Computer Vision and Image Understanding 116 (2012) 981–990

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

MDS-based segmentation model for the fusion of contour and texture cues in natural images ☆

Max Mignotte

DIRO, Univ. de Montréal, C.P. 6128, Succ. Centre-ville, Montréal, Canada H3C 3J7

ARTICLE INFO

Article history:
Received 8 July 2011
Accepted 11 May 2012
Available online 24 May 2012

Keywords:
Color textured image
Berkeley image database
Multidimensional scaling (MDS)
Non-linear dimensionality reduction
Unsupervised image segmentation
Probability Rand Index (PRI)
F-measure
Precision–recall framework

ABSTRACT

In this paper, we present an original image segmentation model based on a preliminary spatially adaptive non-linear data dimensionality reduction step integrating contour and texture cues. This new dimensionality reduction model aims at converting an input texture image into a noisy color image in order to greatly simplify its subsequent segmentation. In this latter de-texturing model, the (spatially adaptive) non-local constraints based on edge and contour cues allows us to efficiently regularize the reduced data (or the resulting de-textured color image) and to efficiently combine inhomogeneous region and edge based features in a data fusion/reduction model used as pre-processing step for a final segmentation task. In addition, a set of color/texture and edge-based adaptive spatial continuity constraints is imposed during the segmentation step. These improvements lead to an appealing and powerful two-step adaptive segmentation model, integrating contour and texture cues. Extensive experimental evaluation on the Berkeley image segmentation database demonstrates the efficiency of this hybrid segmentation model in terms of classification accuracy of pairwise pixels in the resulting segmentation map and in the precision–recall framework widespread used for evaluating contour detectors.

1. Introduction

Image segmentation is a frequent pre-processing step whose goal is to simplify the representation of an image into meaningful and spatially coherent regions (also known as *segments* or *superpixels*) with similar attributes such as consistent parts of objects or of the background. This low-level vision task, which changes the representation of an image into something that is easier to analyze, is often the preliminary and also crucial step in the development of many image understanding algorithms and computer vision systems.

A review of literature indicates that most of natural image segmentation algorithms can be generally divided into two categories, namely the so-called *region-based* and *edge-based* segmentation approaches. Region-based segmentation methods attempt to group spatially coherent regions with similar attributes. They include segmentation methods exploiting clustering schemes [32,22,46,34] (with fuzzy sets [37] or Gaussian mixture models [41] and after a possibly de-texturing approach [34,35]), mean-shift-based (or more generally mode seeking) procedures [9,25,6], watershed or [30] region growing strategies [11], lossy coding and compression models [26,30], MRF-based statistical [19,14,12,39,7], Bayesian [38,8] or graph-based models [43,16,45], variational methods [3,15,40], deformable surfaces [23], spectral clustering [10] and finally by some fusion techniques of multiple weak (region-based) segmentations [32,20,33].

On the other hand, edge-based segmentation methods rely on the prediction of local edge fragments which are simply defined by significant localized changes or discontinuities in some image features. In this way, classical edge detectors, such as Canny's [5] search for discontinuities in the luminance or color intensity while more recent edge-based segmentation methods can also use steerable filters [17], filter-based methods [24], energy-based models [36], probabilistic approaches [13] or take into account color and texture information with a preliminary learning step for the optimal cue combination [28]. It is worth mentioning that, due to the local nature of such contour detectors and thus, their inherent sensitivity to noise artifacts, boundary detection algorithms inevitably produce false and disconnected contours (excepted in [2]). By this fact, and contrary to the result given by a region-based segmentation method, the resulting soft boundary detection map does not generally exhibit, for a given threshold, a set of closed curves corresponding to the boundaries of a segmentation into regions and consequently, this resulting soft edge map often remains more difficult to exploit in a high level image analysis system compared to a classical region map.

In order to find a reliable segmentation, some attention has been given to associating/combining a region and edge based segmentation approach or equivalently to proposing an image segmentation model fusing contour or gradient features with color/texture cues. In this attempt, we can cite the Data-Driven

☆ This paper has been recommended for acceptance by C.S. Li.
E-mail address: mignotte@iro.umontreal.ca
URL: http://www.iro.umontreal.ca/~mignotte/

1077-3142/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2012.05.002

Markov Chain Monte Carlo (DDMCMC) paradigm proposed by [47] in order to combine (possibly) different edge detectors and region-based clustering methods in order to guide the Markov chain search and to select a set of important segmentation solutions at multiple scales of details. We can also cite the hierarchical segmentation method proposed in [2] which transforms the output of any contour detector (and especially the so-called *gPb* contour detector [1] combining local gradient and color/texture cues) into a hierarchical region tree structure represented as an ultrametric contour map (UCM) and producing a set of closed curves for any threshold (and thus also encoding a set of segmentation maps at different scales). Extensive experimental results reported in [2] shows that this latter segmentation algorithm is currently the best to date both in terms of quality of segmented regions and detected contours. More precisely, in terms of two reliable and complementary performance metrics, namely the Probabilistic Rand Index (PRI) [44] and the *F*-measure [28] which appear to be fully effective for evaluating both a region-based segmentation and the quality of contours produced by this segmentation map (comparatively to multiple, manually generated, ground-truth segmentations obtained from human subjects). In fact, the PRI score measures in percentage the number of pairs of pixel labels correctly classified in the segmentation results and the *F*-measure provides a precision score, evaluating the agreement between region boundaries of the machine segmentation and the ground-truth segmentation. This latter measure is, in fact, deduced from the well-known precision/recall values that characterizes, in this case, respectively the fraction of detections that are true boundaries and the fraction of true boundaries detected.

The segmentation approach, proposed in this paper, further explores the MDS-based de-texturing pre-processing step initially presented in [35]. Concretely, this de-texturing step aims at converting the original texture image, in which each pixel is associated with a (local) *D*-dimensional feature vector, in a low (three) dimensional space, i.e., into a noisy color image, thus simplifying the segmentation step (see Fig. 1). In the proposed model, our contribution includes both the integration of contour cues in this region-based dimensionality reduction and clustering procedures, thus allowing to further simplify the segmentation process by efficiently "regularizing" the de-textured color image and finally to greatly improve the segmentation result. We will see that the proposed strategy appears as an interesting alternative to efficiently combine region and edge cues in a hybrid segmentation model. In addition, we also propose a complexity measure which allows us to adaptively set, for each natural image to be segmented, the different parameters of our two-step (de-texturing and clustering) segmentation model. The experimental evaluation demonstrates that our model provides a powerful segmentation framework which appears as an interesting alternative to complex models existing in the literature for the difficult image segmentation problem.

2. Proposed model

As starting point for our segmentation model, we consider, in a first step, the gradient and color/texture feature extraction step proposed in [35] (see Section 2.1) and locally computed on overlapping squared small windows (centered around the pixel to be classified). This yields, for each pixel, to a *D*-dimensional feature vector whose dimension will be non-linearly reduced, in a second step, to three dimensions (3D) by an improved version of the non-linear MDS-based dimensionality reduction technique introduced in [35] (see Section 2.2). This non-linear dimension reduction step can also be viewed as a de-texturing approach which converts the original texture image into a (low) 3D representation of the local color and gradient value distribution of the texture regions. Concretely speaking, this latter step aims at converting the input textured image into a (noisy) color image, without texture, that will be drastically easier to segment (see Fig. 1).

