Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Segmentation by Adaptive Prediction and Region Merging

Marc Van Droogenbroeck[1] and Hugues Talbot[2]

[1] Department of Electricity, Electronics and Computer Science,
Montefiore B28, Sart Tilman, B-4000 Liège, Belgium
[2] CSIRO Mathematical & Information Sciences,
Locked Bag 17, Building E6B, Macquarie University,
North Ryde, NSW 2113 Australia

Abstract This paper presents a segmentation technique based on prediction and adaptive region merging.
While many techniques for segmentation exist, few of them are suited for the segmentation of natural images containing regular textures defined on non-rectangular segments. In this paper, we propose a description of regions based on a deconvolution algorithm whose purpose is to remove the influence of the shape on region contents. The decoupling of shape and texture information is achieved either by adapting waveforms to the segment shape, which is a time-consuming task that needs to be repeated for each segment shape, or by the extrapolation of a signal to fit a rectangular window, which is the chosen path.
The deconvolution algorithm is the key of a new segmentation technique that uses extrapolation as a prediction of neighbouring regions. When the prediction of a region fits the actual content of a connected region reasonably well, both regions are merged. The segmentation process starts with an over-segmented image. It progressively merges neighbouring regions whose extrapolations fit according to an energy criterion. After each merge, the algorithm updates the values of the merging criterion for regions connected to the merged region pair. It stops when no further gain is achieved in merging regions or when mean values of adjacent regions are too different. Simulation results indicate that, although our technique is tailored for natural images containing periodic signals and flat regions, it is in fact usable for a large set of natural images.

1 Introduction

Image segmentation is an essential tool for most image analysis tasks. In particular, many of the existing techniques for object-based image compression or image interpretation rely strongly on segmentation results. According to [8], image segmentation techniques can be grouped in three different classes: (1) *local* techniques, (2) *global* techniques, and (3) *split*, *merge* and *growing* techniques. The last family of techniques consider two regions to be merged if they are similar and are adjacent or connected to each other.

The segmentation algorithm presented in this paper is part of the third family of techniques except that it is limited to splitting and merging. Following SALEMBIER and GARRIDO [9], our algorithm relies on three notions, addressed in this paper in the following order: a *model* for the description of a region's content, a *merging criterion* and a *merging order*.

Although developed independently, our method has a few similarities with the work presented by PARK and RA [7].

2 A region model

The region model defines how regions are represented. The major challenge is to define the notion of texture because there is no unique interpretation of textures [10] and because one can rarely cope with signals defined on arbitrary shaped windows.

When dealing with natural images, a segmentation technique should essentially be able to handle *flat regions*, *textures* or *disorganised regions*. Segmentation techniques are generally not able to treat disorganised regions. Even worse, most techniques restrict themselves to flat regions in which case the image is first simplified.

2.1 Definition of texture

Texture is observed in many natural images. It can be seen as a the repetition of basic texture elements called *texels* or *textons* [5] made of pixels whose placement obey some rule. If we restrict ourselves to periodic textures, there are essentially two methods to describe these signals on arbitrary shaped windows: (1) adapt waveforms to the shape of each region as proposed by GILGE *et al. [4]* or (2) perform an extrapolation. We opted for solution (2) as solution (1) implies that waveforms be adapted for each shape, which is time-consuming.

In this paper, we refer to *texture* as a signal that can be extended in the spatial domain by an extrapolation mechanism (see Figure 1). Note that the extrapolation does not need to be unique. Also the definition is open for any type of extrapolation. For example, we may want to use texture synthesis methods likewise to the method proposed by EFROS and LEUNG [1] to handle random textures. Also, we could try to determine the periodic proportion and the random proportion of textures (see for example [6]), perform two extrapolations and combine the results. However, in this paper, we did not implement a mixed extrapolation scheme because the extrapolation of random textures is particularly slow.

In the next section, we propose to model regions defined on arbitrarily shaped windows as a restricted set of spectral coefficients after extrapolation. The model is effective for textures but it will be used to describe other types of regions as well.

