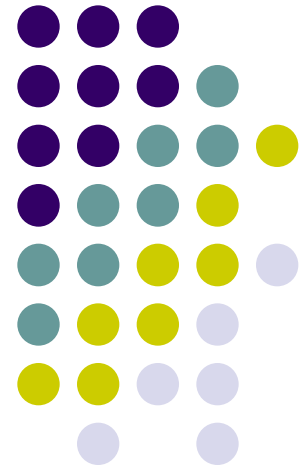


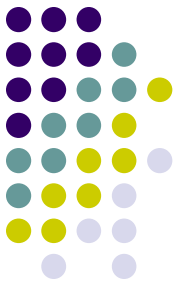
Level Set (I)

Hongxin Zhang, Wei Chen

2009-05-21

State Key Lab of CAD&CG
Zhejiang University

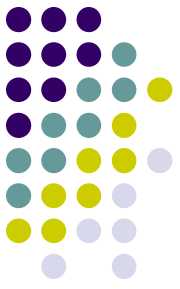




outline

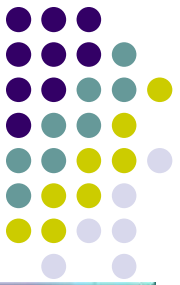
- 数学基础
- Level Set方法
- Level Set方法数值解法
- Level Set在图像图形中的应用

survey



- Level set 与 Poisson方程的比较
 - 一般的讲，level set的应用和效果都要更好！
 - 但是level set由于需要较多数学，所以工程上不被普遍接受。
 - Poisson方程是线性的，而level set方程一般都是非线性的，需要稳定的算法求解。

[Http://cermics.enpc.fr/~paragios/book/book.html](http://cermics.enpc.fr/~paragios/book/book.html)



Nikos Paragios

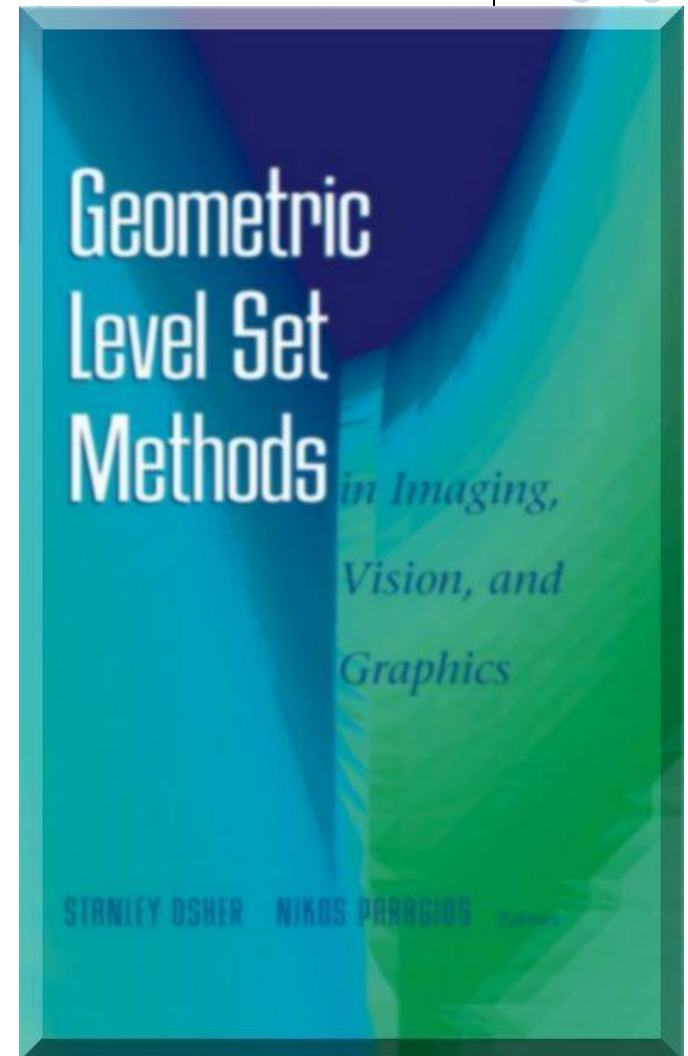
<http://cermics.enpc.fr/~paragios>

Atlantis Research Group
Ecole Nationale des Ponts et Chaussees
Paris, France

Stanley Osher

<http://math.ucla.edu/~sjo>

Department of Mathematics
University of California, Los Angeles
USA





目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 方法概念
- 数值解法
 - 向上差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



数学基础—曲线的微分几何

- 映射 $C(p): [a, b] \text{ in } \mathbf{R} \rightarrow \mathbf{R}^2$ 定义了一个平面的曲线， p 是参数，对每一个 $p_0 \text{ in } [a, b]$ ，我们得到曲线上的一点

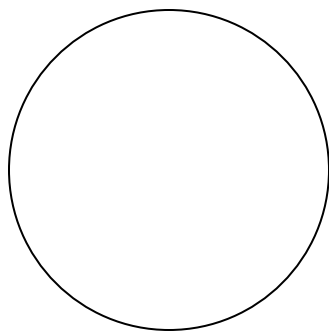
$$C(p_0) = [x(p_0), y(p_0)]$$

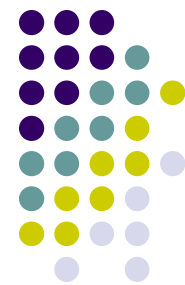
- 正则曲线：如果 $C'(p) = [x'(p), y'(p)] \neq 0$

- 例

$$C(t) = [\cos(t), \sin(t)], \quad t \text{ in } [0, 2\pi]$$

圆





数学基础—曲线的微分几何

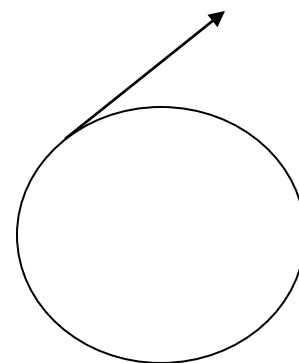
- 曲线的切线

$$C'(p) = [x'(p), y'(p)] = \frac{\partial C}{\partial p}$$

- 弧长参数

- 如果曲线的参数满足 $\left\| \frac{\partial C}{\partial p} \right\| = 1$

p 表示曲线上以某一点为标准的弧长





数学基础—曲线的微分几何

- 弧长

$$L(p_0, p_1) = \int_{p_0}^{p_1} [(x'(p))^2 + (y'(p))^2]^{1/2} dp$$

- 对弧长参数

$$\langle C'(p), C'(p) \rangle = 1$$

求导

$$\langle C'(p), C''(p) \rangle = 0$$



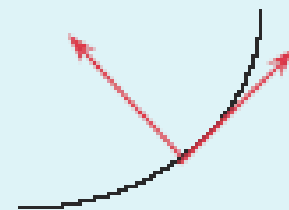
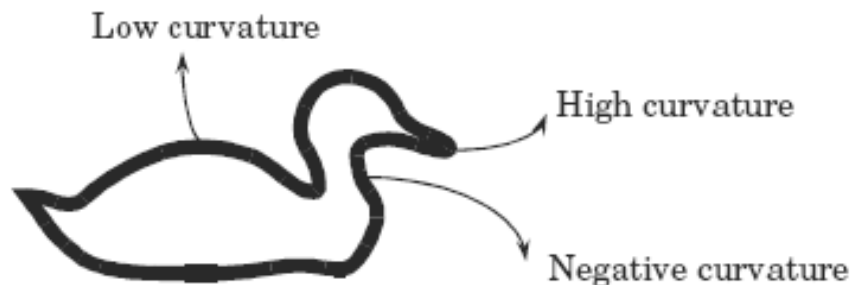
数学基础—曲线的微分几何

- 曲率

$$\kappa = \|C''(p)\|$$

- 假设 T 表示切线, N 表示法线, 则

$$\frac{dC}{dp} = T \quad \frac{d^2C}{dp^2} = \kappa N$$



$$\left\| \frac{\partial C(s)}{\partial s} \right\| = 1$$

$$\langle C_s, C_{ss} \rangle = 0$$

$$C_s \perp C_{ss}$$

$$\kappa := \|C_{ss}\|$$



数学基础—曲线的微分几何

- **Frenet** 公式

$$\frac{dT}{dp} = \kappa N$$

$$\frac{dN}{dp} = -\kappa T$$



数学基础—曲线的微分几何

- 曲率的其他定义

- 假设 θ 为切线T与x轴之间的夹角，则

$$\kappa = \frac{d\theta}{ds}$$

- 隐式曲线的曲率

$$C \equiv \{(x, y) : u(x, y) = 0\}$$
$$\kappa = \frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}$$



数学基础—曲线的微分几何

- 曲率的性质
 - 旋转，平移不变
 - 缩放要变化





数学基础—曲线的微分几何

- 隐式曲线的法向量 $u(x, y) = 0$

$$N = +(-) \frac{\nabla u}{\|\nabla u\|}$$

- 因为**T**和**N**互相垂直，所以平面上任何曲线都可以用曲线上任何一点的**T**和**N**的线性组合来表示

$$\frac{\partial C}{\partial t} = \alpha T + \beta N$$

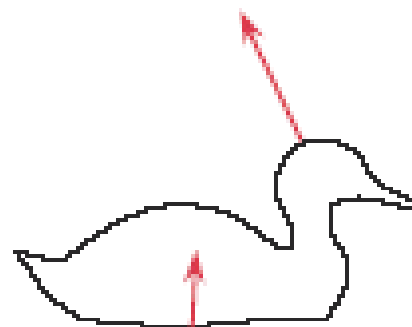




数学基础—曲线的微分几何

- 如果只考虑几何形状的变化，那其变化只跟法线方向的变化有关系，则有

$$\frac{\partial C}{\partial t} = \beta N$$

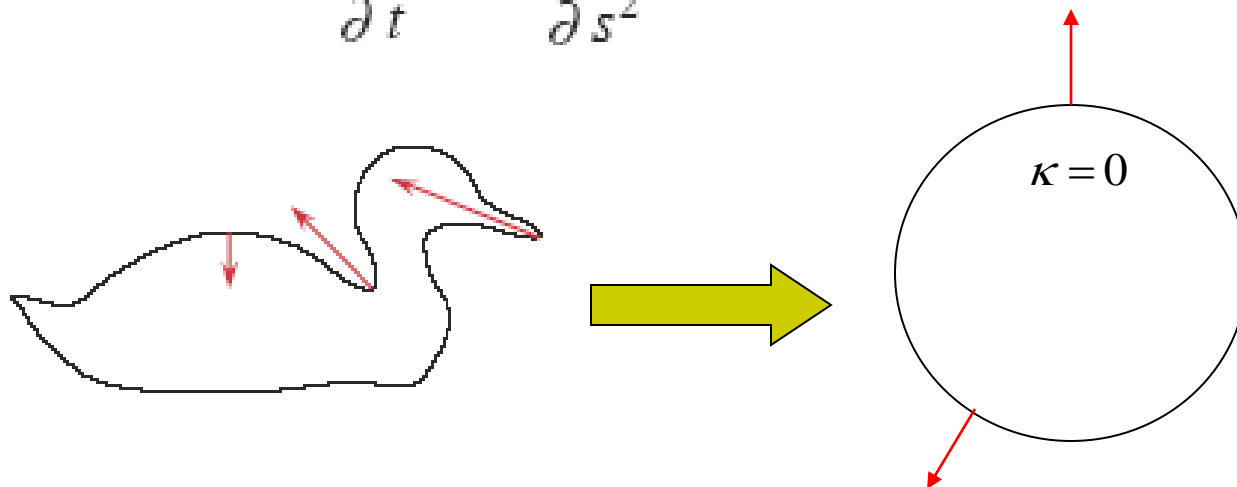




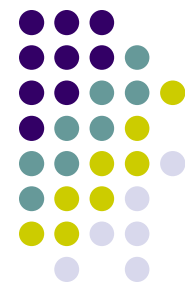
数学基础—曲线的微分几何

- 例：沿着曲率变化最大方向的曲线变形

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial s^2} = \kappa \bar{N}$$



最后变化为曲率都为常数的曲线停止，即圆



数学基础—曲面的微分几何

- 平均曲率和高斯曲率
 - 每个正则曲面都有两个主曲率。
 - 这两个的平均值就是 **平均曲率**，两个的积是 **高斯曲率**。



目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 方法概念
- 数值解法
 - 向上差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



