Beltrami Framework and Geodesic Active Contours

- The Beltrami framework
  - multi-channel diffusion
  - diffusion in HSV color space
- Geodesic active contours
  - Euclidean geodesic active contours
  - affine invariant geodesic active contours

Beltrami Framework

- last week: 2d image U as manifold via

$$\sigma(x,y) = (x, y, \beta\, U(x,y)) \subset \mathbb{R}^3\,, \quad (x,y) \in D$$

  with first fundamental form

$$\mathbf{I}_{(x,y)} = \begin{pmatrix} 1+\beta^2 U_x^2 & \beta^2 U_x U_y \\ \beta^2 U_x U_y & 1+\beta^2 U_y^2 \end{pmatrix}$$

  and

$$\sigma_t = \frac{1}{\sqrt{\det \mathbf{I}_{(x,y)}}}\ \mathrm{div}\left(\sqrt{\det \mathbf{I}_{(x,y)}}\, \mathbf{I}_{(x,y)}^{-1}\, \nabla\sigma\right)$$

- Beltrami flow for grey scale images: projection onto 3rd component:

$$U_t = \frac{U_{xx}(1+\beta^2 U_y^2) + U_{yy}(1+\beta^2 U_x^2) - 2\beta^2 U_{xy} U_x U_y}{(1+\beta^2 U_x^2 + \beta^2 U_y^2)^2}$$

Beltrami Flow for Multi-Channel Images

- Consider an image $U : D \to \mathbb{R}^k$, $D \subset \mathbb{R}^2$, with k-tuples $(k > 1)$ as values
  - Colour images: $k = 3$
  - Vector fields on D: $k = 2$

  Particularly in computer science, tuples are often called vectors. We distinguish the notions here to avoid confusion between e.g. colour images in which the channels have no spatial meaning, and true vector fields

- σ becomes a surface in $\mathbb{R}^{k+2}$, e.g. in $\mathbb{R}^5$ for colour images

- Computation of geometric diffusion flow, and projection on those dimensions corresponding to image data, analogous to scalar-valued images, leads to **Beltrami flow for multi-channel images**

$$\partial_t U_j = \frac{1}{\sqrt{\det \mathbf{I}_{(x,y)}}} \operatorname{div}\left(\sqrt{\det \mathbf{I}_{(x,y)}}\, \mathbf{I}_{(x,y)}^{-1} \nabla U_j \right) , \quad j = 1, \ldots, k$$

- This Beltrami flow is again a gradient descent for the surface area w.r.t. a suitable inner product of multi-channel functions

Beltrami Flow for Multi-Channel Images, cont.

- *Caveat:* Explicit expressions become more involved, since

$$\mathbf{I}_{(x,y)} = \begin{pmatrix} 1+\beta^2 \sum\limits_{j=1}^{k} (\partial_x U_j)^2 & \beta^2 \sum\limits_{j=1}^{k} (\partial_x U_j \cdot \partial_y U_j) \\ \beta^2 \sum\limits_{j=1}^{k} (\partial_x U_j \cdot \partial_y U_j) & 1+\beta^2 \sum\limits_{j=1}^{k} (\partial_y U_j)^2 \end{pmatrix}$$

$$\begin{aligned} \det \mathbf{I}_{(x,y)} &= 1 + \beta^2 \sum_{j=1}^{k} ((\partial_x U_j)^2 + (\partial_y U_j)^2) \\ &\quad + \beta^4 \left(\sum_{j=1}^{k} (\partial_x U_j)^2 \cdot \sum_{j=1}^{k} (\partial_y U_j)^2 - \left(\sum_{j=1}^{k} (\partial_x U_j)(\partial_y U_j) \right)^2 \right) \end{aligned}$$

Beltrami Flow for Multi-Channel Images – Example

Removing salt-and-pepper noise (G. Rosman et al. 2000)