2.1. Texture feature extraction step

In order to validate our segmentation model, we use as texture features (to characterize each textured region) and to be represented in a lower dimensional space:

Fig. 1. From left to right: original Berkeley images (n^0 105019 and n^0 134008 and n^0 134052) and de-textured related images i.e., low (three)-dimensional representation (as a color image) of the local color and gradient value distribution of the texture regions. Second column: our MDS-based de-texturing method integrating contour and texture cues. Third column: MDS-based de-texturing method using only texture cues as proposed in [35].

1. First, the set of values of the local color non-normalized histogram and estimated around the pixel to be classified for an input image expressed in the LAB color space. In our application, this local histogram is equally re-quantized with q_c equidistant binnings for each of the A and B color channels, and $(q_c - 1)$ equidistant binnings for the L channel (in order to be somewhat invariant to shadows effects) in a final $N_b = (q_c - 1) \times q_c^2$ bin descriptor, computed on an overlapping squared fixed-size (N_w) neighborhood centered around the pixel to be classified.
2. Second, the four sets of (respectively vertical, horizontal, right diagonal and left diagonal) N_g equidistant bin values in the interval $[0{:}G_{\max}]$ of the non-normalized local histogram of the gradient magnitude (i.e., the absolute value of the first order difference) computed on the luminance component of each pixel contained in an overlapping N_w-squared fixed-size neighborhood (centered around the pixel to be classified).

In this simpler feature extraction model, a *texton* is thus herein characterized by the values of the re-quantized color and gradient magnitude histograms in the four directions (see [35] for more details), and thus yields to a $D = [N_b + 4N_g] = [(q_c - 1)q_c^2 + 4N_g]$-dimensional feature vector (for each pixel to be segmented). In our application, we take $q_c = 4$, $N_g = 10$ and $N_w = 7$.

Let us note that N_w must be large enough to efficiently model the *texton* feature but should also be not too large in order not to affect (too much) the accuracy of the boundary estimation between distinct textured regions. A good compromise, between good classification of the segmentation and contour accuracy, seems to be the value $N_w = 7$. Some tests have shown that $N_w = 5$ and $N_w = 9$ give slightly similar performance results and a size value greater than $N_w = 11$ affects the accuracy of the boundary location. This observation was also noticed in [34].

It is also important to note that q_c and N_g, respectively the number of bins of each color channel and the number of bins of the gradient magnitude histogram, should not be too large, in order to avoid a resulting over-segmentation map. Indeed, the more the number of bins is small and the more the number of similar texture regions is important in the Bhattacharyya distance sense (see Eq. (2)). Conversely, a large number of bins will favor different region labels. In our application, we have empirically chosen these two parameters after some trial and error.

$G_{\max}$, the maximal value of the gradient histogram, is set to $G_{\max} = 10$ (thus, in our application, a gradient magnitude value greater than 10 thus fall into the last bin).

2.2. MDS-based dimensionality reduction step

As proposed in [35], we separately reduce the dimensionality of the color and the gradient magnitude feature vectors, since these two texture clues are not very interrelated. This strategy allows the MDS algorithm to (computationally) more easily find a non-linear manifold (without altering the accuracy of the final results). Moreover, since the color features seem more important than the gradient magnitude clues to characterize a texture region, we also use twice as much weight by searching for them a non-linear manifold with two times more dimensions. Finally, we construct a low-dimensional representation with three dimensions[1]; this allows us also to visualize this low-dimensional representation as a three-channel (RGB or color) image. In this context, we have therefore a 2D space for the texture/color based feature vector and a 1D space for the gradient magnitude based feature vector.

Mathematically speaking, we consider the set of texture features of an input color image (with N pixels) as a data cube $\mathscr{I}(\mathbf{s}, k)$ or a 3D array, where $\mathbf{s}$ indicates the spatial location $\mathbf{s}$ = (row, column) and $h_{\mathbf{s}}(\cdot) \in IR^D$ is the texture feature vector, indexed by k, at location $\mathbf{s}$. D is the dimensionality of the original texture feature vectors (i.e., the high dimensional space) and $d = 3$, the dimensionality of the target low-dimensional representation. In this context, the MDS approach, which attempts to find an embedding from the N initial feature vectors in the high dimensional space ($d = D$) such that distances are faithfully preserved in the low dimensional ($d = 3$) target space, consists of finding $\hat{\mathbf{u}} = (\widehat{R}\,\widehat{G})^t$ a 2D vector of mappings and $\hat{\mathbf{v}} = (\widehat{B})^t$ a 1D vector solution of the two (independent and) objective *stress* functions to be minimized

$$\begin{cases} \hat{\mathbf{u}} = \arg\min\limits_{\mathbf{u}} \sum\limits_{\mathbf{s},\mathbf{t}_{\mathbf{s}\neq\mathbf{t}}} \left(w_{\mathbf{s},\mathbf{t}} \beta_{\mathbf{s},\mathbf{t}}^{[0:N_b[} - \|\mathbf{u}_{\mathbf{s}} - \mathbf{u}_{\mathbf{t}}\|_2^2 \right)^2 \\ \hat{\mathbf{v}} = \arg\min\limits_{\mathbf{v}} \sum\limits_{\mathbf{s},\mathbf{t}_{\mathbf{s}\neq\mathbf{t}}} \left(w_{\mathbf{s},\mathbf{t}} \beta_{\mathbf{s},\mathbf{t}}^{]N_b:D]} - (\mathbf{v}_{\mathbf{s}} - \mathbf{v}_{\mathbf{t}})^2 \right)^2 \end{cases} \tag{1}$$

where the summation $\sum_{\mathbf{s},\mathbf{t}_{\mathbf{s}\neq\mathbf{t}}}$ is over all pairs of sites (i.e., for all sites $\mathbf{s}$ and for all pairs of sites including $\mathbf{s}$) existing in the image. $\beta_{\mathbf{s},\mathbf{t}}^{[k_0-k_1]}$ denotes the squared Bhattacharyya distance between the pair of feature vectors $h_{\mathbf{s}}(k)$ and $h_{\mathbf{t}}(k)$ (at pixels locations $\mathbf{s}$ and $\mathbf{t}$ and $k \in [k_0{:}k_1]$):

$$\beta_{\mathbf{s},\mathbf{t}}^{\mathscr{I}} = D_{\mathscr{B}}\left[h_{\mathbf{s}}^{\star}, h_{\mathbf{t}}^{\star}\right] = 1 - \sum_{k\in\mathscr{I}} \sqrt{h_{\mathbf{s}}^{\star}(k) h_{\mathbf{t}}^{\star}(k)} \tag{2}$$

In this histogram-based similarity measure (derived from the Bhattacharyya similarity coefficient and whose measures range from 0 to 1), $h_{\mathbf{s}}^{\star}(\cdot)$ denotes the normalized (integrating to 1) texture feature vector or the normalized (color/texture or gradient) histogram related to each pixel. After minimization, the three mapping results ($\widehat{R}\ \widehat{G}\ \widehat{B}$) can be seen as a (noisy) de-textured color image (see Fig. 1).

