Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation

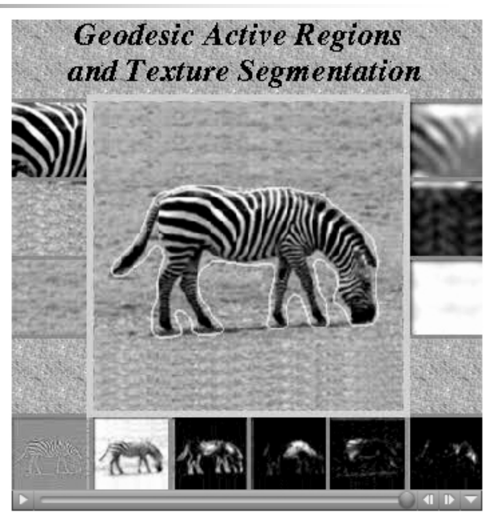
Nikos Paragios, Rachid Deriche

IJCV, 223-247, 2002

ftp://ftp-sop.inria.fr/odyssee/Publications/2002/paragios-deriche:02.pdf

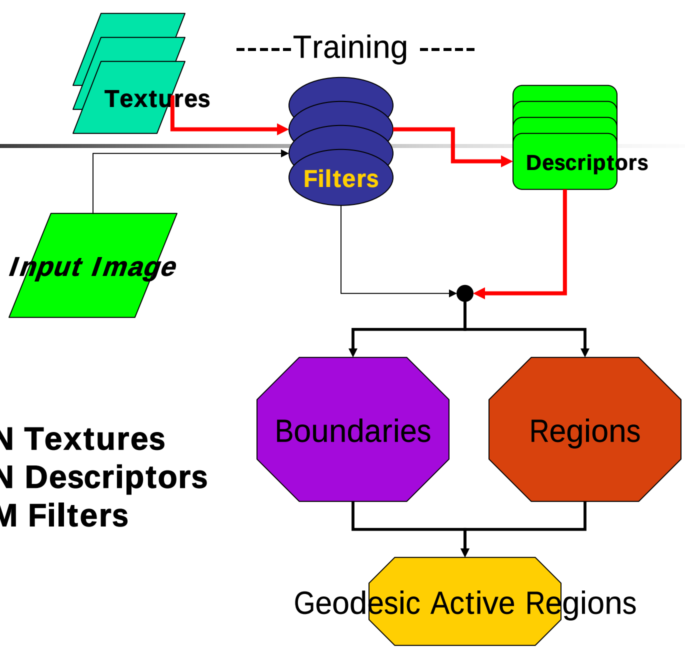
Problem – Supervised Texture Segmentation

- Given an image of foreground and background textures, find the boundary between them.
- "Supervised" – A priori knowledge of the textures



Algorithm Basis

- Paper takes from three areas of research
  - Texture analysis
    - Statistical modeling, filtering theory
  - Boundary-based image segmentation
    - Edges, snakes, active contours
  - Region-based image segmentation
    - Region-growing, statistical homogeneity, …



•N Textures
•N Descriptors
•M Filters

Training - Filters

- Predefined set of filter operators
  - Isotropic
  - Anisotropic
  - Gabor
- Modeling Phase - filters are modeled for each texture pattern
- Validation of filters – reliability measurements (to be used as weights)

Training – Set of filters

- Gaussian operator – $\mathbf{g(x,y|\sigma)}$
- Laplacian of Gaussian – $\mathbf{l(x,y|\sigma)}$
- Spectrum analyzer of the 2D Gabor filters $\mathbf{s(x,y|\sigma,\theta,\varphi)}$
  - Combination of the real and imaginary parts of the 2D Gabor function result
- **(See page 227 in paper for equations)**

Training – Set of filters

· The 2D Gabor operators analyze the image simultaneously in both space [σ], and frequency domains [θ, ϕ].

$$\left[g_G(x, y \mid \sigma, \theta, \phi) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} e^{-j2\pi(\theta x + \phi y)} \right]$$

