ISSN 1512-1232

EDGE DETECTION AND SEGMENTATION BY FUSION OF HISTOGRAM-BASED K-MEANS CLUSTERS IN DIFFERENT COLOR SPACES

Dr.G.Samuel Varaprasad Raju[1], G.Syam Prasad[2], Y.Srikanth[3]

[1] Professor in Scholl of Distance Education,AU,Vizag

[2]Associate Professor in CSE Dept,sriviveka institute of technology , Vijayawada

[3]B.Tech, CSE,sriviveka institute of technology , Vijayawada

ysrikanth557@yahoo.com

Abstract

This paper presents a new, simple, and efficient segmentation, edge detection approach, based on a fusion procedure which aims at combining several segmentation maps associated to simpler partition models in order to finally get a more reliable and accurate segmentation. End edge detection is one of the most commonly used operations in image analysis. The reason for this is that edges form the outline of an object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured. In this paper, we present methods for edge segmentation of satellite images; we used seven techniques for this category; Sobel operator technique, Prewitt technique, Kiresh technique, Laplacian technique, Canny technique, Roberts technique and Edge Maximization Technique (EMT) and they are compared with one another so as to choose the best technique for edge detection segment image. These techniques applied on one satellite images to choose base guesses for segmentation or edge detection image.

Index Terms: *Berkeley image database, color spaces, fusion of segmentations, -means clustering, textured image segmentation. image processing, edge detection, Euclidean distance, canny detector.*

1. INTRODUCTION

Image segmentation is a classic inverse problem which consists of achieving a compact region-based description of the image scene by decomposing it into meaningful or spatially coherent regions sharing similar attributes. This low-level vision task is often the preliminary (and also crucial) step in many video and computer vision applications, such as object localization or recognition, data compression, tracking, image retrieval, of understanding.

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation. This process detects outlines of an object and boundaries between objects and the background in the image. An edge-detection filter can also be used to improve the appearance of blurred image; to this cause more studies take this subject can be give some of these studies briefly: Soft computing techniques have found wide applications.

One of the most important applications is edge detection for image segmentation. The process of partitioning a digital image into multiple regions or sets of pixels is called image segmentation. Edge is a boundary between two homogeneous regions. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. In this paper, the main aim is to survey the theory of edge detection for image segmentation using soft computing approach based on the Fuzzy logic, Genetic Algorithm and Neural Network[1].The Canny algorithm uses an optimal\ edge detector based on a set of criteria which include finding the most edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and markin edges only once when a single edge exists for minimal response[2].The on maximal suppression stage identifies pixels that are local maxima in the direction of the gradient using the magnitude and orientation of the pixels. The major orientation of the gradient, either horizontal or vertical, is obtained by comparing the individual components, dx and dy, which are the result of convolving the smoothed image with the derivative of the Gaussian. Since most edges are at an angle, it is possible to obtain further granularity in the orientation of the gradient by comparing the sign bit of the gradient [3].The designed fuzzy rules are an attractive solution to improve the quality of edges as much as possible.

Because of its simplicity and efficiency, clustering approaches were one of the first techniques used for the segmentation of (textured) natural images [1]. Since different edge detectors work better under different conditions, it would be ideal to have an algorithm that makes use of multiple edge detectors, applying each one when the scene conditions are most ideal for its method of detection. In order to create this system, you must first know which edge detectors perform better under which conditions. That is the goal of our project. We tested four edge detectors that use different methods for detecting edges and compared their results under a variety of situations to determine which detector was preferable under different sets of conditions.

This data could then be used to create a multi-edge-detector system, which analyzes the scene and runs the edge detector best suited for the current set of data. For one of the edge detectors we considered two different ways of implementation, one using intensity only and the other using color information. After the selection and the extraction of the image features [usually based on color and/or texture and computed on (possibly) overlapping small windows centered around the pixel to be classified], the feature samples, handled as vectors, are grouped together in compact but well-separated clusters corresponding to each class of the image. The set of connected pixels belonging to each estimated class thus defined the different regions of the scene. The method known as K -means (or Lloyd’s algorithm) [2] (and its fuzzy version called fuzzy C-means) are some of the most commonly used techniques in the clustering-based segmentation field, and more generally, “by far, the most popular clustering algorithm used in industrial applications and machine learning” [3].

Many other methods have been proposed and studied in the last decades to solve the textured image segmentation problem. Contrary to clustering algorithms.

Years of research in segmentation have demonstrated that significant improvements on the final segmentation results may be achieved by using notably more sophisticated feature selection procedures, more elaborate clustering techniques (involving sometimes a mixture of different or non Gaussian distributions for the multidimensional texture features [9], [10]), taking into account prior distribution on the labels, region processes, or the number of classes [8], [11], [12], finally, involving (in the case of energy-based segmentation models) more costly optimization techniques.

The segmentation approach, proposed in this paper, is conceptually different and explores a new strategy; in fact, instead of considering an elaborate and better designed segmentation model of textured natural image, our technique rather explores the possible alternative of fusing (i.e., efficiently combining) several segmentation maps associated to simpler segmentation models in order to get a final reliable and accurate segmentation result. More precisely, this work proposes a fusion framework which aims at fusing several K-means clustering results (herein using as simple cues the values of the requantized color histogram estimated around the pixel to be classified) applied on an input image expressed by different color spaces. These different label fields are fused together by a simple K-means clustering techniques using as input features, the local histogram of the class labels, previously estimated and associated to each initial clustering result.

This paper demonstrates that the proposed fusion method, while being simple and fast performs competitively and often better (in terms of visual evaluations and quantitative performance measures) than the best existing state-of-the-art recent segmentation methods on the

ISSN 1512-1232

Berkeley natural image database (containing also, for quantitative evaluations, ground truth segmentations obtained from human subjects).

EDGE DETECTION FOR IMAGE SEGMENTATION

Edge detection techniques transform images to edge images benefiting from the changes of grey tones in the images. Edges are the sign of lack of continuity, and ending. As a result of this transformation, edge image is obtained without encountering any changes in physical qualities of the main image[11][25]. Objects consist of numerous parts of different color levels. In an image with different grey levels, despite an obvious change in thegrey levels of the object, the shape of the image can be distinguished in Figure 1

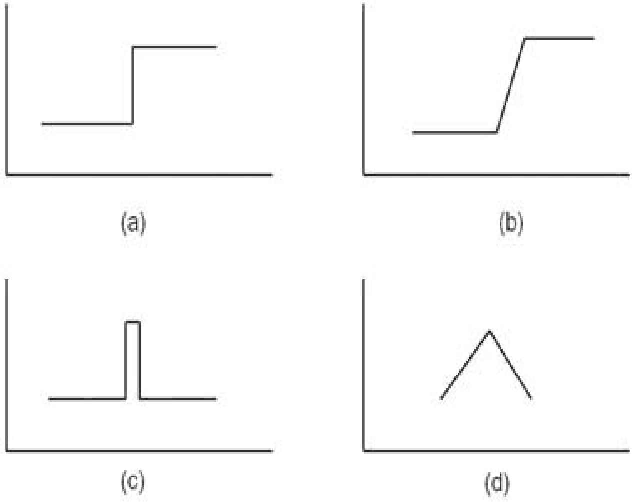


Fig. 1. Type of Edges (a) Step Edge (b) Ramp Edge (c) Line Edge (d) Roof Edge

A. Steps in Edge Detection

Edge detection contain three steps namely Filtering, Enhancement and Detection. The overview of the steps in edge detection are as follows.

