Dictionary Snakes

Dahl, Anders Bjorholm; Dahl, Vedrana Andersen

Published in:
Proceedings of the 22nd International Conference on Pattern Recognition (ICPR 2014)

Link to article, DOI:
10.1109/ICPR.2014.34

Publication date:
2014

Document Version
Peer reviewed version

Citation (APA):
Dahl, A. B., & Dahl, V. A. (2014). Dictionary Snakes. In L. O'Conner (Ed.), *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR 2014)* (pp. 142-147). IEEE. https://doi.org/10.1109/ICPR.2014.34

Dictionary Snakes

Anders Bjorholm Dahl
Vedrana Andersen Dahl
DTU Compute
Technical University of Denmark
{abda,vand}@dtu.dk

***Abstract*—Visual cues like texture, color and context make objects appear distinct from the surroundings, even without gradients between regions. Texture-rich objects are often difficult to segment because algorithms need advanced features which are unique for the image. In this paper we suggest a method for image segmentation that operates without training data. Our method is based on a probabilistic dictionary of image patches coupled with a deformable model inspired by snakes and active contours without edges. We separate the image into two classes based on the information provided by the evolving curve, which moves according to the probabilistic information obtained from the dictionary. Initially, the image patches are assigned to the nearest dictionary element, where the image is sampled at each pixel such that patches overlap. The curve divides the image into an inside and an outside region allowing us to estimate the pixel-wise probability of the dictionary elements. In each iteration we evolve the curve and update the probabilities, which merges similar texture patterns and pulls dissimilar patterns apart. We experimentally evaluate our approach, and show how textured objects are precisely segmented without any prior assumptions about image features. In addition, a texture probability image is obtained.**

I. INTRODUCTION

Images rich in texture often have no well-defined intensity gradient between regions. Without gradients in the image domain the texture must be characterized by considering a neighborhood of pixels. A typical approach is to map the image to a feature domain, and if this mapping is successful, texture regions become distinct with high gradients in between. Successful mapping, however, is dependent on the choice of features, and inadequate features might severely reduce the segmentation quality with a need for a high degree of regularization.

A popular technique is to regularize the segmentation by explicitly modeling the boundary. Kass et al. [1] introduced an active contour model called snakes based on an evolving curve. The curve is deformed by external and internal forces. External forces are obtained from the image and make the curve move towards positions with high gradients. The internal forces are determined by the shape of the curve and include first and second order spline terms – the so called membrane and plate terms. These terms are intuitive and can be weighted to govern the elasticity and stiffness of the curve. Yezzi Jr et al. [2] and Chan and Vese [3] (see also [4], [5]) suggest an active contour model, which evolves the curve using level sets to minimize the Mumford-Shah functional [6]. Hereby, they obtain a robust and flexible segmentation method that also may handle textured objects by a low-dimensional feature representation using image derivatives.

Similarly to image derivatives fractal dimension is a low-dimensional representation that has shown strong texture representation properties [7], [8]. An advantage of having a low-dimensional texture characterization is that gradients are often well-defined and contour evolution can be conducted in a similar manner as with intensity images. However, such a low-dimensional representation has limited discriminative power compared to higher dimensional representations. A simple higher dimensional representation of a texture is the intensity histogram, which has been used for active contour models [9], [10]. The histogram has the advantage of being easy to compute, but the discriminative properties of histograms are still limited. A histogram is the first order statistics of an image, where the gray level co-occurrence matrix is a method for characterizing second order image statistics [11]. The gray level co-occurrence features are typically used as high-dimensional descriptors. Other high-dimensional descriptors are based on filter response methods, e.g. Gabor filters [12]. Scale space methods also show good performance for texture characterization, e.g. basic image features are designed for texture characterization [13]. Local binary patterns is another example of high-dimensional texture descriptor with good discriminative properties [14] and has been used with active contours in [15]. One problem with a high-dimensional features is that most features are far apart in feature space, which makes gradients less well-defined and therefore segmentation methods require higher degree of regularization, e.g. using a Markov random field [16].

Sparse coding has been used in a range of image processing and analysis tasks [17] including texture segmentation [18]. The sparse image representation is a generative image reconstruction model, where an image is coded using an overcomplete dictionary of exemplar image patches. An image patch is reconstructed using a linear combination of a few basis vectors from the dictionary. The idea of sparse coding for texture segmentation is to build a dictionary for each of the texture classes, and then choose the class with least reconstruction error. Here, the method for generating the dictionary significantly influences the performance, where improvements have been obtained by optimizing for discriminative properties [19], [20] instead of reconstructive [21], [22].

