A texture-based energy for active contour image segmentation

Daniel Reska, Cezary Boldak, and Marek Kretowski

Faculty of Computer Science
Bialystok University of Technology, Bialystok, Poland,
{d.reska, c.boldak, m.kretowski}@pb.edu.pl

Summary. This paper presents a two-dimensional deformable model-based image segmentation method that integrates texture feature analysis into the model evolution process. Typically, the deformable models use edge and intensity-based features as the influencing image forces. Incorporation of the image texture information can increase the methods effectiveness and application possibilities. The algorithm generates a set of texture feature maps and selects the features that are best suited for the currently segmented region. Then, it incorporates them into the image energies that control the deformation process. Currently, the method uses the Grey Level Co-occurrence Matrix (GLCM) texture features, calculated using hardware acceleration. The preliminary experimental results, compared with outcomes obtained using standard energies, show a clearly visible improvement of the segmentation on images with various texture patterns.

1 Introduction

Efficient and robust image segmentation is one of the main challenges in computer vision. Deformable models [1] are a widely used class of segmentation methods. Generally, a deformable model is an active shape (e.g., a 2D contour or a 3D surface) that tries to adapt to a specific image region. This adaptation process is influenced by external and internal forces that deform the shape towards the boundaries of the segmented region. The external forces attract the model to desired image features, while the internal forces control its smoothness and continuity. This formulation allows the models to overcome many problems, like image noise and boundary irregularities. External forces can come from a variety of image features. The most common approaches use the edge or intensity statistics of the segmented region [2, 3, 4]. This makes the deformable models well suited for extraction of areas with distinctive (but not necessary continuous) borders and fairly uniform texture. However, larger patterns with high contrast pose a greater challenge to traditional methods.

Deformable models were introduced with a seminal "snake" algorithm [2], which was a parametric active contour with an edge-based image force and a

R.S. Choras (ed.), *Image Processing & Communications Challenges 6*,
Advances in Intelligent Systems and Computing Volume 313

set of internal energies that controlled bending and stretching of the curve. Since then, the classical model has been heavily modified and extended, e.g., with addiction of expansion forces [5], edge-based vector field energies [3], adaptive topology mechanisms [6] and region-based image energies [4, 7]. Furthermore, methods combining deformable models with texture extraction and classification [8] were also introduced. Texture-based methods generally fall into two categories: utilising information obtained from supervised texture analysis, or using the calculated features without earlier classification. The supervised methods can require an initial analysis stage, which provides information about the number and characteristics of distinctive patterns in the image [9], or they can use a selected set of features to create a deformation map that influence the evolution of the model [10]. Unsupervised methods can extract the required information using the initial region intensity statistics to create a likelihood deformation map [11], or, in case of patters with a larger scale, use a small bank of Gabor filters to extract the features [12, 13]. Texture features can also be used to improve the segmentation process alongside other image characteristics [14].

In this paper, we present a texture-based image energy for a two-dimensional parametric active contour. The energy incorporates texture features into our template active contour framework [15]. The proposed energy can be used alongside other image forces, expanding the application possibilities of the method. Currently, the energy utilises features based on Gray-Level Co-occurrence Matrices (GLCM) [16]. The features are generated using hardware-accelerated implementation, which allows an efficient creation of a large feature set. The preliminary experiments were performed on synthetic images with various texture patterns and show a visible improvement over typical image energies.

2 Texture-based active contour

The proposed algorithm incorporates a texture-based image energy into a discrete parametric snake [17] with an adaptive topology reformulation abilities [6]. The energy makes the snake to expand into a region with a uniform texture that is similar to the initial region of the contour.

Firstly, the snake is manually initialised inside the segmented region. Next, the algorithm generates a set of texture feature maps and selects the features that are best suited for the currently segmented area. Finally, the contour evolution process expands the snake under the influence of the texture energy, which is based on the selected features. This algorithm can be presented in a pseudocode:

Require: Initialisation of the snake $\mathbf{s}$

Generate the initial texture feature maps set $\mathbf{T}_{init}$

Initialise empty set $\mathbf{T}_{best}$ for selected texture feature maps

for all texture feature map $\mathbf{t} \in \mathbf{T}_{init}$ **do**

if $\mathbf{t}$ meets the selection condition **then**
    add $\mathbf{t}$ to $\mathbf{T}_{best}$
end if
end for
Create external energy $\mathbf{E}_{tex}$ using $\mathbf{T}_{best}$
repeat
for all snake point $\mathbf{p} \in$ contour $\mathbf{s}$ **do**
    Minimise the local energy of $\mathbf{p}$ using $\mathbf{E}_{tex}$ and other energies
end for
until convergence of snake $\mathbf{s}$
return the segmented region.

