Project Report

Contour and Texture Analysis for Image Segmentation

JITENDRA MALIK, SERGE BELONGIE, THOMAS LEUNG AND JIANBO SHI

IJCV, 2001

Saikat Kumar Das

SMT2017016

IIIT-Bangalore

Introduction

The authors propose an algorithm [1] for partitioning grayscale images into disjoint coherent regions based on combination of texture and contour analysis. Traditionally, the primary step for image segmentation prior to this work comprised several steps to find edges and segment the image using those edges. However, as rightly pointed out by the authors, such methods are clearly limited in images where we find highly textured regions and repetitive patterns. This is because *edge detection* in such cased produce a 'unmanageable web' of edges which do not impart any additional meaningful information to assist in segmenting the image. Thus, to deal with such images analysing the textures in an image becomes necessary but only analysing texture is not useful for images having majorly homogeneous regions. To solve these multifaceted problems the authors outline a list of properties of good segmentation algorithm which are :

I. The algorithm should be equally effective for general images i.e. not specific to homogenous or textured regions analysis but a combination of both cues to exploit both properties simultaneously
II. In terms of contours the algorithm should be equally effective for brightness step and lines in case of synthetic graphics images like cartoons
III. In terms of texture, the algorithm should be able to handle regular, stochastic or any other patterns which lie in between the above categories

The authors proposed a unified framework for joint analysis of contours and texture. The basis of the algorithm is formed by a set of oriented filters and scaled versions of them which is used to derive a '*hyper coloumn transform'* of the images. This transform is nothing but the collection of filter responses of convolving the image with the oriented filter banks. However, rather than considering each point by a histogram of these filter responses authors propose to map the pixels into higher dimensional space by considering each pixel to be a vector of the collection of all responses at that point.

The vectors are quantized in to texture prototypes named as 'textons' [2] in this work while the contour analysis is done using 'Intervening Contour' framework proposed in an earlier work [3] by the authors. The image segmentation is performed using 'normalized cut' framework, which was also proposed in an earlier work [4] by the authors, after assigning weights to edges between every pixel and its neighbours by jointly analysing the two cues together in a gating process. The motivation behind the gating process is to let the cues work independent of each other in areas where one is preferred over the other.

Available Resources

The publicly available resources for this publication are the prior works of the author and

- Dataset use in this work which was extended eventually to create the widely popular BSDS500 dataset containing 500 images and their contour maps and segmentation
- Original MATLAB code of the Normalized Cut framework from the authors

- An independent implementation of the oriented filter bank mentioned in the paper
- Skimage API for normalized cut

Implementation

The filter bank mentioned in the paper consists of 6 oriented filters in 3 different scales. The filters are actually rotated copies of second order Gaussian derivatives and their Hilbert transfoms. There are 4 filters with radially symmetric receptive field implemented as DoG with different values of ***σ*,** also in 3 scales. The code for creating the filter bank is borrowed from the public git repository of Tony Joseph in https://github.com/CVDLBOT/LM_filter_bank_python_code

$$f_1(x, y) = \frac{d^2}{dy^2}\left(\frac{1}{C}\exp\left(\frac{y^2}{\sigma^2}\right)\exp\left(\frac{x^2}{\ell^2\sigma^2}\right)\right)$$
$$f_2(x, y) = Hilbert(f_1(x, y))$$

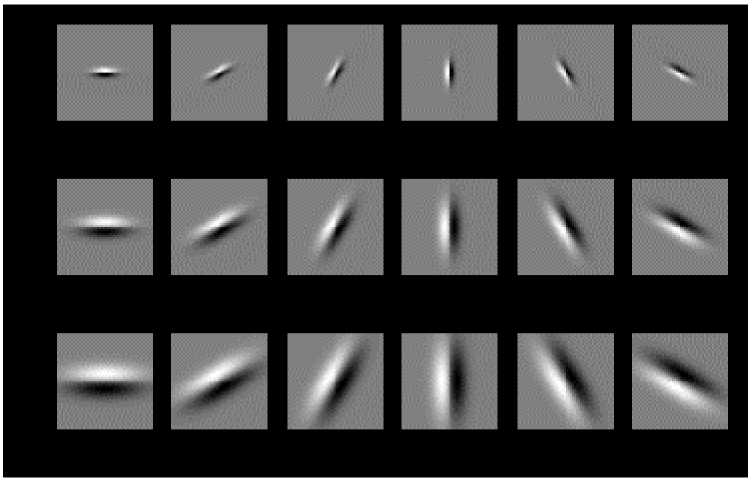

Figure 1. f1(x, y)

