Model Building in Image Processing by Meta-Learning based on Case-Based Reasoning

Anja Attig and Petra Perner

Institute of Computer Vision and Applied Computer Sciences, IBaI, Leipzig, Germany
www.ibai-institut.de, pperner@ibai-institut.de

Abstract. We propose a framework for model building in image processing by meta-learning based on case-based reasoning. The model-building process is seen as a classification process during which the signal characteristics are mapped to the right image-processing parameters to ensure the best image-processing output. The mapping function is realized by case-based reasoning. Case-based reasoning is especially suitable for this kind of process, since it incrementally allows one to learn the model based on the incoming data stream. To find the right signal/image description of the signal/image characteristics that are in relationship to the signal-processing parameters is one important aspect of this work. In connection with this work intensive studies of the theoretical, structural, and syntactical behavior of the chosen image-processing algorithm have to be done. Based on this analysis we can propose several signal/image descriptions. The selected image description should summarize the cases into groups of similar case and map these to the same processing parameters. Having found groups of similar cases, these should be summarized by prototypes that allow fast retrieval of several groups of cases. This generalization process should permit building up the model over the course of time based on the incrementally obtained data stream. We studied this task for image segmentation based on the Watershed Transformation. First, we studied the theoretical and the implementation aspects of the Watershed Transformation and drew conclusions for suitable image descriptions. Four different descriptions were chosen - statistical and texture features, marginal distributions of columns, rows, and diagonal similarity between the regional minima of two images, and the shape descriptor based on central moments. Our study showed that the weighted statistical and texture features and the shape descriptor based on central moments have yielded the best image description so far for the Watershed Transformation. It can best separate the cases into groups having the same segmentation parameters and it sorts out rotated and rescaled images. Generalization over cases can also be performed over the groups of case. It helps to speed up the retrieval process and to learn incrementally the general model.

1 Introduction

The aim of image processing is to develop methods for automatically extracting from an image or a video the desired information. The developed system should assist a
user in processing or understanding the content of a complex signal, such as an image. Usually, an image consists of thousands of pixels. This information can hardly be quantitatively analyzed by the user. In fact, some problems related to the subjective factor or to the tiredness of the user arise, which may influence reproducibility. Therefore, an automatic procedure for analyzing an image is necessary.

Although in some cases it might make sense to process a single image and to adjust the parameters of the image processing algorithm to this single image manually, mostly the automation of the image analysis makes only sense if the developed methods have to be applied to more than one single image. This is still an open problem in image processing. The parameters involved in the selected processing method have to be adjusted to the specific image. It is often hardly possible to select the parameters for a class of images in such a way that the best result can be ensured for all images of the class. Therefore, methods for parameter learning are required that can assist a system developer in building a model [1] for the image processing task.

While the meta-learning task has been extensively studied for classifier selection, it has not been studied so extensively for parameter learning. Soares et. al [2] studied parameter selection for the identification of the kernel width of a support-vector machine, while Perner [3] studied parameter selection for image segmentation.

The meta-learning problem for parameter selection can be formalized as follows: For a given signal that is characterized by specific signal properties $A$ and domain properties $B$ find the parameters $P$ of the processing algorithm that ensure the best quality of the resulting output signal/information:

$$f : A \cup B \rightarrow P$$

(1)

Meta-data for images may consist of image-related meta-data (gray-level statistics) and non-image related meta-data (sensor, object data) [4]. In general, the processing of meta-data from signals and images should not require too heavy processing and should allow characterizing the properties of the signal that influence the signal processing algorithm.

The mapping function $f$ can be realized by any classification algorithm, but the incremental behavior of case-based reasoning (CBR) fits best to many data/signal processing problems, where the signal-class cannot be characterized ad-hoc, since the data appear incrementally. The right similarity metric that allows mapping data to parameter-groups and, as a consequence, allows obtaining good output results, should be studied more extensively. Performance measures that allow to judge the achieved output and to automatically criticize the systems performance are another important problem [5].

Abstraction of cases to learn domain theory would allow better understanding the behavior of many signal processing algorithms that cannot anymore be described by means of the standard system theory [6].

The aim of our research is to develop methods that allow us to learn a model for the desired task from cases without heavy human interaction (see Fig. 1). The specific emphasis of this work is to develop a methodology for finding the right image description for the case that group similar images in terms of parameters within the same group and map the case to the right parameters in question.
In section 2, we describe the image segmentation based on case-based reasoning. In section 3, we review the theoretical and behavioral aspects of the Watershed Transformation and how to use case-based reasoning to form the image segmentation model. An overview about some of the test images and the corresponding best segmentation parameters is given in section 4, where also the problems concerning the evaluation of the results are briefly addressed. The derived image descriptions from the theoretical and behavioral study in section 3 are given in section 5. A discussion of the result is done in section 6. The process of generalization is described in section 7. Finally, we give conclusions in section 8.

2 CBR-based Image Segmentation

2.1 Case-Based Meta-Control of Image Segmentation

The segmentation problem can be seen as a classification problem for which we have to learn the best classifier. Depending on the segmentation task, the output of the classifier can be the labels for the image regions, the segmentation algorithm selected as the most adequate, or the parameters for the selected segmentation algorithm. In any case, the final result is a segmented image. The learning of the classifier should be done on a sufficiently large test data set, which should represent the entire domain well enough, in order to be able to build up a general model for the segmentation problem. However, often it is not possible to obtain a sufficiently large data set and, therefore, the segmentation model does not fit the entire data set and needs to be adjusted to process new data. We note that a general model does not guarantee the best segmentation for each image; rather, it guarantees an average best fit over the entire set of images.

Another aspect of the problem is related to the changes in image quality caused by variations in environmental conditions, image devices, etc. Thus, the segmentation
performance needs to be adapted to changes in image quality. All this suggests the use of case-based reasoning [7] as a basic methodology for image segmentation, since CBR can be seen as a method for problem solving as well as a method to capture new experience. It can be seen as a learning and knowledge discovery approach. The CBR process consists of six phases: extracting the case description, indexing, retrieval, learning, adaptation and application of the solution.

The CBR process for image segmentation is shown in Fig. 2. The actual image characteristics are described by mean features computed on the whole image. These features are used for indexing the case-base and for retrieval of a set of cases close to the current problem, based on a proper similarity measure.