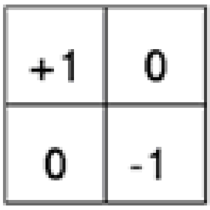
1) Filtering: Images are often corrupted by random variations in intensity values, called noise. Some common types of noise are salt and pepper noise, impulse noise and Gaussian noise. Salt and pepper noise contains random occurrences of both black and white intensity values. However, there is a trade-off between edge strength and noise reduction. More filtering to reduce noise results in a loss of edge strength[26].

2) Enhancement: In order to facilitate the detection of edges, it is essential to determine changes in intensity in the neighborhood of a point. Enhancement emphasizes pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude[14].

3) Detection: Many points in an image have a nonzero value for the gradient, and not all of these points are edges for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, thresholding provides the criterion used for detection[13].

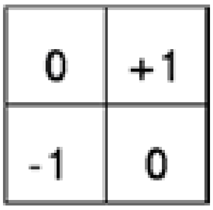
B. Edge Detection Methods Three most frequently used edge detection methods are used for comparison. These are (1) Roberts Edge Detection, (2) Sobel Edge Detection and (3) Prewitt edge detection[27]. The details of methods as follows,

1) The Roberts Detection:



+1	0
0	-1

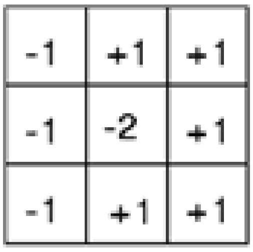
Gx



0	+1
-1	0

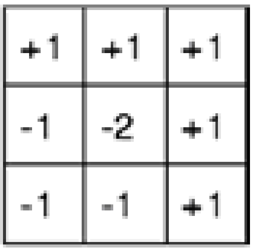
Gy

2) The Prewitt Detection:



-1	+1	+1
-1	-2	+1
-1	+1	+1

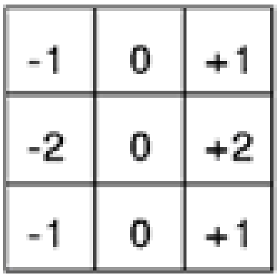
0°



+1	+1	+1
-1	-2	+1
-1	-1	+1

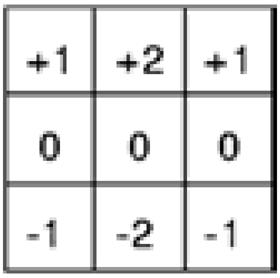
45°

3) The Sobel Detection:



-1	0	+1
-2	0	+2
-1	0	+1

Gx



+1	+2	+1
0	0	0
-1	-2	-1

Gy

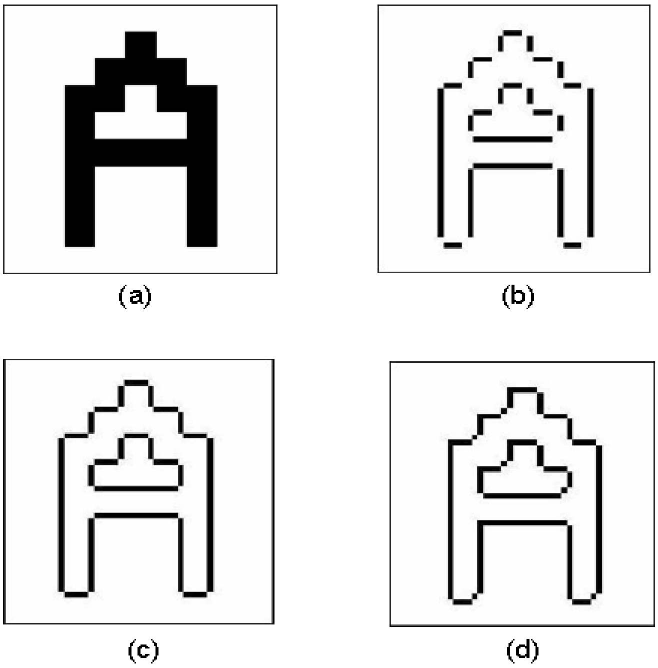


Fig. 5. The comparison of the edge detections for the example image. (a) Original Image (b) using Prewitt Edge Detection (c) using Roberts Edge Detection (d) using Sobel Edge Detection

ISSN 1512-1232

A.Fuzzy based Approach

$$\mu_{Edge}\big(g(x,y)\big)=1-\frac{1}{1+\frac{\sum_{N}\| g(x,y)-g(i,j)\|}{\Delta}}$$

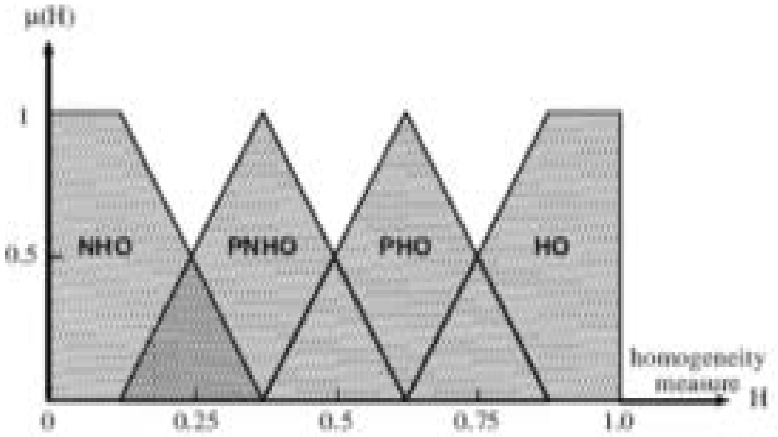


Fig. 6. The fuzzy sets used for homogeneity inference.

