Deformable Models & Applications (Part I)

Ye Duan

Department of Computer Science
University of Missouri at Columbia

December 21, 2004

Outline

- Overview
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

Outline

- Overview
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

Why Deformable Models?

- Very successful for visual computing.
- Combines knowledge from mathematics, physics and mechanics.
- Inherited smoothness, more robust to noise.
- Built-in dynamic behavior, well suited for time-varying phenomena.

What is a deformable Model?

- A deformable model is a geometric object whose shape can change over time.
- The deformation behavior of a deformable model is governed by variational principles (VPs) and/or partial differential equations (PDEs).

Deformable Model

- Introduced by Terzopoulos et al. in late 80s.
- Consists of deformable curves, deformable surfaces, and deformable solids.

Outline

- Overview
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

Outline

- Overview
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

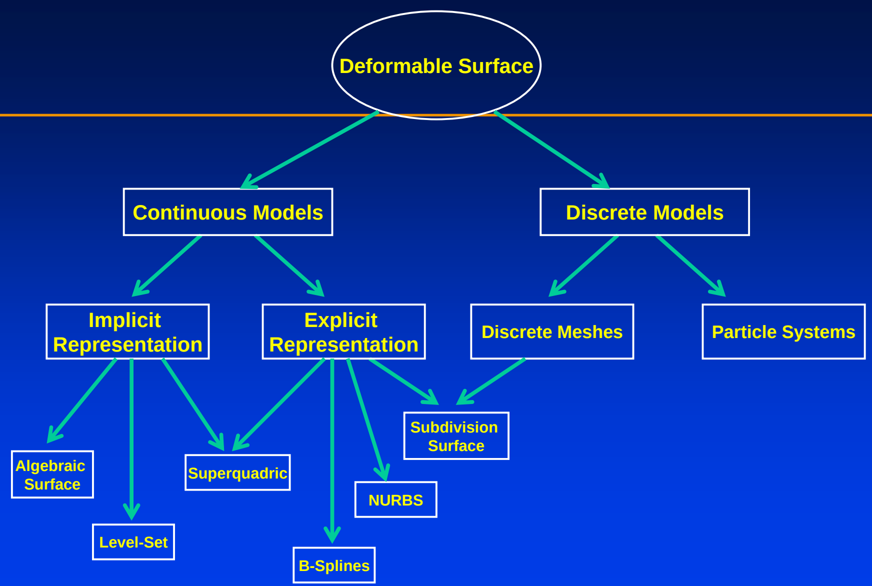
Deformable Surface

Three components:

- Geometric representation
- Evolution law
- Topology change

Geometric Representations

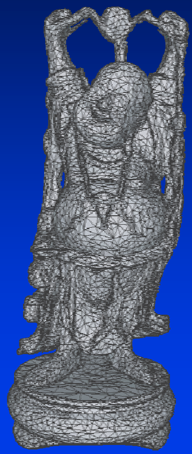
- Continuous representation
  - Explicit representation
  - Implicit representation
- Discrete representation
  - Triangle meshes
  - Points/Particles



Surfaces

- Triangle meshes.
- Tensor-product surfaces.
  - Hermite surface, Bezier surface, B-spline surfaces, NURBS.
- Non-tensor product surfaces.
  - Sweeping surface, ruling surface, etc.
- Subdivision surfaces.
- Implicit surfaces.
- Particle systems.

Triangle Meshes



Quadratic Surfaces

- Implicit representation

$$a_0x^2+a_1y^2+a_2z^2+a_3xy+a_4xz+a_5yz+a_6x+a_7y+a_8z+a_9$$

- Sphere

$$x^2+y^2+z^2-r^2=0$$

- Ellipsoid

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}-1=0$$

Tensor Product Surface

- From curves to surfaces
- A simple curve example (Bezier)

$$\mathbf{c}(u)=\sum_{i=0}^{3}\mathbf{p}_iB_i(u)$$

where $u \in [0,1]$

- Consider p_i is a curve $p_i(v)$
- In particular, if p_i is also a bezier curve, where $v \in [0,1]$

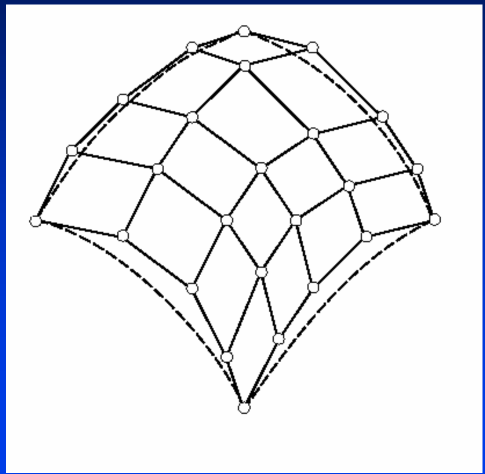
$$\mathbf{p}_i=\sum_{j=0}^{3}\mathbf{p}_{i,j}B_j(v)$$

From curve to surface

- Then we have

$$\mathbf{s}(u,v)=\sum_{i=0}^{3}(\sum_{j=0}^{3}\mathbf{p}_{i,j}B_j(v))B_i(u)=\sum_{i=0}^{3}\sum_{j=0}^{3}\mathbf{p}_{i,j}B_i(u)B_j(v)$$

Bezier Surface



B-Splines Surface

- B-Spline curves

$$c(u) = \sum_{i=0}^{n} p_i B_{i,k}(u)$$

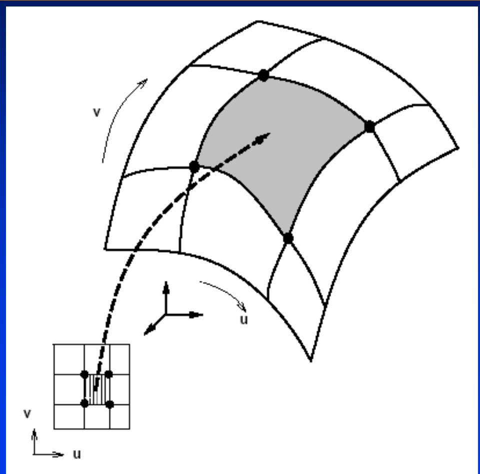
- Tensor product B-splines

$$\mathbf{s}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{p}_{i,j} B_{i,k}(u) B_{j,l}(v)$$

where u ∈[0,1], and v ∈[0,1]

- Can we get NURBS surface this way?

