Still Image Segmentation Tools for Content-based Multimedia Applications *

Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis

Information Processing Laboratory
Electrical and Computer Engineering Department
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece

Informatics and Telematics Institute
1st Km Thermi-Panorama Rd,
Thessaloniki 57001, Greece
Email: strintzi@eng.auth.gr

ABSTRACT

In this paper, a novel color image segmentation algorithm and a novel approach to large-format image segmentation are presented, both focused on usage for image segmentation in content-based multimedia applications. The novel color image segmentation algorithm uses the Discrete Wavelet Frames decomposition to extract texture features and performs pixel classification using a novel initial clustering procedure and applying a novel variant of the K-Means-with-connectivity-constraint algorithm in the combined intensity and texture feature space. This novel scheme enables the unsupervised operation of the proposed segmentation algorithm. The novel large-format image segmentation scheme employs the aforementioned segmentation algorithm and exploits spatial redundancy to provide an elegant framework for the fast segmentation of large-format images. As shown by experimental evaluation, these novel algorithms provide fast segmentation with high perceptual segmentation quality.

1 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. 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 [1, 2], region-of-interest coding using the JPEG2000 standard [3]) and for video coding and indexing using the recently introduced MPEG4 and upcoming MPEG7 standards. The cornerstone of any such content-based multimedia application is the segmentation algorithm. The present work concentrates on addressing the issue of effective segmentation of still color images.

Segmentation methods for 2D images may be divided primarily into region-based and boundary-based methods [4]. Region-based approaches rely on the homogeneity of spatially localized features such as intensity and texture, whereas boundary-based methods use primarily gradient information to locate object boundaries. In this paper, a region-based approach is adopted. A novel segmentation algorithm is presented using a combination of position, intensity and texture information for the image pixels. An overview of the proposed algorithm can be seen in figure 1.

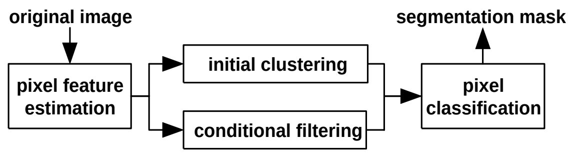

Figure 1: Overview of the proposed segmentation algorithm.

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. For this reason, a novel framework for the fast segmentation of large-format images is proposed (figure 2). The proposed framework 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.

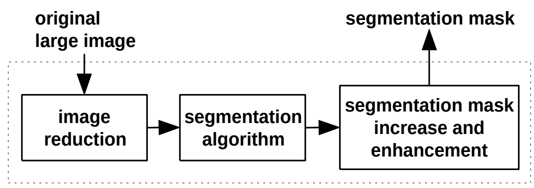

Figure 2: Proposed scheme for the fast segmentation of large-format images.

The paper is organized as follows: The segmentation algorithm is presented in section 2. In section 3, the framework for the fast segmentation of large-format images is developed, and the issues of time efficiency and perceptual quality are discussed. Section 4 contains an experimental evaluation of the developed methods and comparisons, and finally, conclusions are drawn in section 5.

2 COLOR IMAGE SEGMENTATION

2.1 Color and Texture Features

The color features used are the three intensity coordinates of the CIE L*a*b* color space. This color space is more suitable for the proposed algorithm than

*This work was supported by the European IST project IST-2000-32795 SCHEMA and the European IST project HISCORE. The assistance of COST211 quat is also greatfully acknowledged.

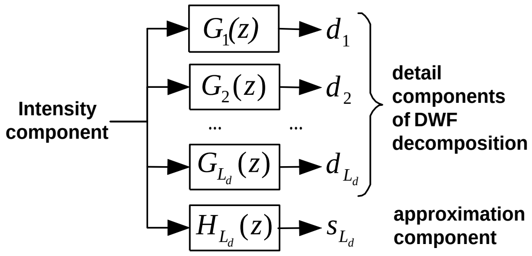

Figure 3: 1-D Discrete Wavelet Frames decomposition of L_d levels.