In [35], $w_{\mathbf{s},\mathbf{t}}$ was a factor equal to 0.5 for sites $\mathbf{t}$ belonging to the first neighborhood (i.e., four nearest neighbors) of $\mathbf{s}$ and equal to 1.0 elsewhere. The goal of this was to include, in this dimension reduction step, a prior on the (de-textured color image) solution favoring homogeneity between neighboring sites and to somewhat constraint the low-dimensional mapping to exhibit homogeneous regions (and thus allowing to simplify the subsequent segmentation procedure). In order to make this de-textured image even easier to subsequently segment, we have decided to make this factor depending on the result of a coarse edge-based segmentation. More precisely, a soft edge map (in the textural sense) is preliminary computed. One possibility, which we exploit here, is the simple texture gradient based edge map proposed in [34] (see Fig. 2). In order now to favor homogeneity (for the color value) of the sites $\mathbf{s}$ and $\mathbf{t}$ when these ones are not separated by an edge in the soft edge map and *vice versa*, we have used, as empirical choice for the estimator of $w_{\mathbf{s},\mathbf{t}}$:

$$w_{\mathbf{s},\mathbf{t}} = 2\Psi(\mathbf{s},\mathbf{t}) \tag{3}$$

where $\Psi(\mathbf{s},\mathbf{t})$ denotes the maximal contour potential found on the straight line existing between the sites $\mathbf{s}$ and $\mathbf{t}$ in the (soft) edge map plane (see Fig. 2) normalized between [01]. With this additional constraint, two feature vectors will be even further away (in distance) in the (3D low dimensional) target space if they are separated by an edge (thus penalizing two same colors for two regions separated by an edge). Inversely, two feature vectors will be even closer (in distance) in the target space if they are not separated by an edge (thus favoring an homogeneous region). This allow us to easily include a global prior constraint, based on contour cues,

[1] Experimental tests have shown that a higher dimension representation does not provide more information but, on the other hand, may affect the efficiency of the clustering process. Conversely, a representation with less than three dimension results in significant loss of information in the non-linear dimensionality reduction method and thus in the accuracy of the segmentation result.

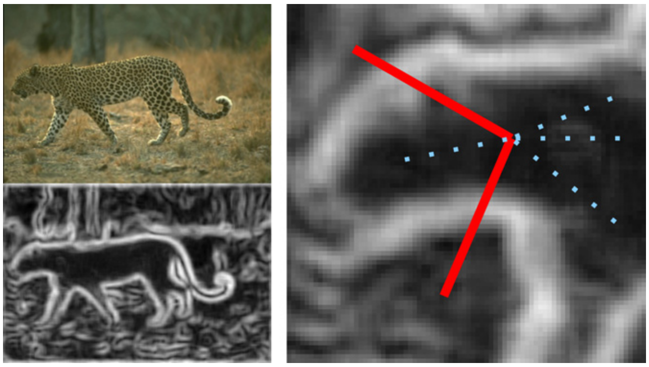

Fig. 2. From top to bottom and left to right; a natural image (number 134052) from the Berkeley database, its soft edge map and six pairs of sites represented by six line segments whose the size is proportional to the value of the edge potential crossed by the line segment.

within our energy-based dimensionality reduction model exploiting pairwise (texture) region-based distances. In addition, as additional region-based regularization effect, and in order to take into account that the soft edge map or the input texture image may be noisy, we have added the following hard thresholding or constraint:

$$\text{if } w_{\mathbf{s},\mathbf{t}}\beta_{\mathbf{s},\mathbf{t}} < T_{Bh}(=0.1) \text{ then } w_{\mathbf{s},\mathbf{t}}\beta_{\mathbf{s},\mathbf{t}} = 0.0 \tag{4}$$

Fig. 1 shows three original Berkeley images and their de-textured representation (as a color image), with and without contour cues, and thus showing the "regularizing" effect of the non-local constraints based on contour cues (which will greatly simplify the subsequent segmentation problem and consequently to increase its robustness).

Finally in order to minimize our energy-based dimensionality reduction model (see Eqs. (1)–(4)), we use the optimization strategy described in [35].

To this end (and qualitatively explained), we rely on a coarse-to-fine minimization method. The goal of this multiresolution optimization strategy is first to estimate (at coarser resolution level) an approximate (and hence computationally simpler) solution of the original minimization problem and then exploits this coarser estimate to obtain (*via* an interpolation scheme) a good initial guess that guides and accelerates the minimization process of the following finer resolution level. At the coarser level, the solution is initialized at random. At each resolution level, a conjugate gradient descent procedure is then first used and, after convergence, the solution is then refined by a stochastic local exploration around the current solution using the Metropolis criteria and a low radius of exploration. For further details, see [35].

2.3. Segmentation step

The above-presented MDS-based de-texturing approach, greatly simplifies our segmentation problem (see Fig. 1) for which we have decided to use a simple spatially constrained *K*-means clustering procedure. In this clustering process, we have used as input multi-dimensional feature descriptor, the set of (color) values estimated on an overlapping squared fixed-size ($N_{km} = 5$) neighborhood centered around each pixel (of our de-textured image) to be classified

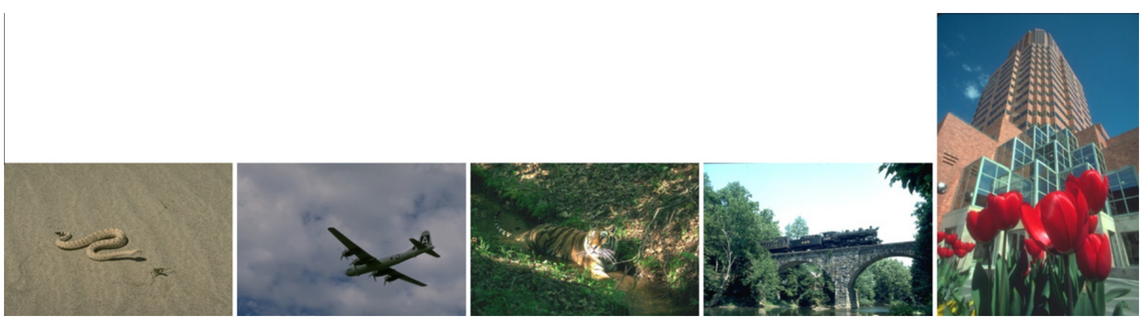

Fig. 3. Examples of complexity values on some images of the Berkeley database. From left to right, complexity value = 0.235, 0.364, 0.515, 0.661, 0.803 setting the number of classes (*K*) of the *K*-mean clustering algorithm respectively to 2, 4, 5, 7 and 8 in our segmentation model.

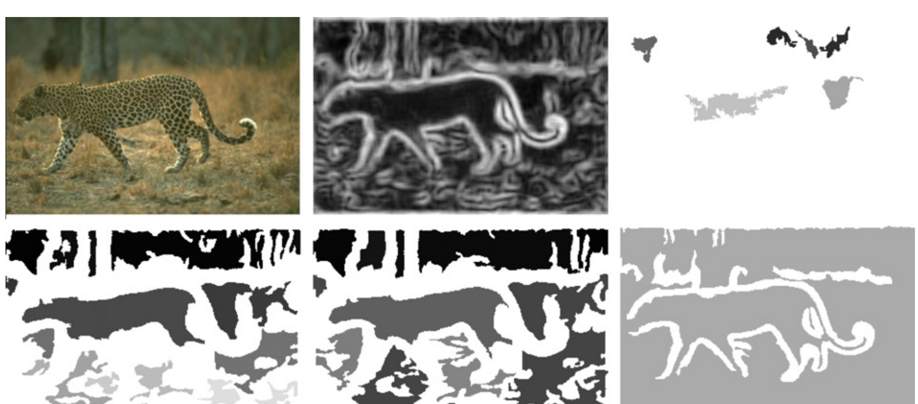

Fig. 4. From top to bottom and left to right; a natural image from the Berkeley database (n^0 134052), its soft edge map and the sets of connected pixels (i.e., regions) whose edge potential is respectively below $\xi = 0.1$ (7 homogeneous regions), $\xi = 0.2$ (18 homogeneous regions), $\xi = 0.3$ (12 homogeneous regions) and $\xi = 0.4$ (1 homogeneous region). The white region corresponds to the sets of pixels whose edge potential is above the threshold ξ (and thus corresponding to inhomogeneous regions in the textural sense). By selecting the segmentation map exhibiting the maximum number of regions (in this example; the map related to $\xi = 0.2$), this allows, almost surely, to avoid a merging of different textural regions during the spatially constrained *K*-means algorithm.

