Annals of the BMVA Vol. 2007, No. 6, pp 1–10 (2007)

Colour Image Segmentation using Texems

Xianghua Xie and Majid Mirmehdi
Department of Computer Science,
University of Bristol, Bristol BS8 1UB, England
⟨xie@cs.bris.ac.uk⟩ ⟨majid@cs.bris.ac.uk⟩

Abstract

We present two methods to perform colour image segmentation using a generative three-dimensional model that is based on the assumption that an image can be generated through an overlapped placement of a few primitive, exemplar image patches, i.e. texems. Multiscale analysis is used in order to capture sufficient image features and pixel neighbourhood interactions at relatively lower computational costs. Experimental results on synthetic and real images are presented to demonstrate this as a promising alternative approach to popular discriminative methods.

1 Introduction

Numerous features have been reported for the benefit of image segmentation, including histogram properties, co-occurrence matrices, local binary patterns, fractal dimension, Markov random field features, multiscale multidirectional filter responses, and simply pixel colour. These features are then spatially and / or spectrally grouped together to form image regions. Region growing, merge-split, Bayesian classification, and neural network classification are examples of common techniques applied to achieve the latter.

Most colour image segmentation techniques are derived from methods designed for graylevel images, usually taking one of three forms: (1) Processing each channel individually by directly applying graylevel based methods Caelli and Reye [1993], Haindl and Havlicek [2002]: The channels are assumed independent and only their intra-spatial interactions are considered. (2) Decomposing the image into luminance and chromatic channels Paschos et al. [1999], Dubuisson-Jolly and Gupta [2000]: After transforming the image data into the desired (usually application dependent) colour space, texture features are extracted from the luminance channel while chromatic features are extracted from the chromatic channels, each in a specific manner. (3) Combining spatial interaction within each channel and interaction between spectral channels Jain and Healey [1998], Bennett and Khotanzad [1998], Palm [2004], Mirmehdi and Petrou [2000]: Graylevel texture analysis techniques are applied in each channel, while pixel interactions between different channels are also taken into account.

Also, some works perform global colour clustering analysis, followed by spatial analysis in each individual stack.

The importance of extracting correlation between the channels for colour texture analysis has been addressed by several authors. For example, in Jain and Healey [1998], Jain and Healey used Gabor filters to obtain texture features in each channel and opponent features that capture the spatial correlation between the channels.

Original 3D models to analyse colour textures have also been developed, where the spatial and spectral interactions are simultaneously handled, e.g. Jojic et al. [2003]. The main difficulties arise in effectively representing, generalising, and discriminating three dimensional data. The epitome Jojic et al. [2003] provides a compact 3D representation of colour textures. The image is assumed to be a collection of epitomic primitives relying on raw pixel values in image patches. The neighbourhood pixels of a central pixel in a patch are assumed to be statistically conditionally independent. A hidden mapping guides the relationship between the epitome and the original image. This compact representation inherently captures the spatial and spectral interactions simultaneously. The epitome model inspired the development of the texem model Xie and Mirmehdi [2007], a compact mixture representation for colour images, used for novelty based defect detection.

In this paper, we use a simple generative model to represent and segment colour images. The spatial arrangement and interspectral properties of pixels are modelled simultaneously without decomposing images into separate channels. In section 2, the colour texem model is presented. Section 3 describes two segmentation methods based on this model. Texem grouping is discussed in Section 4 for multimodal textures. Some experimental results are given in Section 5. Section 6 concludes the paper.

2 Modelling Colour Images

Representing colour images using 3D space models is considered a challenging task in that it is difficult to keep a compact representation and still sufficiently characterise the image data. In Jojic et al. [2003], Jojic *et al.* introduced the epitome model as a small, condensed representation of a given image which contains its primitive shapes and textural elements. The recently proposed texem (texture exemplar) model Xie and Mirmehdi [2007] is based on the same assumption that a given image can be generated from a collection of image patches and the variation in placement results in appearance variations in the images. However, unlike the epitome, the texem model discards the hidden mapping and uses multiple, much smaller epitomic representations. In Xie and Mirmehdi [2007], it was shown that texems is a much more efficient way of performing novelty detection in colour images. Each of the texems learnt from the image contain partial degrees of image micro-structures. In other words, texems are implicit representations of image primitives, as opposed to textons Julesz [1981] which are explicit representations and very often use base functions Zhu et al. [2005].

