Efficient texture representation using multi-scale regions*

Horst Wildenauer[1], Branislav Mičušík[1,2] and Markus Vincze[1]

[1] Automation and Control Institute
[2] Institute of Computer Aided Automation, PRIP Group
Vienna University of Technology, Austria

Abstract. This paper introduces an efficient way of representing textures using connected regions which are formed by coherent multi-scale over-segmentations. We show that the recently introduced covariance-based similarity measure, initially applied on rectangular windows, can be used with our newly devised, irregular structure-coherent patches; increasing the discriminative power and consistency of the texture representation. Furthermore, by treating texture in multiple scales, we allow for an implicit encoding of the spatial and statistical texture properties which are persistent across scale. The meaningfulness and efficiency of the covariance based texture representation is verified utilizing a simple binary segmentation method based on min-cut. Our experiments show that the proposed method, despite the low dimensional representation in use, is able to effectively discriminate textures and that its performance compares favorably with the state of the art.

1 Introduction

Textures and structured patterns are important cues towards image understanding, pattern classification and object recognition. The analysis of texture properties and their mathematical and statistical representation is attracting the interest of researchers since many years; with the primary goal of finding low dimensional and expressive representations that allow for reliable handling and classification of texture patterns. Texture representations, which have been successfully applied to image segmentation tasks, include steerable filter responses [1], color changes in a pixel's neighborhood [2], covariance matrices of gradients, color, and pixel coordinates [3], Gaussian Mixture Models (GMM) computed from color channels [4, 5], color histograms [6], or multi-scale densities [7, 8].

Since textures "live" at several scales, a scale-dependent discriminative treatment should be aimed for. In this paper, we explore the possibility to refine coarse texture segmentation by matching textures between adjacent scales, taking advantage of spatial and statistical properties which persist across scale. We

* The research has been supported by the Austrian Science Foundation (FWF) under the grant S9101, and the European Union projects MOVEMENT (IST-2003-511670), Robots@home (IST-045350), and MUSCLE (FP6-507752).

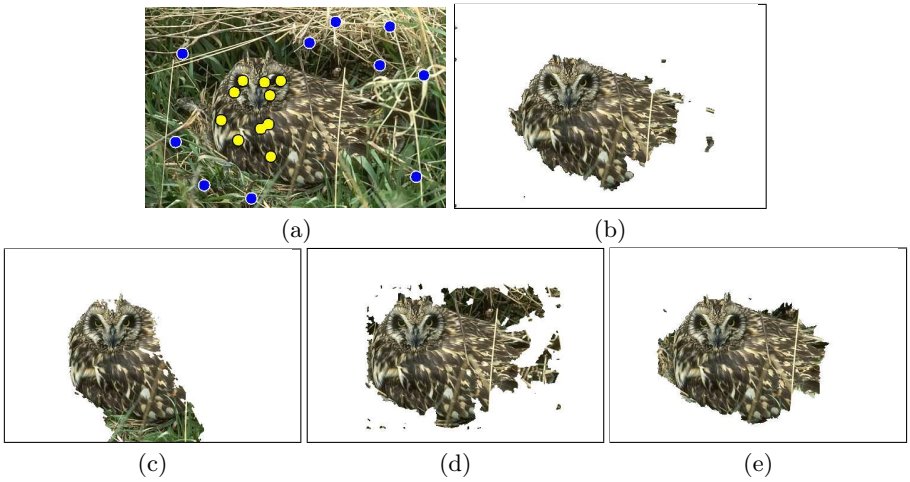


Fig. 1. Segmentation results using the min-cut algorithm [12] with different texture representations. (a) Input image with user specified foreground and background markers. (b) The proposed multi-scale texture representation. (c) Color Histograms [6]. (d) GrabCut [4] using GMMs. (e) Color changes in the pixel neighbourhoods [2].

show that texture segments can be efficiently treated in a multi-scale hierarchy similarly to [8], however building on superpixels.

