CAP6411 Computer Vision Systems Lecture 14

Alper Yilmaz
Office: CSB 250
Email: yilmaz@cs.ucf.edu
Web: http://www.cs.ucf.edu/courses/**cap6411/cap6411**/spring2006

Paper Presentations

- Veenman, C., Reinders, M., and Backer, E. 2001. Resolving motion correspondence for densely moving points. PAMI 23, 1, 54–72.
- Shi, J. and Tomasi, C. 1994. Good features to track. In CVPR. 593–600.
- Paragios, N. and Deriche, R. 2002. Geodesic active regions and level set methods for supervised texture segmentation. IJCV 46, 3, 223–247.
- R. Vidal, Y. Ma and S. Sastry, 2003, Generalized Principal Component Analysis (GPCA). CVPR, 621--628
- Kentaro Toyama and Andrew Blake 2001. Probabilistic Tracking in a Metric Space. ICCV (Marr Prize)

1. Functional

$$E\left(\partial\Gamma(s)\right)=\iint_R h\left(x(s),y(s)\right)dxdy$$

$$h\left(x(s),y(s)\right)=\int_{-m}^{m}\int_{-m}^{m}\log P_R\left(I(x+f(s),y+g(s))\right)1_R^{\partial\Gamma}\,dxdy$$

$$R=\{\text{object},\text{background}\}$$

2. Green's Theorem

- For a planar region R=object $\cup$ background (P(x,y),Q(x,y)) is any vector field with continuous first order derivatives, then

$$h\left(x,y\right)=\frac{\partial Q}{\partial x}-\frac{\partial P}{\partial y}$$

example $\begin{cases} Q(x,y)=\frac{1}{2}\int_0^x h(t,y)dt \\ P(x,y)=-\frac{1}{2}\int_0^y h(x,t)dt \end{cases}$

$$\iint_R h\left(x,y\right)dxdy=\iint_R\left(\frac{\partial Q}{\partial x}-\frac{\partial P}{\partial y}\right)dxdy$$

2.1. Derivation

$$\iint_R h(x,y)dxdy = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right)dxdy$$

$$= \iint_R \frac{\partial Q}{\partial x}dxdy - \iint_R \frac{\partial P}{\partial y}dydx$$

$$= \int_{\partial\Gamma} Qdy - \int_{\partial\Gamma} Pdx$$

$$= \int_0^l P\dot{x}ds + \int_0^l Q\dot{y}ds$$

change of variables

$$\dot{x} = \frac{\partial x}{\partial s} \qquad \dot{y} = \frac{\partial y}{\partial s}$$

3. Minimization

$$E(\partial\Gamma(s)) = \iint_R h(x,y)dxdy$$

$$= \int_0^l (P(x,y)\dot{x} + Q(x,y)\dot{y})ds$$

$$= \int_0^l L(x,\dot{x},y,\dot{y})ds$$

Minimizing in the *steepest descent* results in the following Euler-Lagrange equations.

$$\frac{\delta E}{\delta x} = \frac{\partial L}{\partial x} - \frac{d}{ds}\frac{\partial L}{\partial \dot{x}} \qquad \frac{\delta E}{\delta y} = \frac{\partial L}{\partial y} - \frac{d}{ds}\frac{\partial L}{\partial \dot{y}}$$

3.1. Euler-Lagrange Equations

$$\frac{\delta E}{\delta x}=\frac{\partial L}{\partial x}-\frac{d}{ds}\frac{\partial L}{\partial \dot{x}} \qquad \frac{\delta E}{\delta y}=\frac{\partial L}{\partial y}-\frac{d}{ds}\frac{\partial L}{\partial \dot{y}}$$

$$\frac{\partial L}{\partial x}=\frac{\partial Q(x,y)}{\partial x}\dot{x}+\frac{\partial P(x,y)}{\partial x}\dot{y}$$

$$\frac{\partial L}{\partial y}=\frac{\partial Q(x,y)}{\partial y}\dot{x}+\frac{\partial P(x,y)}{\partial y}\dot{y}$$

$$\frac{\partial L}{\partial \dot{x}}=Q(x,y) \qquad \frac{\partial L}{\partial \dot{y}}=P(x,y)$$