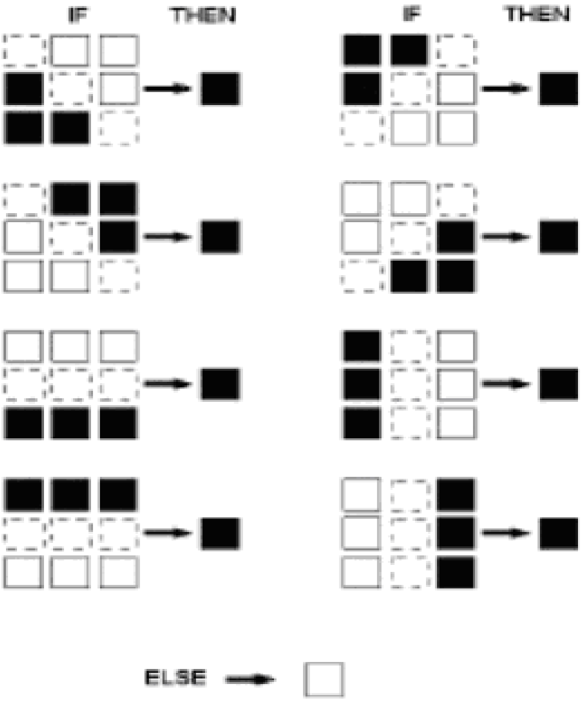


Fig. 7. Neighborhood of a central pixel

Neural networks are formed by several elements that are connected by links with variable weights[8][21]. Artificial neural networks (ANN) are widely applied for pattern recognition[27]. Their processing potential and

nonlinear characteristics are used for clustering[12][16]. Self organization of Kohonen Feature Map (SOFM) network is a powerful tool for clustering[24]. Ji and Park proposed an algorithm for watershed segmentation based on SOM[10]. This method finds the watershed segmentation of luminance component of color image[2] [27]. The method can be explained as follows. It consists of two independent neural networks one each for saturation and intensity planes[7][17]. The neural network consists of three layers namely input layer, hidden layer, and output layer as depicted in the following Figure 8.

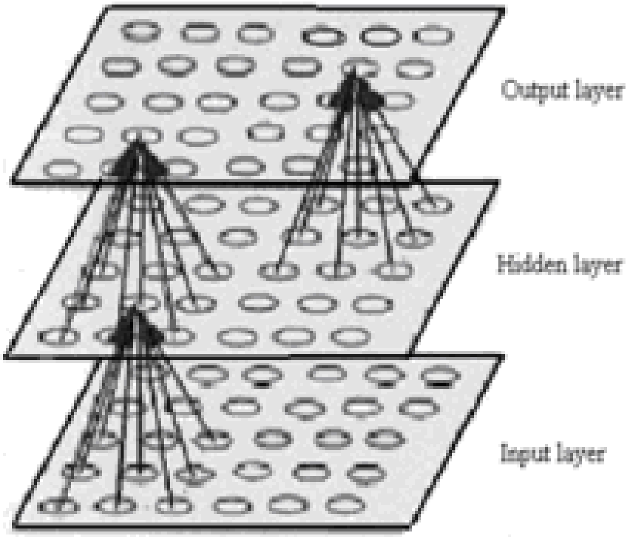


Fig. 8. Neural network approach for Image Segmentation Process

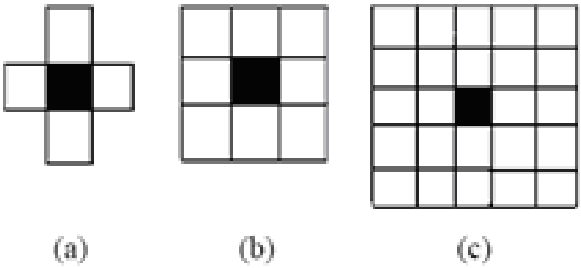


Fig. 9. Neighborhoods of a pixel (a) First order neighborhood, (b) Second order neighborhood, (c) Sequence of neighborhood.

In edge detection process, Initialize the synaptic weights of the network, Vj(0) to small, different, random numbers at iteration k=0. draw a sample y from the input set. Find the best matching (winning) neuron r(y) at iteration k, using the minimum distance Euclidean criterion[24]

$$r(y) = \min \| y - V_j \|, j = 1,2,\ldots,L$$

Update the synaptic weight vectors using the update formula

$$V_{r(y)}^{k+1} = V_{r(y)}^{k} + \eta^{k} (y - V_{r(y)}^{k}),$$
$$V_j^{k+1} = V_j^{k} + (\eta^{k})^2 (y - V_j^{k})$$
$$\forall_j \exists \Omega_{r(y)}(k)$$

where Ωr(y)(k) is the neighborhood pixel of r(y).

Increment k by 1, go to input set, and continue until the synaptic weights__ reach their steady-state values.

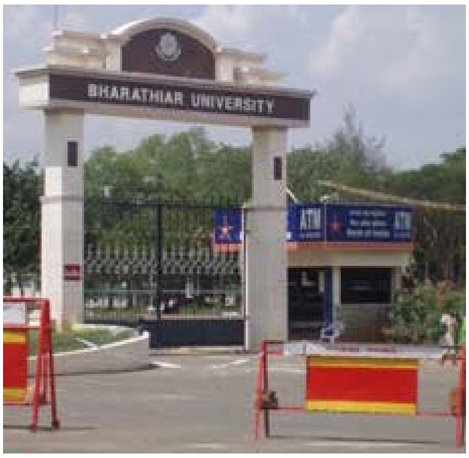


Fig. 10. Original Image.

