Morphological Segmentation on Learned Boundaries

Allan Hanbury [a,*] and Beatriz Marcotegui [b]

[a] *Pattern Recognition and Image Processing Group (PRIP), Institute of Computer-Aided Automation, Favoritenstraße 9/1832, A-1040 Vienna, Austria*

[b] *Centre de Morphologie Mathématique, Ecole des Mines de Paris, 35, rue Saint-Honoré, 77305 Fontainebleau cedex, France*

Abstract

Colour information is usually not enough to segment natural complex scenes. Texture contains relevant information that segmentation approaches should consider. Martin et al. (2004) proposed a particularly interesting colour-texture gradient. This gradient is not suitable for Watershed based approaches because it contains gaps. In this paper we propose a method based on the distance function to fill these gaps. Then two hierarchical Watershed-based approaches, the Watershed using volume extinction values and the Waterfall, are used to segment natural complex scenes.

Resulting segmentations are thoroughly evaluated and compared to segmentations produced by the Normalised Cuts algorithm using the Berkeley segmentation dataset and benchmark. Evaluations based on both the area overlap and boundary agreement with manual segmentations are performed.

Key words: image segmentation, watershed, waterfall, normalised cuts, segmentation evaluation, volume extinction values

1 Introduction

Image segmentation is often used as a first step in general object recognition in complex natural scenes, for example in [1–3]. The object recognition is simplified if the regions produced by the segmentation algorithm already correspond

[*] Corresponding author
Email addresses: `hanbury@prip.tuwien.ac.at` (Allan Hanbury), `marcoteg@cmm.ensmp.fr` (Beatriz Marcotegui).
URL: `http://www.prip.tuwien.ac.at/people/hanbury` (Allan Hanbury).

to “meaningful” objects. Nevertheless, unless it is made clear what the objects of interest in a scene are, even humans may not agree on the best segmentation of such a scene [4]. If a number of people are instructed to segment an image of an arbitrary scene, then each person will most likely produce a different segmentation of the image. This could be due to different interpretations of the scene or considering the scene at different scales.

Many algorithms for image segmentation are available, two of the most popular being the Normalised Cuts (NCuts) [5] and the Watershed [6]. Both of them require a way of measuring the similarity (or difference) between pixels in an image. The Watershed, for example, is usually applied to the gradient of an image. A particularly promising algorithm was presented in [7] for detecting the boundaries in an image based on brightness, colour and texture cues learned from human segmentations of a set of images. It calculates for every pixel in an image the probability that it is part of a boundary. Unfortunately, these boundaries are not suitable to be used as a gradient for a Watershed algorithm due to gaps in the boundary lines. In this paper, we present a solution to this problem. This solution consists in filling the small gaps by applying a distance transform to the boundary image. We make use of the recently introduced distance function for greyscale images, the *quasi-distance function* [8]. This has the advantage that it can be applied directly to the greyscale boundary probability image and that no parameters need to be set. This is in contrast to the classic distance function, for which the boundary probability image would first have to be thresholded, and to smoothing filters, for which the size of the filter must be chosen.

Two different hierarchical segmentation approaches based on the Watershed are studied: the *hierarchy based on the volume extinction values* of the Watershed catchment basins that produces a partition with a specified number of regions, and the *Waterfall* that is iterated a given number of times producing a variable number of regions according to the image complexity. The segmentations produced are thoroughly evaluated and compared to segmentations produced by the NCuts algorithm using area and boundary-based segmentation evaluation measures. The manual segmentations from the Berkeley Segmentation Dataset and Benchmark are used as ground truth.

The paper is structured as follows. Section 2 is devoted to an overview of boundaries based on learning [7] and a presentation of the technique we propose to close the gaps in order to make them suitable for the Watershed approach. Section 3 summarises the two hierarchical Watershed algorithms. Area and boundary based segmentation evaluation is discussed in Section 4 and applied to evaluate the segmentations in Section 5. Section 6 concludes.

2 Gradients Based on Learning

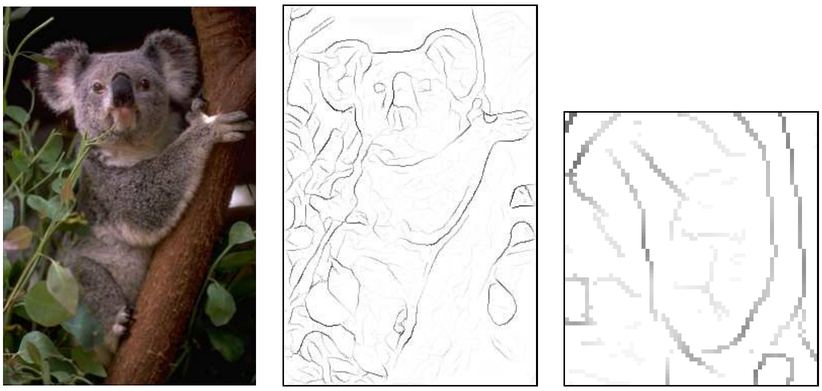
The literature abounds with algorithms for computing gradients of both colour and greyscale images. While most of them may be used in conjunction with a Watershed algorithm to segment an image, almost all gradients tend to suffer from many strong responses in highly textured image regions, which prevents them from clearly delimiting textured areas. To solve this problem, Martin et al. introduced the boundaries based on learning [7], which we briefly review in the first part of this section. While these boundaries are better at delimiting highly textured areas, they cannot be directly used with a Watershed segmentation technique as the boundaries are usually not closed. We solve this problem by applying a distance function to close the gaps in the boundary image, as described in the second part of this section.

2.1 Boundaries Based on Learning

The boundaries based on learning approach introduced by Martin et al. [7] makes use of brightness, colour and texture gradients to compute the boundaries. To calculate the gradients, a circular region is moved over the image. At each pixel, for a number of orientations of a line dividing the circle into two halves, the χ^2 histogram difference is evaluated for histograms of the features in the two halves. For colour, three 32-bin histograms of the values of L^*, a^* and b^* in the CIELAB space (taken separately) are used; for texture, one 64-bin histogram of the textons used in [7] is used. For each feature, the gradient is taken to be the maximum value obtained over all the orientations of the line dividing the circle. The result of this algorithm is therefore a vector of four gradient values at every pixel (3 colour and 1 texture).