Each texem $\mathbf{m}$ is defined by a mean, μ, and a corresponding variance, ω, i.e. $\mathbf{m} = \{\mu, \omega\}$. An image is then considered as a superposition of patches of various sizes. This forces the image properties not into a single texem, but a family of them. We use a mixture model to learn the texems which together characterise a given image. The original image $\mathbf{I}$ is broken down into a set of P overlapping patches $\mathbf{Z} = \{\mathbf{Z}_i\}_{i=1}^{P}$, each containing pixels from a subset of image coordinates. For simplicity, square patches of size $d = N \times N$ are used. We assume that there exist K texems, $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^{K}$, $K \ll P$, for image $\mathbf{I}$ such that each patch in $\mathbf{Z}$ can be generated from a texem with certain added variations:

$$p(\mathbf{Z}_i|\theta_k) = p(\mathbf{Z}_i|\boldsymbol{\mu}_k, \boldsymbol{\omega}_k) = \prod_{j\in S} \mathcal{N}(\mathbf{Z}_{j,i}; \boldsymbol{\mu}_{j,k}, \boldsymbol{\omega}_{j,k}), \tag{1}$$

where θ_k denotes the kth texem's parameters with mean $\boldsymbol{\mu}_k$ and variance $\boldsymbol{\omega}_k$, $\mathcal{N}(.)$ is a Gaussian distribution over $\mathbf{Z}_{j,i}$, S is the patch pixel grid, $\boldsymbol{\mu}_{j,k}$ and $\boldsymbol{\omega}_{j,k}$ denote mean and variance at the jth pixel position in the kth texem. The mixture model is given by:

$$p(\mathbf{Z}_i|\Theta) = \sum_{k=1}^{K} p(\mathbf{Z}_i|\theta_k)\alpha_k, \tag{2}$$

where $\Theta = (\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K)$, and α_k is the *priori* probability of kth texem constrained by $\sum_{k=1}^{K} \alpha_k = 1$. The Expectation and Maximisation (EM) technique can be used to estimate the model parameters. The E-step involves a soft-assignment of each patch $\mathbf{Z}_i$ to texems, $\mathcal{M}$, with an initial guess of the true parameters, Θ. We denote the intermediate iteration t parameters as $\Theta^{(t)}$. The probability that patch $\mathbf{Z}_i$ belongs to the kth texem may then be computed using Bayes' rule:

$$p(\mathbf{m}_k|\mathbf{Z}_i, \Theta^{(t)}) = \frac{p(\mathbf{Z}_i|\mathbf{m}_k, \Theta^{(t)})\alpha_k}{\sum_{k=1}^{K} p(\mathbf{Z}_i|\mathbf{m}_k, \Theta^{(t)})\alpha_k}. \tag{3}$$

The M-step then updates the parameters according to:

$$\begin{aligned}
\hat{\alpha}_k &= \frac{1}{P}\sum_{i=1}^{P} p(\mathbf{m}_k|\mathbf{Z}_i, \Theta^{(t)}), \quad \hat{\boldsymbol{\mu}}_k = \{\hat{\boldsymbol{\mu}}_{j,k}\}_{j\in S}, \quad \hat{\boldsymbol{\omega}}_k = \{\hat{\boldsymbol{\omega}}_{j,k}\}_{j\in S}, \\
\hat{\boldsymbol{\mu}}_{j,k} &= \frac{\sum_{i=1}^{P} \mathbf{Z}_{j,i} p(\mathbf{m}_k|\mathbf{Z}_i, \Theta^{(t)})}{\sum_{i=1}^{P} p(\mathbf{m}_k|\mathbf{Z}_i, \Theta^{(t)})}, \\
\hat{\boldsymbol{\omega}}_{j,k} &= \frac{\sum_{i=1}^{P} (\mathbf{Z}_{j,i} - \hat{\boldsymbol{\mu}}_{j,k})(\mathbf{Z}_{j,i} - \hat{\boldsymbol{\mu}}_{j,k})^T p(\mathbf{m}_k|\mathbf{Z}_i, \Theta^{(t)})}{\sum_{i=1}^{P} p(\mathbf{m}_k|\mathbf{Z}_i, \Theta^{(t)})}.
\end{aligned} \tag{4}$$