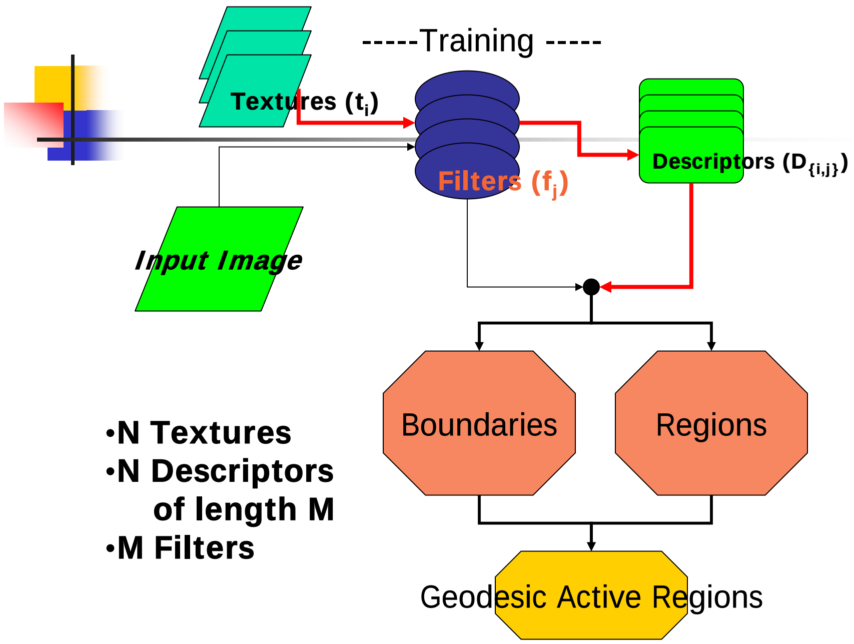
These Gabor functions can be decomposed into two components; the real part [$g_R(x, y \mid \sigma, \theta, \phi)$] and the imaginary part [$g_I(x, y \mid \sigma, \theta, \phi)$]. The texture features are captured by the spectrum analyzer {s(|σ, θ, ϕ)} of the Gabor components,

$$s(x, y \mid \sigma, \theta, \phi) = \sqrt{(g_R * I)(x, y)^2 + (g_I * I)(x, y)^2}$$

smoothed by a Gaussian function, where ($G_R * I$) denotes the convolution operation between the image I and the filter G_R.

Training – Modeling Phase

- **T,t:** N texture patterns
- **P:** N texture pattern images
- **F,f:** M filters
- **D:** NxM multidimensional feature training data space
- **$D_{\{i,j\}}$:** data component of the i^{th} texture with the j^{th} filter
  - **(refer to page 228 in paper)**



- **N Textures**
- **N Descriptors of length M**
- **M Filters**

Training – Modeling Phase (cont)

- $p_{\{i,j\}}$: conditional probability density of the data component $D_{\{i,j\}}$ (normalized histogram)
- Decompose each $D_{\{i,j\}}$ into C_N=2 Gaussians of mean μ and std σ (use maximum likelihood principle – Duda and Hart, 1973)
- $P^k_{\{i,j\}}$: is the probability of component k of $D_{\{i,j\}}$

Training – Modeling Phase (cont)

- $p_{\{i,j\}}(x|\theta_{\{i,j\}})$ =
- $p_{\{i,j\}}(x|P^k_{\{i,j\}},\mu^k_{\{i,j\}},\sigma^k_{\{i,j\}})$ =

$$\sum P^k_{\{i,j\}} * p_{\{i,j\}}(x|\mu^k_{\{i,j\}},\sigma^k_{\{i,j\}})$$

- Basically just saying a PDF can be modeled by a weighted sum of Gaussian functions

Training Phase - Validation

- Goal – Get a set of weights for each filter
- Texture Descriptor ($\mathbf{p}_t$) : Each of the N textures (t) now has a M length tuple of conditional PDFs
- Want to measure the error rate of each of our filter operators.
- Misclassification rate: A pixel s is misclassified, iff there is another texture pattern for which the cond. probability given the observed value is greater than the true case.