B-Splines Surface



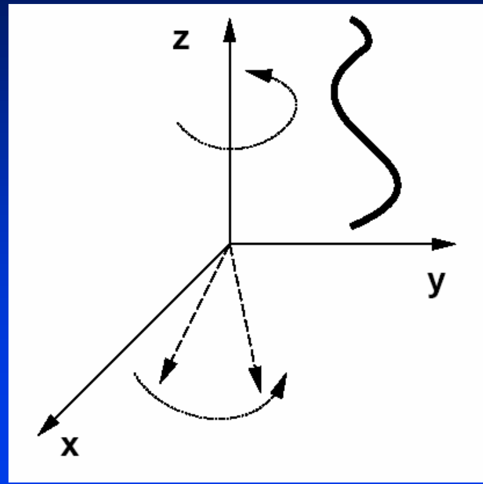
Tensor Product Properties

- Inherit from their curve generators.
- Continuity across boundaries
- Interpolation and approximation tools.

NURBS Surface

$$\mathbf{s}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{P}_{i,j} w_{i,j} B_{i,k}(u) B_{j,l}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} B_{i,k}(u) B_{i,j}(v)}$$

Surface of Revolution



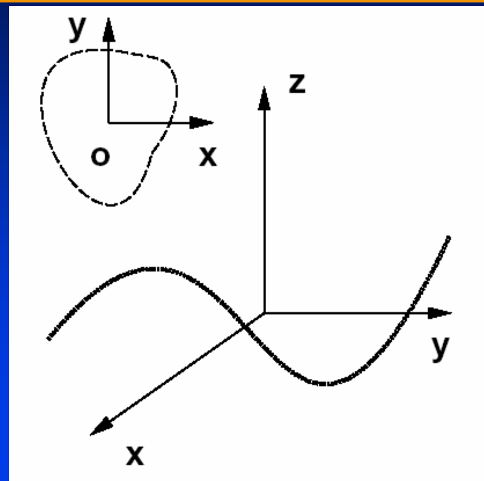
Surface of Revolution

- Geometric construction
  - Specify a planar curve profile on y-z plane
  - Rotate this profile with respect to z-axis
- Procedure-based model

$$\mathbf{c}(u) = \begin{bmatrix} 0 \\ y(u) \\ z(u) \end{bmatrix}$$

$$\mathbf{s}(u,v) = \begin{bmatrix} -y(u)\sin v \\ y(u)\cos v \\ z(u) \end{bmatrix}$$

Sweeping Surface



Sweeping Surface

- Surface of revolution is a special case of a sweeping surface.
- Idea: a profile curve and a trajectory curve.
- Move a profile curve along a trajectory curve to generate a sweeping surface.

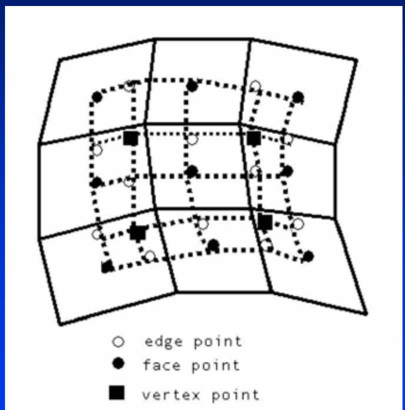
Subdivision Scheme

Overview:

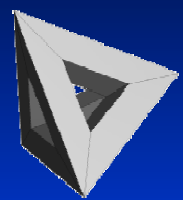
- Start from an initial control polygon.
- Recursively refine it by some rules.
- A smooth surface in the limit.

$$s(x, p) = J(x)p$$

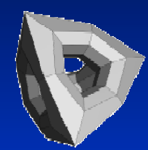
Catmull-Clark Scheme



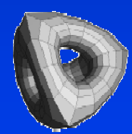
Catmull-Clark Subdivision Surface



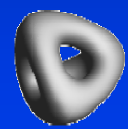
Initial mesh



Step 1



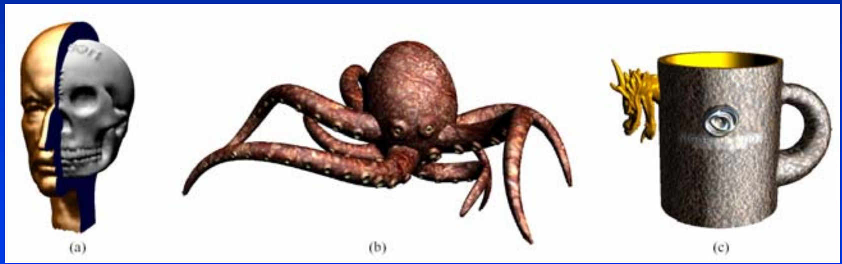
Step 2



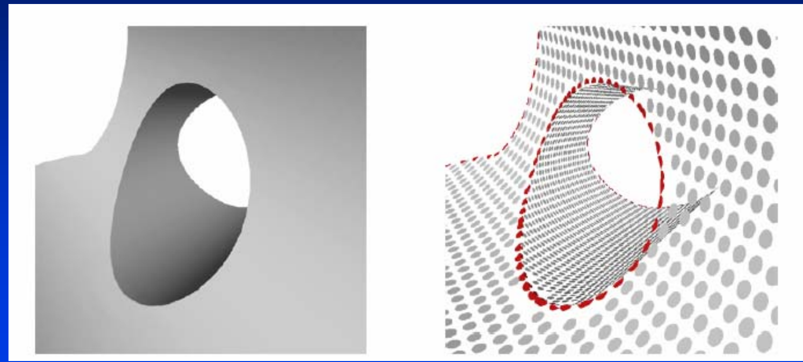
Limit surface

Points

- Very popular primitives for modeling, animation, and rendering.



Points

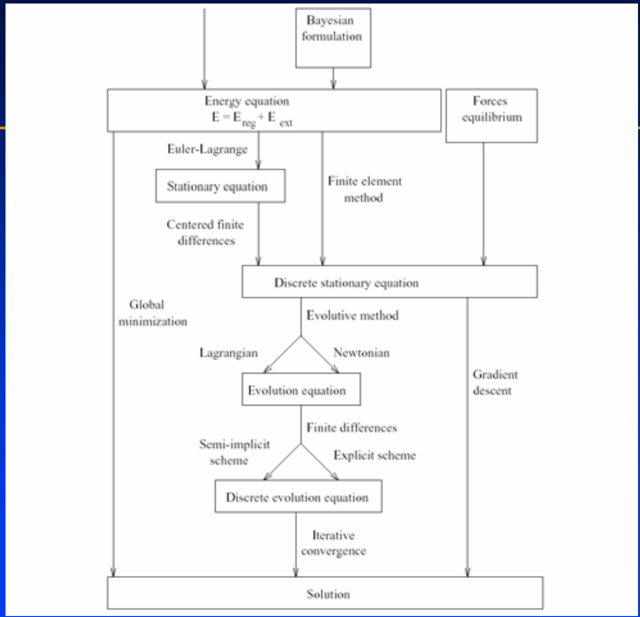


Outline

- Overview
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

Deformable Surface Evolution

- The deformation involves a data term and a regularization term.
- Data is used to drive the model deformation toward the boundary.
- Regularization term enforcing a smooth behavior of the model.
  - Combat noise or outliers, stabilize the model evolution.