To adjust the segmentation parameters, indexing should find out images sharing the same segmentation parameters. Among the cases close to the current problem, the closest one is selected and its associated solution 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. Finally, the output is the segmented image.

A case consists of a description of the image characteristics and the solution. The description of the image characteristics can take into account non-image and image information. Based on image description, we can reduce our complex solution space to a subspace of relevant cases, where variation in image quality among the cases is limited.

The solution of a case can be one of the outputs of the classifiers described above. Supposing that our aim is to control the parameters of the segmentation unit, the solution is the set of parameters applicable for the segmentation of the current image. The solution is given as an input to the segmentation unit and the current image is processed by the segmentation unit, based on the selected parameters.

If we want to control the selection of the best algorithm in a set of possible algorithms, then the solution given to the image segmentation unit would be the selected algorithm. If we want to label the regions, then the output would be the labeled regions.

After the image has been processed, the segmentation result is evaluated. This means that the segmentation quality is judged either by an expert or automatically. Depending on the obtained evaluation, the knowledge containers (case description, similarity, solution) are modified to ensure a better segmentation result by processing again the same image. This task is done by the case-base maintenance unit.

The case-base maintenance unit is shown in Fig. 3. Different from a conventional segmentation process, CBR also includes the evaluation of the segmentation result and takes it as a feedback to improve the system performance semi-automatically or automatically. Usually this is an open problem in many segmentation applications.
When the evaluation of the segmentation result is done manually, the expert compares the original image with the labeled image on display. If the expert detects significant differences in the two images, the result is tagged as incorrect and the case-base management will start.

The evaluation procedure can also be done automatically. However, there is no general procedure available and evaluation can be done automatically only in a domain-dependent fashion.

2.2. Case-Based Maintenance and Model Learning

Case-based maintenance is done for several purposes: 1. to enter a new case, when no similar cases are available in the case-base, 2. to update an existing case by case refinement, and 3. to obtain case generalization.

Once an incorrect result is observed for an input image either by the user or by the automatic evaluation procedure, the case is tagged as a bad case. In a successive step, the best segmentation parameters for that image are determined and the attributes, necessary for similarity determination, are computed from the image. Both the segmentation parameters and the attributes calculated from the image are stored into the case-base as a new case. In addition to that, non-image information is extracted from the file header or from any other associated source of information and is stored together with the other information in the case-base. If the case-base is organized hierarchically, the new case has to be stored at the position in the hierarchy suggested by its similarity-relation to the other cases in the case-base.

During storage, case generalization is done to ensure that the case-base does not become too large unnecessarily. Cases that are similar to each other are grouped together in a case class and, for this case-class, a prototype is computed by averaging the values of the attributes. Case generalization ensures that a case is applicable to a wider range of segmentation problems.
It can also happen that several cases are quite similar to each other and, therefore, they get retrieved for a new problem at the same time. Then, learning the similarity by updating the local weights associated to each attribute is advised.

Fig. 3. Case-base maintenance.

3 The Watershed Algorithm: What Influences the Result of the Segmentation?

An easy way to explain the idea of the Watershed Transform is the interpretation of the image as 3D landscape. Therefore we allocate for every pixel its grey value as z-coordinate. Now we flood the landscape from the regional minima, after having bored the local minima and sunk the landscape into water. Lakes are created (basins, catchment basins) in correspondence with the regional minima. We build dams (watershed lines or simply watersheds), where two lakes meet. Alternatively, instead of sinking the landscape, the latter can be flooded by rainfall and the watershed lines will be the lines of attracting the rain that will fall on the landscape. Whichever of the paradigms is used – the immersion or rainfall paradigm - to obtain its simulation two approaches are possible: either the basins are identified, then the watershed lines are obtained by taking a set complement, or the complete image partition is computed, then successively the watersheds are found by detecting boundaries between basins.

Many algorithms were developed for computing the Watershed Transform. For a survey see e.g. [8]. In this work we deal mainly with the Watershed Transformation scheme suggested by Vincent and Soille in [9], and use this scheme also for the new
implementation that we have done for the segmentation algorithm from Frucci et al. [10], [11].

The first definition of the so-called Watershed Transform by immersion was given by Vincent and Soille [9]. Let \( D \) be a digital grey value image with \( h_{\text{min}} \) and \( h_{\text{max}} \) as the minimal and maximal gray value. \( \text{MIN}_h \) is the union of all regional minima with grey value \( h \) with \( h \in [h_{\text{min}}, h_{\text{max}}] \). Furthermore, let \( B \subseteq D \) and suppose that \( B \) is partitioned in \( k \) connected sets \( B_i \) (\( i \in \{1, \ldots, k\} \)). Then, the geodesic influence zone of the set \( B_i \) within \( D \) is computed as:

\[
iZ_D(B_i) = \{ p \in D \mid \forall j \in \{1 \ldots k\} / \{ i \} : d_D(p, B_i) < d_D(p, B_j) \},
\]

where \( d_D(p, B_i) = \min_{q \in B_i} d_D(p, q) \) with \( d_D(p, q) \) as the minimum path among all paths within \( D \) between \( p \) and \( q \).

The union of the geodesic influence zones of the connected components \( B_i \), \( IZ_D(B) \), is computed as follows:

\[
IZ_D(B) = \sum_{i=1}^{k} iZ_D(B_i)
\]

The Watershed Transform by immersion is defined by the following recursion:

\[
\begin{align*}
X_{h_{\text{min}}} & = \{ p \in D \mid f(p) = h_{\text{min}} \} = T_{h_{\text{min}}} \\
X_{h+1} & = \text{MIN}_{h+1} \cup IZ_{T_{h+1}}(X_h), \quad h \in [h_{\text{min}}, h_{\text{max}}]
\end{align*}
\]

where \( T_{h} \), defined as \( T_{h} = \{ p \in D \mid f(p) \leq h \} \), is the set of pixels with grey-level smaller than or equal to \( h \), and \( \text{MIN}_h \) is the union of all the regional minima at level \( h \).

The watershed line of \( D \) is then the complement of \( X_{h_{\text{max}}} \) in \( D \).