3.1. Euler-Lagrange Equations

$$\frac{\delta E}{\delta x}=\frac{\partial L}{\partial x}-\frac{d}{ds}\frac{\partial L}{\partial \dot{x}} \qquad \frac{\delta E}{\delta y}=\frac{\partial L}{\partial y}-\frac{d}{ds}\frac{\partial L}{\partial \dot{y}}$$

$$\frac{d}{ds}\frac{\partial L}{\partial \dot{x}}=\frac{d}{ds}Q\big(x(s),y(s)\big)=\frac{\partial Q}{\partial x}\frac{\partial x}{\partial s}+\frac{\partial Q}{\partial y}\frac{\partial y}{\partial s}=\frac{\partial Q}{\partial x}\dot{x}+\frac{\partial Q}{\partial y}\dot{y}$$

$$\frac{dL}{dx}-\frac{d}{ds}\frac{\partial L}{\partial \dot{x}}=\left(\frac{\partial Q}{\partial x}\dot{x}+\frac{\partial P}{\partial x}\dot{y}\right)-\left(\frac{\partial Q}{\partial x}\dot{x}+\frac{\partial P}{\partial y}\dot{y}\right)=\underbrace{\left(\frac{\partial Q}{\partial x}-\frac{\partial P}{\partial y}\right)}_{h(x,y)}\dot{y}$$

$$\boxed{\frac{\delta E}{\delta x}=h(x,y)\dot{y}} \qquad \boxed{\frac{\delta E}{\delta y}=-h(x,y)\dot{x}}$$

4. The Motion Equation

$$\frac{\delta E}{\delta x} = h(x,y)\dot{y} \qquad \frac{\delta E}{\delta y} = -h(x,y)\dot{x}$$

Normal vector along the contour is $\vec{\mathbf{n}} = (\dot{y}, -\dot{x})$

Let $\vec{v} = (x,y)$ thus $\frac{\delta E}{\delta \vec{v}} = h(x,y)\vec{\mathbf{n}}$

4. The Motion Equation

$$\frac{\delta E}{\delta \vec{v}} = h(x,y)\vec{\mathbf{n}}$$

$$h(x(s), y(s)) = \int_{-m}^{m}\int_{-m}^{m} \log P_R\big(I(x+f(s), y+g(s))\big) 1_R^{\partial\Gamma}\, dxdy$$

$$\frac{\delta E}{\delta \vec{v}} = -\int_{-m}^{m}\int_{-m}^{m} \log P_{R_{obj}}\big(I(\mathbf{x})\big) 1_{obj}^{\partial\Gamma}\{\mathbf{x}\} dxdy\, \vec{\mathbf{n}}_{obj}^{\mathbf{x}} - \int_{-m}^{m}\int_{-m}^{m} \log P_{R_{bck}}\big(I(\mathbf{x})\big) 1_{bck}^{\partial\Gamma}\{\mathbf{x}\} dxdy\, \vec{\mathbf{n}}_{bck}^{\mathbf{x}}$$

Contour Representations

Level sets

1. Fluid Dynamics

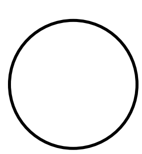
- Predict the motion of fluids
  - Flow of heat
  - Mass transfers (perspiration…), etc.
- Non-rigid transformation of particles
  - Mathematical formulation
    - Scientific knowledge?
  - Numerical implementation
    - Accuracy?