Energy-Minimizing Deformable Models

For a parametric Contour: $v(s) = (x(s), y(s), z(s))^T$

Define an energy function: $\varepsilon(v) = S(v) + P(v)$

With regular term: $S(v) = \int_0^1 w_1(s) \left| \frac{\partial v}{\partial s} \right|^2 + w_2(s) \left| \frac{\partial^2 v}{\partial s^2} \right|^2 ds$

Data term: $P(v) = \int_0^1 p(v(s)) ds$

$$p(u) = -c \left| \nabla [G_\sigma * I(u)] \right|$$

Energy-Minimizing Deformable Models

Obtain a Euler-Lagrange equation by calculus of variations:

$$-\frac{\partial}{\partial s}\left(w_1 \frac{\partial v}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2 \frac{\partial^2 v}{\partial s^2}\right) + \nabla P(v(s,t)) = 0$$

Gradient-Descent Based Energy Minimization

Evolve an initial surface in the steepest energy direction by the following equation:

$$S_{k+1} = S_k - \Delta t \, \nabla E(S_k)$$

Energy-Minimizing Deformable Models

By gradient descent, obtain a dynamic equation:

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial s}(w_1 \frac{\partial v}{\partial s}) - \frac{\partial^2}{\partial s^2}(w_2 \frac{\partial^2 v}{\partial s^2}) - \nabla P(v(s,t))$$

Deformable Surface Evolution

In general, the deformation of a deformable surface $S(t)$ can be described by an evolution equation:

$$\frac{\partial S}{\partial t} = F(S, N, K, f, ...)$$

F: speed vector

N: surface normal

K: surface curvature

f: internal and external force.

Second order Lagrange dynamic equation

Lagrange equations of motion is obtained by associating a mass density function $\mu(s)$ and a damping density $\gamma(s)$:

$$\mu \frac{\partial^2 v}{\partial t^2} + \gamma \frac{\partial v}{\partial t} - \frac{\partial}{\partial s}(w_1 \frac{\partial v}{\partial s}) + \frac{\partial^2}{\partial s^2}(w_2 \frac{\partial^2 v}{\partial s^2}) = -\nabla P(v(s,t))$$

Discretized Lagrange dynamic equation

$$M\ddot{u} + D\dot{u} + Ku = f$$

Can be solved using finite difference or finite element.

Probabilistic Deformable Models

- Casting the model fitting process in a probabilistic framework.
- Incorporation of prior model and sensor model characteristics in terms of probability distributions.
- Provides a measure of uncertainty of the estimated shape parameters after the model is fitted to the image data.

Probabilistic Deformable Models

- Let u be the shape parameters with a prior probability p(u)
- Let p(I|u) be the imaging model
  - The probability of producing an image I given a model u.
- Bayes' theorem:

  p(u|I) = p(I|u)p(u)/p(I)

  expresses the posterior probability p(u|I) of a model given the image.
- Assume a Gibbs distribution of the form:

  p(u)=exp(-S(u))/Z_s

  and p(I|u)=exp(-p(u))/Z_I

  Then model is fitted by finding u maximizing p(u|I), called maximum a posteriori (MAP).

  A Kalman filter can be used for a time-varying prior model.

Outline

- Overview
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

State-Of-Art Deformable Models

- Explicit parametric model
  - Snake: Kass, Witkin and Terzopoulos, 87.
  - D-Superquadrics: Metaxas and Terzopoulos, 92.
  - D-NURBS: Qin and Terzopoulos, 94.
  - T-Snake: McInerney and Terzopoulos 95
  - Oriented-Particles: Szeliski and Terzopoulos, 95
  - D-Subdivision: Qin and Mandal. 98.
  - …
- Implicit level-set model
  - Osher and Sethian, 88.
  - Malladi, Sethian and Vemuri, 95.
  - Caselles, Kimmel, and Sapiro, 95.
  - Zhao, Osher and Fedkiw 2001.
  - ...

State-Of-Art Deformable Models

- Explicit parametric model
  - Snake: Kass, Witkin and Terzopoulos, 87.
  - D-Superquadrics: Metaxas and Terzopoulos, 92.
  - D-NURBS: Qin and Terzopoulos, 94.
  - T-Snake: McInerney and Terzopoulos 95
  - Oriented-Particles: Szeliski and Terzopoulos, 95
  - D-Subdivision: Qin and Mandal. 98.
  - …
- Implicit level-set model
  - Osher and Sethian, 88.
  - Malladi, Sethian and Vemuri, 95.
  - Caselles, Kimmel, and Sapiro, 95.
  - Zhao, Osher and Fedkiw 2001.
  - ...

Snake: Active Contour Models

History

- A seminal work in Computer vision, and imaging processing.
- Appeared in the first ICCV conference in 1987.
- Michael Kass, Andrew Witkin, and Demetri Terzopoulos.

Overview

- A snake is an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges.
- Snakes are active contour models: they lock onto nearby edges, localizing them accurately.
- Snakes are very useful for feature/edge detection, motion tracking, and stereo matching.

Energy Formulation

Parametric splines

$$\mathbf{v}(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix}$$

Energy functional

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))ds$$

$$= \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))ds$$

where

E_{int}: internal energy

E_{image}: image energy

E_{con}: constraint energy

Energy Formulation

Internal spline energy (membrane + thin-plate)

$$E_{int}(\mathbf{v}(s)) = \frac{1}{2}(\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2)$$

Image energy (line + edge + termination)

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}$$

Line functional

$$E_{line} = I(x,y)$$

Edge functional

$$E_{edge} = -|\nabla I(x,y)|^2$$

where

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

Energy Formulation

Termination functional

$$E_{term} = \frac{\partial \theta}{\partial \mathbf{n}_\perp} = \frac{\partial^2 C / \partial \mathbf{n}_\perp^2}{\partial C / \partial \mathbf{n}}$$

$$= \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}}$$

where

$$C(x,y) = G_\sigma(x,y) * I(x,y)$$

$$\tan(\theta) = \frac{C_y}{C_x}$$

$$\mathbf{n} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

Feature detection need domain knowledge

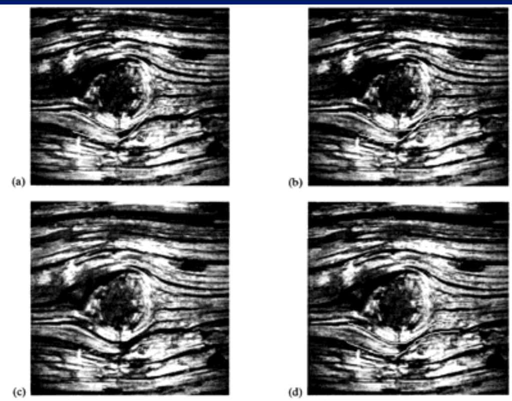


Fig. 1. Lower-left: Original wood photograph from Brodatz. Others: Three different local minima for the active contour model.

