vec{v}_{ijk} \right) \right)$$

- **how can we find the v_{ijk} that satisfy the pairs given by the artist (v_i, p_i) ?**

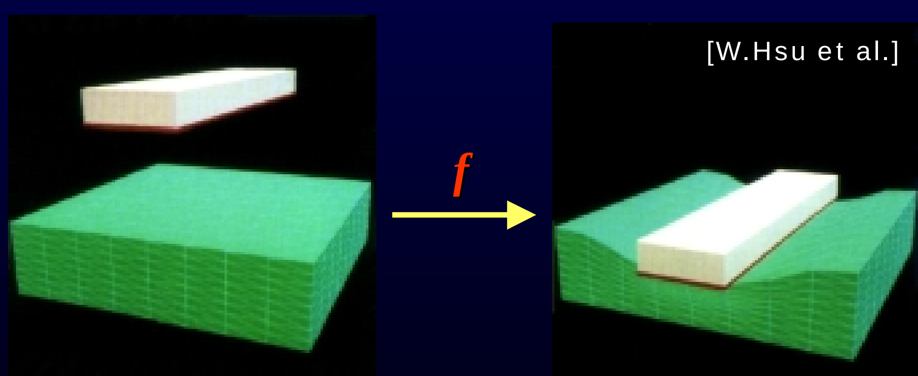
$$\vec{v}(p) = B(p) \begin{pmatrix} \vec{v}_{000} \\ \vdots \\ \vec{v}_{lmn} \end{pmatrix} \qquad \vec{v}_i = B(p_i) \begin{pmatrix} \vec{v}_{000} \\ \vdots \\ \vec{v}_{lmn} \end{pmatrix} \qquad \begin{pmatrix} \vec{v}_1 \\ \vdots \\ \vec{v}_n \end{pmatrix} = \begin{pmatrix} B(p_1) \\ \vdots \\ B(p_n) \end{pmatrix} \begin{pmatrix} \vec{v}_{000} \\ \vdots \\ \vec{v}_{lmn} \end{pmatrix}$$

Circumventing the Lattice with FFD (2/2)

[W.Hsu J.Hughes & H.Kaufman'92]

- **Find a matrix's pseudo-inverse**

$$\begin{pmatrix} \vec{v}_1 \\ \vdots \\ \vec{v}_n \end{pmatrix} = \begin{pmatrix} B(p_1) \\ \vdots \\ B(p_n) \end{pmatrix} \begin{pmatrix} \vec{v}_{000} \\ \vdots \\ \vec{v}_{lmn} \end{pmatrix} \longrightarrow \begin{pmatrix} B(p_1) \\ \vdots \\ B(p_n) \end{pmatrix}^{+} \begin{pmatrix} \vec{v}_1 \\ \vdots \\ \vec{v}_n \end{pmatrix} = \begin{pmatrix} \vec{v}_{000} \\ \vdots \\ \vec{v}_{lmn} \end{pmatrix}$$

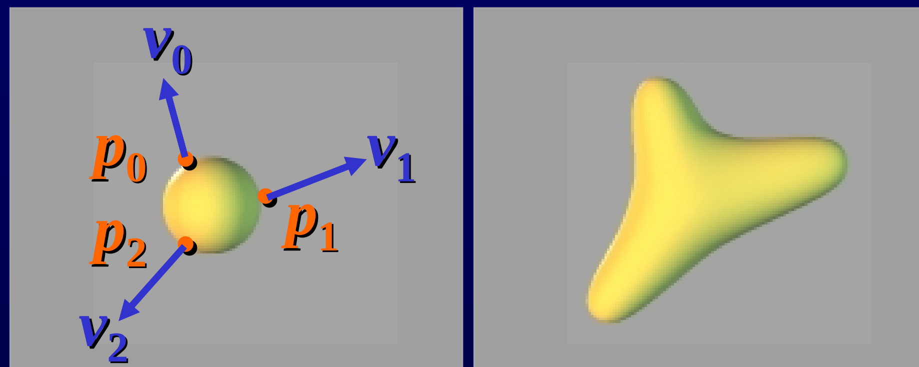


[W.Hsu et al.]

- **Too many constraints $\Rightarrow$ system over-determined**

Circumventing the Lattice with polynomial basis functions

[P.Borrel & D.Bechmann'91]



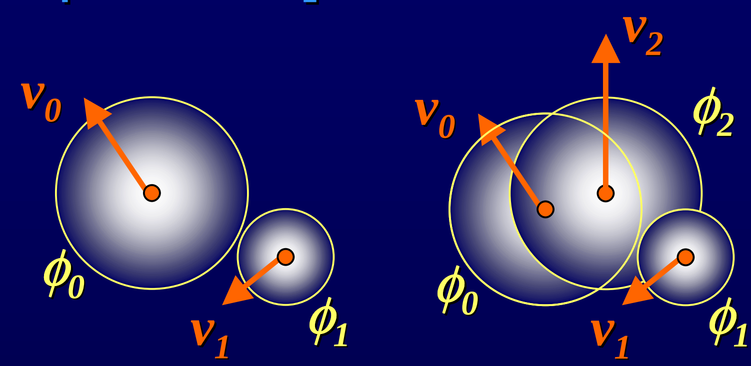
$$\begin{pmatrix} \vec{v}_1 \\ \vdots \\ \vec{v}_n \end{pmatrix} = \begin{pmatrix} B(p_1) \\ \vdots \\ B(p_n) \end{pmatrix} \begin{pmatrix} \vec{v}_{000} \\ \vdots \\ \vec{v}_{lmn} \end{pmatrix}$$

- ***B*** **does not have to be a filter. It could be anything**
  - **Polynomials**
  - **Piecewise Polynomials (B-Spline)**
- **Enables to generalize FFD to IR^n**
- **Hard to predict behaviour**

Circumventing the Lattice with RBF

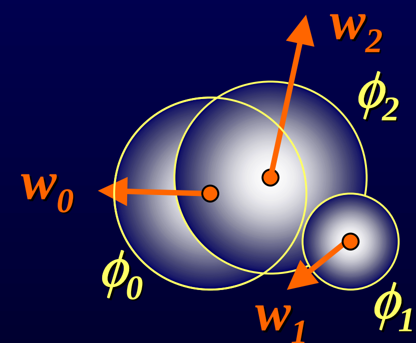
[P.Borrel & A.Rappoport'94]

- **SCODEF: triplets** $(p_i, \vec{v}_i, r_i)$ $\quad \vec{v}(p_i) = \vec{v}_i$
- ***B* is a radial basis function**
- **naïve deformation :** $\vec{v}(p) = \sum_i \left(\vec{v}_i \varphi_i(p)\right)$



- **exact control : replace v_i with some w_i**

$$\vec{v}(p) = \sum_i \left(\vec{v}_i \varphi_i(p)\right)$$



- **All in a matrix:**

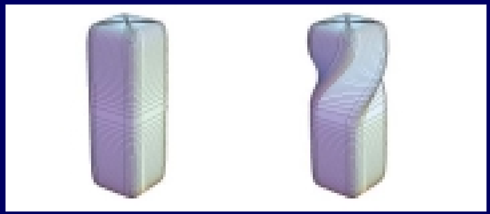
$$\begin{pmatrix} \vec{v}_1^T \\ \vdots \\ \vec{v}_n^T \end{pmatrix} = \begin{pmatrix} \varphi_1(p_1) & \cdots & \varphi_n(p_1) \\ \vdots & & \\ \varphi_1(p_n) & \cdots & \varphi_n(p_n) \end{pmatrix} \begin{pmatrix} w_1^T \\ \vdots \\ w_n^T \end{pmatrix}$$

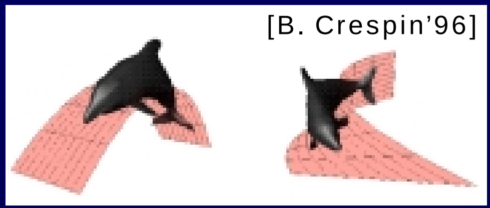
[P.Borrel & A.Rappoport'94]

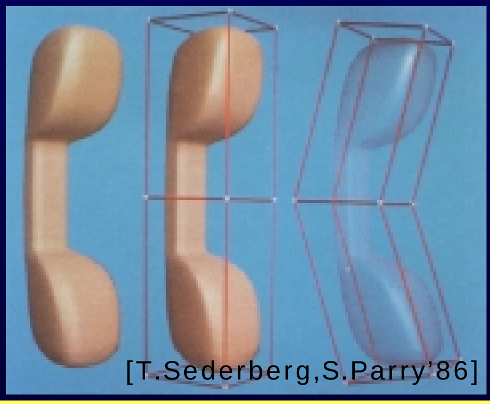
- **Finding the w_i $\Leftrightarrow$ finding the inverse of a square matrix**