The detailed descriptions of the algorithm steps are presented below.

Initialisation

Initialisation of the contour is the crucial step in the current version of our algorithm. The initial snake should be manually placed inside the segmented area and in a region of a possibly uniform texture. Furthermore, the contour area should be large enough to cover the texture pattern and capture the entirety of its characteristics, preferably by including several texture pattern tiles.

Texture feature maps generation and selection

The next step of our algorithm is generation of the texture feature maps. The features are calculated for each pixel of the source image, that results in a set of feature maps.

The currently used texture features, generated from the Grey-Level Co-occurrence Matrix, are: Entropy, Correlation, Homogeneity, Contrast and Energy. The maps are generated for different sets of GLCM parameters: window size (from 3×3 to 11×11 by default), displacement (from 1 to 3 pixels) and orientation (0°, 45°, 90°, 135°and for all four angles). The algorithm, however, is not limited to the GLCM approach – any method that can generate a feature map of the segmented image can be used.

The following algorithm step selects the texture feature maps that will be used in the segmentation process. The algorithm prefers features that have a low dispersion inside the initial contour region. In order to mark the region, a binary mask is created from the initial curve. For the pixels covered by the mask in each of the feature images t_i, the mean of the feature values $\bar{x}_i$, standard deviation σ_i and Relative Standard Deviation $\%RSD_i = \frac{\sigma_i}{\bar{x}_i} \times 100$ are calculated. The texture feature maps used for the segmentation process must have the $\%RSD$ lower than a user-specified threshold (equal to 65% by default). As there is no supervised texture classification step and the number of different textures in the image is not known, this condition selects the texture features that, hopefully, will distinguish the segmented area from regions with different textures.

Additionally, the selection step can reduce the number of maps by analysing their similarity for different orientations in groups with the same feature, window size and displacement. By default, the algorithm generates one map for each of the orientations and one extra map with an averaged response for all angles. Then, it selects only the directional maps which have their feature value mean (inside the initial region) sufficiently different (at least 50% by default) from the feature mean in the averaged map. However, in case of a clearly isotropic texture pattern, generation of the maps with different orientations can be manually turned off and simplified by creating only the averaged response map.

Contour evolution process

The snake evolution process aims to minimise the energies of the contour points (snaxels) by moving them to the positions of the lowest local energy. A typical intensity energy E^s_{int} for a snaxel s and a potential destination point p can be defined as:

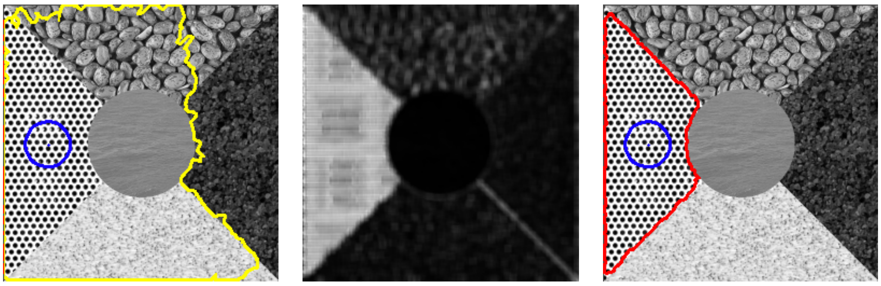
$$E^s_{int}(p) = \begin{cases} \frac{1}{d(s,p)+1} & \text{if } |I(p) - \bar{x}| \leq \theta \times \sigma \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where $I(p)$ is the intensity value in point p, $\bar{x}$ and σ are the intensity mean and standard deviation in the snake initial region, θ is the user-specified sensitivity threshold and $d(s,p)$ is the distance between the location of snaxel s and point p. As the energy is inversely proportional to the distance between the current snaxel and its potential position, it prefers more distant points, enabling the expansion of the contour.