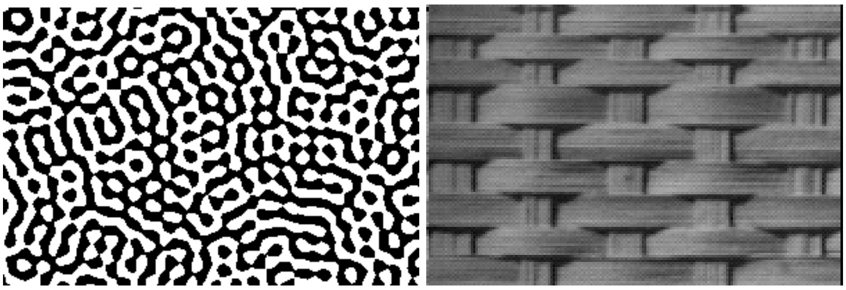

Figure 1. Two textures (the left-hand side texture was provided by LANTUÉJOUL).

2.2 A spectral representation of textures

Textures that are periodic in nature are well suited for a spectral characterisation. Unfortunately, in most cases the signal is provided on an arbitrarily shaped window. Therefore we need to develop a model for texture defined on an arbitrarily shaped window or a technique that can remove the windowing effect. The algorithm presented hereafter belongs to a family of descriptors that extrapolate the signal to fit a rectangular window.

Let an initial texture signal be defined on the original window domain R_w; we assume that the texture is not known outside R_w. The texture signal is given by $p(\boldsymbol{x})$, where $\boldsymbol{x} = (x, y)$, and the window function $w(\boldsymbol{x})$ associated with R_w is defined by

$$w(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \boldsymbol{x} \in R_w \\ 0 \text{ otherwise} \end{cases}$$

The aim of an extrapolation is to enlarge R_w or, equivalently, to have more locations $\boldsymbol{x}$ where $w(\boldsymbol{x})$ is equal to 1. As such the extrapolation process may be modelled as a convolution. Indeed, let $f(\boldsymbol{x})$ be the function to extrapolate and $g(\boldsymbol{x})$ the values observed over R_w. The functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are related by the expression

$$g(\boldsymbol{x}) = w(\boldsymbol{x}) \times f(\boldsymbol{x}) \tag{1}$$

With the transform formalism, the equation is equivalent to the convolution

$$\mathcal{G}(\boldsymbol{u}) = \mathcal{W}(\boldsymbol{u}) \otimes \mathcal{F}(\boldsymbol{u}) \tag{2}$$

It is clear from this equation that the desired spectrum $\mathcal{F}(\boldsymbol{u})$ is affected by the window spectrum $\mathcal{W}(\boldsymbol{u})$. This is why spectral extrapolation techniques are also referred to as *deconvolution* techniques. GERCHBERG [3] and other authors have developed algorithms for the extrapolation of bandwidth-limited signals. Unfortunately one can not generally assume that a texture is a bandwidth-limited signal. Furthermore a deconvolution with respect to the spectrum analysis is an iterative process (except for particular types of signals), which makes it time consuming. In our work, we have used the algorithm proposed by FRANKE [2]. This