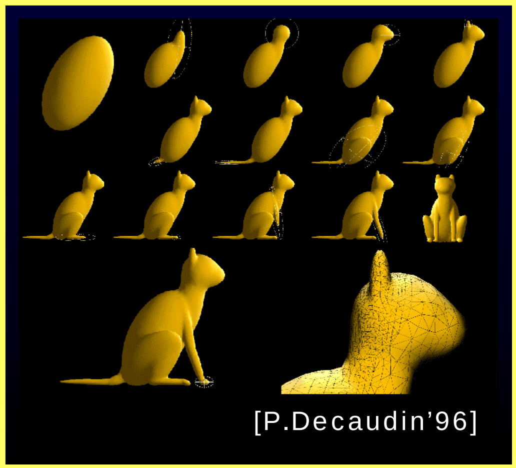
Outline

- **More control**
  - **Axial Space Deformations**
  - **Surface Space Deformations**
  - **Lattice Space Deformations**
  - ➡ **Specialized Space Deformations**





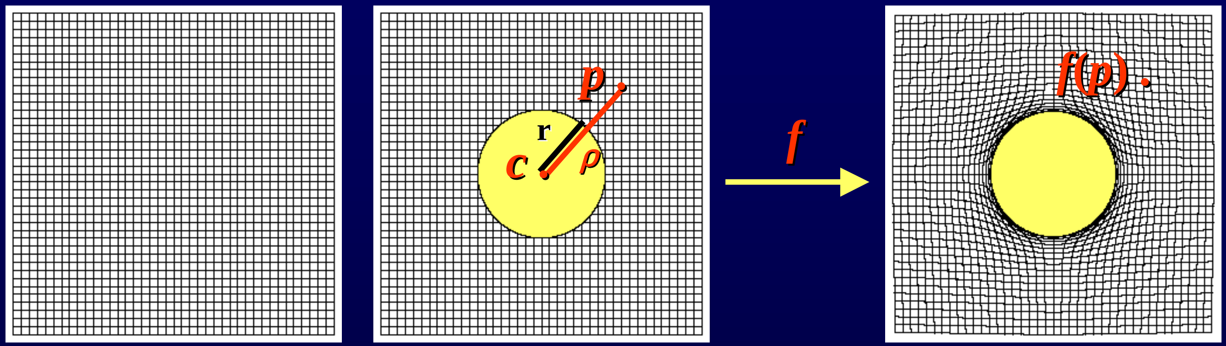




Controlled Volume increase

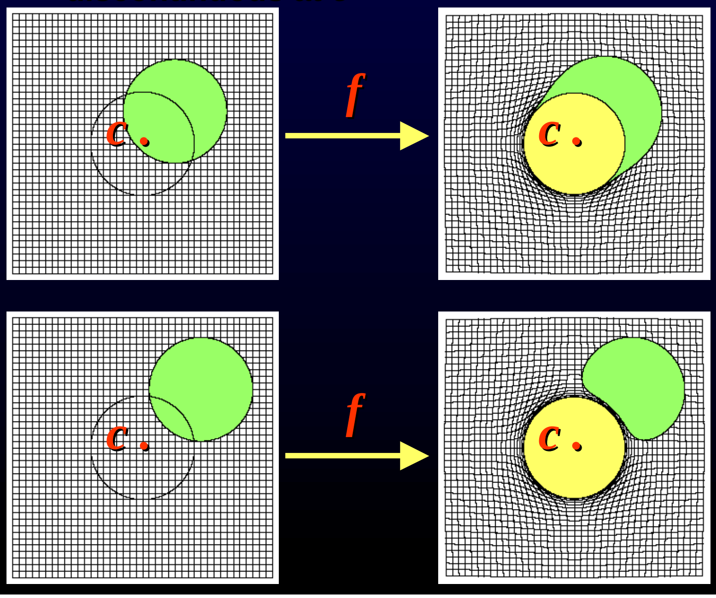
[P.Decaudin'96]

- **Insert an object of volume V in space**



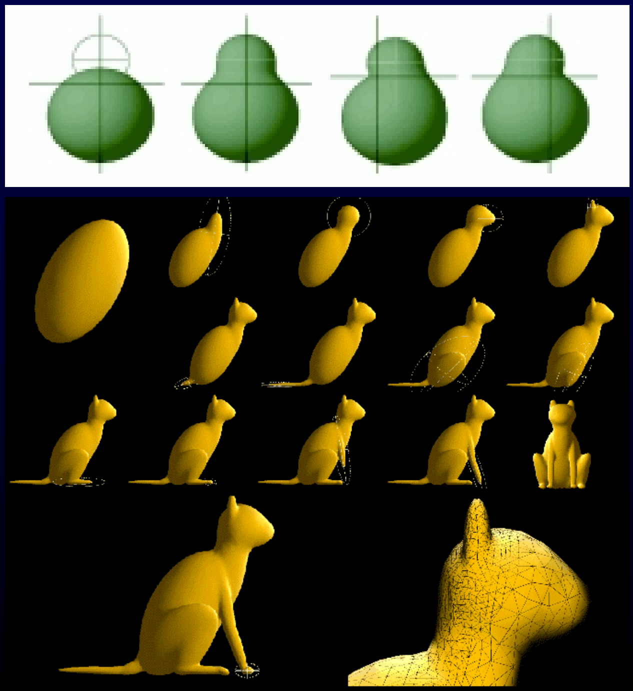
$$f(p) = c + \sqrt[3]{r^3 + \rho^3}$$

- **Restrictions:**
  - c inside tool
  - discontinuous at c



c inside shape:
Volume is increased by V

c outside shape:
volume is maintained



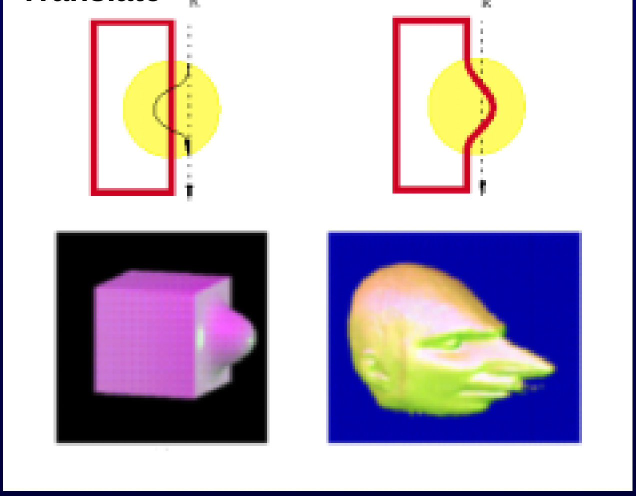
[P. Decaudin'96]

Reversible

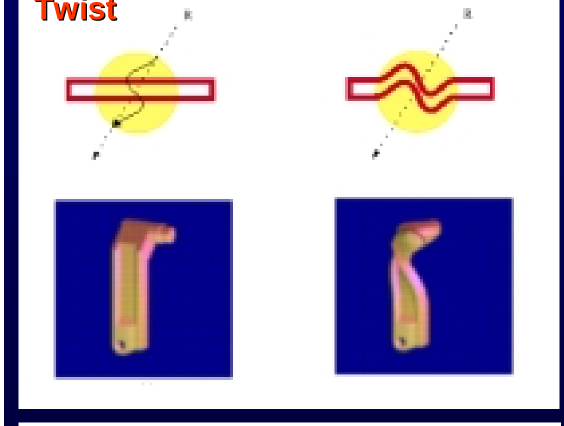
[Y.Kurzion & R.Yagel'97]

