

# Still Image Segmentation Tools for Object-based Multimedia Applications

Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis.

*Abstract*— In this paper, a color image segmentation algorithm and an approach to large-format image segmentation are presented, both focused on breaking down images to semantic objects for object-based multimedia applications. The proposed color image segmentation algorithm performs the segmentation in the combined intensity–texture–position feature space in order to produce connected regions that correspond to the real-life objects shown in the image. A preprocessing stage of conditional image filtering and a modified K-Means-with-connectivity-constraint pixel classification algorithm are used to allow for seamless integration of the different pixel features. Unsupervised operation of the segmentation algorithm is enabled by means of an initial clustering procedure. The large-format image segmentation scheme employs the aforementioned segmentation algorithm, providing an elegant framework for the fast segmentation of relatively large images. In this framework, the segmentation algorithm is applied to reduced versions of the original images, in order to speed-up the completion of the segmentation, resulting in a coarse-grained segmentation mask. The final fine-grained segmentation mask is produced with partial reclassification of the pixels of the original image to the already formed regions, using a Bayes classifier. As shown by experimental evaluation, this novel scheme provides fast segmentation with high perceptual segmentation quality.

**Keywords** : *image segmentation; image analysis; large-format image segmentation; Bayes classifier*

## I. INTRODUCTION

In recent years, the proliferation of digital media has established the need for the development of tools for the efficient representation, access and retrieval of visual information. These tools, targeted at applications of large image and video collections as well as the Web, become increasingly important as the amount of digital media both on the Web and in proprietary collections increases. While several approaches have been proposed to address these issues, most recent approaches rely on the analysis of the content of the medium in semantic objects. This is true both for still image manipulation (image indexing [2], [4], [7], [19], region-of-interest coding using the JPEG2000 standard [35]) and for video representation [23], coding [14], [44] and indexing using the MPEG-4 [38] and recently in-

troduced MPEG-7 [29] standards. In still image indexing, for example, state of the art systems are based on the segmentation of images into regions corresponding to objects and the use of a separate set of indexing features for each object [2], [7].

The cornerstone of any such object-based multimedia application is the segmentation algorithm, which for every still image or every frame of an image sequence produces a corresponding segmentation mask: a gray-scale image in which different gray levels denote different regions identified by the algorithm. The segmentation mask is then used, depending on the specific application, for extracting region-specific indexing features or for identifying regions of interest. The present work concentrates on addressing the issue of effective segmentation of still color images, aiming to applications requiring the automatic segmentation of heterogeneous images, thus excluding the availability of a priori knowledge about the objects contained in each image. Content-based indexing and retrieval of images is a typical application of this category.

Segmentation methods for 2D images may be divided primarily into region-based and boundary-based methods [39], [40], [10], [15]. Region-based approaches [13], [22], [21] rely on the homogeneity of spatially localized features such as intensity. The K-means algorithm [26] and evolved variants of it (KMCC [22]) have been used as the basis of several region-based approaches. Region-growing and split and merge techniques also belong to the same category. On the other hand, boundary-based methods use primarily gradient information to locate object boundaries. Deformable whole boundary methods [17], [41], rely on the values of gradients in parts of an image near an object boundary. Hybrid techniques which integrate the results of boundary detection and region growing have also been proposed [9].

Other techniques include the segmentation using the Expectation-Maximization (EM) algorithm [2], [4] and the segmentation by anisotropic diffusion [31], [1]. The EM algorithm is used for finding maximum likelihood estimates when there is missing or incomplete data; the cluster membership for each pixel can be seen as such. Anisotropic diffusion can be seen as a robust procedure which estimates a piecewise smooth image from a noisy input image. The “edge-stopping” function in the anisotropic diffusion equation, allows the preservation of edges while diffusing the rest of the image. The Recursive Shortest Spanning Tree (RSST) algorithm [28], [43], starting from a very fine partitioning of the image, performs merging of neighboring nodes while considering the minimum of a cost function. Mathematical morphology [25], [37], [34] methods,

V. Mezaris and M.G. Strintzis are with the Information Processing Laboratory, Electrical and Computer Engineering Dept., Aristotle University of Thessaloniki, Thessaloniki 54124, Greece, and with the Informatics and Telematics Institute, Centre for Research and Technology Hellas, 1st Km Thermi-Panorama road, Thessaloniki 57001, Greece. I. Kompatsiaris is with the Informatics and Telematics Institute, Centre for Research and Technology Hellas, 1st Km Thermi-Panorama road, Thessaloniki 57001, Greece. This work was supported by the EU projects aceMedia “Integrating knowledge, semantics and content for user centred intelligent media services” (FP6-001765) and SCHEMA “Network of Excellence in Content-Based Semantic Scene Analysis and Information Retrieval” (IST-2001-32795). The assistance of COST211 quat is also gratefully acknowledged.

including in particular the watershed transformation [47], [12], have also received considerable attention for use in image segmentation. The watershed transformation determines the minima of the gradients of the image to be segmented, and associates a segment to each minimum. Conventional gradient operators generally produce many local minima which are caused by noise or quantization errors, and hence, the watershed transformation with a conventional gradient operator usually results in oversegmentation. To alleviate this problem, the use of multiscale morphological gradient operators has been proposed [48].

In this paper, a region-based approach is adopted. An unsupervised segmentation algorithm is presented using a combination of conditional image filtering by a moving average filter and pixel classification by means of a novel variant of the K-Means-with-connectivity-constraint algorithm (KMCC), in order to form connected regions that correspond to the objects contained in the image. Performing the segmentation in the combined intensity–texture–position feature space allows for effective handling of textured objects, as opposed to most previous algorithms, including those based on K-means-family pixel classifiers, which do not utilize texture information.

Although this segmentation algorithm is quite fast when applied to images of relatively small dimensions, its efficiency degrades quickly as the dimensions of the image increase. This is in fact the case for any segmentation algorithm with computational complexity proportional to the number of pixels of the image to be segmented. Since large-format images are becoming increasingly popular, partly as a result of recent advances in storage and communication technologies, time-efficient methods for their segmentation become essential. For this reason, a novel framework for the fast segmentation of relatively large images employing a Bayes classifier is proposed. This effectively addresses the issues of time efficiency and perceptual segmentation quality and, as will be seen, can also be combined with most segmentation algorithms found in the literature.

The paper is organized as follows: The proposed segmentation algorithm is presented in Sec. II. In Sec. III, the framework for the fast segmentation of large-format images using the segmentation algorithm presented in the previous section is developed, and the issues of time efficiency and perceptual segmentation quality are discussed. Section IV contains experimental evaluation and comparisons of the developed methods, and finally, conclusions are drawn in Sec. V.

## II. COLOR IMAGE SEGMENTATION

### A. Segmentation System Overview

The segmentation system described in this section is based on a novel variant of the K-Means-with-connectivity-constraint algorithm (KMCC) [21], [22], a member of the popular K-Means family. The KMCC algorithm is an algorithm that classifies the pixels into regions taking into account not only the intensity information associated with each pixel but also the position of the pixel, thus producing

connected regions rather than sets of chromatically similar pixels. The novel variant presented in this paper introduces the use of texture features in combination with the intensity and position features; this, along with the texture-dependent filtering of pixel intensities (conditional filtering), endow the segmentation algorithm with the capability to handle textured objects effectively, by forming large, chromatically non-uniform regions instead of breaking down the objects to a large number of chromatically uniform regions, as was the case in a previous preliminary version of the algorithm [22], [3]. In addition, in the proposed algorithm the required initial values are estimated using a novel initial clustering procedure, based on breaking down the image to square blocks and assigning an intensity feature vector and a texture feature vector to each block. This automated initial clustering procedure makes any user intervention at this stage unnecessary, thus facilitating the processing of large image collections.

The overall segmentation algorithm consists of the following stages:

- Stage 1. Extraction of the intensity and texture feature vectors corresponding to each pixel. These will be used along with the spatial features in the following stages.
- Stage 2. Estimation of the initial number of regions and their spatial, intensity and texture centers, using a novel initial clustering procedure. These values are to be used by the KMCC algorithm.
- Stage 3. Conditional filtering using a moving average filter.
- Stage 4. Final classification of the pixels, using the KMCC algorithm.

The result of the application of the segmentation algorithm to a color image is the segmentation mask, i.e. a grayscale image in which different gray values correspond to different regions formed by the KMCC algorithm.

### B. Color and Texture Features

For every pixel  $\mathbf{p} = [p_x \ p_y]$ ,  $p_x = 1, \dots, x_{max}$ ,  $p_y = 1, \dots, y_{max}$ , where  $x_{max}$ ,  $y_{max}$  are the image dimensions in pixels, a color feature vector and a texture feature vector are calculated. The color features used are the three intensity coordinates of the CIE  $L^*a^*b^*$  color space. What makes CIE  $L^*a^*b^*$  more suitable for the proposed algorithm than the widely used RGB color space is perceptual uniformity: the CIE  $L^*a^*b^*$  is approximately perceptually uniform, i.e. the numerical distance in this color space is approximately proportional to the perceived color difference [24]. The color feature vector of pixel  $\mathbf{p}$ ,  $\mathbf{I}(\mathbf{p})$  is defined as

$$\mathbf{I}(\mathbf{p}) = [I_L(\mathbf{p}) \ I_a(\mathbf{p}) \ I_b(\mathbf{p})] \quad (1)$$

In order to detect and characterize texture properties in the neighborhood of each pixel, the Discrete Wavelet Frames (DWF) decomposition, proposed by [45], is used. This is a method similar to the Discrete Wavelet Transform (DWT), that uses a filter bank to decompose each intensity component of the image to a set of subbands (Fig. 1). The main difference between the two methods is that in the DWF decomposition the output of the filter bank is

not subsampled. The DWF approach has been proven to decrease the variability of the estimated texture features, thus improving classification performance [45].

