BIL 717
Image Processing

Mar. 7, 2016

Active Contours/Snakes
Variational Segmentation Models

Erkut Erdem
Hacettepe University
Computer Vision Lab (HUCVL)

Acknowledgement: The slides on active contours are adapted from the slides prepared by K. Grauman of University of Texas at Austin

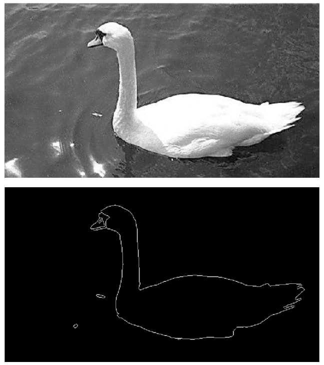
Today

- Active Contours
- Variational Segmentation Models

Today

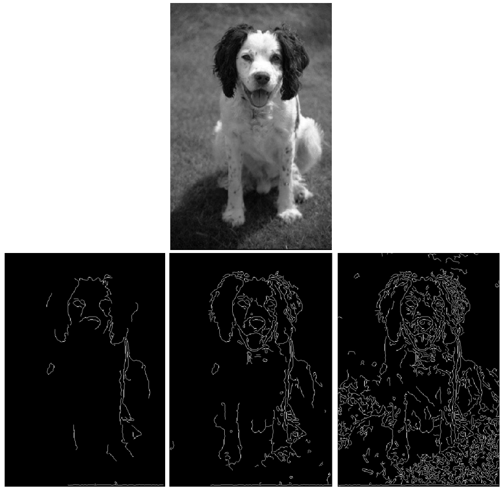
- Active Contours
- Variational Segmentation Models

Fitting: Edges vs. boundaries



Edges useful signal to indicate occluding boundaries, shape.

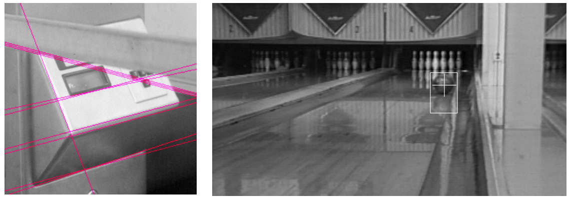
Here the raw edge output is not so bad…



…but quite often boundaries of interest are fragmented, and we have extra “clutter” edge points.

Images from D. Jacobs

Fitting: Edges vs. boundaries



Given a model of interest, we can overcome some of the missing and noisy edges using **fitting** techniques.

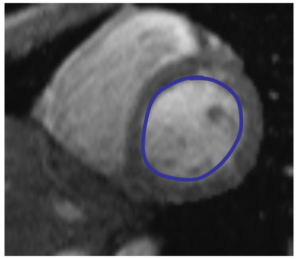
With voting methods like the **Hough transform**, detected points vote on possible model parameters.

Deformable contours

a.k.a. active contours, snakes

Given: initial contour (model) near desired object

Goal: evolve the contour to fit exact object boundary



Main idea: elastic band is iteratively adjusted so as to

- be near image positions with high gradients, and
- satisfy shape "preferences" or contour priors

[Snakes: Active contour models, Kass, Witkin, & Terzopoulos, ICCV1987]

Figure credit: Yuri Boykov

Deformable contours: intuition

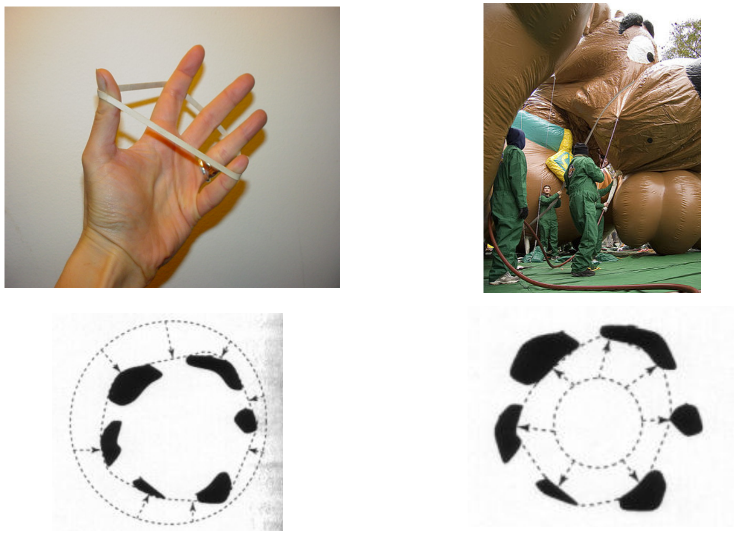
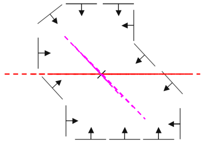
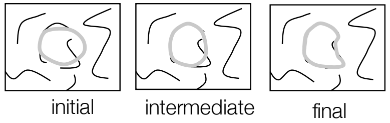


Image from http://www.healthline.com/blogs/exercise_fitness/uploaded_images/HandBand2-795868.JPG

Deformable contours vs. Hough

Like generalized Hough transform, useful for shape fitting; but





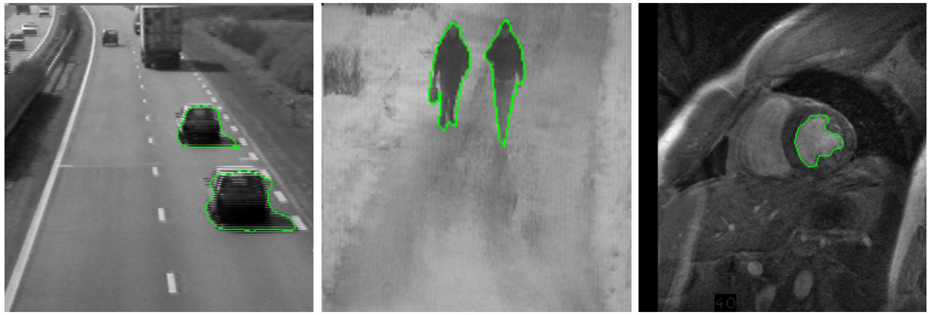
Hough
Rigid model shape
Single voting pass can detect multiple instances

Deformable contours
Prior on shape types, but shape iteratively adjusted (deforms)
Requires initialization nearby
One optimization "pass" to fit a single contour

Why do we want to fit deformable shapes?

- Some objects have similar basic form but some variety in the contour shape.

Why do we want to fit deformable shapes?



- Non-rigid, deformable objects can change their shape over time.

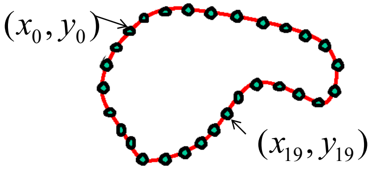
Figure credit: Julien Jomier

Aspects we need to consider

- Representation of the contours
- Defining the energy functions
  - External
  - Internal
- Minimizing the energy function

Representation

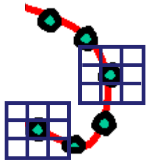
- We'll consider a discrete representation of the contour, consisting of a list of 2d point positions ("vertices").



$$v_i = (x_i, y_i),$$

$$\text{for} \quad i = 0, 1, \ldots, n-1$$

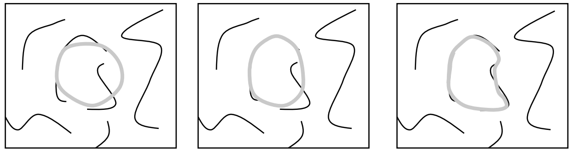
- At each iteration, we'll have the option to move each vertex to another nearby location ("state").



Fitting deformable contours

How should we adjust the current contour to form the new contour at each iteration?

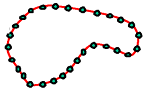
- Define a cost function (“energy” function) that says how good a candidate configuration is.
- Seek next configuration that minimizes that cost function.



initial intermediate final

Energy function

The total energy (cost) of the current snake is defined as:



$$E_{total} = E_{internal} + E_{external}$$

Internal energy: encourage prior shape preferences: e.g., smoothness, elasticity, particular known shape.

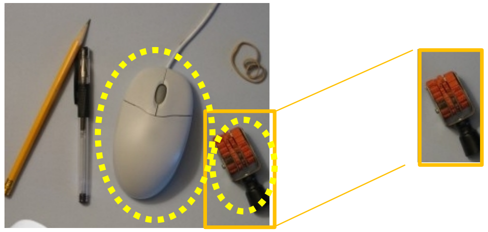
External energy (“image” energy): encourage contour to fit on places where image structures exist, e.g., edges.

A good fit between the current deformable contour and the target shape in the image will yield a low value for this cost function.