- **Deforming a shape ⇔ undeforming rays. 4 local operators:**

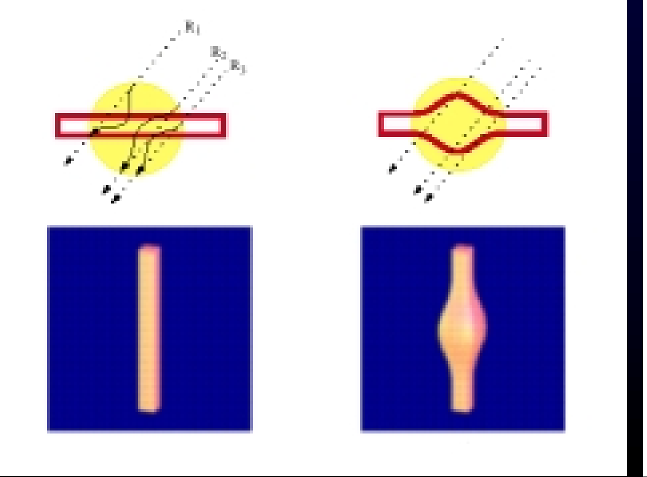
Translate



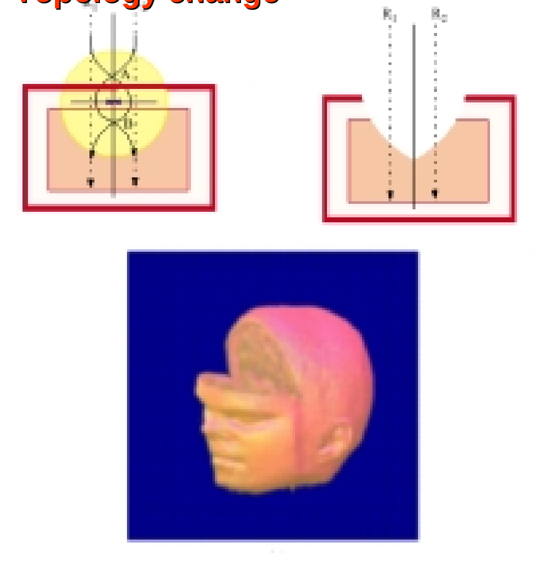
Twist



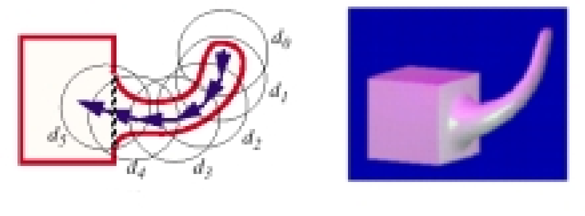
Scale



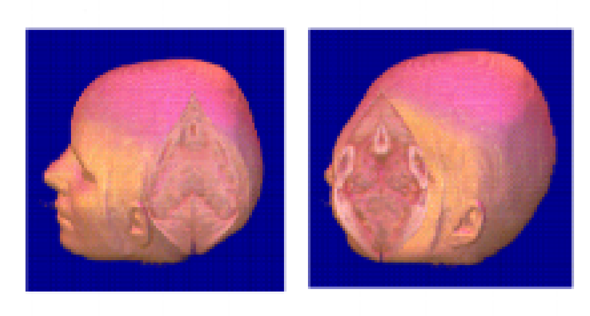
Topology change



Modeling by composition of operators



Application: exploring volume data



“Implicit” FFD

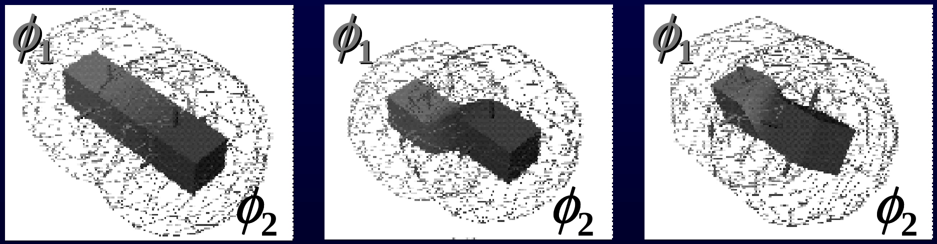
[B. Crespin’96]

- **Single tool**

$$f(p) = p + \varphi(p)(f(p) - p)$$

- **Idea:**
  replace SCODEF’s expensive parameter tuning with cunning blending
  - **partitions of unity blending :**

$$f(p) = p + \frac{\sum_i \varphi_i(p)(f_i(p) - p)}{\sum_i \varphi_i(p)}$$



[B. Crespin’96]

  - **Combination functions :**

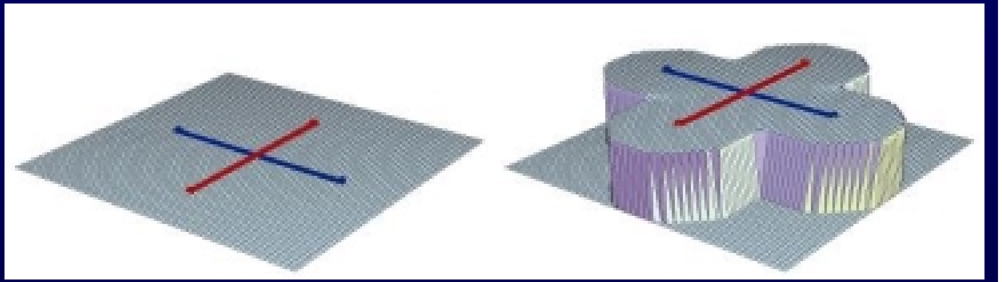
$$f(p) = p + \frac{\sum_i \Gamma_i(p)\varphi_i(p)\Delta p_i}{\sum_i \varphi_i(p)}$$

“Implicit” FFD

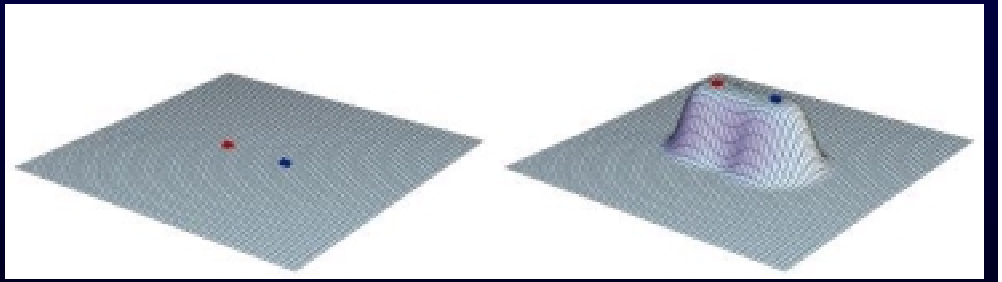
[B. Crespin’96]

- **Blending**

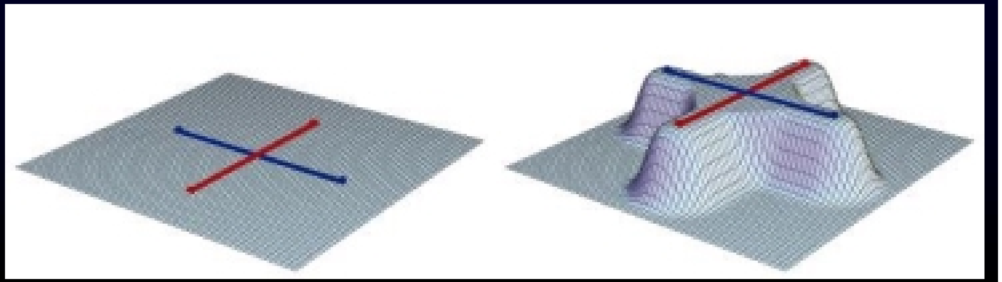
$$f(p) = p + \frac{\sum_i \varphi_i(p)\Delta p_i}{\sum_i \varphi_i(p)}$$



$$f(p) = p + \frac{\sum_i \gamma_i(p)\varphi_i(p)\Delta p_i}{\sum_i \varphi_i(p)}$$



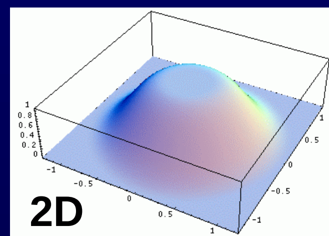
Discontinuity at crossing skeletons,
where $\sum_i \varphi_i(p) > 1$



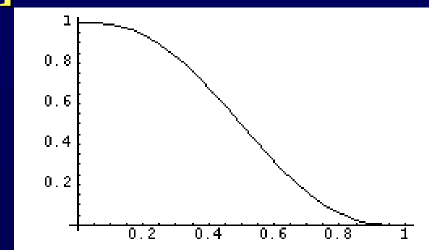
Sweepers

[A.Angelidis et al. ’02]

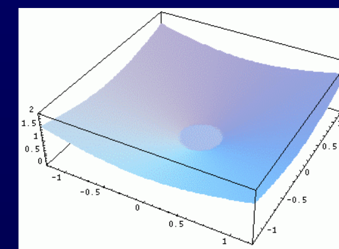
- **Tool = amount of transformation, in [0,1]**
  - **Smooth**
  - **Local**



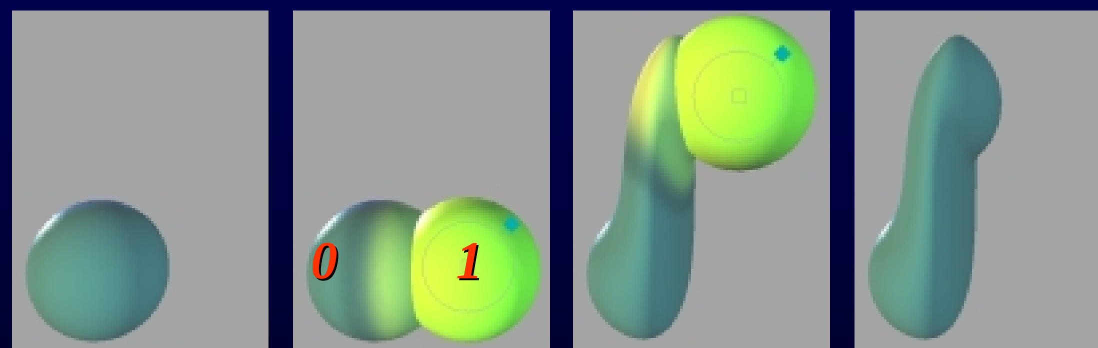
=



∘



- **User input = transformation**
  - **Translation, scale, rotation, modulated with amount of transformation**



