Deformable Mesh Evolved by Similarity of Image Patches

Nguyen, Tuan T.; Dahl, Vedrana Andersen; Barentzen, J. Andreas; Dahl, Anders Bjorholm

Published in:
Proceedings of 2019 IEEE International Conference on Image Processing

Link to article, DOI:
10.1109/icip.2019.8803264

Publication date:
2019

Document Version
Peer reviewed version

DEFORMABLE MESH EVOLVED BY SIMILARITY OF IMAGE PATCHES

Tuan T. Nguyen, Vedrana A. Dahl, J. Andreas Bærentzen, Anders B. Dahl

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Kogens Lyngby 2800, Denmark

ABSTRACT

We propose a deformable model for manually initialized segmentation of images, which may contain both textured and non-textured regions. Image segments and segment boundaries are represented using a deformable triangle mesh, providing all advantages of an explicit geometry representation, but allowing for adaptive topology. Deformation forces are computed using a probabilistic model of local self-similarity, based on clustering of image patches. Both our curve representation and our similarity model naturally support multi-label segmentation. We demonstrate the properties of our approach on a number of natural color images as well as composed textured images.

***Index Terms*—** textured segmentation, adaptive triangle mesh, Mumford-Shah

1. INTRODUCTION

A deformable contour is a curve in the image domain which evolves under the influence of forces inferred from the image and from the curve itself. The approach is very popular in image segmentation, since tracking segment boundaries opens the possibility for regularization and for incorporation of shape priors. Proposed methods vary largely in two characteristics: the representation of the deformable curve and the derivation of the forces deforming the curve.

In general, the curve representation falls in two groups, each with its advantages. An explicit curve (e.g. snakes [1]) can be as simple as a sequence of points connected by line segments. Such a representation is straightforward, uses any desired resolution of points, and a curve deforms easy by displacing points. An implicit curve (largely based on level sets, e.g. [2]) defines an auxiliary function on the whole image domain, and the curve is located where the function changes sign. The most important advantage of an implicit curve is topological adaptivity.

Forces for deforming the curve are numerous. Local methods [1, 3] attract curves to edges or other features characterizing segment boundaries. More global (region-based) forces [4, 2] utilize characterization of segment regions providing a higher robustness to the method.

Our approach offers advances in both curve representation and in deforming forces. The mesh-based curve representation which we use is explicit, but it provides topological adaptivity. Further advantages of this representation are natural multi-label support, adaptive curve resolution and full control of topological changes.

Our curve deforming forces are region-based and bear resemblance to active contours without edges. However, instead of averaging over pixel intensities we perform averaging over image patches, consequently encoding local self-similarity. The resulting forces are therefore able to segment any type of regions, characterized by both texture and intensity.

1.1. Related work

The most important limitation of the explicit curve representation originally proposed in [1] is the lack of topological adaptivity. Providing an explicit curve with topological adaptivity requires resolving curve intersections, examples include both 2D [5] and 3D [6] representations. This may be greatly simplified by using a deformable triangle mesh [7]. Each triangle in such mesh is given a label that indicates to which segment it belongs. The curve (segment boundary) consists of the edges shared by triangles that have different labels.

Applications of deformable meshes in volume and image segmentation include [8] and [9], both with meshes of uniform resolution. The resolution adaptivity is introduced in [10, 11] for segmentation of intensity-based regions. In this paper we extend upon [10] by incorporating forces based on self-similarity, allowing us to segment textured regions.

The typical approach to texture segmentation involves mapping the image to a texture descriptor space. Here the assumption is that descriptors within textures are similar while they differ between textures. Such an approach was suggested by [2] using texture orientation, which has been extended in e.g. [12] using the structure tensor and level sets. For better performance, the scale of the structure tensor is automatically estimated in [13], while [14] utilizes diffusion.

Many other texture descriptors characterizing the local image structure have been suggested. These include local fractal features [15], gradient histograms [16], local binary patterns [17], textons [18], and more. Images often contain

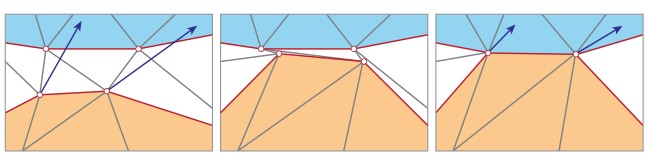
texture on various scales, or areas with deformed or rotated versions of a certain texture. Typically, this is handled in by designing descriptors invariant to such properties.