The filter bank used is based on the lowpass Haar filter  $H(z) = \frac{1}{2}(1 + z^{-1})$ , which satisfies the lowpass condition  $H(z)|_{z=1} = 1$ . The complementary highpass filter  $G(z)$  is defined with respect to the lowpass  $H(z)$  as  $G(z) = zH(-z^{-1})$ . The filters of the filter bank,  $H_V(z), G_i(z), i = 1, \dots, V$  are generated by the prototypes  $H(z), G(z)$ , according to equations presented in [45]. Despite its simplicity, the above filter bank has been demonstrated to perform surprisingly well for texture segmentation in [45] and is, for this reason, a good choice for our system. The use of such simple filters has the additional advantage of correspondingly reduced computational complexity. The texture of pixel  $\mathbf{p}$  is then characterized by the standard deviations of all detail components, calculated in a square neighborhood  $\Phi$  of pixel  $\mathbf{p}$ .

For images of relatively small dimensions, e.g.  $150 \times 100$  pixels, a two-dimensional DWF decomposition of two levels has been chosen, thus  $V = 2$ . Since three detail components are produced for each level of decomposition and each one of the three intensity components, this results in a  $9 \times V = 18$ -component texture feature vector  $\mathbf{T}(\mathbf{p})$ :

$$\mathbf{T}(\mathbf{p}) = [\sigma_1(\mathbf{p}) \ \sigma_2(\mathbf{p}) \ \dots \ \sigma_{9 \times V}(\mathbf{p})] \quad (2)$$

Moving towards larger images, or large-format versions of the same image, any given texture becomes coarser-grained, in terms of the size of its basic structural element, calculated in pixels. Thus, for images of significantly larger dimensions, more levels of decomposition may be required to effectively characterize texture. In the experiments where the segmentation algorithm of this section was applied directly to large-format images, four levels of decomposition were used instead of two, resulting in a 36-component texture feature vector.

### C. Initial Clustering

Similarly to any other variant of the K-Means algorithm, the KMCC algorithm requires initial values: an initial estimation of the number of regions in the image and their spatial, intensity and texture centers (all these initial values can and are expected to be altered during the execution of the algorithm). In order to compute them, the image is broken down to square, non-overlapping blocks of dimension  $f \times f$ . In this way, a reduced image composed of a total of  $L$  blocks,  $b_l, l = 1, \dots, L$ , is created. A color feature vector  $\mathbf{I}^b(b_l) = [I_L^b(b_l) \ I_a^b(b_l) \ I_b^b(b_l)]$  and a texture feature vector  $\mathbf{T}^b(b_l)$  are then assigned to each block, as follows:

$$\mathbf{I}^b(b_l) = \frac{1}{f^2} \sum_{\mathbf{p} \in b_l} \mathbf{I}(\mathbf{p}), \quad (3)$$

$$\mathbf{T}^b(b_l) = \frac{1}{f^2} \sum_{\mathbf{p} \in b_l} \mathbf{T}(\mathbf{p}), \quad (4)$$

The distance between two blocks is defined as follows:

$$D^b(b_l, b_n) = \|\mathbf{I}^b(b_l) - \mathbf{I}^b(b_n)\| + \lambda_1 \|\mathbf{T}^b(b_l) - \mathbf{T}^b(b_n)\|, \quad (5)$$

where  $\|\mathbf{I}^b(b_l) - \mathbf{I}^b(b_n)\|, \|\mathbf{T}^b(b_l) - \mathbf{T}^b(b_n)\|$  are the Euclidean distances between the block feature vectors. In our experiments,  $\lambda_1 = 1$ , since experimentation showed that using a different weight  $\lambda_1$  for the texture difference would result in erroneous segmentation of textured images if  $\lambda_1 \ll 1$ , respectively non-textured images if  $\lambda_1 \gg 1$ . As shown in the experimental results section, the value  $\lambda_1 = 1$  is appropriate for a variety of textured and non-textured images; small deviations from this value have little effect on the segmentation results.

The number of regions of the image is initially estimated by applying a variant of the maximin algorithm to this set of blocks. This algorithm consists of the following steps:

- Step 1. The block in the upper left corner of the image is chosen to be the first intensity and texture center.
- Step 2. For each block  $b_l, l = 1, \dots, L$ , the distance between  $b_l$  and the first center is calculated; the block for which the distance is maximized is chosen to be the second intensity and texture center. The distance  $C$  between the first two centers is indicative of the intensity and texture contrast of the particular image.
- Step 3. For each block  $b_l$ , the distances between  $b_l$  and all centers are calculated and the minimum of those distances is assigned to block  $b_l$ . The block that was assigned the maximum of the distances assigned to blocks is a new candidate center.
- Step 4. If the distance that was assigned to the candidate center is greater than  $\gamma \cdot C$ , where  $\gamma$  is a predefined parameter ( $\gamma \in [0, 1]$ ;  $\gamma = 0$  results to all non-identical blocks being identified as region centers, while  $\gamma = 1$  restricts region centers to the two already identified in step 2), the candidate center is accepted as a new center and step 3 is repeated; otherwise, the candidate center is rejected and the maximin algorithm is terminated.

In the experimental results section, the values of parameters  $L, \gamma$  that were used throughout the experiments and the effect of using values deviating from the employed ones are presented.

The number of centers estimated by the maximin algorithm constitutes an estimate of the number of regions in the image. Nevertheless, it is not possible to directly determine whether these regions are connected or not. Furthermore, there is no information regarding their spatial centers. In order to solve these problems, a simple K-Means algorithm is applied to the set of blocks, using the information produced by the maximin algorithm for its initialization. When the K-Means algorithm converges, the connectivity of the regions that were formed is evaluated; those that are not connected are easily broken down to the minimum number of connected regions using a recursive four-connectivity component labelling algorithm [16], so that a total of  $K'$  connected regions  $s_k, k = 1, \dots, K'$  are identified. Their intensity, texture and spatial centers,  $\mathbf{I}^s(s_k) = [I_L^s(s_k) \ I_a^s(s_k) \ I_b^s(s_k)]$ ,  $\mathbf{T}^s(s_k) = [T_1^s(s_k) \ \dots \ T_{9 \times V}^s(s_k)]$  and  $\mathbf{S}(s_k) = [S_x(s_k) \ S_y(s_k)]$ ,  $k = 1, \dots, K'$ , must now be calculated. Let  $M_k$  be the number of pixels belonging to region  $s_k$ :  $s_k = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{M_k}\}$ ; the region centers to be used

for the initialization of the KMCC are calculated as follows:

$$\mathbf{I}^s(s_k) = \frac{1}{M_k} \sum_{\mathbf{p} \in s_k} \mathbf{I}(\mathbf{p}), \quad (6)$$

$$\mathbf{T}^s(s_k) = \frac{1}{M_k} \sum_{\mathbf{p} \in s_k} \mathbf{T}(\mathbf{p}), \quad (7)$$

$$\mathbf{S}(s_k) = \frac{1}{M_k} \sum_{\mathbf{p} \in s_k} \mathbf{p}, \quad (8)$$

#### D. Conditional Filtering

Images may contain parts in which intensity fluctuations are particularly pronounced, even when all pixels in these parts of the image belong to a single object (Fig. 2a). In order to facilitate the grouping of all these pixels in a single region based on their texture similarity, it is useful to reduce their intensity differences. This is achieved by applying a moving average filter to the appropriate parts of the image, thus altering the intensity information of the corresponding pixels.

The decision of whether the filter should be applied to a particular pixel  $\mathbf{p}$  or not is made by evaluating the norm of the texture feature vector  $\mathbf{T}(\mathbf{p})$  (Sec. II-B); the filter is not applied if that norm is below a threshold  $\tau$ . The output of the conditional filtering module can thus be expressed as:

$$\mathbf{J}(\mathbf{p}) = \begin{cases} \mathbf{I}(\mathbf{p}) & \text{if } \|\mathbf{T}(\mathbf{p})\| < \tau \\ \frac{1}{f^2} \sum \mathbf{I}(\mathbf{p}) & \text{if } \|\mathbf{T}(\mathbf{p})\| \geq \tau \end{cases} \quad (9)$$

Correspondingly, region intensity centers calculated similarly to Eq. (6) using the filtered intensities  $\mathbf{J}(\mathbf{p})$  instead of  $\mathbf{I}(\mathbf{p})$  are symbolized  $\mathbf{J}^s(s_k)$ .

An appropriate value of threshold  $\tau$  was experimentally found to be

$$\tau = \max\{0.65 \cdot T_{max}, 14\} \quad (10)$$

where  $T_{max}$  is the maximum value of the norm  $\|\mathbf{T}(\mathbf{p})\|$  in the image. The term  $0.65 \cdot T_{max}$  in the threshold definition serves to prevent the filter from being applied outside the borders of textured objects, so that their boundaries are not corrupted. The constant bound 14, on the other hand, is used to prevent the filtering of images composed of chromatically uniform objects; in such images, the value of  $T_{max}$  is expected to be relatively small and would correspond to pixels on edges between objects, where filtering is obviously undesirable.

The output of the conditional filtering stage (e.g. Fig. 2b) is used as input by the KMCC algorithm.

#### E. The K-Means with Connectivity Constraint Algorithm

Clustering based on the K-Means algorithm, originally proposed by McQueen [26], is a widely used region segmentation method [36], [33], [20] which, however, tends to produce unconnected regions. This is due to the propensity of the classical K-Means algorithm to ignore spatial information about the intensity values in an image, since it only takes into account the global intensity or color information. Furthermore, previous pixel classification algorithms of the

K-Means family do not take into account texture information. In order to alleviate these problems, we propose the use of a novel variant of the KMCC algorithm. In this algorithm the *spatial proximity* of each region is also taken into account by defining a new center for the K-Means algorithm and by integrating the K-Means with a component labeling procedure. In addition to that, texture features are combined with the intensity and position information to permit efficient handling of textured objects.

