

Supplementary Material: End-to-End Trainable Deep Active Contour Models for Automated Image Segmentation: Delineating Buildings in Aerial Imagery

Ali Hatamizadeh, Debleena Sengupta, and Demetri Terzopoulos

Computer Science Department
University of California, Los Angeles, CA 90095, USA
{ahatamiz,debleenas,dt}@cs.ucla.edu

1 Derivation of the ACM Evolution PDE

Following [1], we derive the Euler-Lagrange PDE governing the evolution of the ACM. Let C be a 2D closed time-varying contour represented in $\Omega \in R^2$ by the zero level set of the signed distance map ϕ , and $X_1 = (u, v)$ and $X_2 = (x, y)$ represent two independent spatial variables that each represent a point in Ω . The interior of C is represented by the smoothed Heaviside function

$$H(\phi) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{\phi}{\epsilon}\right), \quad (1)$$

the derivative of which is the smoothed Dirac delta function

$$\frac{\partial H(\phi)}{\partial \phi} = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + \phi^2} = \delta(\phi). \quad (2)$$

Using the characteristic function W_s , which selects regions within a square window of size s , the energy functional of C may be written in terms of a generic internal energy density F as

$$E(\phi) = \int_{\Omega_{X_1}} \delta(\phi(X_1)) \int_{\Omega_{X_2}} W_s F(\phi, X_1, X_2) dX_2 dX_1. \quad (3)$$

To compute the first variation of the energy functional, we add to ϕ a perturbation function $\epsilon\psi$, where ϵ is a small number; hence,

$$E(\phi + \epsilon\psi) = \int_{\Omega_{X_1}} \delta(\phi(X_1) + \epsilon\psi) \int_{\Omega_{X_2}} W_s F(\phi + \epsilon\psi, X_1, X_2) dX_2 dX_1. \quad (4)$$

Taking the partial derivative of (4) with respect to ϵ and evaluating at $\epsilon = 0$ yields, according to the product rule,

$$\begin{aligned} \left. \frac{\partial E}{\partial \epsilon} \right|_{\epsilon=0} &= \int_{\Omega_{X_1}} \delta(\phi(X_1)) \int_{\Omega_{X_2}} \psi W_s \nabla_{\phi} F(\phi, X_1, X_2) dX_2 dX_1 + \\ &\quad \psi \int_{\Omega_{X_1}} \gamma \phi(X_1) \int_{\Omega_{X_2}} W_s F(\phi, X_1, X_2) dX_2 dX_1, \end{aligned} \quad (5)$$

where $\gamma\phi$ is the derivative of $\delta(\phi)$. Since $\gamma\phi$ is zero on the zero level set, it does not affect the movement of the curve. Thus the second term in (5) and can be ignored. Exchanging the order of integration, we obtain

$$\left. \frac{\partial E}{\partial \epsilon} \right|_{\epsilon=0} = \int_{\Omega_{X_2}} \int_{\Omega_{X_1}} \psi \delta(\phi(X_1)) W_s \nabla_\phi F(\phi, X_1, X_2) dX_1 dX_2. \quad (6)$$

Invoking the Cauchy–Schwartz inequality yields

$$\frac{\partial \phi}{\partial t} = \int_{\Omega_{X_2}} \delta(\phi(X_1)) W_s \nabla_\phi F(\phi, X_1, X_2) dX_2. \quad (7)$$

Adding the contribution of the curvature term and expressing the spatial variables by their coordinates, we obtain the desired curve evolution PDE:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \int_{\Omega} W_s \nabla_\phi F(\phi) dx dy \right], \quad (8)$$

where, assuming a uniform internal energy model and defining $m_1(x, y)$ and $m_2(x, y)$ as the mean image intensities inside and outside C and within W_s , we have

$$\nabla_\phi F = \delta(\phi) (\lambda_1(u, v)[I(u, v) - m_1(x, y)]^2 - \lambda_2(u, v)[I(u, v) - m_2(x, y)]^2). \quad (9)$$

2 TDAC Backbone Architecture

In Tables 1 and 2 we present the details of the encoder and decoder in the TDAC backbone CNN architecture. BN, Add, Pool, Upsample, Conv and Conv1 denote batch normalization, addition, 2×2 max pooling, bilinear upsampling, 3×3 convolutional, and 1×1 convolutional layers, respectively.