- **What is a natural way of taking fraction of a transformation ?**
  - **Matrix exponentiation**

$$f(p) = M^{\varphi(p)} p = \exp(\varphi(p)\log(M))\,p$$

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

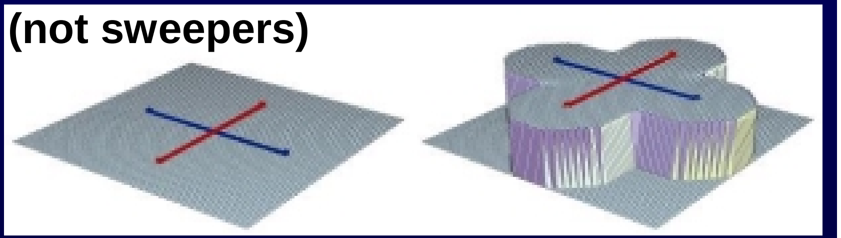
$$\log(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} \ldots$$

Sweepers

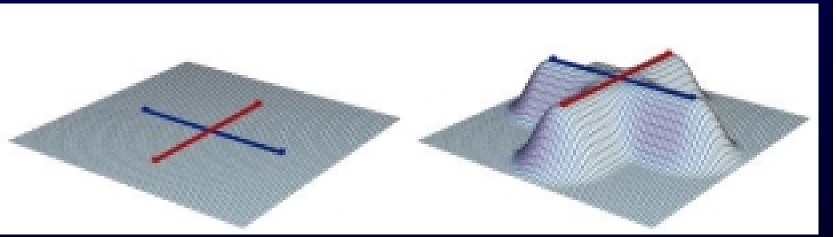
[A.Angelidis et al. ’02]

- **Blending multiple tools**

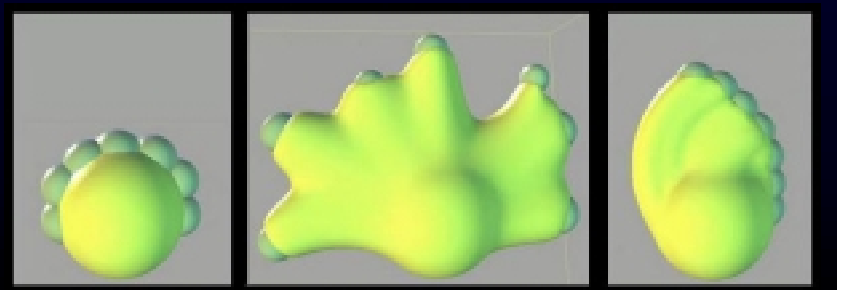
$$f(p) = p + \frac{\sum_i \varphi_i(p)\Delta p_i}{\sum_i \varphi_i(p)}$$



$$f(p) = p + \frac{\sum_i \varphi_i(p)\Delta p_i}{\sum_i \varphi_i(p)}\left(1-\prod_i\left(1-\varphi_i(p)\right)\right)$$



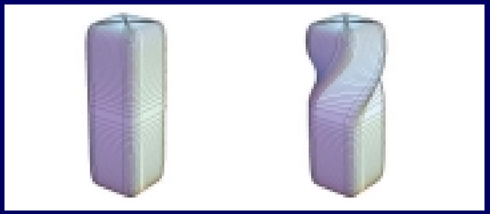
$$f(p) = \exp\left(\frac{\left(1-\prod_i\left(1-\varphi_i(p)\right)\right)\sum_i \varphi_i(p)\log M_i}{\sum_i \varphi_i(p)}\right)p$$

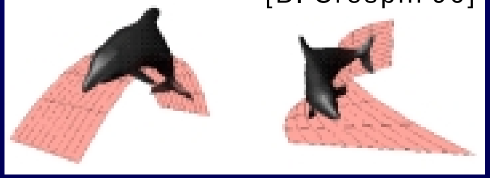


Outline

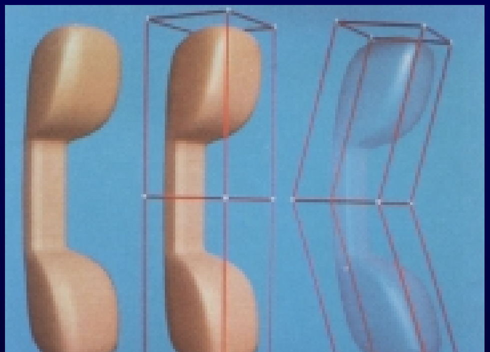
- More control
  - Axial Space Deformations
  - Surface Space Deformations
  - Lattice Space Deformations
  - Specialized Space Deformations

- ➡ **More on Space Deformation**
  - Morphing, modeling, animation, rendering, blending, coherency and volume.

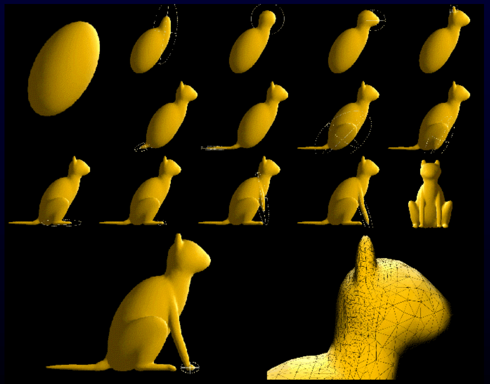




[B. Crespin'96]



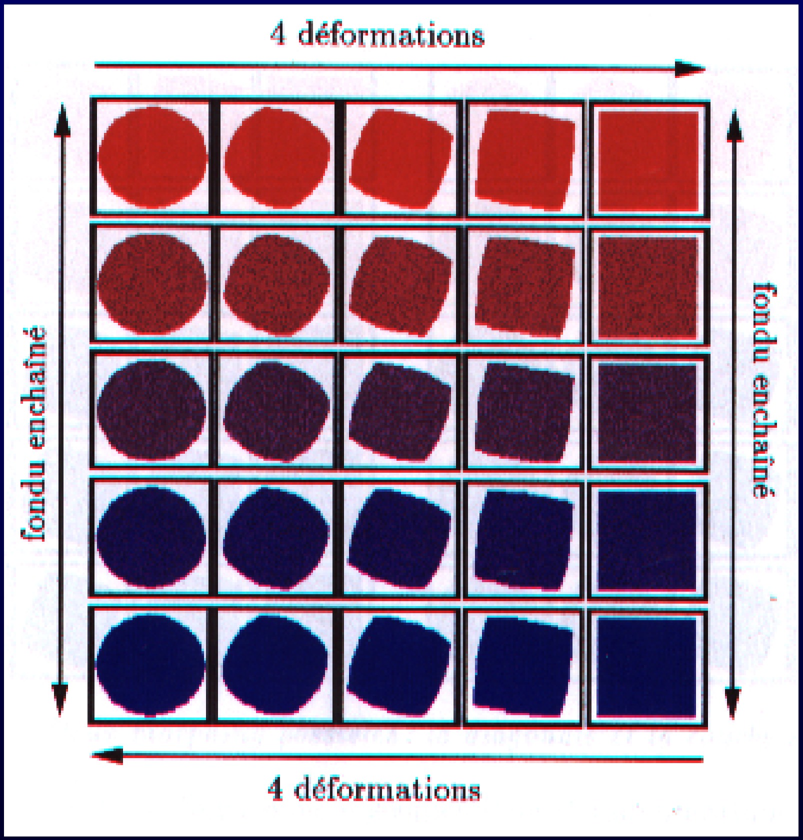
[T.Sederberg,S.Parry'86]

