Deformable Models & Applications (Part I)

Ye Duan
Department of Computer Science
University of Missouri at Columbia
December 21, 2004

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.

Outline

• Overview
• Deformable Surface
  – Geometry Representation
  – Evolution Low
  – Topology
• State-of-art deformable models
• Applications
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).

Outline

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

Deformable Model

- Introduced by Terzopoulos et al. in late 80s.

- Consists of deformable curves, deformable surfaces, and deformable solids.
**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
  \[ a_0 x^2 + a_1 y^2 + a_2 z^2 + a_3 xy + a_4 xz + a_5 yz + a_6 x^2 + a_7 y^2 + a_8 z^2 + a_9 \]

- 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 (Beziers)

\[ c(u) = \sum_{i=0}^{3} p_i B_i(u) \]
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 \( v \in [0,1] \)

\[ p_i = \sum_{j=0}^{3} p_{i,j} B_j(v) \]

From curve to surface

- Then we have

\[ s(u, v) = \sum_{i=0}^{3} \left( \sum_{j=0}^{3} p_{i,j} B_j(v) \right) B_i(u) = \sum_{i=0}^{3} \sum_{j=0}^{3} p_{i,j} B_i(u) B_j(v) \]
Beziers Surface

B-Splines Surface

- B-Spline curves
  \[ c(u) = \sum_{i=0}^{n} p_i B_{i,k}(u) \]

- Tensor product B-splines
  \[ s(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} p_{ij} B_{i,k}(u) B_{j,l}(v) \]

  where \( u \in [0,1] \), and \( v \in [0,1] \)

- 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

\[
s(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{n} b_{i,j} B_i(u) B_j(v) / \sum_{i=0}^{n} \sum_{j=0}^{n} b_{i,j} B_i(u) B_j(v)
\]

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

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

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

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.

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:
\[
P(v) = \int_{s_0}^{s_f} p(v(s)) \, ds
\]

\[
p(u) = -c |\nabla[G_u * f(u)]|
\]

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:

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

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 \( \gamma(s) \):

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

Discretized Lagrange dynamic equation

Can be solved using finite difference or finite element:

\[
M\ddot{u} + D\dot{u} + Ku = f
\]
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.

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:
$$p(u|I) = \frac{p(I|u)p(u)}{p(I)}$$
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
$$p(I|u) = \exp(-p(I))/Z_I$$

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
  – Topology
• State-of-art deformable models
• Applications

State-Of-Art Deformable Models

• Explicit parametric model
  – 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.
  – …
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.
  - ...

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**

\[ E_{\text{snake}} = \int_0^1 E_{\text{snake}}(\mathbf{v}(s)) \, ds = \int_0^1 E_{\text{int}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s)) \, ds \]

where

- \( E_{\text{int}} \): internal energy
- \( E_{\text{image}} \): image energy
- \( E_{\text{con}} \): constraint energy

Energy Formulation

**Internal spline energy (membrane + thin-plate)**

\[ E_{\text{int}}(\mathbf{v}(s)) = \frac{1}{2} (\alpha(s)|x(s)|^2 + \beta(s)|y(s)|^2) \]

**Image energy (line + edge + termination)**

\[ E_{\text{image}} = w_{\text{line}} E_{\text{line}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{term}} E_{\text{term}} \]

**Line functional**

\[ E_{\text{line}} = I(x, y) \]

**Edge functional**

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

where

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

Energy Formulation

**Termination functional**

\[ E_{\text{term}} = \frac{\partial \theta}{\partial n_1} + \frac{\partial^2 C}{\partial n^2} \]

\[ = C_{yy} C_{x}^2 - 2C_{xy} C_x C_y + C_{xx} C_y^2 \]

\[ (C_x^2 + C_y^2)^{3/2} \]

where

\[ C(x, y) = C_0(x, y) + I(x, y) \]

\[ \tan(\theta) = \frac{C_y}{C_x} \]

\[ n = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \]

Feature detection need domain knowledge
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 $-k(x_1-x_2)^2$ 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 neurite/glia. Neurites are shown in black, synapses and the venule are in white.

Edge detection

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

Scale space

Subjective contour detection

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

Fig. 6: Above left: Dynamic subjective contour illusion. Sequence of left to right translation. Above Right: Virtual translation to edges and intersections. At the moving horizontal line slides to the right, the seems breaks until it falls off the line. Bringing the line close enough subjects the subject movement.

Stereo vision

Fig. 7: Bottom: Stereoscan of a torn piece of paper. Below: Surface reconstruction from the outline of the paper matched using stereo vision. 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 2D model.

Motion tracking

Fig. 8: Selected frames from a 2-second video sequence showing motion tracking. After being initialized to the speaker's lips in the first frame, the systems 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.
  - ...

Topology Adaptive 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.

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

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:

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

\[M: \text{mass matrix}\]
\[D: \text{damping matrix}\]
\[K: \text{stiffness matrix}\]
\[f: \text{generalized force}\]

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
  \[ f_{ij}(r) = -\nabla_{r_{ij}} \phi_{LJ}(|r_{ij}|) \]
  where \( r_{ij} = p_j - p_i \)
Lennard-Jones 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 $r_{ij}$ is computed from the global coordinates $p_i$ and $p_j$ of particles $i$ and $j$. The local distance $d_{ij}$ is computed from $r_{ij}$ and the rotation matrix $R_i$.

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.

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 joints to become one seamless surface.

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

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

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

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. Increasing the range would allow the sparse areas of the cheek and neck to fill in.

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

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

**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.
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.

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

Differentiating both sides of Eq. (1)

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{dx}{dt}$$  \hspace{1cm} (2)

Define the speed function $F$ as:

$$F(x, n, \phi, \ldots) = n \cdot \frac{dx}{dt}$$

We have,

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \cdot F(x, n, \phi, \ldots)$$  \hspace{1cm} (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)
Artificial Fish

Preying Behavior

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

Virtual Chinese Brush