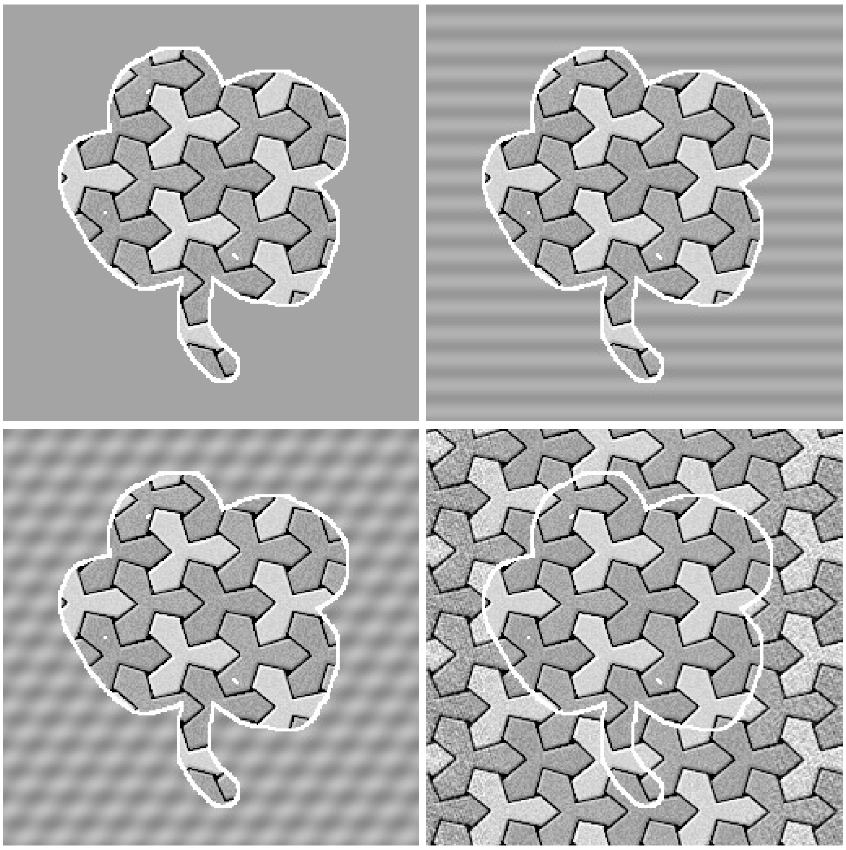

Figure 2. A 256 × 256 synthetic texture restricted to a flower shape and extrapolated signals with respectively 3, 5 and 500 spectral lines (Fourier coefficients) after extrapolation.

method, called *selective deconvolution*, progressively selects the largest spectral lines by iteration.

The selective deconvolution method produces a fairly satisfactory approximation of $f(\boldsymbol{x})$ for textures. This is illustrated in Figure 2.

It should be to emphasised that the spectral transformation used for extrapolation has to be equally sensitive to all directions, which means that the DCT should be avoided. Therefore we have chosen the Fourier transform. Our experiments have shown that:

– a small number of spectral coefficients (typically 5%) is sufficient to extrapolate a regular texture, even if it results that $f(\boldsymbol{x})$ slightly differs from $g(\boldsymbol{x})$ on R_w. This small number of coefficients is sufficient to describe and model a region.