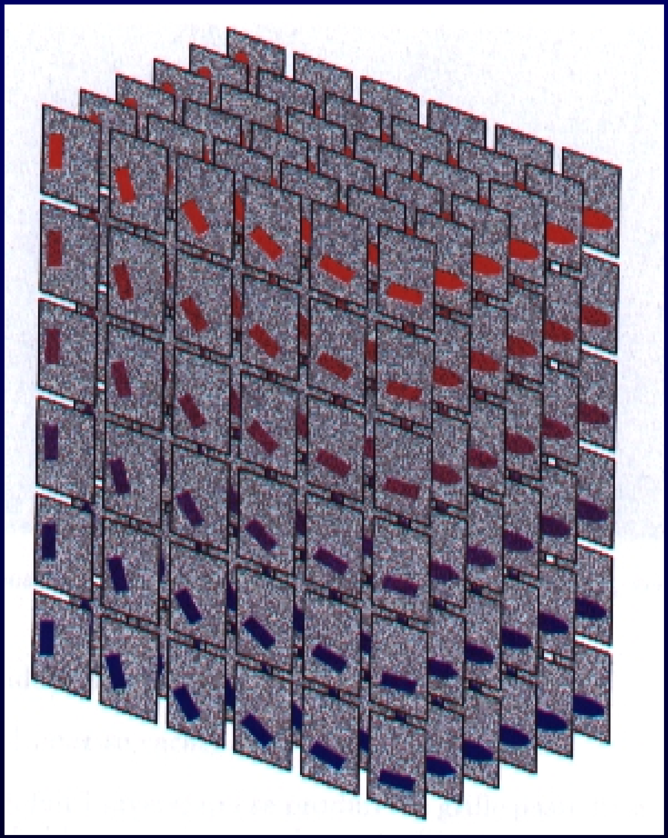


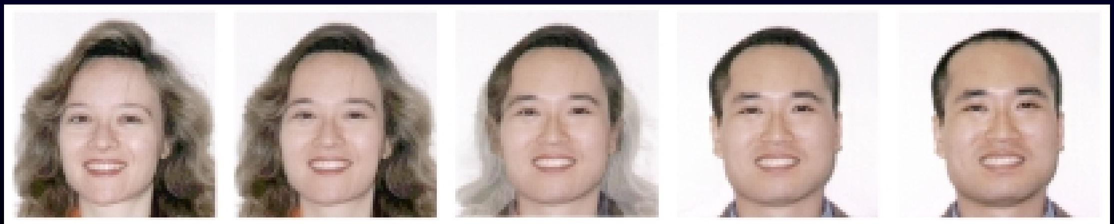
[P.Decaudin'96]

Morphing

2D change of shape & color







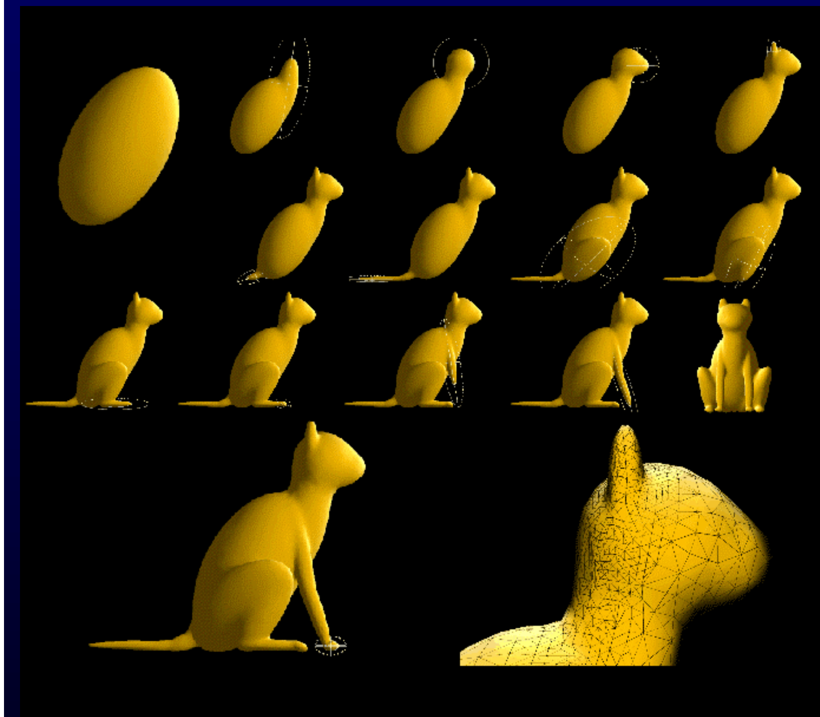
[Lee, Chwa, Shin'95]

Deformations handle
change of shape
Color is some other function

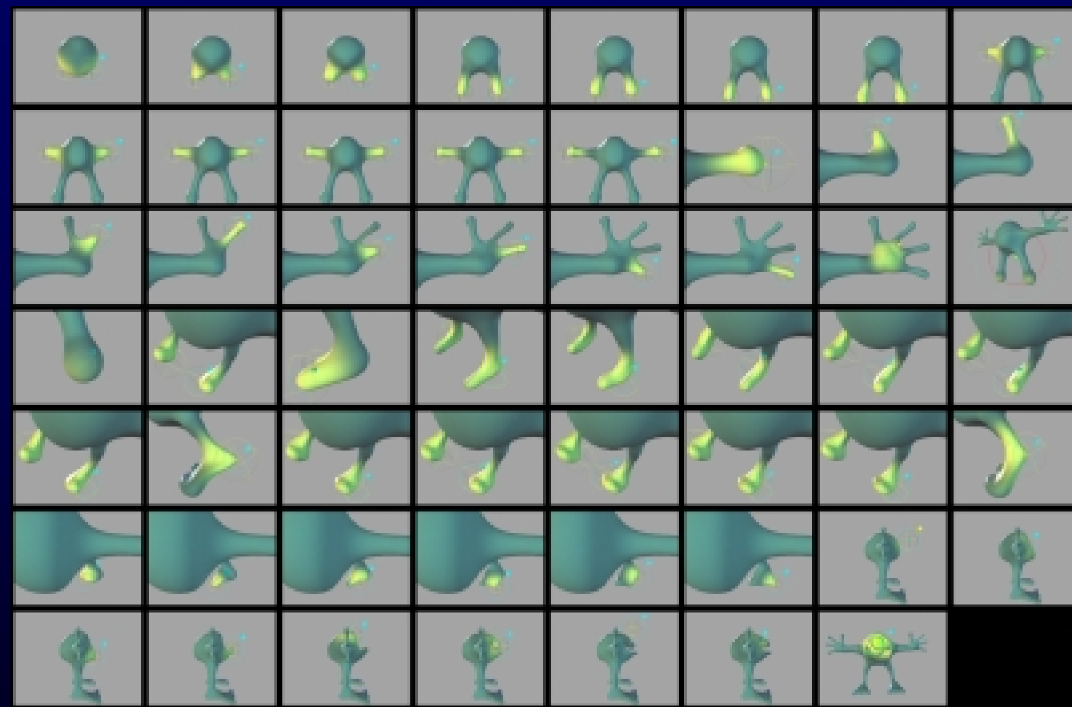
Modeling

- **Apply a series of functions:**

$$S_n(p) = f_n(f_{n-1}(\ldots f_3(f_2(f_1(S_0)))))$$



[P.Decaudin'96]

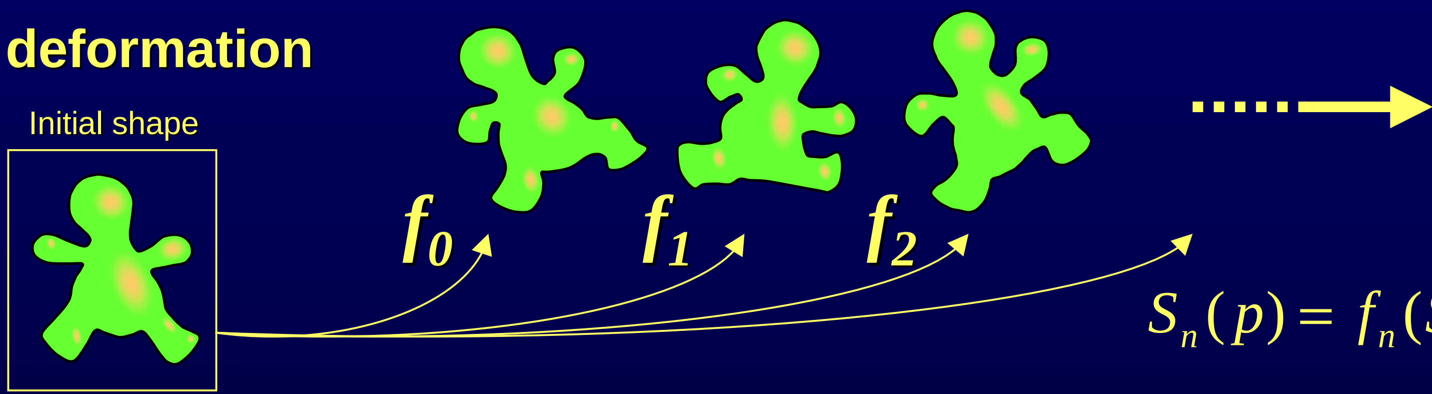


[A.Angelidis et al.'04]