2. Representations

- **Parametric:** Lagrangian approach has problems during evolution
- **Implicit:** Eulerian approach.
  - *Marker string* methods
  - *Volume fluid* methods
  - *Level set* methods
- *Level set* approach is numerically most stable implicit representation

3. Two-dimensional Contour

- Closed form contour equation
  - $C(x,y)=0$
- Parametric contour equation
  - $C(f(s),g(s))=0$
- For instance *circle*:



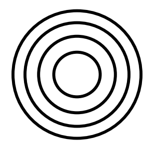
$$x^2 + y^2 - 1 = 0$$

$$\left.\begin{aligned} x &= f(s) = \cos s \\ y &= g(s) = \sin s \end{aligned}\right\} \text{parametric form}$$

3.1. Family of Contours

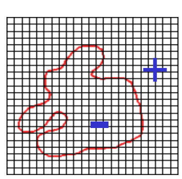
- Family parameter t is introduced
  - $C_t(x,y)=C(x,y,t)=0$
- The parametric form is
  - $C_t(f(s,t),g(s,t))=C(\mathbf{x}(s,t),t)=0$
- For instance for the circle

$$x = f(s,t) = t\cos s$$
$$y = g(s,t) = t\sin s$$



Contour Representations

- Explicit “parametric form”
- Explicit “marker-String method”
- Implicit “volume fluid method”
- $\boxed{\text{Implicit “level-set methods”}}$



implicit curve $\phi(\partial\Gamma(s,t),t)=0$

surface gradient $\phi_t = F.|\nabla\phi_t|$

curve motion $\dfrac{\partial\phi(\partial\Gamma(s,t),t)}{\partial t}=0$

update $\phi^n = \phi^{n-1} - \Delta t\,|\nabla\phi_t|\,F$

The Level Set Method

- Osher-Sethian (1987)
  - Earlier: Dervieux, Thomassett, (1979, 1980)
- Introduced in the area of fluid dynamics
- Vision and image segmentation
  - Caselles-Catte-coll-Dibos (1992)
  - Malladi-Sethian-Vermuri (1994)
- Level Set Milestones
  - Faugeras-keriven (1998) stereo reconstruction
  - Paragios-Deriche (1998), active regions and grouping
  - Chan-Vese (1999) mumford-shah variant
  - Leventon-Grimson-Faugeras-etal (2000) shape priors
  - Zhao-Fedkiew-Osher (2001) computer graphics

The Level Set Method

- Let us consider in the most general case the following form of curve propagation:

  $C(p,t) = F(\mathcal{K})\mathcal{N}$

- Addressing the problem in a higher dimension…

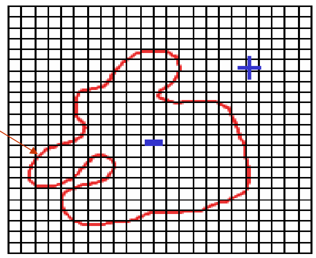
- The level set method represents the curve in the form of an implicit surface:

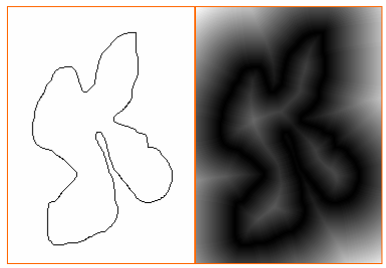
  $\varphi(x,y,t) : \mathcal{R}^2 \times [0,T) \rightarrow \mathcal{R}$

Level Set Representation

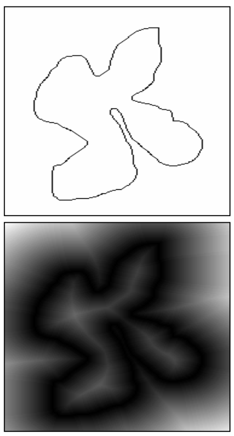
- Contour is represented in discrete grid
  - Grid values are distances from the contour
  - Contour inside is negative
  - Contour outside is positive