The E-step and M-step are iterated until the estimations stabilise or the rate of improvement of the likelihood falls below a pre-specified convergence threshold.

3 Colour Image Segmentation

Clearly each image patch from an image has a measurable relationship with each texem according to the *posteriori*, $p(\mathbf{m}_k|\mathbf{Z}_i, \Theta)$, which can be conveniently obtained using Bayes' rule in (3). Thus, every texem can be viewed as an individual textural class component, and the *posteriori* can be regarded as the component likelihood with which each pixel in the image can be labelled. Based on this, we present two different multiscale approaches to carry out segmentation. One performs segmentation at each level separately, and then updates the label probabilities from coarser to finer levels, and the other simplifies the procedure by learning the texems across the scales to gain efficiency.

3.1 Segmentation with interscale post-fusion

Various sizes of texems are necessary to capture sufficient image properties. Alternatively, the same size texems can be applied to a multiscale image, as in Xie and Mirmehdi [2007], where texems were generated from each scale independently for novelty detection. Detection results from individual scales were then combined to produce a final novelty defect map. For image segmentation, however, the fusion procedure is more involved, e.g. a relaxation process Mirmehdi and Petrou [2000] can be used to update the class probabilities from coarser to finer levels.

We first layout the image in multiscale. Besides computational efficiency, exploiting information at multiscale offers other advantages. Characterising a pixel based on local neighbourhood pixels can be more effectively achieved by examining various neighbourhood relationships. A simple Gaussian pyramid was found to be sufficient.

Let us denote $\mathbf{I}^{(n)}$ as the nth level image of the pyramid, $\mathbf{Z}^{(n)}$ as all the patches extracted from $\mathbf{I}^{(n)}$, and l as the total number of levels. We then extract texems from individual pyramid levels. Similarly, let $\mathbf{m}^{(n)}$ denote the nth level of multiscale texems and $\Theta^{(n)}$ the associated parameters. During the EM process, the stabilised estimation of a coarser level is used as the initial estimation for the finer level, i.e. $\hat{\Theta}^{(n,t=0)} = \Theta^{(n+1)}$, which helps speed up the convergence and achieve a more accurate estimation.

For segmentation, each pixel needs to be assigned a class label, $c = \{1, 2, ..., K\}$. We can perform this labelling using the measurable relationship between each patch at its central pixel position and the texems, as given in (3). So, at each scale n, there is a random field of class labels, $C^{(n)}$. The probability of a particular image patch, $\mathbf{Z}_i^{(n)}$, belonging to a texem (class), $c = k, \mathbf{m}_k^{(n)}$, is determined by the *posteriori* probability, $p(c = k, \mathbf{m}_k^{(n)} | \mathbf{Z}_i^{(n)}, \Theta^{(n)})$, simplified as $p(c^{(n)}|\mathbf{Z}_i^{(n)})$, given by:

$$p(c^{(n)}|\mathbf{Z}_i^{(n)}) = \frac{p(\mathbf{Z}_i^{(n)}|\mathbf{m}_k^{(n)})\boldsymbol{\alpha}_k^{(n)}}{\sum_{k=1}^{K} p(\mathbf{Z}_i^{(n)}|\mathbf{m}_k^{(n)})\boldsymbol{\alpha}_k^{(n)}}, \tag{5}$$

which is equivalent to the stablised solution of (3). The class probability at given pixel location $(x^{(n)}, y^{(n)})$ at scale n then can be estimated as $p(c^{(n)}|(x^{(n)}, y^{(n)})) = p(c^{(n)}|\mathbf{Z}_i^{(n)})$. Thus, this labelling assignment procedure initially partitions the image in each individual scale. As the image is laid hierarchically, there is inherited relationship among parent and children pixels. Their labels should also reflect this relationship. Next, building on this initial labelling, the partitions across all the scales are fused together to produce the final segmentation map.