the widely used RGB color space because it is approximately perceptually uniform, i.e. the numerical distance in this color space is approximately proportional to the perceived color difference. The color feature vector of pixel $\mathbf{p}$, $I(\mathbf{p})$ is defined as

$$I(\mathbf{p}) = [I_L(\mathbf{p}), I_a(\mathbf{p}), I_b(\mathbf{p})]^T$$

In order to detect and characterize texture properties in the neighborhood of each pixel, the Discrete Wavelet Frames (DWF) decomposition, proposed by [5], is used. This is a method similar to the Discrete Wavelet Transform (DWT), that employs a filter bank based on a lowpass filter $H(z)$ to decompose each intensity component of the image to a set of subbands (figure 3) and uses the standard deviations of all detail components calculated in a neighborhood F of pixel $\mathbf{p}$ to characterize its texture. The complementary highpass filter $G(z)$ is defined with respect to the lowpass $H(z)$ as follows:

$$G(z) = zH(-z^{-1}). \tag{1}$$

The filters of the filter bank, $H_{L_d}(z), G_i(z), i = 1, \ldots, L_d$ are generated by the prototypes $H(z)$, $G(z)$, according to equations:

$$H_{i+1}(z) = H(z^{2^i})H_i(z),$$

$$G_{i+1}(z) = G(z^{2^i})H_i(z), i = 0, \ldots, L_d - 1$$

where $H_0(z) = 1$ is the necessary initial condition and L_d is the number of levels of decomposition. The 2-D fast iterative scheme presented in [5] is used by our segmentation algorithm. The filter bank employed is based on the lowpass Haar filter, that was shown to be a suitable choice.

$$H(z) = \frac{1}{2}(1 + z^{-1}). \tag{2}$$

For images of relatively small dimensions, e.g. 150×100 pixels, a two-dimensional DWF decomposition of two levels has been chosen ($L_d = 2$), resulting in an 18-component texture feature vector,

$$T(\mathbf{p}) = [\sigma_1(\mathbf{p}), \sigma_2(\mathbf{p}), \ldots, \sigma_{18}(\mathbf{p})]^T \tag{3}$$

whereas in the experiments where the 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.

2.2 Initial Clustering

In order to compute the initial values required by the KMCC algorithm, 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, $b_l, l = 1, \ldots, L$, is created. Let the center of block b_l be pixel $\mathbf{p}^l_{cntr}$. A color feature vector $I(b_l)$ and a texture feature vector $T(b_l)$ are then assigned to each block, as follows:

$$I(b_l) = \frac{1}{f^2}\sum_{m=1}^{f^2} I(\mathbf{p}^l_m), \;\; T(b_l) = T(\mathbf{p}^l_{cntr}), \tag{4}$$

where $\mathbf{p}^l_m, m = 1, \ldots, f^2$ are the pixels belonging to block b_l.

The distance between two blocks is defined as follows:

$$D(b_{l_1}, b_{l_2}) = \|I(b_{l_1}) - I(b_{l_2})\| + \|T(b_{l_1}) - T(b_{l_2})\|, \tag{5}$$

where $\|I(b_{l_1}) - I(b_{l_2})\|$, $\|T(b_{l_1}) - T(b_{l_2})\|$ are the Euclidean distances of the intensity and texture feature vectors.

The number of regions of the image is initially estimated by applying a variant of the maximin algorithm to this set of blocks. In the employed variant, the intensity and texture distance Db_{max} between the first two centers is calculated and candidate centers are accepted as region centers until the minimum distance between the candidate center and a region center is lower than $\gamma \cdot Db_{max}$, where $\gamma = 0.4$.

In order to obtain an estimate of the spatial centers of these regions, a simple K-Means algorithm is applied to the set of blocks, using the information produced by the maximin algorithm for its initialization. This is followed by the application of a recursive four-connectivity component labelling algorithm [6], so that a total of K' connected regions are identified. Their centers, including their spatial centers $S(s_k) = [S_x(s_k), S_y(s_k)]^T, k = 1, \ldots, K'$, are then calculated as the mean values of the intensity, texture and position features of the pixels belonging to the blocks assigned to each region.

2.3 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 (figure 5(a)). In order to facilitate the grouping of all these pixels in a single region based on their texture similarity, their intensity differences are reduced by conditionally applying a moving average filter.

