Making Image Segmentation Fully Automatic by Case-Based-Reasoning

Maria Frucci [1)], Petra Perner [2)], Gabriella Sanniti di Baja [1)]

1) Institute of Cybernetics "E.Caianiello", CNR, Via Campi Flegrei 34 80078 Pozzuoli, Naples, Italy, {m.frucci, g.sannitidibaja}@cib.na.cnr.it, http://perseo.cib.na.cnr.it/cibcnr/

2) Institute of Computer Vision and Applied Computer Sciences, Körnerstrasse 10, 04107 Leipzig, Germany, pperner@ibai-institut.de, http://www.ibai-research.de/

Abstract: *Image segmentation methods involve a number of parameters whose values have to be tuned depending on image domain. In this communication, a watershed-based segmentation algorithm is considered and Case-Based-Reasoning is used for the automatic selection of the values that, assigned to the parameters, produce a satisfactory segmentation. In this way, the segmentation algorithm can be applied to a wider image domain.*

***Keywords*:** Image segmentation, Watershed transform, Case-Based-Reasoning.

1. INTRODUCTION

One of the most important tasks in image processing is segmentation, which is accomplished in order to distinguish the objects of interest (the foreground) from the rest of the image (the background). A number of different approaches to image segmentation can be found in the literature (see, e.g., [1,2]), among which Watershed-Based-Segmentation, WBS for short, has a key position. In fact, WBS integrates both region-based and edge-detection-based tools, which are often individually considered in other segmentation schemes.

WBS is based on two main phases, respectively aimed at the detection of a suitable set of pixels in the gray-level image, the *seeds*, and at the identification of the regions of influence of the seeds by means of a growing process.

For gray-level images, the seeds are mostly detected as the sets of pixels with locally minimal gray-level in the gradient image. These sets, also called *regional minima*, are found in the regions where the gray-level distribution is mostly uniform and, hence, far from the edges that, in turn, result to be enhanced in the gradient image.

The region of influence of any seed *s* is determined by means of a growing process that, starting from the seed *s*, incorporates in its associated region the pixels that are closer to *s* more than to any other seed, in terms of gray-level homogeneity.

A drawback of WBS standard algorithms is that the image results to be partitioned into a high number of regions, which are not all significant. This phenomenon occurs because the regional minima, even if detected in the gradient image, generally include a number of non-significant seeds. Thus, to be effectively used for image segmentation, WBS algorithms should always include a phase to filter out irrelevant seeds, before computing the watershed partition, or to merge adjacent regions, once the watershed partition is obtained, or to both filter out irrelevant seeds, before the computation of the partition, and merge adjacent regions, once the partition is available.

Whichever way is used to reduce over-segmentation, a number of parameters have to be introduced in the WBS algorithm, whose values depend on image domain. Thus, though in principle WBS can be applied to images belonging to different domains, some fine tuning of the values of the parameters involved by the selected WBS algorithm is necessary to adapt the same algorithm to a different class of images. Tuning can be performed by running the WBS algorithm on various images, belonging to the selected domain, with different values for the parameters. The obtained segmentation results are then analyzed and the values of the parameters producing in the average the best segmentation results can be taken as the resulting tuning for that class of images.

In this work, we show that Case-Based-Reasoning (CBR) can be used for the automatic selection of the values for the segmentation parameters involved in a WBS algorithm. The values expected to produce a good segmentation of an input image are found by comparing the features characterizing the input image with the features characterizing the images already analyzed and recorded as cases in a case-base. In fact, our basic idea is that if two images are characterized by similar features, then both images should be reasonably well segmented by using the same values for the parameters of the WBS algorithm. Thus, CBR can be used to select from the case-base cases having similar image features and to assign to the current image the values of the segmentation parameters associated to the most similar case.

2. CBR FOR SEGMENTATION

A segmentation algorithm needs to be tested on a sufficiently large test data set, which should represent properly the image domain. However, often the test data set is not sufficiently large and, therefore, the parameters involved in the algorithm have to be adjusted to process new data, even if still belonging to the same general domain. Moreover, changes in image quality caused by variations in environmental conditions or by the image devices used in the acquisition phase, also require some adjustment of the values of the parameters.

The above considerations and the fact that CBR can be seen as a method for problem solving as well as a method to capture new experiences, suggest to use Case-Based-Reasoning as a useful tool for image segmentation. The whole CBR system for image segmentation consists of six phases: extracting the case description, indexing, retrieval, learning, adaptation and application of the solution, as shown in Fig. 1. The image features are computed on the whole image and are used for indexing the case-base and for retrieval of the cases close to the current problem, based on a proper similarity measure.