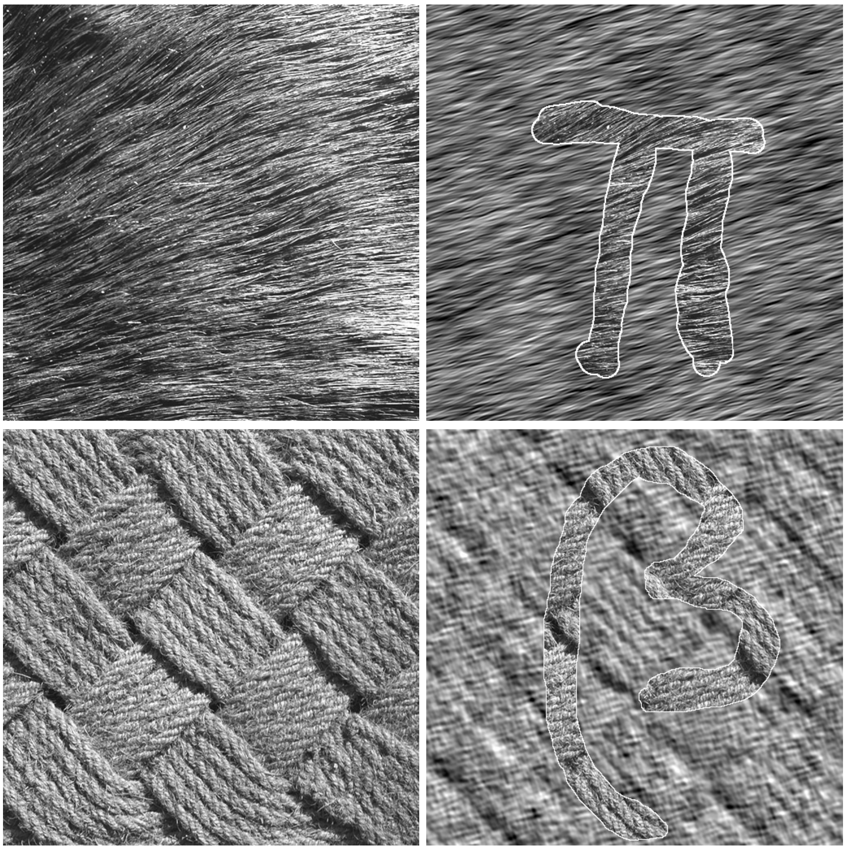

Figure 3. Extrapolation of two original textures 512×512 (left) with 500 spectral lines.

– outside the window, the extrapolation is more reliable for locations close to R_w, and
– larger or convex windows provide better extrapolation results.

Illustrations of the extrapolation on textures taken from the public VisTex database are shown in Figure 3.

3 Segmentation scheme

Our segmentation scheme follows a bottom-up approach starting from an over-segmented partition of the image. Initial partitions are obtained by the watershed algorithm. The input image to the watershed is the result of BEUCHER's morphological gradient [11] in 8-connectivity thresholded to 5. Figure 4 shows the contours of two initial partitions superimposed on the corresponding images.

Figure 4. The contours of two over-segmented images.

It is usual for region merging techniques to be conditioned by the original partitions. Therefore it is important to ensure that all contours have been identified.

The initial partition is used for the construction of an directed adjacency graph $G = (N, E)$, whose nodes N and edges E represent regions and connectivity links in space. Afterwards each edge will be given a merging value.

3.1 Merging criterion

The basis for merging regions is that, if the difference between an extrapolated neighbouring signal and the original signal is smaller than the energy of the original signal, regions should be merged.

Let us assume that R_{w1} and R_{w2} are two neighbouring regions. The knowledge of $p(\boldsymbol{x})$ on R_{w1} yields the extrapolated texture signal denoted $f(\boldsymbol{x})$. We extrapolate $p(\boldsymbol{x}) - \mu_{w1}$ (where μ_{w1} is the statistical mean of $p(\boldsymbol{x})$ on R_{w1}) instead of $p(\boldsymbol{x})$ to save time and to reduce the influence of the DC coefficient that is the largest coefficient. After extrapolation, the signal $f(\boldsymbol{x})$ is compared to the original values of R_{w2}. More precisely, we compare the original variance of a region $\sigma^2_{w2}[p - \mu_{w2}]$ to $\sigma^2_{w2}[p - \mu_{w2} - f]$, which is the residual energy after extrapolation. When $\sigma^2_{w2}[p - \mu_{w2} - f] < \sigma^2_{w2}[p - \mu_{w2}]$, the extrapolation results in an *extrapolation gain EG* defined as

$$EG = \sigma^2_{w2}[p - \mu_{w2}] - \sigma^2_{w2}[p - \mu_{w2} - f] \tag{3}$$

From Parseval's equality, we observe that a similar gain is obtained in the Fourier domain or, more generally, in any spectral domain. However the quality of the extrapolation and, hence the quality of the prediction, relies on the number of selected coefficients. We have found that the average extrapolation gain for positive EG increases with the amount of selected coefficients, but the number of extrapolations leading to positive EG decreases. As a compromise, we have chosen to limit the number of coefficients to 10% of the area of the region to be extrapolated, with an absolute minimum of 3, and maximum of 30 coefficients. Also, we have observed that relying only on the variance leads to unsuitable segmentation results. Therefore the statistical mean difference $\mu_{diff} = \mu_{w2} - \mu_{w1}$ was used. Note that μ_{diff} is equal to the statistical expectation of $p - f - \mu_{w1}$ on R_{w2}, as $E_{w2}[p] = \mu_{w2}$ and $E_{w2}[f] = 0$. In order to obtain a merging criterion, we adopted the following rules:

– very small regions should have a large probability for merging with larger regions.
– small adjacent regions with low variance and close mean values should be merged, even if $EG < 0$.
– EG is more suitable for merging than the mean difference.
– although EG might be positive, regions should not be merged when μ_{diff} is too large.