In our approach, textures are represented by covariance matrices, for which an effective similarity measure based on the symmetric generalized eigenproblem was introduced in [9]. In contrast to the rectangular windows used in [3], covariance matrices are computed from irregular structure-coherent patches, found at different scales. In order to allow for an efficient image-partitioning into scale-coherent regions we devised a novel superpixel method, utilizing watershed segmentations imposed by extrema of an image's mean curvature. However, the suggested framework of multi-scale texture representation is generally applicable for other superpixel methods, such as [10, 11], depending on accuracy and time complexity constraints imposed by the application domain.

We verify the feasibility and meaningfulness of the multi-scale covariance based texture representation by a binary segmentation method based on the min-cut algorithm [12]. Figure 1 shows an example of how different types of texture descriptors influence the min-cut segmentation of a particularly challenging image, consisting of textured regions with highly similar color characteristics.

The remainder of the paper is organized as follows. We present the details of the proposed method in Section 2. Section 3 reports experimental results and compares them to the results obtained using state-of-the-art methods. The paper is concluded with a discussion in Section 4.

2 Our approach

2.1 Superpixels

Probably one of the most commonly used blob detectors is based on the properties of the Laplacian of Gaussians (LoG) or its approximation, the Difference of Gaussians (DoG) [13]. Given a Scale-Space representation $L(t)$ obtained by repeatedly convolving an input image by Gaussians of increasing sizes t, the shape of the intensity surface around a point $\boldsymbol{x}$ at scale t can be described using the Hessian matrix

$$\boldsymbol{H}(\boldsymbol{x}, t) = \begin{bmatrix} L_{xx}(\boldsymbol{x}, t) & L_{xy}(\boldsymbol{x}, t) \\ L_{xy}(\boldsymbol{x}, t) & L_{yy}(\boldsymbol{x}, t) \end{bmatrix}. \tag{1}$$

The LoG corresponds to the trace of the Hessian:

$$\nabla^2 L(\boldsymbol{x}, t) = L_{xx}(\boldsymbol{x}, t) + L_{yy}(\boldsymbol{x}, t), \tag{2}$$

and equals the mean intensity curvature multiplied by two. The LoG's computation results in strong negative or positive responses for bright and dark blob-like structures of size $\sqrt{t}$ respectively. Using this, the position and characteristic scale of blobs can by found by detecting Scale-Space extrema of scale normalized LoG responses [14].

In our approach, we do not directly search for blob positions and scales, but rather use spatial response extrema as starting points for a watershed-based over-segmentation of an image's mean curvature surface. Specifically, we proceed as follows:

1. Computation of LoG responses at scales of $\sqrt{t} = 2^{m/3}$, with $m = 1 \dots M$, where M denotes a predefined number of scales. I.e., we calculate 3 scale levels per Scale-Space octave.
2. Watershed-segmentation:
   (a) Detection of spatial response extrema at all scales. Extrema with low contrast, i.e. those with a minimum absolute difference to adjacent pixels smaller than a predefined threshold, are discarded.
   (b) At each scale, segment the image into regions assigned to positive or negative mean curvature. This is achieved by applying the watershed to the negative absolute Laplacian $-|\nabla^2 L(\boldsymbol{x}, t)|$ using the seeds from (a).

The majority of watersheds thus obtained follow the zero-crossings of the Laplacian; i.e., the edges where the mean curvature of the intensity surface changes its sign. Though, for irregularly shaped blobs, which exhibit significant variations in mean curvature, usually several seed-points are detected. This results in an over-segmentation of regions with otherwise consistent curvature signs. Figure 2 shows a direct comparison of the superpixels produced by our method at a single scale and the normalized-cut based superpixels suggested in [10]. Another method for image over-segmentation, which is partially favoured for its speed, utilizes the Minimum Spanning Tree [11]. However, for larger superpixels, which are needed to stably compute the covariance-based descriptor