Gao et al. [23] recently suggested a segmentation procedure using sparse texture and active contours. By utilizing rough user input for foreground and background regions, they are able to build a sparse dictionary for the two image classes. Then they evolve an active contour using the residual error of the reconstructions. This allows for high flexibility in the type of image that can be segmented.

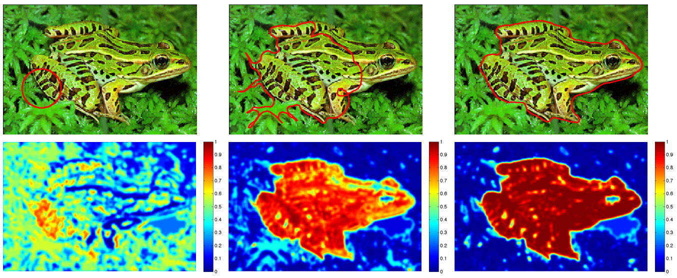


Fig. 1. Segmentation example. Top row shows the countour and bottom row shows the pixelwise probability for inside. First column is the initialization, middle column is after 20 iterations and right is after 50 iterations when the segmentation has converged. Note how the probability for inside changes to cover the segment with the frog. The parameters were patch size of 5, dictionary size of 250, 200 curve points, and no normalization.

We present a method that is based on ideas from both active contours, and texture characterization using a dictionary of exemplar image patches like for sparse texture segmentation. However, we employ an additional probability dictionary, which encodes the probability of dictionary elements, similar to [24]. A segmentation example is shown in Figure 1 and the details of our method are described below.

II. METHOD

Our segmentation algorithm is inspired by parametric active contours [1] and region-based segmentation approaches [2], [3]. In region-based approaches, intensity images are segmented by minimizing the Mumford-Shah potential, but instead of using pixel intensity we suggest using pixel probability. This lets us segment both texture and intensity. Information from a partitioning obtained from the curve makes it possible to estimate the probability of a pixel belonging to the regions inside and outside. These probabilities allow a curve evolution that pulls apart dissimilar textures. Before explaining the details we will briefly describe the snakes model and the region-based curve evolution.

A. *Snakes and curve evolution*

In the snakes [1] formulation the closed curve is defined as $\mathbf{X} : [0,1] \to \Omega \subset \mathbb{R}^2$ where $\mathbf{X}(s) = [x(s), y(s)]$ for $s \in [0,1]$ is a point on the curve and $\mathbf{X}(0) = \mathbf{X}(1)$. This curve is dynamic, i.e. it is also a function of time t, and it evolves under the influence of external and internal forces

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{F}_{\text{int}}(\mathbf{X}) + \mathbf{F}_{\text{ext}}(\mathbf{X}) \,. \tag{1}$$

The internal force discourages stretching and bending by minimizing the membrane potential and the plate potential of the curve

$$\mathbf{F}_{\text{int}}(\mathbf{X}) = \frac{\partial}{\partial s}\left(\alpha \frac{\partial \mathbf{X}}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 \mathbf{X}}{\partial s^2}\right) \,, \tag{2}$$

where α and β are regularization weights.

Considering the external forces, an object with sharp edges has large gradients along its boundary, making it possible to define a Gaussian potential that will adequately guide the curve towards the boundary of the object. The resulting external force is

$$\mathbf{F}_{\text{ext}}(\mathbf{X}) = \nabla \left[\nabla \left(G_\sigma * Q\right)\right]^2 \,, \tag{3}$$

where G_σ is the Gaussian kernel with standard deviation σ.

Instead of the Gaussian potential, which is dependent on high image gradients, Yezzi Jr et al. [2] and Chan and Vese [3] suggested to evolve the curve considering region information provided by the curve. With this they minimize the piecewise constant Mumford-Shah functional. In an images with two distinct means, the region-based external force is given by

$$\mathbf{F}_{\text{ext}}(\mathbf{X}) = (\mu_2 - \mu_1)(Q - \mu_2 + Q - \mu_1)\mathbf{N} \,, \tag{4}$$

where $Q(x,y)$ is an intensity image, and $\mathbf{N}$ is an inwards pointing normal. Values μ_1 and μ_2 are the mean intensities for the areas inside and outside the curve repeatedly evaluated as the curve evolves. Hereby, a point on the curve will move outwards if its intensity is closer to the inside mean and vice versa, leading to a segmentation that separates the two intensity means.

Both the Gaussian potential and the regional approach are based on the intensity difference between image segments. The difference between segments is however often based on texture. Our suggestion is therefore to obtain a texture based external energy, utilizing probabilities of textures obtained by the evolving curve.

B. *External dictionary energy*

In a manner similar to region-based approaches [2], [3] we extract the inside and outside information from the current curve position. We pass this information trough a dictionary of image patches and obtain a current texture probability image. The texture probabilities then define forces on the curve.