数学基础—数学形态学

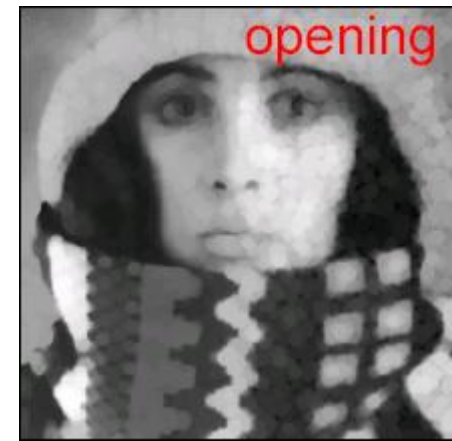
- 是一个经典的基于几何的理论
- 广泛应用于图像处理



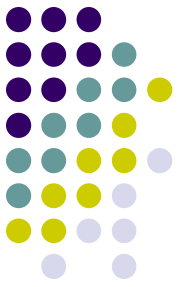
形态算子

- 一组空间滤波操作
- 用于改变二值区域的形状
 - 腐蚀：减少物体边界的像素数
 - 膨胀：增加物体边界的像素数
 - 复合方法
 - 开：腐蚀，然后膨胀
 - 闭：膨胀，然后腐蚀

Original Image



膨胀与腐蚀 (Dilation, Erosion)



- 数学形态学里面最重要的操作
- 腐蚀将图像的尺寸减少
- 膨胀增加图像的尺寸
- 可以用来消除图像上小的亮斑噪声和不规则的边



腐蚀（续）

- 定义：物体的颜色是白，背景是黑
- 定义腐蚀模板为
 - 1 1 1
 - 1 1 1
 - 1 1 1
- 将模板与图像进行加操作
- 如果有，则结果为1，否则为0



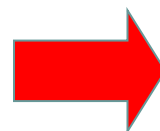
腐蚀（续）

- 模板的效果相当于去掉物体边界处的单个象素
- 4种情况：
 - 当前处理象素为1，邻域象素为1 \rightarrow 1
 - 当前处理象素为0，邻域象素为1 \rightarrow 0
 - 当前处理象素为0，邻域象素为1、0的混合 \rightarrow 0
 - 当前处理象素为1，邻域象素为1、0的混合 \rightarrow 1

腐蚀（续）



原始图像



腐蚀后的图像



膨胀

- 膨胀是腐蚀的逆操作
- 模板文件是
 - 0 0 0
 - 0 0 0
 - 0 0 0
- 其效果相当于在物体的边界添加单个像素



膨胀（续）

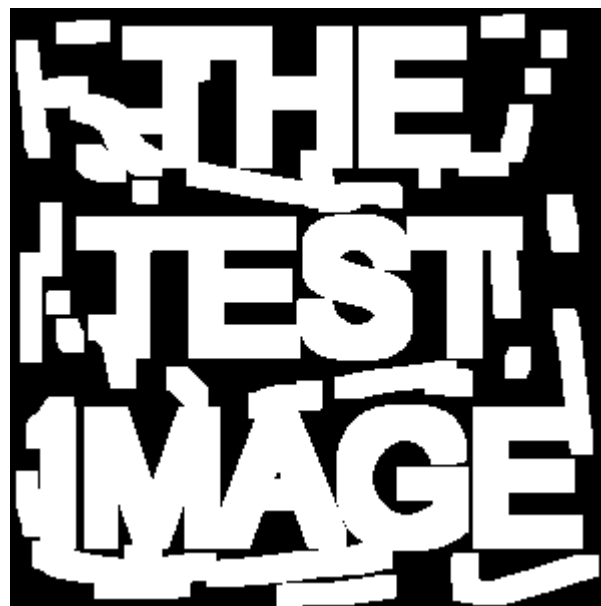
- 4种情况
 - 当前处理象素为0，邻域象素为0—》0
 - 当前处理象素为1，邻域象素为1—》1
 - 当前处理象素为1，邻域象素为1、0的混合—》1
 - 当前处理象素为0，邻域象素为1、0的混合—》1
- 逻辑操作算子是Or



膨胀（续）



原始图像



膨胀膨胀图像图像



开操作

- 开操作相当于先做腐蚀操作，再做膨胀操作
- 效果相当于去掉单个像素，但是保留原来的形状何尺寸。



原始图像

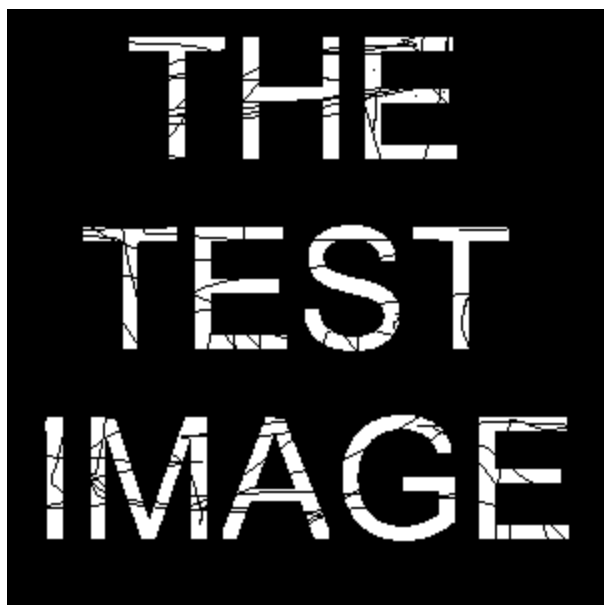


腐蚀两次，然后膨胀两次（开操作）



闭操作

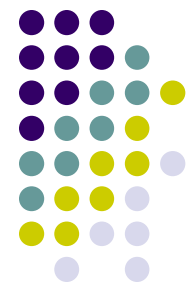
- 闭操作是开操作的相互操作
- 先膨胀，然后腐蚀
- 它可以用来填补一些小洞



原始图像

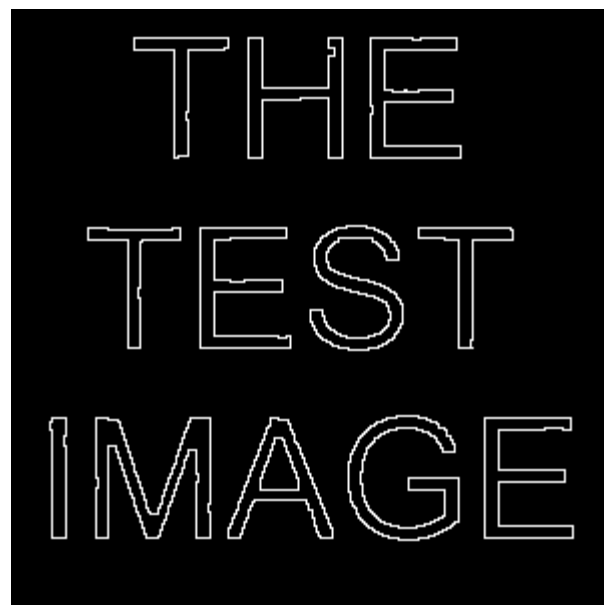


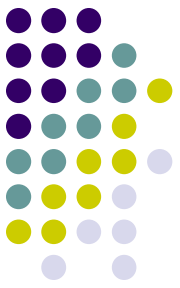
闭操作结果



轮廓抽取

- 先做腐蚀操作，再将腐蚀结果图像减去原始图像





数学基础—数学形态学

$$A \oplus B := \{a + b, a \in A, b \in B\} = \bigcup_{b \in B} A_b$$

$$A \ominus B := (A^c \oplus B)^c = \bigcap_{b \in B} A_b$$

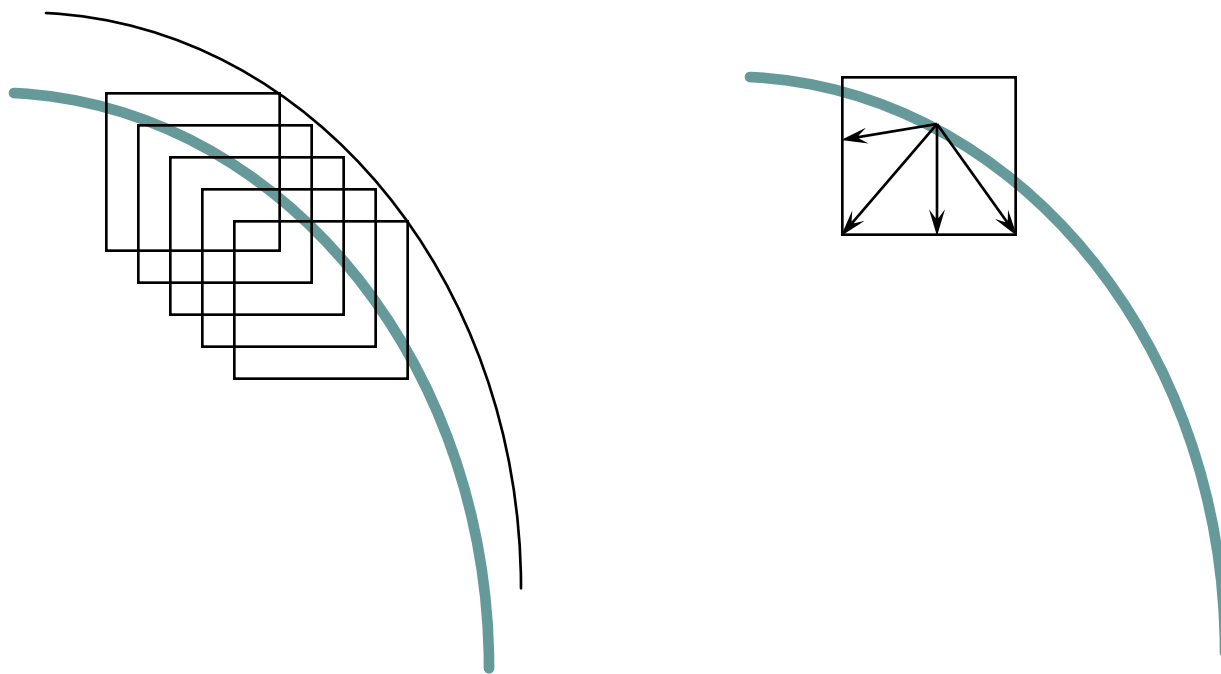
$$A \circ B := (A \ominus B) \oplus B = \bigcup_{\{y: B_y \subseteq A\}} A_y$$

$$A \bullet B := (A \oplus B) \ominus B$$



数学基础—形态学与曲线演化

- $$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$
$$A \oplus (rB) = A \oplus r_1B \oplus r_2B \oplus \dots$$





数学基础—形态学与曲线演化

- 法线速度的一般表示

$$\beta = \sup_{\theta} \{ r(\theta) \cdot N \}$$

- 例子

$$\beta = N$$

$$B = \textit{disk}$$

$$\beta = \max \{ N_x, N_y \}$$

$$B = \textit{diamond}$$

$$\beta = |N_x| + |N_y|$$

$$B = \textit{square}$$



目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 方法概念
- 数值解法
 - 向上差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