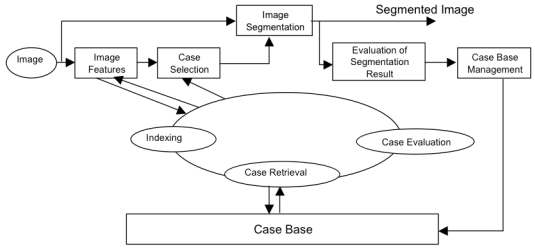

Fig. 1. The CBR image segmentation scheme.

Each case consists of a description of the image features and of the solution, i.e., the values of the parameters that produce, for that case, a satisfactory segmentation. Once the cases close to the current problem have been retrieved, the closest case is selected and the solution associated to it is given as control input to the image segmentation unit. The image segmentation unit takes the current image and processes it according to the current control state. The obtained segmentation result is evaluated, either by an expert or, preferably, automatically. The case-base maintenance unit receives the evaluation of the segmentation result and takes it as a feedback to improve the system performance.

3. WATERSHED BASED SEGMENTATION

The interpretation of a gray-level image as a 3D *landscape*, where the gray-level of a pixel is used as its z-coordinate, can be used to easily explain how the watershed transform is computed. The bottom of each valley is, in the 2D gray-level image, a connected set of pixels with locally minimal gray-level, while the top of each hill is, in the 2D image, a connected set of pixels with locally maximal gray-level. If the landscape is taken under falling rain, the valleys will start to be filled by rain and lakes are created in the catchment basins. Dams have to be built, if we want to avoid that different lakes merge, when the level of rain water reaches the lowest point between any pair of adjacent catchment basins.

The watershed transformation ends once the level of rain has reached the highest hill in the landscape. The top lines of the dams correspond, in the 2D image, to the watershed lines of the transform, which separate the regions into which the image results to be partitioned.

As already pointed out in the Introduction, a drawback of standard WBS algorithms is the large number of regions generated in the partition, when all the regional minima are used as seeds for the growing process. As an example, see Fig. 2, where for the input image (Fig. 2 left) a partition into 3237 regions is obtained (Fig. 2 right), which are, clearly, not all significant.

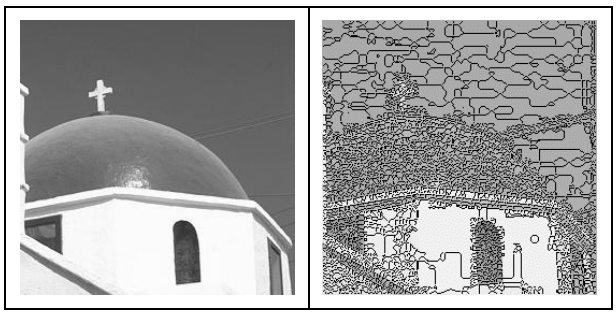

Fig. 2. An image, left, and its watershed partition, right.

To reduce over-segmentation, only seeds corresponding to significant regions should be used. Two techniques, called *flooding* and *digging*, have been suggested in [3] to cause disappearance in the gradient image of those seeds that are recognized as corresponding to non-significant regions. Only significant basins should be preserved in the final watershed partition (i.e., their seeds should all be regarded as relevant) and non-significant basins should be removed by aggregating them to adjacent significant basins (i.e., their seeds should be regarded as irrelevant). The whole process (i.e., flooding, digging and computation of the watershed transform) is iterated until all basins result to be significant.

The definition of significant region is crucial to obtain a meaningful partition. In [3], the relative significance of a basin X with respect to an adjacent basin Y is computed in terms of two measures. Namely: i) the relative depth D_{XY} of X with respect to Y (computed as the difference between the smallest gray-level along the watershed line separating X and Y, and the gray-level of the regional minima in X), and ii) the difference in gray-level G_{XY} between the regional minima of X and Y.

The basin X is considered as relatively significant with respect to Y if $G_{XY}>At$ or $D_{XY}>Dt$, where At and Dt are two threshold values, computed automatically by using statistics on the initial watershed partition of the gray-level image.

Once the relative significance of X has been computed with respect to all the basins adjacent to X, the basin X can be classified in three possible ways: if X is relatively significant with respect to each adjacent region Y, then X is classified as *strongly significant* (the corresponding seed is relevant); if X is not relatively significant with respect to every adjacent region Y, then X is classified as *non-significant* (the corresponding seed is irrelevant and X has to be absorbed by the adjacent regions by means of flooding, i.e., by setting all pixels of X with gray-level lower than the smallest value along the watershed line separating X and Y to such a higher value); if X is relatively significant in correspondence of some adjacent regions only, X is classified as *partially significant* (the seed is irrelevant and X has to be merged with proper regions, selected among those with respect to which X is relatively non-significant. This is obtained by digging a canal to connect the regional minima of X and of each basins Y with respect to which X is relatively non-significant. The gray-level of the pixels in the canal is set to the lower value between those of the regional minima of X and Y). At the end of the process, a noticeably reduction of over-segmentation is obtained.