Why called snake?

- Because of the way the contour slither while minimizing their energy, they are called snakes.
- The model is active.
- It is always minimizing its energy functional and therefore exhibits dynamic behavior.
- Snakes exhibit hysteresis when exposed to moving stimuli.

Snake

- Initialization is not automatically done.
- It is an example of matching a deformable model to an image by means of energy minimization.

Basic snake behavior

- It is a controlled continuity spline under the influence of image forces and external constraint forces.
- The internal spline forces serve to impose a piecewise smoothness constraint.
- The image forces push the snake toward salient image features such as line, edges, and subjective contours.
- The external constraint forces are responsible for putting the snake near the desired minimum.

External constraint forces

- The user can connect a spring to any point on a snake.
- The other end of the spring can be anchored at a fixed position, connected to another point on a snake, or dragged around using the mouse.
- Creating a spring between x1 and x2 simply adds –k(x1-x2)^2 to the external energy E_{con}.
- In addition to springs, the user interface provides a 1/r^2 repulsion force controllable by the mouse.

Snake Pit

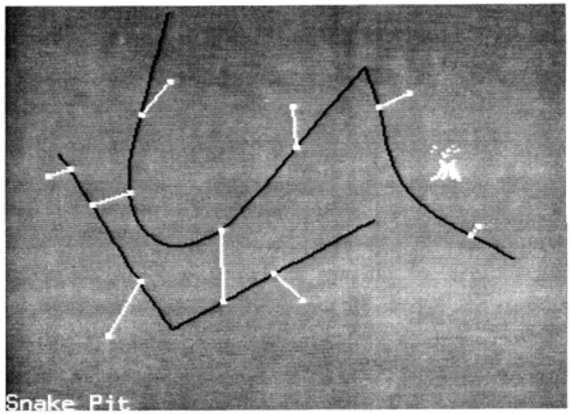


Fig. 2. The Snake Pit user-interface. Snakes are shown in black, springs and the volcano are in white.

Edge detection

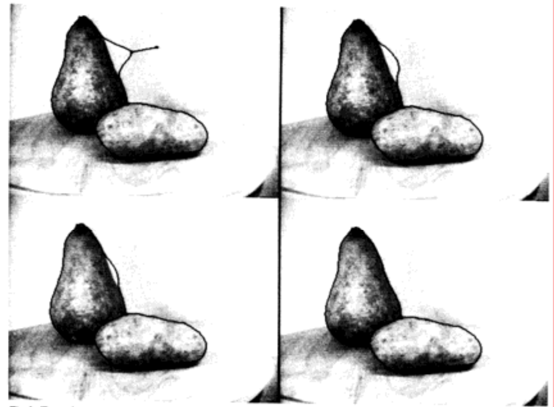


Fig. 3. Two edge snakes on a pear and potato. Upper-left: The user has pulled one of the snakes away from the edge of the pear. Others: After the user lets go, the snake snaps back to the edge of the pear.

Scale space

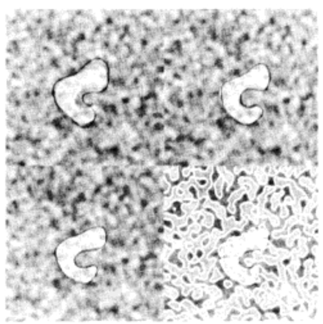


Fig. 4. Upper-left: Edge snake in equilibrium at coarse scale. Upper-right: Snake in equilibrium at intermediate scale. Lower-left: Final snake equilibrium after scale-space continuation. Lower-right: Zero-crossings overlayed on final snake position.

Subjective contour detection

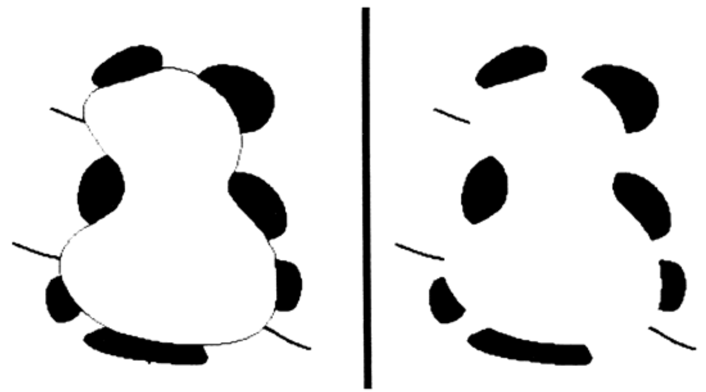


Fig. 5. Right: Standard subjective contour illusion. Left: Edge/termination snake in equilibrium on the subjective contour.

Dynamic subjective contour

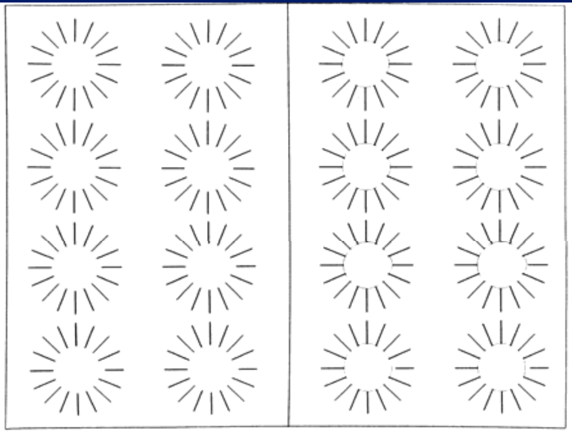


Fig. 6. Above left: Dynamic subjective contour illusion. Sequence is left to right, top to bottom. Above Right: Snake attracted to edges and terminations. As the moving horizontal line slides to the right, the snake bends until it falls off the line. Bringing the line close enough makes the snake reattach.

Stereo vision

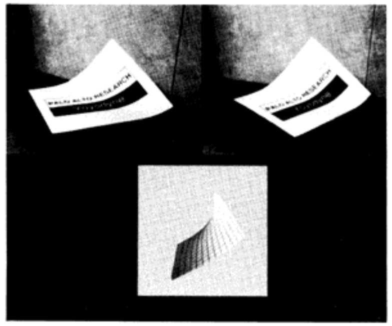


Fig. 7. Bottom: Stereogram of a bent piece of paper. Below: Surface reconstruction from the outline of the paper matched using stereo snakes. The surface model is rendered from a very different viewpoint than the original to emphasize that it is a full 3D model, rather than a 2.5D model.