We start by describing the construction of a probability image given a curve. The image to be segmented is denoted $Q : \Omega \to \mathbb{R}$ (RGB case is covered later). The curve determines a label image $L_{\text{in}} : \Omega \to \{0,1\}$ given by

$$L_{\text{in}}(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \Omega_{\text{in}} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where Ω_{in} is the part of Ω inside the closed curve $\mathbf{X}$.

To handle textured images we consider a window of size $m = M \times M$ around each grid position, both in the intensity image Q and in the label image L_{in}. We have chosen to concatenate the collected values in vectors. As a result, for each grid position (except close to the image boundary) we have sampled intensities in $\mathbf{q}(x,y) \in \mathbb{R}^m$ and sampled labels in a binary vector $\mathbf{l}(x,y) \in \{0,1\}^m$. We will refer to $\mathbf{q}$ and $\mathbf{l}$ as image and label patches respectively, to make the description of the model clear, even though they in our mathematical notation are vectors.

We model the texture using a dictionary consisting of an intensity dictionary and a label dictionary $\mathbf{D} = (\mathbf{D}_Q, \mathbf{D}_L)$. The intensity dictionary is fixed – here we obtain it by clustering. Dictionary elements are arranged in a matrix $\mathbf{D}_Q \in \mathbb{R}^{m \times n}$ and each image patch is assigned to the nearest dictionary element using the Euclidian distance

$$I(x,y) = \arg\min_i ||\mathbf{q}(x,y) - \mathbf{d}_Q(i)||_2^2, \; i \in \{1, ..., n\} \,, \tag{6}$$

where $\mathbf{d}_Q(i)$ is the i'th column of $\mathbf{D}_Q$. By doing so we obtain a dictionary assignment image $I : \Omega \to \mathbb{N}$.

Just as we have a label image L_{in} corresponding to an intensity image Q, we want to compute a label dictionary $\mathbf{D}_L \in [0,1]^{m\times n}$ corresponding to an intensity dictionary, so that $\mathbf{D}_L$ contains a pixel-wise probability of belonging to inside. For a given label image (i.e. a curve) the sample frequency of each dictionary element can be estimated as the average of assigned label patches

$$\mathbf{d}_L(i) = \frac{1}{|S(i)|} \sum_{(x,y)\in S(i)} \mathbf{l}(x,y)\,, \tag{7}$$

where $S(i)$ are the positions of image patches assigned to the ith dictionary element and $|S(i)|$ is the number of elements in $S(i)$. Note that the dictionary label patches are no longer binary and they indeed contain an observed frequency of being inside. The label dictionary is therefore also refered to as probability dictionary.

In the case of RGB images we collect intensities from all three color channels, resulting in intensity patches three times longer than label patches. The fundamentals of the method and all computations are the same for any number of color channels, so we refrain from describing a RGB case in more detail.

Having obtained the probabilities of dictionary elements we go back to the image space and construct a probability image $P_{\text{in}} : \Omega \to [0,1]$. The probability image is build as the mean of overlapping label patches from $\mathbf{D}_L$ which are placed in a grid according to the dictionary assignment $I(x,y)$. Note that P_{in} is different from L_{in}, because image patches from both inside and outside can be assigned to the same dictionary element. This results in confusion with probabilities different from zero and one. We will utilize this confusion to drive the curve evolution.