These four gradients are combined to form a boundary probability. The weight for each gradient is obtained by logistic regression. As ground truth, human segmentations of the 200 images in the training group of the Berkeley segmentation dataset were used. Every pixel marked as a boundary by at least one person was considered as part of the ground truth boundaries. We made use of the weights provided by the authors of [7] in their software [1] . The resultant boundary probabilities are in the range $[0, 1]$. As an example, the boundaries detected in Figure 1(a) are shown in Figure 1(b).

[1] Downloadable on the Berkeley Segmentation Benchmark page: `http://www.cs.berkeley.edu/projects/vision/grouping/segbench/`



(a) (b) (c)

Fig. 1. (a) An image and (b) its boundary probabilities (darker pixels indicate higher probability). (c) Detail of (b) showing the gaps in the contour.

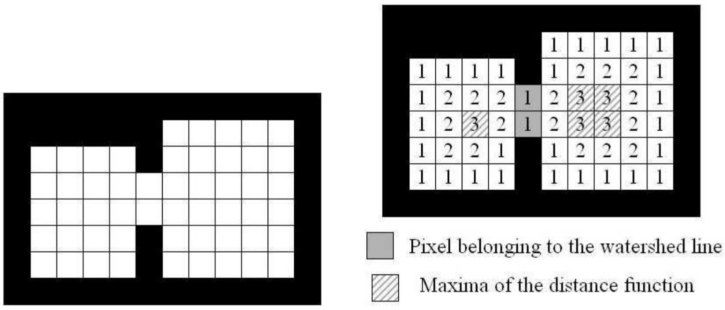
2.2 Distance Functions Applied to Boundary Images

The boundary image produced by the algorithm outlined in the previous section (see Figure 1(b)) seems to be a good gradient approximation: the values in the koala fur are low, while its body is well delimited. But if we look more closely (see Figure 1(c)), we can clearly see gaps in the boundary lines. This results in very few local minima in the boundary image (often only one), which makes applying Watershed-based segmentation difficult. Our solution to the problem is to attempt to close the gaps by calculating a *distance function* of the boundary image.

The classic distance function takes as input a binary image. It associates with each foreground pixel the distance to the closest background pixel (see Figure 2 for an example). Overlapping binary objects may be segmented using the well known approach [9,10] that combines the distance function and the Watershed. If a connected component contains several particles, its distance function will have a maximum in each particle. Thus maxima of the distance function (pixels represented with a hatched pattern in Figure 2(b)) mark the different particles contained in the connected component. The Watershed applied to the complement of the distance function (grey pixels in figure 2(b)) correctly separates the different particles of the connected component

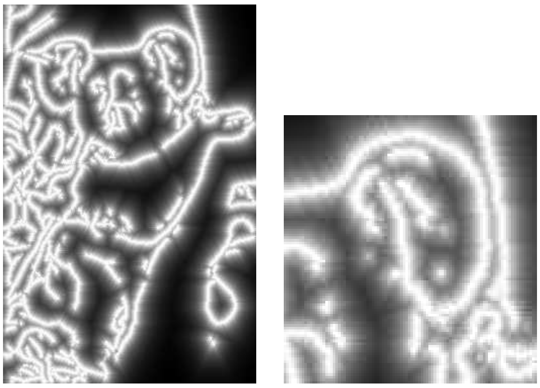
As the classic distance function must be applied to a binary image, applying it to the boundary image would require that the boundary image first be thresholded. To avoid the necessity of choosing this threshold, we make use of the quasi-distance introduced by Beucher [8]. The quasi-distance qd of a greyscale image I is defined as:

$$qd(x,y) = \arg\max_i(\epsilon_{i-1}(x,y) - \epsilon_i(x,y)) \qquad (1)$$



(a) (b)

Fig. 2. (a) Binary image. (b) Associated distance function.



(a) (b)

Fig. 3. (a) Complement of the quasi-distance on the boundary image. (b) Detail of (a).

where ϵ_i is the morphological erosion of size i, and (x, y) is a given pixel of the image I. In other words, the quasi-distance associates with each pixel (x, y) the size i of the erosion that produces the biggest change in greylevel, among all possible sizes of erosions. Thus the quasi-distance is able to characterize the size of objects in a greylevel image without first applying a threshold.

If we take the boundaries detected by the Martin et al. algorithm as the background, the distance function encodes the shortest distance to each of the detected boundary lines. The value of the distance function on the detected boundaries will be zero. Within small gaps in the detected boundaries the value of the distance function will be small. As we want the Watershed to take these boundaries as the edges of regions, we use the complement of this distance function, in which the detected boundaries will have the maximum possible value. The lower values of the distance function in small gaps lead to higher values in the complement, effectively closing the gaps in the topographical representation of the image used by the Watershed. The complement of the quasi-distance function applied to the boundary image in Figure 1(b) is shown in Figure 3(a), with a zoomed in area shown in Figure 3(b).

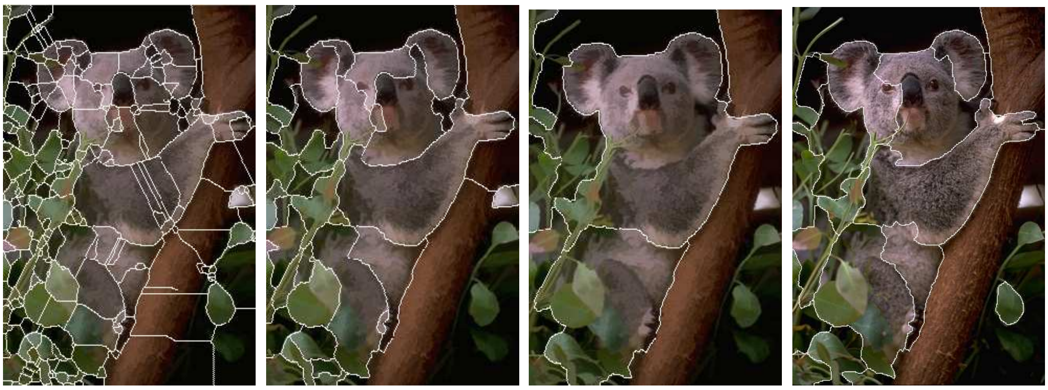


Fig. 4. (a) Watershed of the complement of the quasi-distance on the boundary probability image (level 0). (b) Waterfall level 1. (c) Waterfall level 2. (d) Watershed using volume extinction values (18 regions).