The Vincent-Soille algorithm, which does not strictly implement equation (4) (for details see below and [8]), consists of two steps:

1. The pixels of the input image are sorted in increasing order of grey values.
2. A flooding step is done, level after level, starting from level \( h_{\text{min}} \) and terminating at level \( h_{\text{max}} \). For every grey value, breadth-first-search is done to determine the label (label of existing basin, new label or watershed label) to be ascribed to the pixel.

The location of regional minima is important for segmentation by Watershed Transform. To extract the objects of interest in the image, the minima should not lie along the border line of the objects (see Fig. 4 and Fig. 5). On the contrary, regional maxima should be located on the border lines. To correctly identify the minima far from the border lines, in practice the gradient image of the original image is used as an input image for the Watershed Transformation (compare Fig. 4 and Fig. 6).
The Watershed Transform is also dependent on the connectivity. For example, the gradient image in Fig. 4 has three regional minima if 4-connectedness is used, but only one minimum if 8-connectedness is used. In the following, we always use 4-connectedness.

An advantage of the Vincent-Soille algorithm is that it runs in linear running time respectively to the number of pixels of the input image (if the used sorting algorithm is linear). To obtain linear computation time, Vincent and Soille modified the theoretical recursion of expression 4, so as to access each pixel only once. To this aim, at iteration $t$, only pixels labeled $h$ are checked, while expression 4 requires that
pixels labeled less than or equal to h are checked. This modification to the recursion schemes introduces some drawbacks. Specifically, watershed lines thicker than just one pixel can be created and differently labeled basins (see Fig. 7). A pixel that becomes during an iteration t the watershed label and is neighbor of 2, 4, or more odd-numbered pixels with different labels, can get the wrong label during the same iteration after having visited all adjacent basins (see Fig. 8).

![Fig. 7. Comparing the results of the theoretical recursion with Vincent-Soille algorithm under condition 1](image)

Another example of thick watershed lines created by the above modification is shown in Fig. 9.

![Fig. 9. A segmentation result using the Vincent-Soille algorithm (4-connected)](image)

Theoretically, the results of the Watershed Transformation do not depend on the order in which the neighbors of a pixel are visited. Because of the constraints introduced by Vincent-Soille into the implementation of the algorithm, the order of visitation heavily influences the result. For example, if for a pixel \((x, y)\) its neighbors are visited, in the order \(\{(x - 1, y), (x, y + 1), (x + 1, y), (x, y - 1)\}\) (see Fig. 10b), or in the order \(\{(x, y + 1), (x - 1, y), (x, y - 1), (x + 1, y)\}\) (see Fig. 10c), different results will be obtained.
Fig. 10. Cross image and segmentation results using the Vincent-Soille algorithm (4-connected) when different pixel visiting orders are used.

This property also explains why the algorithm is not invariant with respect to rotation (see Fig. 11) and scaling (see Fig. 12).

Fig. 11.: Cross image and segmentation results using the Vincent-Soille algorithm (4-connected) for rotated images.

Fig. 12. A scaled image and the segmentation result using the Vincent-Soille algorithm (4-connected).

The cross in image Fig. 12 has double the width in comparison with the cross in Fig. 10. The light-colored pixel at the intersection of the light-colored and dark-
colored branches of the cross has now only two darker neighbors. In such a condition we obtain the correct result for the Watershed Transformation.

Scaling algorithms generally use interpolations, which can produce more regional minima in the rescaled image than in the original image. As a consequence, a larger number of components can be obtained for the scaled image, which will result in a different segmentation from that of the original image.

The Vincent-Soille algorithm does not generate always connected watershed lines. A reason for that is the constraint demonstrated in Fig. 6. No watershed line separates the two basins A and B, if all the highest grey pixels between the two basins belong to the geodesic influence zone of either A or B. For example, consider Fig. 13 (a), showing an image completely black except for a grey stripe that is four pixels wide. For this image the Vincent-Soille algorithm creates no watershed line between the two basins, because no pixel of the stripe has the same distance to the two basins (compare Fig. 13b and c).

Fig. 13: An image, the geodesic distances from pixel of the stripe to the different basins and the segmentation result using the Vincent-Soille algorithm

To solve the two above drawbacks, Roerdink and Meijster [8] proposed a slightly modified algorithm, which is, however, remarkably more expensive with regard to computation, since its cost is $O(N^2)$, where $N$ is the number of pixels in the image. The original version of Roerdink’s and Meijster’s algorithm is not quite correct. Therefore, Andreà and Haufe give a corrected version in [12].

As we already pointed out, watershed segmentation performs better when using the gradient image of the input image. During our study we tested different edge detectors. The Prewitt (with Chessboard Distance) and the Sobel (with Euclidian Distance) (described e.g. in [13]) edge detectors produced the best results coupled with the use of the new implementation of algorithms [10] with the Vincent-Soille Watershed Transformation. But neither of them was clearly better than the other. All of the following tests were performed with the Prewitt detector, because we obtained the best results with it when working with our test images.

Furthermore, independently of the selected approach, Watershed Transformation tends to highly oversegment due to many regional minima that can possibly be interpreted as noisy minima. To solve this problem different approaches exist, like intensive preprocessing (e.g. [14]), marker controlled watershed (e.g. [15]) or region-merging (e.g. [16], [17]). For preprocessing in order to reduce the number of local minima, smoothing algorithms or extended edge detectors eliminating unnecessary edges, sharpen edges or produce edges with less gaps are adopted. Often a combination of different preprocessing methods is used. A problem of the majority of
the preprocessings methods is the dependence on the result of the particular kind of images. By the marker-controlled watershed a set of regions called markers are used in place of the set of minima. These regions are often manually determined by the user. Therefore this Watershed Transform approach is often qualified for interactive use and less for automation.

To solve the problem of oversegmentation, Frucci [17] combines an iterative computation of the Watershed Transform with processes called digging and flooding. Flooding merges adjacent basins. This is achieved by letting water increase the level, so that it can overflow from one basin into an adjacent one if the level of water is higher than the lowest height along the watershed line separating the two basins. Also digging merges adjacent regions. In this case, to merge a basin A, regarded as non-significant, with a basin B, a canal is dug in the watershed line separating A and B to allow water to flow from B into A. The effect of merging is that the number of local minima found at each iteration diminishes. Flooding and digging are iterated until only significant basins are left.