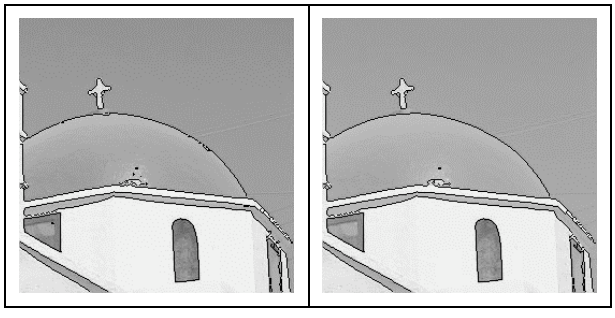

Fig. 3. Reduction of over-segmentation by the algorithm [3], left, and by the new criterion, right.

For the running example, only 82 regions are found by using the algorithm [3], see Fig. 3 left .

To improve the performance of the segmentation algorithm [3], we suggest to change the crisp test based on the OR of the two conditions on G_{XY} and D_{XY}, into a new test, where both G_{XY} and D_{XY} are simultaneously taken into account and different weights can be used for their contributions. Specifically, we define a basin X as relatively significant if:

$$\frac{1}{2}(a \cdot \frac{G_{XY}}{At} + b \cdot \frac{D_{XY}}{Dt}) > T$$

where a and b are suitable weights and T is a threshold. In this way, depending on the image at hand, we can give the same emphasis to both measures by assigning the same value to a and b, or privilege one of them. Moreover, we can reduce over-segmentation for a given selection of the values for a and b, by increasing the value of the threshold T. Of course, while in [3] the two thresholds At and Dt were automatically computed, the main problem with the modified test is the selection of a, b and T, which depends on the image at hand. The three values producing the best segmentation can be found, for a given image, by running the segmentation algorithm several times with a different selection of values for the parameters, and by evaluating the so obtained segmentation results. This process is obviously time consuming if it has to be done for every new image to be segmented. Moreover, this process requires a strong interaction of the user and his ability to judge the various results.

For the running example, we have tried a number of different selections for a, b and T, which are summarized in Table 1.

Table 1. Selection of the values for the three parameters for the running example.

a	*b*	*T*	# regions	evaluation
1.	1.	0.9	45	Under-segmented: the dome is merged to the sky
1.	1.	0.7	62	Under-segmented: part of the church to the left is merged to the sky
1.	1.	0.6	76	A good result with small over-segmentation
0.75	1.25	0.9	47	Under-segmented: the dome is merged to the sky
0.75	1.25	0.8	56	A very good result. The best one.
0.75	1.25	0.7	64	A good result with small over-segmentation
1.5	0.5	1.	29	Extremely under-segmented
1.5	0.5	0.5	75	Over- and under-segmented: the dome is merged to the sky and non meaningful region are detected
1.5	0.5	0.2	104	Over-segmented

From Table 1, we note that the best segmentation result for the running example is obtained by weighting D_{XY} more than G_{XY}. In Fig. 3 right, the segmentation obtained for the running example by selecting for the three parameters the values a=0.75, b=1.25 and T=0.8 that produced the best segmentation into only 56 regions is shown.

To automatically compute the values of the parameters, our idea is to analyze the image features of a number of images to determine whether these images are similar to each other; then, we assume that the values for the weights a and b and the threshold T that, empirically found for one of the similar images gave for it the best segmentation result, can be used for all the remaining similar images to produce always a satisfactory segmentation result. Of course, we are aware that with the same set of values for the three parameters we will not necessarily obtain the best segmentation for each similar image, but we believe that we will obtain an average best fit over the entire set of similar images. We will use CBR to compare images and decide on their similarity, as well as to associate the same values for a, b and T to all similar images, i.e., to those images with almost the same features. A first step in this direction has been done in [4].

4. IMAGE FEATURES

To characterize an image, different features can be used. In this work we use both statistical and texture features. Statistical features are computed in terms of statistical measures of the gray-levels, like mean, variance, skewness, kurtosis, variation coefficient, energy, entropy, and centroid, as suggested in [5]. The statistical features that we adopt for image characterization are shown in Table 2, where the first order histogram $H(g)$ is equal to $N(g)/S$, being g the grey-level, $N(g)$ the number of pixels with grey-level g and S the total number of pixels.