数学基础—隐函数

- 隐函数(implicit function): 自变量和因变量之间的法则是由一个方程式所确定

$$F(x, y) = 0$$

- 例子 $y = y(x)$

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



数学基础—距离场函数

- 距离函数定义

$$d(x) = \min(|x - x_l|) \quad \text{for all } x_l \in \partial\Omega$$

- 距离函数的性质

$$|\nabla d| = 1$$



数学基础—距离场函数

- 带符号的距离场隐函数 $\Phi(x)$

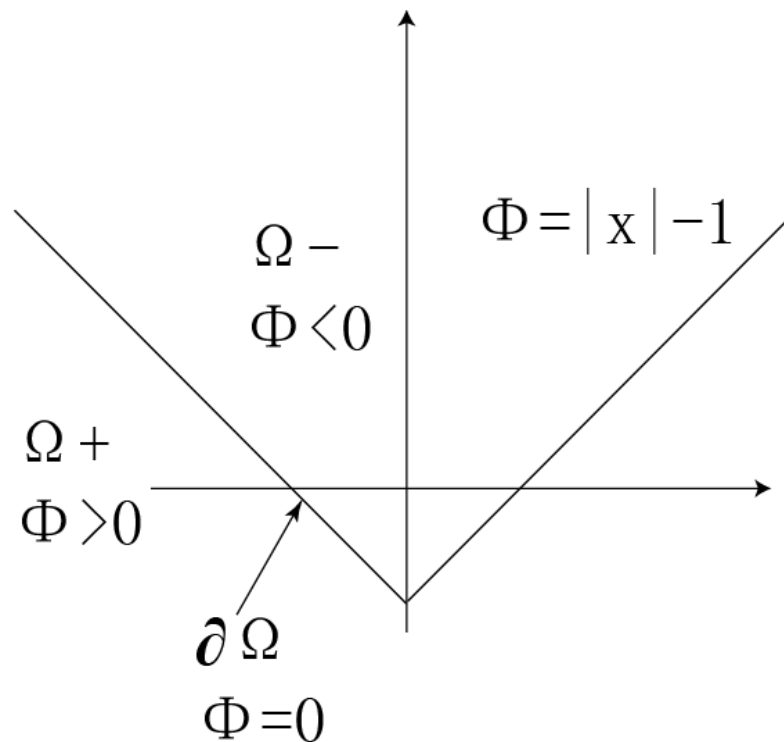
$$|\Phi(x)| = d(x)$$

$$\Phi(x) = \begin{cases} -d(x) & \text{for } x \in \Omega^- \\ 0 & \text{for } x \in \partial\Omega \\ -d(x) & \text{for } x \in \Omega^+ \end{cases}$$

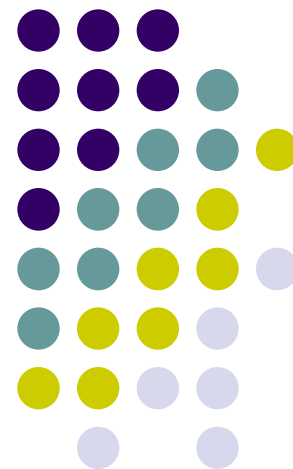


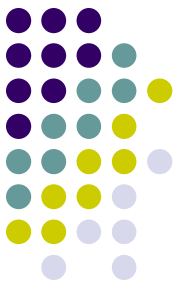
数学基础—距离场函数

- 例子 $\Phi(x) = |x| - 1$



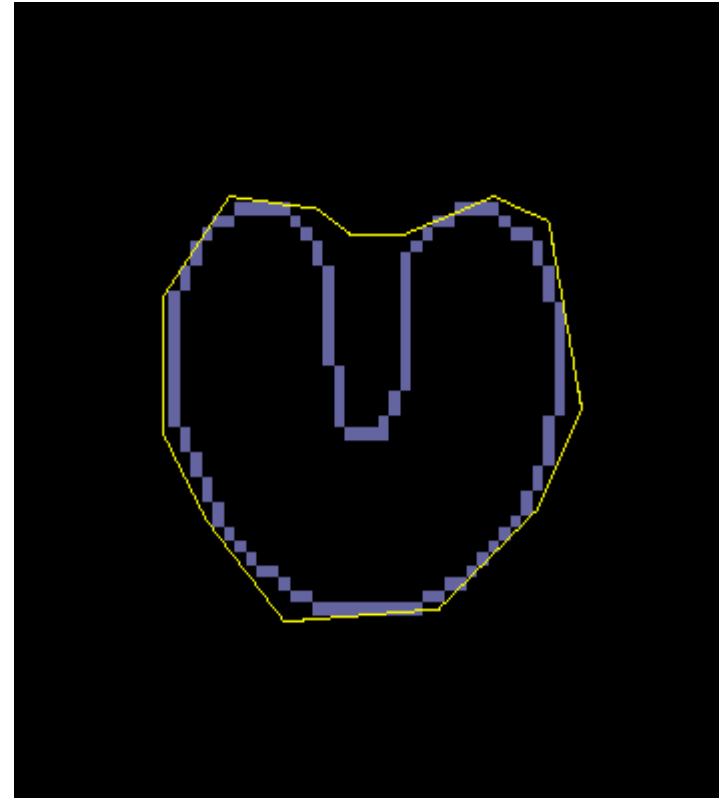
动态可变形模型

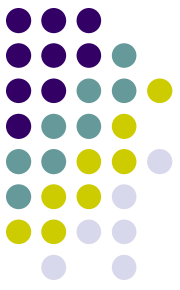




What is Snake?

- Active contour model; parametric model
- Result from Kass, Witkin, and Terzopoulos, 1987
- Energy minimizing formulation
- Depends on its shape and location within image





- Example: cell nucleus
- Initialisation: external
- Fitting: iterative
 - Using the current geometry (“Contour influence”) and the image contents (“Image influence”)
 - “Influence”:
 - scalar = energy
 - vector = force
 - Until the energy is minimum, or the forces are null

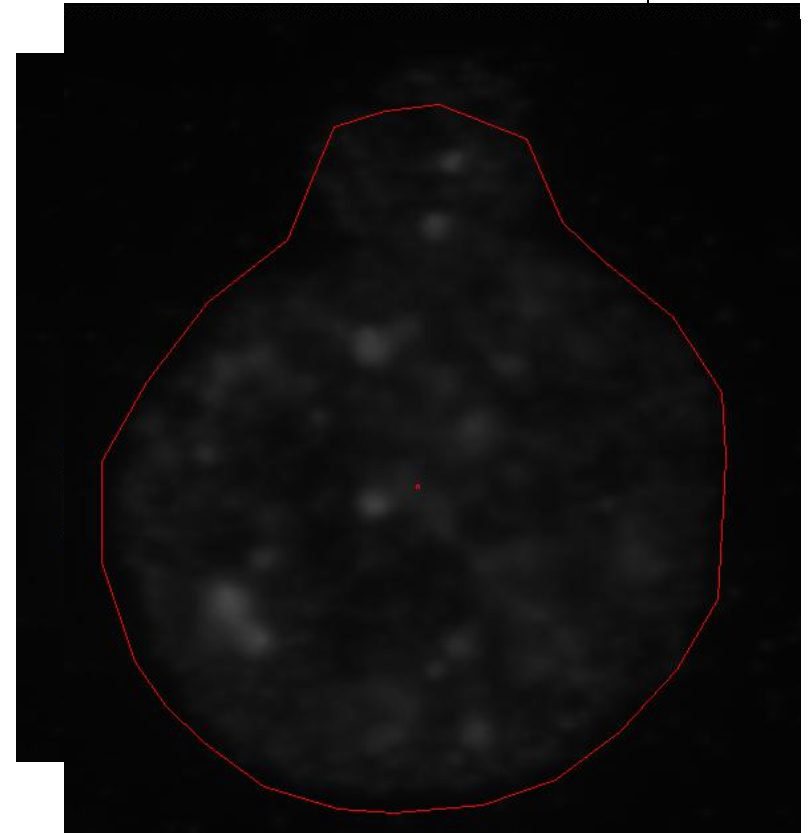


Image: Ian Diblik 2004

Traditional Snake model

□ Snake Model (1987) [Kass-Witkin-Terzopoulos]

- Planar parameterized curve $C: \mathbb{R} \rightarrow \mathbb{R} \times \mathbb{R}$
- A cost function defined along that curve

$$E[(C)(p)] = \alpha \int_0^1 E_{int}(C(p)) dp + \beta \int_0^1 E_{img}(C(p)) dp + \gamma \int_0^1 E_{con}(C(p)) dp$$

- The **internal term** stands for regularity/smoothness along the curve and has two components (resisting to stretching and bending)
- The **image term** guides the active contour towards the desired image properties (strong gradients)
- The **external term** can be used to account for user-defined constraints, or prior knowledge on the structure to be recovered
- The lowest potential of such a cost function refers to an equilibrium of these terms

Active Contour Components

□ The internal term...

$$E_{int}(C(p)) = w_{tension}(C(p)) \left| \frac{\partial C}{\partial p}(p) \right|^2 + w_{stiffness}(C(p)) \left| \frac{\partial^2 C}{\partial p^2}(p) \right|^2$$

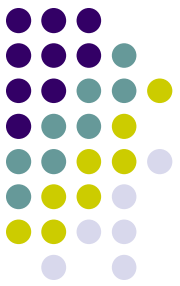
- The first order derivative makes the snake behave as a membrane
- The second order derivative makes the snake act like a thin plate

□ The image term...

$$E_{img}(C(p)) = w_{line}E_{line}(C(p)) + w_{edge}E_{edge}(C(p)) + w_{term}E_{term}(C(p))$$

- Can guide the snake to
 - Iso-phot $E_{line}(C(p)) = I(C(p))$, edges $E_{edge}(C(p)) = |\nabla I(C(p))|^2$
 - and terminations

□ Numerous Provisions...: balloon models, region-snakes, etc...

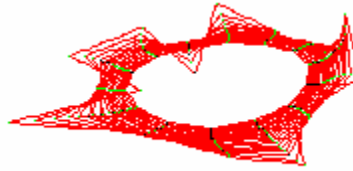


Internal force

- Elastic force (alpha force)—pull inward



(a) Initial contour



(b) after executing 20 times

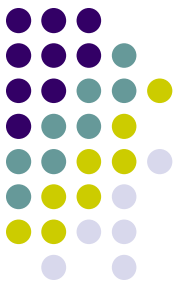
- Bending force (beta force) —smooth rather than shrinkage



(a) Initial contour



(b) after executing 20 times

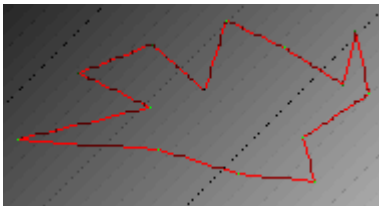


External force--- traditional snake model

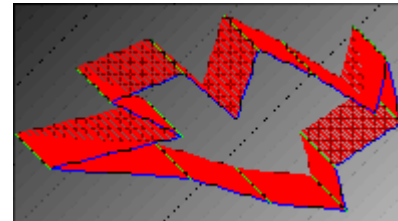
- Image gradient force
- The main difference among snake models
- F_{ext} in traditional snake model: $F_{ext}^{(P)} = -\nabla E_{ext}$.
- How to choose E_{ext} ---take on smaller value at the boundaries

$$E_{ext}^{(1)}(x, y) = -|\nabla I(x, y)|^2$$

$$E_{ext}^{(2)}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2$$



(a) Initial contour



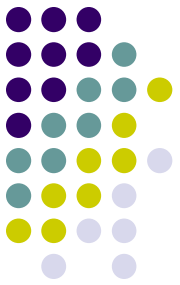
(b) after executing 20 times

Optimizing Active Contours

- Taking the Euler-Lagrange equations:

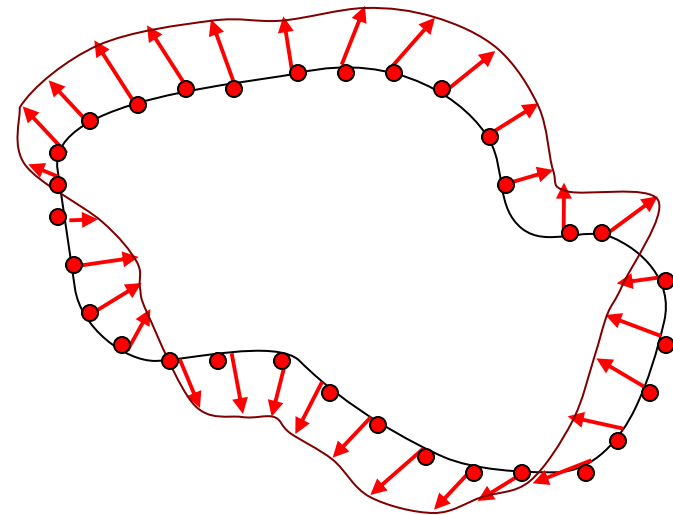
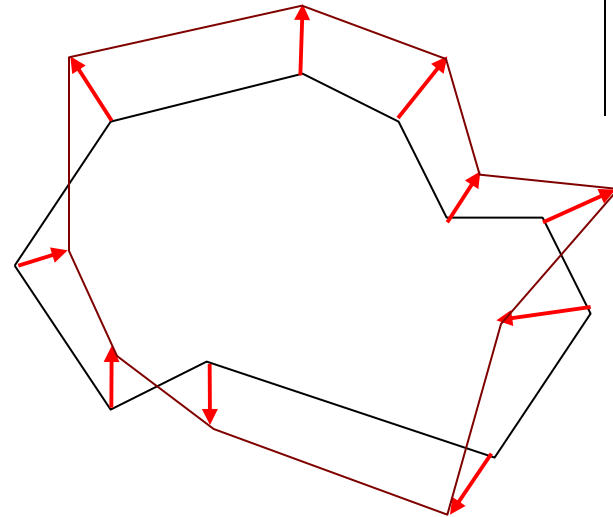
$$\alpha \left(w_{tension} \frac{\partial^2 C}{\partial p^2}(p) - w_{stiffness} \frac{\partial^4 C}{\partial p^4}(p) \right) - \beta \nabla E_{img}(C(p)) = 0$$

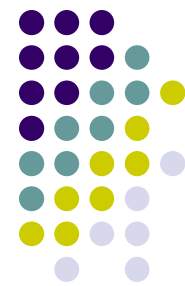
- That are used to update the position of an initial curve towards the desired image properties
 - Initial the curve, using a certain number of control points as well as a set of basic functions,
 - Update the positions of the control points by solving the above equation
 - Re-parameterize the evolving contour, and continue the process until convergence of the process...



- **Parametric AC:**
 - Stored as vertices
 - Each vertex is moved iteratively

- **Geometric AC:**
 - Stored as coefficients
 - Sampled before each iteration
 - Each sample is moved
 - New coefficients are computed (interpolation)





动态可变形模型

- 经典的可变形模型—snakes

- 假设：物体在图像中的边界是分段连续或光滑的，并且用参数表示为 $v(s) = (x(s), y(s)), s \in [0,1]$

- 轮廓的能量约束表示为 $E(v) = S(v) + P(v)$

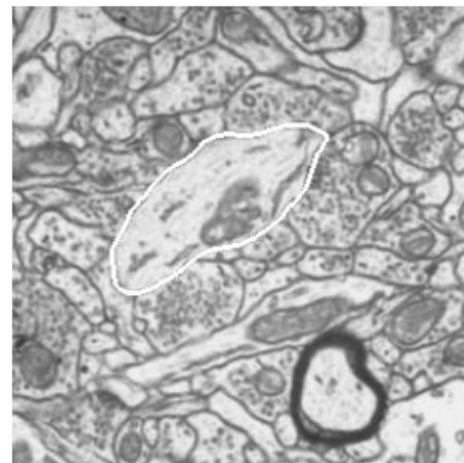
- 其中 **S** 表示对边界线的内部约束，**P** 表示对边界线的外部约束

- **S** 可以定义为 $S(v) = \frac{1}{2} \int_0^1 w_1(s) \left| \frac{\partial v}{\partial s} \right|^2 + w_2(s) \left| \frac{\partial^2 v}{\partial s^2} \right|^2 ds$
分别表示对曲线的张量和硬度的限制

- **P** 可以定义为 $P(x, y) = -c |\nabla[G_\sigma * I(x, y)]|$
表示曲线被吸引到梯度大（可能是边界）的区域

- 上式的 **Euler-Lagrange** 方程为

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





动态可变形模型

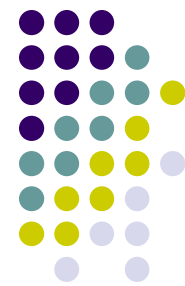
- 动态snake

- 经典的snake是一个静态问题，如果将问题的求解看作是曲线的演化，曲线变为 $v(s,t) = (x(s,t), y(s,t))$ 则上面问题可以化为

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

该式的稳定解就是原静态问题的解，其中前两项可以看作曲线的惯性力和粘滞力。





动态可变形模型

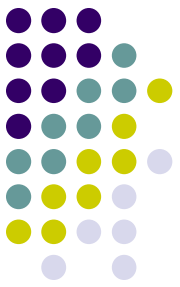
- 动态可变形模型的求解
 - 一般的方法式将几何模型 \mathbf{v} 表示为局部或全局基函数的线性组合（利用有限元法，有限差分法，或者样条函数法等），然后将问题转化为求解线性参数。

- 一般的形式为
$$E(u) = \frac{1}{2} u^T K u + P(u)$$

其中 \mathbf{K} 叫做刚性矩阵。最后的方程可以化为一个二次常微分形式

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

Problem of traditional snake



- Capture range
- Poor convergence

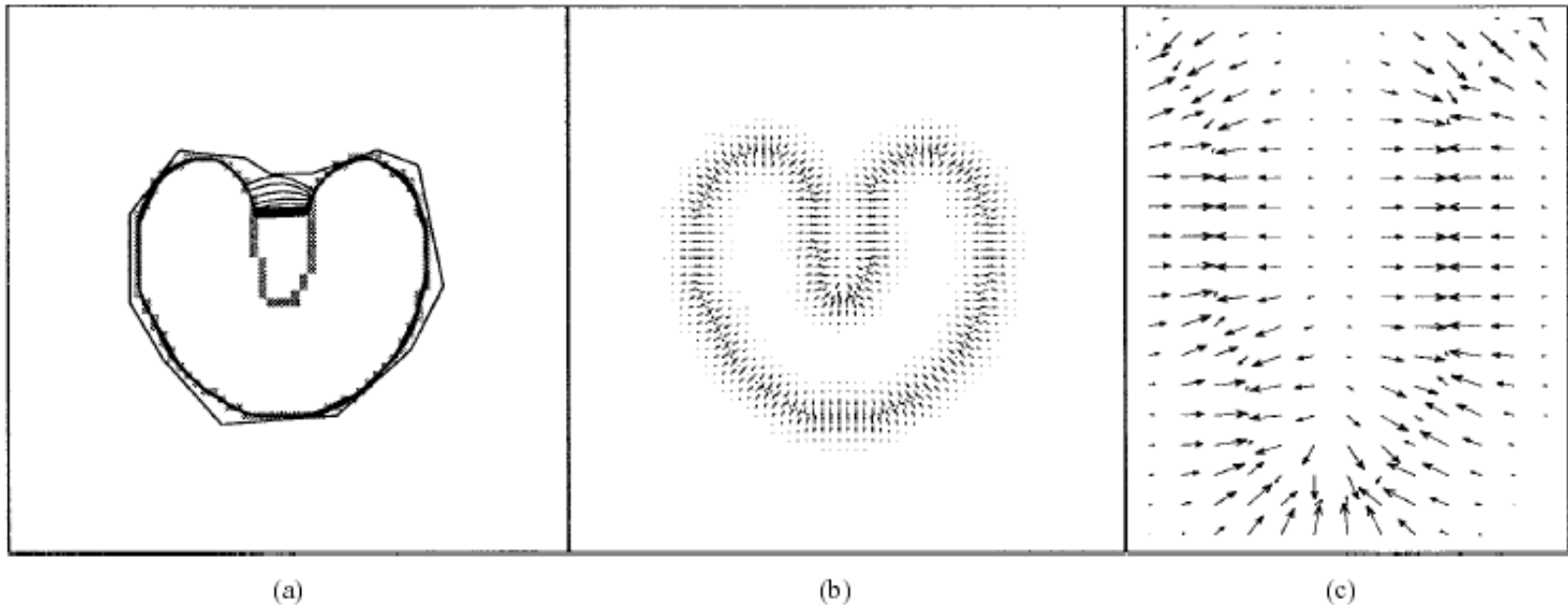
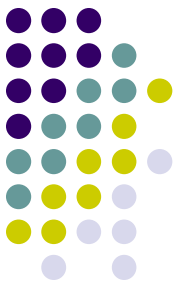


Fig. 1. (a) Convergence of a snake using (b) traditional potential forces, and (c) shown close-up within the boundary concavity.



External force---balloons

- L.D.Cohen and I.Cohen,1993

- Push the curve outward

$$F = k_1 \vec{n}(s) - k \frac{\nabla P}{\|\nabla P\|}(v(s)) \quad (0 < k_1 < k < 1)$$

- Problem: Poor convergence

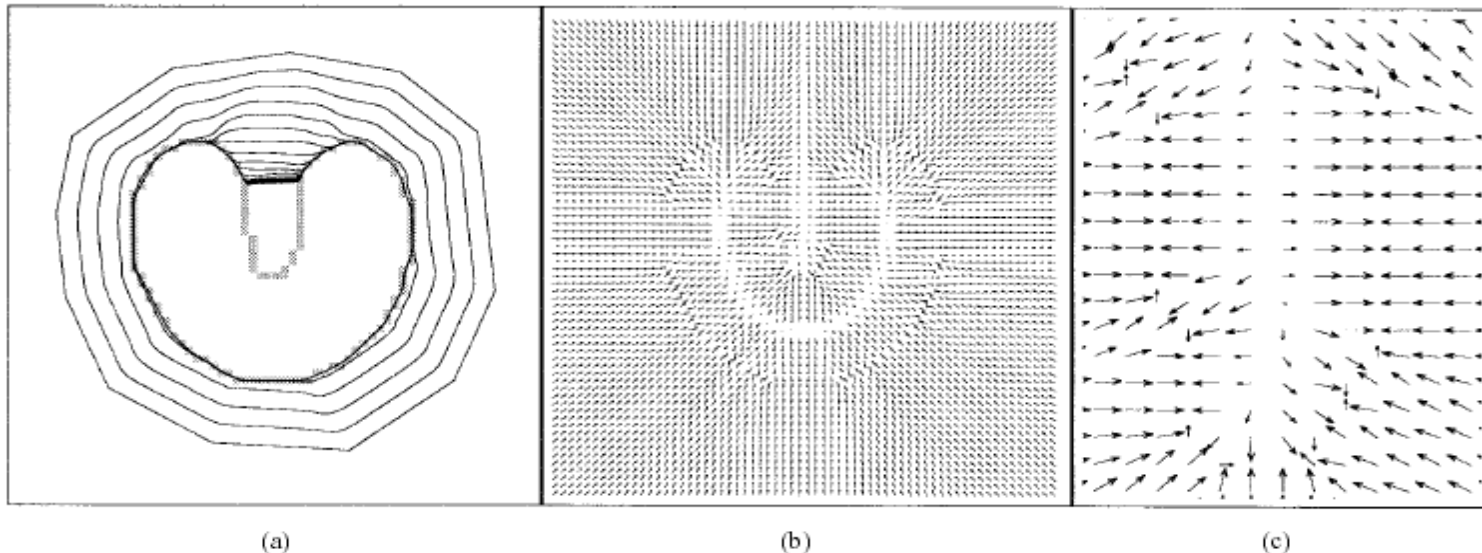
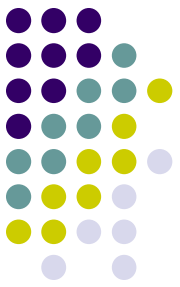
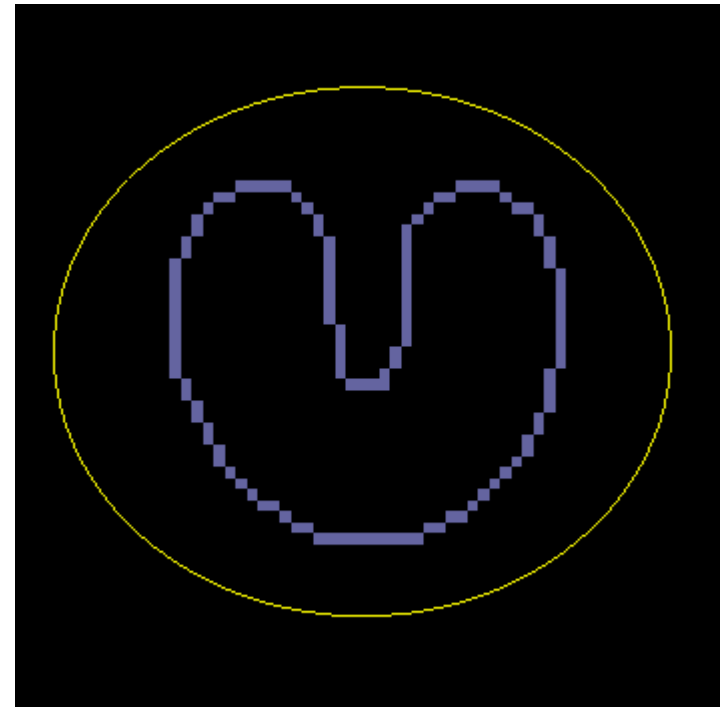