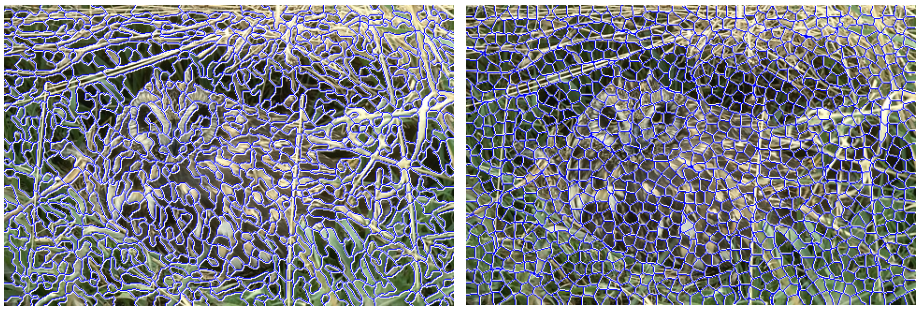


Fig. 2. Left: Superpixels obtained by the proposed method. Right: Superpixels obtained by the method of Ren et al. [10].

on, the regions obtained by this method are highly irregular and often do not align well with object boundaries. Figure 3 shows the effect of using different superpixels in conjunction with the method proposed in this paper. As one can see, our method gives acceptable results compared to the normalized-cut based approach, which needs more than 100 times longer to compute the segmentation.

The outlined approach is similar in spirit to the watershed segmentation of principal curvature images proposed by Deng et al. [15]. In their approach, the first principle curvature image (i.e., the image of the larger eigenvalue of the Hessian matrix) is thresholded near zero and either the positive, or negative remainder is flooded starting from the resulting zero-valued basins. Hence, as opposed to our method, the watersheds follow the ridges of the image's principal curvature surface. In experiments we found that this approach was not suitable for our purposes since it tends to under-segment images, aggressively merging regions with same-signed principal curvature.

2.2 Covariance-based texture similarity

Recently, Tuzel et al. [3] have introduced region covariance matrices as potent, low-dimensional image patch descriptors, suitable for object recognition and texture classification. One of the authors' main contributions was the introduction of an integral-image like preprocessing stage, allowing for the computation of covariances from image features of an arbitrarily sized rectangular window in constant time. However, in the presented work covariances are directly obtained from irregularly shaped superpixels, the aforementioned fast covariance computation is not applicable.

We proceed to give a brief description of the covariance-based texture descriptor in use. The sample covariance matrix of feature vectors collected inside a superpixel is give by:

$$\boldsymbol{M} = \frac{1}{N-1} \sum_{n=1}^{N} (\boldsymbol{z}_n - \boldsymbol{\mu})(\boldsymbol{z}_n - \boldsymbol{\mu})^\top, \tag{3}$$

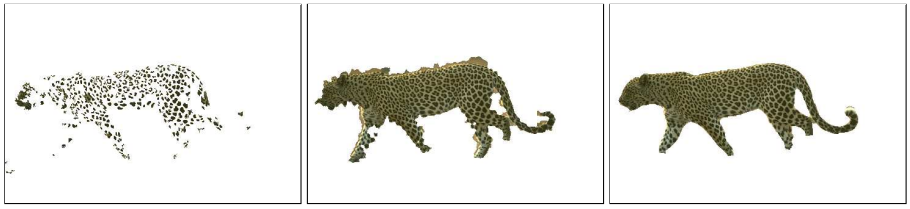


Fig. 3. Effect of superpixels on the final image segmentation. From left to right: Superpixels through a color-based Minimum Spanning Tree [11]. The proposed approach. Superpixels based on normalized-cuts using combined color and textured gradient [10].

where $\boldsymbol{\mu}$ denotes the sample mean, and $\{\boldsymbol{z}_n\}_{n=1\ldots N}$ are the d-dimensional feature vectors extracted at N pixel positions. In our approach, these feature vectors are composed of the values of the RGB color channels R, G, and B and the absolute values of the first derivatives of the Intensity I at the n-th pixel

$$\boldsymbol{z}_n = \left[R_n,\ G_n,\ B_n,\ \left|\frac{\partial I}{\partial x}\right|,\ \left|\frac{\partial I}{\partial y}\right|\right]^\top. \tag{4}$$

The resulting 5×5 covariance matrix gives a very compact texture representation with the additional advantage of exhibiting a certain insensitivity to illumination changes. And, as will be shown experimentally, offers sufficient discriminative power for the segmentation task described in the remainder of the paper.