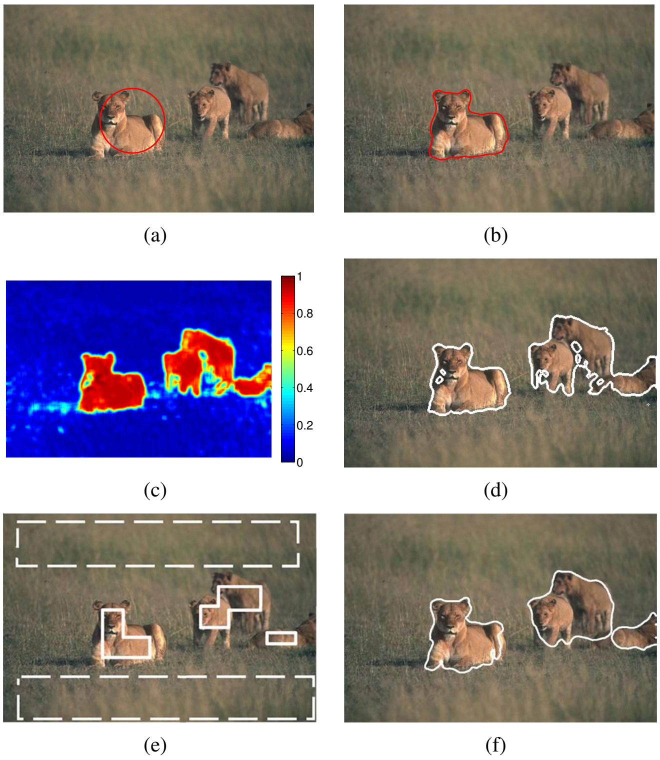
A probability image defines forces deforming the curve $\mathbf{X}$. Curve points at locations with large P_{in} should move outwards, curve points with large $P_{\text{out}} = 1 - P_{\text{in}}$ should move inwards, and the curve should converge in a band where $P_{\text{in}} = P_{\text{out}}$. This motivates us to consider a force with the magnitude $\ln(P_{\text{in}}/P_{\text{out}})$. However, our probability estimation is biased by the ratio between the area of the inside and outside. If $\mathbf{X}$ encloses a relatively small part of Ω, estimated probability of being inside will also be small, which will lead to further shrinkage of the curve and eventually result in a trivial solution with all pixels being outside. To avoid the trivial solutions we normalize the inside and the outside probability with the area of inside $|\Omega_{\text{in}}|$ and outside $|\Omega_{\text{out}}| = |\Omega| - |\Omega_{\text{in}}|$ respectively. Consequently, the external force takes the form of

$$\mathbf{F}_{\text{ext}}(x,y) = \ln\left(\frac{|\Omega_{\text{out}}|}{|\Omega_{\text{in}}|}\,\frac{P_{\text{in}}(x,y)}{P_{\text{out}}(x,y)}\right)\mathbf{N}\,. \tag{8}$$

C. Algorithm

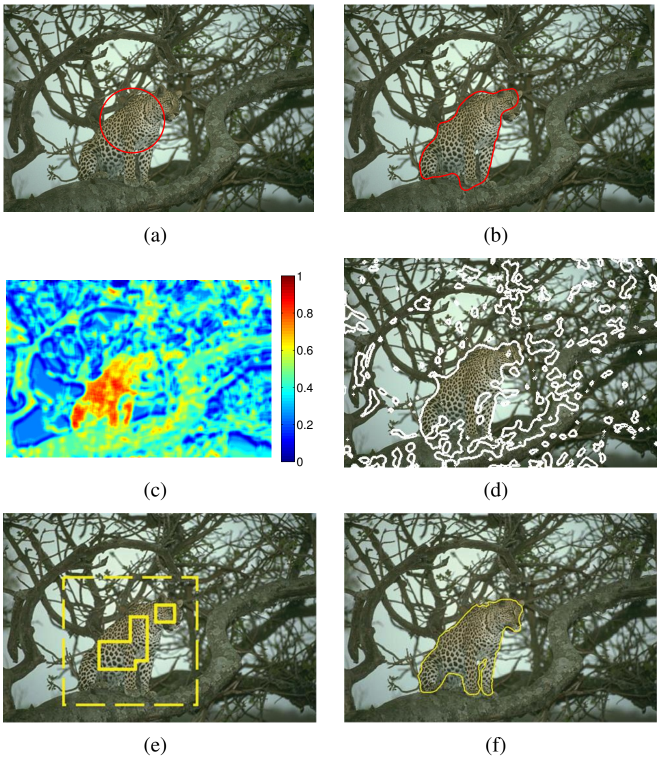
The algorithm for the dictionary snakes is initialized with an image Q, initial curve $\mathbf{X}^0$, and the set of dictionary parameters: patch size M, dictionary size n and the normalization flag. Parameters for evolving the curve are: internal force weights (α and β), and time step size Δ.

Initially all patch vectors $\mathbf{q}$ are extracted from Q, and an intensity dictionary $\mathbf{D}_Q$ is build by k-means clustering.



(a) (b) (c) (d) (e) (f)

Fig. 2. Lion image. (a) The initial curve, (b) the final curve, (c) probability image, (d) edges of thresholded probability image, (e) initialization of [23], (f) result of [23].



(a) (b) (c) (d) (e) (f)

Fig. 3. Snow leopard image. (a) The initial curve, (b) the final curve, (c) probability image, (d) edges of thresholded probability image, (e) initialization of [23], (f) result of [23].