Fig. 2. (a) Convergence of a snake using (b) distance potential forces, and (c) shown close-up within the boundary concavity.



External force---GVF

- C Xu and J.L.Prince, 1998
- GVF (gradient vector flow):
 $\mathbf{F}_{\text{ext}}^{(g)} = \mathbf{v}(x, y)$
- Edge map: $f(x, y) = -E_{\text{ext}}^{(g)}(x, y)$
- Energy minimize:



$$\mathcal{E} = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dx dy.$$

Result

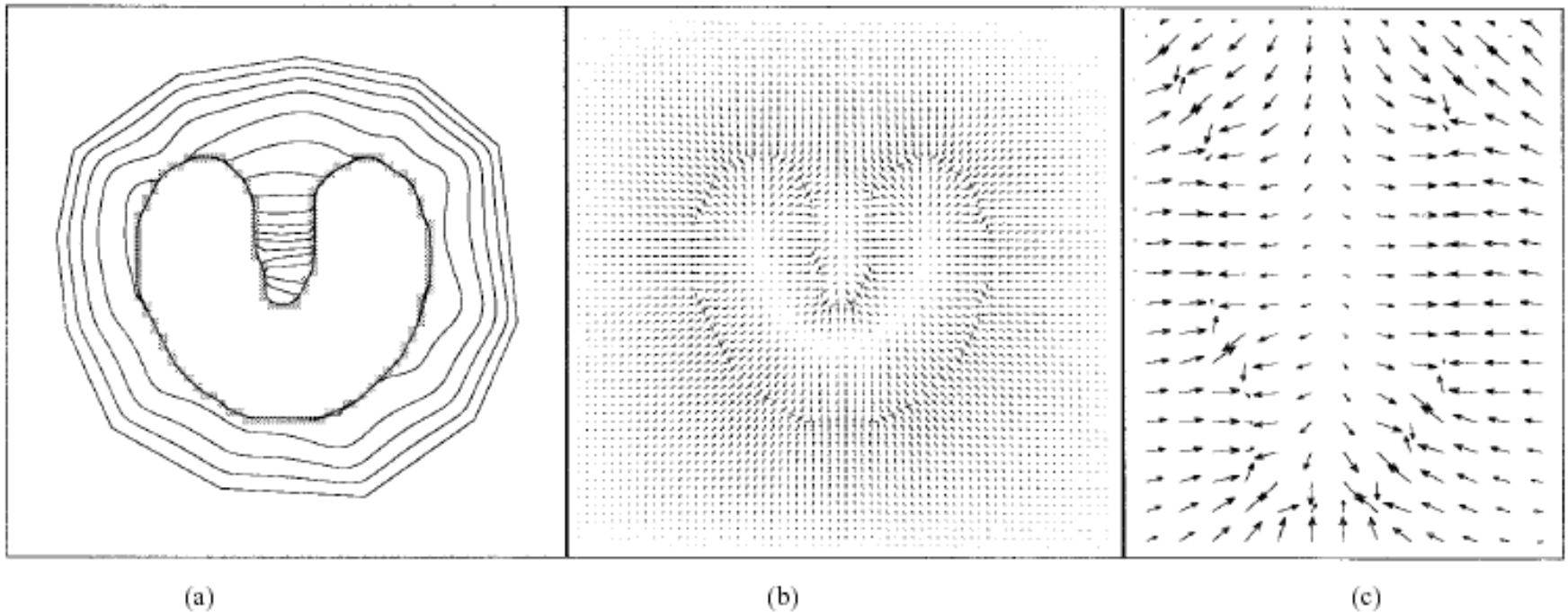
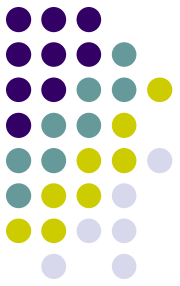
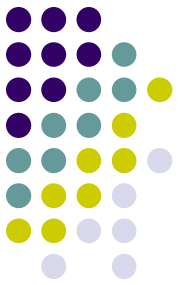
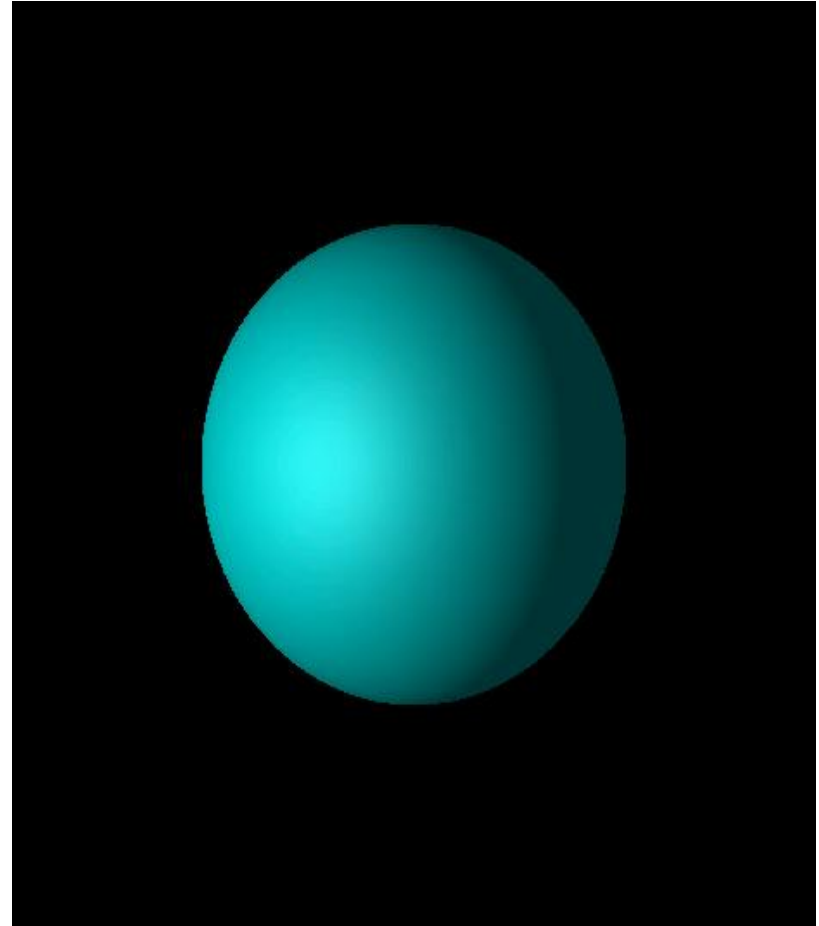


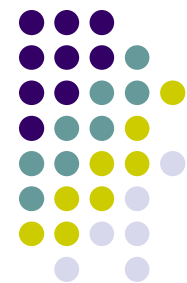
Fig. 3. (a) Convergence of a snake using (b) GVF external forces, and (c) shown close-up within the boundary concavity.



More about snake...

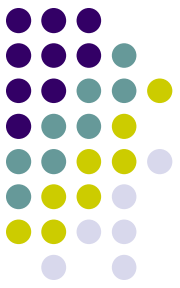
- Can't change topology
- Other model:
 - **Snakes with Topology Control**
-----*Stephan Bischoff, Leif Kobbelt*
- Higher dimension:
 - GVF—C.Xu,98
 - L.Cohen,95
 - ...





目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 方法概念
- 数值解法
 - 向上差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



Level Set — 水平集

- Level set 的数学定义

假设隐函数 $\varphi(x, t)$ 表示一个高维空间的方程，其在低维空间上的接触面为 $\varphi(x, t) = 0$ ，其中

$$x = (x_1, x_2, \dots, x_n) \in R^n$$

则level set 方程 $\Gamma(t)$ 有如下的性质，其中接触面表示为

$$\varphi(x, t) < 0 \quad \text{for } x \in \Omega$$

$$\varphi(x, t) > 0 \quad \text{for } x \notin \overline{\Omega}$$

$$\varphi(x, t) = 0 \quad \text{for } x \in \partial\Omega = \Gamma(t)$$

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:

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

- That is derived from the initial contour according to the following condition:

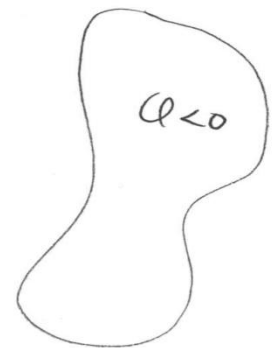
$$C(p, 0) = \{(x, y) : \varphi(x, y, 0) = 0\}$$

$$\{x | \varphi(x, t) = 0\}$$

defines $\Gamma(t)$.



$\ell > 0$

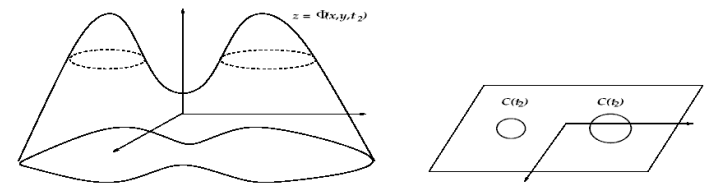
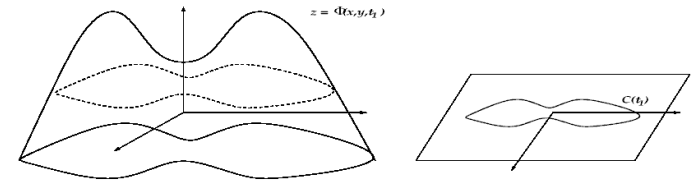


The Level Set Method

- Construction of the implicit function

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

$$C(p, t) = \{(x, y) : \varphi(x, y, t) = 0\}, C(t) = \varphi^{-1}(0)$$



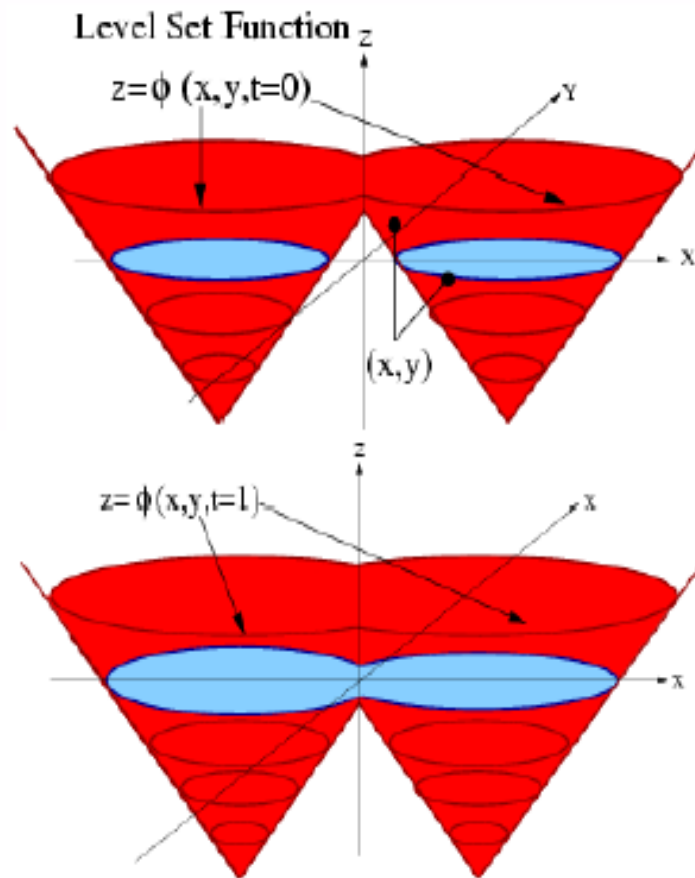
- And taking the derivative with respect to time (using the chain rule)

$$\varphi(C(t), t) = 0 \Rightarrow \frac{\partial \varphi}{\partial C} \cdot \underbrace{\frac{\partial C}{\partial t}}_{FN} + \underbrace{\frac{\partial \varphi}{\partial t}}_{\varphi_t} = 0 \quad (1)$$

- And we are DONE...

