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 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 :

1. 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
2. In terms of contours the algorithm should be equally effective for brightness step and lines in case of synthetic graphics images like cartoons
3. 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’ in this work while the contour analysis is done using ‘Intervening Contour’ framework proposed in an earlier work by the authors. The image segmentation is performed using ‘normalized cut’ framework, which was also proposed in an earlier work 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.

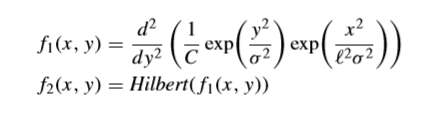
Available Resources

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>



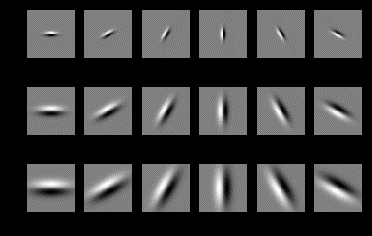


Figure . f1(x, y)

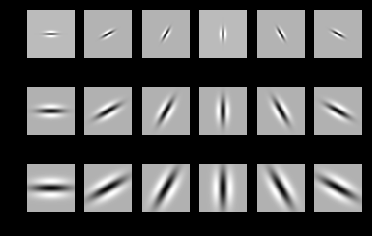


Figure . f2(x,y) = Hilbert(f1(x,y))