This led us to propose the merging criterion, called *Merging Factor* (MF) hereafter, that combines the effects of EG and μ_{diff}

$$\begin{array}{ll} \text{if}\, EG > 0 & MF = \alpha_1 \left(1 - \frac{EG}{\sigma^2_{w2}[p-\mu_{w2}]} \times \frac{\mu_{diff}}{\beta}\right) \\ \text{else} & \text{if}\, \mu_{diff} \leq \gamma, \;\; MF = \alpha_2 \frac{\beta}{\beta + \mu_{diff}} \\ & \text{else}\, MF = -1 \end{array}$$

with $\frac{\alpha_1}{\alpha_2} \gg 1$. Four typical situations, summarised in Table 1, occur. The first column corresponds to regions that have the same texture, while the first row holds for regions with similar means.

	$EG > 0$	$EG \leq 0$
$\mu_{diff} \leq \beta$	Good extrapolation, close means: $MF \gg 0$	No extrapolation gain, close means: $MF > 0$
$\mu_{diff} > \beta$	Good extrapolation, different means: $MF < 0$ or $MF > 0$	No extrapolation gain, different means: $MF < 0$

Table 1. Typical combinations of EG and μ_{diff}.

3.2 Merging order

As a direct application of the defined merging factor, we have developed a region merging segmentation algorithm. The proposed region merging algorithm and merging order work as follows:

Evaluate all the edge values of graph G. Note that we have to evaluate both the edges from R_i to R_j and from R_j to R_i as they differ.

Select the edge with the largest MF and merge the corresponding regions.

Update all the edges of regions previously connected to the merged regions.

Repeat steps 2 and 3 as long as the largest MF is positive.

There are many possible strategies for extrapolation and merging but the baseline is that the algorithm should favour regions with a large EG before regions with similar means.

4 Simulation results

All the simulations were performed with the same set of parameter values: $\alpha_1/\alpha_2 = 10^4$, $\beta = 30$ and $\gamma = \beta/2 = 15$. Figure 5 provides some segmentation results for natural images containing textures.

Images (a) and (b) show that the algorithm gradually merges regions belonging to the same texture; it stops when all regions have been merged. The same behaviour is observed in images (d) where large areas in the roof and the wall have been identified. Image (f) shows that textures with similar means but different orientations are not merged. This was the expected behaviour when choosing the FOURIER transform.

5 Discussion and future work

This algorithm does not perform object recognition, thus for example the shirt in Fig. 5(e) and (f) is not segmented as a single object. The textured regions present in the shirt detected by the proposed algorithm are all similar, but modified via affine tranforms. Further work is needed to perform this analyis. However since the number of regions is much reduced from the original, a region comparison approach invariant under affine transform might be feasible.

The starting point of the segmentation is the result of a watershed, which is debatable as the watershed uses edge cues, possibly producing too many small regions from which extracting texture information might be challenging. In practice it doesn't seem that this starting point is a problem.

The texture extrapolation method is relatively slow but the other techniques investigated so far are slower still.

We plan to investigate the possiblity of using a more sophisticated, MDL-based approach for region merging after extrapolation.

6 Conclusions

In this paper, we propose a model for the description of regions and a region merging segmentation technique. The algorithm is based on a merging criterion that compares the variances and the means of a signal to an extrapolated neighbouring region content on the same support. It has been shown to work well for images that contain flat regions or textures.

Acknowledgements

The authors are grateful to Carolyn Evans for fruitful discussions on textures.