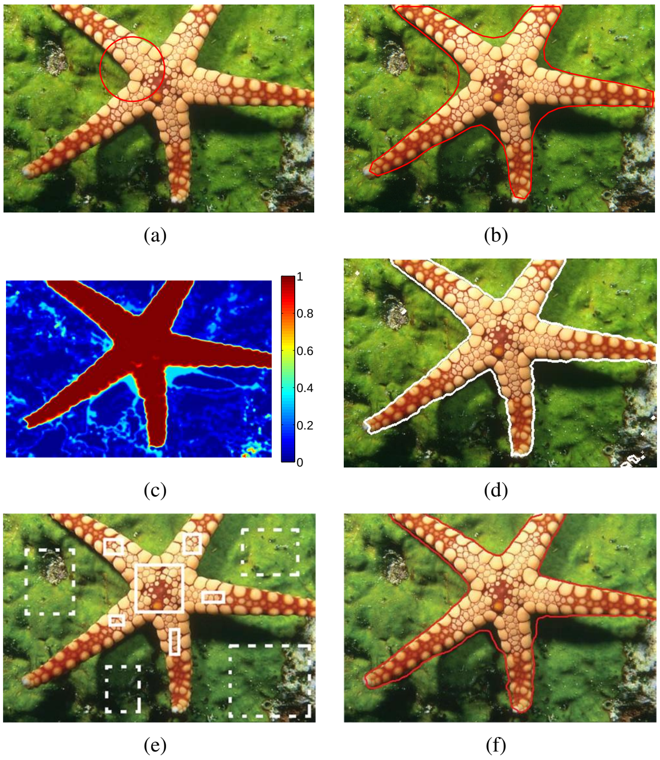


Fig. 4. Sea star image. (a) The initial curve, (b) the final curve, (c) probability image, (d) edges of thresholded probability image, (e) initialization of [23], (f) result of [23].

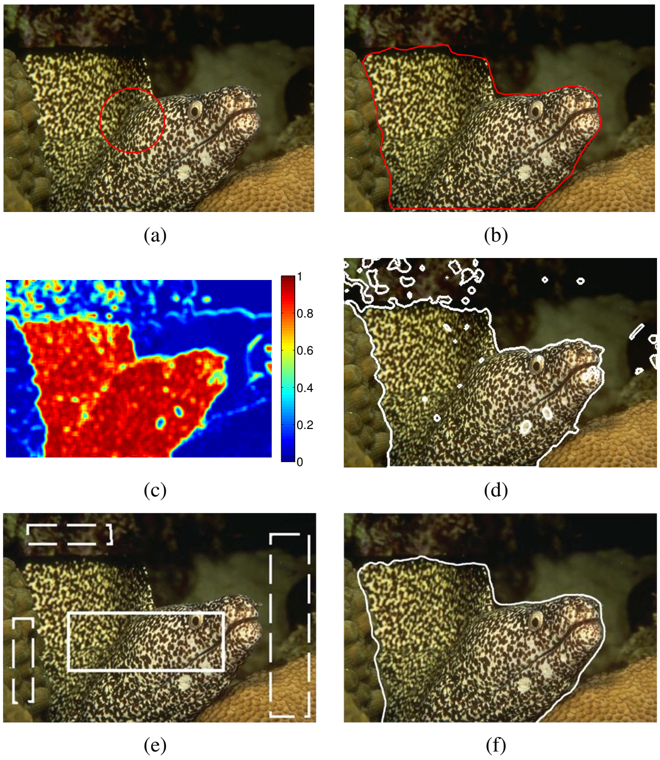


Fig. 5. Fish image. (a) The initial curve, (b) the final curve, (c) probability image, (d) edges of thresholded probability image, (e) initialization of [23], (f) result of [23].

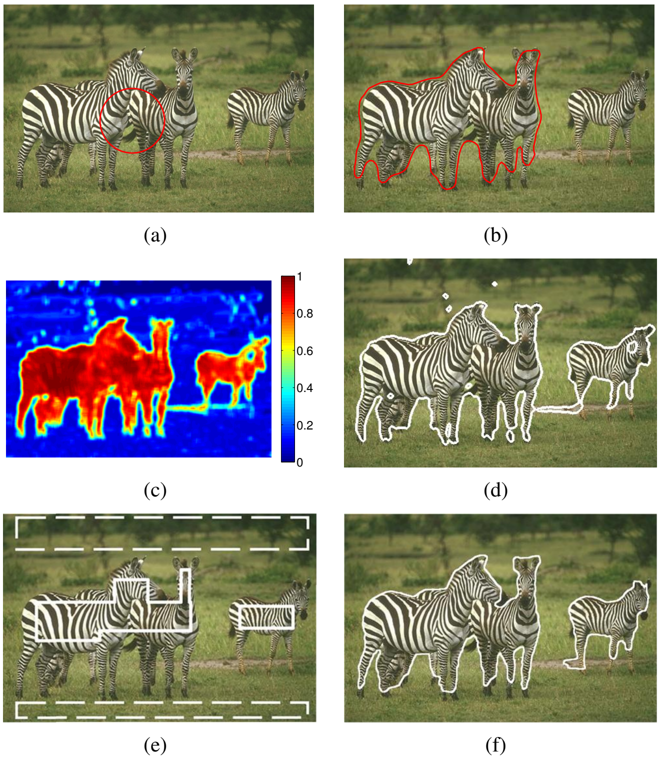


Fig. 6. Zebra image. (a) The initial curve, (b) the final curve, (c) probability image, (d) edges of thresholded probability image, (e) initialization of [23], (f) result of [23].