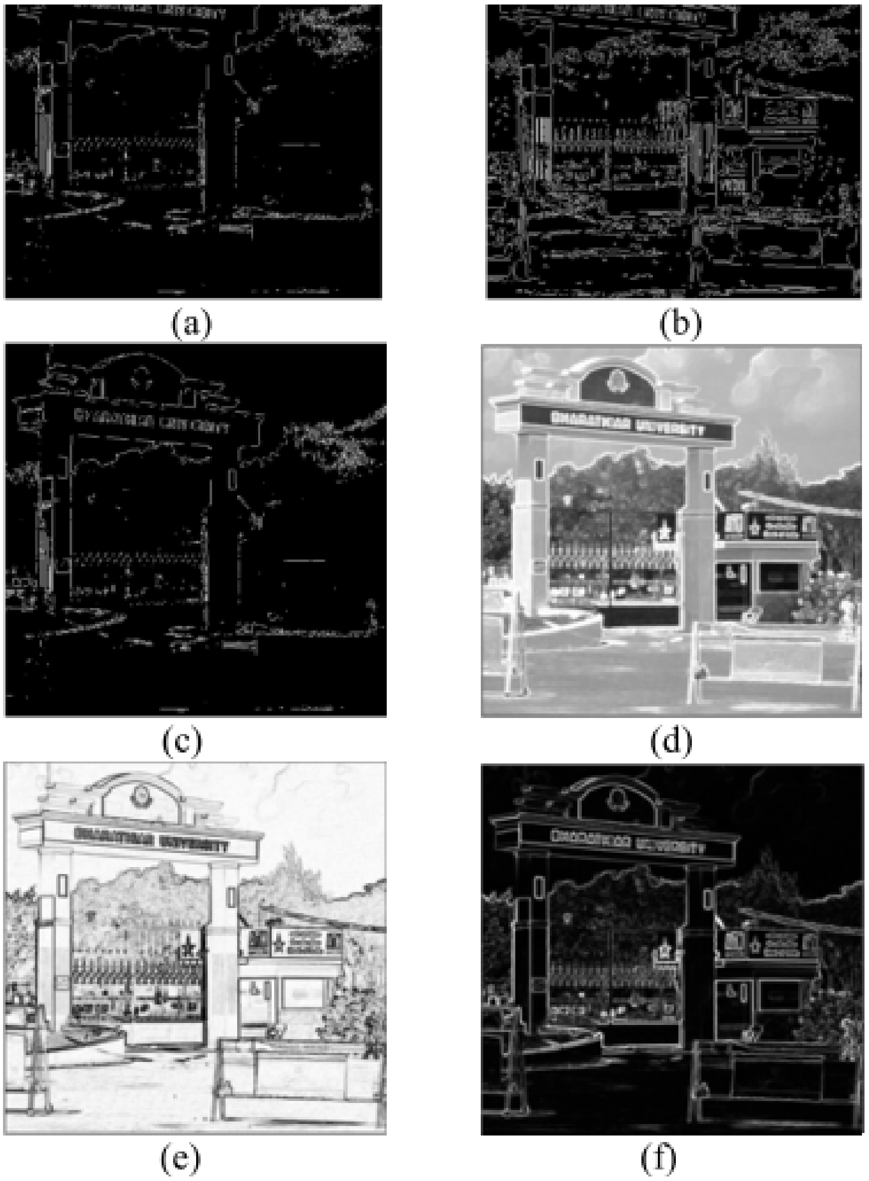


Fig. 11. Using Edge Detection Methods. (a) using Prewitt Method, (b) using Roberts Method, (c) using Sobel Method, (d) using Fuzzy Method, (e) using Genetic algorithm Method, (f) using Neural Network Method

II. INITIAL SEGMENTATIONS TO BE FUSED

The initial segmentation maps which will then be fused together by our fusion framework (see Section III) are simply given, in our application, by a K-means [2] clustering technique, applied on an input image expressed by different color spaces, and using as simple cues (i.e., as input multidimensional feature descriptor) the set of values of the re-quantized color histogram (with equidistant binning) estimated around the pixel to be classified. In our application, this local histogram is equally re-quantized (for each of the three color channels) in a N_b=5×5×5=125 bins descriptor, computed on an overlapping squared fixed-size (Nw=7) neighborhood centered around the pixel to be classified. This estimation can be quickly computed by using a more coarsely requantized color space and then computing the bin index that represents each re-quantized color (see Fig. 1 and Algorithm 1).

Mathematically, let b(x) € {0,…..,N_b-1}denote the bin index associated with the color vector y(x) at pixel location x (lying on a pixel grid) and N(x) be the set of pixel locations within the squared neighborhood region (of fixed-size $N_w \times N_w$) centered at pixel location x (in which local color information will be gathered). An estimate h(x)={h(n;x)}n=o,…,N_{b-1} of 125 bins descriptor, characterizing the color distribution for each pixel to be classified, is given by the following standard bin counting procedure:

$$h(n;\mathbf{x}) = \mathcal{K} \sum_{\mathbf{u}\in\mathcal{N}(\mathbf{x})} \delta[b(\mathbf{u}) - n]$$

Where δ is the Kronecker delta function and is $\mathcal{K} = 1/(N_w)^2$ a normalization constant ensuring $\sum_{n=0}^{N_b\ 1} h(n;\mathbf{x}) = 1$ (see Fig. 1 and Algorithm 1).

Algorithm I. Estimation, for each pixel , of the bins descriptor.

```
Estimation of the $N_b = q^3$ bins descriptor.
$\mathcal{N}_\mathbf{x}$ Set of pixel locations x within the $N_w \times N_w$ neighborhood region centered at x.
h[] Bins descriptor: Array of $N_b$ floats (h[0],h[1],…,h[$N_{b-1}$]).
[.] integer part
For each pixe l $\mathbf{x} \in \mathcal{N}_\mathbf{x}$ with color value $R_\mathbf{x}, G_\mathbf{x}, B_\mathbf{x}$ do
    • $k \leftarrow q^2 \cdot \lfloor q \cdot R_\mathbf{x}/256 \rfloor + q \cdot \lfloor q \cdot G_\mathbf{x}/256 \rfloor + \lfloor q \cdot B_\mathbf{x}/256 \rfloor$.
    • $h[k] \leftarrow h[k] + 1/(N_w)^2$.
```

In this simpler model, a *texton* (i.e., the repetitive character or element of a textured image, also called a texture primitive) is herein characterized by a mixture of colors or more precisely

by the values of the re-quantized (local) color histogram. This model is simple to compute, allows significant data reduction while being robust to noise and local image transformations and has already demonstrated all its efficiency for tracking applications [13].

Finally, these (125-bin) descriptors are grouped together into different clusters (corresponding to each class of the image) by the classical K -means algorithm [2] with the classical Euclidean distance. This simple segmentation strategy of the input image into K_1 classes is repeated for different color spaces which can be viewed as different image channels provided by various sensors or captors (or as a multichannel filtering where the channels are represented by the different color spaces). In our application, we use

N_s segmentations provided by the color spaces, namely the $N_s = 6$ color spaces, namely the $\mathcal{C} = \{\mathrm{RGB}, \mathrm{HSV}, \mathrm{YIQ}, \mathrm{XYZ}, \mathrm{LAB}, \mathrm{LUV}\}$

color spaces [1], [14]–[16]. Of course, these initial segmentations to be fused can result of the same initial and simple model used on an input image filtered by another filter bank (e.g., a bank of Gabor filters [11], [17] or any other 2-D decomposition of the frequential space) or can also be provided by different segmentation models or different segmentation results provided by different seeds of the same stochastic segmentation model.

Each color space has an interesting property, which can efficiently be taken into account in order to make more reliable thefinal fusion procedure. For example, RGB is an additive color system based on tri-chromatic theory and nonlinear with visual perception. This space color seems to be the optimal one for tracking applications [18]. The HSV is interesting in order to decouple chromatic information from shading effect [13]. The YIQ color channels have the property to code the luminance and chrominance information which are useful in compression applications (both digital and analogue). Besides, this system is intended to take advantage of human color characteristics. XYZ has the advantage of being more psycho-visually linear, although they are nonlinear in term of linear component color mixing. The LAB color system approximates human vision, and its component closely matches human perception of lightness [1]. The LUV components provide an Euclidean color space yielding a perceptually uniform spacing of color approximating a Riemannian space [17]. Each of these properties will be efficiently combined by our fusion technique.