What is Level Set

- Adding an extra dimension to the problem



The level set function:

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

Contour at time t :

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

The level set PDE:

$$\Phi_t + F|\nabla\Phi| = 0$$

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

Evolving the level set

- Initialization of \emptyset :

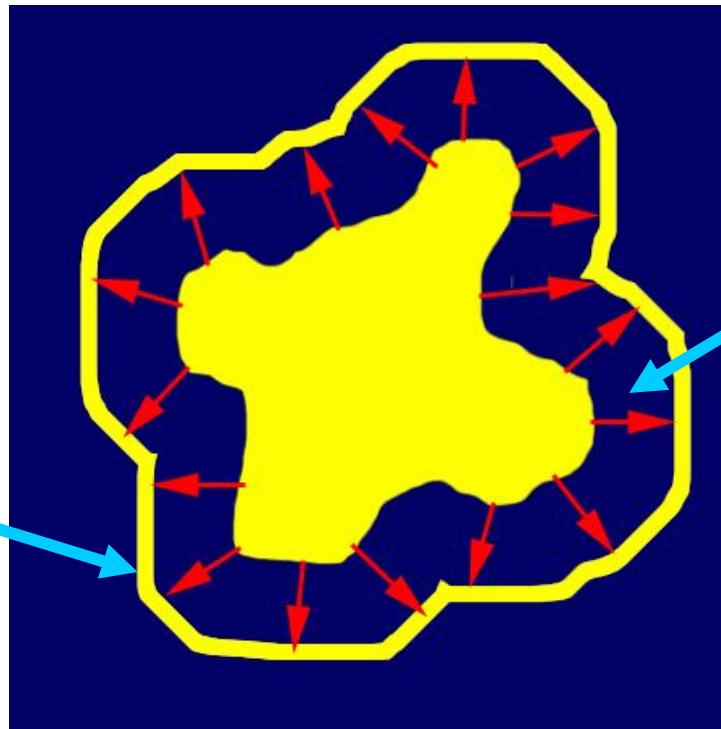
$$\psi(x, t=0) = \pm d$$

$$\varphi(x, t) > 0 \text{ for } x \in \Omega$$

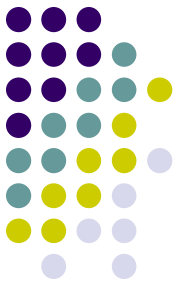
$$\varphi(x, t) < 0 \text{ for } x \notin \bar{\Omega}$$

- Pr $\varphi(x, t) = 0$ for $x \in \partial\Omega = \Gamma(t)$

New front



Add:
 $F|\nabla\Phi|$
(velocity)



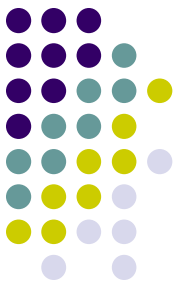
Level Set — 水平集

- Level set 的运动可以表示为

$$\frac{\partial \varphi}{\partial t} + v \cdot \nabla \varphi = 0$$

- 假设 $v_N = v \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$ 则上式成为

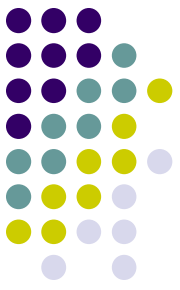
$$\frac{\partial \varphi}{\partial t} + v_N |\nabla \varphi| = 0$$



Level Set —水平集

- Level set 方法的几何意义很直观：
 - 给定一个高维空间在低维空间 (n 维) 定义上的接触面，分析和计算其边界在速度 v 下的运动轨迹
 - 速度 v 是与位置，时间和接触面的几何形状有关的（如平均曲率，法向），还有外部的物理作用力。

(UCLA的Osher和Sethian首先提出了这个方法)



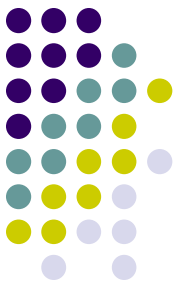
Level Set —水平集

- 小结

- **Level set** 方法实际上就是求解一个随时间变化的偏微分方程

$$\frac{\partial \varphi}{\partial t} + v_N |\nabla \varphi| = 0$$

- 其中 v_N 表示可以是任何关于时间，位置，几何等量的函数。
- 有时提到 **Level set** 方法是指其数值解方法
- 应用**Level set**方法需要解决两个问题
 - 如何列出有意义的方程求解实际问题
 - 如何能快速、稳定地求出方程的数值解

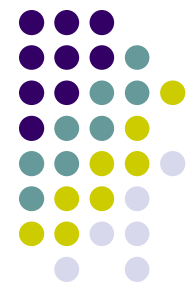


Level Set—水平集

- 通常，演化速度 F 含有三种成份：

$$\frac{\partial C}{\partial t} = g(\alpha + \beta k) \mathbf{n} \quad \alpha, \beta \in \mathbf{R}, \beta \geq 0$$

1. 与曲率相关的所谓扩散项起到保持曲线的光滑性的作用
2. 对流项为曲线演化提供动力支持
3. 速度衰减因子使速度在边缘轮廓处停止



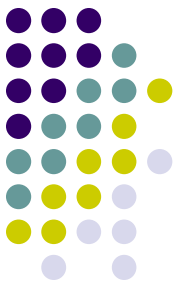
Level Set —水平集

- Level set 的主要优点就是其考虑了物体几何的一些更本质的特征（如曲率，梯度等），所以得到的结果能够比已有的一些其他方法要好。



目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 概念
- Level set 的数值解法
 - 向上差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



Level set 的数值解法

- 几种差分方法的稳定性分析

- 假设求解一阶一维方程 $u_t + au_x = 0, u(x,0) = u_0(x)$

- 向前差分格式

$$u_j^{n+1} = u_j^n - ar(u_j^n - u_{j-1}^n)$$

- 向后差分格式

$$u_j^{n+1} = u_j^n - ar(u_{j+1}^n - u_j^n)$$

- 中心差分格式

$$u_j^{n+1} = u_j^n - ar(u_{j+1}^n - u_{j-1}^n) / 2$$



Level set 的数值解法

- 向前差分格式的稳定性条件 $0 \leq ar \leq 1$
- 向后差分格式的稳定性条件 $-1 \leq ar \leq 0$
- 中心差分格式的稳定性条件 恒不稳定

(稳定性条件的分析方法 (如Von Neumann方法) 参见数值分析的教材)



Level set 的数值解法—CIR格式

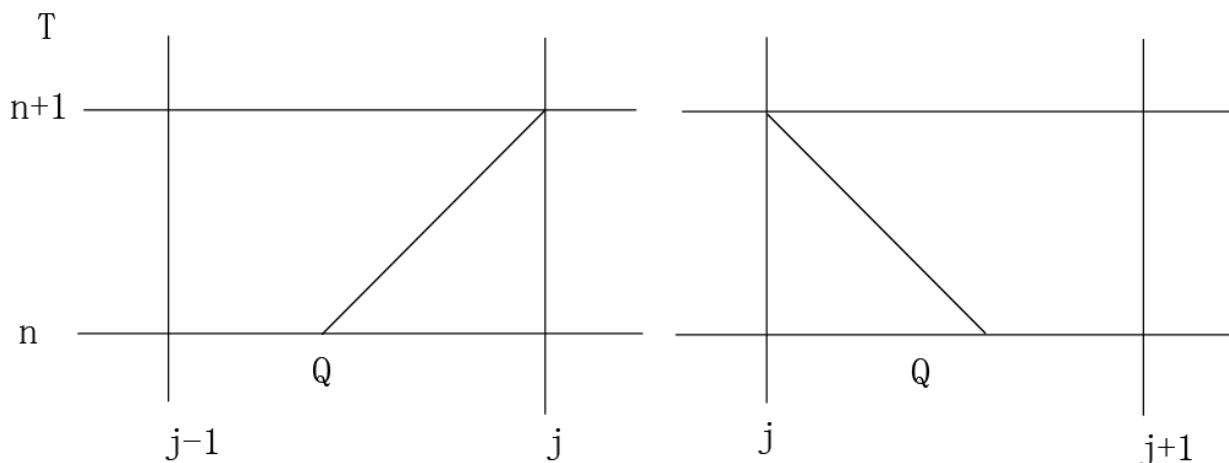
- 所以选择差分格式要根据 a 的符号来判断

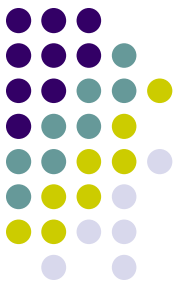
解为

$$u_j^{n+1} = \begin{cases} u_j^n - ar(u_j^n - u_{j-1}^n), & a \geq 0 \\ u_j^n - ar(u_{j+1}^n - u_j^n), & a < 0 \end{cases}$$

或

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a^+ \frac{u_j^n - u_{j-1}^n}{h} - a^- \frac{u_{j+1}^n - u_j^n}{h} = 0$$





Level set 的数值解法—CIR格式

- 其中

$$a^{\pm} = \frac{1}{2}(|a| \pm a)$$

上式称为Courant-Isaacson-Rees格式（CIR格式）

在求解其它类型的偏微分方程时也要用到类似的格式，不过分析其稳定性要复杂很多。



Level set 的数值解法—LF格式

- 如果将中心差分格式的 u_j^n 替换为 $\frac{1}{2}(u_{j+1}^n + u_{j-1}^n)$ 则得到Lax-Friedrichs格式

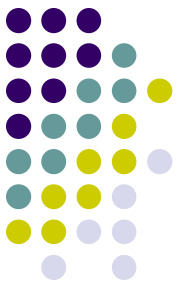
$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{ar}{2}(u_{j+1}^n - u_{j-1}^n)$$

其稳定性条件为 $|ar| \leq 1$



目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 概念
- Level set 的数值解法
 - Upwind 差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系

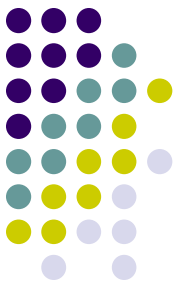


Upwind 差分法

- 假设 $t^n = n\Delta t$

$$\boxed{\frac{\partial \varphi}{\partial t} + v \cdot \nabla \varphi = 0} \quad \longrightarrow \quad \frac{\varphi^{n+1} - \varphi^n}{\Delta t} + v^n \cdot \boxed{\nabla \varphi^n} = 0$$

根据CIR格式，先考虑一维的情况，当 $v^n > 0$ 时，曲线从左往右移动，所以要用到 φ_i^n 左边的值，即用向后差分来估计 $\nabla \varphi^n$ 。同理，当 $v^n < 0$ 时，用向前差分。



Upwind 差分法

- 算法的精度

$$O(\Delta x)$$

- 算法的稳定条件

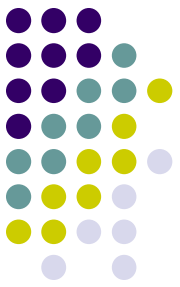
$$\Delta t \max\left\{ \frac{|v_x|}{\Delta x} + \frac{|v_y|}{\Delta y} + \frac{|v_z|}{\Delta z} \right\} \in (0,1)$$

可见*upwind*差分法虽然简单，但是精度不高，计算速度慢



目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 概念
- Level set 的数值解法
 - Upwind 差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



Hamilton-Jacobi ENO 方法

- ENO: Essentially Nonoscillatory (不波动, 不摇动)
- 基本思想: 用尽量光滑的多项式插值 φ 然后再求解 φ_x 。用HJ ENO方法可以更精确地估计 φ_x^+ 或者 φ_x^- 。

$$D_i^0 = \varphi_i$$

- 定义算子 $D_{i+1/2}^1 \varphi = \frac{D_{i+1}^0 \varphi - D_i^0 \varphi}{\Delta x}$

$$D_i^2 \varphi = \frac{D_{i+1/2}^1 \varphi - D_{i-1/2}^1 \varphi}{2\Delta x}$$

...