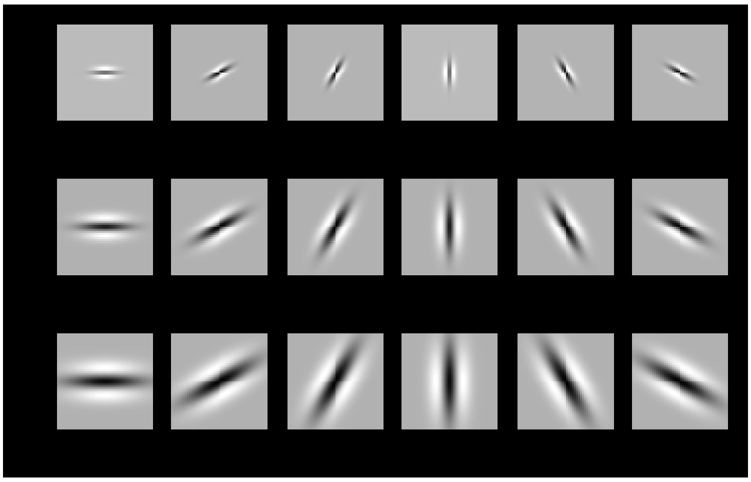

Figure 2. f2(x,y) = Hilbert(f1(x,y))

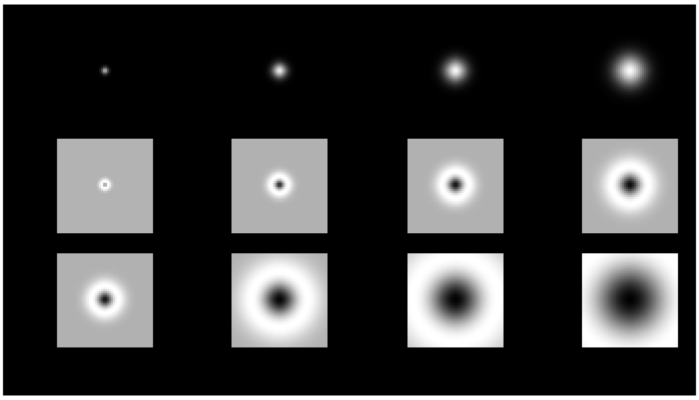

Figure 3. Filters with radially symmetric receptive fields

This Filter bank is the used to create the hyper coloumn transform of the image which is the collection of responses from convolving the image with the filterbank $I * f_i$