External energy: intuition

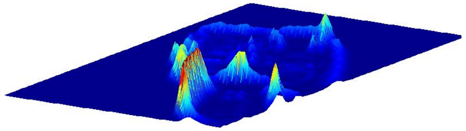
- Measure how well the curve matches the image data
- “Attract” the curve toward different image features
  - Edges, lines, texture gradient, etc.

External image energy



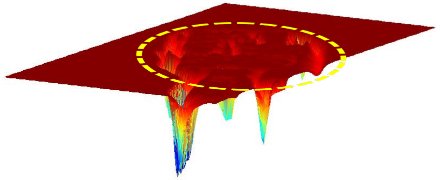
How do edges affect “snap” of rubber band?

Think of external energy from image as gravitational pull towards areas of high contrast



Magnitude of gradient

$$G_x(I)^2 + G_y(I)^2$$

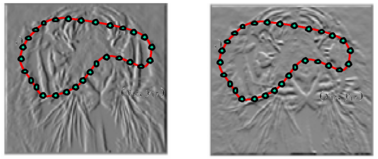


- (Magnitude of gradient)

$$-\left(G_x(I)^2 + G_y(I)^2\right)$$

External image energy

- Gradient images $G_x(x,y)$ and $G_y(x,y)$



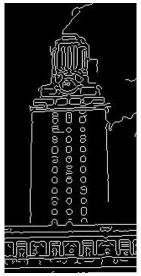
- External energy at a point on the curve is:

$$E_{external}(\nu) = -(|G_x(\nu)|^2 + |G_y(\nu)|^2)$$

- External energy for the whole curve:

$$E_{external} = -\sum_{i=0}^{n-1} |G_x(x_i,y_i)|^2 + |G_y(x_i,y_i)|^2$$

Internal energy: intuition

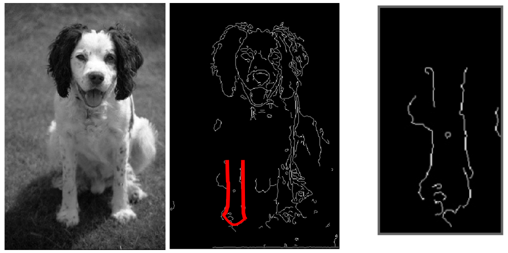


What are the underlying boundaries in this fragmented edge image?

And in this one?

Internal energy: intuition

A priori, we want to favor **smooth** shapes, contours with **low curvature**, contours similar to a **known shape**, etc. to balance what is actually observed (i.e., in the gradient image).



Internal energy

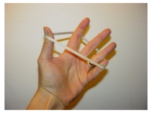
For a *continuous* curve, a common internal energy term is the "bending energy".

At some point *v(s)* on the curve, this is:

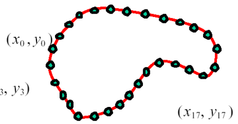
$$E_{internal}(\nu(s)) = \boxed{\alpha \left|\frac{d\nu}{ds}\right|^2} + \boxed{\beta \left|\frac{d^2\nu}{d^2s}\right|^2}$$

Tension, Elasticity

Stiffness, Curvature



Internal energy



- For our discrete representation,

$$\nu_i = (x_i, y_i) \qquad i = 0 \;\ldots .\; n-1$$

$$\frac{d\nu}{ds} \approx \nu_{i+1} - \nu_i \qquad \frac{d^2\nu}{ds^2} \approx (\nu_{i+1} - \nu_i) - (\nu_i - \nu_{i-1}) = \nu_{i+1} - 2\nu_i + \nu_{i-1}$$

Note these are derivatives relative to position---not spatial image gradients.

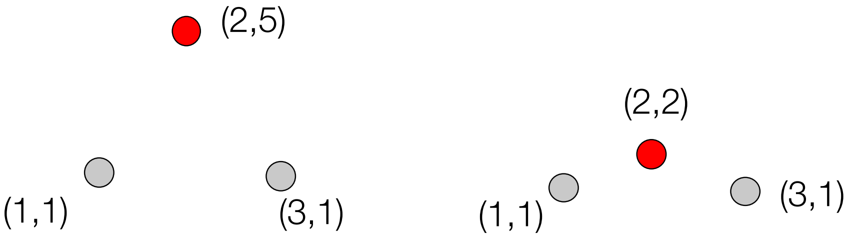
- Internal energy for the whole curve:

$$E_{internal} = \sum_{i=0}^{n-1} \alpha \left\|\nu_{i+1} - \nu_i\right\|^2 + \beta \left\|\nu_{i+1} - 2\nu_i + \nu_{i-1}\right\|^2$$

Why do these reflect tension and curvature?

Example: compare curvature

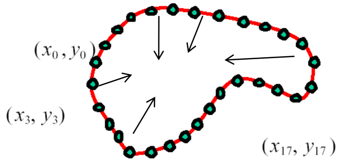
$$E_{curvature}(\nu_i) = \left\|\nu_{i+1} - 2\nu_i + \nu_{i-1}\right\|^2$$
$$= (x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2$$



Penalizing elasticity

- Current elastic energy definition uses a discrete estimate of the derivative:

$$E_{elastic} = \sum_{i=0}^{n-1} \alpha \left\|\nu_{i+1} - \nu_i\right\|^2$$
$$= \alpha \cdot \sum_{i=0}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2$$



What is the possible problem with this definition?

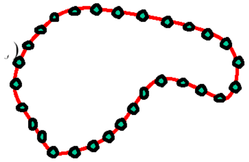
Penalizing elasticity

- Current elastic energy definition uses a discrete estimate of the derivative:

$$E_{elastic} \cancel{= \sum_{i=0}^{n-1} \alpha \left\|\nu_{i+1} - \nu_i\right\|^2}$$

Instead:

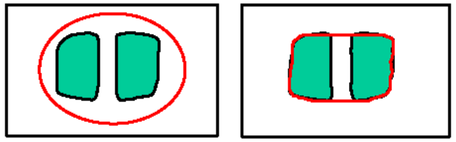
$$= \alpha \cdot \sum_{i=0}^{n-1} \left((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 - \overline{d}\right)^2$$

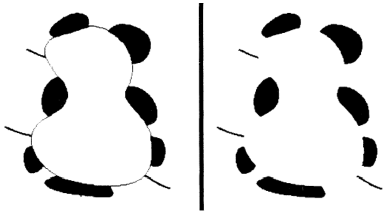


where d is the average distance between pairs of points – updated at each iteration.

Dealing with missing data

- The preferences for low-curvature, smoothness help deal with missing data:



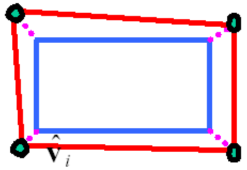


Illusory contours found!

[Figure from Kass et al. 1987]

Extending the internal energy: capture shape prior

- If object is some smooth variation on a known shape, we can use a term that will penalize deviation from that shape:



$$E_{internal} += \alpha \cdot \sum_{i=0}^{n-1} (v_i - \hat{v}_i)^2$$

where $\{\hat{v}_i\}$ are the points of the known shape.

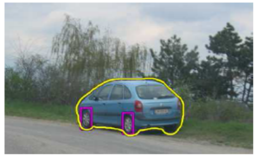


Fig from Y. Boykov

Total energy: function of the weights

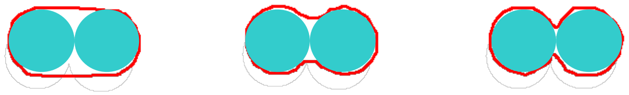
$$E_{total} = E_{internal} + \gamma E_{external}$$

$$E_{external} = -\sum_{i=0}^{n-1} |G_x(x_i, y_i)|^2 + |G_y(x_i, y_i)|^2$$

$$E_{internal} = \sum_{i=0}^{n-1} \alpha \left(\bar{d} - \|v_{i+1} - v_i\|\right)^2 + \beta \|v_{i+1} - 2v_i + v_{i-1}\|^2$$

Total energy: function of the weights

- e.g., α weight controls the penalty for internal elasticity

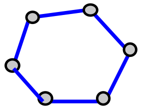


large α medium α small α

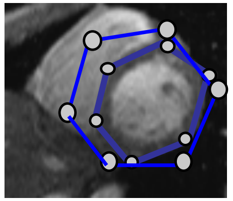
Fig from Y. Boykov

Recap: deformable contour

- A simple elastic snake is defined by:
  - A set of *n* points,
  - An internal energy term (tension, bending, plus optional shape prior)
  - An external energy term (gradient-based)



- To use to segment an object:
  - Initialize in the vicinity of the object
  - Modify the points to minimize the total energy

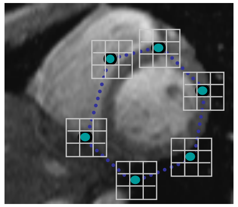


Energy minimization

- Several algorithms have been proposed to fit deformable contours:
  - Greedy search
  - Dynamic programming (for 2d snakes)
  - etc.