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 $T(\mathbf{p})$; the filter is not applied if that norm is below a threshold T_{th}. The output of the conditional filtering module can thus be expressed as:

$$J(\mathbf{p}) = \begin{cases} I(\mathbf{p}) & \text{if } \|T(\mathbf{p})\| < T_{th} \\ \frac{1}{f^2}\sum_{m=1}^{f^2} I(\mathbf{p}_m) & \text{if } \|T(\mathbf{p})\| \geq T_{th} \end{cases} \tag{6}$$

$$T_{th} = \max\{0.65 \cdot T_{max}, 14\} \tag{7}$$

where T_{max} is the maximum value of the norm $\|T(\mathbf{p})\|$ in the image.

The output of the conditional filtering stage is used as input by the KMCC algorithm.

2.4 The K-Means with Connectivity Constraint Algorithm

Clustering based on the K-Means algorithm, originally proposed by McQueen [7], is a widely used region segmentation method [8, 9, 10] 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 K-Means-with-connectivity-constraint algorithm. In this algorithm, texture features are combined with the intensity and position information to permit efficient handling of textured objects.

The K-Means with connectivity constraint (KMCC) algorithm applied to the pixels of the image consists of the following steps:

1. The region number and the region centers are initialized, using the output of the initial clustering procedure described in section 2.2.

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:

$$D(\mathbf{p}, s_k) = \|J(\mathbf{p}) - J(s_k)\| + \\ + \|T(\mathbf{p}) - T(s_k)\| + \lambda \frac{\bar{A}}{A_k} \|\mathbf{p} - S(s_k)\| \quad (8)$$

where $S(s_k) = [S_x(s_k), S_y(s_k)]^T$ is the spatial center of region s_k,

$$\|J(\mathbf{p}) - J(s_k)\| = \sqrt{\sum_{r \in \{L,a,b\}} (J_r(\mathbf{p}) - J_r(s_k))^2},$$

$$\|T(\mathbf{p}) - T(s_k)\| = \sqrt{\sum_{q=1}^{18} (\sigma_q(\mathbf{p}) - T_q(s_k))^2},$$

$$\|\mathbf{p} - S(s_k)\| = \sqrt{\sum_{q \in \{x,y\}} (p_q - S_q(s_k))^2}$$

The area A_k of each region is defined as $A_k = M_k$, where M_k is the number of pixels assigned to region s_k, and $\bar{A}$ is the average area of all regions.

The regularization parameter λ is defined as:

$$\lambda = 0.4 \cdot \frac{Db_{max}}{\sqrt{p_{x,max}^2 + p_{y,max}^2}}$$

The regularization parameter λ is used to ensure that a pixel is assigned to a region primarily due to their similarity in intensity and texture characteristics. Being proportional to the intensity and texture contrast Db_{max} of the image, it ensures that even in low-contrast images, where intensity and texture differences are small, these will not become insignificant compared to spatial distances. The opposite would result in the formation of regions that would not correspond to the objects of the image.

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 [6].

4. Region centers are recalculated. Regions with areas below a size threshold $KMCC_{size}$ are dropped. In our experiments, the threshold $KMCC_{size}$ was equal to 0.5% of the total image area. This is lower than the minimum accepted region size, which in our experiments was defined as $th_{size} = 0.75\%$ of the total image area. The latter is used to ensure that no particularly small, meaningless regions are formed. Here, the slightly lower threshold $KMCC_{size}$ is used instead of th_{size}, to avoid dropping, in one iteration of the KMCC algorithm, regions that are close to threshold th_{size} and are likely to exceed it in future iterations. The number of regions K is also recalculated, taking into account only the remaining regions.

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

$$D(s_{k_1}, s_{k_2}) = \|J(s_{k_1}) - J(s_{k_2})\| + \\ + \|T(s_{k_1}) - T(s_{k_2})\| \leq th_{merge}$$

Threshold th_{merge} is image-specific, defined in our experiments by

$$th_{merge} = \begin{cases} 7.5 & \text{if } Db_{max} < 25 \\ 15 & \text{if } Db_{max} > 75 \\ 10 & \text{otherwise} \end{cases} \quad (9)$$

where Db_{max} is the intensity and texture contrast of the particular image, as defined in section 2.2

6. Region number K and region centers are reevaluated.

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

$$\|J(s_k) - J(s_k^{old})\| \leq th_I,$$

$$\|T(s_k) - T(s_k^{old})\| \le th_T,$$
$$\|S(s_k) - S(s_k^{old})\| \le th_S,$$

for $k = 1, \ldots, K$. 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.