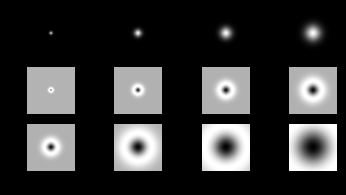


Figure . Filters with radially symmetric receptive fields

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



Figure . Sample Image

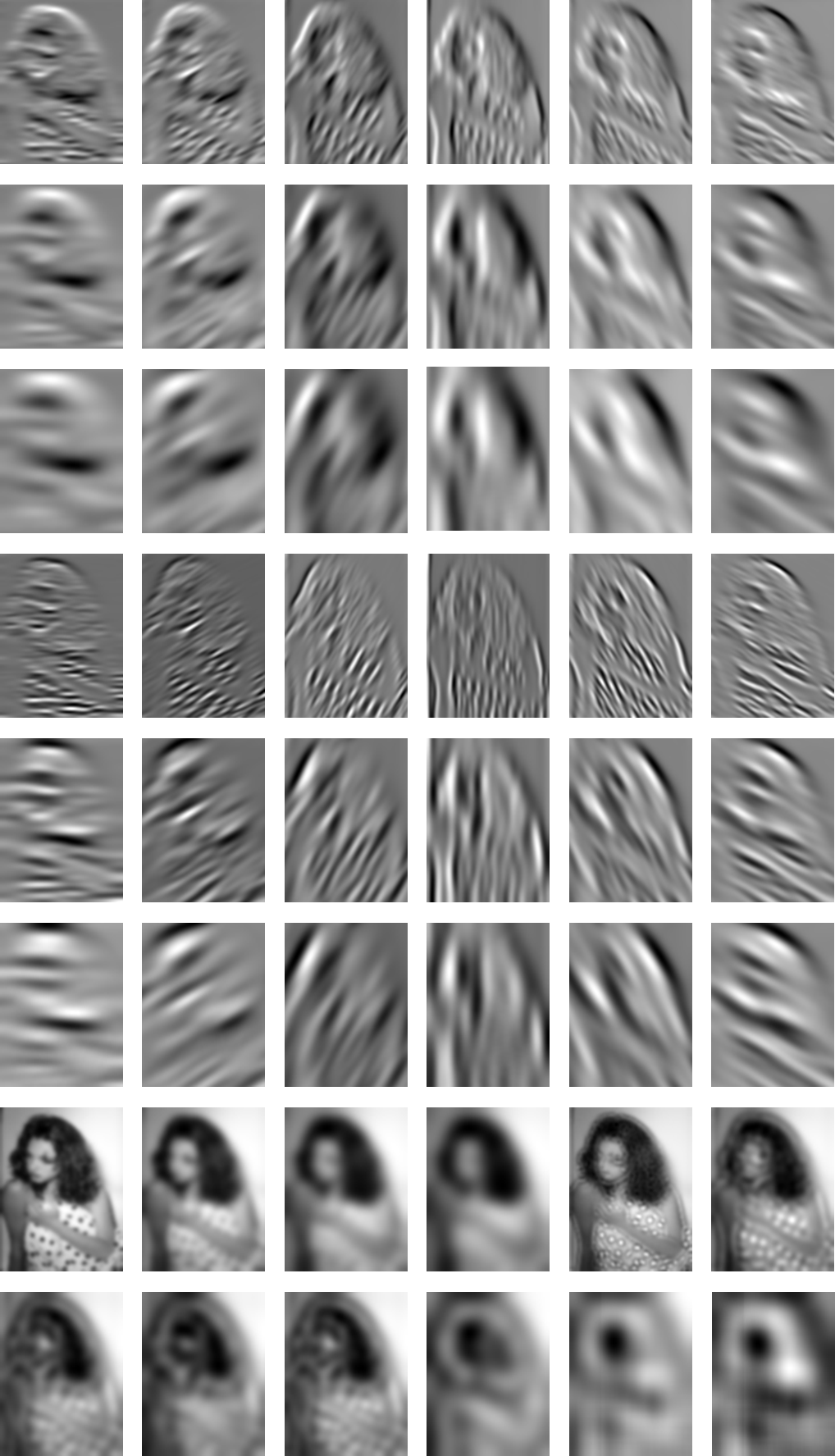


Figure . Hyper Coloumn Transform of the image

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 . 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)