Figure 4. Sample Image

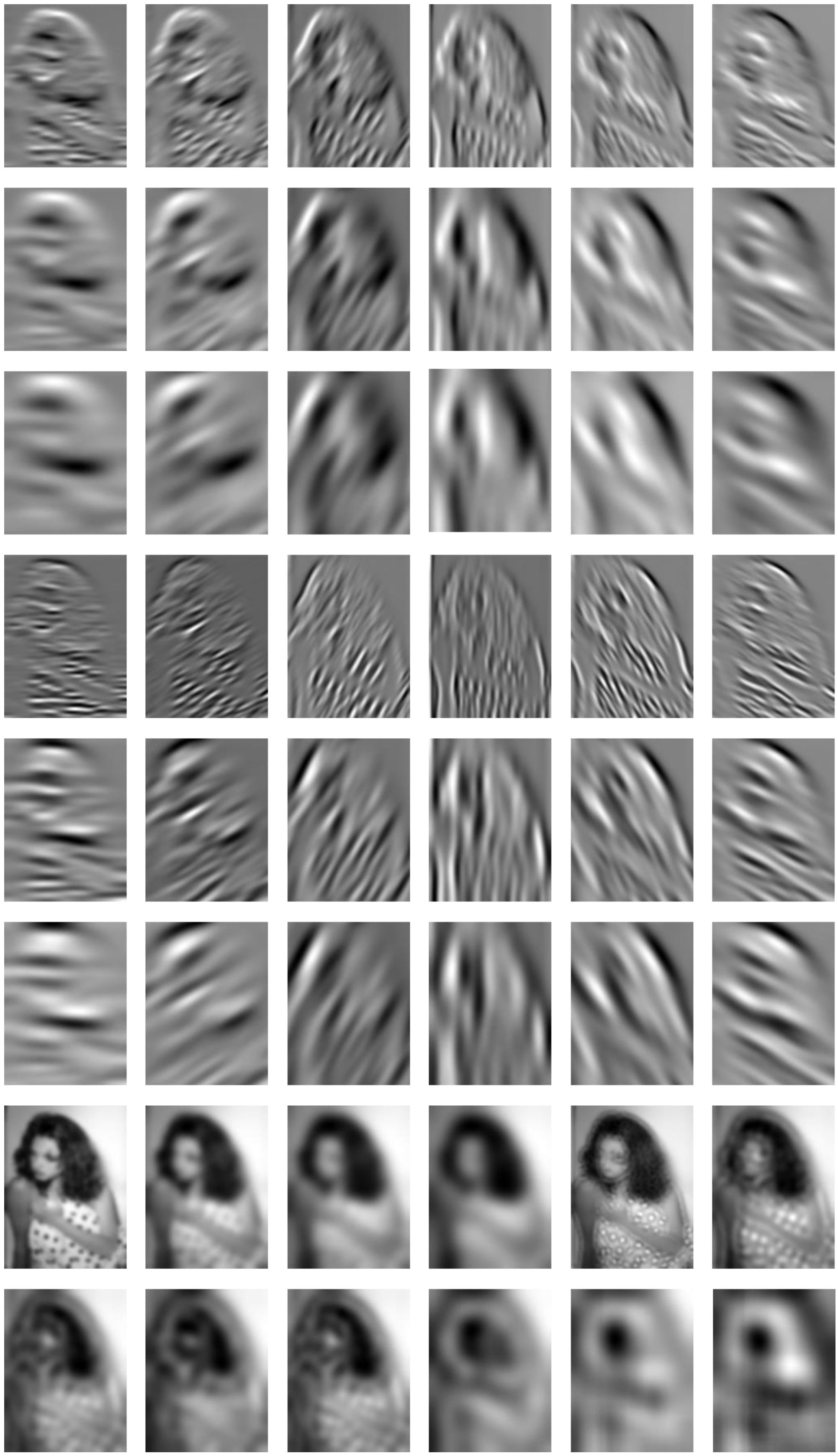

Figure 5. Hyper Coloumn Transform of the image

The hyper coloumn transform allows us to visualise each pixel as a vector made up of the all the different responses of the filter. Thus, these vectors can be quantized in 'textons' in a higher dimensional space. This is done using K-means clustering of the pixel-vectors with K=36

Figure 6. Clustering of the filter response vectors

The corresponding image patches (textons) to each centres of the clustering can be found by premultiplying the centre vectors with pseudoinverse of the filetrbank matrix (created by turning each filter into a coloumn vector)