The algorithm involves expensive computation, but it has the advantage of producing acceptable segmentation results independently of the kind of image. In fact, the approach followed in [17] rather than identifying the watershed lines during the Watershed Transformation, first builds the complete partition of the image into basins and only after that detects the watershed lines by boundary detection. In this way, both drawbacks affecting the Vincent-Soille approach are overcome at the expense of higher computation costs. Therefore, neither the Frucci algorithm presented in [17], nor the extension of the Frucci algorithm introduced in [10], are suitable for real-time processing.

We have used the new implementation of the algorithm [10], [11] for our case-based reasoning studies, since the computation costs are much lower than, due to the use of the Watershed Transform computed by the Vincent and Soille scheme.

Let X and Y be two adjacent basins in the watershed partition of a grey-level image D and W and Z some other basins. As in [17], we denote $LO_{XY}$, the local overflow of X with respect to Y, as the pixel with the minimal grey value along the border line between X and Y. The value of the pixel with the lowest value is the overflow of X, $O_x$. Furthermore, $R_x$ is the grey value of the regional minimum of X, $SA_{XY} = |R_X - R_Y|$ denotes the similarity parameter and $D_{xy} = LO_{XY} - R_x$ defines the relative depth of X at $LO_{XY}$ (see Fig. 14).
In order to determine if a basin $X$ has to be merged with a basin $Y$, Frucci et al. [10] introduce the notion of relative significance of $X$ with respect to $Y$ and perform the following check:

$$\frac{1}{2} \left( a \cdot \frac{SA_{XY}}{At} + b \cdot \frac{D_{ST}}{Dt} \right) \geq T,$$

where $At$ and $Dt$ are the threshold value (for the automatic computation see [17]) $a$, $b$ and $T$ constants, and the left side of 5 is the relative significance of $X$ with respect to $Y$.

If condition 5 is satisfied, the basin $X$ is significant with respect to $Y$. The relative significance $f_X$ is evaluated with respect to all its adjacent basins. Three cases, as shown in Table 1, are possible. When merging is possible, Table 1 also specifies if digging or flooding has to be performed. After flooding or digging have been performed for all basins that are not strongly significant, the Watershed Transformation is executed again with an obviously smaller number of local minima.

### Table 1. Cases for the significance of $X$ by taking into account the relative significance with respect to all its adjacent basins.

<table>
<thead>
<tr>
<th>$X$ is denoted</th>
<th>if</th>
<th>type of merging</th>
</tr>
</thead>
<tbody>
<tr>
<td>strongly significant</td>
<td>$X$ is significant with respect to each adjacent region $Y$.</td>
<td>no merging</td>
</tr>
<tr>
<td>non significant</td>
<td>$X$ is non-significant with respect to each adjacent region $Y$.</td>
<td>flooding</td>
</tr>
<tr>
<td>partially significant</td>
<td>all other cases</td>
<td>digging</td>
</tr>
</tbody>
</table>
**Flooding:** All the pixels of $X$ with a lower grey value than $O_x$ are set to the value $O_x$. If Watershed Transformation is executed again, the basin $X$ results to be merged with any adjacent basin $Y$ for which the value of $LO_{XY}$ is $O_x$.

**Digging:** The basin $X$ is only merged with any adjacent basin $Y$, being non-significant with respect to $X$ and in such a way that $R_x \geq R_y$. Therefore, a canal is dug from the regional minimum of $X$ to the regional minimum of $Y$ passing through $LO_{XY}$. All the pixels of the canal whose grey values are greater than $R_x$ are set to $R_x$.

We have implemented the segmentation algorithm [10] by computing the Watershed Transform according to the scheme of Vincent and Soille, so as to reduce the overall computation cost. However, using such a watershed model creates some problems due to the fact that watershed lines may be missing or thick, which inhibits in some cases the possibility to check adjacent basins and, hence, to apply flooding or digging. For an example of thick watershed lines, see Fig. 15, where $LO_{XY}$ is equal to 12, the basins $X$ and $Y$ are those respectively characterized by the local minima equal to 8, and 6.

If, instead of flooding, Table 1 had suggested to apply digging, we would have obtained mainly the same result for the image in Fig. 15 (b). The only difference would have been that pixels labeled 12 would have been labeled 8. In that case one obtains the parameters $A_t=D_t=8$ if the surrounding image of this cutout shown in Fig. 15 is suitable.

Furthermore, it should be noted that digging over $O_x$ is not always the best way. Other minimal criteria for path digging are explained by Bleau and Leon in [16].

In our study, we are interested in analyzing the image properties in order to detect the proper values for the constants $a$, $b$ and $T$. The constants $a$ and $b$ control the influence of the similarity parameter and the depth. $T$ can be regarded as a threshold. If $T$ is 0 and $a,b \geq 0$, then we obviously get the same segmentation like the one...
produced by the Vincent-Soille algorithm. For $a, b = 2$ and $T = 1$ we often obtain similar results to those achieved by using the algorithm [17].

Since we have used the Vincent-Soille Watershed Transformation, which is not invariant for image rotation and scaling, in our implementation of algorithms [10] and [11], the best value of $a$, $b$ and $T$ can be different for two images after scaling or rotation.

To decide which watershed-based segmentation algorithm to use, we have also considered the algorithm described in [10] in its original implementation. Making a compromise between computation costs and quality of the obtained segmentation results, we have finally opted for the Vincent-Soille algorithm as a basis for the CBR-based watershed algorithm. In future work, we will carry out further tests on the different behavior of the basic watershed algorithms. The hope is that the choice of the basic algorithm can be included as a parameter into a CBR-based watershed algorithm.

### 4 Test Images and Parameters for Watershed Segmentation

For our study we used a data base comprised of different types of images, such as e.g. landscape, medical and biological images, and face images. Nine of these images are shown in Fig. 16a-j and are used for demonstration purposes of the results throughout this chapter. The neurone images {neu1; neu2; neu3; neu4; neu4-r180} seem to belong to the same class, except that neu4_r180 is the 180 degree rotated image of neu4 and neu4 seems to be a rescaled cut-out of one of the image neurons.