The class labels $c^{(n)}$ are assumed conditionally independent given the labelling in the coarser scale $c^{(n+1)}$. Thus, each label field $C^{(n)}$ is assumed only dependent on the previous coarser scale label field $C^{(n+1)}$. This offers efficient computational processing, while preserving the complex spatial dependencies in the segmentation. The label field $C^{(n)}$ becomes a Markov chain structure in the scale variable n:

$$p(c^{(n)}|c^{(>n)}) = p(c^{(n)}|c^{(n+1)}), \tag{6}$$

where $c^{(>n)} = \{c^{(i)}\}_{i=n+1}^{l}$ are the class labels at all coarser scales greater than the nth, and $p(c^{(l)}|c^{(l+1)}) = p(c^{(l)})$ as l is the coarsest scale. The coarsest scale segmentation is directly based on the initial labelling.

A quadtree structure for the multiscale label fields is assumed, and $c^{(l)}$ only contains a single pixel, although a more sophisticated context model can be used to achieve better interaction between child and parent nodes, e.g. a pyramid graph model Cheng and Bouman [2001]. The transition probability $p(c^{(n)}|c^{(n+1)})$ can be efficiently calculated numerically using a lookup table. The label assignment at each scale are then updated, from coarsest to the finest level, according to the joint probability of the data probability and the transition probability:

$$\begin{cases} \hat{c}^{(l)} = \arg\max_{c^{(l)}} \log p(c^{(l)}|(x^{(l)}, y^{(l)})), \\ \hat{c}^{(n)} = \arg\max_{c^{(n)}} \{\log p(c^{(n)}|(x^{(n)}, y^{(n)})) + \log p(c^{(n)}|c^{(n+1)})\} \quad \forall n < l. \end{cases} \tag{7}$$

The segmented regions will be smooth and small isolated holes are filled.

3.2 Segmentation using branch partitioning

Starting from the pyramid layout described in Section 3.1, each pixel in the finest level can trace its parent pixel back to the coarsest level forming a unique route or branch. In Section 3.1, the conditional independence assumption amongst pixels within the local neighbourhood makes the parameter estimation tractable. Here, we assume pixels *in the same branch* are conditionally independent, i.e.

$$p(\mathbf{Z}_i|\theta_k) = p(\mathbf{Z}_i|\mu_k, \omega_k) = \prod_{n\in l} \mathcal{N}(\mathbf{Z}_i^{(n)}; \mu_k^{(n)}, \omega_k^{(n)}), \tag{8}$$

where $\mathbf{Z}_i$ here is a branch of pixels, $\mathbf{Z}_i^{(n)}$, $\mu_k^{(n)}$, and $\omega_k^{(n)}$ are the colour pixel at level n in ith branch, mean at level n of kth texem, and variance at level n of kth texem, respectively. This is essentially the same form as (1), hence, we can still use the EM procedure described previously to derive the texem parameters. However, the image is not partitioned into patches, but rather laid out in multiscale first and then separated into branches. The pixels are collected across scales, instead of from its neighbours. The class labels are then directly given by the component likelihood, again using Bayes' rule, $p(c|\mathbf{Z}_i) = p(\mathbf{m}_k|\mathbf{Z}_i, \Theta)$. Thus, we simplify the approach presented in Section 3.1 by avoiding the inter-scale fusion after labelling each scale.