The KMCC algorithm applied to the pixels of the image consists of the following steps:

- Step 1. The region number and the region centers are initialized, using the output of the initial clustering procedure described in Sec. II-C.
- Step 2. For every pixel  $\mathbf{p}$ , the distance between  $\mathbf{p}$  and all region centers is calculated. The pixel is then assigned to the region for which the distance is minimized. A generalized distance of a pixel  $\mathbf{p}$  from a region  $s_k$  is defined as follows:

$$D(\mathbf{p}, s_k) = \|\mathbf{J}(\mathbf{p}) - \mathbf{J}^s(s_k)\| + \lambda_1 \|\mathbf{T}(\mathbf{p}) - \mathbf{T}^s(s_k)\| + \lambda_2 \frac{\bar{M}}{M_k} \|\mathbf{p} - \mathbf{S}(s_k)\|, \quad (11)$$

where  $\|\mathbf{J}(\mathbf{p}) - \mathbf{J}^s(s_k)\|$ ,  $\|\mathbf{T}(\mathbf{p}) - \mathbf{T}^s(s_k)\|$  and  $\|\mathbf{p} - \mathbf{S}(s_k)\|$  are the Euclidean distances between the pixel feature vectors and the corresponding region centers; pixel number  $M_k$  of region  $s_k$  is a measure of the area of region  $s_k$ , and  $\bar{M}$  is the average area of all regions,  $\bar{M} = \frac{1}{K} \sum_{k=1}^K M_k$ . The regularization parameter  $\lambda_2$  is defined as  $\lambda_2 = 0.4 \cdot \frac{C}{\sqrt{x_{max}^2 + y_{max}^2}}$ , while the choice of the parameter  $\lambda_1$  has been discussed in Sec. II-C.

In (11), normalization of the spatial distance,  $\|\mathbf{p} - \mathbf{S}(s_k)\|$  by division by the area of each region  $\frac{M_k}{\bar{M}}$ , is necessary in order to encourage the creation of large connected regions; otherwise, pixels would tend to be assigned to smaller rather than larger regions due to greater spatial proximity to their centers. In this case, large objects would be broken down to more than one neighboring smaller regions instead of forming one single, larger region. The regularization parameter  $\lambda_2$  is used to ensure that a pixel is assigned to a region primarily due to their similarity in intensity and texture characteristics, even in low-contrast images, where intensity and texture differences are small compared to spatial distances.

- Step 3. The connectivity of the formed regions is evaluated; those which are not connected are easily broken down to the minimum number of connected regions using a recursive four-connectivity component labelling algorithm [16].
- Step 4. Region centers are recalculated (Eq. (6)-(8)). Regions with areas below a size threshold  $\xi$  are dropped. In our experiments, the threshold  $\xi$  was equal to 0.5% of the total image area. This is lower than the minimum accepted region size  $\psi$ , which in our experiments was equal to 0.75% of the total image area. The latter is used to ensure that no particularly small, meaningless regions are formed. Here, the slightly lower threshold  $\xi$  is used to avoid dropping, in one iteration of the KMCC algorithm, regions that are close to threshold  $\psi$  and are likely to exceed it in future

iterations. The number of regions  $K$  is also recalculated, taking into account only the remaining regions.

- Step 5. Two regions are merged if they are neighbors and if their intensity and texture distance is not greater than an appropriate merging threshold:

$$\| \mathbf{J}^s(s_{k_1}) - \mathbf{J}^s(s_{k_2}) \| + \lambda_1 \| \mathbf{T}^s(s_{k_1}) - \mathbf{T}^s(s_{k_2}) \| \leq \mu \quad (12)$$

Threshold  $\mu$  is image-specific, defined in our experiments by

$$\mu = \begin{cases} 7.5 & \text{if } C < 25 \\ 15 & \text{if } C > 75 \\ 10 & \text{otherwise} \end{cases} \quad (13)$$

where  $C$  is an approximation of the intensity and texture contrast of the particular image, as defined in Sec. II-C

- Step 6. Region number  $K$  and region centers are reevaluated.

- Step 7. If the region number  $K$  is equal to the one calculated in *Step 6* of the previous iteration and the difference between the new centers and those in *Step 6* of the previous iteration is below the corresponding threshold for all centers, then stop, else goto *Step 2*. If index "old" characterizes the region number and region centers calculated in *Step 6* of the previous iteration, the convergence condition can be expressed as  $K = K^{old}$  and

$$\begin{aligned} \| \mathbf{J}^s(s_k) - \mathbf{J}^s(s_k^{old}) \| &\leq c_I, \\ \| \mathbf{T}^s(s_k) - \mathbf{T}^s(s_k^{old}) \| &\leq c_T, \\ \| \mathbf{S}(s_k) - \mathbf{S}(s_k^{old}) \| &\leq c_S, \end{aligned}$$

for  $k = 1, \dots, K$ . Since there is no certainty that the KMCC algorithm will converge for any given image, the maximum allowed number of iterations was chosen to be 20; if this is exceeded, the method proceeds as though the KMCC algorithm had converged.

### III. FAST LARGE-FORMAT IMAGE SEGMENTATION

The approach presented in the previous section is considerably fast when the algorithm is applied to images of relatively small dimensions, e.g.  $150 \times 100$  pixels. When the image size increases, time efficiency degrades quickly, since the computational complexity of the algorithm is approximately proportional to the number of pixels of the image. In order to provide a more efficient scheme for the segmentation of relatively large images, one could take advantage of a reasonable assumption already made in the previous section, namely that regions falling below a size threshold  $\psi$ , that was defined to be equal to 0.75% of the total image area in subsection II-E, are insignificant for the multimedia applications where segmentation is required. For relatively large images, this threshold corresponds to a large number of pixels. This reveals the potential of applying the segmentation algorithm of the previous section to reduced versions of the original images [18], [42], [32]. These would be large enough for even insignificant objects to be detectable, yet significantly smaller than the original ones, thus faster to segment.

In this paper, the reduced image is derived from the original image by associating each  $R \times R$  block of the original image with a pixel of the reduced one (Fig. 3), where  $R$  is the reduction factor. A necessary condition for all significant objects to be detectable in the reduced image is that the size threshold for the reduced image, expressed as the minimum number of pixels, be much greater than one; otherwise, even significant objects could be difficult or even impossible to detect. Thus,

$$\frac{\psi}{R^2} \gg 1 \quad (14)$$

where  $\psi$  has been defined as  $\psi = \frac{0.75}{100} \cdot y_{max} \cdot x_{max}$  and  $x_{max}, y_{max}$  are the original image dimensions. A graphical representation of this three-layer segmentation scheme is presented in Fig. 3. The segmentation algorithm proposed in Sec. II is applied to the Layer 1 image. Consequently, its initial clustering process described in subsection II-C is performed on the Layer 2 image.

The use of a reduced image improves the time efficiency of the segmentation process, but does so at the expense of the quality of the segmentation result; edges between objects are crudely approximated by piecewise linear segments, lowering the perceptual quality of the result. To alleviate this problem, the use of the Bayes classifier for the reclassification of pixels is proposed. Reclassification of all pixels of the original image is unnecessary, since only those close to the edges of each region may have been misclassified due to the use of a reduced image. Thus, reclassification is restricted to the latter. The proximity of pixels to edges is evaluated using the output of the segmentation algorithm: the segmentation mask corresponding to the Layer 1 image. If a pixel of that mask, assigned to one region, is neighboring to pixels of  $\Gamma$  other regions,  $\Gamma \neq 0$ , the assignments of all pixels of the original image represented by that pixel of the Layer 1 image must be reevaluated, since each of them may belong to any one of the possible  $\Gamma + 1$  regions. In this way,  $G$  sets  $g_i^p, i = 1, \dots, G$  of disputed pixels are formed, each associated with a different set  $g_i^s, i = 1, \dots, G$  of possible regions (Fig. 4).

The reclassification of the disputed pixels is then performed using their intensity values only, as follows: Let  $\omega_k$  be the class of pixels of region  $s_k$ . According to the Bayes classifier [11], [8], a disputed pixel  $\mathbf{p}$ ,  $\mathbf{p} \in g_i^p$ , is assigned to region  $s_k$  if

$$p(\omega_k | \mathbf{I}(\mathbf{p})) > p(\omega_q | \mathbf{I}(\mathbf{p})), \quad \forall s_k, s_q \in g_i^s, k \neq q, \quad (15)$$

Using the Bayes Theorem, Eq. (15) can be rewritten as:

$$p(\mathbf{I}(\mathbf{p}) | \omega_k) \cdot p(\omega_k) > p(\mathbf{I}(\mathbf{p}) | \omega_q) \cdot p(\omega_q), \quad \forall s_k, s_q \in g_i^s, k \neq q, \quad (16)$$

The probability  $p(\omega_k)$  is the a priori probability of class  $\omega_k$ , whereas probability  $p(\mathbf{I}(\mathbf{p}) | \omega_k)$  is the density function of the intensities of pixels belonging to class  $\omega_k$ . The latter can be easily determined using the normalized histogram  $hist_k$  of each intensity component for the non-disputed pixels of region  $s_k$ :

$$p(\mathbf{I}(\mathbf{p}) | \omega_k) = \prod_{x \in \{L, a, b\}} hist_k^x(I_x(\mathbf{p})), \quad (17)$$

Regarding class prior probabilities, it can be assumed that among the pixels of group  $g_i^p$  the a priori probability of class  $\omega_k$ ,  $s_k \in g_i^s$ , is equal for all regions  $s_k \in g_i^s$ . The fact that reclassification is restricted to disputed pixels, i.e. pixels on edges between regions, along with the fact that each pixel of group  $g_i^p$  belongs to a block that is on the edges of all regions of group  $g_i^s$ , make the size of each region  $s_k \in g_i^s$  irrelevant to the employed methodology (Fig. 4); therefore, using region size for estimating prior probabilities would be unjustifiable. Consequently, the classification criterion of Eq. (16) is simplified to: pixel  $\mathbf{p}$ ,  $\mathbf{p} \in g_i^p$ , is assigned to region  $s_k$  if

$$\prod_{x \in \{L,a,b\}} \text{hist}_k^x(I_x(\mathbf{p})) > \prod_{x \in \{L,a,b\}} \text{hist}_q^x(I_x(\mathbf{p})), \quad \forall s_k, s_q \in g_i^s, k \neq q, \quad (18)$$

The block diagram of the pixel reclassification scheme is presented in Fig. 5.