A related approach for image segmentation is based on sparse dictionaries of image patches [19, 20] where a dedicated dictionary is built for each texture class. Similar methods focusing on optimal reconstruction have been proposed [21], and improved performance has been obtained by also optimizing for discrimination [22]. More recently [23] suggested to use sparse dictionaries together with an user-initiated active contour.

Our approach is closely related to the methods in [24] and [25]. These employ a dictionary that encodes patch-based self-similarities in the image for evolving a deformable boundary. In [24] a snake curve of is used. Consequently, only a single closed curve of a constant resolution may be tracked. These issues are alleviated in [25], where a level set is employed. However, multi-phase forces used there are based on heuristics, while implicit curve representation lacks the compactness and gives only limited possibilities for incorporating shape priors.

2. METHOD

Our aim is to obtain a segmentation where similar patterns in the image belong to the same segment. We utilize a triangle mesh to define a piece-wise constant function, which represents segments of an image (Fig. 1). Each triangle is labeled with an integer $l = 1, .., K$ indicating the segment it belongs to. The number of segments K is fixed.



(a) (b) (c)

Fig. 1: A mesh representing moving segments. Edges constituting the boundary are shown in red

Alg. 1 describes the general algorithm. The segmentation starts with an initialization and then deforms the mesh under forces derived by a model for encoding self-similarity in the image. A related model has been used for evolving active contours [24, 25]. In this work we provide a more efficient clustering algorithm, efficient probability update, and the derivation of forces on explicit curve.

To handle the moving mesh, we utilize the deformable simplical complex (DSC) framework [7], which provides an explicit curve with the topological adaptivity. Fig. 1 illustrates the DSC algorithm, where the boundary moves iteratively. In each iteration, the segments move as far as possible without making inverted triangles (Fig. 1b), and the conflict-tion are resolved by mesh refinement (Fig. 1c). The DSC may requires several iterations to move the boundary to its destination (Fig. 1a).

Algorithm 1: General algorithm

Input: Label initialization
while *Residual error is large* **do**
  Update pixel probability /* Section 2.1 */
  Compute curve force /* Section 2.2 */
  Deform mesh /* Using the DSC */

2.1. Self-similarity model

2.1.1. Patch-based texture probability

We use dictionary of image patches to compute pixelwise probabilities of pixel belonging to the segment $\mathcal{P}_i : \Omega \rightarrow \mathbb{R}$, $i = 1, \ldots, K$, where Ω is the image domain, and K is the number of segments.

During initialization, we encode self-similarity in the image by extracting patches of size $M \times M$ from the image, and then grouping similar image patches using clustering. The idea is that the patches that group together should belong to the same segment, while each segment can contain many different clusters. Since patches are overlapping, every pixel is influenced by multiple clusters. Likewise, every cluster is influenced by all its patches. The detailed description of this approach can be found in [25].

For curve evolution, the computation of probabilities is based on two steps. Since we know the current labeling of all patches in the image, in the first step we compute the pixelwise label probability for each cluster from the occurrence of a given label in each pixel in the cluster. In the second step we use the label probabilities in the cluster and go back to the image and compute the label probabilities in the image by averaging overlapping image patches.

While our approach is similar to [25], we improve upon [25] by utilizing a variant of k-means trees for clustering, which results in a more efficient implementation of the algorithm.

2.1.2. Improved clustering the image patches

Intensity-based clustering can result in that image patches are not grouped together even though they contain similar patterns. In order to increase the accuracy, in our experience, it is more important to have very large number of clusters than the precision in clustering; therefore, we have chosen to use a k-means tree [26]. A k-means tree is a graph build from consecutive k-means clusterings resulting in a directed rooted tree with a fixed branching factor b and number of layers t.

In order to limit the computational burden and memory usage when building the k-means tree we extract a subset of

patches of size $M \times M$ from the image and collect pixel intensities in vectors of length $\rho = M^2 l$ (l is the number of channels in the image, e.g. $l = 1$ for grayscale images and $l = 3$ for RGB images). These are clustered into b clusters and each cluster center makes up a node in the tree. Repeated k-means clustering for all image vectors belonging to a node results in b new child nodes. This continues for all nodes until the desired number of layers, t, is reached. If a node contains less image vectors than the branching factor b, then no further clustering is carried out and child nodes are marked as empty. In total n tree nodes indexed by $k \in \{1, \ldots n\}$ are obtained.

Each k-means clustering is initialized by choosing a random subset of b image vectors and clustering is obtained by iteratively updating these centers. Our experience is that good performance is obtained without running the k-means until convergence, and therefore a fixed number of iterations is chosen, e.g. 10 iterations.