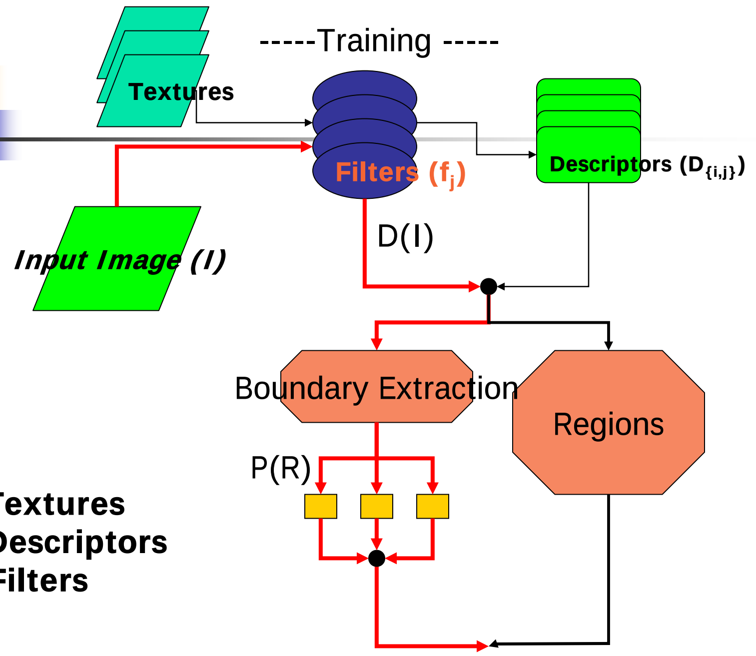
Training Phase - Validation

- For every pixel in convolved pattern image, for each component ($1..C_N$), return a 1 if the true $\mathbf{P_f(t)}$ (probability) of a texture is always greater than the $\mathbf{P_f(t)}$ of all other textures. This gives you a correctness probability weight for a given filter operator. **(page 229-230)**
- For a given filter, average the correctness probability across all textures to give us a reliability measurement, $\boldsymbol{\omega_f}$.
- (page 229-230)

Training Phase - Validation

$$P_{f_o}(t_p) = \frac{1}{|D|} \int_D \int_{\{t_p, f_o\}} \underbrace{H_{\{t_p, f_o\}}\left(D_{\{t_p, f_o\}}(x, y)\right)}_{\text{Correctness function}} \, dx \, dy \qquad (2)$$

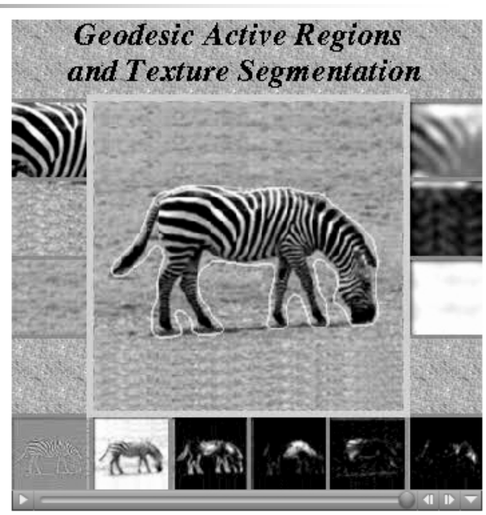
$$\underbrace{H_{\{t_p, f_o\}}(a) = \prod_{i=1, i \neq t_p}^{N} \left[p_{\{t_p, f_o\}}(a) \geq p_{\{i, f_o\}}(a) \right]}_{a \text{ comes from image } D_{\{t_p, f_o\}}} \qquad (3)$$



•N Textures
•N Descriptors
•M Filters

Big Picture

- Find an equation whose zero level set represents the boundary. This equation is composed of boundary and region forces.