#### IV. EXPERIMENTAL RESULTS

The segmentation algorithm described in Sec. II was applied to a variety of synthetic and natural color images of typical dimensions  $150 \times 100$  and  $192 \times 128$ ; these were mainly selected from the Corel gallery [6] and the MIT Vision Texture (VisTex) database [27], while some were collected from the web. Results for natural images are presented in Fig. 6, along with results obtained using two simpler variants of the proposed algorithm: one that neither uses texture features nor enforces connectivity constraints during pixel classification (denoted KM1), and one that differs from the proposed algorithm in that texture is used only by the conditional filtering module and not by KMCC (denoted KM2). Figure 6 illustrates the shortcomings of such simpler variants and the improvement attained using the proposed algorithm.

Additional results for  $192 \times 128$  pixel images [6], [27] are presented in Figs. 7 and 8, along with the corresponding results of the Blobworld segmentation algorithm [2], [4] (obtained using source code from <http://elib.cs.berkeley.edu/src/blobworld/>). The Blobworld algorithm is one that has been extensively tested and has produced very satisfactory results. It is based on modelling the joint distribution of color, texture and position features with a mixture of Gaussians; the Expectation-Maximization (EM) algorithm is employed to estimate the parameters of this model. From the results presented here it can be seen that the proposed algorithm tends to produce more accurate region boundaries. This, along with the fact that every pixel of the image is assigned to a region, make the proposed algorithm suitable not only for content-based image retrieval but also for region-of-interest coding, a task for which the Blobworld algorithm is not suited. A comparison, using an 800Mhz Intel PIII PC, of the time efficiency of the algorithm of Sec. II and Blobworld (uses mostly Matlab code) can be seen in table I.

Objective evaluation of segmentation quality was performed using synthetic images, created using the reference textures of the VisTex database [27], and natural images

of the Corel gallery [6]; reference masks for the latter were manually generated. The employed evaluation criterion is based on the measure of *spatial accuracy* proposed in [46] for foreground/background masks. For the purpose of evaluating still image segmentation results, each reference region  $r_q$  of the reference mask is associated with a different created region  $s_k$  on the basis of region overlapping (i.e.  $s_k$  is chosen so that  $r_q \cap s_k$  is maximized). Then, the spatial accuracy of the segmentation is evaluated by separately considering each reference region as a foreground reference region and applying the criterion of [46] for the pair of  $\{r_q, s_k\}$ ; during this process, all other reference regions are treated as background. A weighted sum of misclassified pixels for each reference region is the output of this process. The sum of these error measures for all reference regions is used for the objective evaluation of segmentation accuracy; values of the sum closer to zero indicate better segmentation. The test images used for objective evaluation are presented in Fig. 9, along with their reference masks and results of the algorithm proposed here (KMCC), the Blobworld algorithm [4] and a modified RSST algorithm [30]. The latter is based on adding to the original RSST algorithm [28] a second stage of region merging, using a distance function that does not discourage the creation of large regions. The values of the evaluation metric for the images of Fig. 9 are shown in table II; results for the simpler variants KM1 and KM2 of the proposed algorithm are also shown. These results clearly demonstrate that algorithms using only color features (KM1, RSST) perform poorly on synthetic or natural images containing textured regions; however, they may be useful in interactive applications requiring some degree of over-segmentation [5]. Both the Blobworld and the proposed KMCC algorithm produce significantly better results. The superiority of the proposed algorithm in producing accurate region boundaries without over-segmentation is demonstrated in Fig. 9 and is numerically verified.

The efficiency of the fast large-format image segmentation framework of Sec. III was also evaluated, by comparing its time-efficiency and perceptual segmentation quality with two other segmentation schemes: the direct application of the algorithm of Sec. II to the large-format images and the application of the same algorithm to reduced images, as in Sec. III, without the subsequent application of the quality improvement stage that employs the Bayes classifier. The time-efficiency of the three aforementioned segmentation schemes was evaluated on an 800MHz Intel Pentium III PC, using a set of 100  $730 \times 490$  pixel images from the Corel gallery [6]. The average image segmentation time for the images of this set is presented in table III. The perceptual quality of the three schemes can be evaluated using the segmentation examples of Fig. 10. As can be seen, the perceptual quality of the proposed fast large-format image segmentation scheme is generally higher than that of the direct approach, due to superiority of the Bayes classifier, compared to the euclidian distance classification used by the KMCC algorithm. The quality of the reduced image approach is clearly lower, due to the fact that regions

are composed of blocks of pixels rather than pixels. Note that the three different schemes do not necessarily produce the same number of regions for a given image. This is due to the fact that the segmentation algorithm of Sec. II is applied to different images under each of the three schemes.

To test the sensitivity of the proposed algorithms to variations of the threshold values, additional tests were conducted using threshold values deviating from those described in the previous sections and summarized in table IV. The values used for these tests and the corresponding results for the images of Fig. 10 are illustrated in table IV; in all cases the results are satisfactory. Note that, with the exception of the results presented in table IV, all results were produced using the original threshold values reported in Sec. II and table IV. The plethora of heterogeneous images shown in Figs. 6 through 10 are seen to be properly segmented without changing any threshold values, which is another indication of the low threshold dependency of the proposed algorithms.

Finally, an important observation regarding the proposed large-format image segmentation methodology is that it requires nothing of the employed segmentation algorithm, apart from an image as its input and a segmentation mask of the same dimensions as the input image be its output. Thus, this methodology can be used in combination not only with the proposed segmentation algorithm but also with a variety of other segmentation algorithms described in the literature.

## V. CONCLUSIONS

A methodology was presented for the segmentation of color images using intensity, position and texture features to facilitate the formation of regions corresponding to the objects contained in the image. Furthermore, a framework for the fast segmentation of large-format color images was presented, to improve the time efficiency of the segmentation process. This framework combines the segmentation algorithm of Sec. II with a Bayes classifier and, as discussed in Sec. IV, features improved time-efficiency and higher perceptual segmentation quality compared to the algorithm of Sec. II, when applied to large-format images. Not only the proposed segmentation algorithm but also others could be easily combined with the fast large-format image segmentation framework to their benefit.

The proposed algorithms are appropriate for use as part of an object-based multimedia application, such as object-based image querying, or for defining regions of interest for content-based coding of still images, in the context of the JPEG2000 standard.

## REFERENCES

- [1] S. T. Acton, A. C. Bovik and M. M. Crawford, "Anisotropic diffusion pyramids for image segmentation", *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994.
- [2] S. Belongie, C. Carson, H. Greenspan and J. Malik, "Color- and texture-based image segmentation using EM and its application to content-based image retrieval", *Proc. Sixth International Conference on Computer Vision*, Jan. 1998.
- [3] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris and M. G. Strintzis, "Content-based Watermarking for Indexing Using Robust Segmentation", *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, Tampere, Finland, May 2001.
- [4] C. Carson, S. Belongie, H. Greenspan and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, Aug. 2002.
- [5] S. Cooray, N. O'Connor, S. Marlow, N. Murphy and T. Curran, "Semi-Automatic Video Object Segmentation using Recursive Shortest Spanning Tree and Binary Partition Tree", *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, Tampere, Finland, May 2001.
- [6] *Corel stock photo library*, Corel Corp., Ontario, Canada.
- [7] Y. Deng and B.S. Manjunath, "An efficient low-dimensional color indexing scheme for region-based image retrieval", *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, pp. 3017-3020.
- [8] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley-Interscience, 2001.
- [9] J. Fan, D.K.Y. Yau, A.K. Elmagarmid and W.G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing", *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1454-1466, Oct. 2001.
- [10] K. S. Fu and J. K. Mui, "A Survey on Image Segmentation", *Pattern Recognition*, vol. 13, pp. 3-16, 1981.
- [11] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition*, Academic Press, 1990.
- [12] H. Gao, W.-C. Siu and C.-H. Hou, "Improved techniques for automatic image segmentation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1273-1280, Dec. 2001.
- [13] D. Geiger and A. Yuille, "A Common Framework for Image Segmentation", *International Journal of Computer Vision*, vol. 6, pp. 227-243, 1991.
- [14] N. Grammalidis, S. Malassiotis, D. Tzovaras and M.G. Strintzis, "Stereo Image Sequence Coding Based on Three-Dimensional Motion Estimation and Compensation", *Signal Processing: Image Communication*, vol. 7, no. 2, pp. 129-145, Aug. 1995.
- [15] R. M. Haralick and L. G. Sapiro, "Image Segmentation Techniques", *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 100-132, 1985.
- [16] R. Jain, R. Kasturi and B. G. Schunck, *Machine Vision*, McGraw-Hill International Editions, 1995.
- [17] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, vol. 1, pp. 312-331, 1988.
- [18] M.D. Kelly, "Edge Detection by Computer Using Planning", *Machine Intell.*, vol. VI, pp. 397-409, 1971.
- [19] I. Kompatsiaris, E. Triantafyllou and M. G. Strintzis, "Region-Based Color Image Indexing and Retrieval", *Proc. IEEE International Conference on Image Processing*, Thessaloniki, Greece, Oct. 2001.
- [20] I. Kompatsiaris and M. G. Strintzis, "3D Representation of Videoconference Image Sequences Using VRML 2.0", *European Conference for Multimedia Applications Services and Techniques (ECMAST'98)*, Berlin, Germany, May 1998.
- [21] I. Kompatsiaris and M. G. Strintzis, "Content-Based Representation of Colour Image Sequences", *IEEE International Conference on Acoustics, Speech, and Signal Processing 2001 (ICASSP 2001)*, Salt Lake City, USA, May 2001.
- [22] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, Dec. 2000.
- [23] I. Kompatsiaris and M. G. Strintzis, "Visualisation of Videoconference Image Sequences Using VRML 2.0", *IX European Signal Processing Conference*, Island of Rhodes, Greece, Sep. 1998.
- [24] S. Liapis, E. Sifakis and G. Tziritas, "Color and/or Texture Segmentation using Deterministic Relaxation and Fast Marching Algorithms", *Proc. Intern. Conf. on Pattern Recognition*, vol. 3, pp. 621-624, Sep. 2000.
- [25] M. Matheron, *Random Sets and Integral Geometry*, Wiley, New York, 1975.
- [26] J. McQueen, "Some Methods for Classification and Analysis of Multivariate Observations", *Proc. 5th Berkely Symp. on Math. Stat. and Prob.*, vol. 1, pp. 281-296, 1967.
- [27] *MIT Vision Texture (VisTex) database*, <http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.
- [28] O.J. Morris, M.J. Lee and A.G. Constantinides, "Graph Theory