Motion tracking

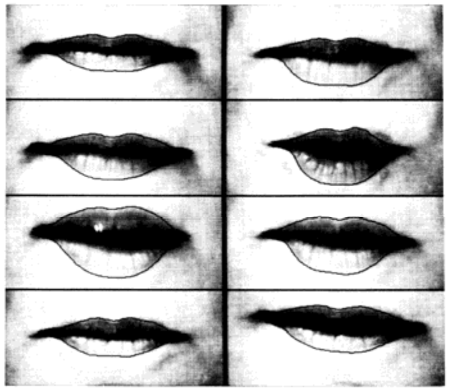
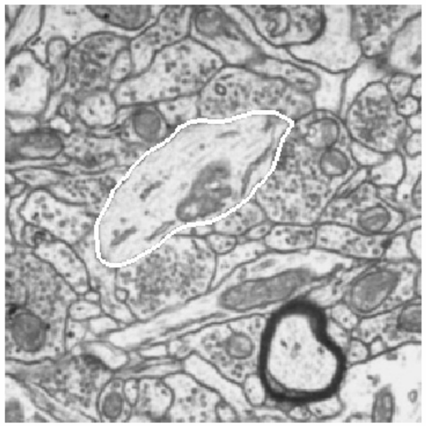
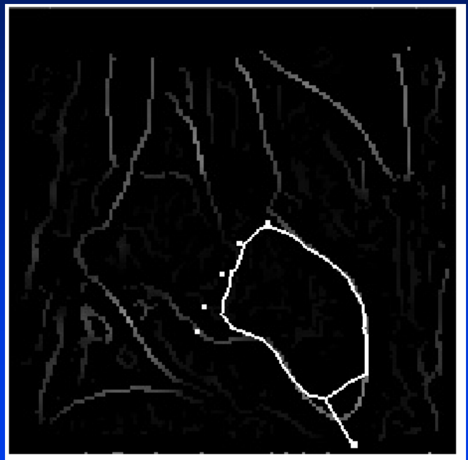


Fig. 8. Selected frames from a 2-second video sequence showing snakes used for motion tracking. After being initialized to the speaker's lips in the first frame, the snakes automatically track the lip movements with high accuracy.

Medical Imaging Segmentation



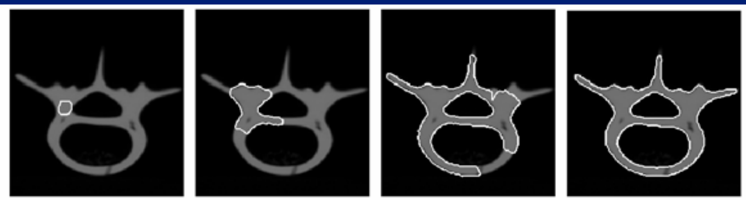
Medical Imaging Segmentation



State-Of-Art Deformable Models

- Explicit parametric model
  - Snake: Kass, Witkin and Terzopoulos, 87.
  - D-Superquadrics: Metaxas and Terzopoulos, 92.
  - D-NURBS: Qin and Terzopoulos, 94.
  - T-Snake: McInerney and Terzopoulos 95
  - Oriented-Particles: Szeliski and Terzopoulos, 95
  - D-Subdivision: Qin and Mandal. 98.
  - …
- Implicit level-set model
  - Osher and Sethian, 88.
  - Malladi, Sethian and Vemuri, 95.
  - Caselles, Kimmel, and Sapiro, 95.
  - Zhao, Osher and Fedkiw 2001.
  - ...

Topology Adaptive Snake



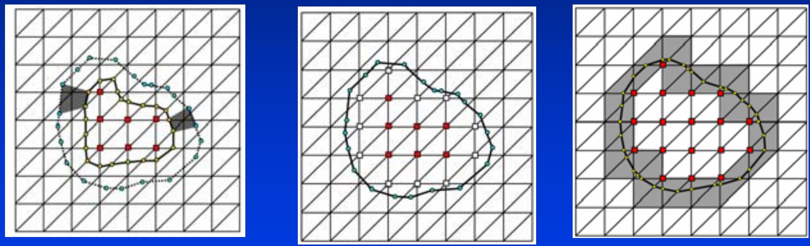
(a) (b) (c) (d)

Figure 5: Segmentation of a cross sectional image of a human vertebra phantom with a topologically adaptable snake (McInerney and Terzopoulos 1995b). The snake begins as a single closed curve and becomes three closed curves.

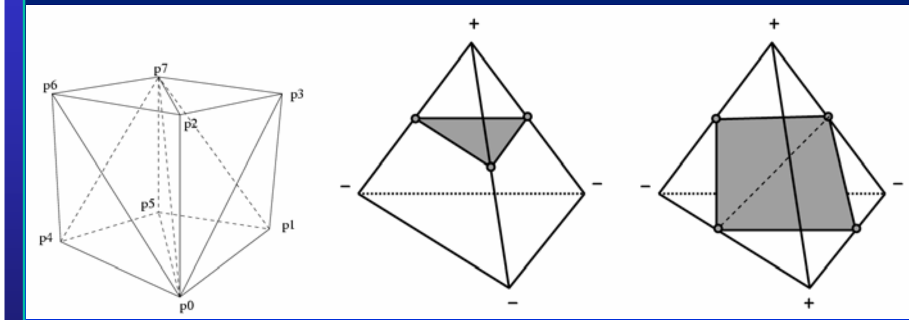
Topology Adaptable Snake

- Introduced by McInerney and Terzopoulos in 1995.
- A deformable discrete contour superimposed with an underlying simplicial grid.
- The deformation of the model is guided by the Lagrangian mechanics of law.
- The geometry and topology of the model is approximated by resampling the triangulation on the simplicial grid.

Topology Adaptable Snake

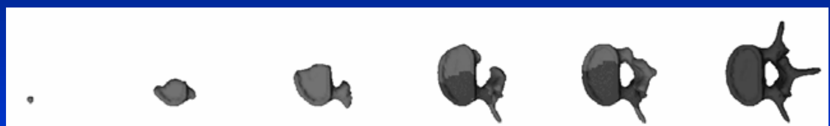


Topology Adaptable Snake



Department of Computer Science University of Missouri at Columbia

Topology Adaptable Snake



Department of Computer Science University of Missouri at Columbia

State-Of-Art Deformable Models

- Explicit parametric model
  - Snake: Kass, Witkin and Terzopoulos, 87.
  - D-Superquadrics: Metaxas and Terzopoulos, 92.
  - D-NURBS: Qin and Terzopoulos, 94.
  - T-Snake: McInerney and Terzopoulos 95
  - Oriented-Particles: Szeliski and Terzopoulos, 95
  - D-Subdivision: Qin and Mandal. 98.
  - …
- Implicit level-set model
  - Osher and Sethian, 88.
  - Malladi, Sethian and Vemuri, 95.
  - Caselles, Kimmel, and Sapiro, 95.
  - Zhao, Osher and Fedkiw 2001.
  - ...