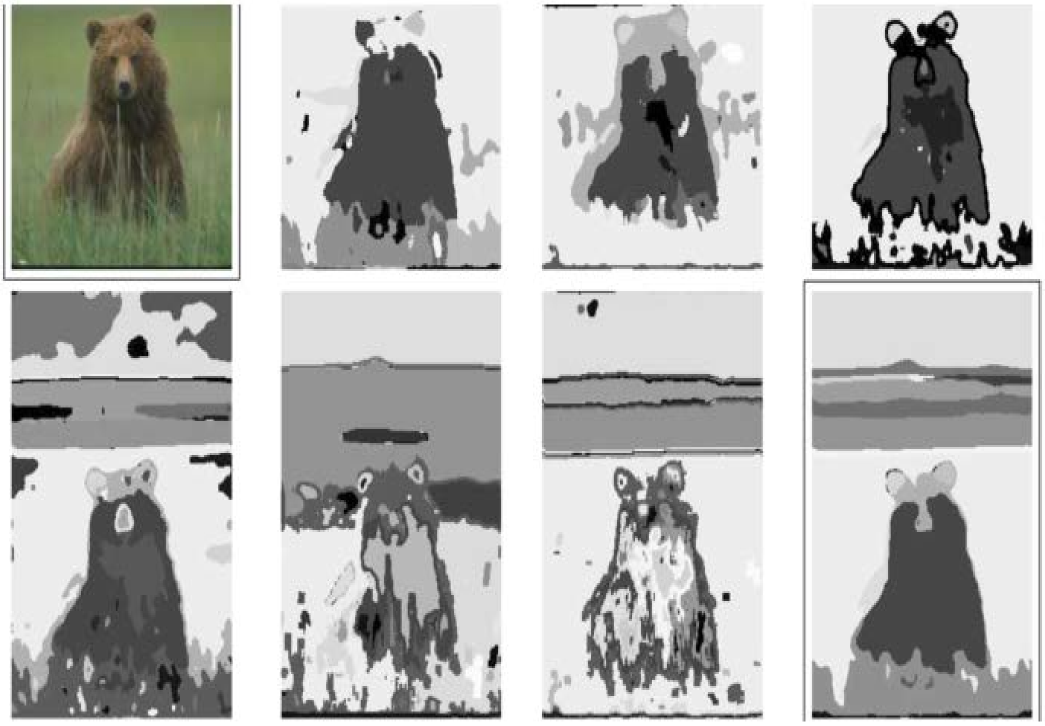


Fig.12. Examples of fusion results (FCR). From top to bottom and left to right: (top left) input natural image from the Berkeley image database. Six segmentation results (into K = 6 classes)

associated to clustering model described in Section II on the top left input image expressed in the RGB, HSV, YIQ, XYZ, LAB, and LUV color spaces and final segmentation map (into K = 6classes) resulting of the fusion of these six clusterings (bottom right) (see Table I for an objective and quantitative comparison).

III. FUSION OF SEGMENTATION MAPS

The key idea of the proposed fusion procedure simply consists of considering, for each site (or pixel to be classified), the local histogram of the class (or *texton*) labels of each segmentation to be fused, computed on a squared fixed-size (N_w)neighborhood centered around the pixel, as input feature vector of a final clustering procedure. For a fusion of N_s segmentation with K_1 classes into a segmentation withK_2classes, the preliminary feature extraction step of this fusion procedure thus yields to $(K_1\text{-bin})$ histograms which are then gathered together in order to form, a $K_1 \times N_s$-dimensional feature vector or a final $(K_1 \times N_s)$ bin histogram which is then normalized to sum to one, so that it is also a probability distribution function.

The proposed fusion procedure is then herein simply considered as a problem of clustering local histograms of (preliminary estimated) class labels computed around and associated to each site. To this end, we use, once again, a K-means clustering procedure exploiting, for this fusion step, an histogram-based similarity measure derived from the Bhattacharya similarity coefficient. Given a normalized histogram

$\{h(n;\mathbf{x})\}_{n=0,\ldots,N_b-1=K_1 N_s-1}$ (at pixel location) and a reference histogram

$\{h^\star(n)\}_{n=0,\ldots,N_b\ \ 1}$ (representing one of the K2 cluster centers of each class of a K – means procedure), the Bhattacha distance between these two histograms is defined as

$$D_B[h^\star, h(\mathbf{x})] = \left(1 - \sum_{n=0}^{N_b-1} \sqrt{h^\star(n) h(n;\mathbf{x})}\right)^{1/2}$$

and a -means algorithm based on this distance converges in all tested examples.

The preestimated label fields to be fused (see Section II), along with the fusion procedure can be viewed (and qualitatively explained) as a two-step hierarchical segmentation procedure in which, first, a texton segmentation map (in each color space) is estimated and, second, a final clustering, taking into account this mixture of textons (expressed in the set of color space), is then used for a final clustering.We recall that a texton, in our framework, is defined by a nonparametric mixture of colors (see Section II).

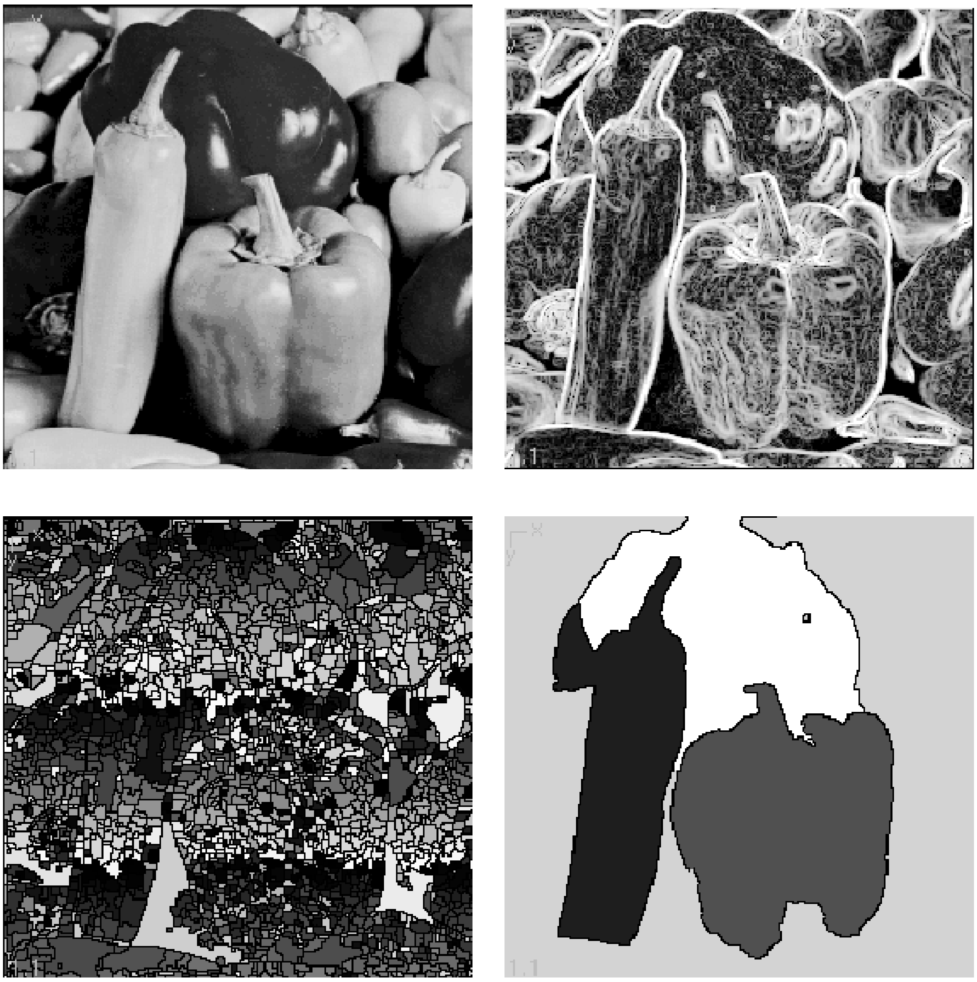
Consequently, in this final clustering (the fusion procedure), two sites for which the local-class-label histogram (i.e., the mixture of textons in the different color spaces given by the $K_1 \times N_s$ bins histogram) are not too far away from each other will be in the same class in the resulting fused segmentation. Inversely, two sites associated to different local-class-label histograms will likely belong to different classes in the final segmentation.