- for Image Analysis: an Approach Based on the Shortest Spanning Tree", *IEE Proceedings*, vol. 133, no. 2, pp. 146-152, 1986.
- [29] MPEG Requirements Group, "MPEG-7 Context and Objectives", *Doc. ISO/MPEEC 2460*, MPEG Atlantic City Meeting, Oct. 1998.
- [30] N. O'Connor, T. Adamek, S. Sav, N. Murphy and S. Marlow, "QIMERA: A Software Platform for Video Object Segmentation and Tracking", *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, London, UK, April 2003.
- [31] P. Perona and J. Malik, "Scale Space and Edge-Detection Using Anisotropic Diffusion", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 12, no. 7, pp. 629-639, Jul. 1990.
- [32] M.R. Rezaee, P.M.J. van der Zwet, B.P.E. Lelieveldt, R.J. van der Geest and J.H.C. Reiber, "A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering", *IEEE Trans. on Image Processing*, vol. 9, no. 7, pp. 1238-1248, July 2000.
- [33] S. Sakaïda, Y. Shishikui, Y. Tanaka and I. Yuyama, "Image segmentation by integration approach using initial dependence of k-means algorithm", *Proc. Picture Coding Symposium 97*, pp. 265-269, Berlin, Germany, Sep. 1997.
- [34] P. Salembier, L. Torres, F. Meyer and C. Gu, "Region-Based Video Coding Using Mathematical Morphology", *Proc. IEEE*, vol. 83, no. 6, pp. 843-857, Jun. 1995.
- [35] D. Santa-Cruz and T. Ebrahimi, "An analytical study of JPEG 2000 functionalities", *Proc. IEEE International Conference on Image Processing*, 2000, vol. 2, pp. 49-52.
- [36] S. Z. Selim and M. A. Ismail, "K-means-type algorithms", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 6, no. 1, pp. 81-87, Jan. 1984.
- [37] J. Serra, *Image Analysis and Mathematical Morphology*, Academic, London, 1982.
- [38] T. Sikora, "The MPEG-4 Video Standard Verification Model", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19-31, Feb. 1997.
- [39] "Special Issue", *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image and Video Processing*, vol. 8, no. 5, pp. 19-31, Sep. 1998.
- [40] "Special Issue", *Signal Processing, Special Issue on Video Sequence Segmentation for Content-Based Processing and Manipulation*, vol. 66, no. 2, Apr. 1998.
- [41] L. H. Staib and J. S. Duncan, "Boundary Finding With Parametric Deformable Models", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 14, pp. 161-175, 1992.
- [42] M.M. Trivedi and J.C. Bezdek, "Low level segmentation of aerial images with fuzzy clustering", *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-16, no. 4, pp. 589-598, 1986.
- [43] E. Tuncel and L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modeling", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 776-781, Aug. 2000.
- [44] D. Tzovaras, N. Grammalidis and M. G. Strintzis, "Object-Based Coding of Stereo Image Sequences using Joint 3-D Motion/Disparity Compensation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 801-811, Apr. 1997.
- [45] M. Unser, "Texture classification and segmentation using wavelet frames", *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1549-1560, Nov. 1995.
- [46] P. Villegas, X. Marichal and A. Salcedo, "Objective Evaluation of Segmentation Masks in Video Sequences", *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, pp. 85-88, Berlin, May 1999.
- [47] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 13, no. 6, pp. 583-598, June 1991.
- [48] D. Wang, "A multiscale gradient algorithm for image segmentation using watersheds", *Pattern Recognition*, vol. 30, no. 12, pp. 2043-2052, 1997.



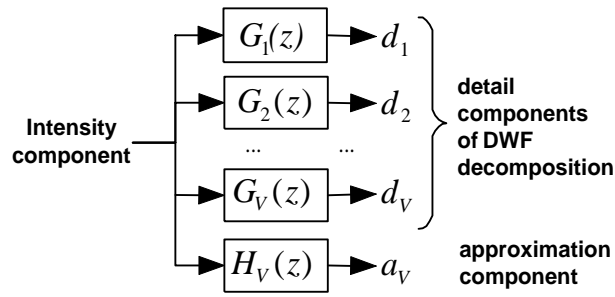


Fig. 1. 1-D Discrete Wavelet Frames decomposition of  $V$  levels.

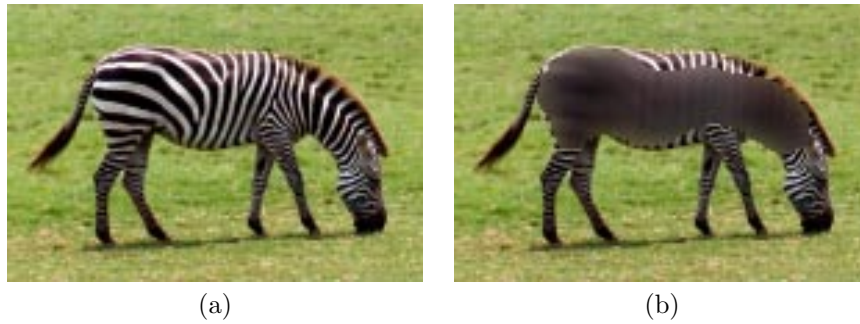


Fig. 2. (a) Original image “zebra”,  $150 \times 100$  pixels. (b) Filtered image.

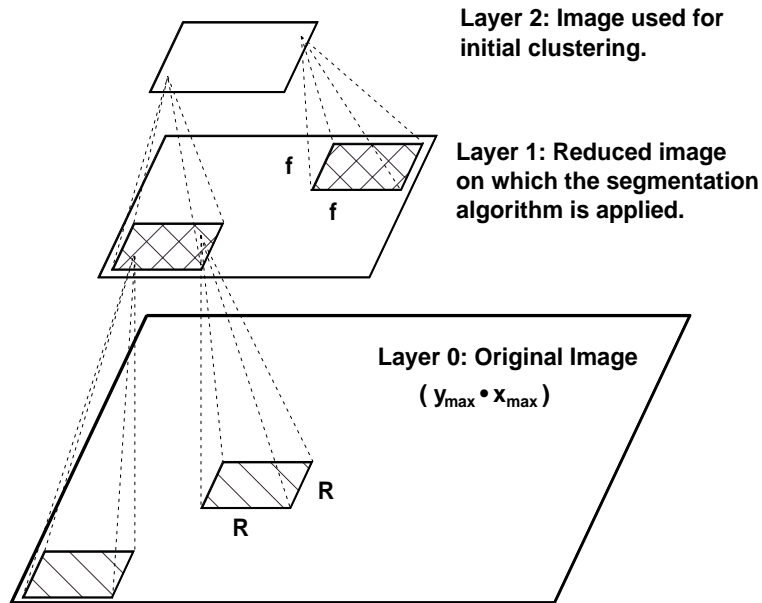


Fig. 3. Three-layer segmentation scheme.

TABLE I  
AVERAGE SEGMENTATION TIME FOR  $192 \times 128$  PIXEL IMAGES.