![Fig. 16. Images used for the study](image)

The parameters for the Watershed segmentation were obtained by running the CBR-based Watershed segmentation and adjusting the parameters until the result had the best segmentation quality. The resulting parameters are shown in Table 2.
Table 2. Segmentation parameters $a$, $b$, and $T$ of the test images shown in Fig. 16

Table 3 shows the Images having the same segmentation parameters. Based on segmentation parameters the images neu1 and neu3 and neu4 and parrot should be grouped together by the image description.

<table>
<thead>
<tr>
<th>ImageName</th>
<th>$a$</th>
<th>$b$</th>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>monroe</td>
<td>0.5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>gan128</td>
<td>0.75</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>parrot</td>
<td>0.75</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>cellr</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu2</td>
<td>0.25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu4</td>
<td>0.75</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>neu4_r180</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. Images having the same segmentation parameters
**Fig. 17.** Clustering of Test Images into Groups based on the Segmentation Parameters

Fig. 17 and Table 3 demonstrate the clustering of images grouped according to their parameters. Based on this result we found an image description which grouped the images monroe, parrot and neu4 as well as the images neu1 and neu3.

We noted that the image monroe provides also a good result (with only one basin more) with the parameters $a=0.75$, $b=2$ and $T=1$.

Table 4 and Fig. 18 show the four images of the type “neurone”, where only two images have the same parameters. One reason is that image “neu4” contains larger objects than the other neurone images. So image neu4 must not get the same parameters, because the Vincent-Soille algorithm is not invariant with respect to scaling. Image neu2 differs from neu1 and neu3 by more connected objects.

<table>
<thead>
<tr>
<th>ImageName</th>
<th>a</th>
<th>b</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>neu1.bmp</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu2.bmp</td>
<td>0.25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu3.bmp</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>neu4.bmp</td>
<td>0.75</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4. Parameters for the image class neurone Fig. 18a) to Fig. 18d)

We also tested a simple pre-processing method, which reduced well the oversegmentation of many images.

**Fig. 18 a-h.** The different neuron images with their best segmentation, the parameters (a,b,T), and the number of basins BS
By our preprocessing method we eliminated some small edges in the gradient image, by setting all pixels having a grey value no greater than a threshold to zero. The threshold must be quite small (compare Fig. 19 and Fig. 20), eliminating only edges caused by background noise and not too many significant edges in the image.

Fig. 19. Different segmentation results by elimination of smaller edges in the gradient image with threshold $th$

Fig. 20. Different segmentation results by elimination of smaller edges in the gradient image with threshold $th$
Fig. 21. The influence of the selected parameter-set to the segmentation result and the number of basins shown on image gan128

A problem related to the determination of the segmentation parameters for the CBR-based watershed algorithm is how to judge the best segmentation quality. Evaluation done by humans is subjective and can result in different segmented images of the same input image. This is of importance if we only want to separate the parrot (see Fig. 23 a) from the background (Fig. 23 b), or segment parts of the parrot as well (see Fig. 23c), for example the check. The result of the automatic evaluation is shown in Fig. 23d.

Another aspect is that interesting textures in the image may not be visible for the human eye (compare Fig. 21 with the Pseudo Color Images from Fig. 22).

Even the automatic evaluation has some weak points. The automatic evaluation proposed in [10] takes the original images, produces an edge image out of it based on the chosen standard edge operation, and labels the edges based on a simple thresholding algorithm (see Fig. 23d, with Otsu’s thresholding algorithm) and then compares this image with the output image of the Watershed Transformation based on case-based reasoning according to Zamperoni’s similarity measure [18]. For each of these processing steps different standard image processing operations can be used, producing different results for the same image.
Fig. 22. Different pseudo-colored images of gan128

<table>
<thead>
<tr>
<th></th>
<th>(0.75, 2, 1) BS=96</th>
<th>(1.25, 0.75, 1) BS=92</th>
<th>(0.5, 0.75, 1) BS=73</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Image parrot</td>
<td>(b) Manually evaluated by a human_1</td>
<td>(c) Manually evaluated by a human_2</td>
<td>(d) Automatically evaluated segmentation result</td>
</tr>
</tbody>
</table>

Fig. 23. Different manually evaluated and automatically evaluated segmentation results of image parrot

5 Elicitation of Image Descriptions and Assessment of Similarity for Watershed Transform

The aim of the image description is to find out from a set of images the group of images that needs the same processing parameters for achieving the best segmentation results. To give an example, the images neu1, neu3, neu4 and parrot should be grouped together based on the best parameters a, b, and T.

We consider different image descriptions in our study that should allow us to group images based on the image parameters and by doing so learn a model for image segmentation from samples.

Cases are normally composed of

- non-image information
- parameters specifying image characteristics, and
- parameters for solution (image segmentation parameters).

Non-image information is different depending on the application. In our study we use images from different domains like landscape, faces or biology. Our aim is to describe image similarity only by general image properties. Hence our cases are composed of

- parameters specifying image characteristics and
- parameters a, b, T for segmentation.
Images, which are classified as being similar based on the image features, should be segmented with the same parameters of the segmentation algorithm, which computes then the best segmentation for any of them.

To understand these properties we use hierarchical clustering. This gives us a graphical representation of the different image groups. Single linkage is used to show outliers, while the distance between two classes is defined as the minimal distance.

The question is: What are the right image properties that allow us to map the images to the right image segmentation parameters for the Watershed Transform?

The image description should reflect the behavioral approach of the Watershed Transformation and the particular image characteristics, respectively. Therefore we studied the theoretical details and the implementation limits of the Watershed Transformation in section 4 to get insights into this question. We came to the conclusion that the distribution of the regional minima is an important criterion for the behavior of the algorithm.

From this work we decided to test four image descriptions described in the following Sections:
- Statistical Feature Image Description,
- Image Description as Marginal Distribution for Column and Lines,
- Image Description by Similarity between the Regional Minima of two Images, and
- Image Description by Central Moments.

### 5.1 Statistical and Texture Feature Image Description

According to Perner [3], who used this description for meta-learning of the segmentation parameter for a case-based image segmentation model, we applied statistical features like centroid, energy, entropy, kurtosis, mean, skewness, variance and variation coefficient and texture features (energy, correlation, homogeneity, contrast) for case description. The input image is the gradient image of the original image since the Watershed Transformation works on that image. First results on this image description are reported in Frucci et al.[10]. The texture feature has been chosen to describe the particular distribution of the regional minima in an image, while the statistical features describe the signal characteristics.