3 Waterfall and Volume Extinction Value Hierarchies

The Watershed algorithm usually leads to a strong over-segmentation of the image. Several hierarchical approaches have been proposed to overcome this problem. In this paper we will study two of these methods: the hierarchy based on the volume extinction values [11,12] and the Waterfall [13].

3.1 Watershed Based on Volume Extinction Values

During the flooding process of the Watershed a measure is associated with each merging. This measure, called the *extinction value*, corresponds to a geometric measure of the smallest lake involved in the merging and is used to evaluate the relevance of the merging. After the flooding process is completed, the “extinction” of small lakes is allowed (the merging is performed) whereas the biggest lakes (according to the measure) are preserved (the merging is not performed). In order to obtain a partition with N regions, the $N-1$ fusions with highest extinction values are avoided. Several measures have been proposed in the literature: the area of a lake that tries to obtain big regions regardless of their contrast, the depth of a lake that privileges contrasted regions regardless of their size and the volume of a lake that combines size and contrast. The use of volume provides a good approximation of human perceptual importance of a region and leads to the most useful segmentations. Figure 4(d) shows the segmentation of Figure 1(a) into 18 regions by this algorithm. We abbreviate this segmentation method as *Volume Watershed*.

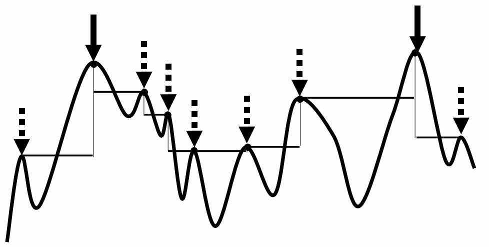


Fig. 5. Waterfall principle.

3.2 Waterfall

The Waterfall [13] is a Watershed-based hierarchical segmentation approach. It consists in two steps:

- first, each region is filled with the value of the lowest pass point of its frontier. The pass point is the pixel where, during the flooding process associated with the Watershed, neighbouring "lakes" (regions) meet for the first time. A morphological reconstruction may be used for this purpose.
- second, the Watershed of the resulting image is computed.

In the example of Figure 5 the Watershed lines are indicated by arrows and only solid line arrows will be preserved by the Waterfall.

The process may be iterated until a single region covers the whole image, establishing a hierarchy among the frontiers produced by the Watershed. An efficient graph-based Waterfall algorithm is presented in [14].

An example of the Waterfall algorithm applied to the complement of the quasi-distance function of the detected boundary image is shown in Figure 4. Image (a) shows the result of applying the Watershed algorithm to the complement of the quasi-distance function, image (b) is the result of applying the Waterfall algorithm once (referred to as level 1 of the hierarchy) and image (c) is the result of two iterations of the Waterfall (level 2).

3.3 Complete Segmentation Algorithm

We summarise here the algorithm used to perform the segmentation:

(1) Calculate the learning-based boundaries (we use the combined colour and texture gradients [7]).
(2) Calculate the complement of the quasi-distance function on the inverse boundary image.
(3) Calculate the final partition using the Waterfall or the volume extinction

value hierarchy on the complement of the distance function.

4 Segmentation Evaluation

A number of methods for evaluating segmentations when ground-truth is available have been proposed. They measure the similarity between a segmentation and a ground-truth segmentation by considering either the amount of region overlap [4], the proximity of the region boundaries to each other [7], or measurements of cluster goodness [15].

We evaluate our algorithms by using the area based method from [4] and a newly introduced boundary based method making use of the distance function. As ground-truth we use the 300 colour images and their human segmentations from the Berkeley Segmentation Dataset and Benchmark. For each image, at least 5 segmentations produced by different people are available.

4.1 Area Based Error Measure

Two measures of the difference between two segmentations based on the overlapping areas of the segmentation regions are introduced in [4]: the Global and Local Consistency Errors (GCE and LCE). As the GCE is a tougher measure, we only use this measure.

Let S_1 and S_2 be two segmentations of an image. The region $R\left(S, p_i\right)$ is the set of pixels corresponding to the region in segmentation S that contains pixel p_i. A segmentation S_1 is a *simple refinement* of S_2 if at every pixel p_i, $R\left(S_1, p_i\right) \subseteq R\left(S_2, p_i\right)$. The GCE is defined in terms of the local refinement error:

$$E\left(S_1, S_2, p_i\right) = \frac{\left|R\left(S_1, p_i\right) \setminus R\left(S_2, p_i\right)\right|}{\left|R\left(S_1, p_i\right)\right|} \tag{2}$$

where $\setminus$ denotes the set difference and $|x|$ is the cardinality of set x. As can be seen, this error measure is not symmetric. If, at pixel p_i, $R\left(S_1, p_i\right) \subseteq R\left(S_2, p_i\right)$, then $E\left(S_1, S_2, p_i\right) = 0$, but $E\left(S_2, S_1, p_i\right) > 0$. The GCE of segmentations S_1 and S_2 is defined as

$$\mathrm{GCE}\left(S_1, S_2\right) = \frac{1}{n} \min\left\{\sum_i E\left(S_1, S_2, p_i\right), \sum_i E\left(S_2, S_1, p_i\right)\right\} \tag{3}$$

where n is the number of pixels and the sums are over all pixels. If S_1 (resp.

S_2) is a simple refinement of S_2 (resp. S_1), then $\mathrm{GCE}\,(S_1, S_2) = 0$. As the local refinement error is not symmetrical, the minimum of the local refinement error sums calculated in both directions is taken. Note that this measure is zero if one of the segmentation is only a single region covering the whole image, or if each pixel of one of the images is taken to be a region. This measure is therefore only useful if segmentations with a similar number of regions are compared.