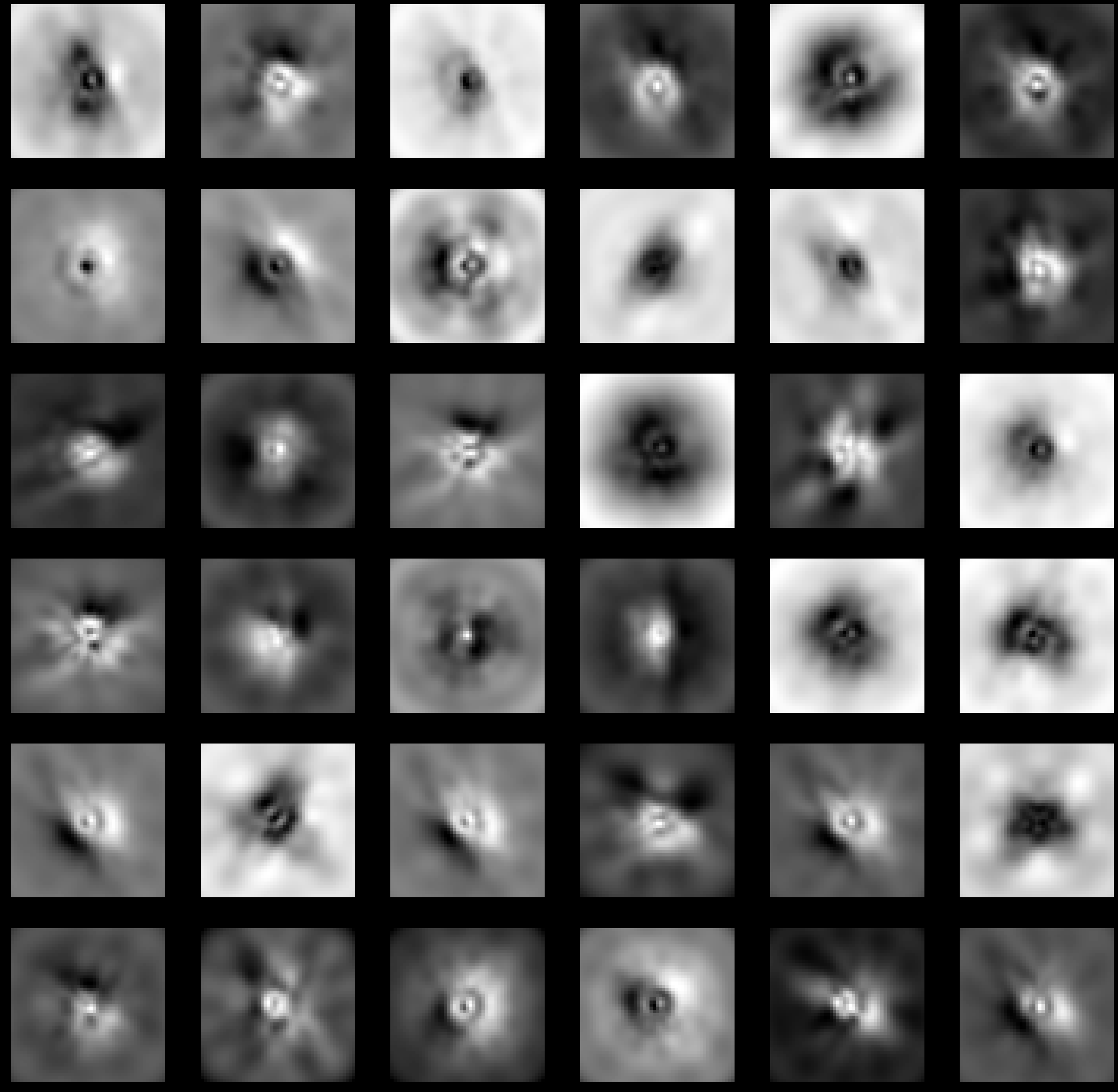


Figure . Texton(s)

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

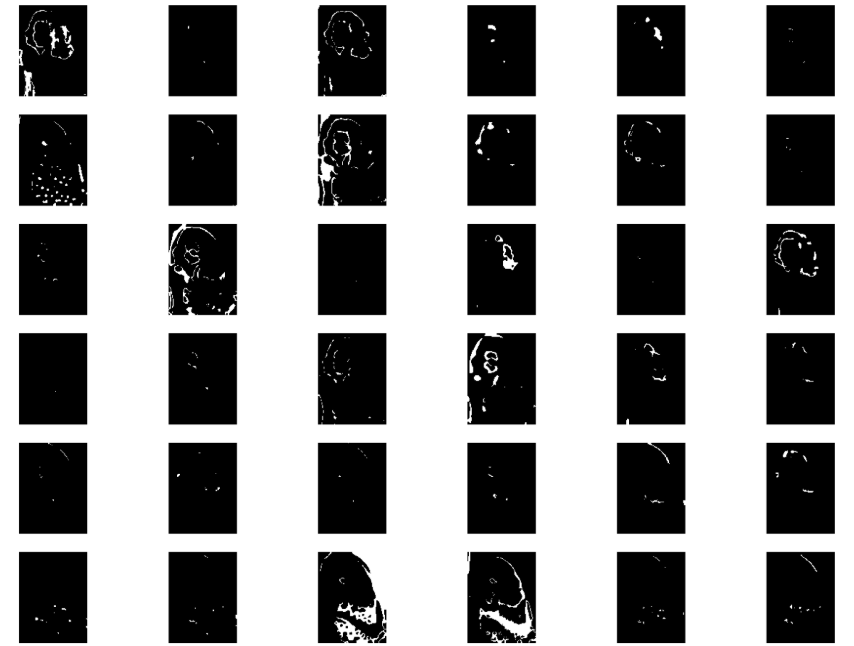


Figure . Texton Channels

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



Figure . Orientation Energies in different directions

We can find the dominant orientation energy by finding the dominant orientation by solving



Figure . OE\* dominant orientation energy

However, a non-maxima suppression step is required by