4 Texem Grouping for Multimodal Texture

A textural region may contain multiple visual elements and display complex patterns. A single texem might not be able to fully represent such textural regions, hence, several texems can be grouped together to jointly represent "multimodal" texture regions. Here, we use a simple but effective method proposed by Manduchi Manduchi [1999, 2000] to group texems. The basic strategy is to group some of the texems based on their spatial coherence. The grouping process simply takes the form:

$$\hat{p}(\mathbf{Z}_i|c) = \frac{1}{\hat{\beta}_c} \sum_{k\in G_c} p(\mathbf{Z}_i|\mathbf{m}_k)\alpha_k, \qquad \hat{\beta}_c = \sum_{k\in G_c} \alpha_k, \tag{9}$$

where G_c is the group of texems that are combined together to form a new cluster c which labels the different texture classes, and $\hat{\beta}_c$ is the *priori* for new cluster c. The mixture model can thus be reformulated as:

$$p(\mathbf{Z}_i|\Theta) = \sum_{c=1}^{\hat{K}} \hat{p}(\mathbf{Z}_i|\mathbf{m}_k)\hat{\beta}_c, \tag{10}$$

where $\hat{K}$ is the desired number of texture regions. Equation (10) shows that pixel i in the centre of patch $\mathbf{Z}_i$ will be assigned to the texture class c which maximises $\hat{p}(\mathbf{Z}_i|c)\hat{\beta}_c$:

$$c = \arg\max_c \hat{p}(\mathbf{Z}_i|c)\hat{\beta}_c = \arg\max_c \sum_{k\in G_c} p(\mathbf{Z}_i|\mathbf{m}_k)\alpha_k. \tag{11}$$

The grouping in (10) is carried out based on the assumption that the *posteriori* probabilities of grouped texems are typically spatially correlated. The process should minimise the decrease of model descriptiveness, D, which is defined as Manduchi [1999, 2000]:

$$D = \sum_{j=1}^{K} D_j, \quad D_j = \int p(\mathbf{Z}_i|\mathbf{m}_j)p(\mathbf{m}_j|\mathbf{Z}_i)d\mathbf{Z}_i = \frac{E[p(\mathbf{m}_j|\mathbf{Z}_i)^2]}{\alpha_j}, \tag{12}$$

where $E[.]$ is the expectation computed with respect to $p(\mathbf{Z}_i)$. In other words, the compacted model should retain as much descriptiveness as possible. This is known as the Maximum Description Criterion (MDC). The descriptiveness decreases drastically when well separated texem components are grouped together, but decreases very slowly when spatially correlated texem component distributions merge together. Thus, the texem grouping should search for smallest change in descriptiveness, ΔD. It can be carried out by greedily grouping two texem components, $\mathbf{m}_a$ and $\mathbf{m}_b$, at a time with minimum ΔD_{ab}:

$$\Delta D_{ab} = \frac{\alpha_b D_a + \alpha_a D_b}{\alpha_a + \alpha_b} - \frac{2E[p(\mathbf{m}_a|\mathbf{Z}_i)p(\mathbf{m}_b|\mathbf{Z}_i)]}{\alpha_a + \alpha_b}. \tag{13}$$

We can see that the first term in (13) is the maximum possible descriptiveness loss when grouping two texems, and the second term in (13) is the normalised cross correlation between the two texem component distributions. Since one texture region may contain different texem components that are significantly different to each other, it is beneficial to smooth the *posteriori* as proposed in Manduchi [2000] such that a pixel that originally has high probability to just one texem component will be softly assigned to a number of components that belong to the same “multimodal” texture. After grouping, the final segmentation map is obtained according to (11).

5 Experimental Results

Here, we present some experimental results and a brief comparison with the well-known JSEG technique Deng and Manjunath [2001].

Fig. 1 shows example results on four different texture collages with the original image in the first row, groundtruth segmentations in the second row, the JSEG result in the third row, the proposed interscale post-fusion method in the fourth row, and the proposed branch partition method in the final row. The two proposed schemes have similar performance, while JSEG tends to over-segment which partially arises due to the lack of prior knowledge of number of texture regions.