Our texture energy E^s_{tex} works in a way similar to the energy described above, but it takes all the generated texture feature map into consideration: the current snaxel can be moved into the new position only if the similarity condition is fulfilled for all the selected texture feature maps, as defined in:

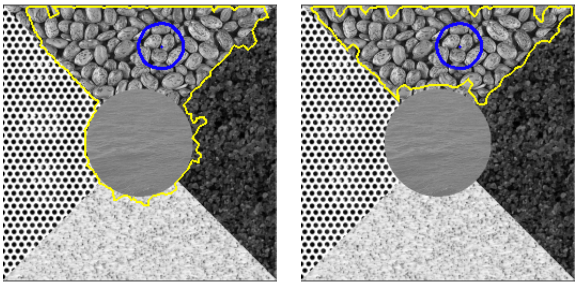
$$E^s_{tex}(p) = \begin{cases} \frac{1}{d(s,p)+1} & \text{if } \forall t \in T_{best} : |val_t(p) - \bar{x}_t| \leq \theta \times \sigma_t \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

where t is a texture feature map in the selected set T_{best}, $\bar{x}_t$ and σ_t are the feature mean and standard deviation in the snake initial region, $val_t(p)$ is the value of the texture feature in the point p, and θ is a user-defined constant. This energy works under two assumptions: (a) features with a low dispersion in the initial snake region have a potential to discriminate it from other patterns and (b) a significant value change in one of the maps blocks the current snaxel from further movement.



(a) (b) (c)

Fig. 1. Segmentation of a high contrast pattern: (a) result of the segmentation with the default intensity energy (initial circular contour visible in blue), (b) calculated GLCM Contrast map and (c) result of the proposed energy.



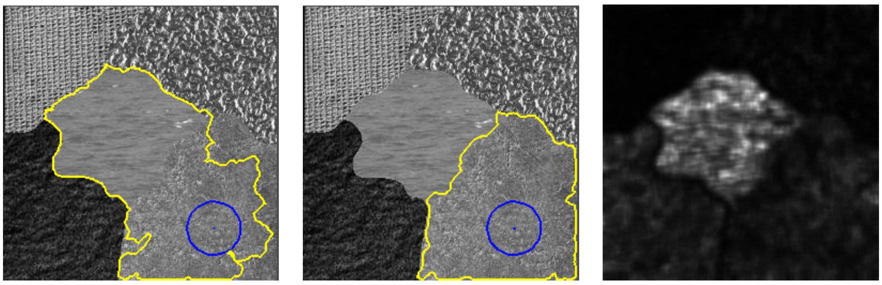
(a) (b)

Fig. 2. Segmentation of a coarse-grained texture: (a) initialisation (in blue) and result (in yellow) using the default intensity energy and (b) result using the proposed energy.

3 Experimental results

This section shows the segmentation results of the proposed method. The experiments were performed on a machine with AMD FX 8150 Eight-Core processor, 16 GB RAM, Nvidia GeForce GTX 660 graphics card (with 960 CUDA cores), and running on Ubuntu 12.04. The total segmentation time was less than 5 seconds for each of the presented examples.

The algorithm was implemented using the MESA system [15] – a platform for designing and evaluation of the deformable model-based segmentation methods. MESA provides a template system for construction of active contours from exchangeable elements (i.e., models, energies and extensions), allowing an easy comparison of the proposed approach with other energies. The GLCM texture generation algorithm was implemented in OpenCL [18] and integrated with the existing code using a Java binding library (JOCL from www.jocl.org). OpenCL allowed utilisation of the graphical processing units,



(a) (b) (c)

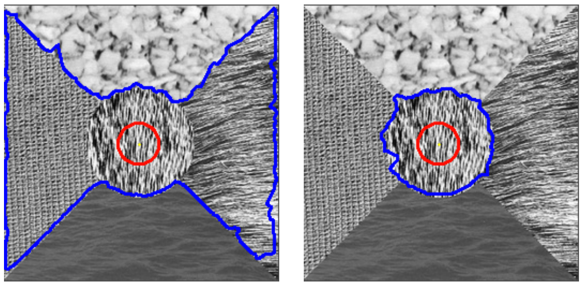
Fig. 3. Example of a fine-grained texture segmentation: (a) result with the intensity energy, (b) result with the texture-based energy and (c) a sample texture map (GLCM Energy) that discriminates the two regions.

which effectively led to a significant speedup of the algorithm (from a few minutes on CPU-based implementation to a few seconds using the hardware-accelerated version).