- **History**
  - **undo**
  - **a description of the shape in itself**

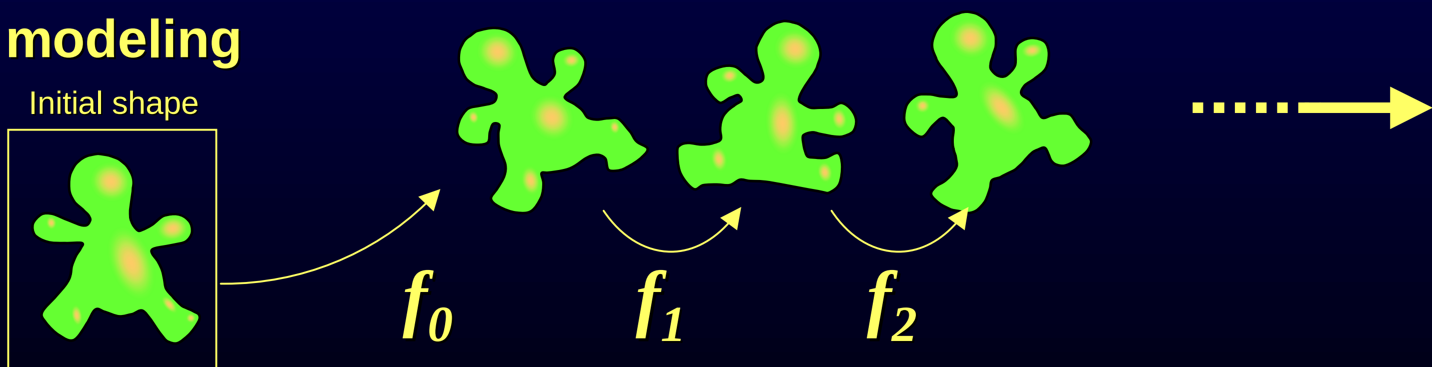
Animation

- **Animated deformation**



$$S_n(p) = f_n(S_0)$$

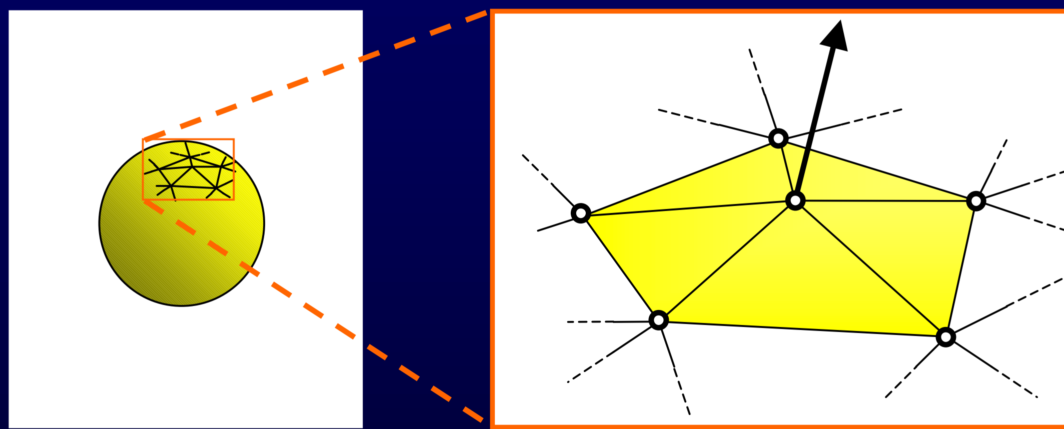
- **Animated modeling**



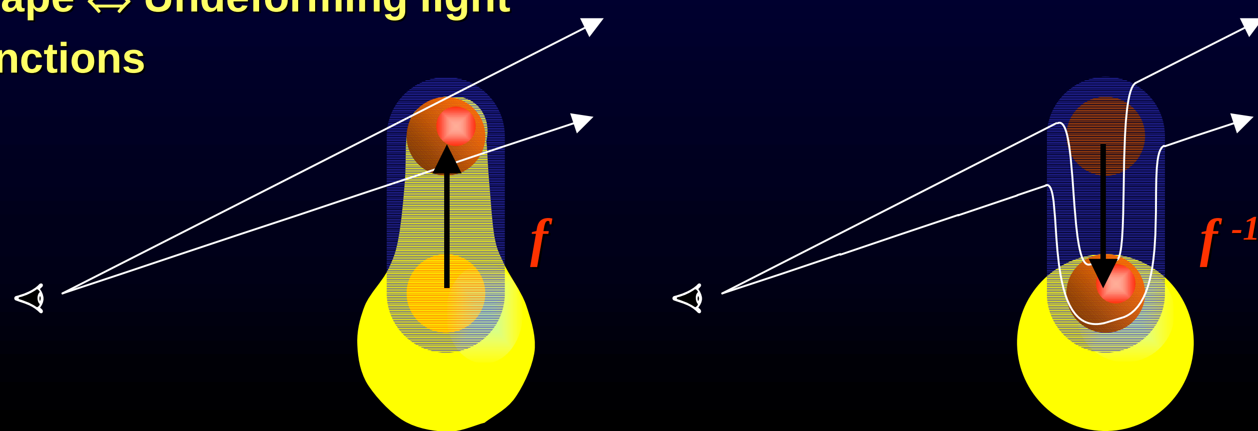
$$S_n(p) = f_n(\dots f_2(f_1(S_0)))$$

Rendering

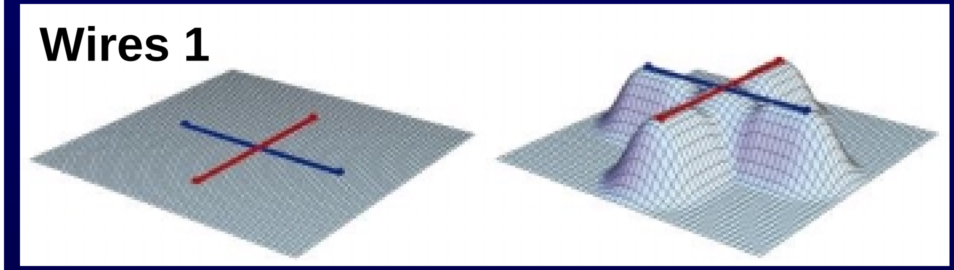
- **Normals/tangent** $f(\vec{n}) = J^{-1T}\vec{n}$ $J = \begin{pmatrix} \frac{\delta f}{\delta x} & \frac{\delta f}{\delta y} & \frac{\delta f}{\delta z} \end{pmatrix}$

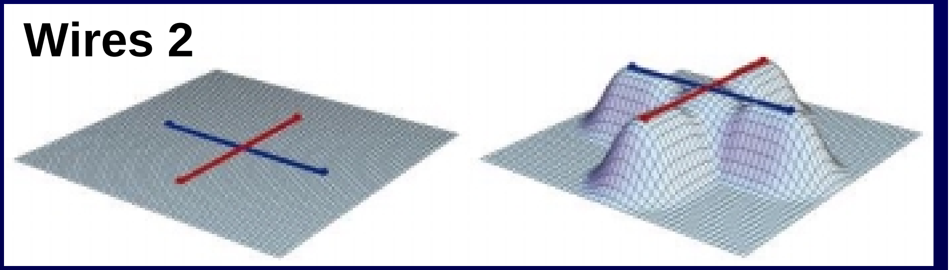


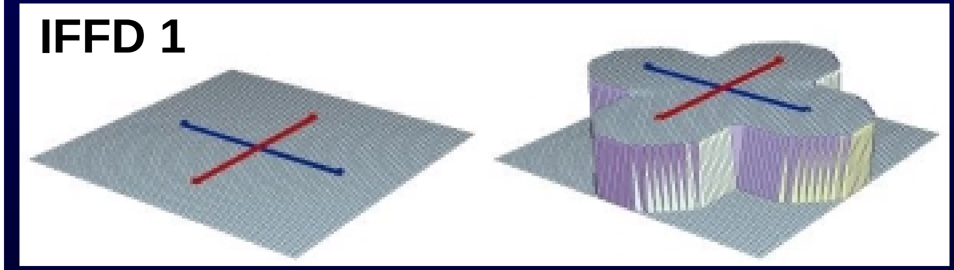
- **Undeformable shapes**
  - **Deforming shape ⇔ Undeforming light**
  - **Reversible functions**