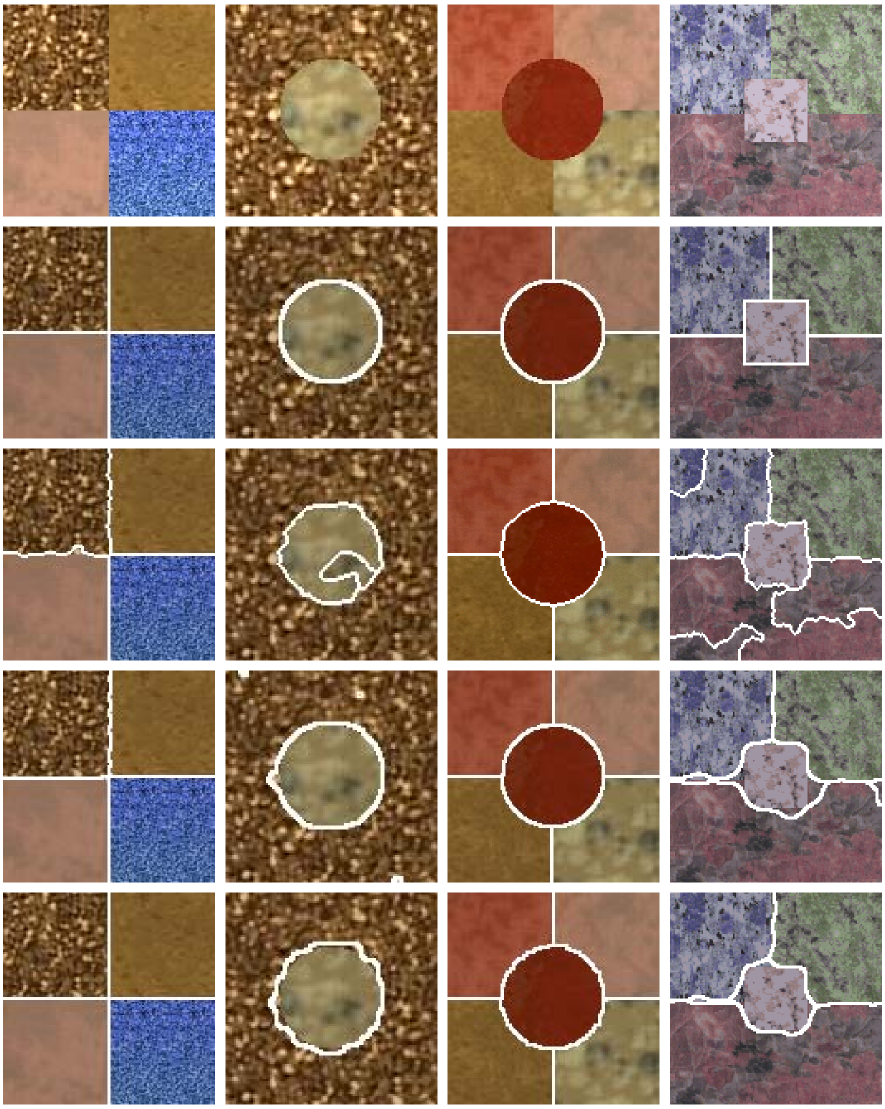

Figure 1: Testing on synthetic images - first row: original image collages, second row: groundtruth segmentations, third row: JSEG results, fourth row: results of the proposed method using interscale post-fusion, last row: results of the proposed method using branch partitioning.

Two real image examples are given in Fig. 2. For each image, we show the original images, its JSEG segmentation and the results of the two proposed segmentation methods. The interscale post-fusion method produced finer borders but is a slower technique.

Fig. 3 focuses on the interscale post-fusion technique followed by texem grouping. The original image and the final segmentation are shown at the top. The second row shows the initial labelling of 5 texem classes for each pyramid level. The texems are grouped to 3 classes as seen in the third row. Interscale fusion is then performed and shown in the last row. Note there is no fusion in the fourth (coarsest) scale.