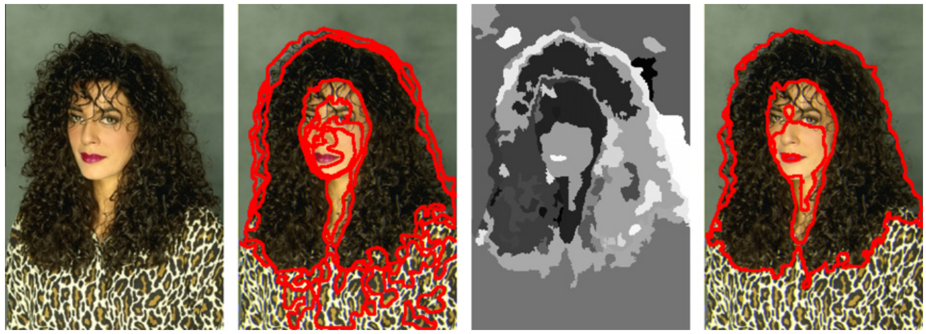

Fig. 5. From left to right; a natural image from the Berkeley database (n^0 198054), its segmentation before the refinement step, its texture-based over-segmentation and the final result after the refinement step.

(this strategy allows us to "regularize" a bit the K-means procedure). In addition, K (the number of classes of the K-means) is herein adaptively set for each image by a metric measuring the complexity, in terms of number of different texture types, of a natural color image (see the following section). The spatial constraints applied on the K-mean clustering process is explained in Section 2.3.2.

2.3.1. Image complexity

This measure will allow us to adaptively set the number of classes of the K-mean algorithm. This metric is herein defined as the measure of the absolute deviation (L_1 norm) of the set of normalized histograms or feature vectors (already computed in Section 2.1) for each overlapping squared fixed-size (N_w) neighborhood contained within the input image. This measure ranges in [01] and an image with several different texture types will result in value of complexity close to 1 (see Fig. 3). In our application,

$$K = \text{ceil}(K_{\max} \times \text{complexity value}) \tag{5}$$

where ceil(x) is a function that rounds x up to the nearest integer and $K_{\max} = 10$ classes in our application since this number seems to be an upper-bound of the number of classes for a very complex natural image (with a minimum of 2 classes).

2.3.2. Spatially constrained K-means

As proposed in [35], to further help the K-means clustering process to succeed in finding an accurate partition, a simple hard constraint enforcing the spatial continuity of each (likely) region is imposed during the iterative K-means labeling process. To this end, we preliminary compute an edge (gradient magnitude) map from our de-textured image. The most likely regions in this edge map (see Fig. 4) are easily estimated by identifying the sets of connected pixels whose edge potential is below a given threshold ξ thus defining a map of likely homogeneous regions. The hard constraint enforcing the spatial continuity of each of the K-means cluster is then simply performed by assigning the majority class label in each pre-defined homogeneous textural region, for each iteration of the K-means algorithm (and this constraint is applied after convergence of the classical K-means clustering). Contrary to [35], and in order to obtain better results by making this procedure somewhat adaptive for each image, we set, (for each edge map) the value of ξ that ensures the maximum number of homogeneous regions in this edge map (see Fig. 4).

Finally, a merging step is added to each segmentation map that simply consists of fusing each small region (i.e., regions whose size is below $Szm = 100$ pixels) with the region sharing its longest boundary. This merging step allows to "regularize" a bit the

Table 1
Average performance, in term of PRI measure, of several segmentation (into regions) algorithms (ranked according to their maximum PRI score) on the BSD300. Some of these segmentation schemes have been directly applied on the BSD300 without training strategy on the training Berkeley dataset. These segmentation methods are indicated by an asterisk.

ALGORITHMS	**PRI**[44]
HUMANS (in [46])	0.875
MDSCCT$_{[T_{Bh}=0.1\|K_{\max}=10]}$	**0.811**
–2011– gPb-owt-ucm [2]	0.810
–2010– PRIF [33]	0.801
–2008– *CTex [22]	0.800
–2009– MIS [23]	0.798
–2011– SCKM [34]	0.796
–2008– FCR [32]	0.788
–2004– *FH [16] (in[46])	0.784
–2011– MD2S [35]	0.784
–2009– HMC [39]	0.783
–2009– *Consensus [20]	0.781
–2009– *Total Var. [15]	0.776
–2009– *A-IFS HRI [37]	0.771
–2001– *JSEG [11] (in[22])	0.770
–2011– *KM [40]	0.765
–2007– *CTM [46,26]	0.762
–2006– *Av. Diss. [3] (in[2])	0.760
–2008– *St-SVGMM [41]	0.759
–2011– *SCL [21]	0.758
–2005– *Mscuts [10] (in[15])	0.756
–2003– *Mean-Shift [9] (in[46])	0.755
–2008– *NTP [45]	0.752
–2010– *iHMRF [7]	0.752
–2005– *NCuts [10] (in[2])	0.750
–2006– *SWA [42] (in[2])	0.750
–2006– *GBMS [6] (in[38])	0.734
–2000– *NCuts [43] (in[46])	0.722
–2010– *JND [4]	0.719
–2010– *DCM [38]	0.708
–2009– *[30]	0.703

Table 2
Average performance, in term of F-measure, of several segmentation (into contours) algorithms and contour detectors (in parentheses) on the BSD300. The score obtained by our algorithm is indicated in bold.

ALGORITHMS	F-measure [29]
HUMANS (in [46])	0.79
MDSCCT	**0.63**
–2011– gPb-owt-ucm [2]	0.71
–2010– PRIF [33]	0.64
–2003– Mean-Shift [9] (in [2])	0.63
–2000– NCuts [10] (in [2])	0.62
–2007– CTM [46] (in [2])	0.58
–2004– FH [16] (in [2])	0.58
–1986– [5] (in [2])	0.58
–2006– SWA [42] (in [2])	0.56
–2008– FCR [32]	0.56
Quad-Tree (in [2])	0.37

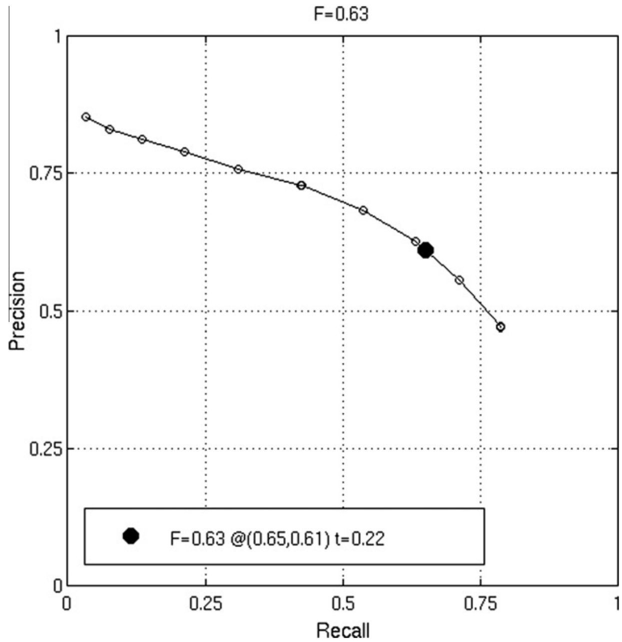

Fig. 6. Evaluation of our algorithm in terms of *F*-measure on the BSD300 benchmark in the precision–recall framework (see also Table 2 for comparison with other algorithms).