The outcome of the clustering is an assignment image, which for every pixel indicates to which cluster the patch centered around the pixel belongs to.

2.2. Deformable adaptive mesh for image segmentation

Our deformation model is largely influenced by approaches in [10, 11], which use intensity to derive the displacements of the mesh. Here we use probability maps $\mathcal{P}_i : \Omega \to \mathbb{R}$, $i = 1, \ldots, K$ of pixel belonging to a segment computed from the current segmentation.

Our objective is to find a segmentation $\cup\Omega_i = \Omega$ that minimizes the energy function

$$E(\cup\Omega_i) = \sum_{i=1}^{N} \int_{\Omega_i} (1 - \mathcal{P}_i)^2 \mathrm{d}\Omega + \alpha \mathrm{Length}\Gamma \qquad (1)$$

where Ω is the image domain, and Γ represents boundaries of the segments. This energy function is closely related to the popular Mumford-Shah functional [27].

2.2.1. *Evolving the segmentation boundary*

Minimizing E leads to iterative displacements of boundary vertices $\frac{\partial \mathbf{v}_i}{\partial t} = \mathbf{F}_{\text{ext}}(\mathbf{v}_i) + \mathbf{F}_{\text{int}}(\mathbf{v}_i)$, where we derive the internal force model

$$\mathbf{F}_{\text{int}}(\mathbf{v}_i) = \sum_{\mathbf{v}_j \in \mathcal{N}_i} \frac{\mathbf{v}_j - \mathbf{v}_i}{\|\mathbf{v}_j - \mathbf{v}_i\|}, \qquad (2)$$

where $\mathcal{N}_i$ denotes neighbor vertices of $\mathbf{v}_i$.

The external force of vertex $\mathbf{v}_i$ is

$$\mathbf{F}_{\text{ext}} = \sum_{e \in \mathcal{N}_i} \left\{ \mathbf{n}_e \sum_{\mathbf{v}_j \in e} \left\| \mathbf{v}_j - \mathbf{v}_i \right\| (2 - \mathcal{P}_1 - \mathcal{P}_2)(\mathcal{P}_1 - \mathcal{P}_2) \right\} \qquad (3)$$

Here e denotes a boundary edge, $\mathbf{n}_e$ is the edge normal, and $\mathcal{P}_1, \mathcal{P}_2$ are the probabilities of point $\mathbf{v}_j$ belongs to the two segments separated by e. The reader may refer to [10, 11] for a similar implementation of the computation of the external force.

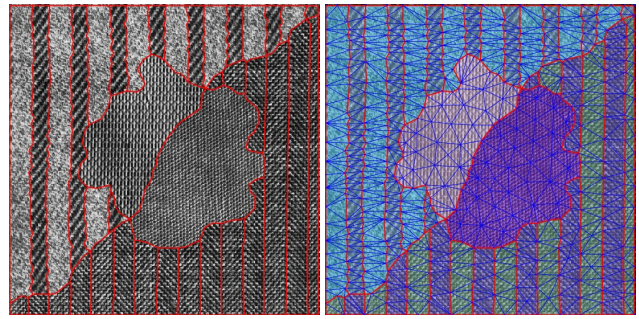


Fig. 2: Multi-label segmentation of a synthetic image

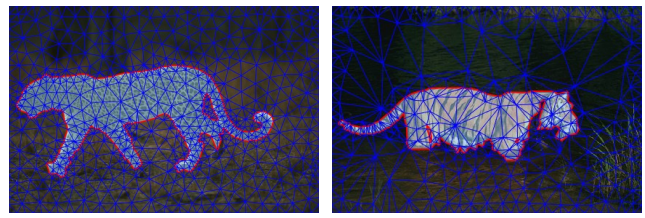


Fig. 3: Adaptive mesh overlaying segmentations

2.2.2. *Relabeling triangles*

Triangle relabeling is a discrete event that assigns a triangle to the label such that the energy in (1) is minimized. This step allows region insertion and speeds up the convergence. Following [10], we perform triangle relabeling after every four steps of mesh evolution for optimal performance.

2.2.3. *Adapting the mesh resolution*

A big advantage of our explicit scheme is that we can define a number of discrete events for adapting the triangle sizes as needed. We follow [10], which starts with a coarse mesh and locally subdivides the triangle, where variance of the probability is high. This approach requires a threshold for triangle variance, which may be difficult to set. The reader may refer to [10] for meshing procedure.