The method was tested on 256×256 synthetic images created using the Brodatz texture database [19]. The initial contours were manually placed inside the desired region and scaled to the preferred size. During the experiments, only the sensitivity parameter θ was modified (between 3 and 4), while the other parameters were left constant on default values.

The first example (see Fig. 1) presents a segmentation of a region with a high contrast texture. The size and intensity dispersion of the pattern make it impossible to segment with an edge-driven force or with the described default intensity-based energy (see Fig. 1(a)). However, using a GLCM Contrast map (see Fig. 1(b)), the texture-based energy managed to correctly drive the snake to the region boundaries (see Fig. 1(c)).

The second example (see Fig. 2) shows the result for a more coarse-grained pattern. Again, the intensity-based energy failed to distinguish between two regions with a similar average intensity. The third example (see Fig. 3) shows a difficult situation, where two regions with different textures not only have a similar average intensity, but also do not have an evident border between them. While the intensity-based energy could not be adjusted to segment the region, the texture-based approach easily separates the two problematic areas. The last example (see Fig. 4) highlights the importance of the texture map orientation selection. In a case of a highly directional texture, the energy without angle selection fails to segment the region (see Fig. 4(a)). With this selection enabled, the region is correctly separated from the other textures (see Fig. 4(b)).



(a) (b)

Fig. 4. Importance of the texture orientation selection: (a) result without the angle selection and (b) segmentation with orientation selection enabled.

4 Conclusions and future work

In this paper a texture-based energy for the two-dimensional parametric active contour is presented. This energy improves the segmentation performance on images with textures of various size, contrast and complexity. Moreover, the algorithm does not require any previous information about the texture classes in the segmented image. Despite the relatively large number of the utilised features, an efficient GPU-accelerated texture generation method enables a high performance of the segmentation process. The currently proposed approach gives promising results on artificially composed images. We are currently investigating the performance and possible application of the method on natural and medical images.

The present form of the algorithm is in an early stage of development and can benefit from many possible improvements. An initial analysis of the start region can be used to find an optimal parameter set for the generated features. Furthermore, the texture feature maps in the initial step can be calculated only for the start region of the snake, while the generation of the complete maps can be performed after the selection, which will improve the performance. Moreover, the currently utilised feature set (based on GLCM) can be easily extended by incorporation of other texture feature extraction methods, like Gabor filters [12].

Acknowledgement. This work was supported by Bialystok University of Technology under Grant W/WI/5/2014 and S/WI/2/2013.

References

1. P. Moore and D. Molloy. A survey of computer-based deformable models. In *International Machine Vision and Image Processing Conference, IMVIP 2007*, pages 55–66, 2007.

2. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
3. Ch. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.
4. R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.
5. L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53:211–218, 1991.
6. T. Mcinerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. In *Medical Image Analysis*, pages 840–845, 1999.
7. T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.
8. T. Reed and J. DuBuf. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image understanding*, 57(3):359–372, 1993.
9. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.
10. O. Pujol and P. Radeva. Texture segmentation by statistical deformable models. *International Journal of Image and Graphics*, 4(03):433–452, 2004.
11. T. Shen, S.g Zhang, J. Huang, X. Huang, and D. Metaxas. Integrating shape and texture in 3D deformable models: from Metamorphs to Active Volume Models. In *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, pages 1–31. Springer, 2011.
12. A. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
13. X. Huang, Z. Qian, R. Huang, and D. Metaxas. Deformable-model based textured object segmentation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 119–135. Springer, 2005.
14. T. Brox, M. Rousson, R. Deriche, and J. Weickert. Colour, texture, and motion in level set based segmentation and tracking. *Image and Vision Computing*, 28(3):376–390, 2010.
15. D. Reska, K. Jurczuk, C. Boldak, and M. Kretowski. MESA: Complete approach for design and evaluation of segmentation methods using real and simulated tomographic images. *Biocybernetics and Biomedical Engineering*, 2014 (in press).
16. R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, (6):610–621, 1973.
17. D. Reska and M. Kretowski. HIST - an application for segmentation of hepatic images. *Zeszyty Naukowe Politechniki Bialostockiej. Informatyka*, 7:71–93, 2011.
18. J. Stone, D. Gohara, and G. Shi. OpenCL: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(3):66, 2010.
19. P. Brodatz. *Textures: a photographic album for artists and designers*. Dover Publications, 1966.