Fig. 2 shows an example of the clustering segmentation model presented in Section II (into classes) of an input image expressed in the RGB, HSV, YIQ, XYZ, LAB, and LUV color spaces and the final segmentation map (into$K_2 = 6$ classes) which results of the fusion of these N_s clusterings. We can notice that none of them can be considered as reliable except the final segmentation result (at bottom right) which visually identify quite faithfully the different objects of the scene.

A final merging step is necessary and is used to avoid oversegmentation for some images. It consists of fusing each region (i.e., set of connected pixels belonging to the same class) of the resulting segmentation map with one of its neighboring region $\mathcal{R}$ if the distance $\mathcal{D}_{\text{MERGING}}$ is below a given threshold\ (or if its size is below 50 pixels with the closest region in the $\mathcal{D}_{\text{MERGING}}$ distance sense)

$$\mathcal{D}_{\text{MERGING}} = \min_{\mathbf{x}\in\mathcal{R}} \left\{ \sum_{\mathcal{C}} D_{\mathcal{B}}[h^{\circ}(n), h^{\ddagger}(n;\mathbf{x})] \right\}.$$

In (3), the first summation is done on the six used color spaces, and $h^{\circ}(n)$ designates the normalized nonparametric histogram of the set of pixels belonging to the region to be merged and $h^{\ddagger}(n;\mathbf{x})$ is the normalized histogram, computed on a squared fixed-size (N_w) windows (at pixel location x and totally included in the region). For this merging procedure, the two histograms are equally re-quantized (for each of the three color channels) with $4 \times 4 \times 4$ bins (see Fig. 3 where this merging strategy is, for this example, intensively used).



Example of final merging step using the Bhattacharya distance on different color spaces as merging criterion on a fused segmented image of the Berkeley database.

IV. EXPERIMENTAL RESULTS

A. Set Up

In all the experiments, we have considered our fusion methods on initial segmentations obtained with the following parameters: the size of the squared window, used to compute the local histogram for the initial segmentations or the fusion procedure is set to $N_w = 7 \times 7$.The number of bins for each local re-quantized histogram is set to $N_b = 5 \times 5 \times 5$.We use $N_s = 6$ segmentations provided by the following color spaces RGB, HSV, YIQ, XYZ, LAB, and LUV. Several quantitative performance measures will be given for several values (comprised between 6 and 13) of K_1 and K_2 respectively, the number of classes of the segmentation to be fused and theresulting number of classes of the final fused segmentation map. The optimal value of K seems to be comprised between 0.10 and 0.15.

B. Comparison With State-of-the-Art Methods

We have replicated the scenario used in the evaluation of state-of-the-art segmentation methods described in [23] and [25] . In these experiments, we have to test our segmentation algorithm on the Berkeley segmentation database [21] consisting of 300 color images of size 481× 321. For each color image, a set of benchmark segmentation results, provided by human observers (between 4 and 7), is available and will be used to quantify the reliability of the proposed segmentation algorithm. As proposed in [23]–[25], we have compared our segmentation algorithm (called FCR for fusion of clustering results) against four unsupervised algorithms, available publicly. For each of these algorithms, their internal parameters are set to their optimal value (see [23]) and/or corresponds to the internal values suggested by the authors. These algorithms are namely the mean-shift [5] (with $h_s = 13, h_r = 19$) Ncuts [6] (with a number of segments $K = 20$, agreeing with the average number of regions found in the segmentation maps given by the human observers [25]), and FH [22] (with a smoothing parameter $\sigma = 0.5$a threshold value $k = 500$ and a minimal region size equals to 200 pixels), and, finally, the CTM algorithm proposed in [23] and [24] (with $\eta = 0.1 \text{ and } \eta = 0.2$)

As in [23] and [25], all color images are normalized to have the longest side equals to 320 pixels (in this paper, this operation was done by the Linux command *convert* which is a member of the ImageMagick suite of tools). The comparison is based on the following performance measures, namely a probabilistic measure called PRI (higher probability is better) and three metrics VoI, GCE, and BDE (lower distance is better). The qualitative meaning of these performance measures are recalled as follows.

1) The Rand index [19] counts the fraction of pairs of pixels whose labellings are consistent between the computed segmentation and the groun truth. This quantitative measure is easily extended to the probabilistic Rand index (PRI) [26] by averaging the result across all human segmentations of a given image.

2) Contrary to the PRI, based on pairwise relationships, the variation of information (VoI) metric [20] 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. More precisely, it measures the amount of information that is lost or gained in changing from one clustering to another (and, thus, can be viewed as representing the amount of andomness in one segmentation which cannot be explained by the other).