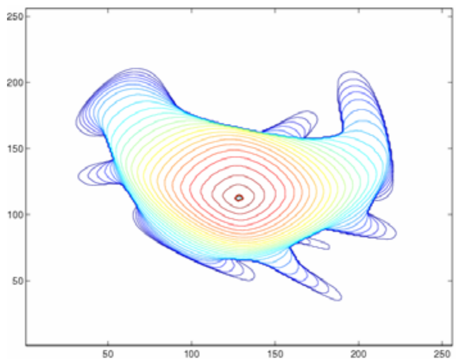
$C(\mathbf{x}(s,t),t)=0$



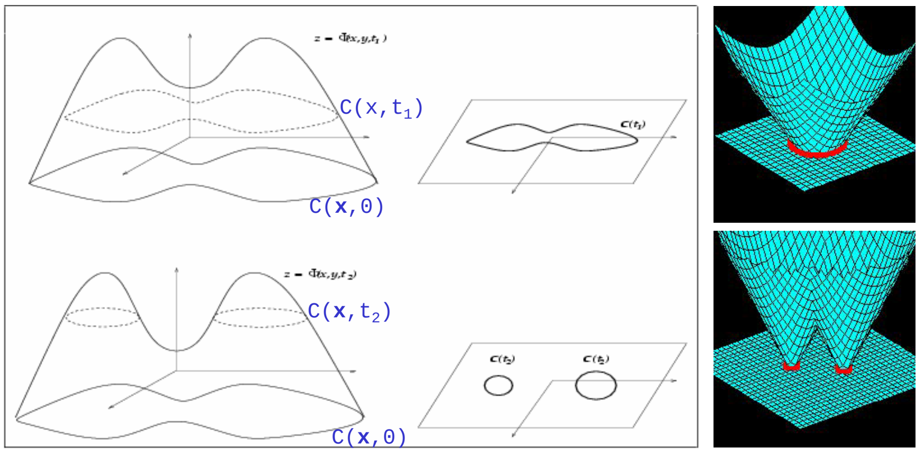


Level Set Representation and Contour Evolution





Evolving the Contour



Evolution Equations

- Evolution = Displacement in normal direction

$$C\left(\mathbf{x}(s,t),t\right)=0$$

$$\frac{\partial C\left(\mathbf{x}(s,t),t\right)}{\partial t}=\frac{\partial}{\partial t}0 \quad \text{take derivate of both sides}$$

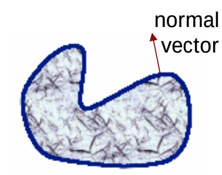
$$C_t+\nabla C\left(\mathbf{x}(s,t),t\right).\underbrace{\mathbf{x}'(t)}_{\text{distance in }\Delta t}=0$$

Evolution Equations

$$C_t + \nabla C\big(\mathbf{x}(s,t),t\big)\mathbf{x}'(t) = 0$$

Divide both sides by |∇C|

$$\frac{C_t}{|\nabla C|} + \underbrace{\frac{\nabla C\big(\mathbf{x}(s,t),t\big)}{|\nabla C|}}_{\text{normal vector } \vec{\mathbf{n}}}\mathbf{x}'(t) = 0$$

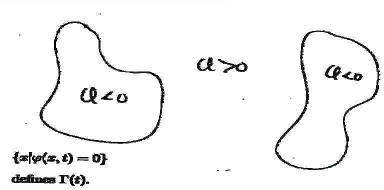


$$C_t + |\nabla C|\,\underbrace{\vec{\mathbf{n}}.\overbrace{\mathbf{x}'(t)}^{\text{speed}}}_{\text{velocity}} = 0$$

The Level Set Method

- Let us consider in the most general case the following form of curve propagation: $C(p,t) = F(\mathcal{K})\mathcal{N}$
- Addressing the problem in a higher dimension…
- The level set method represents the curve in the form of an implicit surface: $\varphi(x,y,t) : \mathcal{R}^2 \times [0,T) \to \mathcal{R}$
- That is derived from the initial contour according to the following condition:

$C(p,0) = \{(x,y) : \varphi(x,y,0) = 0\}$



Overview of the Method

- The level set flow can be re-written in the following form $\varphi_t + H(\varphi_x, \varphi_y) = 0,$
  where H is known to be the Hamiltonian.