Energy minimization: greedy

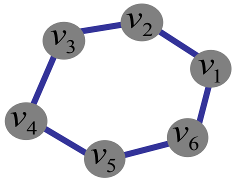
- For each point, search window around it and move to where energy function is minimal
  - Typical window size, e.g., 5 x 5 pixels

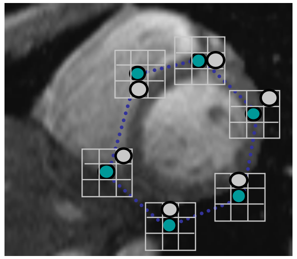


- Stop when predefined number of points have not changed in last iteration, or after max number of iterations

- Note:
  - Convergence not guaranteed
  - Need decent initialization

Energy minimization: dynamic programming





With this form of the energy function, we can minimize using dynamic programming, with the Viterbi algorithm.

Iterate until optimal position for each point is the center of the box, i.e., the snake is optimal in the local search space constrained by boxes.

Fig from Y. Boykov
[Amini, Weymouth, Jain, 1990]

Energy minimization: dynamic programming

- Possible because snake energy can be rewritten as a sum of pair-wise interaction potentials:

$$E_{total}(v_1,\ldots,v_n) = \sum_{i=1}^{n-1} E_i(v_i, v_{i+1})$$

- Or sum of triple-interaction potentials.

$$E_{total}(v_1,\ldots,v_n) = \sum_{i=1}^{n-1} E_i(v_{i-1}, v_i, v_{i+1})$$

Snake energy: pair-wise interactions

$$E_{total}(x_1,\ldots,x_n,y_1,\ldots,y_n) = -\sum_{i=1}^{n-1} |G_x(x_i,y_i)|^2 + |G_y(x_i,y_i)|^2$$

$$+ \quad \alpha \cdot \sum_{i=1}^{n-1} (x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2$$

Re-writing the above with $v_i = (x_i, y_i)$:

$$E_{total}(v_1,\ldots,v_n) = -\sum_{i=1}^{n-1} \| G(v_i) \|^2 + \alpha \cdot \sum_{i=1}^{n-1} \| v_{i+1} - v_i \|^2$$

$$E_{total}(v_1,\ldots,v_n) = E_1(v_1,v_2) + E_2(v_2,v_3) + \ldots + E_{n-1}(v_{n-1},v_n)$$

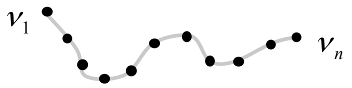
where $E_i(v_i, v_{i+1}) = -\| G(v_i) \|^2 + \alpha \| v_{i+1} - v_i \|^2$

Kristen Grauman

Energy minimization: dynamic programming

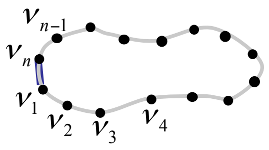
DP can be applied to optimize an open ended snake

$$E_1(v_1,v_2) + E_2(v_2,v_3) + \ldots + E_{n-1}(v_{n-1},v_n)$$



For a closed snake, a "loop" is introduced into the total energy.

$$E_1(v_1,v_2) + E_2(v_2,v_3) + \ldots + E_{n-1}(v_{n-1},v_n) + \boxed{E_n(v_n,v_1)}$$



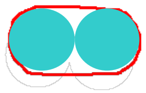
Work around:

1) Fix v_1 and solve for rest .

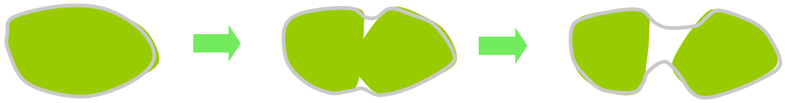
2) Fix an intermediate

Limitations

- May over-smooth the boundary



- Cannot follow topological changes of objects



Limitations

- External energy: snake does not really “see” object boundaries in the image unless it gets very close to it.

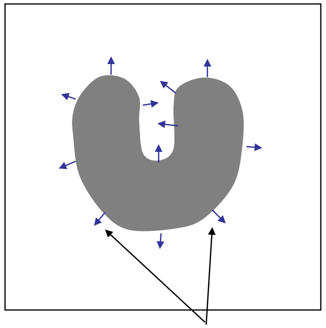
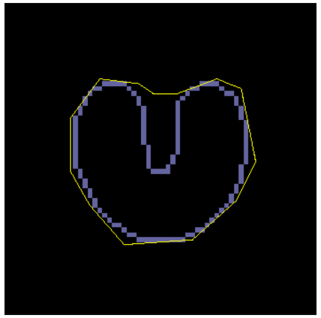
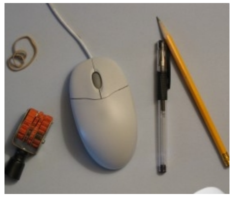


image gradients ∇I
are large only directly on the boundary

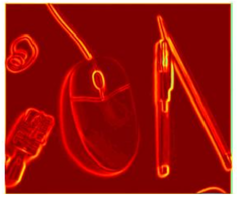


Distance transform

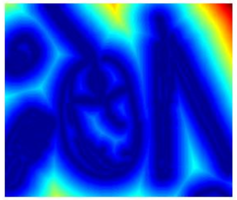
- External image can instead be taken from the **distance transform** of the edge image.



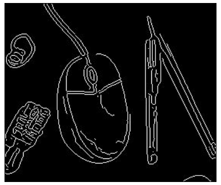
original



-gradient



distance transform



edges

Value at (x,y) tells how far that position is from the nearest edge point (or other binary mage structure)

```
>> help bwdist
```

Deformable contours: pros and cons

Pros:

- Useful to track and fit non-rigid shapes
- Contour remains connected
- Possible to fill in “subjective” contours
- Flexibility in how energy function is defined, weighted.

Cons:

- Must have decent initialization near true boundary, may get stuck in local minimum
- Parameters of energy function must be set well based on prior information

Summary

- Deformable shapes and active contours are useful for
  - Segmentation: fit or “snap” to boundary in image
  - Tracking: previous frame’s estimate serves to initialize the next
- Fitting active contours:
  - Define terms to encourage certain shapes, smoothness, low curvature, push/pulls, …
  - Use weights to control relative influence of each component cost
  - Can optimize 2d snakes with Viterbi algorithm.
- Image structure (esp. gradients) can act as attraction force for *interactive* segmentation methods.

Today

- Active Contours
- Variational Segmentation Models

Review – Nonlinear Diffusion

- use nonlinear PDEs to create a scale space representation
  - consists of gradually simplified images
  - some image features such as edges are maintained or even enhanced.
- Perona-Malik Type Nonlinear Diffusion (1990)
- Total Variation (TV) Regularization (1992)
- Weickert's Edge Enhancing Diffusion (1994)

Review - Perona-Malik Type Nonlinear Diffusion

- Perona-Malik equation is:

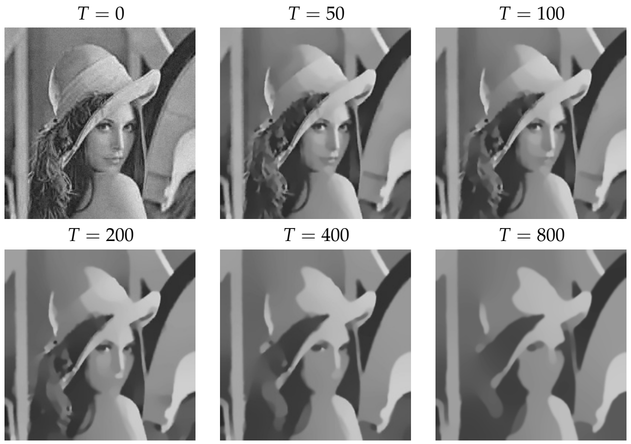
$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u)$$

  with homogeneous Neumann boundary conditions and the initial condition *u*0(*x*) = *f* (*x*), *f* denoting the input image.

- Constant diffusion coefficient of linear equation is replaced with a smooth non-increasing diffusivity function *g* satisfying
  - $g(0) = 1$,
  - $g(s) \geq 0$,
  - $\lim_{s\to\infty} g(s) = 0$
- <u>The diffusivities become variable in both space and time</u>.

Review - Perona-Malik Type Nonlinear Diffusion

- Smoothing process diminishes noise while retaining or enhancing edges


$T = 0$ $T = 50$ $T = 100$

$T = 200$ $T = 400$ $T = 800$

$(\lambda = 1, \sigma = 1)$

Review - Total Variation (TV) Regularization

- Rudin et al. (1992) formulated image restoration as minimization of the total variation (TV) of a given image under certain assumptions on the noise.
- Total Variation (TV) regularization model is generally defined as:

$$E_{TV}(u) = \int_{\Omega} \left(\frac{1}{2}(u-f)^2 + \alpha |\nabla u| \right) dx$$

  - $\Omega \subset \mathbf{R}^2$ is connected, bounded, open subset representing the image domain,
  - f is an image defined on Ω,
  - u is the smooth approximation of f ,
  - $\alpha > 0$ is a scalar.