Segmentation Scheme	Average Time (sec.)
Direct application of the segmentation algorithm of Sec. II	65.5
Application of the blobworld algorithm	226.2

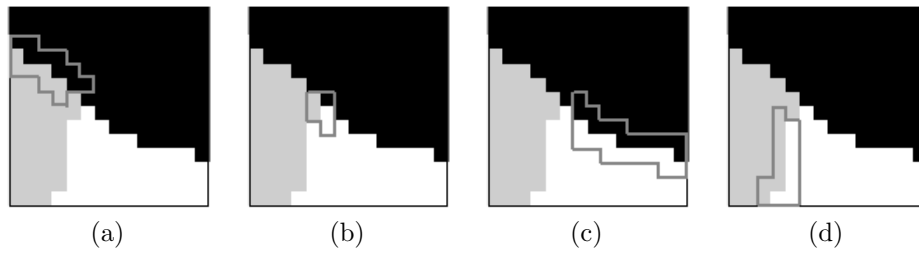


Fig. 4. A magnified  $14 \times 14$  pixel area of a layer 1 segmentation mask. If black, gray and white pixels belong to regions  $s_0$ ,  $s_1$  and  $s_2$  respectively, then the marked areas denote (a) pixel set  $g_1^p$ , associated with region set  $g_1^s = \{s_0, s_1\}$ , (b) pixel set  $g_2^p$ , associated with region set  $g_2^s = \{s_0, s_1, s_2\}$ , (c) pixel set  $g_3^p$ , associated with region set  $g_3^s = \{s_0, s_2\}$ , (d) pixel set  $g_4^p$ , associated with region set  $g_4^s = \{s_1, s_2\}$ .

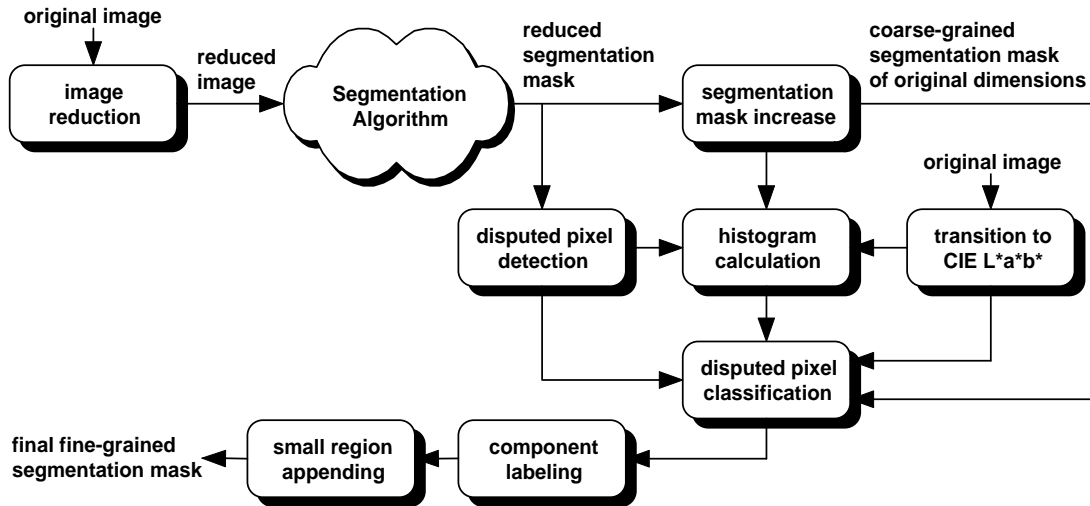


Fig. 5. Block diagram of the proposed framework for fast segmentation of large-format color images.

TABLE II  
NUMERICAL EVALUATION OF THE SEGMENTATIONS OF FIG. 9

images	KM1	KM2	RSST	Blobworld	KMCC
synth1	142.923679	6.453033	105.339744	12.188144	<b>1.260071</b>
synth2	205.812701	15.309401	187.123016	40.027074	<b>1.787774</b>
synth3	66.207026	39.597181	105.995881	45.812201	<b>2.167452</b>
synth4	226.903358	<b>16.022338</b>	78.353790	56.613260	42.442787
synth5	147.495911	70.91871	136.206447	<b>34.720163</b>	50.283481
synth6	127.208613	1.898975	73.851239	10.601577	<b>1.197819</b>
butterfly1	85.742792	11.218476	57.476854	29.533668	<b>9.940959</b>
butterfly2	71.658535	62.490798	22.572128	48.468529	<b>7.800168</b>
sunset	44.383718	44.386698	68.794582	89.307062	<b>5.722744</b>
bear	61.268402	62.992715	86.269010	<b>55.090216</b>	60.948571

TABLE III  
AVERAGE SEGMENTATION TIME FOR  $730 \times 490$  PIXEL IMAGES.

Segmentation Scheme	Average Time (sec.)
Direct application of the segmentation algorithm of Sec. II to $730 \times 490$ pixel images	2494.28
Application of the segmentation algorithm of Sec. II to reduced images (reduction factor $R = 8$ )	18.92
Application of the large-format image segmentation framework of Sec. III (reduction factor $R = 8$ )	47.55

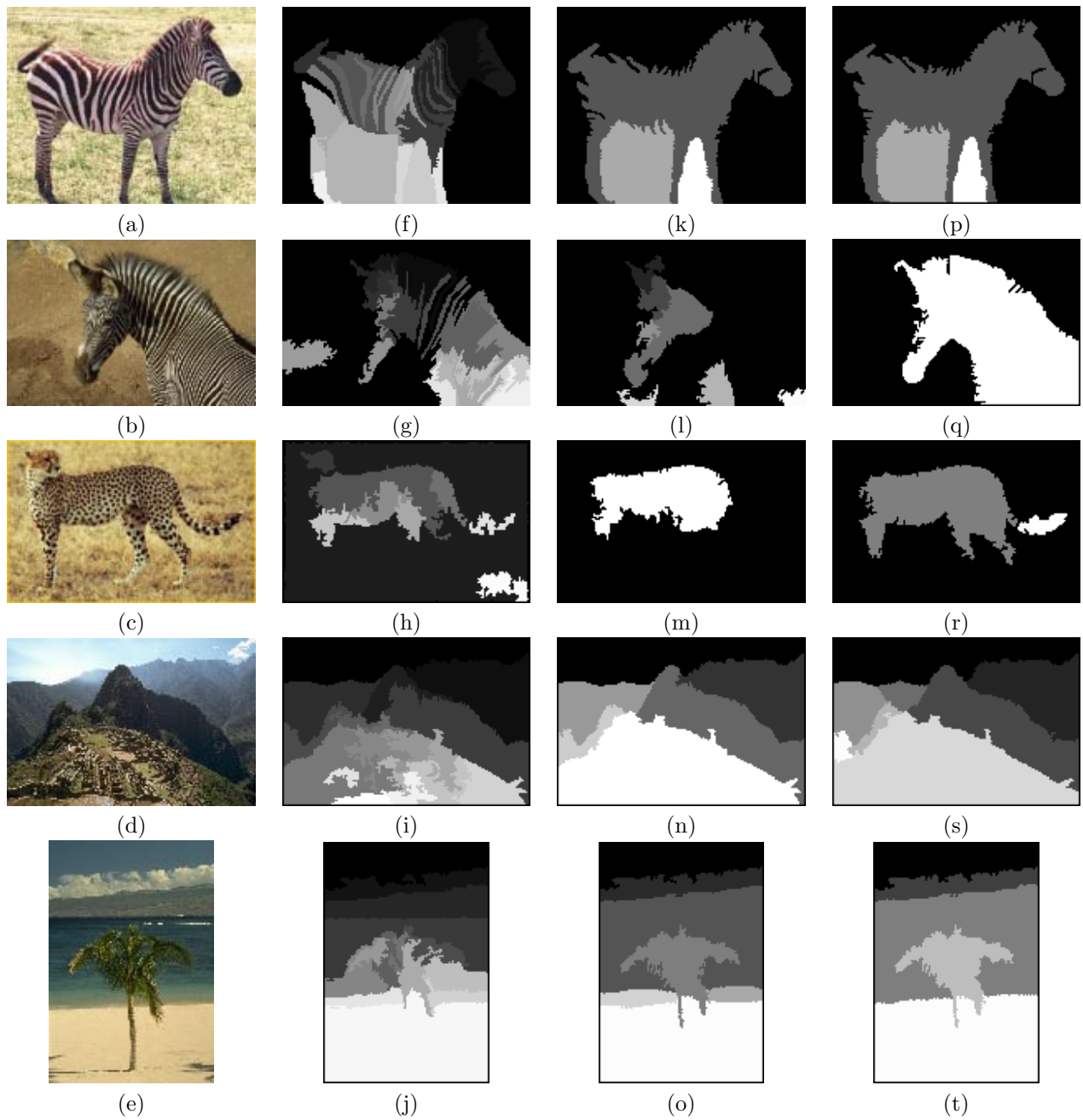


Fig. 6. Image segmentation examples: (a)-(e) Original images of approximate dimensions  $150 \times 100$  pixels. (f)-(j) Segmentation masks, produced by a variant of the algorithm of Sec. II, that neither uses texture features nor enforces connectivity constraints. (k)-(o) Segmentation masks, produced by a variant of the algorithm of Sec. II, that uses texture features only by the conditional filtering module and not by KMCC. (p)-(t) Segmentation masks, produced by the algorithm of Sec. II.