- Determine the initial implicit function (distance transform)
  - Evolve it locally according to the level set flow
  - Recover the zero-level set iso-surface (curve position)
  - Re-initialize the implicit function and Go to step (1) of the loop
- Computationally expensive
- Open Questions: re-initialization…and numerical approximations

Implementation Details…

Level Set Method and Internal Curve Properties

- The normal to the curve/surface can be determined directly from the level set function: $\left[\mathcal{N} = -\frac{\nabla\varphi}{|\nabla\varphi|}\right]$
- The curvature can also be recovered from the implicit function, by taking the second order derivative at the arc length

$$\begin{aligned}\frac{\partial^2\varphi}{\partial s^2} &= \frac{\partial}{\partial s}(\varphi_x x_s + \varphi_y y_s) \\ &= \varphi_{xx}x_s^2 + 2\varphi_{xy}x_s y_s + \varphi_{yy}y_s^2 + \varphi_x x_{ss} + \varphi_y y_{ss} \\ &= \varphi_{xx}x_s^2 + 2\varphi_{xy}x_s y_s + \varphi_{yy}y_s^2 + \langle\nabla\varphi, C_{ss}\rangle \;=\; 0\end{aligned}$$

Level Set Method and Internal Curve Properties

- Where we observe no variation since the implicit function has constant "zero" values, and given that $[C_{ss} = (x_{ss}, y_{ss}) = \mathcal{K}\mathcal{N}]$ as well as $\left[\mathcal{N} = -\frac{\nabla\varphi}{|\nabla\varphi|}\right]$ one can easily prove that:

$$\mathcal{K} = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}}$$

HOMEWORK

- That can also be extended to higher dimensions

Examples: Mean/Gaussian Curvature Flow

- Minimize the Euclidean length of a curve/surface:

$$C_t = \mathcal{K}\mathcal{N}$$

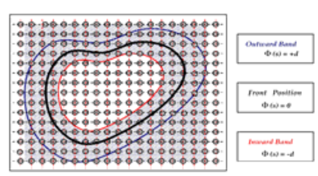
- The corresponding level set variant with a distance transform as an implicit function:

$$\phi_t = \mathcal{K}|\nabla\phi|$$

$$\mathcal{K} = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}}$$

From theory to Practice (Narrow Band)

- Central idea: we are interested on the motion of the zero-level set and not for the motion of each iso-phote (grid) of the surface
  - Extract the latest position
  - Define a band within a certain distance
  - Update the level set function
  - Check new position with respect the limits of the band
  - Update the position of the band regularly, and re-initialize the implicit function



- Significant decrease on the computational complexity, in particular when implemented efficiently and can account for any type of motion flows

Handling the Distance Function

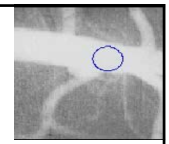
- The distance function has to be frequently re-initialized…
  - Extraction of the curve position & re-initialization:
    - Using the marching cubes one can recover the current position of the curve, set it to zero and then re-initialize the implicit function: the Borgefors approach, the Fast Marching method, explicit estimation of the distance for all image pixels…
  - Preserving the curve position and refinement of the existing function (Susman-smereka-osher:94)

$$\frac{d}{d\tau}\phi_m = \operatorname{sgn}\left(\phi_m^0\right)\left(1 - |\nabla\phi_m|\right)$$

  - Modification on the level set flow such that the distance transform property is preserved (gomes-faugeras:00)
    - Extend the speed of the zero level set to all iso-photes, rather complicated approach with limited added value?

Level Sets in imaging and vision…

Emigration from Fluid Dynamics to Vision



- (Caselles-Cate-Coll-Dibos:93,Malladi-Sethian-Vemuri:94) have proposed geometric flows to boundary extraction

$$\phi_t(;) = g(;)\,(F_A(;) + F_I(;))\,|\nabla\phi|$$

- Where g(;) is a function that accounts for strong image gradients

$$g(;) = \frac{1}{1 + |\nabla G_s * I(;)|}$$

- And the other terms are application specific…that either expand or shrink constantly the initial curve

- Distance transforms have been used as embedding functions

Geodesic Active Regions