Review - Total Variation (TV) Regularization

- Total Variation (TV) regularization model:

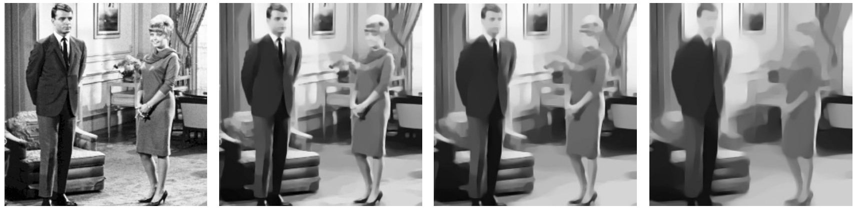
$$E_{TV}(u) = \int_{\Omega} \left(\frac{1}{2}(u-f)^2 + \alpha |\nabla u| \right) dx$$

- The gradient descent equation for Equation (10) is defined by:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \frac{1}{\alpha}(u-f); \quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = 0$$

- The value of α specifies the relative importance of the fidelity term.
- It can be interpreted as a scale parameter that determines the level of smoothing.

Review - TV Restoration results



$\alpha = 50$ $\alpha = 100$ $\alpha = 200$

- The value of α specifies the relative importance of the fidelity term and thus the level of smoothing.

Review - TV Regularization and TV Flow

- TV regularization can be associated with a nonlinear diffusion filter, the so-called *TV flow*
- Ignoring the fidelity term in the TV regularization model leads to the PDE:

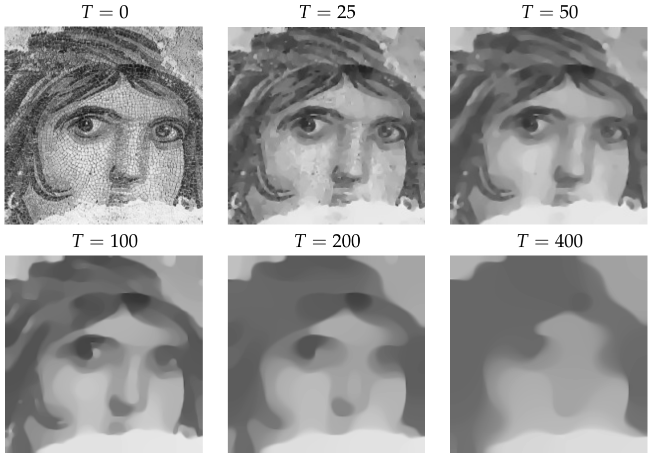
$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u)$$

with $u^0 = f$ and the diffusivity function $g(|\nabla u|) = \frac{1}{|\nabla u|}$

- Notice that this diffusivity function has no additional contrast parameter as compared with the Perona-Malik diffusivities.

Review - Sample TV Flow results

- Corresponding smoothing process yields segmentation-like, piecewise constant images.



$T=0$ $T=25$ $T=50$

$T=100$ $T=200$ $T=400$

Review - Edge Enhancing Diffusion

- Proposed by Weickert (1994)
- an anisotropic nonlinear diffusion model with better edge enhancing capabilities than the Perona-Malik model
- can be described by the equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(\nabla u)\nabla u)$$

where

- *u* is the smoothed image,
- *f* is the input image ($u^0(x) = f(x)$),
- *D* represents a matrix-valued diffusion tensor that describes the smoothing directions and the corresponding diffusivities

Review - Edge Enhancing Diffusion

- Suggested eigenvalues are

$$\begin{aligned} \lambda_1(|\nabla u_\sigma|) &= \begin{cases} 1 & \text{if } |\nabla u_\sigma| = 0 \\ 1 - exp\left(-\frac{3.31488}{(|\nabla u_\sigma|/\lambda)^8}\right) & \text{otherwise,} \end{cases} \\ \lambda_2(|\nabla u_\sigma|) &= 1 \end{aligned}$$

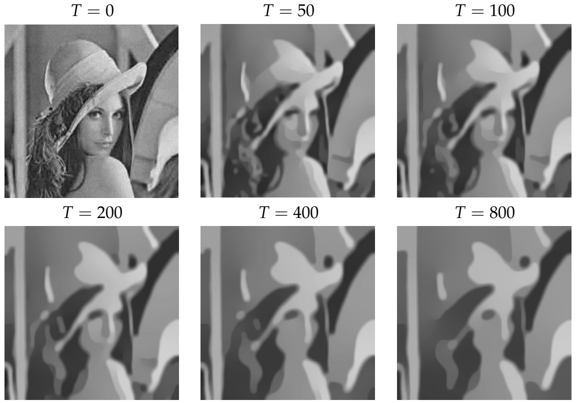
where λ denotes the contrast parameter.

- preserves and enhances image edges by reducing the diffusivity λ_1 perpendicular to edges for sufficiently large values of $|\nabla u_\sigma|$.
- Specifically, the diffusion tensor is given by the formula:

$$D = \begin{bmatrix} (u_\sigma)_x & -(u_\sigma)_y \\ (u_\sigma)_y & (u_\sigma)_x \end{bmatrix} \cdot \begin{bmatrix} \lambda_1(|\nabla u_\sigma|) & 0 \\ 0 & \lambda_2(|\nabla u_\sigma|) \end{bmatrix} \cdot \begin{bmatrix} (u_\sigma)_x & -(u_\sigma)_y \\ (u_\sigma)_y & (u_\sigma)_x \end{bmatrix}^{-1}$$

Review - Sample Results of Edge Enhancing Diffusion

- Smoothing process diminishes noise and fine image details while retaining and enhancing edges as in the Perona-Malik type nonlinear diffusion.



$T=0$ $T=50$ $T=100$

$T=200$ $T=400$ $T=800$

$(\lambda = 2, \sigma = 1)$

Variational Segmentation Models

- Segmentation is formalized as a functional minimization.

- Mumford-Shah Model (1989)
- Ambrosio-Tortorelli Model (1990)
- Shah's Model (1996)
- Chan-Vese Model (2001)

- Context-guided Mumford-Shah Model (2009)

Mumford-Shah (MS) Segmentation Model

- Mumford & Shah, Comm. Pure Appl. Math., 1989
- Segmentation is formalized as a functional minimization: Given an image ***f***, compute a piecewise smooth image ***u*** and an edge set Γ

$$E_{MS}(u,\Gamma) = \beta \int\limits_{\Omega} (u-f)^2 dx + \alpha \int\limits_{\Omega\backslash\Gamma} |\nabla u|^2 dx + length(\Gamma)$$

  - $\Omega \subset \mathbf{R}^2$ is connected, bounded, open subset representing the image domain,
  - *f* is an image defined on Ω,
  - $\Gamma \subset \Omega$ is the edge set segmenting Ω,
  - *u* is the piecewise smooth approximation of *f* ,
  - $\alpha, \beta > 0$ are the scale space parameters.

Mumford-Shah (MS) Segmentation Model

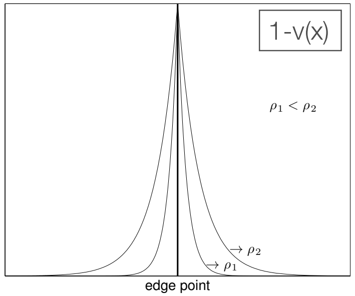
$$E_{MS}(u,\Gamma) = \underbrace{\beta \int\limits_{\Omega} (u-f)^2 dx}_{\text{data fidelity term}} + \underbrace{\alpha \int\limits_{\Omega\backslash\Gamma} |\nabla u|^2 dx + length(\Gamma)}_{\text{regularization or smoothness term}}$$

- Smoothing and edge detection processes work jointly to partition an image into segments.
- Unknown edge set Γ of a lower dimension makes the minimization of the MS model very difficult.
- In literature several approaches for approximating the MS model are suggested.

Ambrosio-Tortorelli (AT) Approximation

$$E_{AT}(u,v) = \int\limits_{\Omega} \left(\beta(u-f)^2 + \alpha(v^2|\nabla u|^2) + \underbrace{\frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1-v)^2}{\rho}\right)}_{\text{length}(\Gamma)} \right) dx$$

- Unknown edge set Γis replaced with a continuous function v(x)
  - v ≈ 0 along image edges
  - v grows rapidly towards 1 away from edges
- The function *v* can be interpreted as a blurred version of the edge set.
- The parameter ρ specifies the level of blurring.