To measure the similarity $\rho(\boldsymbol{M}_i, \boldsymbol{M}_j)$ of two covariance matrices $\boldsymbol{M}_i$ and $\boldsymbol{M}_j$ we utilize the distance metric initially proposed by Förstner [9]:

$$\rho(\boldsymbol{M}_i, \boldsymbol{M}_j) = \sqrt{\sum_{k=1}^{d} \ln \lambda_k^2(\boldsymbol{M}_i, \boldsymbol{M}_j)}, \tag{5}$$

where the $\{\lambda_k\}_{k=1\ldots d}$ are the eigenvalues obtained by solving the generalized eigenvalue problem

$$\boldsymbol{M}_i \boldsymbol{e}_k = \lambda_k \boldsymbol{M}_j \boldsymbol{e}_k,\ \ k = 1\ldots d \tag{6}$$

with $\boldsymbol{e}_k \neq 0$ denoting the generalized eigenvectors.

The cost for computing ρ is in the order of $O(d^3)$ flops which, due to the low dimensionality of the representation, leads to speed advantages compared to histogram matching methods. For a detailed discussion among the topic, other choices of feature combinations as well as the useful properties of region covariances see [3].

2.3 Foreground and background codebooks

From the covariance-based descriptors proposed in Subsection 2.2 we compute representative codebooks for foreground and background regions. These are used later on to drive the image segmentation.

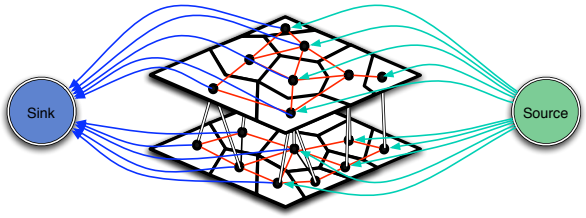


Fig. 4. Two-layer MRF with superpixels detected at two different scales. To avoid clutter not all superpixels are connectd to sink/source nodes.

For the foreground and background codebook, we require user-specified markers, as shown in Figure 1(a), computing the covariance matrices $\boldsymbol{M}_i^{\mathcal{F}}$ and $\boldsymbol{M}_i^{\mathcal{B}}$ from all points under the marker region. Usually, the background contains more textures and cluttered areas requiring more seeds to be establish. Moreover, in applications like object detection or recognition, the background can significantly vary across images while objects of interest usually remain quite consistent in appearance.

To somewhat alleviate the burden of manually selecting many seeds, we propose to avoid the need of background markers by following a simple strategy: We take a rim at the boundary of the image and feed all superpixels under the rim into a hierarchical clustering method with a predefined stopping distance threshold, with the distance between superpixels given by Equation 5. After clustering we take the K most occupied clusters and compute the mean covariance matrix for each cluster out of all covariance matrices belonging to the cluster. For efficiency reasons, we do not calculate the mean covariance matrix by polling over all participating feature vectors, but use the method described in [16, 3], which has its roots in formulations of symmetric positive definite matrices lying on connected Riemannian manifolds. Using this procedure, we arrive at the background codebook matrices $\boldsymbol{M}_i^{\mathcal{B}}$. Of course, the applicability of this ad-hoc technique is limited to cases where the object of interest touches the boundary, or when the rim is not representative enough. However, in most cases the approach lead to background codebooks with sufficient explanatory power for a successful segmentation.

2.4 Multi-scale graph-cut

In order to verify the validity of the covariance-based texture representation, taking into account the superpixel behaviour across different scales we adopted a binary segmentation method based on the min-cut algorithm [12].