Department of Computer Science University of Missouri at Columbia

Dynamic Subdivision Surface

- Introduced by Qin and Mandal, 98.
- Integrate physics-based modeling with geometric subdivision schemes.
- Formulate the smooth limit surface of any subdivision scheme as a single type of finite elements.
- Allow users to directly manipulate the limit subdivision surface via physics-based force tools.

Dynamic Subdivision Surface

Consider the control vertex positions as time-varying variables, the velocity of the surface model can be expressed as:

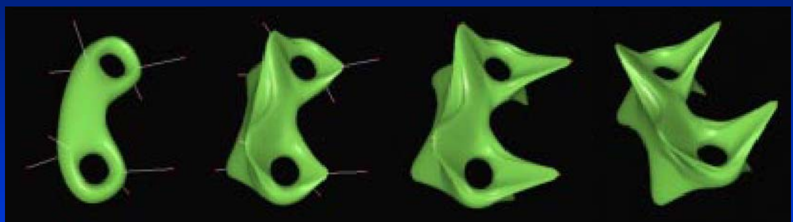
$$\dot{s}(x,p) = J(x)\dot{p}$$

The motion equation of the dynamic subdivision surface is guided by the Lagrange mechanics of law:

$$M\ddot{p} + D\dot{p} + Kp = f_p$$

M: mass matrix
D: damping matrix
K: stiffness matrix
f: generalized force

Dynamic Subdivision Surface



Qin et al. IEEE TVCG'98

State-Of-Art Deformable Models

- Explicit parametric model
  - Snake: Kass, Witkin and Terzopoulos, 87.
  - D-Superquadrics: Metaxas and Terzopoulos, 92.
  - D-NURBS: Qin and Terzopoulos, 94.
  - T-Snake: McInerney and Terzopoulos 95
  - Oriented-Particles: Szeliski and Terzopoulos, 95
  - D-Subdivision: Qin and Mandal. 98.
  - …
- Implicit level-set model
  - Osher and Sethian, 88.
  - Malladi, Sethian and Vemuri, 95.
  - Caselles, Kimmel, and Sapiro, 95.
  - Zhao, Osher and Fedkiw 2001.
  - ...

Oriented Particles

Particle Physics

- Point masses
- Long range attraction forces
- Short range repulsion forces
- Pairwise potential energy

Energy formulation

- Potential energy is a function of distance.

Lennard-Jones potential energy

$$\phi_{LJ}(r) = \phi_R(r) + \phi_A(r)$$

$$\phi_{LJ}(r) = \frac{B}{r^n} - \frac{A}{r^m}$$

– long-range potential
– short-range potential

Particle interaction

$$f_{ij}(r) = -\nabla_{\mathbf{r}_{ij}} \phi_{LJ}(|\mathbf{r}_{ij}|)$$

where $\mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i$

Energy formulation

- Potential energy of the system:
  – Summation of all pairs
- Inter-particle forces:
  – Summation of all pairwise forces

Lennard-Jone type function

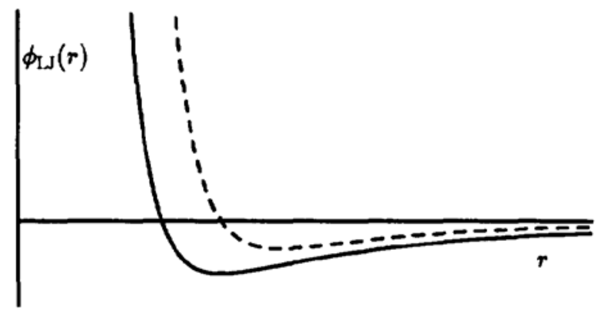


Figure 1: Lennard-Jones type function, $\phi_{LJ}(r) = B/r^n - A/r^m$. The solid line shows the potential function $\phi_{LJ}(r)$, and the dashed line shows the force function $f(r) = -\frac{d}{dr}\phi_{LJ}(r)$.

Oriented Particles

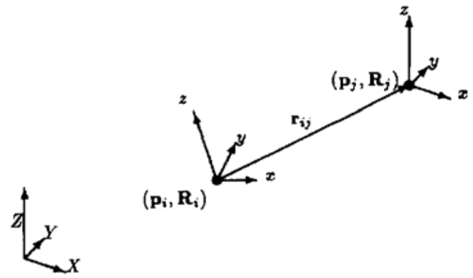


Figure 2: Global and local coordinate frames. The global interparticle distance $\mathbf{r}_{ij}$ is computed from the global coordinates $\mathbf{p}_i$ and $\mathbf{p}_j$ of particles i and j. The local distance $\mathbf{d}_{ij}$ is computed from $\mathbf{r}_{ij}$ and the rotation matrix $\mathbf{R}_i$.

Oriented Particles

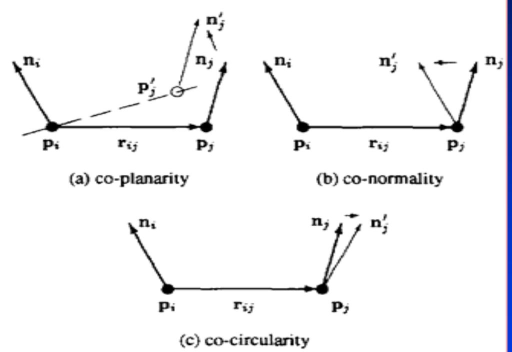


Figure 3: The three oriented particle interaction potentials. The open circles and thin arrows indicate a possible new position or orientation for the second particle which would lead to a null potential.

Oriented Particles

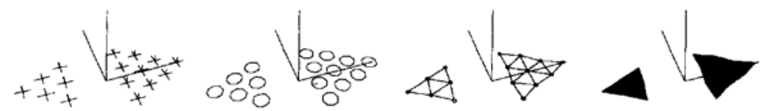


Figure 4: Rendering techniques for particle-based surfaces: (a) axes, (b) discs, (c) wireframe triangulation (d) flat-shaded triangulation.

Oriented Particles

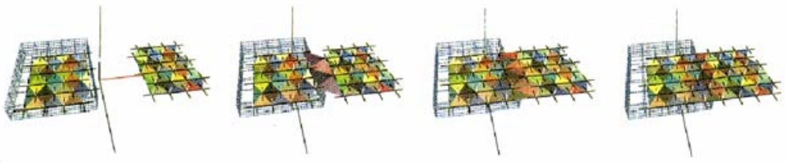


Figure 5: Welding two surfaces together. The two surfaces are brought together through interactive user manipulation, and join to become one seamless surface.

Oriented Particles

Figure 6: Cutting a surface into two. The movement of the knife edge pushes the particles in the two surfaces apart.