Ambrosio-Tortorelli (AT) Approximation: u and v processes

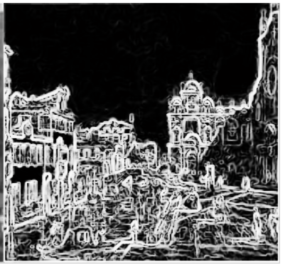
- Piecewise smooth image *u* and the edge strength function *v* are simultaneously computed via the solution of the following system of coupled PDEs:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \nabla \cdot (v^2 \nabla u) - \frac{\beta}{\alpha}(u - f); \quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = 0 \\ \frac{\partial v}{\partial t} &= \nabla^2 v - \frac{2\alpha|\nabla u|^2 v}{\rho} - \frac{(v-1)}{\rho^2}; \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0\end{aligned}$$

Ambrosio-Tortorelli (AT) Approximation: u and v processes

f: raw image

u: smooth image



v: edge strength function

Ambrosio-Tortorelli (AT) Approximation: u and v processes

- Piecewise smooth image *u* and the edge strength function *v* are simultaneously computed via the solution of the following system of coupled PDEs:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \nabla \cdot (v^2 \nabla u) - \frac{\beta}{\alpha}(u - f); \quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = 0 \\ \frac{\partial v}{\partial t} &= \nabla^2 v - \frac{2\alpha|\nabla u|^2 v}{\rho} - \frac{(v-1)}{\rho^2}; \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0\end{aligned}$$

- PDE for each variable can be interpreted as a biased diffusion equation that minimizes a convex quadratic functional in which the other variable is kept fixed.

Ambrosio-Tortorelli (AT) Approximation: u process

- Keeping *v* fixed, PDE for the process *u* minimizes the following convex quadratic functional:

$$\int_{\Omega} \left(\alpha v^2 |\nabla u|^2 + \beta (u - f)^2 \right) dx$$

- Data fidelity term provides a bias that forces *u* to be close to the original image *f* .
- In the regularization term, the edge strength function *v* specifies the boundary points and guides the smoothing accordingly.
- Since *v* ≈ 0 along the boundaries, no smoothing is carried out at the boundary points, thus the edges are preserved.

Ambrosio-Tortorelli (AT) Approximation: v process

- Keeping *u* fixed, PDE for the process *v* minimizes the following convex quadratic functional:

$$\frac{\rho}{2}\int_{\Omega}\left(|\nabla v|^2+\frac{1+2\alpha\rho|\nabla u|^2}{\rho^2}\left(v-\frac{1}{1+2\alpha\rho|\nabla u|^2}\right)^2\right)dx$$

- The function *v* is nothing but a smoothing of $\frac{1}{1+2\alpha\rho|\nabla u|^2}$
- The smoothness term forces some spatial organization by requiring the edges to be smooth.
- Ignoring the smoothness term and letting ρ go to 0, we have

$$v \approx \frac{1}{1+2\alpha\rho|\nabla u|^2}$$

Relating with the Perona-Malik Diffusion

- Replacing *v* with $1/(1+2\alpha\rho|\nabla u|^2)$, PDE for the process *u* can be interpreted as a biased Perona-Malik type nonlinear diffusion:

$$\frac{\partial u}{\partial t}=\nabla\cdot(g(|\nabla u|)\nabla u)-\frac{\beta}{\alpha}(u-f)$$

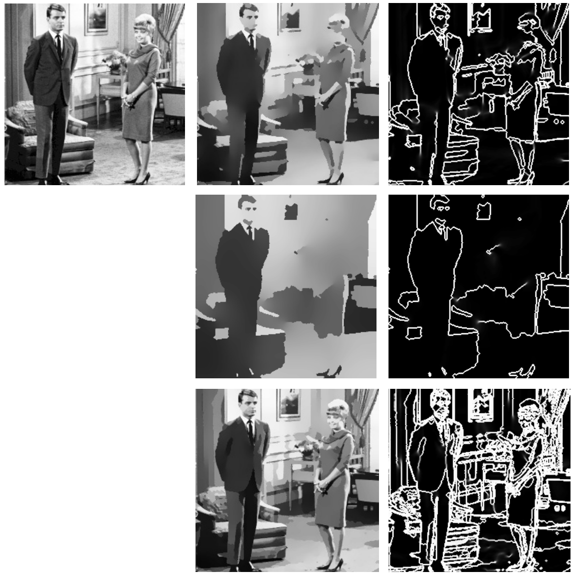
with

$$g(|\nabla u|)=\left(\frac{1}{1+|\nabla u|^2/\lambda^2}\right)^2$$

$$\lambda^2=1/\ (2\alpha\rho)$$

- $\sqrt{1/\ (2\alpha\rho)}$ as a contrast parameter
- Relative importance of the regularization term (scale) depends on the ratio betweenα and β.

Sample Results of the AT model



$\alpha=1,\beta=0.01,\rho=0.01$

$\alpha=1,\beta=0.001,\rho=0.01$

$\alpha=4,\beta=0.04,\rho=0.01$

Numerical Implementation

- Original model:

$$\frac{\partial u}{\partial t} \;=\; \nabla\cdot(v^2\nabla u)-\frac{\beta}{\alpha}(u-f);\quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega}=0$$

- Space discrete version:

$$\begin{aligned}\frac{du_{i,j}}{dt} &= v^2_{i+\frac{1}{2},j}\cdot\left(u_{i+1,j}-u_{i,j}\right)-v^2_{i-\frac{1}{2},j}\cdot\left(u_{i,j}-u_{i-1,j}\right)\\ &+\; v^2_{i,j+\frac{1}{2}}\cdot\left(u_{i,j+1}-u_{i,j}\right)-v^2_{i,j-\frac{1}{2}}\cdot\left(u_{i,j}-u_{i,j-1}\right)\\ &-\;\frac{\beta}{\alpha}\left(u_{i,j}-f_{i,j}\right)\ ,\end{aligned}$$

with $v_{i,j\pm\frac{1}{2}}=\frac{v_{i,j\pm1}+v_{i,j}}{2}$ and $v_{i\pm\frac{1}{2},j}=\frac{v_{i\pm1,j}+v_{i,j}}{2}$

Numerical Implementation

- Original model:

$$\frac{\partial v}{\partial t} = \nabla^2 v - \frac{2\alpha|\nabla u|^2 v}{\rho} - \frac{(v-1)}{\rho^2}; \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0$$

- Space discrete version:

$$\begin{aligned}\frac{dv_{i,j}}{dt} &= v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j} \\ &- \frac{2\alpha|\nabla u_{i,j}|^2 v_{i,j}}{\rho} - \frac{(v_{i,j}-1)}{\rho^2}.\end{aligned}$$

Numerical Implementation

- Space-time discrete versions:

$$\begin{aligned}\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\Delta t} &= \left(v^k_{i+\frac{1}{2},j}\right)^2 \cdot u^k_{i+1,j} + \left(v^k_{i-\frac{1}{2},j}\right)^2 \cdot u^k_{i-1,j} \\ &+ \left(v^k_{i,j+\frac{1}{2}}\right)^2 \cdot u^k_{i,j+1} + \left(v^k_{i,j-\frac{1}{2}}\right)^2 \cdot u^k_{i,j-1} \\ &- \left(\left(v^k_{i+\frac{1}{2},j}\right)^2 + \left(v^k_{i-\frac{1}{2},j}\right)^2 + \left(v^k_{i,j+\frac{1}{2}}\right)^2 + \left(v^k_{i,j-\frac{1}{2}}\right)^2\right) \cdot u^k_{i,j} \\ &- \frac{\beta}{\alpha}\left(\boxed{u^{k+1}_{i,j}} - f_{i,j}\right),\end{aligned}$$

$$\begin{aligned}\frac{v_{i,j}^{k+1} - v_{i,j}^{k}}{\Delta t} &= v^k_{i+1,j} + v^k_{i-1,j} + v^k_{i,j+1} + v^k_{i,j-1} - 4v^k_{i,j} \\ &- \frac{\alpha\left(\left(u^k_{i+1,j} - u^k_{i-1,j}\right)^2 + \left(u^k_{i,j+1} - u^k_{i,j-1}\right)^2\right)\boxed{v^{k+1}_{i,j}}}{2\rho} - \frac{\left(v^{k+1}_{i,j} - 1\right)}{\rho^2}\end{aligned}$$

A Common Framework for Curve Evolution, Segmentation and Anisotropic Diffusion

- Quadratic cost functions in the data fidelity and the smoothing terms are replaced with $L1$-functions (Shah, CVPR 1996):

$$E_S(u,v) = \int_\Omega \left(\beta\,|u-f| + \alpha\, v^2|\nabla u| + \frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1-v)^2}{\rho}\right)\right) dx$$

- As ρ → 0, this energy functional converges to the following functional:

$$E_{S2}(u,\Gamma) = \frac{\beta}{\alpha}\int_\Omega |u-f|dx + \int_{\Omega\setminus\Gamma} |\nabla u| dx + \int_\Gamma \frac{J_u}{1+\alpha J_u} ds$$

with $J_u = |u^+ - u^-|$ indicating the jump in u across Γ, and u+ and u– denote inten-sity values on two sides of Γ