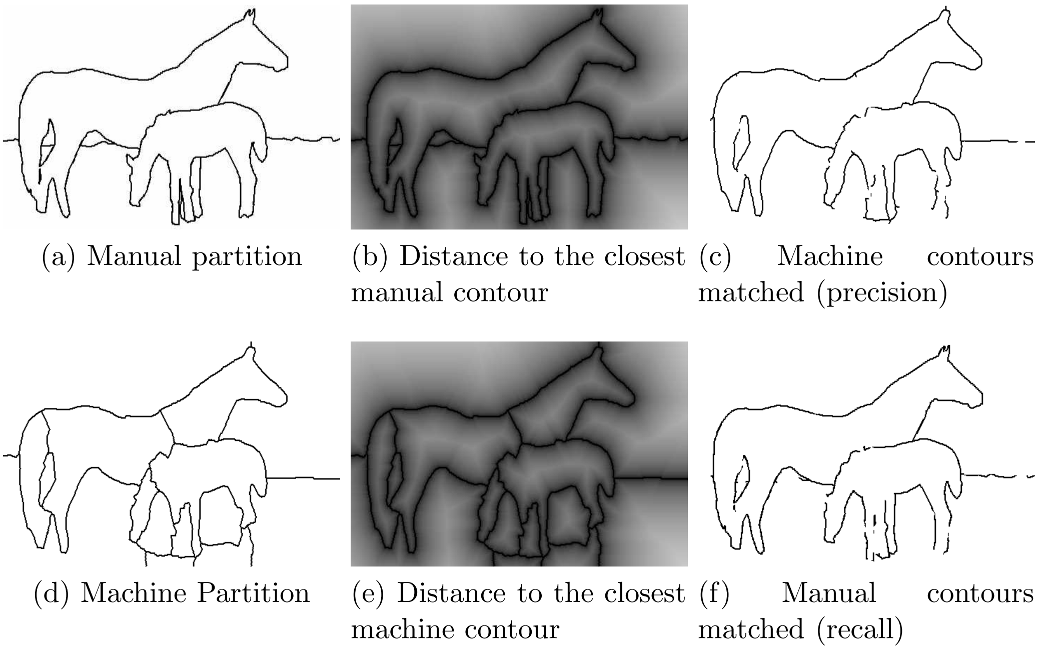
4.2 Boundary based error measure

Martin et al. [7] introduced a boundary based error measure. They first compute the correspondence between machine boundary and human labelled boundary maps. This correspondence is performed by minimizing the distance in the image plane of pairs of matched pixels. If this distance is beyond a given threshold d_{max}, they declare boundary pixels to be non-hits. As this boundary pixel matching procedure is time consuming, the authors propose strategies to speed up the process through the use of a bipartite graph matching algorithm.

We propose a simpler strategy based on the distance function. It allows the evaluation of the quality of a boundary map without a previous bipartite graph matching. Figure 6 illustrates the proposed evaluation algorithm. Let us evaluate the quality of an automatic segmentation (Figure 6(d)) with respect to a human made partition (Figure 6(a)). Figure 6(b) presents the distance of each image pixel to its closest human labelled boundary pixel. This operation has a complexity of $O(n)$. For each machine contour we take the value of the computed distance (Figure 6(b)). Pixels with a distance value lower than d_{max} are considered as matched (i.e. a human boundary pixel is close enough) and pixels above d_{max} are considered as false positives. Figure 6(c) shows the pixels that have been “matched” (close enough to a manual labelled boundary). The parameter d_{max} allows one to vary the maximum deviation accepted to match a contour point. We define the precision (measure currently used in the indexing context) as the ratio of matched machine contour pixels with respect to the number of contour pixels detected by the automatic algorithm:

$$\text{precision} = \frac{\text{Number of Machine Contour Pixels Matched}}{\text{Number of Machine Contour Pixels}} \tag{4}$$

We can repeat the process and compute the distance of each image pixel to the closest machine contour (Figure 6(e)) and consider how many human labeled boundary pixels are close enough ($< d_{max}$, see Figure 6(f)) to machine boundary pixels. We define the recall as the ratio between manual contour



(a) Manual partition (b) Distance to the closest manual contour (c) Machine contours matched (precision)

(d) Machine Partition (e) Distance to the closest machine contour (f) Manual contours matched (recall)

Fig. 6. Computation of the boundary error measure (a) Manual segmentation (b) Distance to the closest manual contour. (c) Machine contours matched (considered for precision calculation). (d) Machine segmentation. (e) Distance to the closest machine contour. (f) Manual contours matched (considered for recall calculation).

pixels matched with respect with the number of manual contour pixels:

$$\text{recall} = \frac{\text{Number of Manual Contour Pixels Matched}}{\text{Number of Manual Contour Pixels}} \tag{5}$$

A high recall is obtained if most human labelled boundary pixels are closer than d_{max} to a machine boundary pixel.

Each human segmentation therefore gives rise to a pair of precision-recall values (P, R). To summarise these values in a single figure, the F-measure, defined as $F = 2PR/(R + P)$ is used.

The proposed measures are similar to those proposed by Martin et al. [7] because they consider that a simple binary counting after matching is sufficient. The advantage of our method is that we avoid the matching procedure which is complex and time consuming.

5 Results

We compare the segmentations of the two hierarchical Watershed based approaches operating on the complement of the quasi-distance of the colour and texture boundary image to the segmentations produced by the NCuts algorithm. For NCuts, we use the implementation by Shi[2] [5], which requires that one specifies in advance the number of regions required. We applied the NCuts algorithm to two types of weighting function. The first is calculated from the multiscale greyscale gradient [16] using the intervening contour method originally introduced in [17] and included in the NCuts implementation used. The second is calculated from the learning based colour and texture boundary probability image using a simplified intervening contour method that does not include orientation energy information (which is not available in the boundary probability images). The weighting based on the former was found to lead to better segmentations, so we present the results using this weighting function in the following analysis.