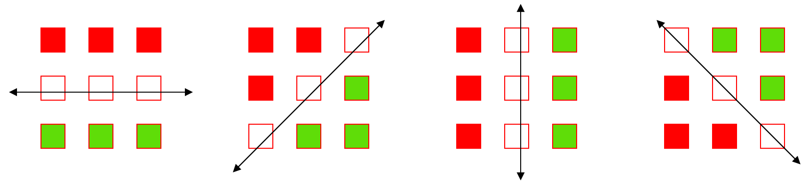
Boundaries -

- Relies heavily on work done by Caselles (95,97) and Kichenassamy (95) on geodesic active contours. Their work continued snake algorithms.
- I: texture input frame
- D(I) = I_j: be the set of M filter responses to the image
- P(R) = R_i: be the set of N region boundaries
- t_i: texture pattern assigned to R_i
- $p_{\{tx,B\}}(s)$: boundary probability for a given pixels s being at the boundary of region R_x

Boundaries -

pp: 237-238

- s: pixel in the image
- N(s): neighborhood of pixels (3x3) with partitions $N_R(s)$ and $N_L(s)$



Boundary of t_k – If a partition exists where the most probable assignment for the left and right partitions are t_k and another texture.
Non-boundary of t_k – If for all partitions, the most probable assignment for the left and right partitions are both t_k or any pair of other textures not including t_k.

Boundaries -

- Equations 20 and 21 gives us the probability that a pixel belongs to a boundary of a particular texture.
- We'll get a set of **M** probabilities for a given pixel and will condense this into a single probability using the previously calculated reliability measurements, $\boldsymbol{\omega_f}$, as weights in a weighted average.

Boundaries -

$$E(\mathcal{P}(\mathcal{R})) = \overbrace{\int_0^1 g\left(\underbrace{p_C(\partial\mathcal{R}(c))}_{boundary\ probability}\right)}^{Boundary\ Attraction} \underbrace{|\partial\dot{\mathcal{R}}(c)|}_{Regularity}\, dc \qquad (23)$$

Regions –

$$p(\mathcal{P}(R) \mid D(I)) = \prod_{i=0}^{N} \prod_{j=1}^{M} p\left(I_j(\mathcal{R}_i) \,\middle|\, t_{R_i}\right) \qquad (25)$$

- For energy function:
  - Use reliability weights
  - Maximization of probability is equivalent to the minimization of the [-log()] of this probability

Geodesic Active Regions

- Minimize the combined boundary and region energy functions

$$E(\partial\mathcal{P}(\mathcal{R})) = (1-\alpha)\int_0^1 \overbrace{g\left(\underbrace{p_C(\partial\mathcal{R}(c))}_{boundary\ probability}\right)\underbrace{|\partial\dot{\mathcal{R}}(c)|}_{Regularity}}^{Boundary\ Attraction} dc$$
$$\underbrace{-\alpha\sum_{i=0}^{N}\int_{R}\int_{i}\sum_{j=1}^{N} w_j \log\left[\overbrace{p_{(\{t_{R_i},j\})}(I_j(x,y))}^{i\ region\ probability}\right] dx\,dy}_{i\ region\ attraction} \qquad (26)$$

Geodesic Active Regions

$$\frac{\partial u}{\partial t} = \Bigg[\alpha \sum_{j=1}^{N} w_j \log\left(\frac{p_{\{t_{R_0}, j\}}(I_j(u))}{p_{\{t_{R_k}, j\}}(I_j(u))} \right) \\ (1-\alpha)(g(p_C(u))\mathcal{K}(u) \\ - \nabla g(p_C(u)) \cdot \mathcal{N}(u)) \Bigg] \mathcal{N}(u) \qquad (27)$$

$$\frac{\partial \phi}{\partial t}(u) = \alpha \sum_{j=1}^{N} w_j \log\left(\frac{p_{\{t_{R_0}, j\}}(I_j(u))}{p_{\{t_{R_k}, j\}}(I_j(u))} \right) |\nabla \phi(u)| \\ + (1-\alpha)\Bigg(g(p_C(u))\mathcal{K}(u) \\ - \nabla g(p_C(u)) \cdot \frac{\nabla \phi(u)}{|\nabla \phi(u)|} \Bigg) |\nabla \phi(u)| \qquad (28)$$

Results