A Common Framework for Curve Evolution, Segmentation and Anisotropic Diffusion

- Minimizing the energy functional results in the following system of coupled PDEs:

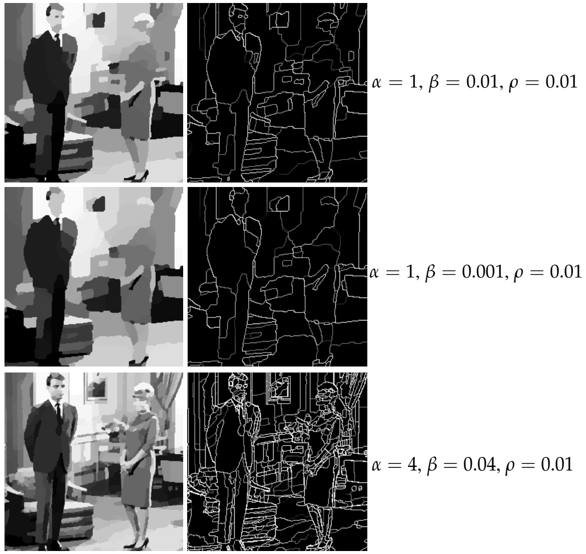
$$\frac{\partial u}{\partial t} = 2\nabla v \cdot \nabla u + v|\nabla u|\; curv(u) - \frac{\beta}{\alpha v}|\nabla u|\frac{(u-f)}{|u-f|}; \quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = 0$$

$$\frac{\partial v}{\partial t} = \nabla^2 v - \frac{2\alpha|\nabla u| v}{\rho} - \frac{(v-1)}{\rho^2}; \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0$$

with $curv(u) = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right)$

- Replacing $L2$-norms in both the data fidelity and the smoothness terms by their $L1$-norms generates shocks in u and thus object boundaries are recovered as actual discontinuities.

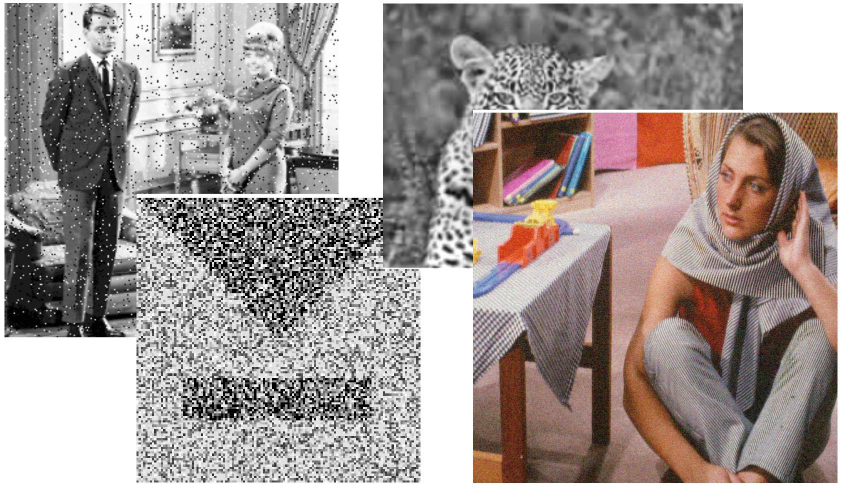
Sample Results of Shah (CVPR96)


$\alpha = 1, \beta = 0.01, \rho = 0.01$
$\alpha = 1, \beta = 0.001, \rho = 0.01$
$\alpha = 4, \beta = 0.04, \rho = 0.01$

- Smoothing process of *u* gives rise to more cartoon-like, piecewise constant images

  but with some unintuitive regions

Challenging Cases



Context-Guided Image Smoothing

- E. Erdem, A. Sancar-Yilmaz, and S. Tari, *"Mumford-Shah Regularizer with Spatial Coherence"*, International Conference on Scale Space and Variational Methods (SSVM) 2007
- E. Erdem and S. Tari, *"Mumford-Shah Regularizer with Contextual Feedback"*, Journal of Mathematical Imaging and Vision, Vol. 33, No.1, pp. 67-84, January 2009

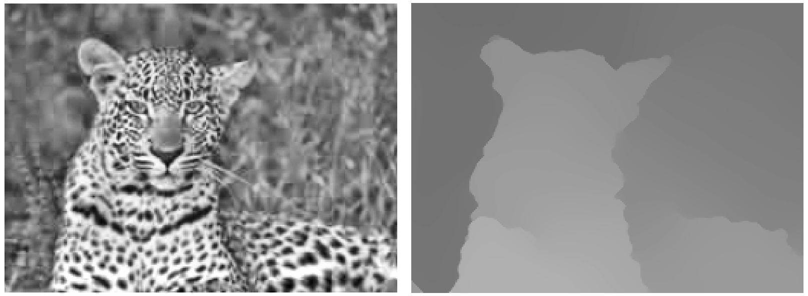
- Contextual knowledge extracted from local image regions guides the regularization process.

Context-Guided Image Smoothing

- E. Erdem, A. Sancar-Yilmaz, and S. Tari, *"Mumford-Shah Regularizer with Spatial Coherence"*, International Conference on Scale Space and Variational Methods (SSVM) 2007
- E. Erdem and S. Tari, *"Mumford-Shah Regularizer with Contextual Feedback"*, Journal of Mathematical Imaging and Vision, Vol. 33, No.1, pp. 67-84, January 2009

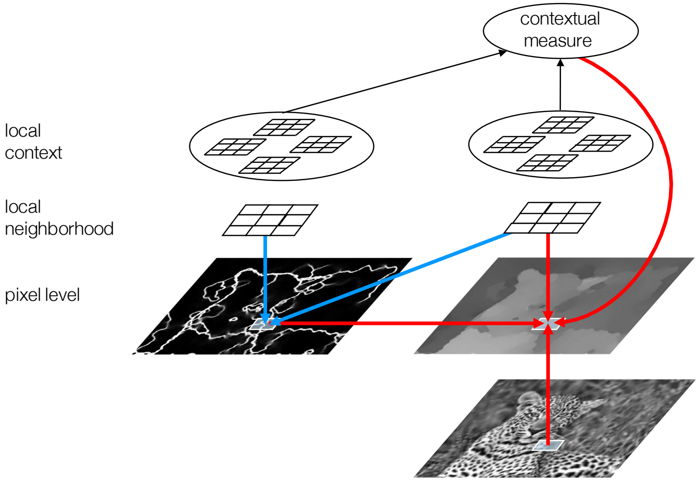
- Contextual knowledge extracted from local image regions guides the regularization process.

Context-Guided Image Smoothing



- Contextual knowledge extracted from local image regions guides the regularization process.

Context-Guided Image Smoothing



Context-Guided Image Smoothing

- 2 coupled processes (u and v modules)

$$\frac{\partial v}{\partial t} = \nabla^2 v - \frac{2\alpha|\nabla u|^2 v}{\rho} - \frac{(v-1)}{\rho^2}; \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0$$

$$\frac{\partial u}{\partial t} = \nabla \cdot ((cv)^2 \nabla u) - \frac{\beta}{\alpha}(u-f); \quad \left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = 0$$

$$cv = \phi v + (1-\phi)V$$

$\phi \in [0,1]$ $\quad$ $V \in \{0,1\}$

The Roles of ϕ and V

1. Eliminating an accidentally occurring event
   - **e.g.**, a high gradient due to noise
   - V=1, ϕ is low for accidental occurrences

$$(cv)_i^2 = (\phi_i v_i + (1-\phi_i)\,1)^2$$

2. Preventing an accidental elimination of a feature of interest
   - **e.g.**, encourage edge formation
   - V=0, ϕ is low for meaningful occurrences

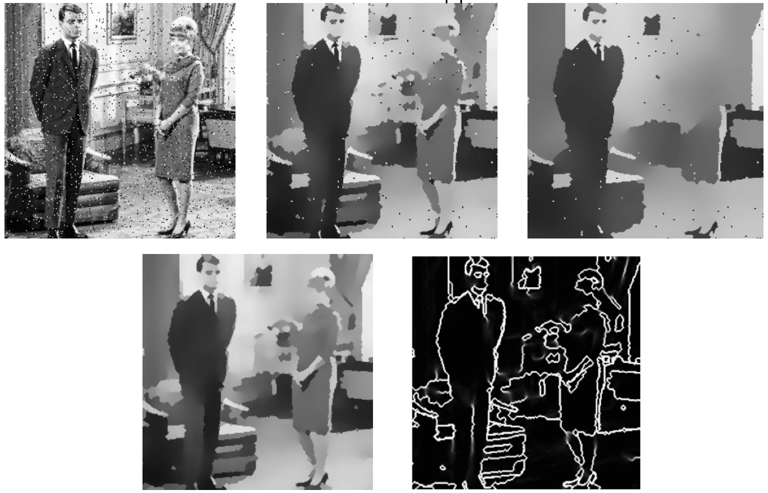
$$(cv)_i^2 = (\phi_i v_i + (1-\phi_i)\,0)^2$$