General Framework

- introduced by Kimmel, Malladi and Sochen
- relies on maps between manifolds $\mathbf{X}\colon \Sigma \to M$
  - (Σ, g) image manifold, $\dim \Sigma = m$
  - (M, h) feature space manifold
- energy functional / measure on maps

$$S[X^i, g_{\nu\mu}, h_{ij}] = \int d^m\sigma \sqrt{g} g^{\nu\mu} \partial_\mu X^i \partial_\nu X^j h_{ij}(\mathbf{X})$$

  - g - determinant of image metric, $g^{\nu\mu}$ - inverse of image metric
  - Einstein summation convention: summation over indices that appear twice, e.g.

$$\langle a, b\rangle = a_i b^i = \sum_{i=1}^{n} a_i b^i$$

  - minimisation wrt. $g_{\mu\nu}$ gives induced metric

$$g_{\mu\nu} = h_{ij} \partial_\mu X^i \partial_\nu X^j$$

Beltrami Framework (6)

- Example 1: surface embedded in $\mathbb{R}^3$ with $(\Sigma, g) = (\mathbb{R}^2, \delta_{ij})$, $(M, g) = (\mathbb{R}^3, \delta_{ij})$, e.g.

$$\mathbf{X}(\sigma_1, \sigma_2) = (X^1(\sigma_1, \sigma_2), X^2(\sigma_1, \sigma_2), X^3(\sigma_1, \sigma_2))$$

$$S[\mathbf{X}] = \int d^2\sigma(|\nabla X^1|^2 + |\nabla X^2|^2 + |\nabla X^3|^2)$$

- Example 2: 2d image as manifold, $(h_{ij}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \beta^2 \end{pmatrix}$

- Euler-Lagrange equations

$$-\frac{1}{2\sqrt{g}} h^{il} \frac{\delta S}{\delta X^l} = \frac{1}{\sqrt{g}} \partial_\mu (\sqrt{g} g^{\mu\nu} \partial_\nu X^i) + \Gamma^i_{jk} \partial_\mu X^j \partial_\nu X^k g^{\mu\nu}$$

with Christoffel symbols

$$\Gamma^i_{jk} = \tfrac{1}{2} h^{il} (\partial_j h_{kl} + \partial_k h_{jl} - \partial_l h_{jk})$$

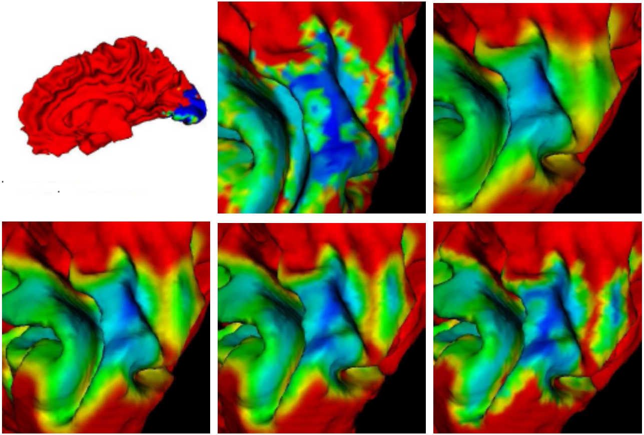
Beltrami Framework (7)

Beltrami Flow for Images on Manifolds

- The Beltrami framework can also be used to establish edge-preserving smoothing procedures for images painted on manifolds (e.g., surfaces with texture)
- Consider surface $\mu : D \to \mathbb{R}^3$, $D \subset \mathbb{R}^2$, and multi-channel image $U : \mu(D) \to \mathbb{R}^k$ on the surface
- Construct new surface $\mathbf{X} : D \to \mathbb{R}^{k+3}$ by
$$\mathbf{X}(\sigma_1, \sigma_2) = (\mu_1, \mu_2, \mu_3, U_1 \circ \mu, \ldots, U_k \circ \mu)^{\mathrm{T}}$$
- Compute intrinsic diffusion flow for $\mathbf{X}$
- Projection to components $U_1, \ldots, U_k$ gives Beltrami flow
- Different projections possible:
  - Project to components $U_1, \ldots, U_k$: Smooth image data only
  - Project to components μ_1, μ_2, μ_3: Smooth manifold only
  - Use all components (no projection) – smoothes manifold and image data