Suppose that the image at a certain scale t is represented by a graph $\mathcal{G}_t = \langle \mathcal{V}_t, \mathcal{E}_t \rangle$, where $\mathcal{V}_t$ is a set of all vertices representing superpixels, and $\mathcal{E}_t$ is a set of all *intrascale* edges connecting spatially adjacent vertices. To capture the Scale-Space behaviour we connect the graphs by *interscale* edges forming a set of edges $\mathcal{S}$. We form the entire graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S} \rangle$ consisting of the union of

all vertices $\mathcal{V}_t$, and all intrascale $\mathcal{E}_t$ and interscale edges $\mathcal{S}$. For more clarity, the resulting graph structure is depicted in Figure 4.

The binary segmentation of the graph $\mathcal{G}$ is achieved by finding the minimum-cut [12], minimizing the Gibbs energy,

$$E(\boldsymbol{x}) = \sum_{i \in \mathcal{V}} E_{data}(x_i, \boldsymbol{M}_i) + \lambda \sum_{(i,j) \in \mathcal{E}} \delta(x_i, x_j)\, E_{sm\_im}(\boldsymbol{M}_i, \boldsymbol{M}_j) + \\ + \gamma \sum_{(i,j) \in \mathcal{C}} \delta(x_i, x_j)\, E_{sm\_sc}(\boldsymbol{M}_i, \boldsymbol{M}_j), \quad (7)$$

where $\boldsymbol{x} = [x_0, x_1, \ldots]^\top$ corresponds to a vector with label x_i for each vertex. We concentrate on a bi-layer segmentation where the label x_i is either 0 (background) or 1 (foreground). $\boldsymbol{M}_i$ corresponds to the measurement in the i-th graph vertex, i.e., to a covariance matrix for a given superpixel. The weight constants λ, γ control the influence of the image (intrascale), and interscale smoothness terms respectively; δ denotes the Kronecker delta.

The data term describes how likely the superpixel is foreground or background. The data term for the foreground is defined as

$$E_{data}(x_i = 1, \boldsymbol{M}_i) = \frac{l(\boldsymbol{M}_i, \mathcal{F})}{l(\boldsymbol{M}_i, \mathcal{F}) + l(\boldsymbol{M}_i, \mathcal{B})}, \quad (8)$$

where $l(M_i, \mathcal{F}) = \exp\left(-\min_{k=1\ldots|\mathcal{F}|} \rho(M_i, M_k^{\mathcal{F}})/(2\sigma_1^2)\right)$ stands for the foreground likelihood of the superpixel i. $\boldsymbol{M}_k^{\mathcal{F}}$ denotes the k-th covariance matrix from a foreground codebook set $\mathcal{F}$, and σ_1 is an experimentally determined parameter. As the derivation of the background terms and likelihoods follows analogously, we will omit its description.

The smoothness term describes how strongly neighborhood pixels are bound together. There are two types of the smoothness terms, see Equation (7), one for intrascale neighborhoods, E_{sm_im}, one for interscale neighborhoods, E_{sm_sc}. The intrascale smoothness term using α blending is defined as

$$E_{sm\_im}(M_i, M_j) = \alpha \exp\Big(- \rho(\boldsymbol{M}_i, \boldsymbol{M}_j)/(2\sigma_2^2)\Big) + \\ + (1-\alpha) \exp\left(-\Big(l(\boldsymbol{M}_i, \mathcal{F}) - l(\boldsymbol{M}_j, \mathcal{F})\Big)^2/(2\sigma_3^2)\right), \quad (9)$$

where σ_2 and σ_3 are pre-defined parameters. The interscale smoothness term is only defined for edges between two vertices from neighboring scales when the corresponding superpixels share at least one same image pixel. The weight on

Fig. 5. Importance of inter-scale graph edges. From left to right: Only one lower scale used. Only one higher scale used. Three consecutive scales used.