Oriented Particles

Figure 7: Putting a crease into a surface. The center row of particles is turned into unoriented particles which ignore smoothness forces.

Oriented Particles

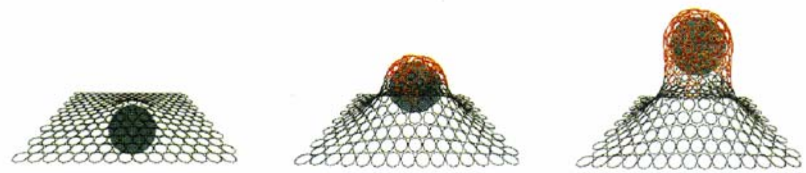


Figure 8: Particle creation during stretching. As the ball pushes up through the sheet, new particles are created in the gaps between pairs of particles.

Oriented Particles

Figure 9: Surface interpolation through a collection of 3-D points. The surface extends outward from the seed points until it fills in the gaps and forms a complete surface.

Oriented Particles

Figure 10: Interpolation of an open surface through a collection of 3-D points. Particles are added between control points until all gaps less than a specified size are filled in. Increasing the range would allow the sparse areas of the cheek and neck to filled in.

Oriented Particles

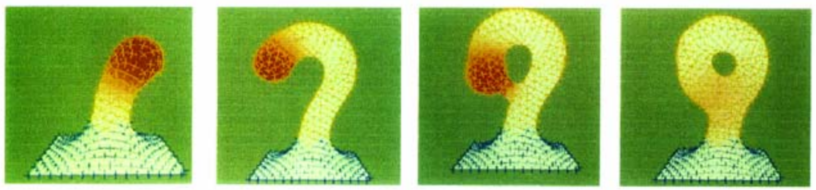


Figure 11: Forming a complex object. The initial surface is deformed upwards and then looped around. The new topology (a handle) is created automatically.

Oriented Particles

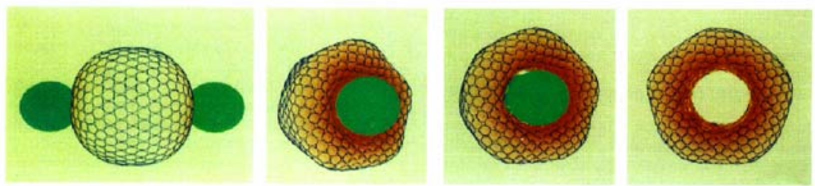


Figure 12: Deformation from sphere to torus using two spherical shaping tools. The final view is from the side, showing the toroidal shape.

Oriented Particles

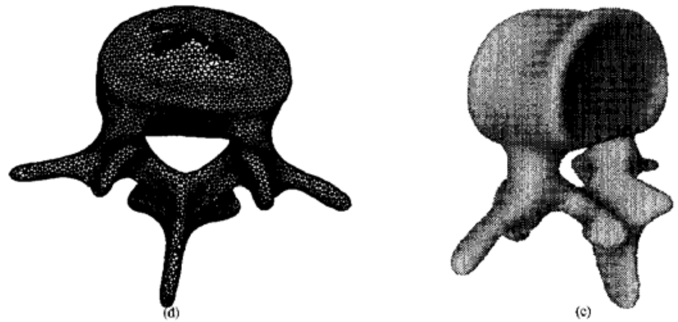


Figure 4: 3-D Reconstruction of a vertebra from 120 × 128 × 52 CT volume data: (a) xy slice, (b) xz slice, (c) yz slice, (d) reconstructed 3-D surface model with triangulated particles, (e) shaded surface.

Oriented Particles

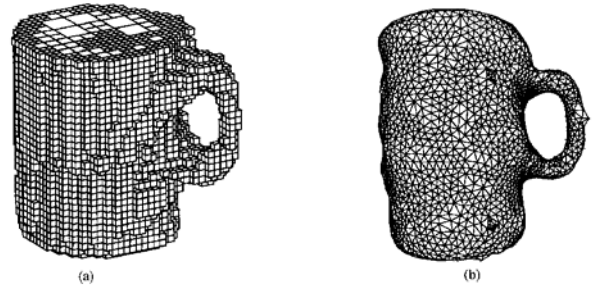


Figure 5: Reconstruction of a surface model of a cup from silhouettes: (a) cup bounding volume represented as an octree, (b) triangulated surface of reconstructed model.

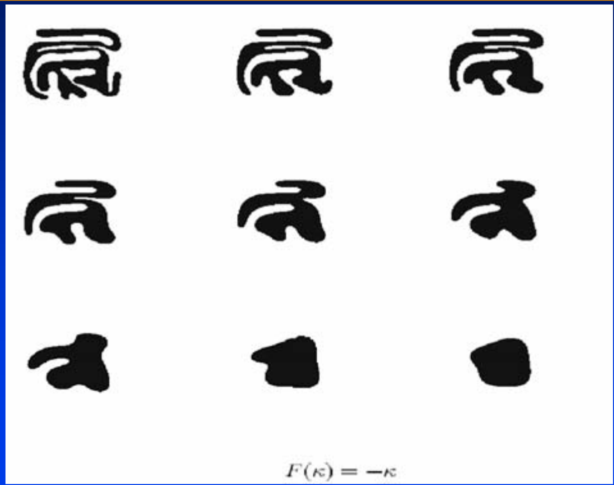
State-Of-Art Deformable Models

- Explicit parametric model
  - Snake: Kass, Witkin and Terzopoulos, 87.
  - D-Superquadrics: Metaxas and Terzopoulos, 92.
  - D-NURBS: Qin and Terzopoulos, 94.
  - T-Snake: McInerney and Terzopoulos 95
  - Oriented-Particles: Szeliski and Terzopoulos, 95
  - D-Subdivision: Qin and Mandal. 98.
  - …
- Implicit level-set model
  - Osher and Sethian, 88.
  - Malladi, Sethian and Vemuri, 95.
  - Caselles, Kimmel, and Sapiro, 95.
  - Zhao, Osher and Fedkiw 2001.
  - …

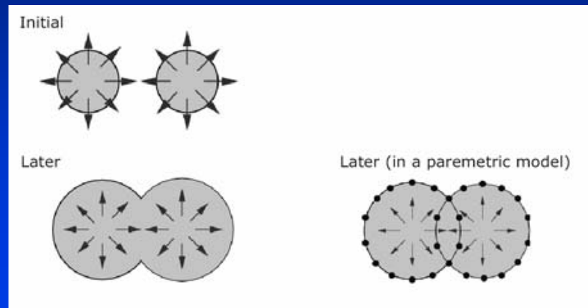
Level Set Models

- Introduced by Osher and Sethian in 1988.
- Level set models are deformable implicit surfaces that have a volumetric representation.
- The deformation of the surface is controlled by a speed function in the partial differential equation (PDE).
- Topology flexible.
- Very popular in recent years.