References

1. A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1033–1038, Corfu, Greece, September 1999.
2. U. Franke. Selective deconvolution: A new approach to extrapolation and spectral analysis of discrete signals. In *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 30.3.1–30.3.4. IEEE, May 1987.
3. R. Gerchberg. Super-resolution through error energy reduction. *Optica Acta*, 21(9):709–720, 1974.
4. M. Gilge, T. Engelhardt, and R. Mehlan. Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. *Signal Processing: Image Communication*, 1(2):153–180, October 1989.
5. B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981.
6. F. Liu and R. Picard. Periodicity, directionality and randomness: Wold features for image modelling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, July 1996.
7. H. Park and J. Ra. Homogeneous region merging approach for image segmentation preserving semantic object contours. In *Proc. of International Workshop on Very Low Bitrate Video Coding*, pages 149–152, Chicago, October 1998.
8. I. Pitas. *Digital image processing algorithms and applications.* John Wiley & Sons, 2000.
9. P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, April 2000.
10. A. Salvatella. On texture description. Master's thesis, Universitat Autònoma de Barcelona, Bellaterra, September 2001.
11. P. Soille. *Morphological image analysis: principles and applications.* Springer-Verlag, Berlin Heidelberg, 1999.

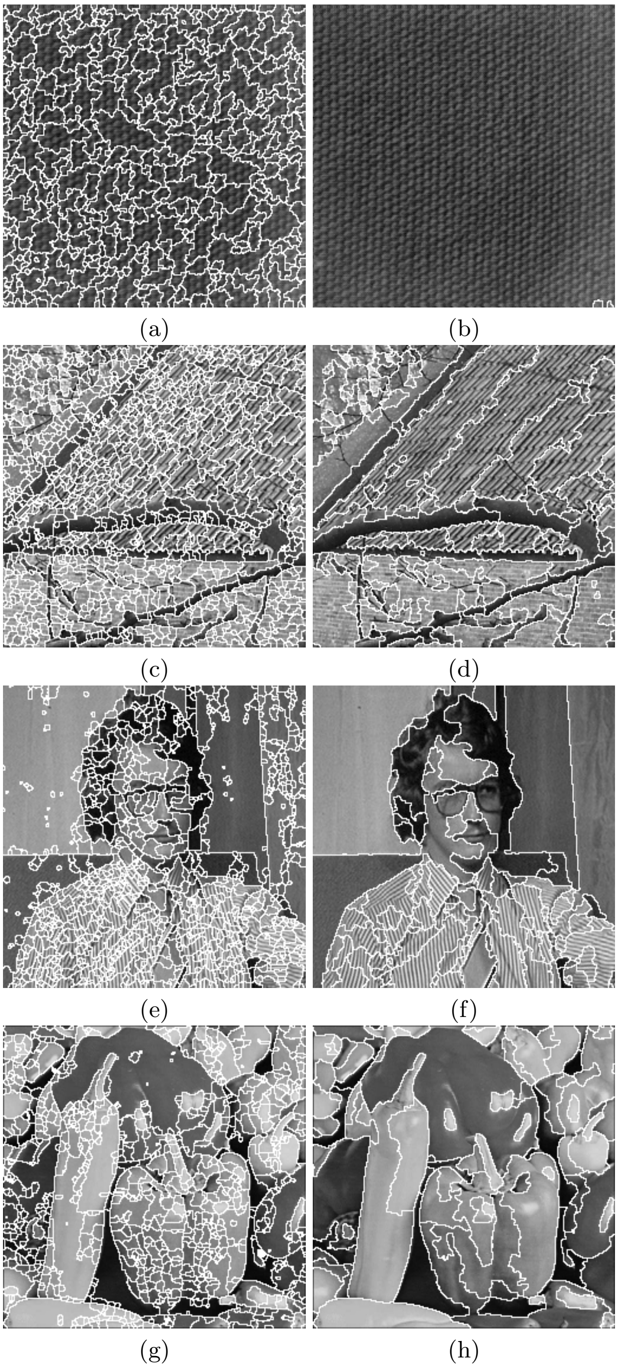

Figure 5. Initial partitions (on the left) and final segmented natural images (on the right).