Then all image patches are assigned to the nearest dictionary element $\mathbf{d}_Q$ using (6). Normalization indicates whether the image patches have been normalized to unit length.

After the initialization the curve is iteratively evolved. In each evolution step the label image L_{in} is updated using (5) and probability dictionary $\mathbf{D}_L$ is estimated using (7). The external energy is calculated for all curve positions using (8). Finally the curve is updated using (1), (2) and

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \Delta \frac{\partial \mathbf{X}}{\partial t} \tag{9}$$

either for fixed number of steps or until a convergence criterion is met, e.g. given by $\partial \mathbf{X}^* / \partial t < \epsilon$, where ϵ is a small number. When the convergence criterion is met the final label probability image P_{in}^* containing the pixel-wise texture probabilities can be obtained together with the final segmentation curve $\mathbf{X}^*$.

To ensure stability of the curve evolution we avoid self-intersections by reordering the nodes if the curve folds. The rule is, that points in the smallest fold are reordered. In addition, we redistribute the points equidistantly along the curve in each iteration, using linear interpolation between curve points. This allows us to capture the high curvature regions. Both these steps are included to speed up the evolution, because they allow for fewer points and larger step sizes.

We have applied this algorithm to a number of images in order to qualitatively investigate its performance and demonstrate the possibilities of using the dictionary snakes. The investigation demonstrates the strong texture modeling properties of the dictionary snakes.

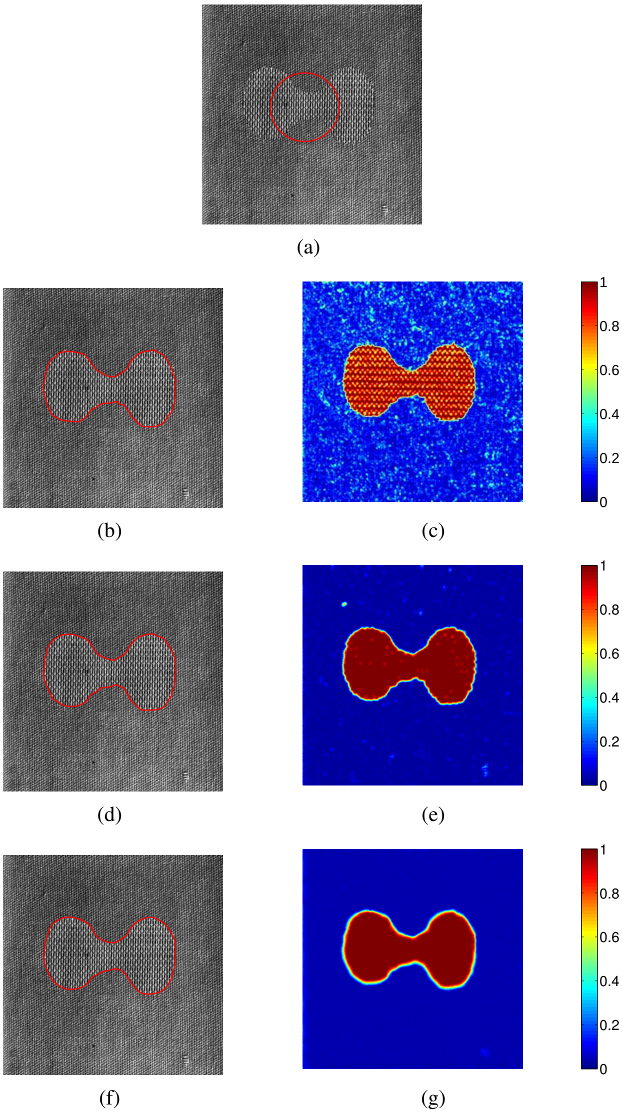


Fig. 7. Easy texture image. (a) The initial curve, (b,c) the final curve and probability image for patch size 3, (d,e) patch size 5, (f,g) patch size 9.