Table 3
Average performance of our algorithm for different performance measures (lower is better) on the BSD300. The score obtained by our algorithm and the best score to date are indicated in bold.

ALGORITHMS	VoI	GCE	BDE
HUMANS	1.10	0.079	4.99
MDSCCT	**2.004**	**0.205**	**7.951**
PRIF [33]	**1.970**	0.209	8.446
SCKM [34]	2.114	0.230	10.090
MD2S [35]	2.361	0.235	10.368
FCR [32]	2.304	0.211	8.995
CTM [46,26]	2.024	**0.188**	9.896
Mean-Shift [9](in[46])	2.477	0.260	9.700
NCuts [43](in[46])	2.933	0.218	9.604
FH [16](in[46])	2.665	0.189	9.950

K-means procedure and to get a slightly better PRI score (see Section 3).

2.4. Refinement step

Finally, in order to increase the contour accuracy of our segmentation map, we now exploit an over-segmentation obtained by a *K*-means clustering technique, applied on the input image and using as input multidimensional feature descriptor, the color/texture cues already computed computed in Section 2.1 and with $K = K_{\max}$ (the upper-bound number of classes already used in Eq. (5)). In order to succeed in finding a segmentation result with accurate contours, a hard constraint is finally imposed to our segmentation result, by simply assigning (as in Section 2.3.2) the majority region label in each homogeneous textural region defined by this over-segmentation (see Fig. 5).

3. Experimental Results

3.1. Setup

In these experiments, we have tested our segmentation algorithms on the Berkeley segmentation database (BSD300) [29] for which the color images are normalized to have the longest side equal to 320 pixels. The segmentation results are then supersampled in order to obtain segmentation images with the original resolution (481×321) before the estimation of the performance metrics. In order to ensure the integrity of the evaluation, the internal parameters of our segmentation algorithm (called MDSCCT), namely $T_{Bh} = 0.1$ (Eq. (4)), $K_{\max} = 10$ (Eq. (5) and Section 2.4) are tuned (according to the PRI measure) on the train image set by doing a local discrete grid search routine, with a fixed step-size, on the parameter space and in the feasible ranges of parameter values (namely $T_{Bh} \in [0–0.5]$ [step-size: 0.05], $K_{\max} \in [5–20]$ [step-size: 1].

3.2. Results and discussion

Two performance metrics have been computed, namely the PRI [44] result, given for the entire image database for comparison with the other segmenters and the *F*-measure [28] (see Tables 1 and 2). For comparison, we illustrate the results of our segmentation algorithm by showing the same segmented images (see Figs. 8 and 9) as those shown in the papers [33,35]. The results for the entire database are available on-line at http address http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/mdscct. It may be noted that our segmentation procedures give the best PRI score among the state-of-the-art segmentation methods recently proposed in the literature (a score equal to PRI = 0.811, for example, simply means that, on average, 81.1% of pairs of pixel labels are correctly classified in the BSD300 segmentation results), even with a final simple clustering scheme based on a simple *K*-means algorithm and a simple texture feature extraction step based on the values of the local re-quantized texture/color and gradient distributions (Fig. 7 shows respectively the distribution of the PRI measure and the number and size of regions obtained by our algorithm $\text{MDSCCT}_{[T_{Bh}=0.1|K_{\max}=10]}$ over the BSD300).

The PRI performance measure, for only the testing set, and thus without fear of the possible over-fitting problem, is PRI = 0.792, a slightly lower score of only 2% compared to the score obtained on the whole dataset. This is often the case for all segmenters, because the testing set contains more difficult images to segment. By comparison, our algorithm without integrating contour cues, i.e.,

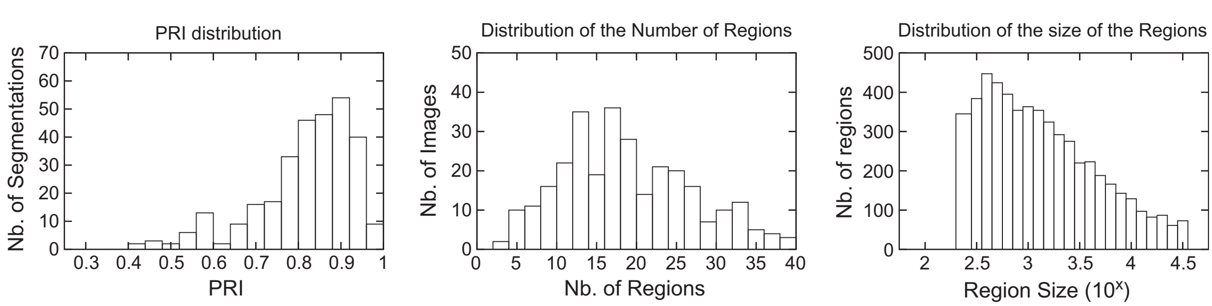

Fig. 7. From left to right, distribution of the −1 – PRI measure −2 – number and −3 – size of regions over the 300 segmented images of the Berkeley image database.

the MD2S algorithm (see [35] and Table 1) allows to obtain PRI = 0.784 on the entire database and also a slightly lower (of about 2%) score of PRI = 0.768 on the testing set. Similarly, the performance score of the PRIF segmentation algorithm (see [33] and Table 1) is PRI = 0.789 and PRI = 0.801, respectively on the testing set and on the entire database.