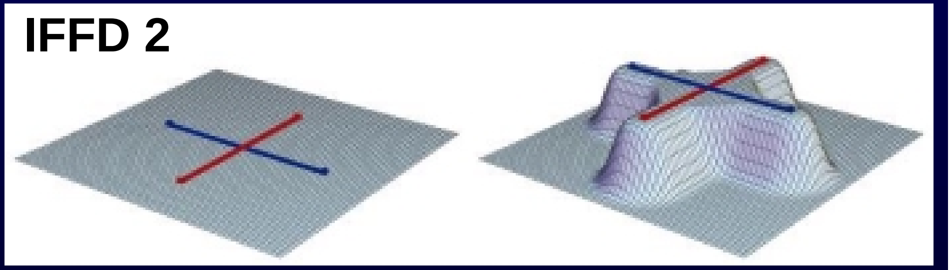


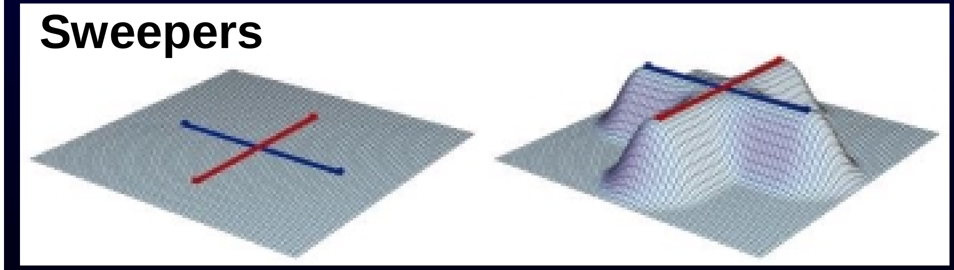
Blending









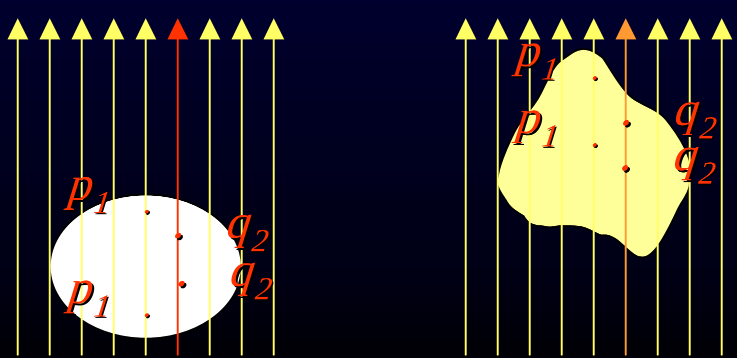


Blendeforming (1/2)

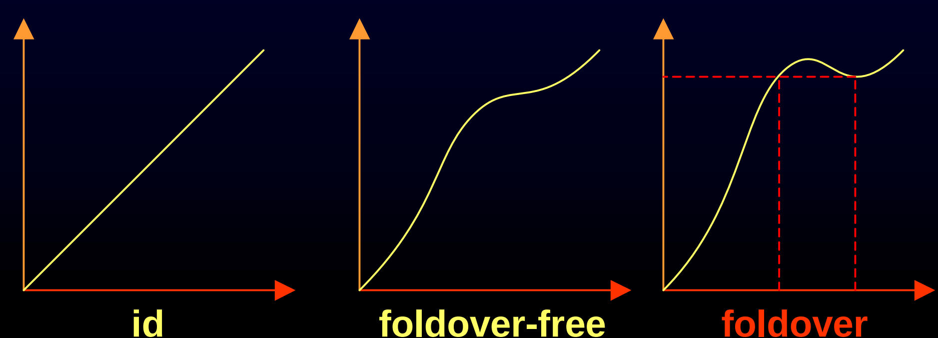
[D.Mason & G.Wyvill'00]

- What is a self-intersection?
  - surface incoherency

- How can they appear?
  - foldover: deformation is surjective
- Cure foldover ⇒ bound *slope* of deformation.
- Blendeforming:
  deformation follows lines of flux ⇒ find a solution for individual lines



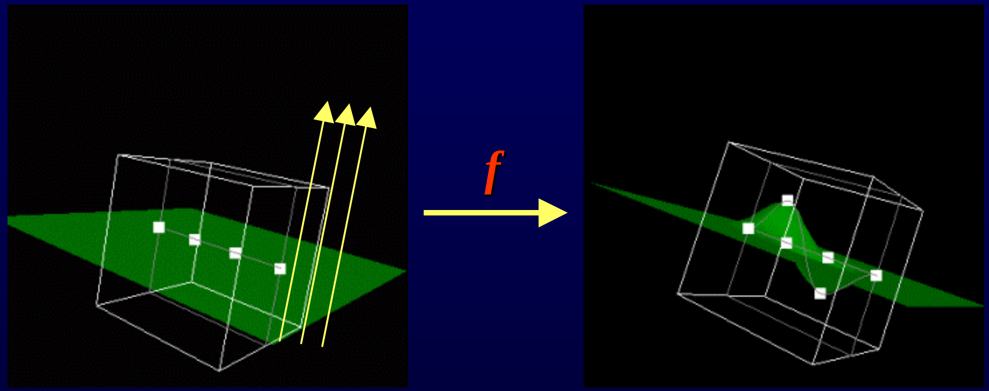
problem solved in 1D ⇒ solved in 3D



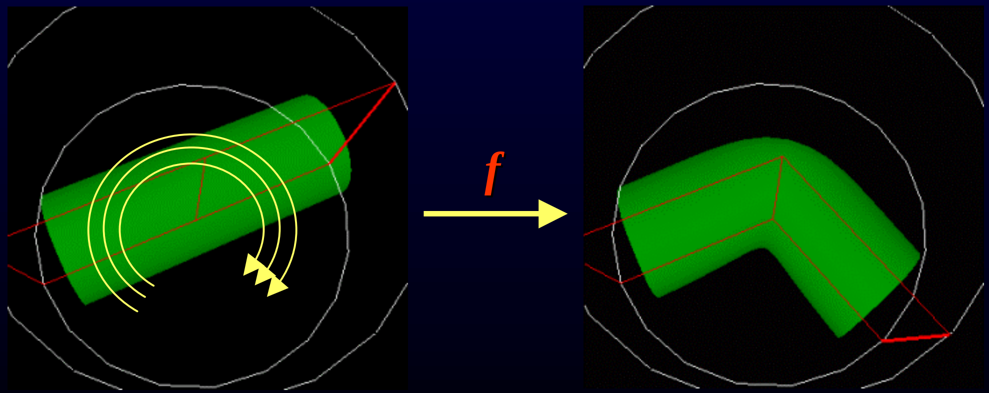
Blendeforming (2/2)

[D.Mason & G.Wyvill'00]

- **Examples**
  - **Straight lines of flux**



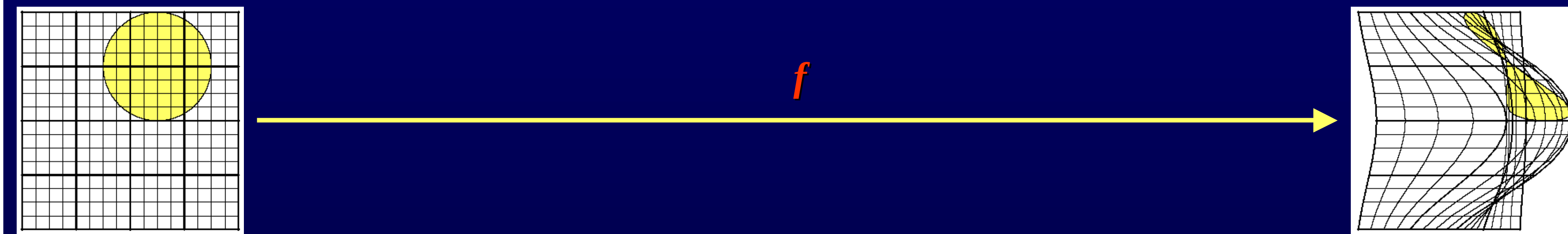
  - **Circular lines of flux**



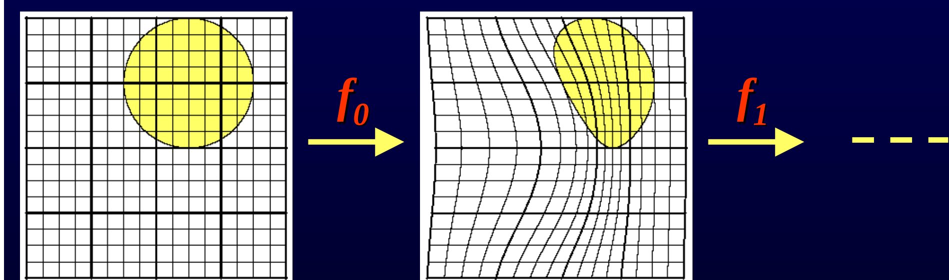
Foldover-free DMFFD

[J.Gain & N.Dodgson'01]