Even though the region centers of particularly small regions are omitted in *Step 4* and the formation of large regions is encouraged in *Step 2*, there is no guarantee that such small regions will be absent from the segmentation mask following the convergence of the algorithm. Furthermore, despite the use of the moving average filter described in section 2.3, regions corresponding to a textured object may have remained separate, despite their texture similarity, due to differences in their intensity.

To alleviate the latter problem, a region merging procedure for regions of significant texture, based on their texture similarity, is employed, following the convergence of the KMCC algorithm. Chromatically homogeneous regions are not eligible for merging through this procedure, since that would lead to the merging of regions corresponding to different non-textured objects. In particular, neighboring regions s_k, s_q for which $\|T(s_k)\| > 10$ and $\|T(s_q)\| > 10$ are merged if their texture difference is lower than an experimentally determined threshold:

$$\|T(s_k) - T(s_q)\| < 7.5$$

This is followed by the elimination of any remaining small regions. Since these regions are not wanted, they are forced to merge with one of their neighboring regions, based on intensity and texture similarity: a small region s_{k_1}, $M_{k_1} < th_{size}$ is appended to region $s_{k_2}, k_2 = 1, \ldots, K, k_2 \neq k_1$, for which the distance

$$D(s_{k_1}, s_{k_2}) = \|J(s_{k_1}) - J(s_{k_2})\| + \|T(s_{k_1}) - T(s_{k_2})\|$$

is minimum. This procedure is performed for all small regions of the segmentation mask, until all such small regions are absorbed.

3 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. 150×100 pixels, but time efficiency degrades quickly as the image size increases. In order to provide a more efficient scheme for the segmentation of large-format images, one could take advantage of a reasonable assumption already made in the previous section, namely that regions falling below a size threshold $th_{size} = 0.75\%$ of the total image area are insignificant. For large-format 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. These would be large enough for even insignificant objects to be detectible, yet significantly smaller than the original ones, thus faster to segment.

A necessary condition for all significant objects to be detectible 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; thus,

$$\frac{th_{size} \cdot y_{max} \cdot x_{max}}{R^2} \gg 1,$$

where x_{max}, y_{max} are the original image dimensions, R is the reduction factor and th_{size} has been defined as $th_{size} = 0.75\%$.

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, since 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 minimization of the mean-square error in the reclassification of pixels belonging to blocks on edges between regions is proposed. If a block, assigned to one region, is neighboring to blocks of Γ other regions, $\Gamma \neq 0$, the assignments of all pixels of the original image represented by that block must be reevaluated, since each of them may belong to any one of the possible $\Gamma + 1$ regions. In this way, G sets $g_i^p, i = 1, \ldots, G$ of disputed pixels are formed, each associated with a different set $g_i^s, i = 1, \ldots, G$ of possible regions.

The reclassification of the disputed pixels is then performed using their intensity values only, as follows: Let ω_k be the class of pixels assigned to region s_k, not including any disputed pixels:

$$\omega_k = \{\mathbf{p} \in s_k, \mathbf{p} \notin g_i^p, i = 1, \ldots, G\}$$

According to the Bayes classifier for minimum classification error [11], a disputed pixel $\mathbf{p}$, $\mathbf{p} \in g_i^p$, is assigned to region s_k if

$$p(\omega_k|I(\mathbf{p})) > p(\omega_q|I(\mathbf{p})), \;\; \forall s_k, s_q \in g_i^s, k \neq q, \quad (10)$$

Using the Bayes Theorem and assuming that among the pixels of group g_i^p the a priory probability of class ω_k, $p(\omega_k)$, $s_k \in g_i^s$, is equal for all regions $s_k \in g_i^s$, the classification criterion of equation (10) is simplified to: pixel $\mathbf{p}$, $\mathbf{p} \in g_i^p$, is assigned to region s_k if

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

Probability $p(I(\mathbf{p})|\omega_k)$ can be calculated as

$$p(I(\mathbf{p})|\omega_k) = \prod_{x \in \{L,a,b\}} hist_k^x(I_x(\mathbf{p})), \qquad (11)$$

where $hist_k^x, \;\; x \in \{L, a, b\}$ are the normalized histograms of region s_k, excluding any disputed pixels.