Experimental Results

- Suggested contextual measures:
  1. Directional consistency of edges
     - shapes have smooth boundaries
  2. Edge Continuity
     - gap filling
  3. Texture Edges
     - boundary between different textured regions
  4. Local Scale
     - Resolution varies throughout the image

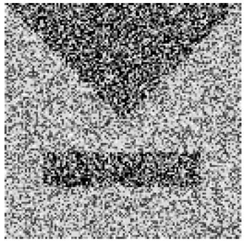
Directional Consistency

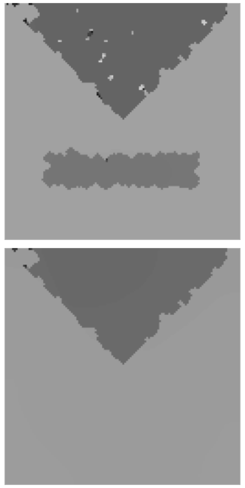
Approximate MS



Context guided filtering result

Directional Consistency

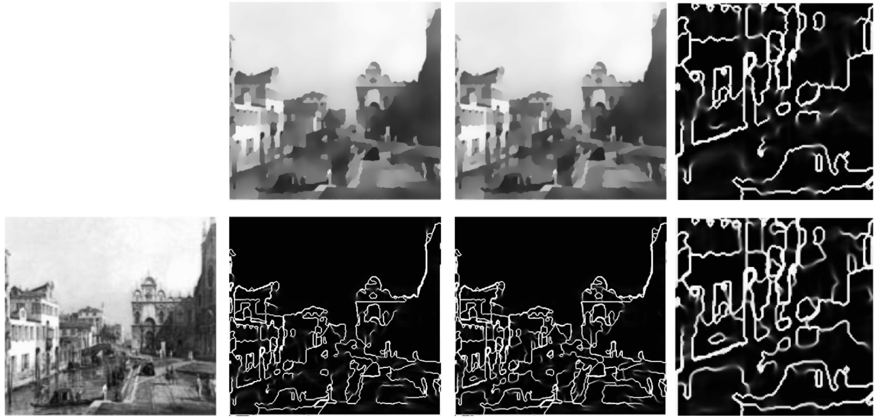




Approximate MS

Context guided filtering result

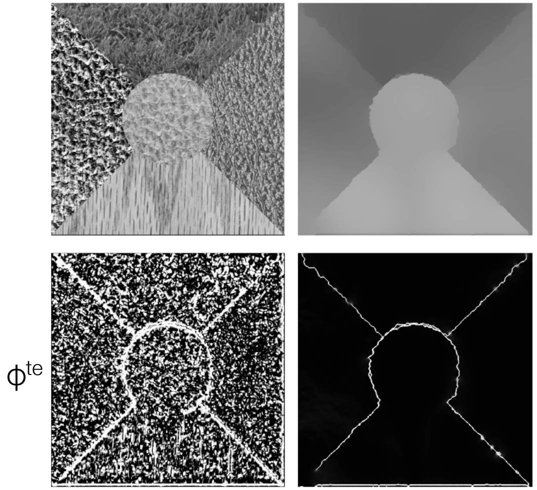
Edge Continuity



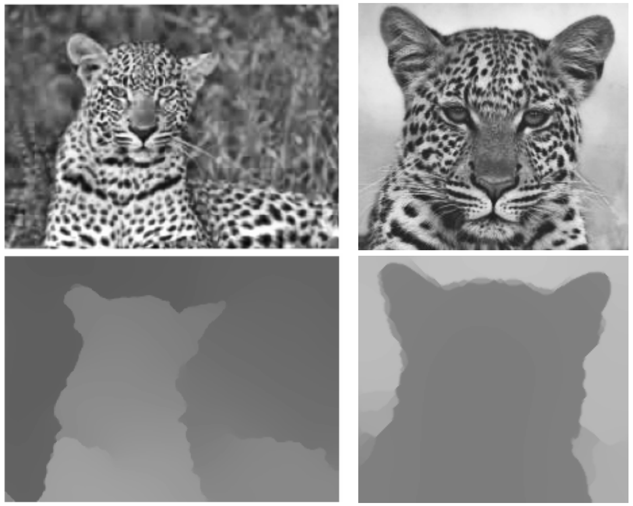
Approximate MS

Context guided filtering result

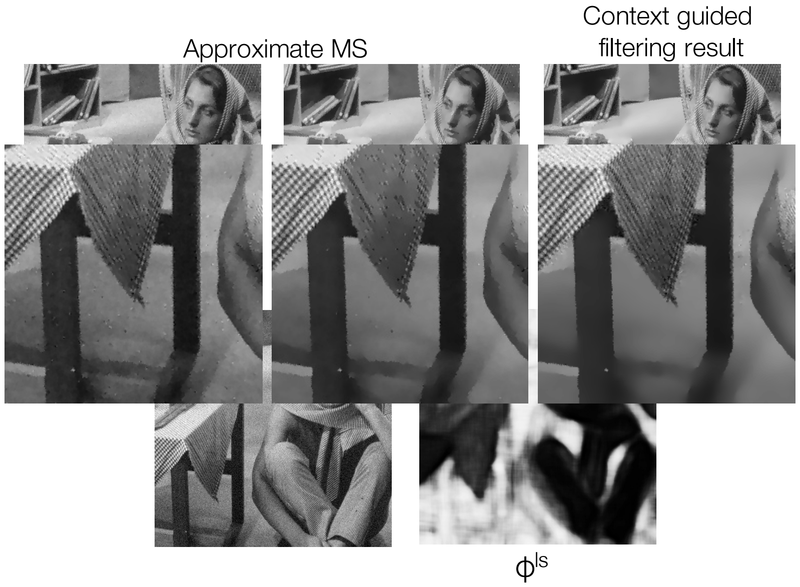
Coalition of Directional Consistency and Texture Edges



Coalition of Directional Consistency, Edge Continuity and Texture Edges



Local Scale



Active Contours Without Edges

- A level-set based approximation of the Mumford-Shah model proposed by Chan and Vese (2001).
- Level sets provide an implicit contour representation where an evolving curve is represented with the zero-level line of a level set function.

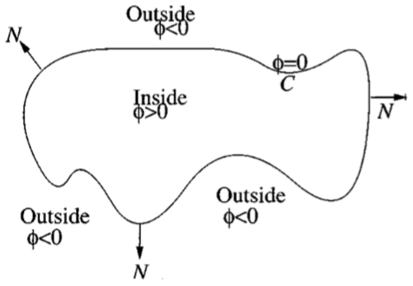


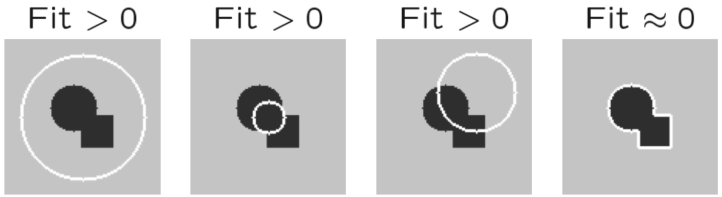
Image credit: Chan & Vese, 2001

Active Contours Without Edges

- **Basic idea:** Fitting term

$$\int_{inside(C)} |u_0 - c_1|^2 dxdy + \int_{outside(C)} |u_0 - c_2|^2 dxdy$$

$$\text{where} \begin{cases} c_1 = \text{average of } u_0 \text{ inside } C \\ c_2 = \text{average of } u_0 \text{ outside } C \end{cases}$$



- **Minimize:** the Fitting term + Length(C)

Slide credit: L. Vese

Active Contours Without Edges

- A level-set based approximation of the Mumford-Shah model proposed by Chan and Vese (2001):

$$\begin{aligned} E_{CV}(c_1, c_2, \phi) &= \lambda_1 \int_\Omega (f - c_1)^2 H(\phi) dx + \lambda_2 \int_\Omega (f - c_2)^2 (1 - H(\phi)) dx \\ &+ \mu \int_\Omega |\nabla H(\phi)| dx \end{aligned}$$

where λ_1, $\lambda_2 > 0$ and $\mu \geq 0$ are fixed parameters.

- Length parameter μ can be interpreted as a scale parameter. It determines the relative importance of the length term.
- Possibility of detecting smaller objects/regions increases with decreasing μ.