Like Perner [3], we determined the distance between two images A and B as follows:

\[
\text{dist}_{AB} = \frac{1}{k} \sum_{i=1}^{k} \omega_i \frac{C_{iA} - C_{iB}}{C_{i\text{max}} - C_{i\text{min}}}
\]

where \( k \) is the number of properties in the data base, \( C_{i\text{max}} \) and \( C_{i\text{min}} \) are the maximum and minimum value of all the images for the \( i \)th features in the data base, \( C_{iA} \) is the value of the \( i \)th feature of image \( A \) (analogous for B) and the \( \omega_i \) are weights with
\[ \sum_{i=1}^{k} \omega_i = 1. \] (7)

We use \( \omega_i = 1/k \ \forall i \in \{1, \ldots, k\} \).

Results are reported in Fig. 24.

Fig. 24. Dendrogram for image description based on texture and statistical features

If we virtually cut the dendrogram by the cophenetic similarity of 0.0043, we obtain the groups \( G_1 = \{\text{neu4; neu4_r_180; neu1; neu3}\}, \ G_2 = \{\text{neu2}\}, \ G_3 = \{\text{parrot}\}, \ G_4 = \{\text{gan128}\}, \ G_5 = \{\text{monroe}\}, \text{and} \ G_6 = \{\text{cell}\} \).

The images neu4 and the image neu4_r_180 rotated by 180 degree are grouped into the same group, although they have completely different segmentation parameters. We obtained the best result for neu4 with the parameter-set \( a=0.75, b=2 \) and \( T=1 \), while for neu4_r180 the best segmentation was obtained with the parameter-set \( a=1, b=0.5 \) and \( T=1 \). By using this latter parameter-set for neu4, we would get an
undersegmented result (compare Fig. 26(c)). Overall we observed that not all the images having the same image segmentation parameters are grouped into one group such as neu4 and gan128.

To sort out rotated images from the group including the un-rotated images, we have to give more emphasis to the feature centroid, because this feature is the only one which is not invariant for rotations. Therefore, we divide the image features set into three groups: texture features, centroid, and the remaining statistical features. Each group gets a total weight \( \omega_g \) of 1/3. The local weights \( \omega_{gl} \) in each group are computed as follow

\[
\omega_g = \sum_{i=1}^{l} \omega_{gl} = \sum_{i=1}^{l} \frac{1}{3*1} = \frac{1}{3},
\]

where \( l \) is the number of features in the group.

In the resulting dendrogram the images neu4 and neu4_r180 are clustered in different groups (see Fig. 25).

![Dendrogram Using Single Linkage](image)

**Fig. 25** Dendrogram for image description based on weighted texture, centroid and statistical features

If we virtually cut the dendrogram by a cophenetic similarity value of 0.0057, we obtain the following groups \( G1=\{neu1;neu3\}, G2=\{neu4_r180\}, G3=\{neu2\}, G4=\{neu4\}, G5=\{gan128\}, G6=\{parrot\}, G7=\{cell\}, \) and \( G8=\{monroe\}. \) Except for the images neu4 and parrot, these groups seem to better reflect the relationship between the image description and the parameter-set. The proper weighting of the features can improve the grouping of the images. For this purpose we need to work out a strategy in a further study.
As another way to summarize the regional minima into an image description, we chose to calculate the marginal distribution over x- and y-direction of an image as well as over the diagonal. The marginal distribution is calculated by counting the foreground pixels by column for the y-direction and row by row for the x-direction of an image as well as for the diagonal. As a result, we have histograms over the columns, the rows and the diagonal showing the frequency of foreground pixels in the gradient image. The normalized marginal histogram for x- and y-direction for image neu3 is presented in Fig. 27.

In the next step we compare the histograms for the columns, rows and diagonals between two images by different distance functions. Before we can calculate the marginal distribution for the foreground pixels, we need to binarize the gradient image in order to decide which pixels are foreground or background pixels. We used the thresholding algorithm of Otsu for the automatic binarization. It is clear that this will put another constraint to our approach. However, if it gives us a good image description and an automatic procedure, this is acceptable.

The Kullback-Leibler divergence is usually the measure of choice when comparing two histograms. The Kullback-Leibler divergence is defined as:
where $H_1$ and $H_2$ are two histograms, $n$ the size of the diagrams and $H_1(i)$ the $i$th row in the histogram $H_1$ (analogous for $H_2$). The problem of this measure is that it cannot handle functional values of zero in the histogram. Then the multiplicand in formula (9) is not defined. A functional value of zero means that for a column or a row no foreground pixel exists, which can always happen in a binarized image. Therefore, although this measure is often cited in the literature as the preferred measure when comparing two histograms, for the purpose proposed in this paper we cannot use this measure.

We used different distance functions, like (squared) Euclidean Distance, city block or Chebyshev Distance to compare two histograms.

The dendrograms of the single-linkage method in Figures Fig. 28 to Fig. 31 show the results of the marginal distribution using the Euclidean Distance, where Fig. 28 (and Fig. 29 and Fig. 31, respectively) shows the results comparing the column histogram (and row and diagonal histograms, respectively) and Fig. 30 shows the results comparing the column and row histograms together.
Fig. 28. Dendrogram for image description. Marginal distribution for column using the similarity measure Euclidean Distance

We virtually cut the dendrogram in Fig. 28 by the cophenetic-similarity value equal to 0.455. However, with this cophenetic-similarity value, images for which we got different best segmentation parameters were also clustered in different cases. Therefore the cut-off will be by the cophenetic-similarity value equal to 0.389, because monroe and neu3 have different best segmentation parameters. By this choice of the cophenetic-similarity value all images are in different clusters, including images with the same best segmentation parameters like neu4 and parrot.

By taking into account the row histograms in Fig. 29, the cut-off should be done by the cophenetic-similarity value equal to 0.25, which is able to separate the images neu4 and neu4_r180. However, with this choice again all the images would be assigned to different clusters.