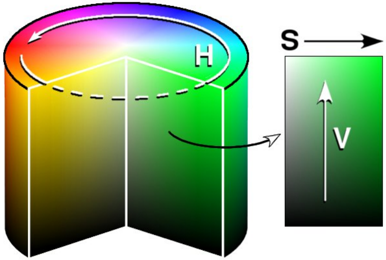
Whether smoothing is performed for the image (texture) or manifold (surface) or both, the smoothing process is always controlled using both pieces of information

Beltrami Flow for Images on Manifolds – Example



Comparison of Beltrami flow and nonlinear surface diffusion. **Top, left to right:** Original noisy image on left cortex surface – detail from original image – Beltrami flow, $\beta = 0$ (isotropic diffusion); **Bottom, left to right:** Beltrami flow, $\beta = 0.1$ – $\beta = 0.5$ – anisotropic diffusion *(N. Sochen, R. Deriche, L. Lopez Perez 2003)*

HSV color space



Source: http://de.wikipedia.org/wiki/HSV-Farbraum

- hue: red: 0° ; green: 120° ; blue: 240°
- saturation: gives distance of the colour to the nearest grey tone
- value: defines how dark or bright a colour is

Beltrami operator on S^1

- S^1 can be described by (U,V) with $U^2+V^2=1$
- as manifold, two charts are needed:
  - on $S^1 - \{(\pm 1, 0)\}$

$$ds^2_{S^1} = dU^2 + dV^2 dU^2 + (d(\sqrt{1-U^2}))^2 = \frac{1}{1-U^2} dU^2$$

  - on $S^1 - \{(0, \pm 1)\}$

$$ds^2_{S^1} = \frac{1}{1-V^2} dV^2$$

Diffusion on $\mathbb{R}^4 \times S^1$

- image $(x, y) \mapsto (x, y, H(x, y), S(x, y), V(x, y))$
- define $U = \cos(H),\ W = \sin(H)$
- we use

$$(h_{ij}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & A(U) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

with $A(U) = \frac{1}{1-U^2}$ (similar for W)

- induced metric:

$$(g_{\mu\nu}) = \begin{pmatrix} 1 + A(U)U_x^2 + S_x^2 + V_x^2 & A(U)U_xU_y + S_xS_y + V_xV_y \\ A(U)U_xU_y + S_xS_y + V_xV_y & 1 + A(U)U_y^2 + S_y^2 + V_y^2 \end{pmatrix}$$

- Christoffel symbols: the only nonvanishing term uses Γ_{33}^3

$$\Gamma_{33}^3 = \frac{U}{1-U^2} = U\,h_{33}$$

Beltrami Framework (12)

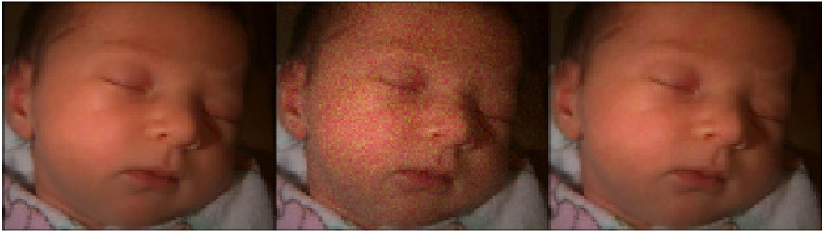
- resulting flow:

$$\begin{aligned} U_t &= \Delta_g U + 2U - U(g^{11} + g^{22}) \\ W_t &= \Delta_g W + 2W - W(g^{11} + g^{22}) \\ S_t &= \Delta_g S \\ V_t &= \Delta_g V \end{aligned}$$

- implementation: compute both U and W simultaneously, use

$$\begin{cases} (U, \operatorname{sgn}(W)\sqrt{1-U^2}) & U^2 \leq W^2 \\ (\operatorname{sgn}(U)\sqrt{1-W^2}, W) & U^2 \geq W^2 \end{cases}$$