Active Contours Without Edges

- A level-set based approximation of the Mumford-Shah model proposed by Chan and Vese (2001):

$$\begin{aligned} E_{CV}(c_1, c_2, \phi) &= \lambda_1 \int_\Omega (f - c_1)^2 H(\phi) dx + \lambda_2 \int_\Omega (f - c_2)^2 (1 - H(\phi)) dx \\ &+ \mu \int_\Omega |\nabla H(\phi)| dx \end{aligned}$$

- Model represents the segmented image with the variables c_1, c_2 and $H(\phi)$, where $H(\phi)$ denotes the Heaviside function of the level set function ϕ:

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Active Contours Without Edges

- A level-set based approximation of the Mumford-Shah model proposed by Chan and Vese (2001):

$$\begin{aligned} E_{CV}(c_1, c_2, \phi) &= \lambda_1 \int_\Omega (f - c_1)^2 H(\phi) dx + \lambda_2 \int_\Omega (f - c_2)^2 (1 - H(\phi)) dx \\ &+ \mu \int_\Omega |\nabla H(\phi)| dx \end{aligned}$$

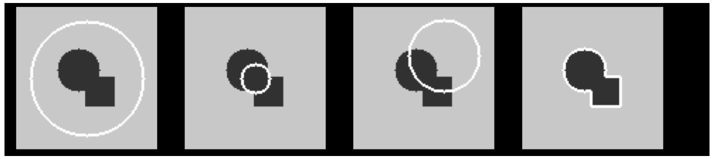
- c_1 and c_2 denote the average gray values of object and background regions indicated by $\phi \geq 0$ and $\phi < 0$, respectively.
- Chan-Vese model can be seen as a two-phase piecewise constant approximation of the MS model.

Active Contours Without Edges

- A level-set based approximation of the Mumford-Shah model proposed by Chan and Vese (2001):

$$\begin{aligned} E_{CV}(c_1, c_2, \phi) &= \lambda_1 \int_\Omega (f - c_1)^2 H(\phi) dx + \lambda_2 \int_\Omega (f - c_2)^2 (1 - H(\phi)) dx \\ &+ \mu \int_\Omega |\nabla H(\phi)| dx \end{aligned}$$

where λ_1, $\lambda_2 > 0$ and $\mu \geq 0$ are fixed parameters.



Active Contours Without Edges

- Segmentation involves minimizing the energy functional with respect to c_1, c_2, and ϕ.
- Keeping ϕ fixed, the average gray values c_1 and c_2 can be estimated as follows:

$$\begin{aligned} c_1 &= \frac{\int_\Omega f(x) H(\phi(x)) dx}{\int_\Omega H(\phi(x)) dx}, \\ c_2 &= \frac{\int_\Omega f(x)(1 - H(\phi(x))) dx}{\int_\Omega (1 - H(\phi(x))) dx} \end{aligned}$$

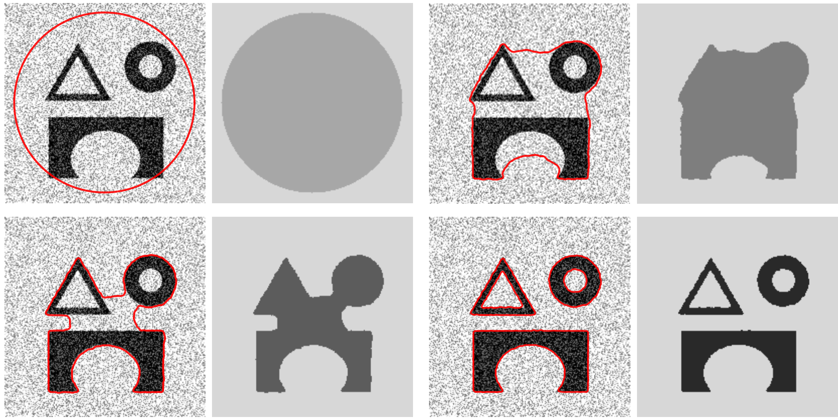
Active Contours Without Edges

- Segmentation involves minimizing the energy functional with respect to c_1, c_2, and ϕ.
- Keeping c_1 and c_2 fixed and using the calculus of variations for the given functional, the gradient descent equation for the evolution of ϕ is derived as:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \, \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2 \right]$$

Sample result of the Chan-Vese Model

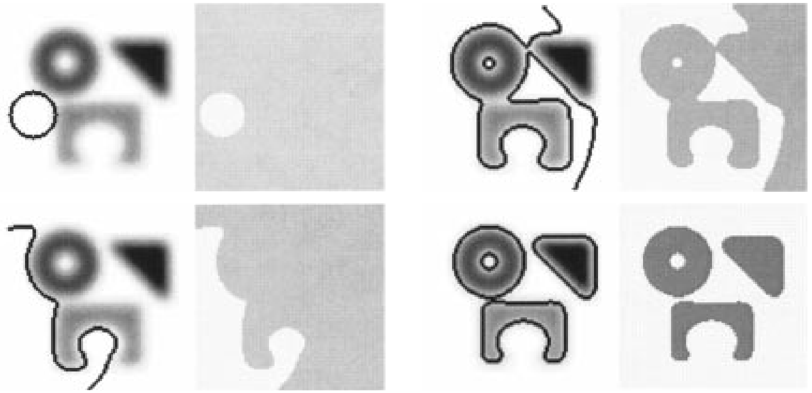
- As the zero-level line of the evolving level set function ϕ is attracted to object boundaries, a more accurate piecewise constant approximations of the original image f is recovered.



$\phi_0 = -\sqrt{(x-100)^2 + (y-100)^2} + 90$ $\lambda_1 = \lambda_2 = 1, \mu = 0.5 \cdot 255^2$

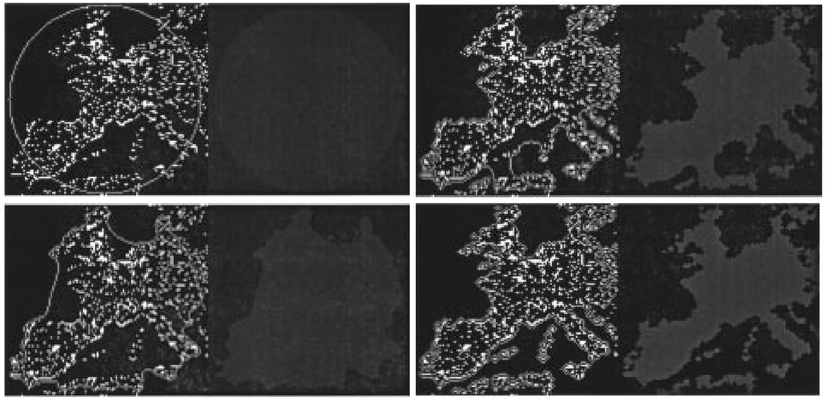
Sample result of the Chan-Vese Model

- As the zero-level line of the evolving level set function ϕ is attracted to object boundaries, a more accurate piecewise constant approximations of the original image f is recovered.



Sample result of the Chan-Vese Model

- As the zero-level line of the evolving level set function ϕ is attracted to object boundaries, a more accurate piecewise constant approximations of the original image f is recovered.



Numerical Implementation

- In the numerical approximation, regularized form of the Heaviside function is used:

$$
\begin{aligned}
H_\varepsilon(z) &= \frac{1}{2}\left(1+\frac{2}{\pi} arctan\left(\frac{z}{\varepsilon}\right)\right) \\
\delta_\varepsilon(z) &= \frac{dH_\varepsilon(z)}{dz} = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2+z^2}
\end{aligned}
$$

Numerical Implementation

- Space-time discrete version:

$$
\begin{aligned}
\frac{\phi_{i,j}^{k+1}-\phi_{i,j}^{k}}{\Delta t} = \quad \delta(\phi_{i,j}^{k}) \quad & \left[\mu\Delta_-^x \cdot \left(\frac{\Delta_+^x\phi_{i,j}^{k+1}}{\sqrt{(\Delta_+^x\phi_{i,j}^{k})^2+(\phi_{i,j+1}^{k}-\phi_{i,j-1}^{k})^2/4}}\right)\right. \\
& + \mu\Delta_-^y \cdot \left(\frac{\Delta_+^y\phi_{i,j}^{k+1}}{\sqrt{(\phi_{i+1,j}^{k}-\phi_{i-1,j}^{k})^2/4+(\Delta_+^y\phi_{i,j}^{k})^2}}\right) \\
& \left. - \lambda_1\left(f_{i,j}-c_1(\phi^k)\right)^2+\lambda_2\left(f_{i,j}-c_2(\phi^k)\right)^2\right]
\end{aligned}
$$

with

$$
\begin{aligned}
\Delta_-^x\phi_{i,j} &= \phi_{i,j}-\phi_{i-1,j}, \quad \Delta_+^x\phi_{i,j} = \phi_{i+1,j}-\phi_{i,j}, \\
\Delta_-^y\phi_{i,j} &= \phi_{i,j}-\phi_{i,j-1}, \quad \Delta_+^y\phi_{i,j} = \phi_{i,j+1}-\phi_{i,j}.
\end{aligned}
$$