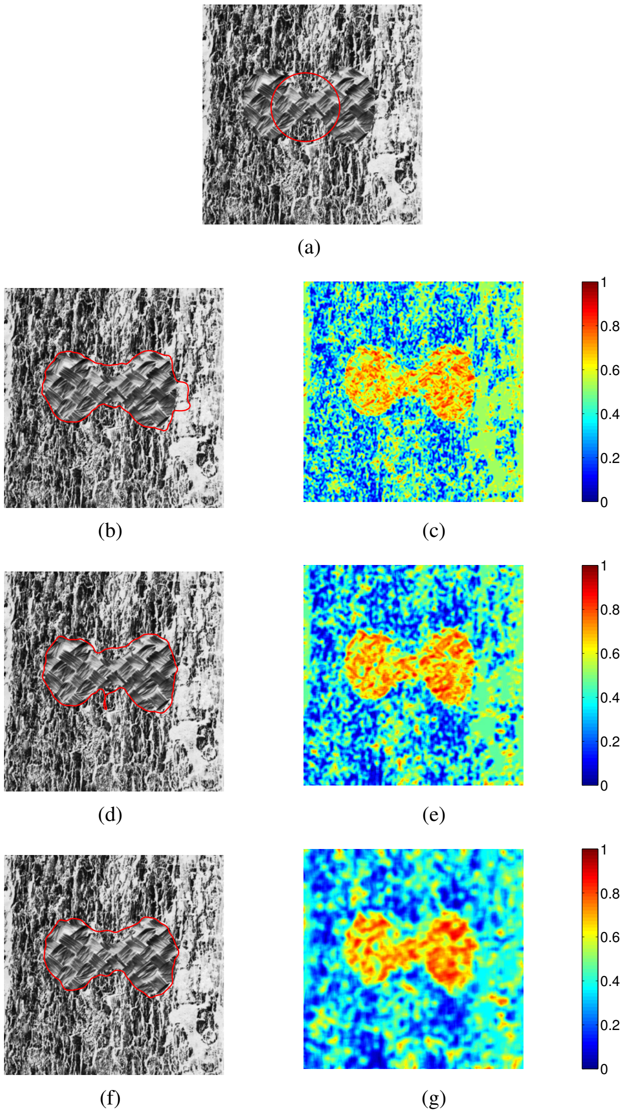


Fig. 8. difficult texture image. (a) The initial curve, (b,c) the final curve and probability image for patch size 3, (d,e) patch size 5, (f,g) patch size 9.

III. EXPERIMENTS AND RESULTS

First we show how the method performs on a number of standard texture examples of real world scenes. Secondly, we use two composed textures based on the Brodatz textures [25] to demonstrate the effect of varying the patch size of the dictionary.

Fig. 2–6 show the performance of our method on a number of photographs also used in Gao et al. [23], which allows us to compare the results. We initialize the curve using the circle shown in (a) and we let the curve evolve for a fixed number of iterations. Hereby we obtain a final curve (b), the probability image (c), and the segmentation by thresholding the probability image (d). The results from [23] are shown in (e) and (f). In Tab. I we list the parameters used for obtaining the results shown in the figures. In all experiments we set $\alpha = 1$, $\beta = 0$, and set step size of $\Delta = 5$ for the first 40 iterations, and $\Delta = 1$ for another 30 iterations to assist convergence. The large step size is chosen to speed up the convergence and the small steps make a final adjustment. Some images converged much faster than the 70 iterations.

TABLE I. PARAMETERS USED IN FIG. 2–6.

	Patch size M	Dict. size n	# curve points	Normalization
Lion	9	250	100	yes
Sea star	5	250	100	no
Snow leopard	9	250	100	yes
Fish	9	250	100	yes
Zebra	9	500	200	yes

Fig. 7 and 8 show segmentation of two composed textures where we vary the size of the image patches using $M = \{3, 5, 9\}$ pixels. All these segmentations are with $\alpha = 1$ and step size of $\Delta = 5$ for the first 40 iterations and $\Delta = 1$ for another 30 iterations. The easy example converged within a few iterations, whereas the difficult example just converged within the 70 iterations.

As for the time performance, the step consuming most time is the k-means clustering. Therefore we chose to randomly extract 5000 image patches for building the dictionary. The clustering then takes at most around a minute for the large patch sizes, whereas for $M = 3$ it takes only a few seconds.

IV. DISCUSSION AND CONCLUSION

We have suggested a segmentation method based on a deformable model using a probabilistic dictionary of image patches. The method is not constrained by any user provided texture, but operates in a manner similar to active contours by [2], [3]. Our method generalizes their intensity separation to include textures as well.

Our method shows very good results for a large number of segmentation tasks, which we illustrate in our experimental section. We obtain performance comparable to the state-of-the-art method by Gao et al. [23], despite using far less restrictions in our modeling. Segmentation of [23] is assisted by the initial user provided foreground and background segments, whereas in our method the user just provides an initial curve. Our comparison to the work of [23] is more an attempt to illustrate the possibilities of the dictionary snakes than making a precise assessment. In this spirit, the parameters for our experiments were chosen to illustrate the possibilities, but without involved parameter optimization.