By comparing column and row histograms together in Fig. 30, we should cut the dendrogram by the cophenetic-similarity value equal to 0.39, in order to separate the rotated images. Then only neu4_r180 and neu3 would be clustered together, but these two images have different best segmentation parameters and hence, should not be assigned to the same case. Thus, the final cut-off should be done by the cophenetic-similarity value equal to 0.34. However, this choice would again, lead to having all images in different clusters.
Fig. 30. Dendrogram for image description of marginal distribution of column and row together using the similarity measure of the Euclidean Distance.

Fig. 31. Dendrogram for image description of marginal distribution of the diagonal use of the
similarity measure of the Euclidean Distance

By considering the diagonal histograms in Fig. 31, we should cut the dendrogram by the cophenetic-similarity equal to 0.66 to separate the images neu4 and neu4_r180. With this choice we would get neu4 and parrot, which have the same best segmentation parameters in the same cluster, as desired. However, also monroe, gan128, neu2, neu3 and neu1 would be assigned to that cluster. Since monroe would result as under-segmented as concerns the hair by the best segmentation parameters of gan128 (compare Fig. 32), we have to cut the dendrogram by the cophenetic-similarity equal to 0.275. Then again all images would be in different clusters.

Fig. 32. The influence of the selected parameter-set to the segmentation result shown on image monroe

In conclusion, the marginal distribution using the Euclidean Distance enables us to separate rotated images, but we did not get images having the same settings in the same group. We got similar results for the marginal distribution using the Chebyshev Distance (compare with Fig. 33) and the city block Distance (compare with Fig. 34).
Fig. 33. Dendrogram for image description on marginal distribution using the similarity measure Chebyshev distance.
In conclusion we have to say, that the marginal distribution is a nice method to summarize the image content in x- and y-direction as well as over the diagonal, but it does not help us in properly grouping images for watershed CBR segmentation.

5.3 Image Description by Similarity between the Regional Minima of two Images

Obviously the position of the regional minima has an influence on the segmentation result. Thus, we study the starting-points of the Watershed Transform (regional minima). A simple idea is to determine the similarity between two images A and B (having the same size) as follows:

\[
SIM = \frac{\text{No Coincidence Points}}{\text{Number Minima Pixel A} + \text{Number Minima Pixel B}}
\]

with \(0 \leq Sim \leq 1\). The similarity measure is null zero if the images A and B are equal and one if A and B have no coincidence by the local minima.

![Dendrogram Using Single Linkage](image)
Fig. 35. Dendrogram based on equation 3

The result is shown in Fig. 35. Our test images are all very dissimilar with this measure as the dendrogram in Fig. 35 shows. The lowest cophenetic similarity value is 0.8627 between the gan128 and parrot image. On closer inspection of the regional minima images from the testing images this result is not surprising, because there are not many coincidences (compare Fig. 36).

Fig. 36. Orginal image and the local minimas from the gradient image

If we use the similarity measure from [18] for the evaluation of testing the local minima images we are getting from the similarity matrix, all images excluding the parrot are most similar to the cell. That is the image with the smallest number of local minima in our set of test images. Furthermore, we observe that the images are in general all dissimilar as the dendrogram in Fig. 37 shows. Therefore, we consider this measure as unsuitable to distinguish between the groups of images versus their segmentation parameters.
More promising are probably the determination of the distances from one regional minima to all the surrounding minima up to a predefined area and the calculation of the statistical distribution. Then the problem would be to compare the distribution of two images. We will further evaluate this idea in another study.

5.4 Image Description by Central Moments

Next we study central moments for the shape description. An image can be interpreted as a two-dimensional density function. So we can compute the geometric moments

\[ M_{pq} = \int \int x^p y^q g(x,y) \quad p, q = 0, 1, 2, \ldots \]  

(11)

with continuous image function \( g(x,y) \).

In the case of digital images, we can replace the integrals by sums:

\[ m_{pq} = \sum x^p \sum y^q f(x,y) \quad p, q = 0, 1, 2, \ldots \]  

(12)

where \( f(x,y) \) is the discrete function of the gray levels.

The seven-moment invariants from Hu [19] have the property of being invariant to translation, rotation and scale. Since our implementation of an algorithm [10] is not invariant with respect to rotation and scale, the invariant moments are unsuitable for our image description.

We can consider the central moments, which are the only ones translation invariant:
For our study we use central moments $m_{pq}$ with $p$ and $q$ between 0 and 3 and as input image the binarized gradient image obtained with the thresholding algorithm of Otsu. To determine the similarity between two images we use the normalized city-block distance in equation 6 its the we

$$m_{pq} = \sum_{x} \sum_{y} (x-x_c)^p (y-y_c)^q f(x,y) \quad p, q = 0, 1, 2, \ldots$$

(13)

where

$$x_c = \frac{\sum_{x} \sum_{y} x f(x,y)}{\sum_{x} \sum_{y} f(x,y)} \quad \text{and} \quad y_c = \frac{\sum_{x} \sum_{y} y f(x,y)}{\sum_{x} \sum_{y} f(x,y)}.$$  

(14)

For our study we use central moments $m_{pq}$ with $p$ and $q$ between 0 and 3 and as input image the binarized gradient image obtained with the thresholding algorithm of Otsu. To determine the similarity between two images we use the normalized city-block distance in equation 6 its the we

$$\omega_i = 1/k \quad \forall i \in \{1, \ldots, k\}.$$

Fig. 38. Dendrogram for CBR based on central moments (on binary gradient image)
Fig. 38 shows the dendrogram of this test. If we virtually cut the dendrogram by the cophenetic similarity measure of 0.0129, we obtain the following groups:

- \( G_1 = \{ \text{Monroe, parrot} \} \)
- \( G_2 = \{ \text{neu3, neu4} \} \)
- \( G_3 = \{ \text{gan128} \} \)
- \( G_4 = \{ \text{neu4_r180} \} \)
- \( G_5 = \{ \text{neu2} \} \)
- \( G_6 = \{ \text{neu1} \} \)
- \( G_7 = \{ \text{cell} \} \)

Compared to the statistical and texture features for image description we obtain one more group, but neu4_r180 gets separated from neu4. The resulting groups seem to represent better the relationship between the image characteristic and the image segmentation parameters.