Hamilton-Jacobi ENO 方法

- 上面的分步差分可以用来重建如下形式的插值多项式

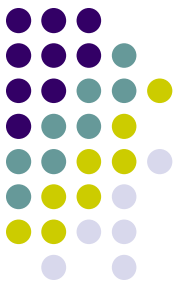
$$\varphi(x) = Q_0(x) + Q_1(x) + Q_2(x) + Q_3(x)$$

然后通过通过对 $\varphi(x)$ 的求导可以得到 φ_x^- 或 φ_x^+ 的估计值。

$$\varphi_x(x_i) = Q_1'(x_i) + Q_2'(x_i) + Q_3'(x_i)$$

- 例如，为了估计 φ_x^- ，应该从 $k=i-1$ 开始，为了估计 φ_x^+ ，应该从 $k=i$ 开始。

然后定义 $Q_1(x) = (D_{k+1/2}^1 \varphi)(x - x_i)$



Hamilton-Jacobi ENO 方法

即 $Q_1'(x) = D_{k+1/2}^1 \varphi$

得到 φ_x 的一阶精度估计（其形式是和upwind方法得到的形式是一样的）

- 同样可以得到二阶和更高阶精度的估计。

二阶精度估计

$$Q_2'(x_i) = D_{k+1/2}^2 \varphi(2(i-k)-1)\Delta x$$

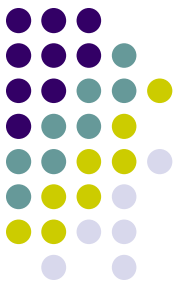
三阶精度估计 $k^*=k-1$

$$Q_3'(x_i) = D_{k+1/2}^3 \varphi(3(i-k^*)^2 - 6(i-k^*)^2 + 2)(\Delta x)^2$$



目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 概念
- Level set 的数值解法
 - Upwind 差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



Hamilton-Jacobi WENO 方法

- WENO: Weighted ENO
- 当计算 $(\varphi_x^-)_i$ 时, 三阶精度的HJ ENO算法需要知道 $\{\varphi_{i-3}, \varphi_{i-2}, \varphi_{i-1}, \varphi_i, \varphi_{i+1}, \varphi_{i+2}\}$ 的值, 共有3种HJ ENO估计 $(\varphi_x^-)_i$ 的方法。定义

$$v_1 = D^- \varphi_{i-2}$$

$$v_2 = D^- \varphi_{i-1}$$

$$v_3 = D^- \varphi_i$$

$$v_4 = D^- \varphi_{i+1}$$

$$v_5 = D^- \varphi_{i+2}$$

则三种估计为

$$\varphi_x^1 = \frac{v_1}{3} - \frac{7v_2}{6} + \frac{11v_3}{6}$$

$$\varphi_x^2 = -\frac{v_2}{3} + \frac{5v_3}{6} + \frac{v_4}{3}$$

$$\varphi_x^3 = \frac{v_3}{3} + \frac{5v_4}{6} - \frac{v_5}{6}$$



Hamilton-Jacobi WENO 方法

- HJ ENO方法的目的是就是从上面3个估计中选出一个最光滑的多项式逼近。
- 后来有人提出这种HJ ENO方法可以进一步通过将3个式子加权来提高精度。
- WENO的形式可以写成

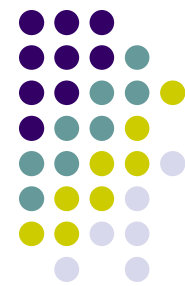
$$\varphi_x = w_1 \varphi_x^1 + w_2 \varphi_x^2 + w_3 \varphi_x^3, \quad w_1 = 0.1, w_2 = 0.6, w_3 = 0.3$$
$$w_1 + w_2 + w_3 = 1, \quad 0 \leq w_i \leq 1$$

可以证明对光滑区域可以达到**5阶精度**！



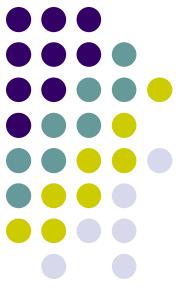
目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 概念
- Level set 的数值解法
 - Upwind 差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



TVD Runge-Kutta 方法

- 前面提到的HJ ENO 和 HJ WENO可以达到5阶精度，向前Euler算法（Upwind算法）只能达到1阶精度。
- 用TVD RK方法可以达到更高阶的精度
- TVD: total variation diminishing
- 一阶TVD RK就是向前Euler算法。
- 二阶TVD RK和二阶RK算法相似。



TVD Runge-Kutta 方法

- 三阶TVD RK算法

$$\frac{\varphi^{n+1} - \varphi^n}{\Delta t} + v^n \cdot \nabla \varphi^n = 0$$

$$\frac{\varphi^{n+2} - \varphi^{n+1}}{\Delta t} + v^{n+1} \cdot \nabla \varphi^{n+1} = 0$$

$$\varphi^{n+1/2} = \frac{3}{4} \varphi^{n+1} + \frac{1}{4} \varphi^{n+2}$$

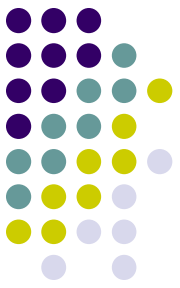
$$\frac{\varphi^{n+3/2} - \varphi^{n+1/2}}{\Delta t} + v^{n+1/2} \cdot \nabla \varphi^{n+1} = 0$$

$$\varphi^{n+1} = \frac{1}{3} \varphi^n + \frac{2}{3} \varphi^{n+3/2}$$



TVD Runge-Kutta 方法

- 虽然4阶以上的TVD RK方法存在，但在实际应用中，在时间上的精度的提高并不会对结果有太大的改进，而且从计算复杂性上考虑会得不偿失。



求解Level set的数值方法小结

- Level set 的一般形式可以表达为

$$\frac{\partial \varphi}{\partial t} + v \cdot \varphi = 0$$

$$v = a(x) \cdot \nabla \varphi - \mu(x) \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right)$$

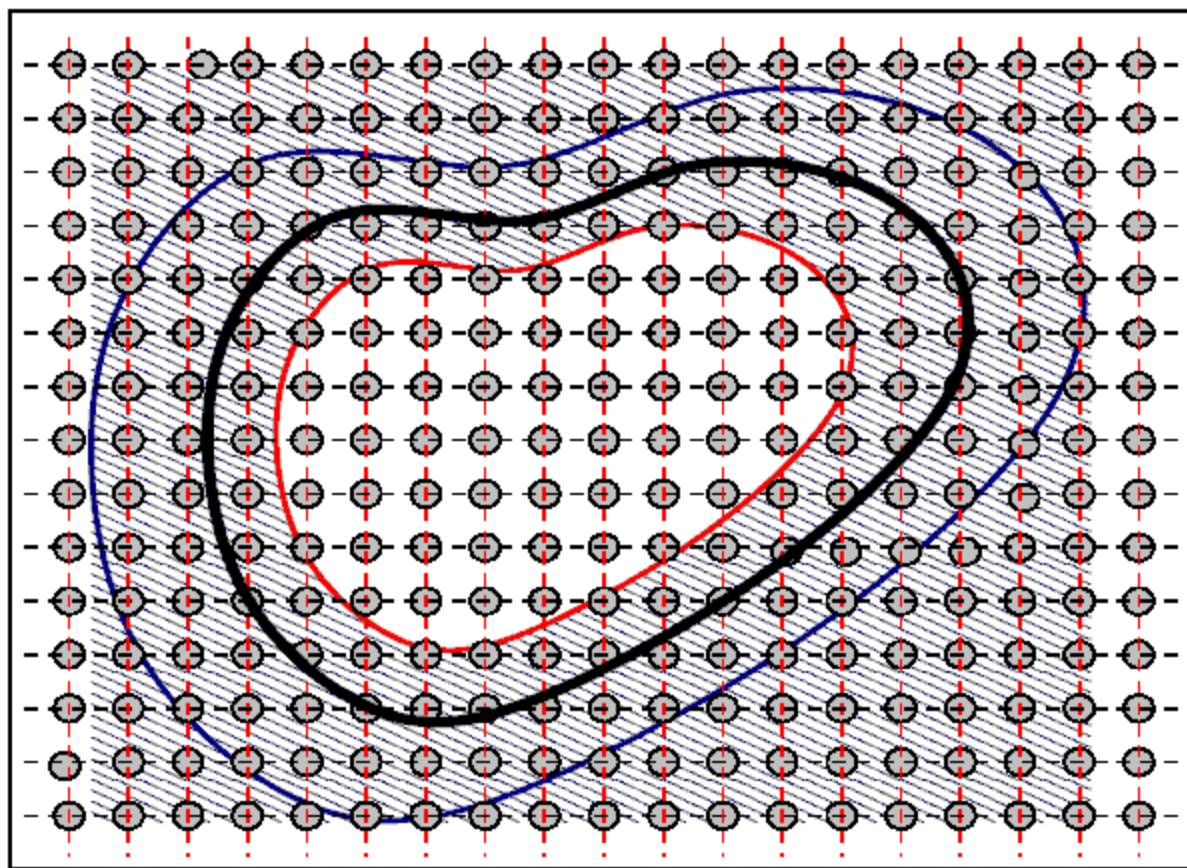
其中 $a(x) \cdot \nabla \varphi$ 称为对流项， $\nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right)$ 为曲率。



求解Level set的数值方法小结

- 求解该方程一般分为3步
 - 用ENO,WENO或upwind方法求解对流项。
 - 用中心差分的方法估算曲率。
 - 用TVD RK方法来求解。

更高级的解法? ! ?