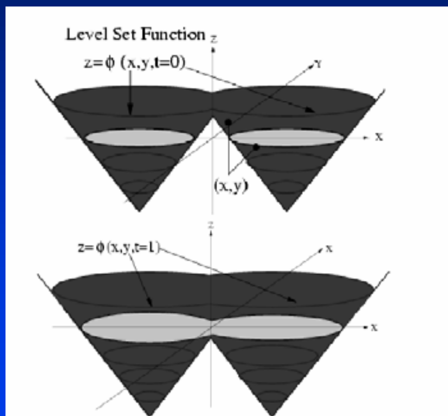
Curve evolution



Propagation of two fronts



Add an extra dimension



The level set function:

$z = \Phi(x,y,t)$

Contour at time *t*:

$0 = \Phi(x,y,t)$

Level Set Models

A deformable surface *S(t)*, is implicitly represented as an iso-surface of a time-varying scalar function, $\phi(x(t))$ embedded in 3D, i.e.

$$S(t) = \{x(t) \mid \phi(x(t), t) = k\} \qquad (1)$$

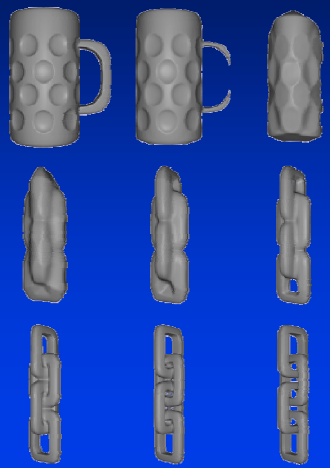
Differentiating both sides of Eq. (1)

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \bullet \frac{dx}{dt} \qquad (2)$$

Define the speed function F as: $F(x, n, \phi, \ldots) = n \bullet \frac{dx}{dt}$

We have, $$\frac{\partial \phi}{\partial t} = |\nabla \phi| F(x, n, \phi, \ldots) \qquad (3)$$

Level Set Models



Breen et al. IEEE TVCG'99

Outline

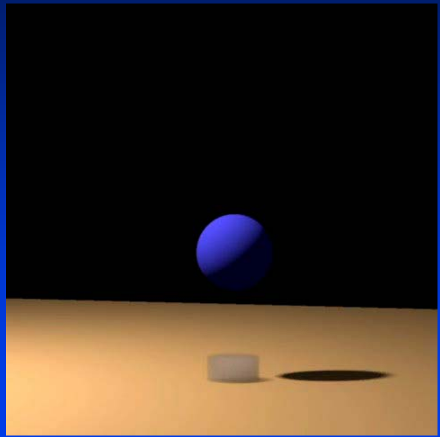
- Overview
  - Why deformable model?
  - What is a deformable model?
  - State-of-art deformable models.
- Deformable Surface
  - Geometry Representation
  - Evolution Low
  - Topology
- State-of-art deformable models
- Applications

Applications

- Computer Animation
- Virtual Environments
- Rapid Prototyping
- Haptic Rendering
- Computer Game Dynamics
- Medical Simulation and Analysis

Physically-Based Modeling of Natural Phenomena (Fire, Smoke & Water)

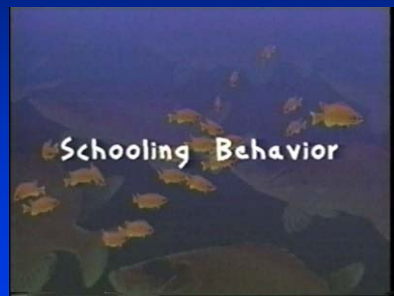
Smoke



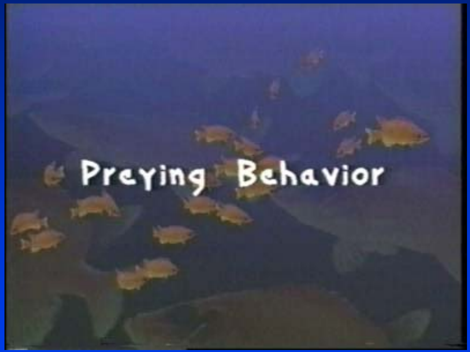
Fire

Cloth modeling

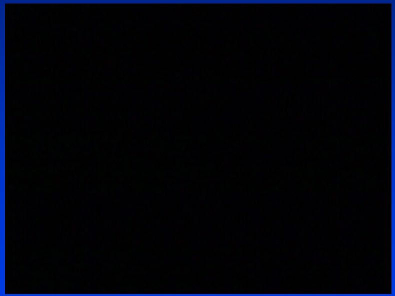
Artificial Fish



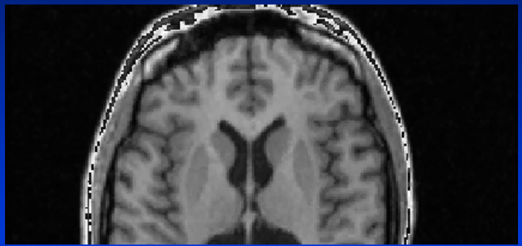
Artificial Fish



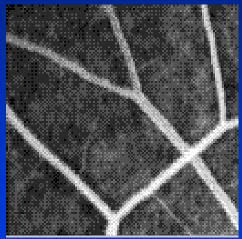
Artificial Fish



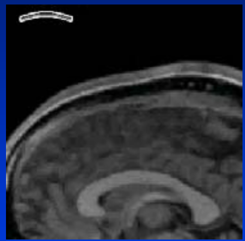
Medical Image Segmentation



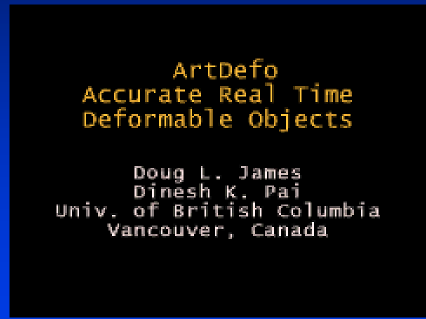
Medical Image Segmentation



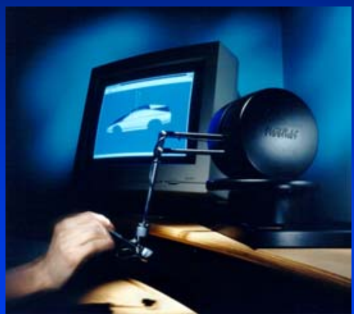
Medical Image Segmentation



Interactive Computer Animation



Force-feedback Rendering



Haptic Interfaces

- hap·tic ('hap-tik)
  adj.
  Of or relating to the sense of touch; tactile.

Glove-based Interaction

Surgical Simulation

Virtual Brush