the edge between superpixels i and j from consecutive scales is set to

$$E_{sm\_sc}(M_i, M_j) = \beta \frac{\text{area}(i \cap j)}{\text{area}(i)} + \\ + (1-\beta) \exp\left(-\Big(l(\boldsymbol{M}_i, \mathcal{F}) - l(\boldsymbol{M}_j, \mathcal{F})\Big)^2 / (2\sigma_3^2)\right). \quad (10)$$

The second term in both Equations (9), (10) increases the dependency of smoothness terms on the foreground likelihood, making it more robust as originally suggested by [8]. However, we rely on this term only partially through the interpolation parameteres α, β, since a full dependency on the likelihood often resulted in compact, but otherwise incomplete segmentations. Figure 5 shows how the use of multiple scales and inter-scale edges improves the segmentation compared to segmentation performed separately for given scales.

3 Experimental Results

We performed segmentation tests on images from the Berkeley dataset[3]. We compare the result to the recent approach proposed by Micusik&Pajdla [2]. Their method looks at color changes in the pixel neighbourhood, yielding superior results on textured images compared to other methods. For both methods the same manually established foreground and background markers were used. To guarantee a fair comparison, the automatic background codebook creation proposed in Section 2.3 was omitted. We present some results where our proposed method performs superior or comparable to [2]. These images typically contain textures with similar colors and are, as stated in [2], the most crucial for their texture descriptor. One must realize that covariance based texture description cannot cope reliably with homogenoeus color regions, see the missing roof of the hut in Figure 6. This should be kept in mind, and use such a descriptor complementary with some color features.

Overall, as experiments show, the newly proposed technique performs very well on textures. The advantage over methods, e.g. [6, 4, 2], is computational effi-

[3] http://www.cs.berkeley.edu/projects/vision/grouping/segbench

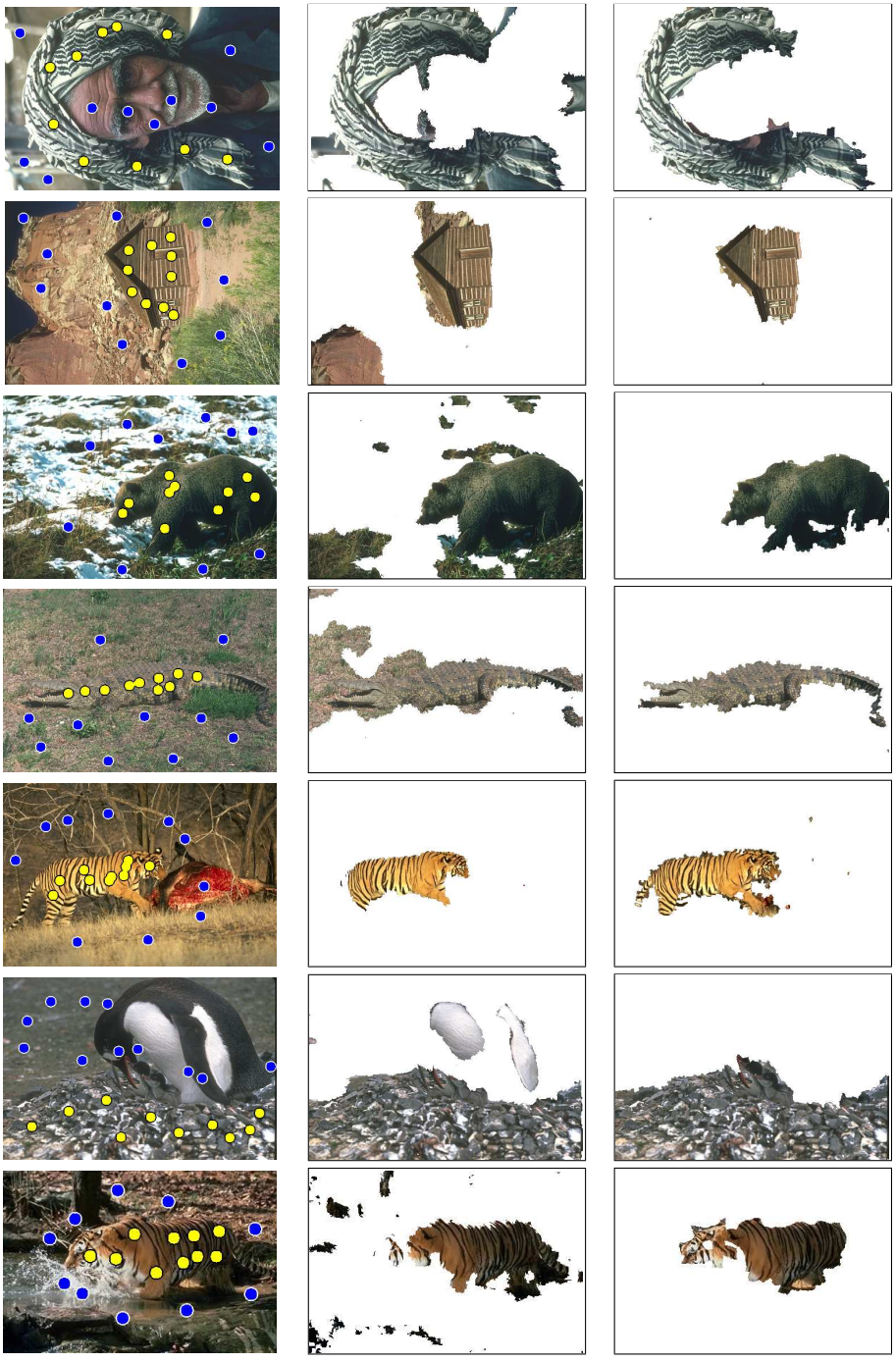
cienty. Moreover, using more accurate superpixels, e.g. [10], improve the accuracy of the result for the price of higher time consumption.