For the Waterfall algorithm, as level 1 of the hierarchy is almost always oversegmented, we evaluate level 2. The mean of the number of regions obtained over all 300 images by this segmentation algorithm is 5.8, so we compare them to the NCuts algorithm producing 6 regions.

For the Watershed using volume extinction values, the number of regions should be specified, as is the case for the NCuts. We chose 18 regions for the comparison, the mean number of regions over all the human segmentations.

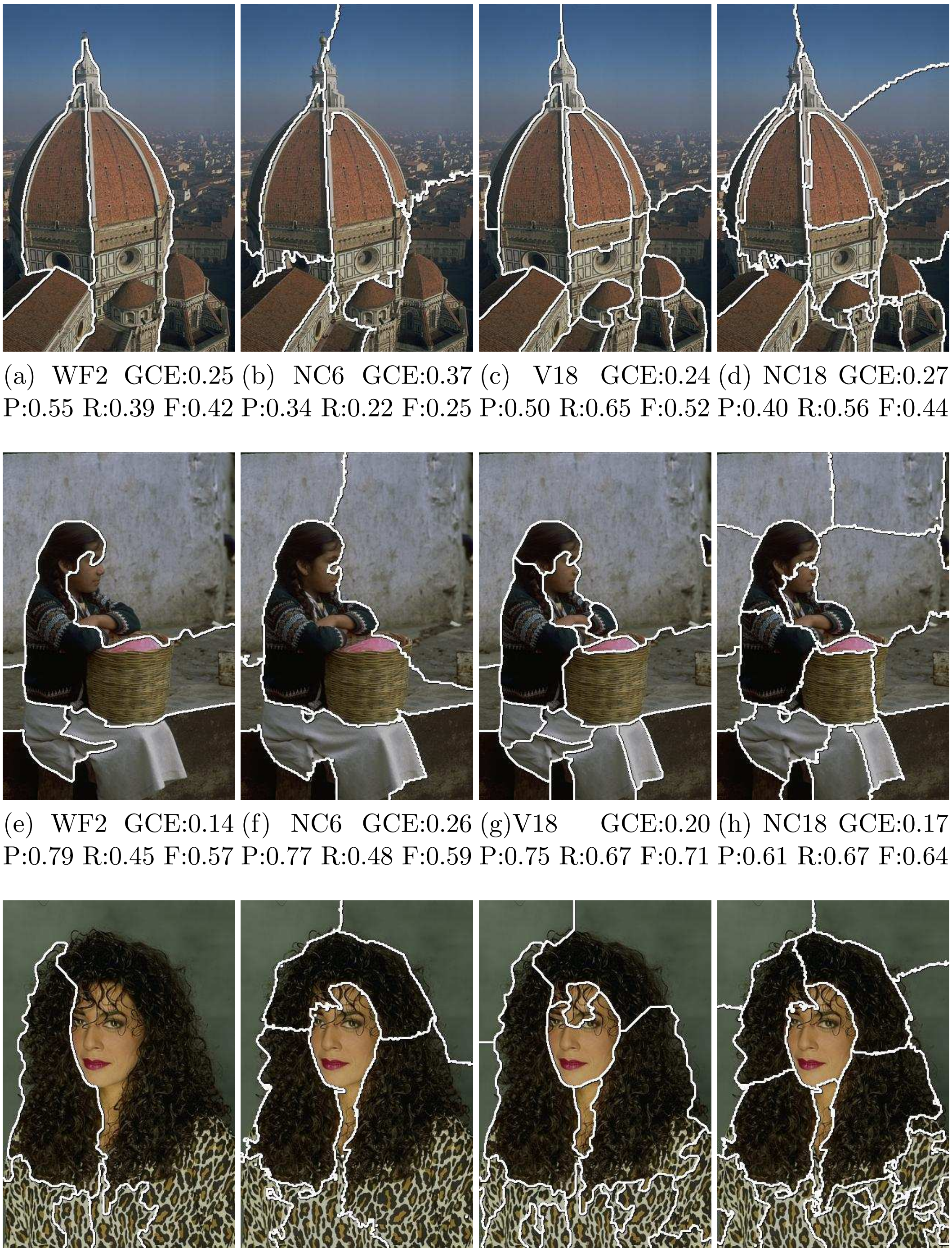
Segmentation results produced by these algorithms applied to the 300 images of the Berkeley segmentation dataset are available on the web[3]. Some example segmentations are shown in Figure 7.

To evaluate a segmentation algorithm, it was first applied to each of the 300 images. Then, for each image, the GCE (area based measure), precision, recall and F-measure (boundary based measures) of the segmentation produced by the algorithm with respect to each of the available human segmentations for that image were calculated. The mean values of these measures were calculated as the mean over the measure for each manual segmentation.

One of the disadvantages of the boundaries based on learning is their long computation time. For the test images used (of size 321 × 481 pixels), the mean computation time for the boundaries was 1.9 minutes on a Pentium 4 computer. The Waterfall and Watershed segmentations require on average 0.05 seconds irrespective of the number of regions produced. In order for the NCuts segmentation to be computationally tractable, the size of the image is reduced to 160 × 240 pixels. The gradient used for the NCuts requires on average 9 seconds computation time on such a reduced size image. For comparison, detecting the learning based boundaries on a reduced size image requires on average 21 seconds. The NCuts segmentation requires an average

[2] available here: `http://www.cis.upenn.edu/~jshi/software/`

[3] `http://muscle.prip.tuwien.ac.at/IVC_segresult/`



(a) WF2 GCE:0.25 P:0.55 R:0.39 F:0.42
(b) NC6 GCE:0.37 P:0.34 R:0.22 F:0.25
(c) V18 GCE:0.24 P:0.50 R:0.65 F:0.52
(d) NC18 GCE:0.27 P:0.40 R:0.56 F:0.44

(e) WF2 GCE:0.14 P:0.79 R:0.45 F:0.57
(f) NC6 GCE:0.26 P:0.77 R:0.48 F:0.59
(g)V18 GCE:0.20 P:0.75 R:0.67 F:0.71
(h) NC18 GCE:0.17 P:0.61 R:0.67 F:0.64

(i) WF2 GCE:0.38 P:0.52 R:0.27 F:0.35
(j) NC6 GCE:0.33 P:0.63 R:0.55 F:0.59
(k) V18 GCE:0.26 P:0.47 R:0.56 F:0.51
(l) NC18 GCE:0.17 P:0.61 R:0.67 F:0.64

Fig. 7. Examples of segmentations produced by the four methods tested on three images. The leftmost column shows the waterfall level 2 (WF2), the column second from left shows the NCuts with 6 regions (NC6), the third column shows the volume Watershed with 18 regions (V18) and the rightmost column shows the NCuts with 18 regions (NC18). Below each image are the values of each of the segmentation comparison measures: Global Consistency Error (GCE), Precision (P), Recall (R) and F-measure (F).