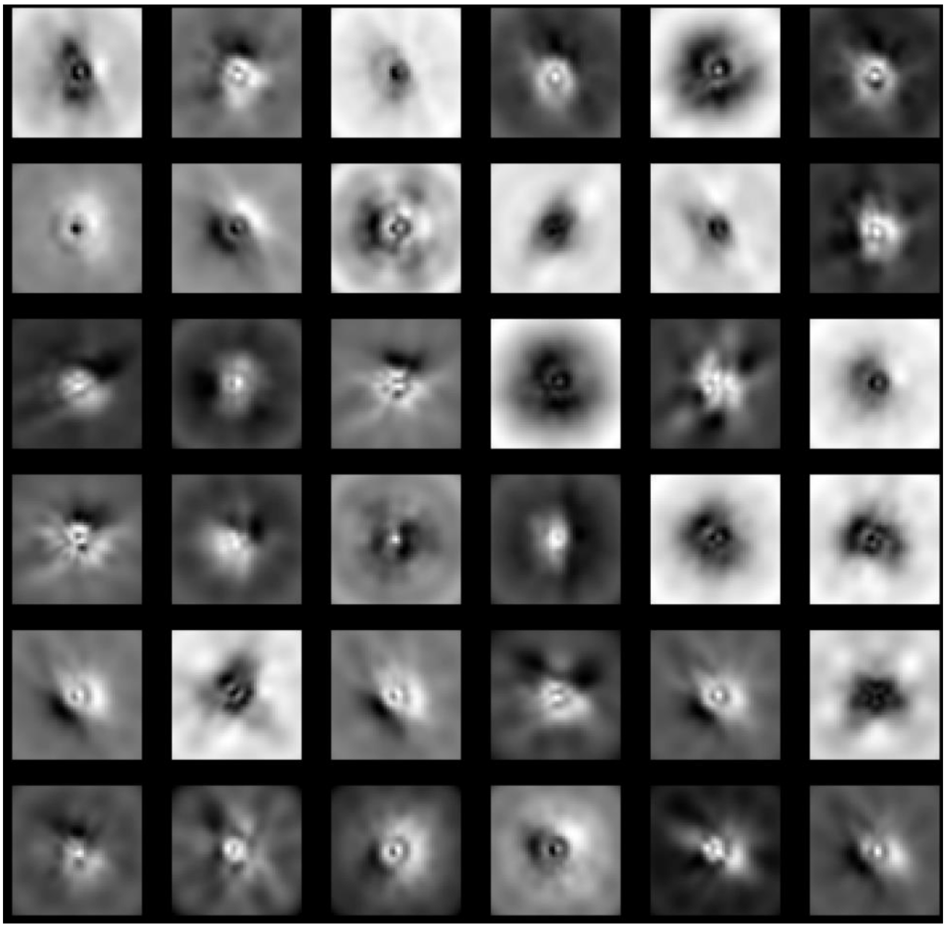

Figure 7. Texton(s)

Using this textons the image can be decomposed into texton channels as depicted below

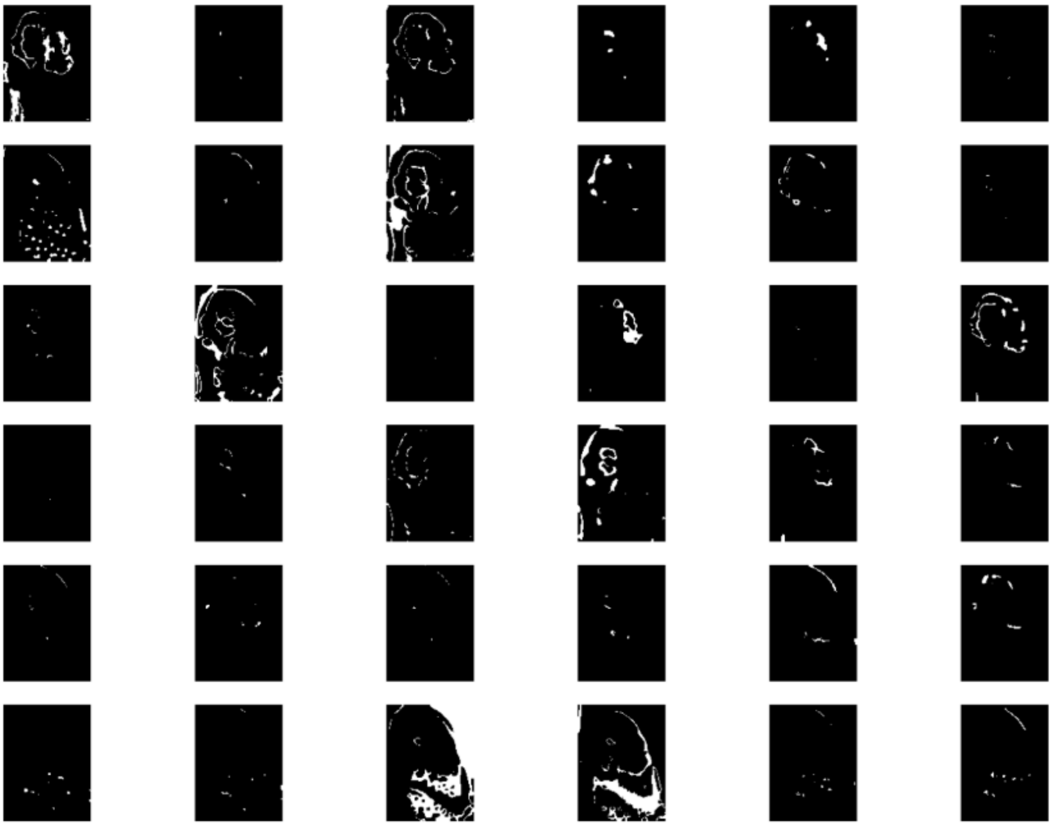

Figure 8. Texton Channels

However, the hyper coloumn transform does not only impart new information in terms of texture analysis only. As the filter pairs $f_1(x, y)$ and $f_2(x, y)$ are perfect quadrature pair thus we can get the *Orientation Energy* at any particular direction by summing the $f_1(x, y)$ and $f_2(x, y)$ filter responses corresponding to that direction

$$OE_\theta = (I * f_{1,\theta})^2 + (I * f_{2,\theta})^2$$

Figure 9. Orientation Energies in different directions

We can find OE^* the dominant orientation energy by finding the dominant orientation by solving

$$\theta^* = \arg\max OE_\theta$$

Figure 10. OE dominant orientation energy*

However, a non-maxima suppression step is required by comparing values of neighbours perpendicular to the dominant orientation to filterout non-contour points. After Non-maxima suppression the values are normalized between 0 and 1 using

$$P_{con} = 1 - \exp(-\frac{OE^*}{\sigma_{IC}})$$

Where P_{con} is the probability like number for the Intervening Contour framework to work and σ_{IC} is the normalising factor related to the image noise.