We have also tested the performance, in term of *F*-measure (see Table 2 and Fig. 6). First, we should remember that this measure is better appropriate for contour detection methods giving a ‘‘soft'' edge map since this benchmark measure is able to find, from a soft edge map, the optimal threshold value ensuring the best *F*-measure [27] over the BSD300. Contrary to the PRI performance measure,

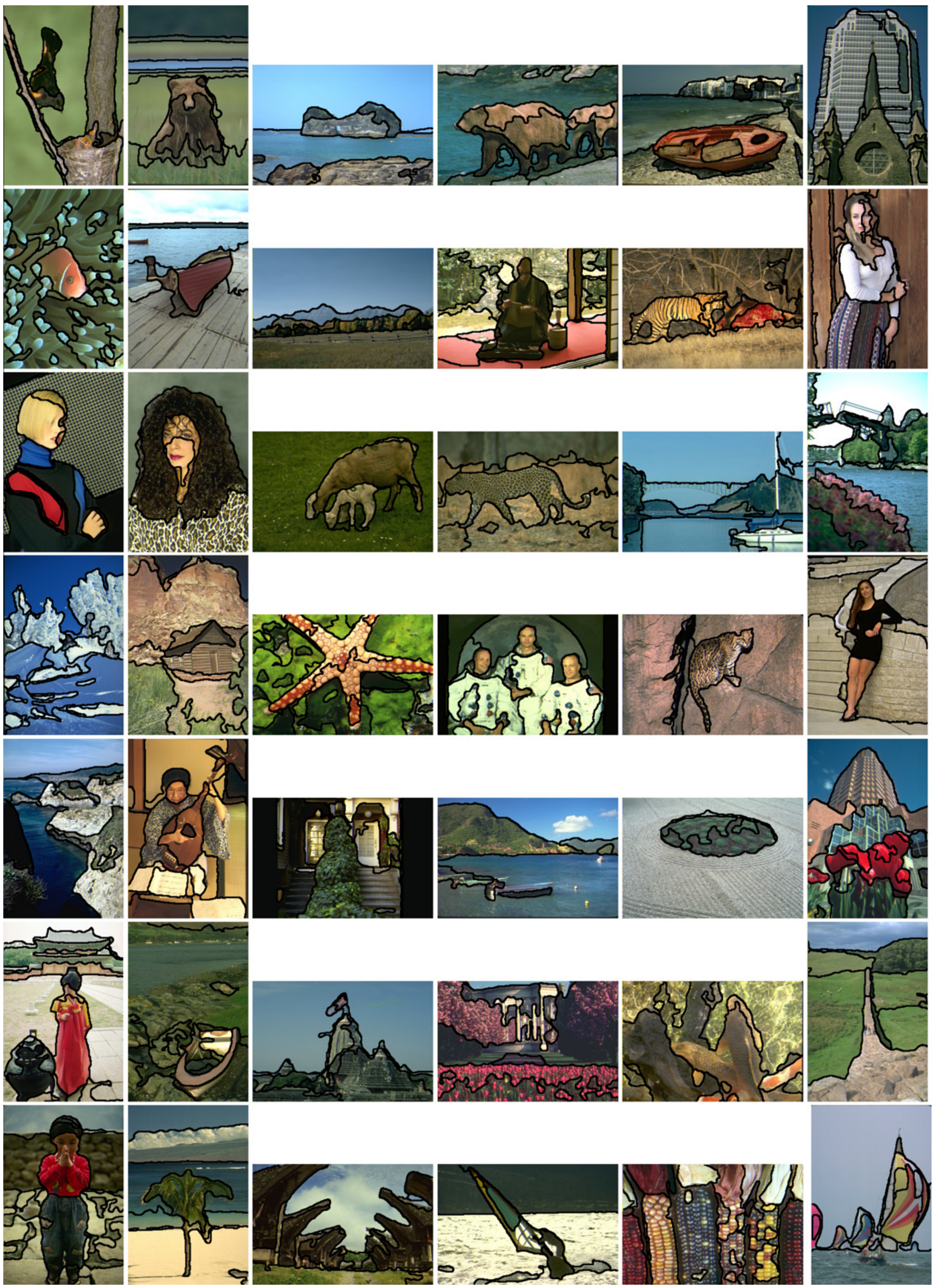

Fig. 8. Example of segmentations obtained by our MDSCCT$_{[T_{Bh}=0.1|K_{\max}=10]}$ algorithm on several images of the Berkeley image database (see also Tables 1 and 2 for quantitative performance measures and http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/mdscct.html for the segmentation results on the entire database).

for which we have noticed that this ''best threshold'' expressed in terms of $K_{\max}$ and T_{Bh} (i.e., the two internal parameters of the algorithm) is thus $K_{\max} = 10|T_{Bh} = 0.1$ (see Table 1). In term of F performance measure, we will let the algorithm in choosing the optimal threshold and we will provide a soft contour map, by simply averaging the set of ''hard'' boundary representations obtained by our segmentation method with $K_{\max}$, the number of classes of the segmentation step, varying in a interval containing an upper and lower bound of the number of classes, e.g., $K_{\max} \in [6{:}16]$ (i.e., an average of 10 different boundary representations obtained by ten different clusterings with ten different values of the number of classes). We have obtained $F = 0.63@(R = 0.65,\ P = 0.61)$ for the BSD300, which is very competitive compared to the state-of-the art existing segmentation methods and not too far to the highest score to date, namely $F = 0.71$ [2]).

We have also compared our segmentation method with three other region-based performance metrics, namely; VoI [31], GCE [29], BDE [18] (for which a lower distance is better), showing that