Table 1: Detailed information about the TDAC encoder.

Operations	Output size
Input	$512 \times 512 \times 3$
Conv, ReLU, BN, Conv, ReLU, BN, Pool	$256 \times 256 \times 16$
Conv, ReLU, BN	$256 \times 256 \times 32$
Conv, ReLU, BN, Conv, ReLU, BN, Add, Pool	$128 \times 128 \times 32$
Conv, ReLU, BN	$128 \times 128 \times 64$
Conv, ReLU, BN, Conv, ReLU, BN, Add, Pool	$64 \times 64 \times 64$
Conv, ReLU, BN	$64 \times 64 \times 128$
Conv, ReLU, BN, Conv, ReLU, BN, Add	$64 \times 64 \times 128$
Conv, ReLU, BN, Conv, ReLU, BN, Add	$64 \times 64 \times 128$
Conv, ReLU, BN, Conv, ReLU, BN, Add	$64 \times 64 \times 128$

Table 2: Detailed information about the TDAC decoder.

Operations	Output size
Input	$64 \times 64 \times 128$
Upsample, Conv, ReLU, BN, Conv, ReLU, BN	$128 \times 128 \times 64$
Upsample, Conv, ReLU, BN, Conv, ReLU, BN	$256 \times 256 \times 32$
Upsample, Conv, ReLU, BN, Conv, ReLU, BN	$512 \times 512 \times 16$
Conv, ReLU, BN	$512 \times 512 \times 16$
Conv1	$512 \times 512 \times 3$

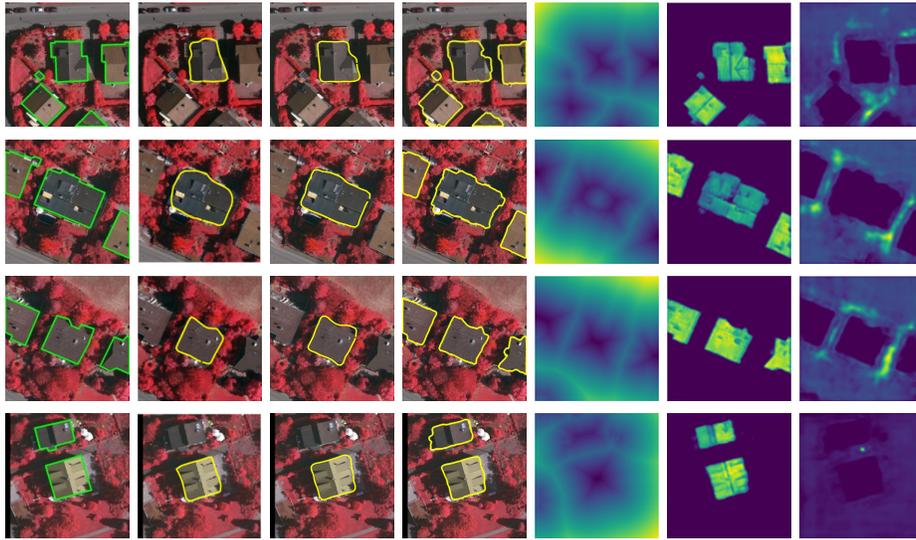
(a) Image (b) DSAC (c) DarNet (d) **TDAC** (e) $\phi_0(x, y)$ (f) $\lambda_1(x, y)$ (g) $\lambda_2(x, y)$

Fig. 1: Additional comparative visualization of the labeled image, the output of DSAC, the output of DarNet, and the output of our TDAC, for the Vaihingen dataset. (a) Image labeled with (green) ground truth segmentation. (b) DSAC output. (c) DarNet output. (d) TDAC output. (e) TDAC’s learned initialization map $\phi_0(x, y)$ and parameter maps (f) $\lambda_1(x, y)$ and (g) $\lambda_2(x, y)$.

3 Comparative Visualization

Additional comparative visualizations are presented in Fig. 1.

References

1. Lankton, S., Tannenbaum, A.: Localizing region-based active contours. *IEEE Transactions on Image Processing* **17**(11), 2029–2039 (2008)