3. RESULTS AND DISCUSSION

Multi-label support is our first advantage (Fig. 2). The method can support unlimited number of labels, defined by user. By using a single mesh, we have the boundary clearly defined, while other methods, e.g. level set method, may have problems with vacuum or overlap.

Adaptive mesh output is our second advantage. In Fig. 3b we show segmentations obtained on natural images with the

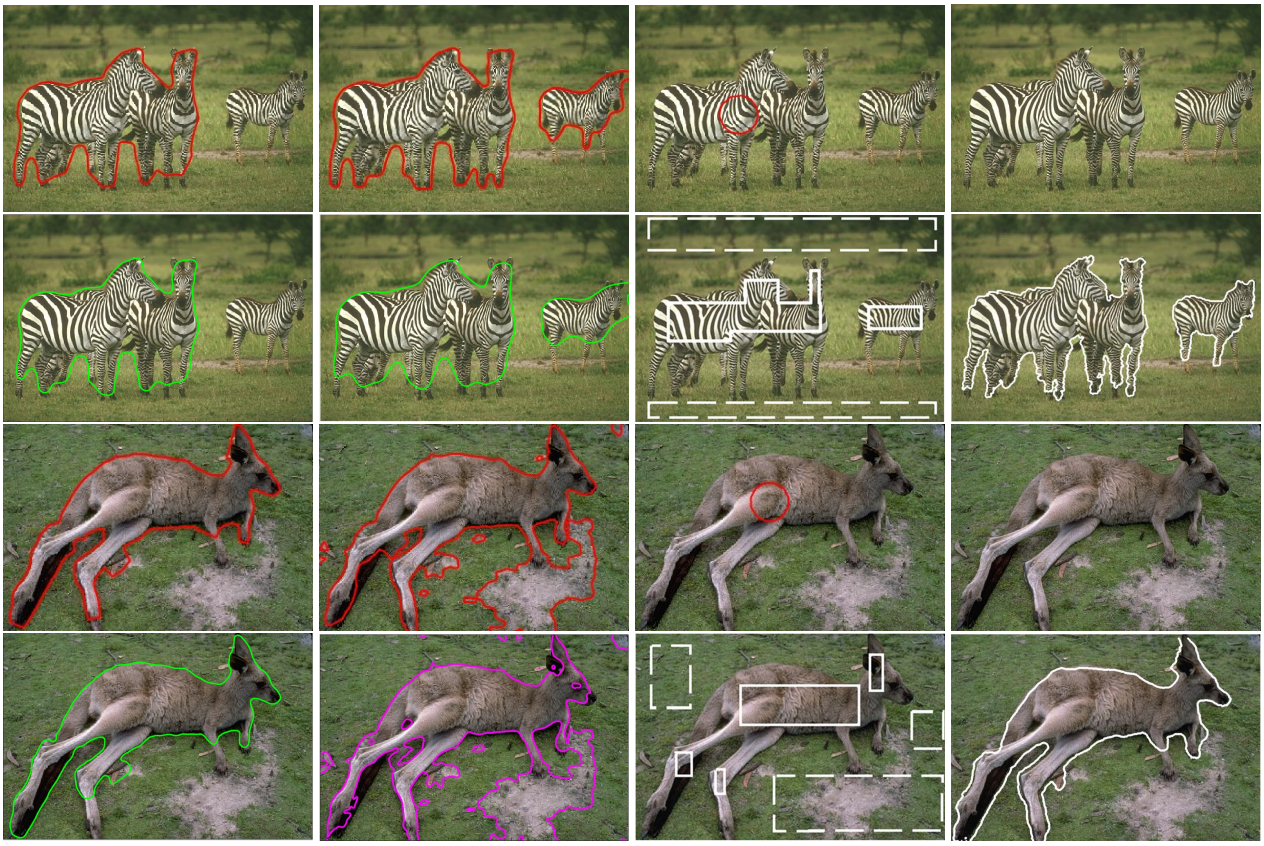


Fig. 4: Segmentation comparison between different methods. There are two rows for each image, where the top row from left shows our method without triangle relabelling, proposed method with triangle relabelling, and initialization (used for the four images from the left), and original. Bottom row shows result by [24], result by [25], initialization and result by [23].

mesh overlayed. Adaptive mesh aids post-processing such as simulations or quantitative analyses.