4 Summary and Conclusions

We present an efficient way of representing textures using connected regions, formed by coherent multi-scale over-segmentations. We show the favourable performance on segmentation of textures images. However, our primary goal is not to segment images accurately, but to demonstrate the feasibility of the covariance matrix based descriptor used in a multi-scale hierarchy built on superpixels. The method is aimed at a further use in recognition and an image understanding systems where so accurate segmentation is not required.

References

1. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. IJCV **43**(1) (2001) 7–27
2. Mičušík, B., Pajdla, T.: Multi-label image segmentation via max-sum solver. In: Proc. CVPR. (2007)
3. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Proc. ECCV. (2006) II:589–600
4. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. In: Proc. ACM SIGGRAPH. (2004) 309–314
5. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Probabilistic fusion of stereo with color and contrast for bi-layer segmentation. PAMI **28**(9) (2006) 1480–1492
6. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proc. ICCV. (2001) 105–112
7. Hadjidemetriou, E., Grossberg, M., Nayar, K.S.: Multiresolution histograms and their use for recognition. PAMI **26**(7) (2004) 831–847
8. Turek, W., Freedman, D.: Multiscale modeling and constraints for max-flow/min-cut problems in computer vision. In: Proc. CVPR Workshop. (2006) 180
9. Förstner, W., Moonen, B.: A metric for covariance matrices. Technical report, Dpt. of Geodesy and Geoinformatics, Stuttgart University (1999)
10. Ren, X., Malik, J.: Learning a classification model for segmentation (2003)
11. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV **59**(2) (2004) 167–181
12. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI **26**(9) (2004) 1124–1137
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 91–110
14. Lindeberg, T.: Scale-Space Theory in Computer Vison. Kluwer (1994)
15. Deng, H., Zhang, W., Dietrich, T., Shapiro, L.: Principal curvature-based region detector for object recognition. In: Proc. CVPR (to appear). (2007)
16. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. International Journal of Computer Vision **66**(1) (2006) 41–66



(a) (b) (c)

Fig. 6. Segmentation comparison. (a) Input image with user marked seeds. (b) The method from [2]. (c) Our approach.