In Fig. 4, JSEG again over-segmented the images when the texture regions were multimodal in nature. The branch partition method followed by texem grouping segmented the

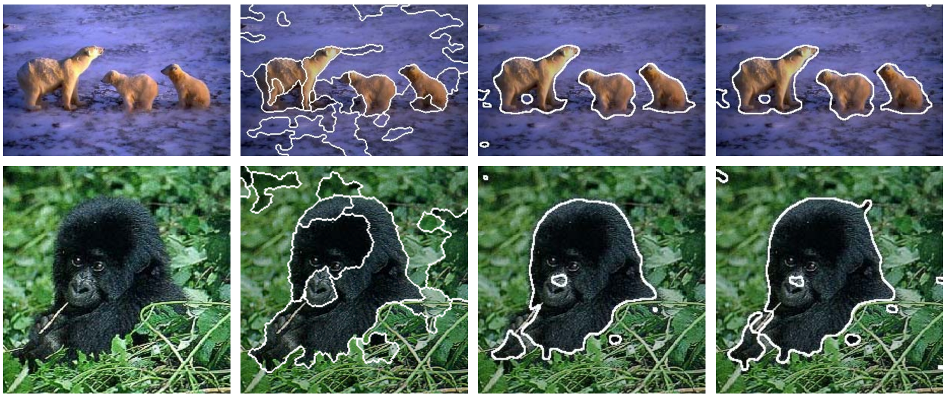

Figure 2: Testing on real images - first column: original images, second column: JSEG results, third column: results of the proposed method using interscale post-fusion, fourth column: results of the proposed method using branch partitioning.

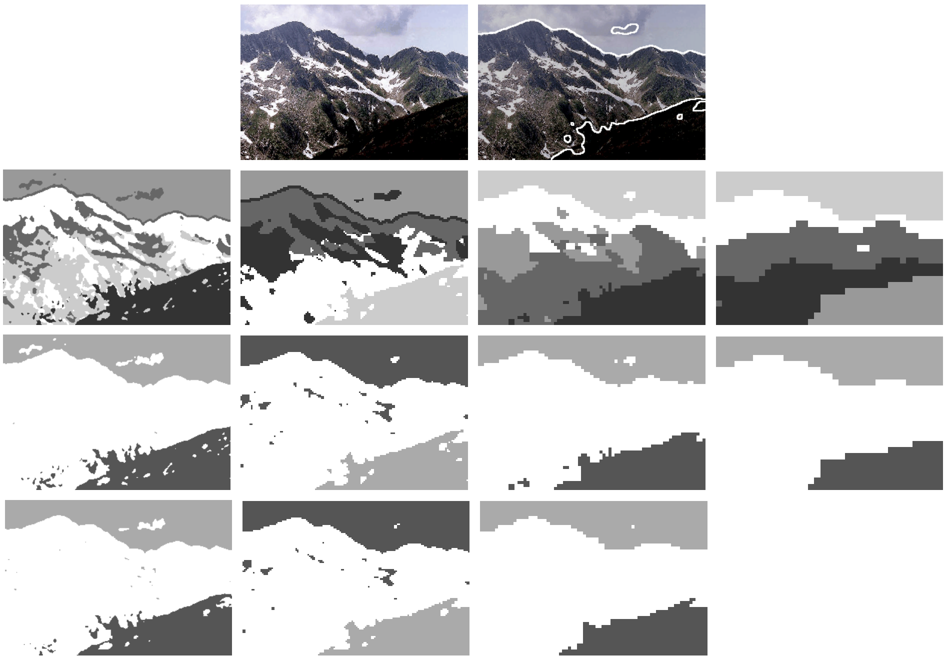

Figure 3: An example of the interscale post-fusion method followed by texem grouping - first row: original image and its segmentation result using the proposed method with interscale post-fusion, second row: initial labelling of 5 texem classes for each scale, third row: updated labelling after grouping 5 texems to 3, fourth row: results of interscale fusion.

images into a more plausible number of texture regions.