Method	GCE	Precision	Recall	F-measure
WF level 2	0.19	0.64	0.37	0.44
NCuts (6 regions)	0.29	0.52	0.38	0.42
WS Vol (18 regions)	0.22	0.54	0.60	0.55
NCuts (18 regions)	0.23	0.44	0.58	0.48

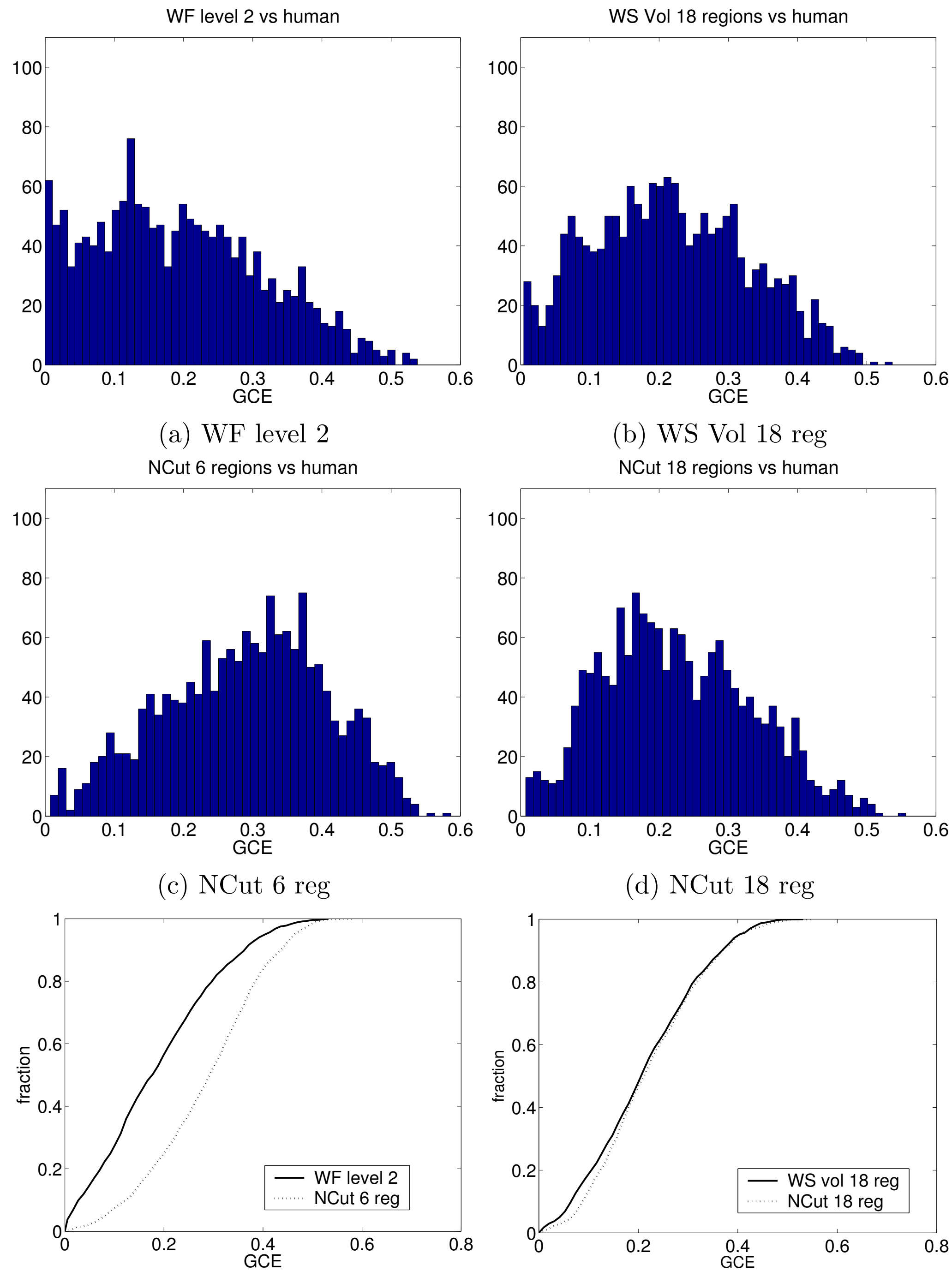
Table 1
The mean values over all manual segmentations for GCE, Precision, Recall and F-measure for various segmentation algorithms. These are: the Waterfall algorithm (WF) for level 2 of the hierarchy, the Watershed using volume extinction values (WS Vol) and the NCuts algorithm. Note that better agreement with the ground-truth is indicated by smaller GCE values, but by larger precision, recall and F-measure values.

of 23 seconds to segment the image into 6 regions, and 35 seconds to segment it into 18 regions. The region labelled images produced by the NCuts algorithms were enlarged to the original image size by pixel replication, leading to ragged region boundaries (see Figure 7).

5.1 Area Based Comparison

The mean GCE values for all segmentation algorithms evaluated are shown in the left column of Table 1. Histograms showing the distributions of the GCE values of each of the manual segmentations are shown in Figure 8(a)-(d). Cumulative histograms are shown in Figure 8(e)-(f). These curves indicate the fraction of GCE values that are below the GCE value on the x-axis. Algorithms with lower GCE values will produce curves that climb faster and hence lie more to the left. Note that some of the segmentations at level 2 of the Waterfall hierarchy consist of only one region. As the GCE for such a segmentation is zero, we chose to use level 1 of the Waterfall hierarchy if level 2 contained only a single region.