Table 2. Statistical features.

Feature	Feature
Mean: $\bar{g} = \sum_g g \cdot H(g)$	Variance: $\delta_g^2 = \sum_g (g-\bar{g})^2 H(g)$
Skewness: $g_s = \frac{1}{\delta_g^3} \sum_g (g-\bar{g})^3 H(g)$	Kurtosis: $g_k = \frac{1}{\delta_g^4} \sum_g (g-\bar{g})^4 H(g) - 3$
Variation Coefficient: $v = \frac{\delta}{\bar{g}}$	Entropy: $g_E = -\sum_g H(g) \log_2 [H(g)]$
Centroid_x: $\bar{x} = \frac{\sum_x \sum_y x f(x,y)}{\bar{g} S}$	Centroid_y: $\bar{y} = \frac{\sum_x \sum_y y f(x,y)}{\bar{g} S}$

The texture features that we use in this work are computed from the co-occurrence matrix [6] and are shown in Table 3.

Table 3. Texture features.

Feature
Energy: $E = \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} \cdot c_{ij}$
Correlation: $C = \frac{\sum_{i=0}^{n} \sum_{j=0}^{n} (i-u_x) \cdot (j-u_y) \cdot c_{ij}}{s_x \cdot s_x}$
Local homogeneity: $H = \sum_{i=0}^{n} \sum_{j=0}^{n} \frac{1}{1+(i-j) \cdot (i-j)} \cdot c_{ij}$
Contrast: $Con = \sum_{i=0}^{n} \sum_{j=0}^{n} (i-j) \cdot (i-j) \cdot c_{ij}$
with $n = 2 \cdot ldGray - 1$, c_{ij}-Entry of Co-occurrence matrix $u_x = \sum_{i=0}^{n} \sum_{j=0}^{n} i \cdot c_{ij}$, $u_y = \sum_{i=0}^{n} \sum_{j=0}^{n} j \cdot c_{ij}$ $s_x^2 = \sum_{i=0}^{n} \sum_{j=0}^{n} (i-u_x)^2 \cdot c_{ij}$, $s_y^2 = \sum_{i=0}^{n} \sum_{j=0}^{n} (i-u_y)^2 \cdot c_{ij}$

The above statistical and texture features are used to evaluate the similarity of a new input image with respect to the images that have already been examined. To this purpose, we compute the image similarity *SIM* between two images *A* and *B* as the distance between *A* and *B*. The smaller is *SIM* the larger is the similarity. The distance between *A* and *B* is computed as follows:

$$dist_{AB} = \frac{1}{k}\sum_{i=1}^{K} w_i \left| \frac{C_{iA} - C_{i\min}}{C_{i\max} - C_{i\min}} - \frac{C_{iB} - C_{i\min}}{C_{i\max} - C_{i\min}} \right|$$

where C_{iA} and C_{iB} are the values of the *i*-th feature of *A* and *B*, respectively, $C_{i\min}$ and $C_{i\max}$ are the minimum and maximum value respectively of the *i*-th feature of all images that have been already characterized in terms of their features, and w_i is the weight for the *i*-th feature. The weights satisfy the condition $w_1 + w_2 + ... + w_k = 1$. We here assign the same value to all weights.

If the new input image is sufficiently similar to one of the cases already included in the case-base, the new image can be seen as belonging to that case. In this event, the values of the segmentation parameters for the new input image are those associated to the case. If the new input image does not result to be sufficiently similar to any case already included in the case-base, it will belong to a new different case that has to be added to the case-base. In this event, the values for the parameters of the new input image have to be determined by running the segmentation algorithm several times with different values of the parameters, and by judging the obtained segmentation results.

The choice of the features to characterize the images and the selection of the threshold on the distance between images to judge about their similarity are crucial points. The use of global features, like the adopted statistical and texture features, is a weak point in our segmentation system, even if their computation is easy and not computationally expensive. The threshold on the distance to judge on similarity should be set properly. A large threshold value will avoid to misunderstand as similar images that actually should not be considered as such, but will force the creation in the case-base of a huge number of different cases, for each of which the segmentation algorithm has to be applied several times to determine the values for *a*, *b* and *T*.

Another crucial point is the evaluation of the segmentation results when building the cases. When a ground truth is available, the best way to evaluate the segmentation result is to compare the obtained partition with the ground truth. The comparison can be done by using the algorithm introduced in [7]. Otherwise, again some interaction with an expert user is necessary. The user will take into account the number of regions for each partition and will select the values of the parameters that produce the partition with the smallest number of regions, provided that the resulting image is not under-segmented.