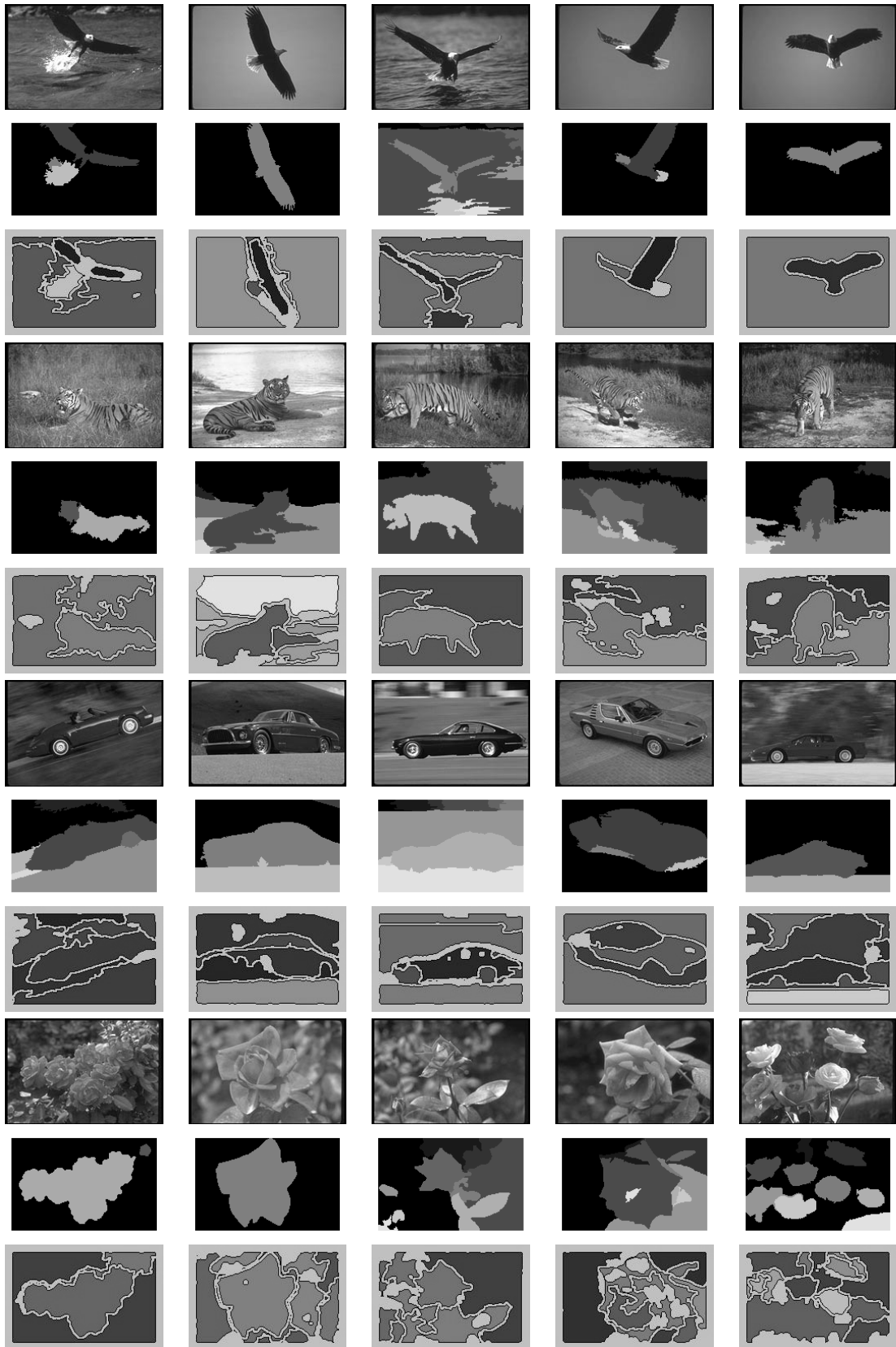


Fig. 7. Segmentation results for images belonging to various classes of the Corel gallery. Results of the proposed algorithm are shown below each original image; below these, results of the Blobworld algorithm are shown.

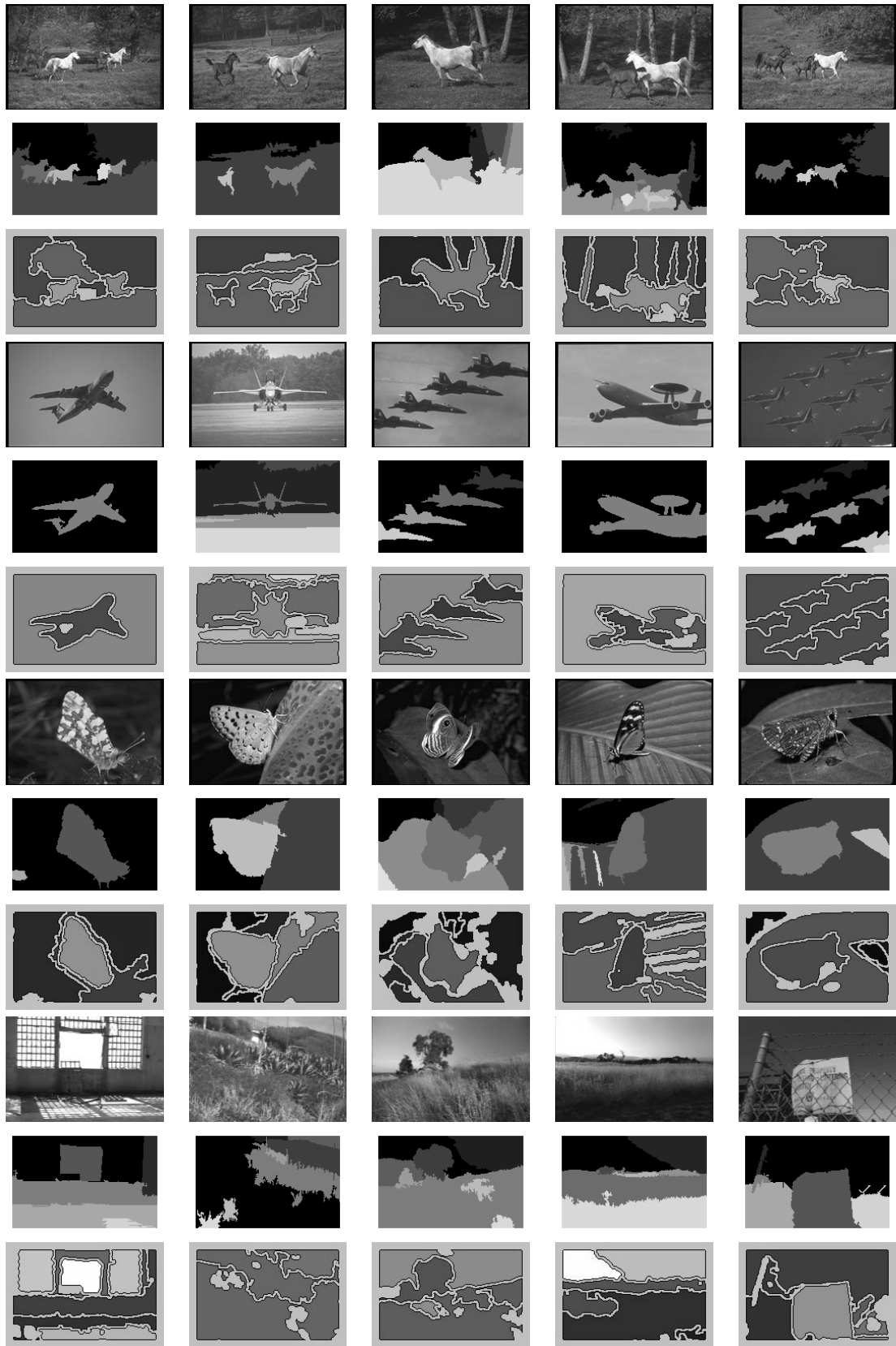


Fig. 8. Segmentation results for images belonging to various classes of the Corel gallery, and MIT's VisTex database. Results of the proposed algorithm are shown below each original image; below these, results of the Blobworld algorithm are shown.

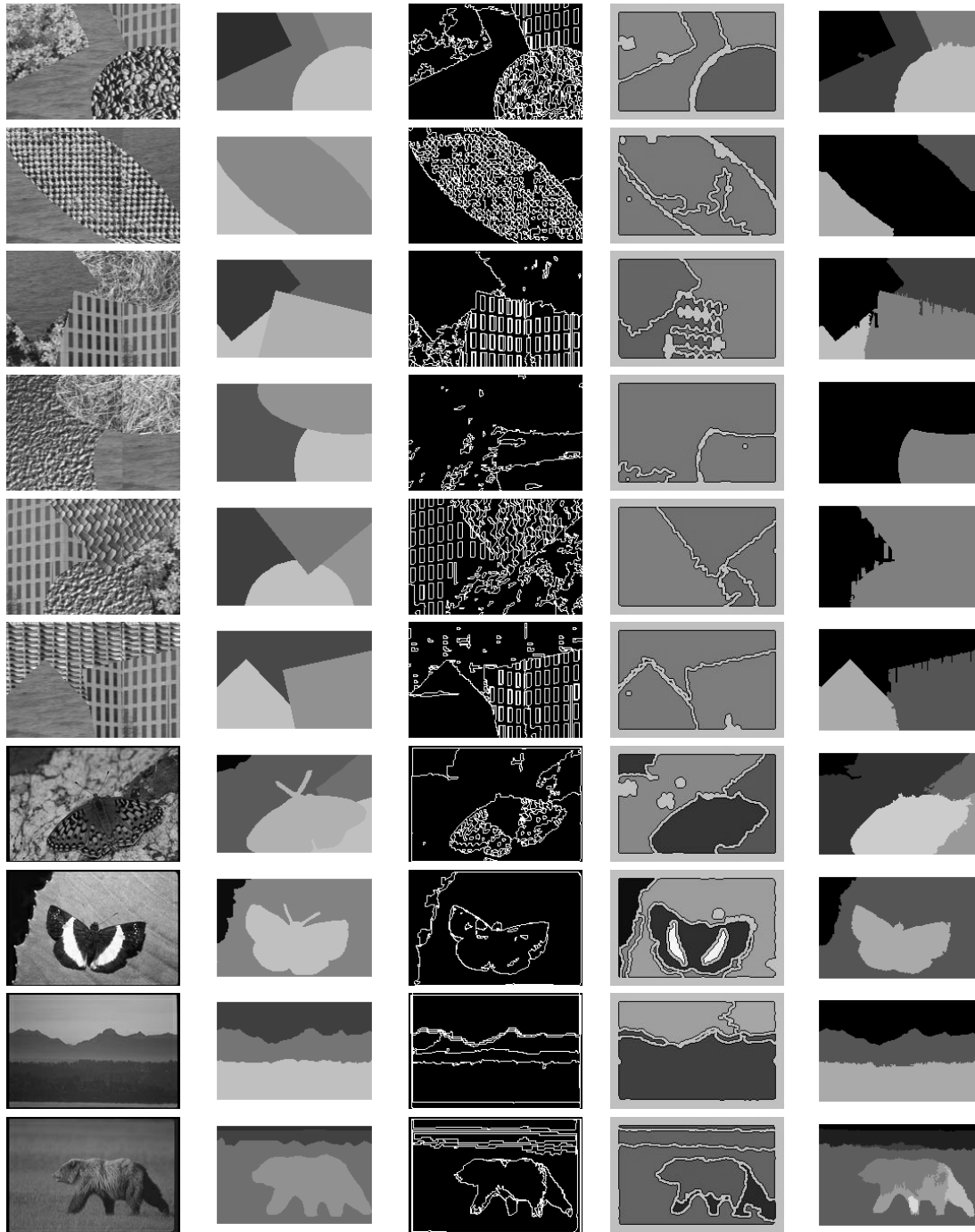


Fig. 9. Segmentation results for synthetic and natural images used for numerical evaluation. Synthetic images were created using the reference textures of MIT's VisTex database. Reference masks are shown in the second column; results for a modified RSST, the Blobworld algorithm and the proposed algorithm are shown in columns 3 to 5, respectively.