Example



Left: original, **middle**: noisy **right**: HSV Beltrami flow

Segmentation Problem

- Assume an image $f : D \to \mathbb{R}$, $D \subset \mathbb{R}^2$ is given
- **Problem:** Find an object in this image
- **Assumption:** Object is a region delineated by a contour of sufficient contrast
- **Interactive proceeding:**
  - Initialise curve with coarse data specified by the user
  - Fit automatically to the precise object contour

Curve Evolution

- Construct curve flow $c(p, t)$ initialised by user-specified contour $c(t = 0) = c_0$
- Modify curvature motion $c_t = \kappa\vec{n}$ by additional **edge-stopping function** $G(f)$ dependent on the given image f:

$$c_t = G(f)\,\kappa\,\vec{n} - \langle \nabla G, \vec{n} \rangle \vec{n}$$

- Typical choice for edge-stopping function:

$$G(f) = g(\|\nabla f\|^2)$$

g: nonnegative decreasing function, e.g. Perona-Malik

$$g(z^2) = \frac{1}{1 + z^2/\lambda^2}$$

Level Set Evolution

- Equivalent level set evolution

$$u_t = G(f)\,\|\nabla u\|\;\kappa - \langle \nabla G(f), \nabla u \rangle$$
$$= -\,\|\nabla u\|\,\mathrm{div}\left(G(f)\,\frac{\nabla u}{\|\nabla u\|} \right)$$

- Initialisation: $u(t=0) = u_0$, e.g. signed distance function for user-specified initial contour c_0
- Edge-stopping function enters diffusivity in a way similar to nonlinear isotropic diffusion
- *Important difference:* G depends on given image f, not on evolving u!
- This process is called (Euclidean) **geodesic active contour evolution**

Variational Interpretation

- Geodesic active contour evolution of a closed curve c is a gradient descent for

$$E_G[c] := \oint_c G(f(c(s)))\,\mathrm{d}s$$

- E_G can be interpreted as **arc-length**

$$E_G[c] = \oint_c \sqrt{g_{(c(s))}(c_s, c_s)}\,\mathrm{d}s$$

  of c w.r.t. a Riemannian metric on D different from the standard Euclidean metric

- New metric is given by the matrix

$$g_{(x,y)} = G(f) \cdot \mathbb{1}$$

- Any non-trivial steady state of the evolution must therefore be a *geodesic* in the image-dependent metric (hence the name "geodesic active contours")

Existence of Solutions

- Consider geodesic active contour evolution in level-set formulation
- In a suitable function space, there exists a unique solution $u(t=T)$ for given u_0 and every $T > 0$
- Solution satisfies maximum-minimum principle

$$\inf(u_0) \leq u(x, y, T) \leq \sup(u_0)$$

- Solution is stable w.r.t. the initial conditions:

$$\|u(\cdot, T) - v(\cdot, T)\|_\infty \leq \|u_0 - v_0\|_\infty$$

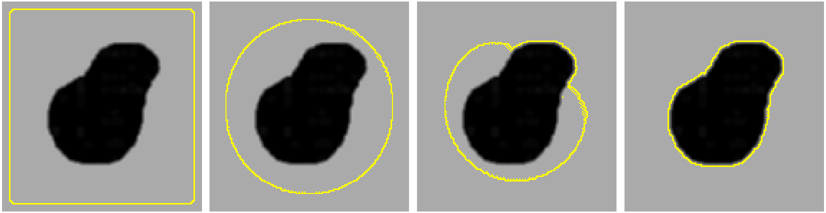
for all $T > 0$ and initial functions u_0, v_0

