Active contour model driven by local histogram fitting energy [1]

Authors: Weiping Liu, Yanfeng Shang, Xin Yang
Presented by: Dakri Abdelmouttaleb

Grenoble-INP ENSIMAG

February 24, 2022

Summary of the presentation

- Introduction
- Background and problem statement
- Proposed method
- Implementation, technical details and results

Introduction

Active contour methods (ACM):

- Edge-driven ACM: Gradient based curve evolution, limited robustness to noise.
- Region driven ACM : Utilises statistics relative to the intensity and textures of the image:

⇒ Parametric Region-driven ACM: Gaussian mixture model, Chan-Vese [2].
Issue: Estimating the object and background distribution.

⇒ Non-Parametric Region-driven ACM: the distributions of different regions are estimated by a nonparametric statistical methods.

The basic idea: strike a balance between local and global features

- Construct a local fitting energy in a local region confined by a kernel function.

⇒ The optimization method can drive the evolving curve according to local nonparametric statistical information represented by a local histogram.

⇒ Choose a suitable width.

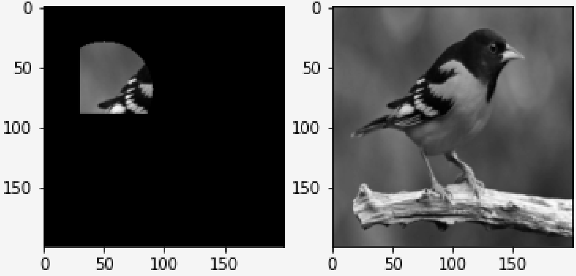
Background and problem statement

Setting up the problem:
Ω an image domain, and $I(\mathbf{x}) : \Omega \to [0, L]$. We define $\vec{C}$ as an initial curve that splits Ω into Σ^{in} and Σ^{out}.
We define N_r^x a radial patch of radius r, the local histogram and cumulative distributions are defined for $z \in \mathbb{R}$ as :

$$P_r^x(z) = \frac{|\{y \in N_r^x \cap \Omega \ : \ I(y) = z\}|}{|N_r^x \cap \Omega|}$$

$$F_r^x(z) = \frac{|\{y \in N_r^x \cap \Omega \ : \ I(y) \leqslant z\}|}{|N_r^x \cap \Omega|}$$

Example:



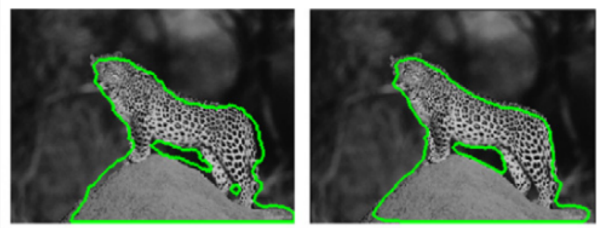
An example of a patch selection (with attention to borders)

The energy functional is defined as such:

$$E(\vec{C}, P^i, P^o) = \text{Length}(\vec{C}) + \lambda(\int_{\Sigma^{in}} D(P^i, P_r^x)dx + \int_{\Sigma^{out}} D(P^o, P_r^x)dx)$$

Where P^i and P^o are two constant histogram inside and outside the evolving curve. λ is a constant parameter. D is a histogram distance. **A histogram distance ?** Two histograms h, g are defined as functions givining a bin value in a bounded set $[0, R]$ a value. Example of L1 distance : $D_1(f, g) = \int_0^R |h(z) - g(z)|dz$. We can use all sorts of other distances (L2, Kullback-Leibler ..).

This method of local histogram based ACM apply local histogram of the current pixels, the histogram of object and background are computed in global domain.
→ if there are some background pixels that the local histogram of them are more similar to the object than the background, for example the mound in the figure below, the segmentation results will be wrong.



Proposed method

Local histogram fitting energy Using a Gaussian Kernel: $K_\sigma(d) = c\ exp(-\frac{d^2}{2\sigma^2})\mathbb{1}_{[-\rho,\rho]}$, where c is a normalizing constant. ρ is the kernel width which defines the size of Gaussian mask, usually $\rho = 2\sigma$. The local histogram fitting energy $E^{LHF}(\vec{C}, P^i, P^o)$:

$$\int_{\Sigma^{in}} K_\sigma(x-y) D_{L1}(P_i^x, P_r^y)dx + \int_{\Sigma^{out}} K_\sigma(x-y) D_{L1}(P_o^x, P_r^y)dx$$

The local histogram fitting energy is a weighted distance of the local histogram P_r^y to the fitting histogram P_i^x and P_o^x with K_σ as the weight assigned to each local histogram P_r^y at y. The weights are dominated by the distance between x and y.