Problematic is only neu3, which appears in the same group as neu4. Fig. 39 shows in (b) the best possible segmentation result and in (c) the segmentation result with the parameter (0.75, 2, 1). Fig. 39(c) is more oversegmented than Fig. 39 (b). To get a better result, we can use a threshold for small edges (compare with section 5 and Fig. 20).

![Fig. 39](image-url)

**Fig. 39.** The influence of the selected parameter-set on the segmentation result and the number of basins shown in image neu3

### 6 Discussion

The image descriptions that work well for the Watershed Transformation are the statistical and texture feature description (STDescript) and the image description based on Central Moments (CMDescript). While the STDescript is only rotation and scale invariant, the STDescript weighted and the CMDescript is not. The obtained groups for the two descriptions are:

**STDescript**

- \( G_1 = \{ \text{neu4, neu4_r180, neu1, neu3} \} \)
- \( G_2 = \{ \text{neu2} \} \)
- \( G_3 = \{ \text{parrot} \} \)
- \( G_4 = \{ \text{gan128} \} \)
- \( G_5 = \{ \text{monroe} \} \)
- \( G_6 = \{ \text{cell} \} \)

**STDescript weighted**

- \( G_1 = \{ \text{neu1, neu3} \} \)
- \( G_2 = \{ \text{neu4_r180} \} \)
- \( G_3 = \{ \text{neu2} \} \)
- \( G_4 = \{ \text{neu4} \} \)
- \( G_5 = \{ \text{gan128} \} \)
- \( G_6 = \{ \text{parrot} \} \)
- \( G_7 = \{ \text{cell} \} \)
- \( G_8 = \{ \text{monroe} \} \)

**CMDescript**

- \( G_1 = \{ \text{Monroe, parrot} \} \)
- \( G_2 = \{ \text{neu3, neu4} \} \)
- \( G_3 = \{ \text{gan128} \} \)
- \( G_4 = \{ \text{neu4_r180} \} \)
- \( G_5 = \{ \text{neu2} \} \)
- \( G_6 = \{ \text{neu1} \} \)
- \( G_7 = \{ \text{cell} \} \)
The obtained groups of the STDescript weighted seem to reflect better the relationship between the image characteristics and the segmentation parameters than the obtained groups by the CMDescript.

The computation time of both image descriptions is more or less the same.

In conclusion, we can say that the statistical and texture feature description weighted and the central moments are the best image description which we have found so far during our study.

7 Case Generalizations and Incremental Model Learning

For each case group we can compute a case-class representative that will be used during case retrieval as matching candidate. The case representative can be the mean over all cases or the median (see for example table 5). If all cases in a case group share the same segmentation parameters, matching will stop after having found the closest case representative among all case representatives. If not all the cases in a case group share the same segmentation parameters, matching will proceed until the closest case of all the cases in a group of cases is found (see Figure 40 for example of the case base based on the image description of STDescript weighted. Over the course of time new cases can be learnt during the application of the Watershed Transformation based on CBR. The matching procedure is done based on the similarity measure described in Section 5.1.

<table>
<thead>
<tr>
<th>ImageName</th>
<th>Mean</th>
<th>StdDev</th>
<th>Kurtosis</th>
<th>VarCoef</th>
<th>…</th>
<th>Energy</th>
<th>Contrast</th>
<th>LocHomog</th>
<th>xCentroid</th>
<th>yCentroid</th>
</tr>
</thead>
<tbody>
<tr>
<td>neu1r.bmp</td>
<td>16.55</td>
<td>23.58</td>
<td>4.48</td>
<td>1.43</td>
<td>…</td>
<td>0.00748</td>
<td>267.45</td>
<td>268.45</td>
<td>56.98</td>
<td>75.67</td>
</tr>
<tr>
<td>neu3r.bmp</td>
<td>15.06</td>
<td>22.43</td>
<td>6.59</td>
<td>1.49</td>
<td>…</td>
<td>0.00558</td>
<td>268.04</td>
<td>269.04</td>
<td>58.55</td>
<td>71.86</td>
</tr>
<tr>
<td>Prototype Σ</td>
<td>15.80</td>
<td>23.01</td>
<td>5.54</td>
<td>1.46</td>
<td>…</td>
<td>0.00653</td>
<td>267.75</td>
<td>268.75</td>
<td>57.77</td>
<td>73.76</td>
</tr>
</tbody>
</table>

Table 5. Case group with its case description and prototype

![Fig. 40. Retrieval of the case base](image-url)
8 Conclusion

We have proposed a framework for model building in image processing by meta-learning based on case-based reasoning. The model-building process is seen as a classification process during which the signal characteristics are mapped to the right image-processing parameters to ensure the best image-processing output. The mapping function is realized by case-based reasoning. Case-based reasoning is especially suitable for this kind of process, since it incrementally allows learning the model based on the incoming data stream. To find the right signal/image description of the signal/image characteristics that are in relationship to the signal-processing parameters is one important aspect of this work. In connection with this work one has to do an intensive study of the theoretical, structural, and syntactical behavior [20] of the chosen image processing algorithm. Based on this analysis we can propose several signal/image descriptions. The selected image description should summarize the cases into groups of similar cases and map these similar cases to the same processing parameters. Having found groups of similar cases, these should be summarized by prototypes that allow fast retrieval over several groups of cases. This generalization process should allow building up the model over time based on the incrementally obtained data stream.

Our theoretical study of different algorithms of the Watershed Transformation showed that the WT produces different results if the image is rotated or rescaled. The particular implementation of the algorithm puts constraints on the behavior of the algorithm. As a result of our study we conclude that we need an image description that describes the distribution of the regional minima and that is not invariant against rotation and scaling.

We studied four different image descriptions: statistical and texture feature description, image description as a marginal description between columns, rows, and diagonals, similarity of regional minima in two images, and central moments.

Two image descriptions of the four image descriptions did not lead to any success. The description based on statistical and texture features is useful but is invariant against rotation and scaling. The best image descriptions are the description based on statistical and texture features weighted and the description based on central moments. These image descriptions seem to well represent the relationship between the image characteristics of the particular image and the segmentation parameters. Cases having the same segmentation parameters could be grouped based on the image descriptions into the same group. This will make possible a generalization on these groups of cases that will lead to a complete image-segmentation model. However, to automatically choose the right weight values for the statistical and texture feature description is up to now an open question.

References