Fig. 4 shows a small comparison between the proposed method, self-similarity with snakes [24], self-similarity with level set [25], and sparse texture active contour [23]. Compared with [24] and [25] our method can obtain similar results, but outcome of [24] and [25] is dictated by curve representation, while we control the outcome by turning triangle relabeling on or off. If disjoint objects are to be segmented, like the image of zebras, it is advantageous to use triangle relabeling, because it allows the objects of the same type to be segmented to have the same label. However, triangle relabeling also gives more flexibility to the model, which in some cases can be a disadvantage, as seen in the bottom image in Fig. 4, where the grey bottom part of the images is labeled the same as the kangaroo.

Compared with [23] is important to note a substantial amount of information given by their initializations, whereas the limited input in the models employing the patch-based self-similarity model.

All segmentations based on self-similarity use patch size of 5×5, and all methods used the same smoothing factor $\alpha = 2$. Run-times were approximately 20 seconds for the proposed method, 40 seconds for [24] and 25 seconds for [25]. Both [24] and [25] are implemented in Matlab using mex-files implemented in C++ whereas the proposed method is only C++.

4. REFERENCES

[1] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, 1988.

[2] Tony F. Chan and Luminita A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, 2001.

[3] Ravi Malladi, James A Sethian, and Baba C Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1995.

[4] Anthony Yezzi Jr, Andy Tsai, and Alan Willsky, "A statistical approach to snakes for bimodal and trimodal

imagery," in *International Conference on Computer Vision*. IEEE, 1999.

[5] T McInerney and D Terzopoulos, "T-snakes: topology adaptive snakes.," *Medical image analysis*, 2000.

[6] Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk, "Deforming meshes that split and merge," *ACM Transactions on Graphics*, 2009.

[7] Marek Krzysztof Misztal and Jakob Andreas Bærentzen, "Topology-adaptive interface tracking using the deformable simplicial complex," *ACM Transactions on Graphics*, 2012.

[8] Alexandre Dufour, Nicole Vincent, and Auguste Genovesio, "3D Mumford-Shah based active mesh," *Progress in Pattern Recognition, Image Analysis and Applications*, 2006.

[9] Vedrana Andersen Dahl, Asger Nyman Christiansen, and Jakob Andreas Bærentzen, "Multiphase Image Segmentation Using the Deformable Simplicial Complex Method," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2014.

[10] Tuan T. Nguyen, Vedrana A. Dahl, and J. Andreas Bærentzen, "Multi-phase image segmentation with the adaptive deformable mesh," *Pattern Recognition Letters*, 2019.

[11] Tuan T Nguyen, Vedrana Andersen Dahl, Jakob Andreas Bærentzen, and Camilla Himmelstrup Trinderup, "Multi-phase Volume Segmentation with Tetrahedral Mesh," in *British Machine Vision Conference (BMVC)*, 2018.

[12] Mikaël Rousson, Thomas Brox, and Rachid Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2003.

[13] Thomas Brox and Joachim Weickert, "A TV flow based local scale estimate and its application to texture discrimination," *Journal of Visual Communication and Image Representation*, 2006.

[14] Thomas Brox, Mikaël Rousson, Rachid Deriche, and Joachim Weickert, "Colour, texture, and motion in level set based segmentation and tracking," *Image and Vision Computing*, 2010.

[15] Manik Varma and Rahul Garg, "Locally invariant fractal features for statistical texture classification," in *International Conference on Computer Vision*. IEEE, 2007.

[16] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2005.

[17] Timo Ojala, Matti Pietikainen, and Topi Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2002.

[18] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi, "Contour and texture analysis for image segmentation," *International Journal of Computer Vision*, 2001.

[19] Michael Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*, Springer, 2010.

[20] Julien Mairal, Francis Bach, and Jean Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[21] Gabriel Peyré, "Sparse modeling of textures," *Journal of Mathematical Imaging and Vision*, 2009.

[22] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman, "Discriminative learned dictionaries for local image analysis," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.

[23] Yi Gao, Sylvain Bouix, Martha Shenton, and Allen Tannenbaum, "Sparse texture active contour," *IEEE Transactions on Image Processing*, 2013.

[24] Anders Bjorholm Dahl and Vedrana Andersen Dahl, "Dictionary snakes," in *2014 22nd International Conference on Pattern Recognition (ICPR)*. IEEE, 2014.

[25] Anders Bjorholm Dahl and Vedrana Andersen Dahl, "Dictionary based image segmentation," in *19th Scandinavian Conference on Image Analysis (SCIA)*. Springer, 2015.

[26] David Nister and Henrik Stewenius, "Scalable recognition with a vocabulary tree," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2006.

[27] David Mumford and Jayant Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, 1989.