As soon as the reclassification is completed, the connectivity of the formed regions is evaluated; this is followed by the appending of any small regions, similarly to what is done following the convergence of the KMCC algorithm.

Table 1: Average Segmentation Time for 730×490 pixel Images.

Segmentation Scheme	Time (sec)
Direct application of the algorithm of section 2 to 730×490 pixel images	2494.28
Application of the algorithm of section 2 to reduced images ($R = 8$)	18.92
Application of the framework of section 3 ($R = 8$)	47.55

4 EXPERIMENTAL RESULTS

The segmentation algorithm described in section 2 was applied to a variety of natural color images of dimensions 150×100 and 192×128, with very good results. A comparison of the proposed algorithm with the Blobworld algorithm [1] and a simpler variant of the algorithm of section 2 (figure 4) indicates the effectiveness of our approach.

The efficiency of the fast large-format image segmentation framework of section 3 was evaluated by comparing its time efficiency and perceptual segmentation quality with two other segmentation schemes: the direct application of the algorithm of section 2 to the large-format images and the application of the same algorithm to reduced images, as in section 3, without the subsequent application of the quality improvement stage that employs the Bayes classifier. The average image segmentation times for the aforementioned schemes, achieved on an 800MHz Intel Pentium III PC for 100 images of 730×490 pixels from the Corel gallery, are presented in table 1.

The perceptual quality of the three schemes can be evaluated using the segmentation examples of figure 5. As can be seen, the perceptual quality of the proposed large-format image segmentation scheme is higher than that of the direct approach, due to the superiority of the Bayes classifier, compared to the euclidian distance classification used by the KMCC algorithm. The quality of the reduced image approach (figure 5(c) and (g)) is clearly lower, due to the fact that regions are composed of blocks of pixels rather than pixels. These findings, along with the results of table 1, lead to the conclusion that the proposed large-format image segmentation framework is particularly efficient.

Another important observation regarding the proposed methodology is that it requires nothing of the employed segmentation algorithm, apart from producing a segmentation mask of the same dimensions as its input image. Thus, the proposed methodology can be used in combination with a variety of other segmentation algorithms described in the literature as well.

5 CONCLUSIONS

A novel methodology was presented for the segmentation of color images using intensity, position and texture features to facilitate the formation of regions corresponding to the real objects. Furthermore, a novel framework for the fast segmentation of large-format color images was presented, to improve the time efficiency of the segmentation process. The proposed algorithms are appropriate for use as part of a content-based multimedia application, such as image querying by example, or for defining regions of interest for content-based coding of images, in the context of the JPEG2000 standard.

REFERENCES

[1] 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," in *Proc. Sixth International Conference on Computer Vision*, January 1998.

[2] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, and M. G. Strintzis, "Content-based Watermarking for Indexing Using Robust Segmentation," in *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, Tampere, Finland, May 2001.

[3] D. Santa-Cruz and T. Ebrahimi, "An analytical study of JPEG 2000 functionalities," in *Proc. IEEE International Conference on Image Processing*, 2000, vol. 2, pp. 49–52.

[4] "Special Issue," *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image and Video Processing*, vol. 8, no. 5, Sept. 1998.

[5] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.

[6] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, McGraw-Hill International Editions, 1995.

[7] J. McQueen, "Some Methods for Classification and Analyis of Multivariate Observations," in *5th Berkely Symp. on Math. Stat. and Prob.*, 1967, vol. 1, pp. 281–296.

[8] S. Z. Selim and M. A. Ismail, "K-means-type algorithms," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 6, no. 1, pp. 81–87, Jan. 1984.

[9] S. Sakaida, Y. Shishikui, Y. Tanaka, and I. Yuyama, "Image segmentation by integration approach using initial dependence of k-means algorithm," in *Picture Coding Symposium 97*, Berlin, Germany, Sept. 1997, pp. 265–269.

[10] I. Kompatsiaris and M. G. Strintzis, "3D Representation of Videoconference Image Sequences Using VRML 2.0," in *European Conference for Multimedia Applications Services and Techniques (ECMAST'98)*, Berlin, Germany, May 1998, pp. 3–12.

[11] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition*, Academic Press, 1990.