3) The global consistency measure (GCE) [21] 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 segmentations 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

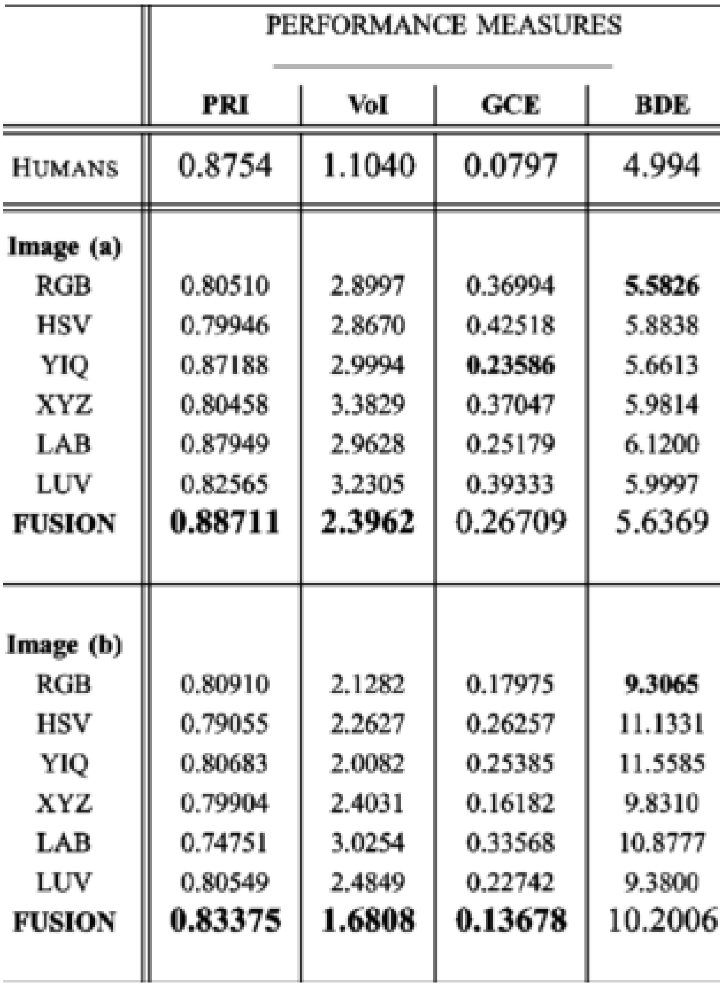
a finer level of detail will merge in such a way that they yield the larger regions proposed by a different observer at a coarser level).

4) The boundary displacement error (BDE) [22] measures the average displacement error of one boundary pixels and the closest boundary pixels in the other segmentation.

As noticed in [23], PRI seems to be more highly correlated with human hand segmentations. Let us also mention that a inherent problem with the GCE measure is that it does not penalize oversegmentation at all (the highest score is given by assigning each pixel to an individual region). Some of these interesting performance measures thus have degenerate cases (i.e., unrealistic bad segmentations give abnormally high score), these complementary measures have thus to be considered all together in order to quantify the performance of a given segmentation method.

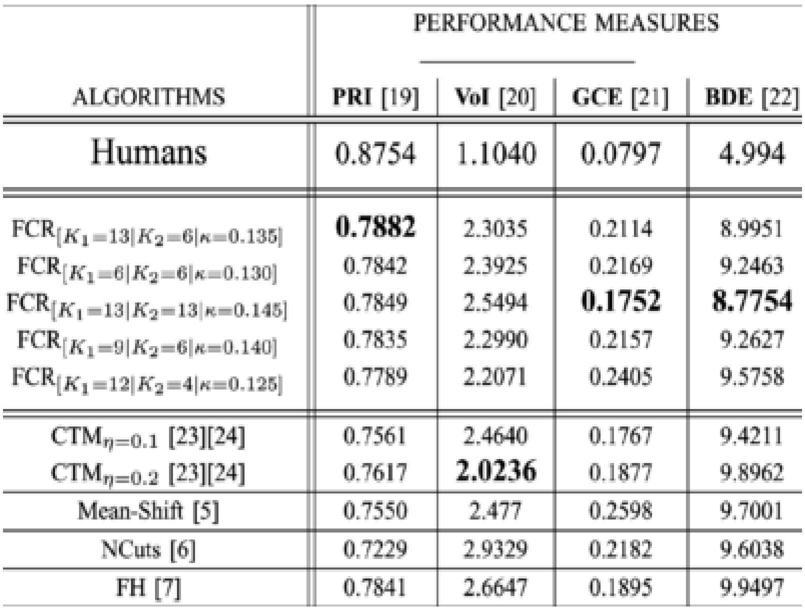
Table I shows the obtained results for the images presented in Fig. 2. Table II shows the obtained results for different values of K_1 and K_2 and k . Fig. 4 shows the distribution of the different performance measure over the 300 images of the Berkeley image database for, $K_1 = 13, K_2 = 6, \kappa = 0.$ We can notice that the discussed fusion strategy allows to give very competitive results among these four different quantitative performance measures with a relative low variance over the set of images of the Berkeley image database. Fig. 6 displays some

Table I. Performance measures for, respectively, the clustering result expressed in each color space and the fusion result given by our algorithm (higher is better for pri and lower is better for voi, gce, and bde) on the two images presented in fig. 2



	PERFORMANCE MEASURES			
	PRI	**VoI**	**GCE**	**BDE**
HUMANS	0.8754	1.1040	0.0797	4.994
Image (a)				
RGB	0.80510	2.8997	0.36994	**5.5826**
HSV	0.79946	2.8670	0.42518	5.8838
YIQ	0.87188	2.9994	**0.23586**	5.6613
XYZ	0.80458	3.3829	0.37047	5.9814
LAB	0.87949	2.9628	0.25179	6.1200
LUV	0.82565	3.2305	0.39333	5.9997
FUSION	**0.88711**	**2.3962**	0.26709	5.6369
Image (b)				
RGB	0.80910	2.1282	0.17975	**9.3065**
HSV	0.79055	2.2627	0.26257	11.1331
YIQ	0.80683	2.0082	0.25385	11.5585
XYZ	0.79904	2.4031	0.16182	9.8310
LAB	0.74751	3.0254	0.33568	10.8777
LUV	0.80549	2.4849	0.22742	9.3800
FUSION	**0.83375**	**1.6808**	**0.13678**	10.2006

Table II. Average performance of our algorithm for several values of its internal parameters (parameter vector [k jk j_] indicated for each experiment) and for different performance measures (higher is better for pri and lower is better for voi, gce and bde) on the berkeley image database
(http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/)



	PERFORMANCE MEASURES			
ALGORITHMS	**PRI** [19]	**VoI** [20]	**GCE** [21]	**BDE** [22]
Humans	0.8754	1.1040	0.0797	4.994
$\text{FCR}_{[K_1=13 \mid K_2=6 \mid \kappa=0.135]}$	**0.7882**	2.3035	0.2114	8.9951
$\text{FCR}_{[K_1=6 \mid K_2=6 \mid \kappa=0.130]}$	0.7842	2.3925	0.2169	9.2463
$\text{FCR}_{[K_1=13 \mid K_2=13 \mid \kappa=0.145]}$	0.7849	2.5494	**0.1752**	**8.7754**
$\text{FCR}_{[K_1=9 \mid K_2=6 \mid \kappa=0.140]}$	0.7835	2.2990	0.2157	9.2627
$\text{FCR}_{[K_1=12 \mid K_2=4 \mid \kappa=0.125]}$	0.7789	2.2071	0.2405	9.5758
$\text{CTM}_{\eta=0.1}$ [23][24]	0.7561	2.4640	0.1767	9.4211
$\text{CTM}_{\eta=0.2}$ [23][24]	0.7617	**2.0236**	0.1877	9.8962
Mean-Shift [5]	0.7550	2.477	0.2598	9.7001
NCuts [6]	0.7229	2.9329	0.2182	9.6038
FH [7]	0.7841	2.6647	0.1895	9.9497