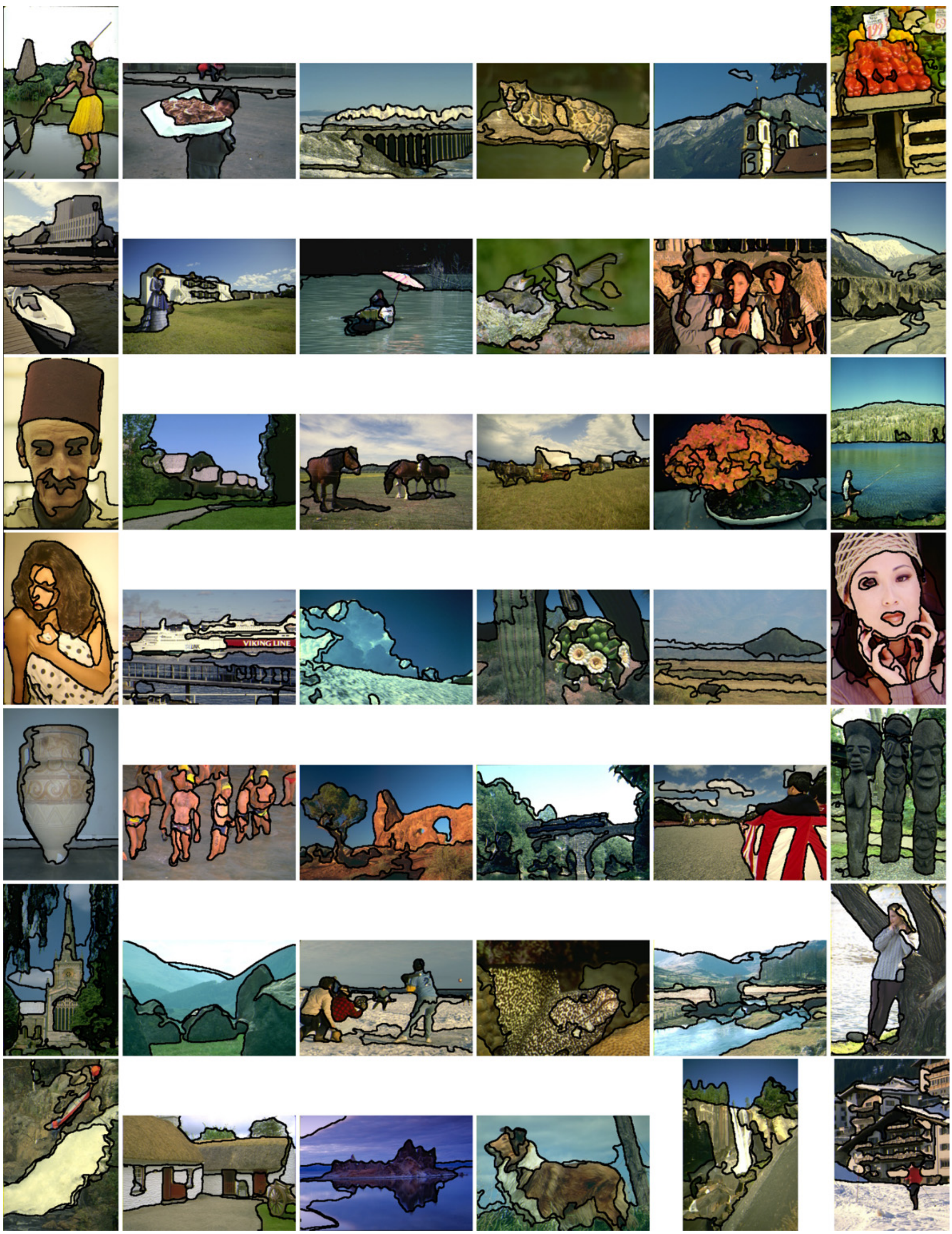

Fig. 9. Example of segmentations obtained by our MDSCCT$_{|T_{Bh}=0.1|K_{\max}=10|}$ algorithm on several images of the Berkeley image database (see also Tables 1 and 2 for quantitative performance measures and http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/mdscct.html for the segmentation results on the entire database).

our method gives competitive results for some other metrics based on different criteria and compared to state-of-the arts (see Table 3). The qualitative meaning of these three metrics are outlined below.

1. The Variation of Information (VoI) metric [31] is based on relationship between a point and its cluster. It uses mutual information metric and entropy to approximate the distance between two clusterings across the lattice of possible clusterings.
2. The Global Consistency measure (GCE) [29] measures the extent to which one segmentation map can be viewed as a refinement of another segmentation. For a perfect match (in this metric sense) every region in one of the segmentation must be identical to, or a refinement (i.e., a subset) of a region in the other segmentation. Segmentation which are related in this manner are considered to be consistent, since they could represent the same natural image segmented at different levels of detail (as the segmented images produced by several human observers for which a finer level of detail will merge in such a way that they yield the larger regions proposed by a different observer at a coarser level).
3. The Boundary Displacement Error (BDE) [18] measures the average displacement error of one boundary pixels and the closest boundary pixels in the other segmentation.

The experiments show us that our segmentation model tends to slightly over-segment for some images containing regions[2] with large texture elements or sometimes to merge a part of an animal (with its texture camouflage) with its natural environment (or more generally a textured region with the same average color of its background). This is also shown by the large number of regions for certain images (see Fig. 7b) and also by the average number of regions per image, obtained by our method, and equals to 19 on this BSD300. Consequently, our segmentation procedure should provide better results if a final grouping at region level (including or not an *a priori* on the number and/or the shape of these segmented regions) would be used as post-treatment.

In order to see which step of our algorithm has contributed most to the final improvements, we have tested the influence and the efficiency of the refinement step (see Section 2.4) on the final PRI score. Without the final refinement step, we obtain a PRI equals to 0.802, a lower score of 1.1% (in terms of pairs of pixel labels correctly classified in the segmentation result) compared to our final score of PRI = 0.811. In addition, we have also tested the influence and the efficiency of the merging step (see Section 2.3.2) on the final PRI score. Without the final merging step (or equivalently with $Szm = 0$), we obtain a PRI equals to 0.808, a lower score of 0.4% compared to our final score of PRI = 0.811. Finally, without the final refinement step and without the merging step, we obtain a PRI equals to 0.790.

It is worth mentioning that, since our MDS-based data/fusion reduction model (used as pre-processing step for a final segmentation step) essentially exploits the notion of pair of (texture/gradient based) feature vectors (and not pair of class-labels) existing in the image; high level knowledge about the semantic meaning of the objects that compose the image (e.g., semantic labels or cooccurence or domain-specific knowledge between labels) thus cannot be integrated in our approach. Nevertheless, our MDS-based de-texturing approach can easily include mid-level features such as (reflection, bilateral, etc.) symmetry cues (of objects present in the image) which can significantly help the segmentation task. This can be done, for example, by favoring homogeneity (with a low $w_{s,t}$ value, see Eq. (1)) for pair of sites (s, t) sharing the same (preliminarily detected) axis of symmetry. In addition, a prior knowledge about the size of the segmented regions can be integrated in our model. For example, in order to limit the size of the segmented regions, we could weight $w_{s,t}$ inversely proportionally to the length of the line segment existing between the sites **s** and **t** (thus favoring a class inhomogeneity between distant pairs of sites).

3.3. Comparison of the MDSCCT versus MD2S algorithm

We can easily compare the segmentation results given by our algorithm compared to its simple version, without contour cues [35], called MD2S, since we herein show the same set of segmented images in Figs. 8 and 9. In addition, let us recall that the results for the entire database can be visually compared with the segmentation result given by the MD2S algorithm at http address: http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/.

It is also interesting to compare the average number of regions per image obtained by our method (19 regions on average) compared to the MD2S segmentation method for which we obtain 21 regions on this BSD300. This also demonstrates, to a certain extend, the "regularizing" effect of the non-local constraints based on contour cues. Table 3 also shows us that the BDE error (among other performance measures) is significantly lower in our model compared to the MD2S model which is certainly due both to our refinement step and the "regularizing" effect of the contour cues based constraints. We can also add that the worst PRI score obtained by our segmentation model is PRI = 0.396 (image number 210088). This score is significantly higher than the worst PRI score obtained by the MD2S segmentation method for which we obtain PRI = 0.313 (image number 130034). This also demonstrates that the take into account of contour cues is also useful to solve some very difficult segmentation problems. In addition, the standard deviation of the PRI performance measures is lower for our algorithm ($\sigma_{\text{PRI}} = 0.115$) compared to the MD2S segmentation method (for which we obtain $\sigma_{\text{PRI}} = 0.117$).

3.4. Algorithm

The segmentation procedure takes about one minute (on average; 50 s for the dimensionality reduction and approximately 10 s for the clustering per image) for an $i7-930$ Intel CPU, 2.8 GHz, 5611 bogomips and for a non-optimized code running on Linux. Source code (in C++ language) of our algorithm (with the set of segmented images and detected contours) are publicly available at the following http address http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/mdscct.html in order to make possible comparisons with future segmentation algorithms or different performance measures.

4. Conclusion

In this paper, we have presented a new and efficient strategy for the integration of inhomogeneous texture and edge-based cues in a segmentation problem. The proposed scheme relies mainly on an original spatially constrained non-linear dimensionality reduction technique able to efficiently fuse region and edge cues in a data pre-processing step. This "de-texturing" process and the sets of color/texture and edge-based adaptive spatial continuity constraints allows us to simplify the image segmentation problem and consequently to increase its robustness. This segmentation framework provides competitive performance comparatively to the state-of-the-art existing methods. In addition, this method remains perfectible by using a more elaborated color clustering algorithm (with eventually a final grouping at region level as

[2] We recall that a region (or a segment) is a set of connected pixels belonging to the same class.

post-treatment) or a better soft edge map and general enough to be applied to various digital image and computer vision applications (e.g., hyperspectral imagery, motion detection, 3-D segmentation, etc.).

Acknowledgments

The author thanks the anonymous reviewers for their many valuable comments and suggestions that helped to improve both the technical content and the presentation quality of this paper.