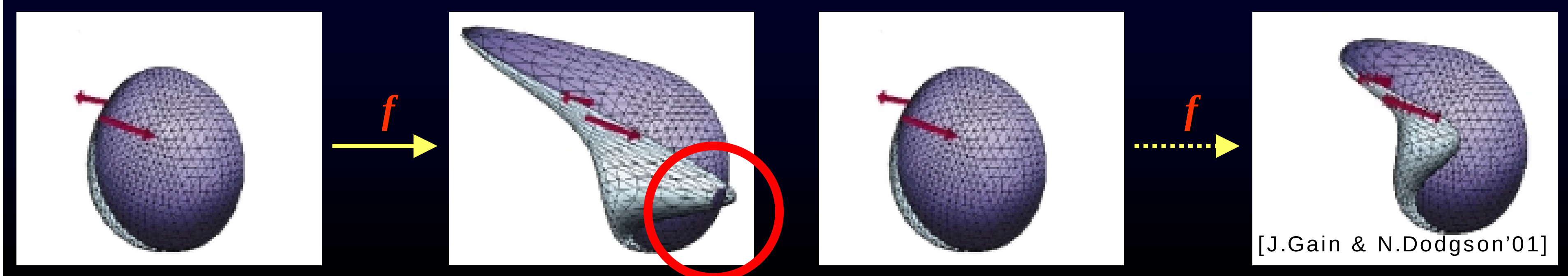
- **Foldover with FFD: destroys coherency**



- **Cure: decompose large DMFFD in a series of "small-enough" DMFFD**



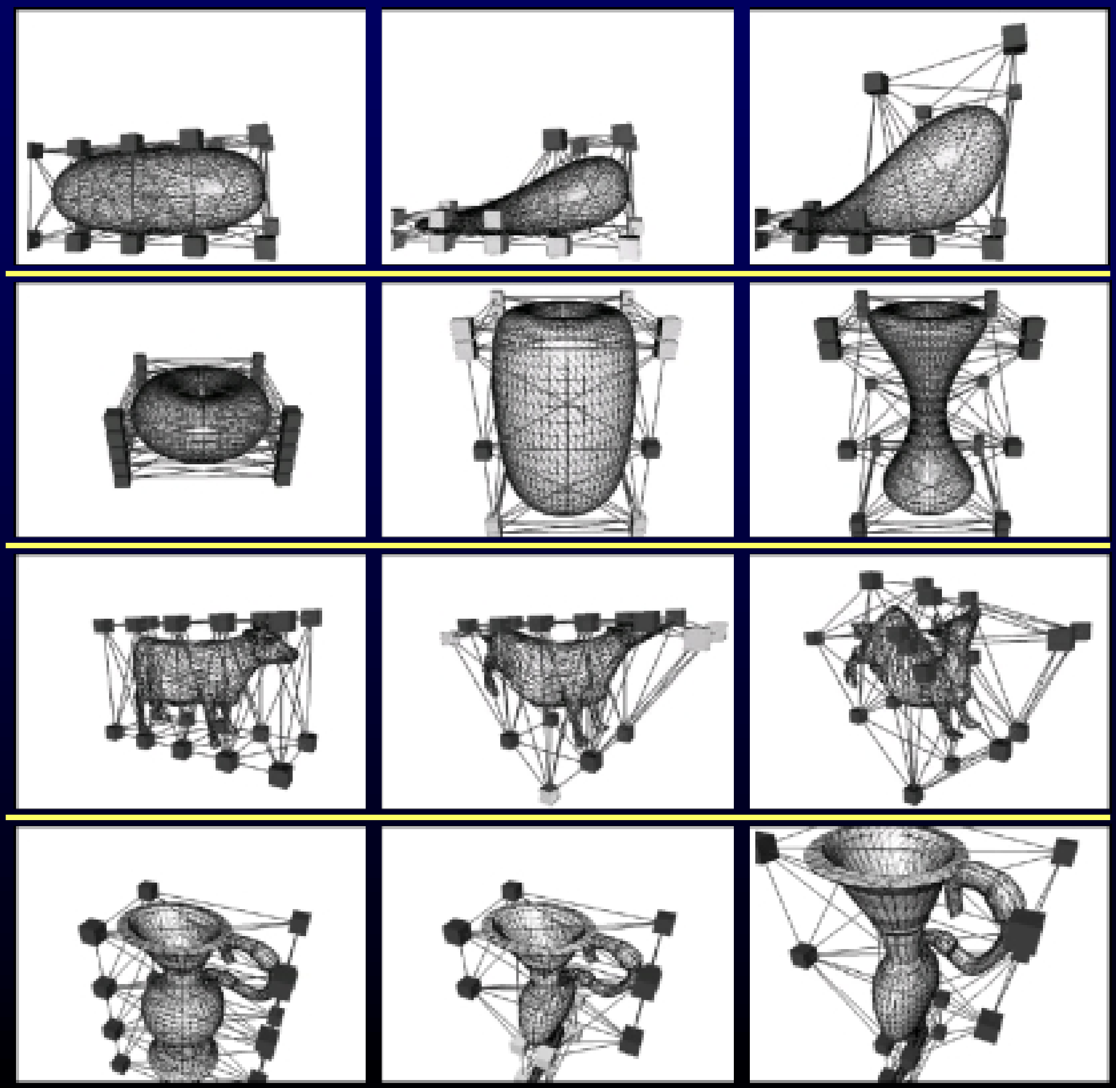
- **In 3D**



Volume with FFD

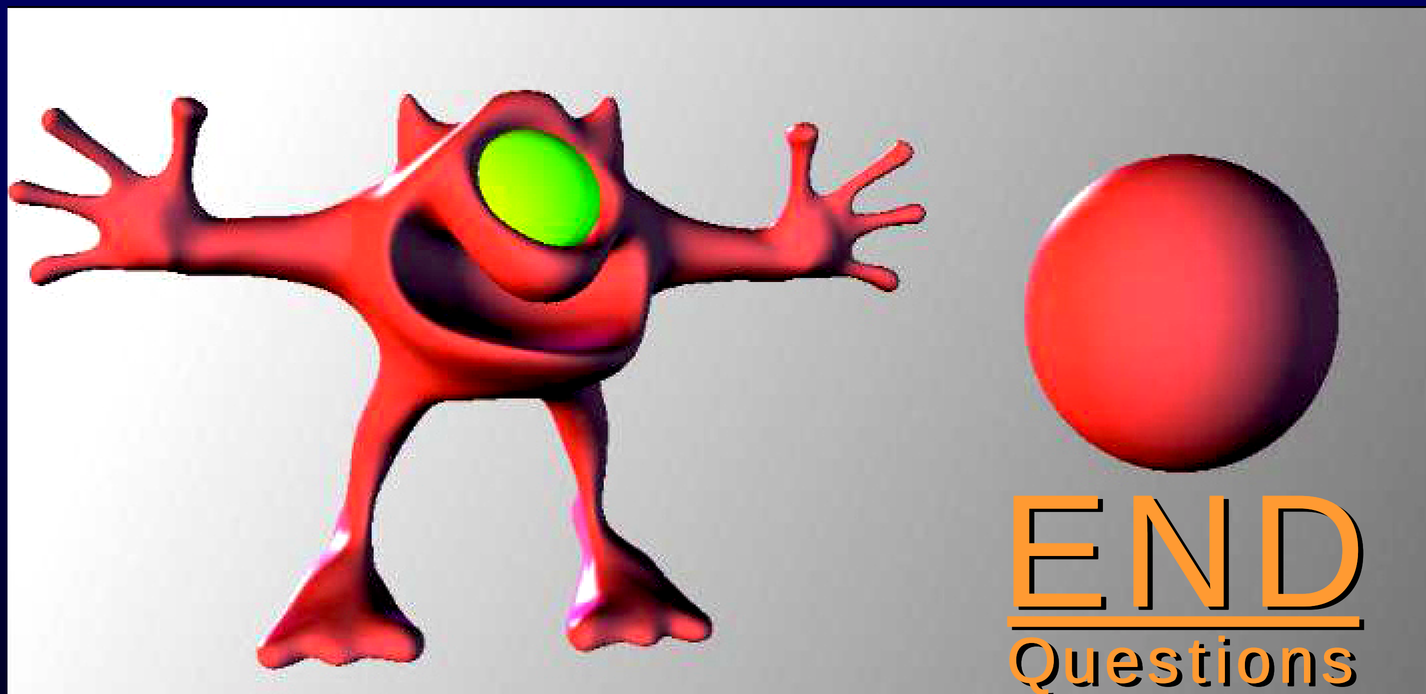
[G.Hirota R.Maheshwari & M.Lin'92]

- **Quantity of material : preserve volume.**



Conclusion

- **Space Deformation is compatible with vertex shaders**
- **Accurate rendering of a deformed shape is still an issue**
- **The race to popularity**
  - **Implicit surfaces = scalar field**
  - **Space deformation = vector field (or worse)**


END
Questions