5. RESULTS

The case-base that is currently available is still a small one and definitely needs to be extended by analyzing a larger number of images. This notwithstanding, the results we have obtained so far are quite satisfactory. As an example, Fig. 4 shows to the left the segmentation obtained by using the values suggested by the CBR system for *a*, *b* and *T*, and to the right the best segmentation that can be obtained for that image by running the segmentation algorithm several times for different selection of the values of the parameters. The set of values for *a*, *b* and *T* suggested by CBR is a=0.75, b=1.25 and T=0.8, which originate a partition into 454 regions. In turn, the set of values for *a*, *b* and *T* found to provide the best segmentation by running several times the WBS algorithm is a=1.5, b=0.5 and T=0.4, which originate a partition into 382 regions. This latter partition is slightly less over-segmented than the partition obtained by using CBR, but we regard the segmentation guided by CBR as still acceptably good. We believe that by including a larger set of images in the case-base a better result can be obtained with segmentation guided by CBR.

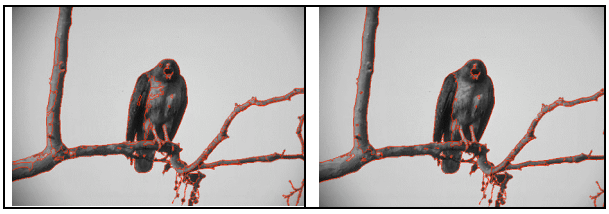

Fig. 4. Segmentation with parameter value selection guided by CBR, left. Best segmentation result obtained by running the segmentation algorithm several time to select the best values of the parameters, right.

In fact, our current case-base includes images belonging to different domains such as biological images, faces, animals and buildings. Unfortunately, some of these images, though appearing to the user as clearly different from each other, are characterized by similar features. As a consequence, they are assigned the values of the parameters associated to the most similar case, but the obtained segmentation results differ from the expected best segmentations. Thus, to extend the validity of our method, further work related to image description is necessary to better discriminate the cases. A possibility is to include other features, e.g., the moments. Alternatively, image similarity could be evaluated between the images with the regional minima used as seeds for the watershed partition, instead of evaluating similarity between the original images. Finally, statistical and texture features could be used in combination with some image description directly based on the images, or by considering also non-image information (such as the position of the camera, the relative movement of the camera, and the object category).

6. CONCLUSION

In this paper, we have presented an approach for watershed-segmentation based on CBR. It is well known that, whatever algorithm is used for computing the watershed partition, the obtained result is likely to be over-segmented. To reduce over-segmentation, significant seeds have to be detected in order only significant regions are obtained in the partition. Seed filtering implies the use of several control criteria, based on features extracted from the initial watershed-partitioned image.

The similarity-based control scheme introduced in the CBR segmentation system gives a flexible way to handle the control criteria, depending on the image features. Currently, the control scheme of our method is global. The same control scheme is applied to the entire image.

We think that a local control scheme could be developed that uses image characteristics of local areas of the image, to control merging in those areas.

An important advantage offered by a CBR segmentation system is that CBR is an incremental knowledge acquisition method as well as a reasoning method. Thus, new situations can be captured in an efficient way and the behavior of the segmentation algorithm can be efficiently studied.

6. REFERENCES

[1] H.D. Cheng, X.H. Jiang, Y. Sun, J. Wang. Color image segmentation: advances and prospects, *Pattern Recognition* 34 (2001). pp. 2259-2281.

[2] J. Freixenet, X. Muñoz, D. Raba, J. Martí, X. Cufí. Yet Another Survey on Image Segmentation: Region and Boundary Information Integration, *Proc. 7th ECCV*, LNCS 2352, Springer (2002) pp. 408-422.

[3] M. Frucci. Oversegmentation Reduction by Flooding Regions and Digging Watershed Lines, *International Journal of Pattern Recognition and Artificial Intelligence* 20 (1) (2006) pp. 15-38.

[4] M. Frucci, P. Perner, G. Sanniti di Baja. Case-based-reasoning for image segmentation, *International Journal of Pattern Recognition and Artificial Intelligence* 22 (5) (2008) pp. 829-842.

[5] H. Dreyer, W. Sauer. *Prozessanalyse*. Verlag Technik. Berlin 1982.

[6] R.M. Haralick, I. Dinstein, K. Shanmugam. Textural features for image classification, *IEEE Transactions on Systems, Man, and Cybernetics* 3 (11) (1973) pp. 610-630.

[7] P. Zamperoni, V. Staroivotov. How dissimilar are two gray-scale images, *Proc. 17th DAGM Symposium*, Springer. Berlin 1995, pp. 448-445.