References

[1] P. Arbelaez, L. Cohen, A metric approach to vector-valued image segmentation, Int. J. Comput. Vision, Spec. Issue Geometr., Variat. Level Sets Methods Comput. Vision 69 (2006) 119–126.
[2] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 898–916.
[3] L. Bertelli, B. Sumengen, B. Manjunath, F. Gibou, A variational framework for multi-region pairwise similarity-based image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 30 (8) (2008) 1400–1414.
[4] K.K. Bhoyar, O.G. Kakde, Color image segmentation based on jnd color histogram, Int. J. Image Process., 2010.
[5] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[6] M.A. Carreira-Perpinan, Fast nonparametric clustering with Gaussian blurring mean-shift, in: Proceedings of the International Conference on Machine Learning (ICML'06), 2006, pp. 153–160.
[7] S. Chatzis, G. Tsechpenakis, The infinite hidden Markov random field model, IEEE Trans. Neural Networks 21 (6) (2010) 1004–1014.
[8] S. Chen, L. Cao, Y. Wang, Image segmentation by ML-MAP estimations, IEEE Trans. Image Process. 19 (9) (2010) 2254–2264.
[9] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.
[10] T. Cour, F. Benezit, J. Shi, Spectral segmentation with multiscale graph decomposition, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2005.
[11] Y. Deng, B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, IEEE Trans. Pattern Anal. Mach. Intell. 23 (8) (2001) 800–810.
[12] F. Destrempes, J.-F. Angers, M. Mignotte, Fusion of hidden Markov Random Field models and its Bayesian estimation, IEEE Trans. Image Process. 15 (10) (2006) 2920–2935.
[13] F. Destrempes, M. Mignotte, A statistical model for contours in images, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 626–638.
[14] F. Destrempes, M. Mignotte, J.-F. Angers, A stochastic method for Bayesian estimation of hidden Markov models with application to a color model, IEEE Trans. Image Process. 14 (8) (2005) 1096–1108.
[15] M. Donoser, M. Urschler, M. Hirzer, H. Bishof, Saliency driven total variational segmentation, in: Proceedings of the IEEE Internatioanl Conference on Computer Vision (ICCV'09), 2009.
[16] P. Felzenszwalb, D. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vision 59 (2004) 167–181.
[17] W. Freeman, E. Adelson, The design and use of steerable filters, IEEE Trans. Pattern Anal. Mach. Intell. 13 (9) (1991) 891–906.
[18] J. Freixenet, X. Munoz, D. Raba, J. Marti, X. Cufi, 2002. Yet another survey on image segmentation: region and boundary information integration, in: Proceedings of 7th European Conference on Computer Vision ECCV02, 408 ff, p. III.
[19] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (6) (1984) 721–741.
[20] S. Ghosh, J.J. Pfeiffer, J. Mulligan, A general framework for reconciling multiple weak segmentations of an image, in: Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'09), Snowbird, UT, 2009.
[21] R. Huanga, N. Sangb, D. Luoc, Q. Tangd, Image segmentation via coherent clustering in $l^*a^*b^*$ color space, Pattern Recogn. Lett. 32 (7) (2011) 891–902.
[22] D.E. Ilea, P.F. Whelan, CTex – an adaptive unsupervised segmentation algorithm on color-texture coherence, IEEE Trans. Image Process. 17 (10) (2008) 1926–1939.
[23] M. Krninidis, I. Pitas, Color texture segmentation based on the modal energy of deformable surfaces, IEEE Trans. Image Process. 7 (18) (2009) 1613–1622.
[24] T. Lindeberg, Edge detection and ridge detection with automatic scale selection, Int. J. Comput. Vision 30 (1998) 117–156.
[25] Q. Luo, T. Khoshgoftaar, Unsupervised multiscale color image segmentation based on mdl principle, IEEE Trans. Image Process. 15 (2006) 2755–2761.
[26] Y. Ma, H. Derksen, W. Hong, J. Wright, Segmentation of multivariate mixed data via lossy coding and compression, IEEE Trans. Pattern Anal. Mach. Intell. 29 (9) (2007) 1546–1562.
[27] D. Martin, C. Fowlkes, The Berkeley segmentation database and benchmark, Image database and source code publicly available at address. <http//www.cs.berkeley.edu/projects/vision/grouping/segbench/>.
[28] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 530–549.
[29] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of 8th Internatioanl Conference on Computer Vision, vol. 2, 2001, pp. 416–423.
[30] I. Mecimore, C.D. Creusere, Unsupervised bitstream based segmentation of images, in: Proceedings of the Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop 2009, pp. 643–647, 2009.
[31] M. sMeila, Comparing clusterings – an axiomatic view, in: Proceedings of the 2005 22nd International Conference on Machine Learning, 2005, pp. 577–584.
[32] M. Mignotte, Segmentation by fusion of histogram-based *K*-means clusters in different color spaces, IEEE Trans. Image Process. 17 (2008) 780–787.
[33] M. Mignotte, A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation, IEEE Trans. Image Process. 19 (6) (2010) 1610–1624.
[34] M. Mignotte, A de-texturing and spatially constrained *K*-means approach for image segmentation, Pattern Recogn. Lett. 32 (2) (2011) 359–367.
[35] M. Mignotte, MDS-based multiresolution nonlinear dimensionality reduction model for color image segmentation, IEEE Trans. Neural Networks 22 (3) (2011) 447–460.
[36] M. Morrone, E. Adelson, Feature detection from local energy, Pattern Recogn. Lett. 6 (1987) 303–313.
[37] M.M. Mushrif, A.K. Ray, A-IFS histon based multithresholding algorithm for color image segmentation, IEEE Signal Process. Lett. 16 (3) (2009) 168–171.
[38] C. Nikou, A. Likas, N. Galatsanos, A Bayesian framework for images segmentation with spatially varying mixtures, IEEE Trans. Image Process. 19 (9) (2010) 2278–2289.
[39] H. Rachid, M. Mignotte, A hierarchical graph-based Markovian clustering approach for the unsupervised segmentation of textured color images, in: Proceedings of the IEEE International Conference on Image Processing, Cairo, Egypt, 2009, pp. 1365–1368.
[40] M.B. Salah, A. Mitiche, I.B. Ayed, Multiregion image segmentation by parametric kernel graph cuts, IEEE Trans. Image Process. 20 (2) (2011) 545–557.
[41] G. Sfikas, C. Nikou, N. Galatsanos, Edge preserving spatially varying mixtures for image segmentation, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, Anchorage, AK, USA, 2008, pp. 1–7.
[42] E. Sharon, M. Galun, D. Sharon, R. Basri, A. Brandt, Hierarchy and adaptivity in segmenting visual scenes, Nature 442 (2006) 810–813.
[43] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[44] R. Unnikrishnan, C. Pantofaru, M. Hebert, A measure for objective evaluation of image segmentation algorithms, in: Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '05), Workshop on Empirical Evaluation Methods in Computer Vision, vol. 3, 2005, pp. 34–41.
[45] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, L. Quan, Normalized tree partitionning for image segmentation, in: Proceedings of the IEEE International Conference on computer vision and pattern recognition, Anchorage, AK, USA, 2008, pp. 1–8.
[46] A.Y. Yang, J. Wright, S. Sastry, Y. Ma, Unsupervised segmentation of natural images via lossy data compression, Comput. Vision Image Understand. 110 (2) (2008) 212–225.
[47] S. Zhu, A. Yuille, Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1996) 884–900.