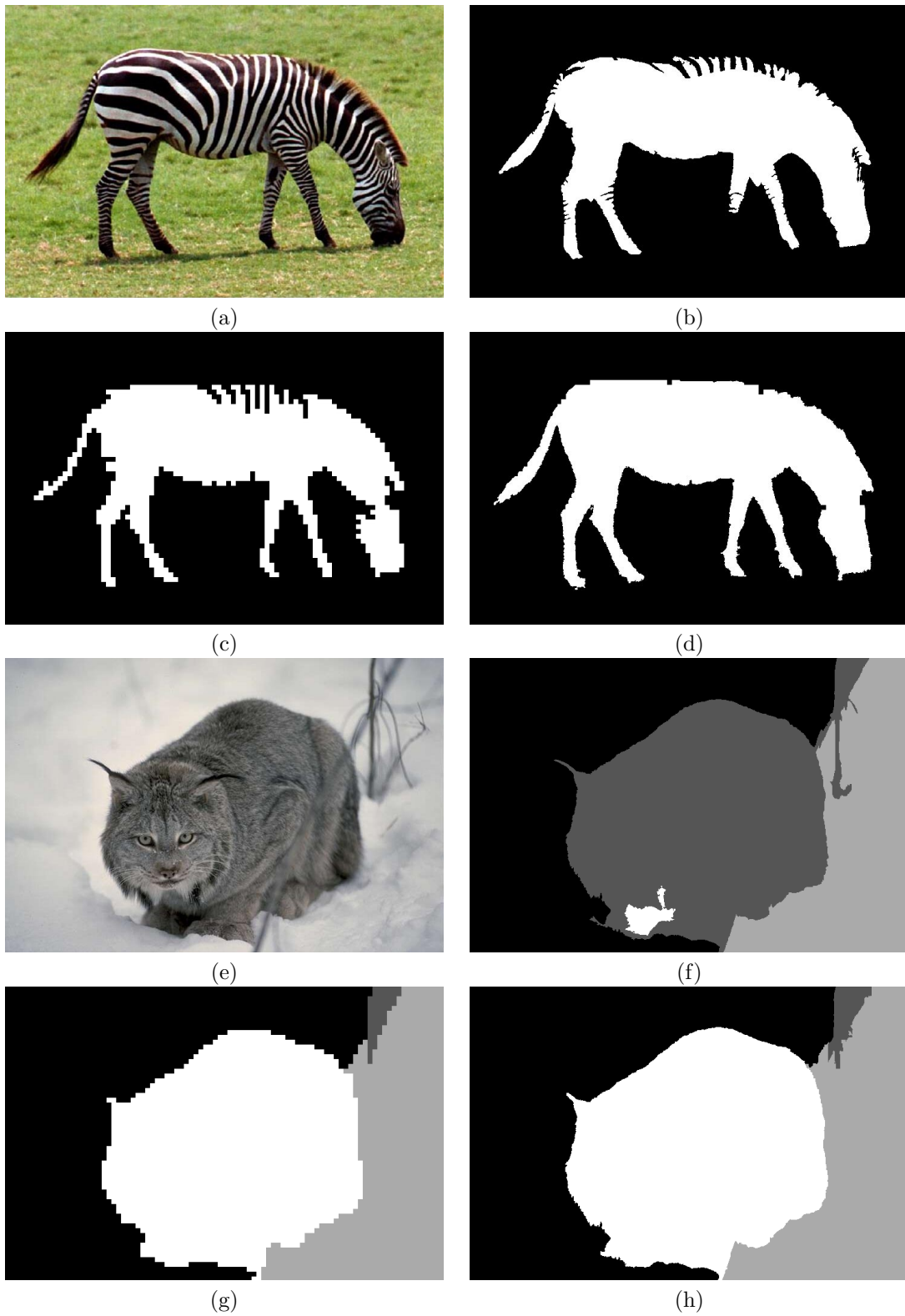


Fig. 10. Segmentation examples: (a)&(e) Original large-format images. (b)&(f) Direct application of the segmentation algorithm of Sec. II. (c)&(g) Application of the same algorithm on reduced images (reduction factor  $R = 8$ ). (d)&(h) Results of the large-format image segmentation framework.

TABLE IV  
THRESHOLD DEPENDENCY EXPERIMENTS







Original threshold	Outcome	
New values		
initial clustering parameter $\gamma = 0.4$ <hr/> 0.3, 0.5	using $\gamma = 0.5$ , no changes were observed; $\gamma = 0.3$ resulted in an additional small region being formed in the “zebra” image, due to the imperfect texture homogeneity of the “zebra” object	
$\tau = \max\{0.65 \cdot T_{max}, 14\}$ <hr/> $\max\{0.6 \cdot T_{max}, 12\}$ , $\max\{0.7 \cdot T_{max}, 16\}$	in the first case ( $\tau = \max\{0.6 \cdot T_{max}, 12\}$ ), no changes were observed; in the second case, an extra region was formed in the “cat” image	
size thresholds $\{\xi, \psi\} = \{0.5\%$ , $0.75\%\}$ of the image area <hr/> $\{0.4\%, 0.6\%\}$ , $\{0.6\%, 0.9\%\}$	in the first case ( $\{0.4\%, 0.6\%\}$ ), no changes were observed; in the second case, one region of the “cat” image was rejected for being too small	
merging threshold $\mu$ (Eq. 13) <hr/> $\mu' = 0.8 \cdot \mu$ , $\mu' = 1.2 \cdot \mu$	in the first case ( $\mu' = 0.8 \cdot \mu$ ), an extra region was formed in the “cat” image; in the second case, no changes were observed	
number of blocks for initial clustering $L \approx 75$ <hr/> 50, 100, 150	using 50 blocks had no effect on the segmentation results; using 100 or 150 blocks resulted in an additional small region being formed in the “zebra” image, due to the imperfect texture homogeneity of the “zebra” object	
$\lambda_1 = 1.0$ <hr/> 0.8, 1.3	using $\lambda_1 = 0.8$ had no effect on the results; using $\lambda_1 = 1.3$ resulted in an additional small region being formed in the “zebra” image	
convergence thresholds $c_I, c_T, c_S$ <hr/> $c'_X = 0.5 \cdot c_X, X \in \{I, T, S\}$ , $c'_X = 1.5 \cdot c_X, X \in \{I, T, S\}$	in both cases, no changes were observed	



TABLE V  
LEGEND OF SYMBOLS

Symbol	Description
$\mathbf{p} = [p_x \ p_y]$	pixel
$x_{max}, y_{max}$	image dimensions in pixels
$H(z), G(z), V$	prototype lowpass filter/ highpass filter/ levels of DWF decomposition
$\Phi$	square neighborhood for pixel texture calculation
$b_l, l = 1, \dots, L$	square blocks for initial clustering
$f$	size of blocks for initial clustering
$s_k$	spatial region
$M_k, \bar{M}$	size of region $s_k$ / average region size
$\mathbf{I}(\cdot), \mathbf{T}(\cdot)$	intensity/texture feature vector of pixel
$\mathbf{I}^b(\cdot), \mathbf{T}^b(\cdot)$	intensity/texture feature vector of block
$\mathbf{I}^s(\cdot), \mathbf{T}^s(\cdot), \mathbf{S}(\cdot)$	intensity/texture/position feature vector (center) of region
$\mathbf{J}(\cdot), \mathbf{J}^s(\cdot)$	pixel/region intensity feature vector after conditional filtering
$\mathbf{D}(\cdot), \mathbf{D}^b(\cdot), \mathbf{D}^s(\cdot)$	distance of pixel from region/ distance between blocks/ distance between regions
$C$	intensity and texture contrast (Sec. II-C)
$\gamma \in [0, 1]$	parameter used by the maximin algorithm (Sec. II-C)
$\tau$	texture threshold for conditional filtering (Eq. (10))
$\lambda_1, \lambda_2$	parameters controlling the weight with which texture/spatial differences are added to intensity differences (Eq. (5), (11), (12))
$\xi, \psi$	size thresholds (Sec. II-E)
$\mu$	merging threshold (Eq. (13))
$c_I, c_T, c_S$	KMCC intensity/texture/position convergence thresholds (Sec. II-E)
$R$	image reduction factor (Sec. III)
$g_i^p, g_i^s$	set of disputed pixels/ corresponding set of possible regions (Sec. III)