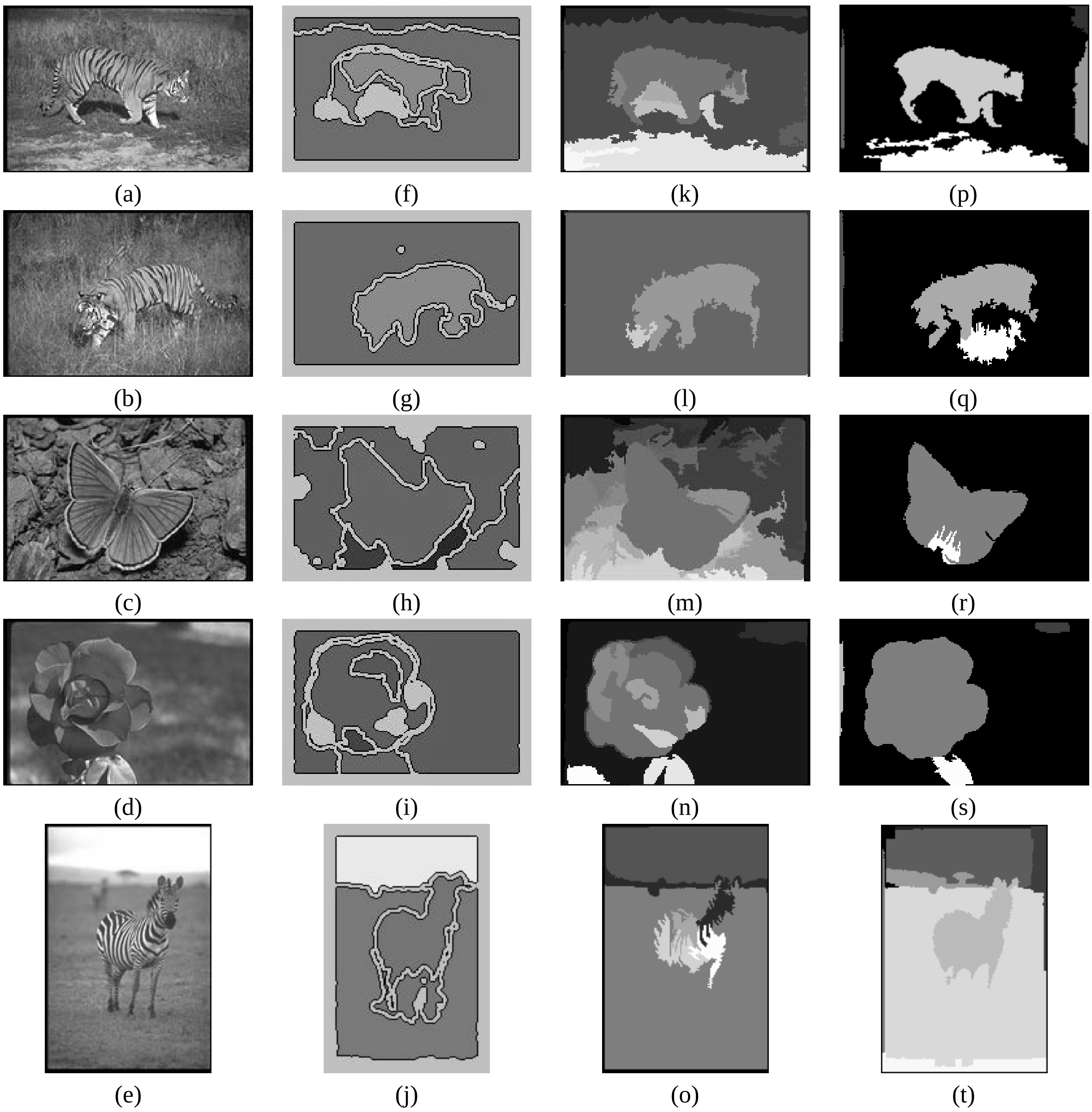

Figure 4: Image segmentation examples: (a)-(e) Original images 192×128 pixels, (f)-(j) Results produced by the Blobworld algorithm (using source code downloaded from http://elib.cs.berkeley.edu/src/blobworld/), (k)-(o) Segmentation masks, produced by a variant of the algorithm of section 2, that neither uses texture features nor enforces connectivity constraints, (p)-(t) Segmentation masks, produced by the algorithm of section 2.