Level set formulation

$$
\begin{aligned}
E_x^{LHF}(\phi, P_i^x, P_o^x) = &\int_\Omega H(\phi) K_\sigma(x-y) \int_0^R |P_i^x(z) - P_r^y(z)| dz dy \\
&+ \int_\Omega (1 - H(\phi)) K_\sigma(x-y) \int_0^R |P_o^x(z) - P_r^y(z)| dz dy
\end{aligned}
$$

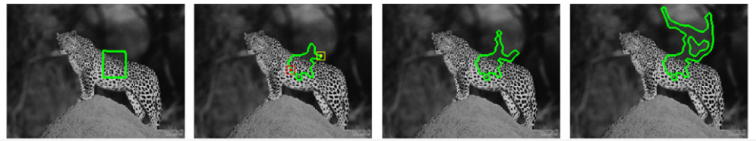
Where H is the heaviside step function.
GOAL: minimize the local histogram fitting energy for all of the points in the whole image domain. Thus, the total energy is defined by:

$$
E(\phi, P_i^x, P_o^x) = \int_\Omega |\nabla H(\phi(x))| dx + \lambda \int_\Omega E^{LHF}(\phi, P_i^x, P_o^x) dx
$$

Gradient algorithm and optimisation

$$\phi_{i+1} = \phi_i + \delta\Big\{ div\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big) + \int_\Omega K_\sigma(x-y)\int_0^R |P_i^x(z) - P_r^y(z)| - |P_o^x(z) - P_r^y(z)| dzdy \Big\}$$

Problem with the proposed method for now : The various parts of the evolving curve may evolve independently.



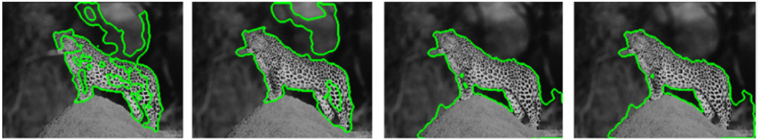
The evolution of the contour curve.

$\rightarrow$ The yellow and red pixels compute their histogram independently (local fitting histograms), which are irrelevant to the object.

Varying the kernel width of the local fitting histogram

- If a small ρ is applied, more fine results in real boundary will be obtained, but it creates too many local minima.
- f a larger ρ is applied, global information will be involved, but the curve will leak out from the real boundary as a result of lacking of the local information.

$\rightarrow$ The suitable Kernel width ρ varies during the curve evolution.



$\rho = 10, 20, 30, 40.$

Strategy: If the two local fitting histograms are similar ρ should be increased, otherwise ρ should remain the same.

$$MinIn < \frac{D(P^x_{i,L}, P^x_{i,C})}{D(P^x_{i,L}, P^x_{o,C})} < MaxIn$$

$$MinOut < \frac{D(P^x_{o,L}, P^x_{i,C})}{D(P^x_{o,L}, P^x_{o,C})} < MaxOut$$

$$D(P^x_{o,C}, P^x_{i,C}) < \varepsilon$$

$\rightarrow$ Compare at each pixel (theoretically) the difference in histograms between the inside and outside the evolving curve.

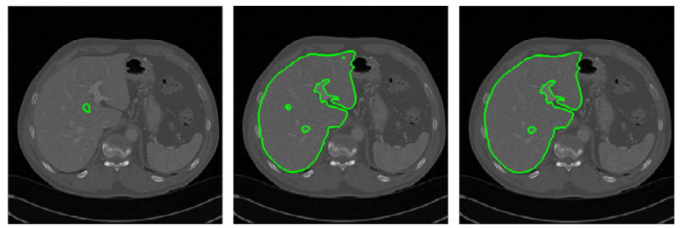
Where $MinIn, MaxIn, MinOut, MaxOut, \varepsilon$ are all pre-defined quantites.

Implementation and technical details

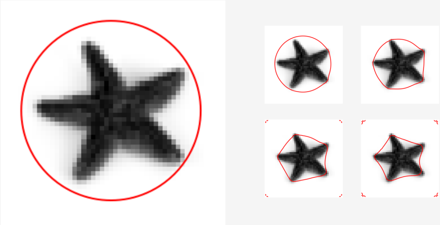
The algorithm

1. Reduce the original gray level of the image to be processed.
2. Select patch scale, current kernel width, larger kernel width, thresholds : $MinIn, MaxIn, MinOut, MaxOut, \varepsilon$.
3. Initialize the evolving curve and compute the local histogram of each pixel.
4. Compute the local fitting histograms of the pixels using current kernel width.
5. Compute the local fitting histograms of the pixels in the bilateral linked list using larger kernel width.
6. Check whether any of the inequalities [prev. slide] is satisfied, if yes, using the larger kernel width and the corresponding local fitting histograms to evolve the curve. If not, using the current kernel width.
7. Check whether convergence or maximum number of iterations is reached. If not, go back to step 4.

Results:



Authors implementation.



Actual implementation and remarks.

Bibliography

[1] W. Liu, Y. Shang, and X. Yang, "Active contour model driven by local histogram fitting energy," *Pattern Recognition Letters*, vol. 34, pp. 655–662, Apr. 2013. DOI: 10.1016/j.patrec.2013.01.005.

[2] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001. DOI: 10.1109/83.902291.