Outward Band

$$\Phi(s) = +d$$

Front Position

$$\Phi(s) = 0$$

Inward Band

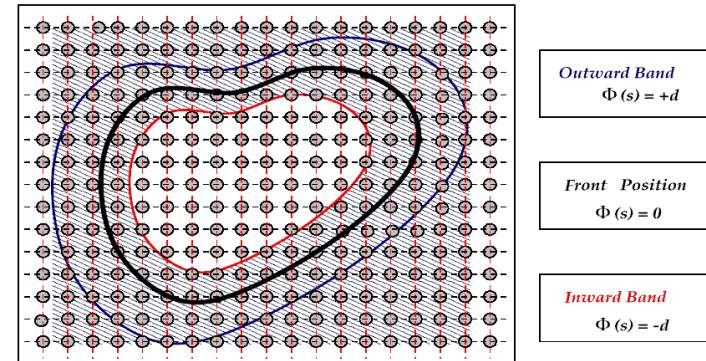
$$\Phi(s) = -d$$

From theory to Practice (Narrow Band)

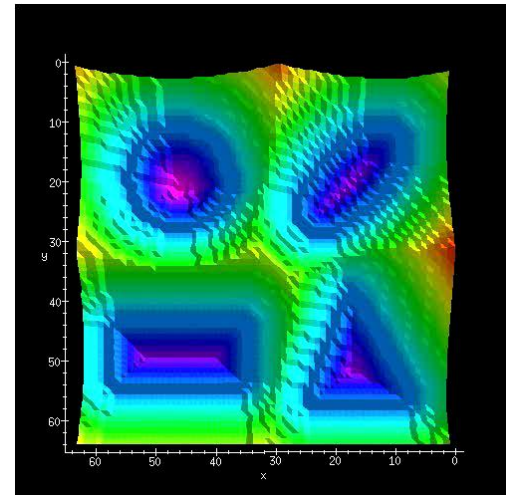
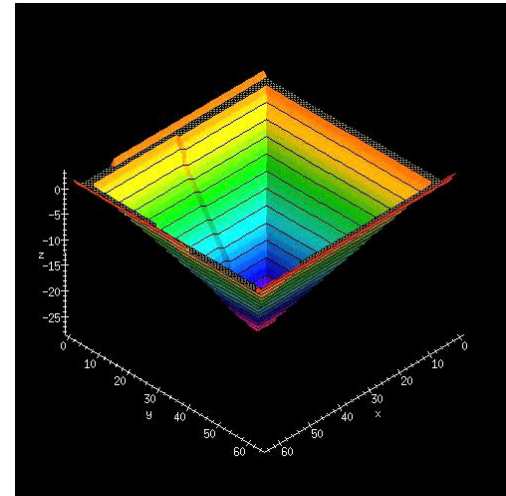
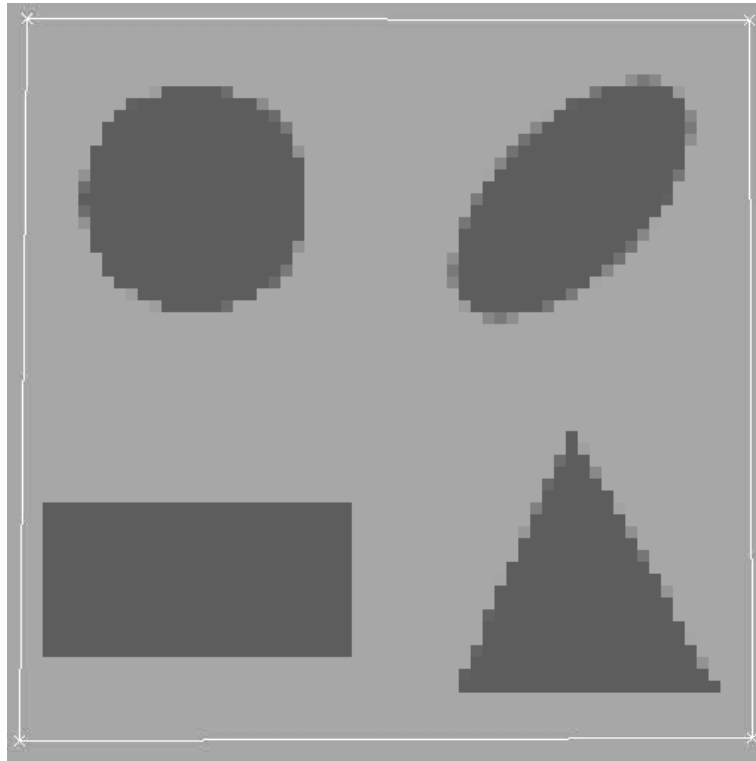
[Chop:93, Adalsteinsson-Sethian:95]

- Central idea: we are interested on the motion of the zero-level set and not for the motion of each iso-phote 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



Narrow Band (the basic derivation)



Results are courtesy: R. Deriche

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 Borgfors 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)

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

- 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-photos, rather complicated approach with limited added value?

From theory to Practice (Fast Marching)

[Tsitsiklis:93, Sethian:95]

- Central idea: “move” the curve one pixel in a progressive manner according to the speed function while preserving the nature of the implicit function
- Consider the stationary equation $F |\nabla T| = 1$.
- Such an equation can be recovered for all $[\frac{\partial C}{\partial t} = F\mathcal{N}]$ flows where the speed function has one sign (either positive or negative), propagation takes place at one direction
- If $T(x,y)$ is the time when the implicit function reaches (x,y) :

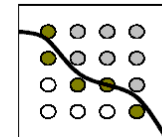
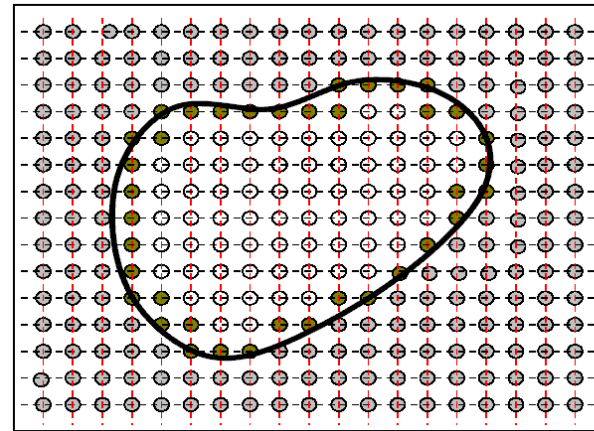
$$\begin{aligned}
 T(C(p,t)) \triangleq t &\Rightarrow \nabla T \cdot C_t = 1 \\
 &\Rightarrow \nabla T \cdot \left(F \frac{\nabla T}{|\nabla T|} \right) = 1 \\
 &\Rightarrow F |\nabla T| = 1
 \end{aligned}$$

Fast Marching (continued)

- Consider the stationary equation $F |\nabla T| = 1$ in its discrete form:

$$\frac{1}{F_{\{i,j\}}^2} = \max(D_{\{i,j\}}^{-x} T, 0)^2 + \min(D_{\{i,j\}}^{+x} T, 0)^2 + \max(D_{\{i,j\}}^{-y} T, 0)^2 + \min(D_{\{i,j\}}^{+y} T, 0)^2$$

- And using the assumption that the surface propagates in one direction, the solution can be obtained by outwards propagation from the smallest T value...



Zoom Window

○ Alive

● Active

○ FarAway

- active pixels, the curve has already reached them
- alive pixels, the curve could reach them at the next stage
- far away pixels, the curve cannot reach them at this stage

Fast Marching (continued)

□ INITIAL STEP

- Initialize $[T = 0]$ for the all pixels of the front (**active**), their first order neighbors **alive** and the rest **far away**
- For the first order neighbors,
estimate the arrival time according to: $[T_{\{i,j\}} = \frac{1}{F_{\{i,j\}}}]$
- While for the rest the crossing time is set to infinity $[T_{\{i,j\}} = \infty]$

□ PROPAGATION STEP

- Select the pixel with the lowest arrival time from the **alive** ones
- Change his label from **alive** to **active** and for his first order neighbors:
 - If they are **alive**, update their T value according to

$$\frac{1}{F_{\{i,j\}}^2} = \max(D_{\{i,j\}}^{-x} T, 0)^2 + \min(D_{\{i,j\}}^{+x} T, 0)^2 + \max(D_{\{i,j\}}^{-y} T, 0)^2 + \min(D_{\{i,j\}}^{+y} T, 0)^2$$

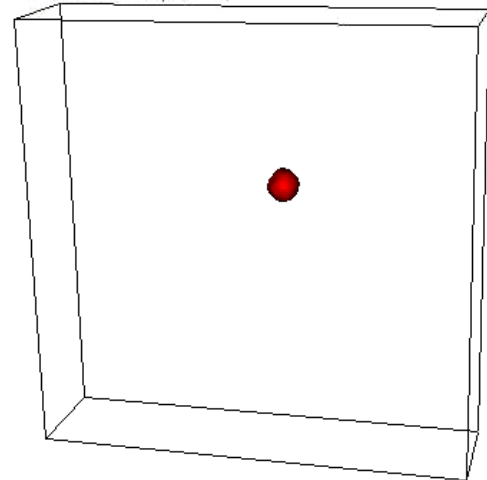
- If they are **far away**, estimate the arrival time according to: $[T_{\{i,j\}} = \frac{1}{F_{\{i,j\}}}]$

Fast Marching Pros/Cons, Some Results

- ❑ Fast approach for a level set implementation
- ❑ Very efficient technique for re-setting the embedding function to be distance transform
- ❑ Single directional flows, great importance on initial placement of the contours
- ❑ Absence of curvature related terms or terms that depend on the geometric properties of the curve...
- ❑ Results are courtesy: J. Sethian, R. Malladi, T. Deschamps, L. Cohen



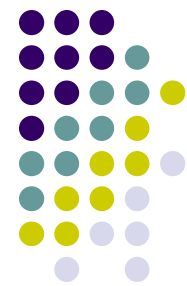
Thomas Deschamps(2003)





目录

- 数学基础
 - 微分几何简介
 - 数学形态学
 - 隐函数，距离函数
- Level Set 概念
- Level set 的数值解法
 - Upwind 差分法
 - Hamilton-Jacobi ENO
 - Hamilton-Jacobi WENO
 - TVD Runge-Kutta
- Level set 与变分方程的关系



Level set 与变分方程的关系

- 考虑Poisson方程的Dirichlet问题

$$\begin{cases} -\Delta u = f(x), & x \in \Omega \\ u = \varphi(x), & x \in \partial\Omega \end{cases}$$

定义能量函数

$$I[w] = \int_{\Omega} \left(\frac{1}{2} |Dw|^2 - wf \right) dx$$

- Dirichlet原理:

$$I(u) = \min_{w \in A} I[w]$$



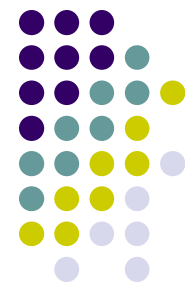
Level set 与变分方程的关系

- 一般的，假定 $Au = f(x)$ 是线性偏微分方程，则 $F(u) = (Au, u) - 2(u, f)$ 的极小函数为 $Au = f(x)$ 的解。其中 $(*, *)$ 表示任何Banach空间算子。
- 对于很多偏微分方程，一般都可以找到一个与之对应的变分方程，其中边界值可以通过Green公式化简到内部积分。



Level set 与变分方程的关系

- 对于大多数变分方程，也可以通过Lagrange公式化归为相应的偏微分方程求解。
- 在后面的应用中我们可以看到很多问题都是先列出变分方程，再求解对应的偏微分方程得到Level set方程的。



Level set 与变分方程的关系

- 例(Total Variation模型):

$$F(u) = \int |\nabla u| dx + \lambda \int |f - u|^2 dx$$

用Euler-Lagrange方程化为偏微分方程为

$$u = f + \frac{1}{2\lambda} \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$$

问题的求解可以化为Level set演化的形式，即

$$u_t = -(u - f) + \frac{1}{2\lambda} \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$$



小结

- **Level Set** 方法可以看作是一类带时间的偏微分方程框架，其意义不仅是其有很多的应用，而且对这一类方程的求解的方法也是其受到广泛重视的重要原因。



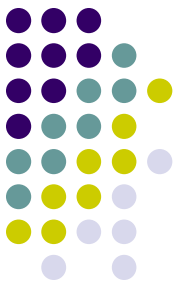
参考资料

- Level set 的Matlab 工具箱

<http://www.cs.ubc.ca/~mitchell/ToolboxLS/>

用matlab实现了三种数值算法

- 有好的C++语言实现的Level Set工具箱吗？



参考文献

- [1] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag (2002).
- [2] James A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press (1999).
- [3] J.A. Sethian. *Level Set Methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press(1996).