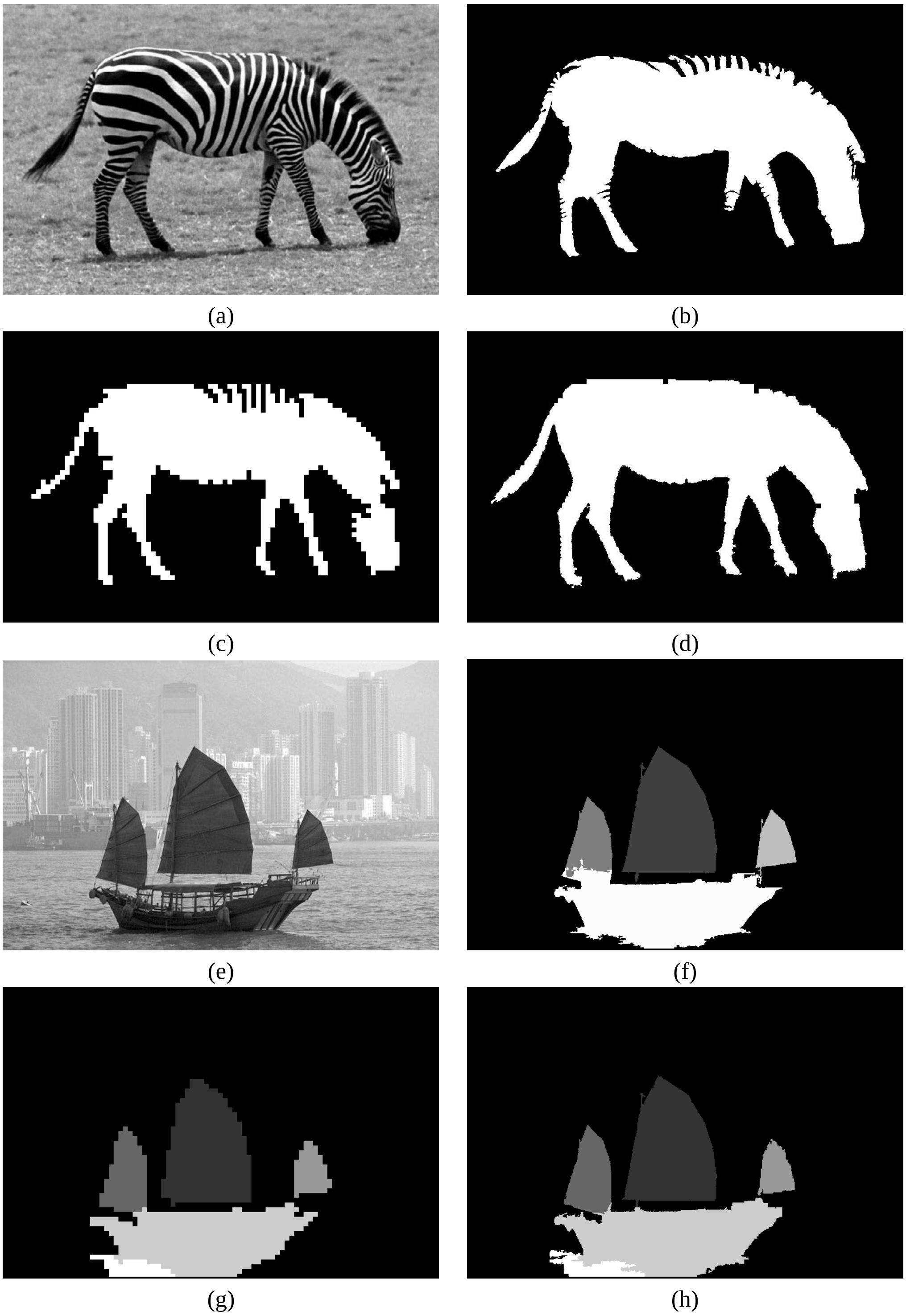

Figure 5: Segmentation examples: (a)&(e) Original large-format images. (b)&(f) Direct application of the segmentation algorithm of section 2. (c)&(g) Application of the same algorithm on reduced images (reduction factor $R = 8$). (d)&(h) Results of the large-format image segmentation framework.