Figure 11P_con probabbility like number for contour cue to work

Weight between two pixels according to the contour cue is calculated using the Intervening Contour framework as described in [4].

$$W_{ij}^{IC} = 1 - \max_{x \epsilon M_{ij}} P_{con}(x)$$

On the other hand, local scale $\alpha(i)$ at pixel i is found by Delaunay Triangulation of neighbours of that pixel in texton channels and finding median distance amongst them. This scale $\alpha(i)$ is used to create a window of $W(i)$ around pixel to find histogram of textures

$$h_i(k) = \sum_{j \in W(j)} I[T(j) = k]$$

Pairwise texture similarity is found by χ^2 test of histograms

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$

And the weight for the texture cue is given as

$$W_{ij}^{TX} = \exp(-\chi^2(h_i, h_j)/\sigma_{TX})$$

But for general images the texture cue and contour cues cannot be used as it is. They have to be combined in a gating framework so that one cue does not interfere with the other. Like in a textured region the texture cue should get more weight than the contour cue. Similarly in a homogenous area with brightness step contour cue should be the dominant one. To get the cues we first find the texturedness of apixel, more precisely the area around a pixel by

$$\chi^2(h_L, h_R) = \frac{1}{2}\sum_{k=1}^{K} \frac{[h_L(k) - h_R(k)]^2}{h_L(k) + h_R(k)}$$

$$P_{texture} = 1 - \frac{1}{1 + \exp[-(\chi_{LR}^2 - \tau)/\beta]}$$

Where χ_{LR}^2 is the maximum of right and left χ^2 test.

The contour cue is gated by

$$P_B = (1 - P_{texture})P_{con}$$

$$W_{ij}^{IC} = 1 - \max_{x \epsilon M_q} P_B(x)$$

The texture cue is gated by changing the number of bins and their counts in histogram according to

$$\widehat{h_\iota}(k) = \sum_{j \in N(i)} P_{texture}(j).I[T(j) = k]$$

$$\widehat{h_\iota}(0) = N_B + \sum_{j \in N(i)} (1 - P_{texture}(j))$$

The weights are then combined by simply multiplying them

$$W_{ij} = W_{ij}^{IC} * W_{ij}^{TX}$$

The normalized cut framework acts on these weights in an iterative manner

$$N\ cut(\boldsymbol{A}, \boldsymbol{B}) = \frac{cut(\boldsymbol{A}, \boldsymbol{B})}{assoc(\boldsymbol{A}, \boldsymbol{V})} + \frac{cut(\boldsymbol{B}, \boldsymbol{A})}{assoc(\boldsymbol{B}, \boldsymbol{V})}$$

$$\begin{aligned} \boldsymbol{y} &= \arg\min Ncut \\ &= \arg\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}} \end{aligned}$$

In paper the authors have used 30 pixel radius for connecting the edge weight but due to limited compute capability in practice only 15 could be used which did not provide desired results for the segmentation

Figure 12. Sparsely connected segmentation

Benchmarking

Due to unavailability of ground truth data the original work has no quantitative analysis

In the current implementation due to unexpected behaviour of the skimage spectral graph theory package of Cut_Normalized() API the results could not be generated for quantification.

The steps prior to the normalized cut has been achieved completely.

Limitations

Due to limited available original resources the work was not extended further in this project.

Although the API cut_normalized() from graph package of skimage.future was based on this paper but over reliance of the API on other helping APIs and complex graph creation method hindered the free use of it. Another factor that was observed was the API was used extensively for graphs having not more than 400 or500 nodes. But as per the original work every pixel had to be considered as a node. Thus, due to constraints of time and compute capability the work could not progressed further.

References

[1] Malik, Jitendra, et al. "Contour and texture analysis for image segmentation." *International journal of computer vision* 43.1 (2001): 7-27.

[2] Julesz, B. ”Textons, the elements of texture perception, and their interactions.” *Nature*, 290.5802 (1981):91–97.

[3] Leung, T. and Malik, J. “Contour continuity in region-based image segmentation” *In Proc. Euro. Conf. Computer Vision*, Vol.1, H.Burkhardt and B.Neumann (Eds.). Freiburg, Germany, (1998) pp.544–559.

[4] Shi,J.andMalik,J. “Normalized cuts and image segmentation” *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, (1997) pp. 731–737.