The results shown demonstrate that the two proposed methods are more able in modelling textural variations than JSEG and are less prone to over-segmentation. However, it is noted that JSEG does not require the number of regions as prior knowledge. On the other hand, texem based segmentation provides a useful description for each region and a measurable relationship between them. The number of texture regions may be automatically de-

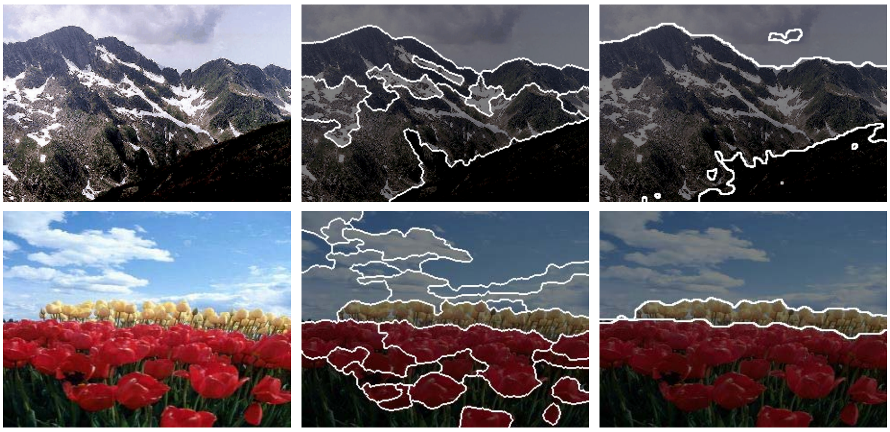

Figure 4: Comparison - from left on each row: original image, JSEG result, the branch partition method followed by texem grouping.

termined using model-order selection methods, such as MDL. The post-fusion and branch partition schemes achieved comparable results, while the branch partition method is faster. However, a more thorough comparison is necessary to draw complete conclusions, which is part of our future work.

6 Conclusions

We presented two colour image segmentation methods based on the texem model. We also showed how to group texems as a potentially useful tool for manipulating them. Future work will focus on methods to automatically estimate the number of texture regions and to further speed-up the texem learning process.

References

J. Bennett and A. Khotanzad. Multispectral random field models for synthesis and analysis of color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):327–332, 1998.

T. Caelli and D. Reye. On the classification of image regions by colour, texture and shape. *Pattern Recognition*, 26(4):461–470, 1993.

H. Cheng and C. Bouman. Multiscale Bayesian segmentation using a trainable context model. *IEEE Transactions on Image Processing*, 10(4):511–525, 2001.

Y. Deng and B. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

M. Dubuisson-Jolly and A. Gupta. Color and texture fusion: Application to aerial image segmentation and GIS updating. *Image and Vision Computing*, 18:823–832, 2000.

M. Haindl and V. Havlicek. A simple multispectral multiresolution Markov texture model. In *International Workshop on Texture Analysis and Synthesis*, pages 63–66, 2002.

A. Jain and G. Healey. A multiscale representation including opponent color features for texture recognition. *IEEE Transactions on Image Processing*, 7(1):124–128, 1998.

N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *IEEE International Conference on Computer Vision*, pages 34–42, 2003.

B. Julesz. Textons, the element of texture perception and their interactions. *Nature*, 290:91–97, 1981.

R. Manduchi. Mixture models and the segmentation of multimodal textures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, 2000.

R. Manduchi. Bayesian fusion of color and texture segmentations. In *IEEE International Conference on Computer Vision*, pages 956–962, 1999.

M. Mirmehdi and M. Petrou. Segmentation of color textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):142–159, 2000.

C. Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965–976, 2004.

G. Paschos, P. Kimon, and P. Valavanis. A color texture based monitoring system for automated surveillance. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(1):298–307, 1999.

X. Xie and M. Mirmehdi. TEXEMS: Texture exemplars for defect detection on random textured surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. Accepted.

S. Zhu, C. Guo, Y. Wang, and Z. Xu. What are textons? *International Journal of Computer Vision*, 62(1-2):121–143, 2005.