For the segmentations into 18 regions, the GCE values produced by both the Volume Watershed and the NCuts are almost identical. On the other hand, when segmenting the image into a small number of regions, the mean GCE for the Waterfall level 2 is much smaller than for the NCuts with 6 regions. This suggests that the regions produced by the Waterfall method are a better match to the human segmentations, although this is discussed further after considering the boundary based evaluation.



(a) WF level 2

(b) WS Vol 18 reg

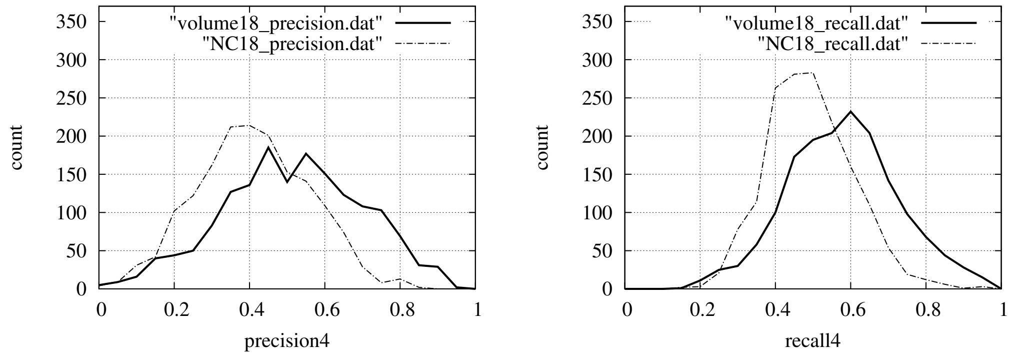
(c) NCut 6 reg

(d) NCut 18 reg

(e) Cum. Hist. WF level 2, NCut 6

(f) Cum. Hist. WS Vol 18, NCut 18

Fig. 8. Histograms of the distribution of the GCE for each of the human segmentations for: (a) level 2 of the Waterfall algorithm, (b) the Watershed with volume extinction values for 18 regions, (c) the NCuts algorithm with 6 regions, and (d) the NCuts algorithm with 18 regions. (e) Cumulative histogram of (a) and (c). (f) Cumulative histogram of (b) and (d).



(a) precision histogram

(b) recall histogram

Fig. 9. Comparison of hierarchy based on volume extinction values and NCuts with 18 regions. (a) Precision histogram. (b) Recall histogram.

5.2 Boundary Based Comparison

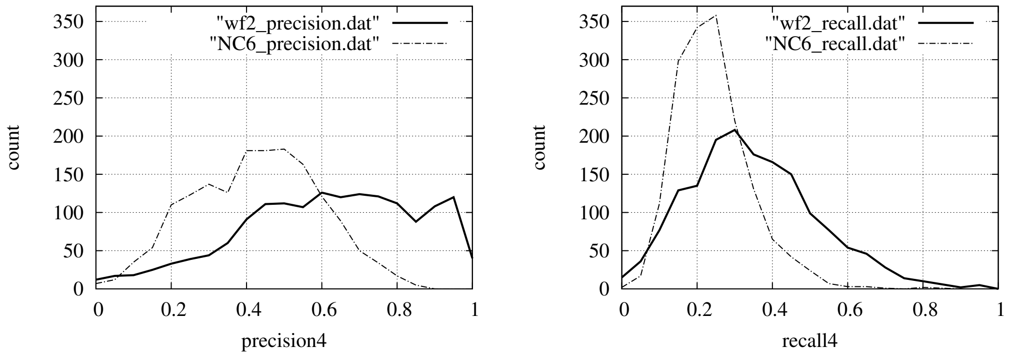
Figure 9 presents the comparison of segmentation results using the NCut approach and the hierarchy based on the volume extinction values, both with 18 regions. The evaluation method used is the one presented in Section 4.2 with d_{max}=4 which represents 0.70% of the image diagonal. In Figure 9(a) we have represented the histogram of the precision (contours of the automatic segmentation that are closer than d_{max} to a manually drawn boundary) and in Figure 9(b) the histogram of the recall (contours of the manual segmentation that are closer than d_{max} to an automatic contour). We see that the hierarchy based on volume extinction values generally outperforms the NCuts, because its histogram lies further to the right than the NCuts histogram, meaning that the precision and recall are concentrated at higher values.

Figure 10 presents the comparison of the NCut with 6 regions and the second level of the hierarchy based on Waterfalls. As stated before, the first level of the Waterfall hierarchy is kept if the second level contains only one region. Again, the histograms of the precision and recall for the Waterfall lie further to the right than the NCut histograms.

The mean precision, recall and F-measure over all the human segmentations are shown in the rightmost columns of Table 1. The mean F-measures resulting from the segmentations are low, demonstrating that the segmentations produced by all methods are not at all close to human segmentations.

5.3 Discussion

In this section we analyse the global trends as well as relating them to segmentations of specific images. For the segmentations in Figure 7, the mean



(a) precision histogram

(b) recall histogram

Fig. 10. Comparison of Waterfall level 2 and NCuts with 6 regions. (a) Precision histogram. (b) Recall histogram.

values of GCE, precision, recall and F-measure calculated over the manual segmentations corresponding to each image are shown for each segmentation.

We begin by considering the evaluation based on boundaries. In Table 1, the mean recall values for the two methods producing a low number of regions are similar to each other, as are the recall values for the two methods producing 18 regions. Larger differences are visible in the mean precision values, where the morphological methods have larger values. This demonstrates that in general, while both segmentation methods find a similar proportion of the segment boundaries corresponding to the ground truth, the morphological methods find fewer false boundaries. A possible explanation for this is that the NCuts has a tendency to produce regions of similar size, often leading to an over-segmentation of homogeneous regions. These spurious region boundaries lead to a lower precision as they are in general not close to any lines in the manual segmentations. A good example can be seen in Figure 7, where images (g) and (h) have identical recall values, but the precision for the Watershed approach is much larger than for the NCuts approach.