The experimental results show, that sometimes the boundary of the curve gives the best segmentation, like in the snow leopard or the fish example, whereas in others, like the sea star, the probability image gives the best result. In the composed examples we see a performance difference between an easy texture and a difficult texture, as well as the influence of patch size. Small patches generally give a more noisy probability image, but with sharper details, where larger patch sizes will generally smooth the result. This smoothing effect is an advantage for the difficult texture example, whereas for the easy example larger patches would potentially smooth away details.

Our method has similarities to the sparse coding used for texture segmentation in e.g. [20], [23]. However, instead of learning a dictionary based on discriminative criteria, we perform k-means clustering. Furthermore, instead of using a linear combination of dictionary vectors, we assign an image patch to the nearest dictionary element, more in a bag-of-features like manner. Still, we obtain results as good as state-of-the-art, due to the use of label probability associated with each dictionary element similar to [24].

We have chosen to initialize the curve to cover both foreground and background regions, but our experience is that dictionary snakes method is robust to the initialization. If the initial curve is fully outside the object of interest, the segmentation will either separate distinct background textures (sometimes not seen obvious from a user perspective), or try to surround the object from the outside. In the latter case the convergence is slow.

In conclusion, we suggest a segmentation method that generalizes the active contour method by Yezzi Jr et al. [2] and Chan and Vese [3] to include textures as well as intensities. The original method evolves a contour for pulling intensities from each other – we do the same, just generalized to textures.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *IJCV*, vol. 1, no. 4, pp. 321–331, 1988.

[2] A. Yezzi Jr, A. Tsai, and A. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery," in *ICCV*, vol. 2. IEEE, 1999, pp. 898–903.

[3] T. F. Chan and L. A. Vese, "Active contours without edges," *TIP*, vol. 10, no. 2, pp. 266–277, 2001.

[4] A. Tsai, A. Yezzi Jr, and A. S. Willsky, "Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification," *TIP*, vol. 10, no. 8, pp. 1169–1186, 2001.

[5] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *IJCV*, vol. 50, no. 3, pp. 271–293, 2002.

[6] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[7] M. Varma and R. Garg, "Locally invariant fractal features for statistical texture classification," in *ICCV*. IEEE, 2007, pp. 1–8.

[8] Y. Xu, H. Ji, and C. Fermuller, "A projective invariant for textures," in *CVPR*, vol. 2. IEEE, 2006, pp. 1932–1939.

[9] A. Adam, R. Kimmel, and E. Rivlin, "On scene segmentation and histograms-based curve evolution," *TPAMI*, vol. 31, no. 9, pp. 1708–1714, 2009.

[10] O. Michailovich, Y. Rathi, and A. Tannenbaum, "Image segmentation using active contours driven by the bhattacharyya gradient flow," *TIP*, vol. 16, no. 11, pp. 2787–2801, 2007.

[11] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, no. 6, pp. 610–621, 1973.

[12] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi, "Integrated active contours for texture segmentation," *TIP*, vol. 15, no. 6, pp. 1633–1646, 2006.

[13] M. Crosier and L. D. Griffin, "Using basic image features for texture classification," *IJCV*, vol. 88, no. 3, pp. 447–460, 2010.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *TPAMI*, vol. 24, no. 7, pp. 971–987, 2002.

[15] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Implicit active contours driven by local binary fitting energy," in *CVPR*. IEEE, 2007, pp. 1–7.

[16] S. C. Zhu, Y. Wu, and D. Mumford, "Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling," *IJCV*, vol. 27, no. 2, pp. 107–126, 1998.

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

[18] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *TPAMI*, vol. 34, no. 4, pp. 791–804, 2012.

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *NIPS*, 2008.

[20] ——, "Discriminative learned dictionaries for local image analysis," in *CVPR*. IEEE, 2008, pp. 1–8.

[21] G. Peyré, "Sparse modeling of textures," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17–31, 2009.

[22] K. Skretting and J. H. Husøy, "Texture classification using sparse frame-based representations," *EURASIP journal on applied signal processing*, vol. 2006, pp. 102–102, 2006.

[23] Y. Gao, S. Bouix, M. Shenton, and A. Tannenbaum, "Sparse texture active contour," *TIP*, 2013.

[24] A. L. Dahl and R. Larsen, "Learning dictionaries of discriminative image patches," in *BMVC*, 2011.

[25] P. Brodatz, *Textures: a photographic album for artists and designers*. Dover New York, 1966, vol. 66.