Euclidean Geodesic Active Contours

Properties

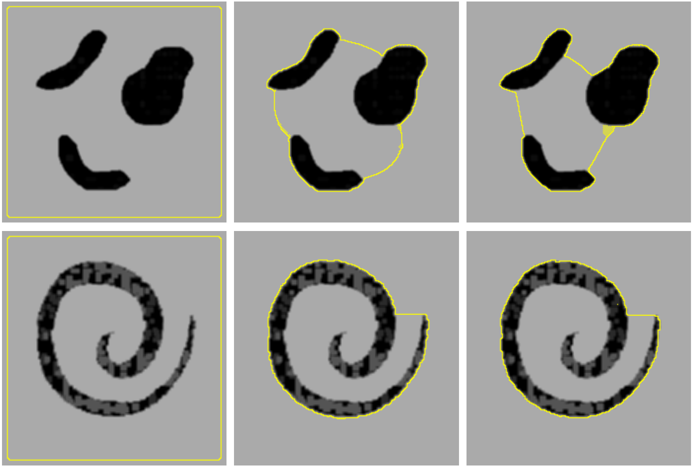
- In homogeneous regions, active contour evolution behaves similar to curvature motion (moving mainly inward)
- Evolution stops at high gradients
- Initial contour should be at least to a considerable part outside the sought object to warrant detection

Example



Feature extraction by active contours. **Left to right:** Synthetic image with initial contour; active contour at evolution times $T = 1000$, $T = 1500$, $T = 2000$

Problematic Examples



Euclidean active contours applied to two synthetic images. **Each row, left to right:** Synthetic image with initial contour – active contour evolution at time $T = 5000$ – same at $T = 20000$

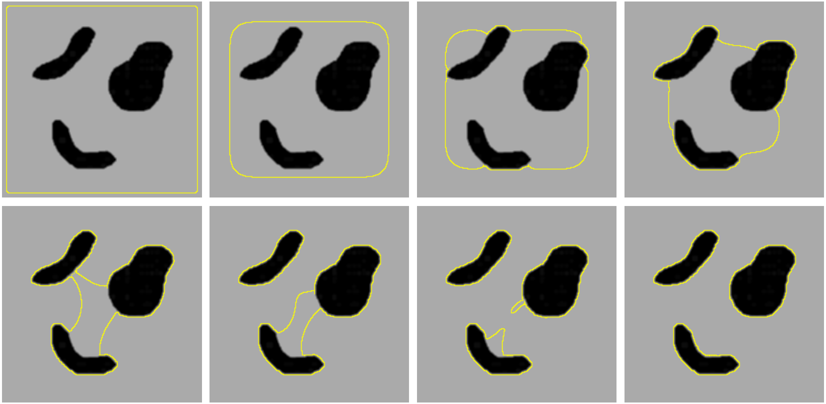
Modification of the Model

- Introduce an additional curve-shrinking term ($\nu > 0$)

$$u_t = G(f)\,\|\nabla u\|\,(\kappa - \nu) - \langle \nabla G(f), \nabla u \rangle$$
$$= -\,\|\nabla u\|\,\mathrm{div}\left(G(f)\frac{\nabla u}{\|\nabla u\|}\right) - \nu G(f)\,\|\nabla u\|$$

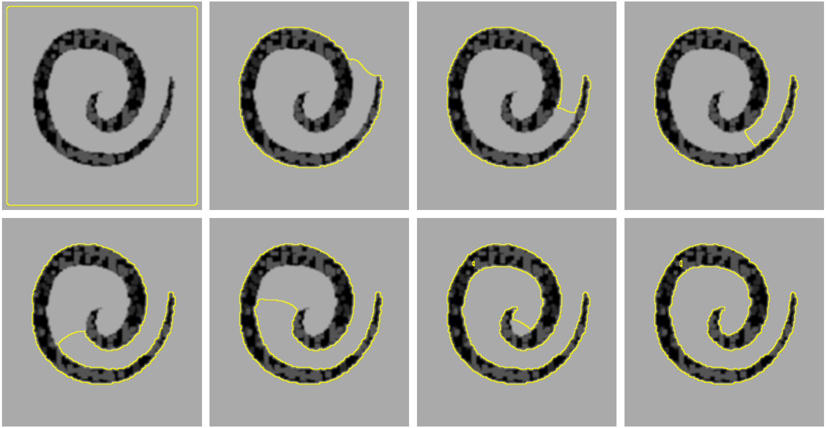
- *Effect:* Constant shrinking force acting on the contour
  - ⊕ Shrinkage of curve towards object contour is speeded up
  - ⊕ Evolution is not as likely to be trapped in false optima
  - ⊖ Additional parameter ν
  - ⊖ Contour tends to shrink further after object detection – stopping time needed