For the methods producing 18 segments, the mean GCE values differ only by 0.1. This is most likely because the GCE is designed so as to ignore over-segmentation. The over-segmented regions produced by the NCuts therefore do not affect this error much. This is also well demonstrated by images (g) and (h) of Figure 7. For this image, the NCuts segmentation has a lower GCE value than the volume Watershed, even though the background is over-segmented.

For the two methods producing a small number of regions, the Waterfall algorithm has a lower mean GCE than the NCuts (also visible in Figure 8(e)). Due to the design of the GCE, this could indicate two possibilities. The first is that the segmentations produced are closer to the manual segmentations, as illustrated by segmentations (a) and (b) in Figure 7, where the GCE for the Waterfall segmentation is much smaller than for the NCut with 6 regions, agreeing with a visual evaluation of the segmentations. The second is that the number of regions in the Waterfall segmentation is less than 6, which also often leads to a smaller GCE. This can be seen in segmentations (e) and (f) of Figure 7, where segmentation (e) is visually worse than segmentation (f), but has a lower GCE. The visual judgement for these two segmentations is better represented by the boundary measures.

There are also images for which the NCuts segmentations are better than the Watershed approaches, as can be seen in the bottom row of Figure 7. Here all measures indicate that both NCuts segmentations perform better, which can be confirmed by visual evaluation.

Over all images, for the segmentations into a small number of regions, 60% of the F-measures are larger for the Waterfall level 2 than for the NCuts (6 regions). For the segmentations into 18 regions, 85% of the F-measures are larger for the Volume Watershed than for the NCuts.

6 Conclusion

In this paper we combine the colour and texture boundaries based on learning introduced by Martin et al. [7] with hierarchical Watershed-based segmentation. These boundaries are not directly suitable for Watershed-based algorithms due to gaps in the boundary lines. We have solved the problem by calculating the complement of the quasi-distance function applied to the boundary image.

Two different hierarchical segmentation approaches based on the Watershed have been studied: volume extinction values and Waterfalls. The segmentations obtained compare favorably with NCuts results. We have used the Berkeley Segmentation Dataset for comparison. For evaluation purposes, we have used the area-based method proposed in [4] and a newly introduced boundary-based evaluation method. The proposed method makes use of the distance function between manual and machine contours.

In general, the Watershed approaches produced boundaries matching ground truth segmentations with higher precision. The recall of both the Watershed and NCuts methods is similar. The Waterfall-based approach has the advantage that the number of regions does not need to be specified in advance. It nevertheless has the disadvantage that it tends to produce too many regions at the first level of its hierarchy and too few at the second level [18]. It should be possible to change the region merging criteria to improve on this. There is a version of the NCuts which determines the number of regions automatically [16], but we currently have no implementation of it.

As further work, we intend to investigate other region merging criterion for the Waterfall algorithm, to choose a level between the over-segmentation of level 1 and the under-segmentation of level 2. We plan to compare this with the version of the NCuts which includes criterion on when to stop splitting regions. The calculation time of the boundaries based on learning is unacceptably high. We plan to either accelerate it in some way or find a good approximation with a lower computation time.

7 Acknowledgements

This work was supported by the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04), and the European Union Network of Excellence MUSCLE (FP6-507752). The area based error measure code was written by Adrian Ion and Branislav Mičušík.

References

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, M. I. Jordan, Matching words and pictures, Journal of Machine Learning Research 3 (2003) 1107–1135.

[2] P. Carbonetto, N. de Freitas, K. Barnard, A statistical model for general contextual object recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2004, pp. I:350–362.

[3] Y. Chen, J. Z. Wang, Image categorization by learning and reasoning with regions, Journal of Machine Learning Research 5 (2004) 913–939.

[4] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int'l Conf. Computer Vision, 2001, pp. II: 416–423.

[5] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[6] S. Beucher, F. Meyer, The morphological approach to segmentation: the watershed transformation, in: E. Dougherty (Ed.), Mathematical Morphology in Image Processing, Marcel Dekker, 1993, Ch. 12, pp. 433–481.

[7] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 530–549.

[8] S. Beucher, Numerical residues, Image and Vision Computing 25 (4) (2007) 405–415.

[9] C. Lantuejoul, S. Beucher, On the use of geodesic metric in image analysis, Journal of Microscopy 121 (1981) 39–49.

[10] P. Soille, Morphological Image Analysis, 2nd Edition, Springer, 2002.

[11] C. Vachier, F. Meyer, Extinction values: A new measurement of persistence, in: Proc. of the IEEE Workshop on Non Linear Signal/Image Processing, 1995, pp. 254–257.

[12] F. Meyer, An overview of morphological segmentation, International Journal of Pattern Recognition and Artificial Intelligence 15 (7) (2001) 1089–1118.

[13] S. Beucher, Watershed, hierarchical segmentation and waterfall algorithm, in: Mathematical Morphology and its Applications to Image Processing, Proc. ISMM'94, 1994, pp. 69–76.

[14] B. Marcotegui, S. Beucher, Fast implementation of waterfall based on graphs, in: Mathematical Morphology and its Applications to Image Processing, Proc. ISMM'05, 2005, pp. 177–186.

[15] X. Jiang, C. Marti, C. Irniger, H. Bunke, Distance measures for image segmentation evaluation, EURASIP Journal on Applied Signal Processing 2006 (2006) Article ID 35909, 10 pages.

[16] J. Malik, S. Belongie, T. Leung, J. Shi, Contour and texture analysis for image segmentation, International Journal of Computer Vision 43 (1) (2001) 7–27.

[17] T. Leung, J. Malik, Contour continuity in region-based image segmentation, in: Proc. Euro. Conf. Computer Vision, 1998, pp. 544–559.

[18] A. Hanbury, B. Marcotegui, Waterfall segmentation of complex scenes, in: Proc. of the Asian Conf. on Computer Vision (ACCV), 2006, pp. I:888–897.