Examples for the Modified Model



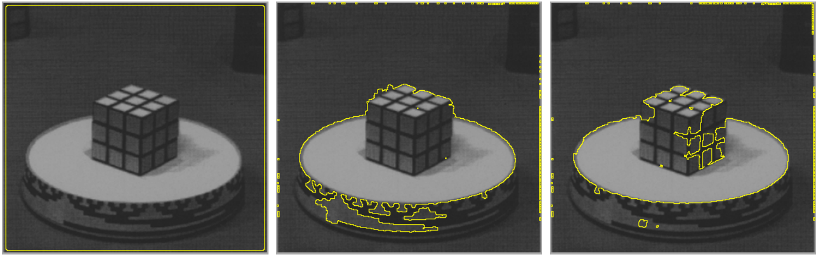
Feature extraction by active contours. **Top, left to right:** Synthetic image with initial contour; active contour with $\nu = 0.1$ at evolution times $T = 200$, $T = 300$, $T = 500$. **Bottom, left to right:** $T = 1000$, $T = 1100$, $T = 1200$, $T = 1300$ *(after Kichenassamy et al. 1996)*

Examples for the Modified Model



Feature extraction by active contours. **Top, left to right:** Synthetic image with initial contour; active contour with $\nu = 0.1$ at evolution times $T = 500$, $T = 1000$, $T = 1500$. **Bottom, left to right:** $T = 2500$, $T = 3000$, $T = 4000$, $T = 4000$ *(after Kichenassamy et al. 1996)*

Examples for the Modified Model



Feature extraction by active contours. **Left to right:** Photograph of Rubik cube on a plate with initial contour – contour evolution ($\nu = 1$) at times $T = 1400$, $T = 2000$ *(after Kichenassamy et al. 1996)*

References
Beltrami Flow

- A. I. El-Fallah, G. E. Ford: Mean curvature evolution and surface area scaling in image filtering. *IEEE Transactions in Image Processing,* 6(5):750–753, 1997
- N. Sochen, R. Kimmel and R. Malladi: A General Framework for Low Level Vision, *IEEE Trans. in Image Processing*, Special Issue on Geometry Driven Diffusion, 7 (1998) pp. 310-318.
- R. Kimmel, R. Malladi, N. Sochen: Images as embedding maps and minimal surfaces: Movies, color, and volumetric medical images. *Computer Vision and Pattern Recognition*, Proceedings, 1997
- A. Yezzi Jr.: Modified curvature motion for image smoothing and enhancement. *IEEE Transactions in Image Processing,* 7(3):345–352, 1998
- R. Kimmel, R. Malladi, N. Sochen: Image processing via the Beltrami operator. *Proc. 3rd Asian Conference on Computer Vision*, Hongkong, 1998
- N. Sochen, R. Deriche, L. Lopez Perez: *The Beltrami Flow over Manifolds.* Preprint no. 4897, INRIA, Sophia Antipolis 2003
- A. Spira, N. Sochen, R. Kimmel: A short time Beltrami kernel for smoothing images and manifolds. *IEEE Transactions in Image Processing,* 16(6):1628–1636, 2007

References
Geodesic Active Contours

- V. Caselles, R. Kimmel, G. Sapiro: Geodesic active contours. *International Journal of Computer Vision* 22(1):61–79, 1997
- S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi: Conformal curvature flows: from phase transitions to active vision. *Arch. Rational Mechanics and Analysis* 134:275–301, 1996
- R. Malladi, J. Sethian, B. Vemuri: Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 17(2):158–175, 1995
- P. Olver, G. Sapiro, A. Tannenbaum: Affine invariant detection: edge maps, anisotropic diffusion, and active contours. *Acta Applicandae Mathematicae* 59:45–77, 1999 19(4):394–398, 1997