Examples of segmentations obtained by our algorithm. The results for the entire database are available online at http://www.iro.umontreal.ca/~mignotte/ResearchMaterial. In short, the proposed algorithm outperforms for several different internal parameters, all the well-known segmentation algorithms presented in Table II in terms of PRI and BDE indices. In term of the VoI

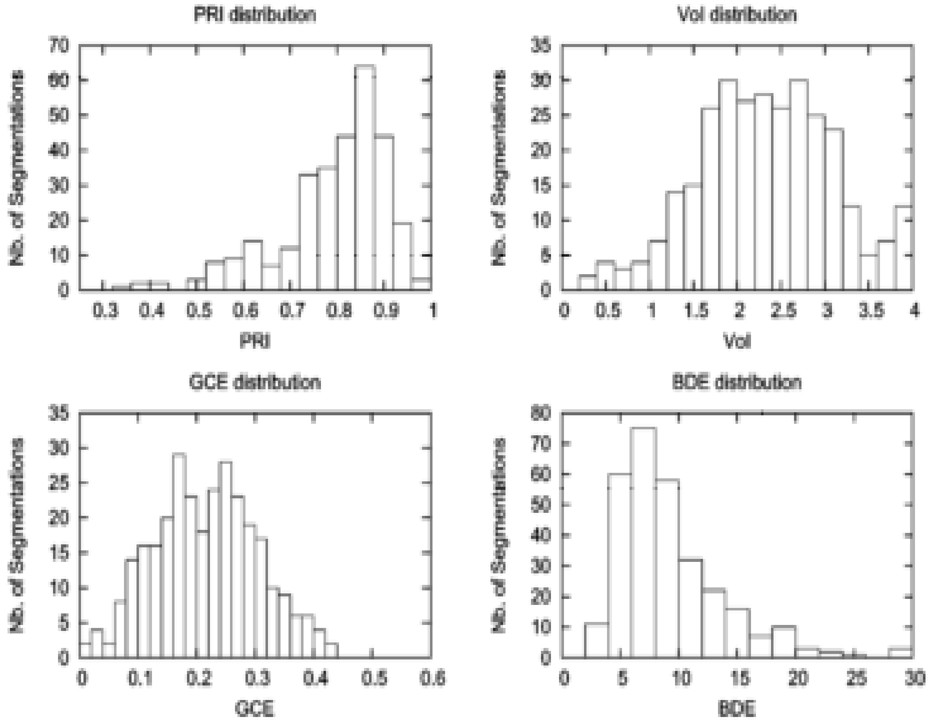
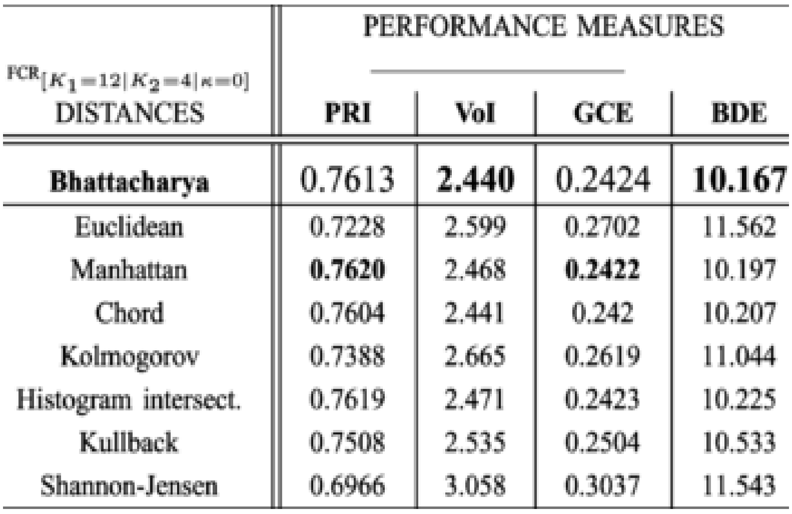


Fig. 4. Distribution of the difference performance measures, respectively from top to bottom; PRI, VoI, GCE, BDE over the 300 images of the Berkeley database for FCR $[K_1=13|K_2=6|\kappa=0\ 135]$.

Table III. Influence of the distance choice used in the final fusion procedure (average performance on the berkeley image database)



${}^{\mathrm{FCR}}[K_1=12\|K_2=4\|\kappa=0]$ DISTANCES	PERFORMANCE MEASURES			
	PRI	**VoI**	**GCE**	**BDE**
Bhattacharya	0.7613	**2.440**	0.2424	**10.167**
Euclidean	0.7228	2.599	0.2702	11.562
Manhattan	**0.7620**	2.468	**0.2422**	10.197
Chord	0.7604	2.441	0.242	10.207
Kolmogorov	0.7388	2.665	0.2619	11.044
Histogram intersect.	0.7619	2.471	0.2423	10.225
Kullback	0.7508	2.535	0.2504	10.533
Shannon-Jensen	0.6966	3.058	0.3037	11.543

Indices, only the CTM algorithm performs equivalently or better and for the GCE measure, our algorithm gives, on average, similar results than the others and outperforms them all for a set of parameters (in which K1 is high, leading to a classical over segmentation).

C. Sensitivity to Parameters

k allows to refine the final segmentation map and allows, to a certain extent, to avoid some over-segmented partition maps results (especially when K2 is high). With k=0 (i.e., without the final fusion step and the other parameters being set to K1=12 and K2=4),we obtain PRI=0.7613 (to be compared to PRI=0.7789.when k=0.125;see Table II)

In order now to quantify the influence of the distance choice, used in the final clustering used as fusion procedure, we have compared the performance measures obtained with our method (K_1=12;K_2=4;k=0) using a Bhattacharya distance and several other metrics [27] (see Table III).We can notice that several distances are as efficient as the Bhattacharya distance, for example, the Manhattan distance (L1 norm) or the histogram intersection-based distance. For the same parameters given in Table II(k≠0) , these two metrics allow to obtain similar performance measures comparatively to the Bhattacharya distance.

Table IV. Influence of the size of the window n_w used to estimate the local histograms (average performance on the berkeley image database)

FCR$_{[K_1=12\|K_2=4\|\kappa=0]}$ SIZES	PERFORMANCE MEASURES PRI	VoI	GCE	BDE
7 × 7	0.7613	2.440	0.2424	10.167
5 × 5	0.7621	2.506	0.2349	9.821
9 × 9	0.7604	2.401	0.2515	10.552

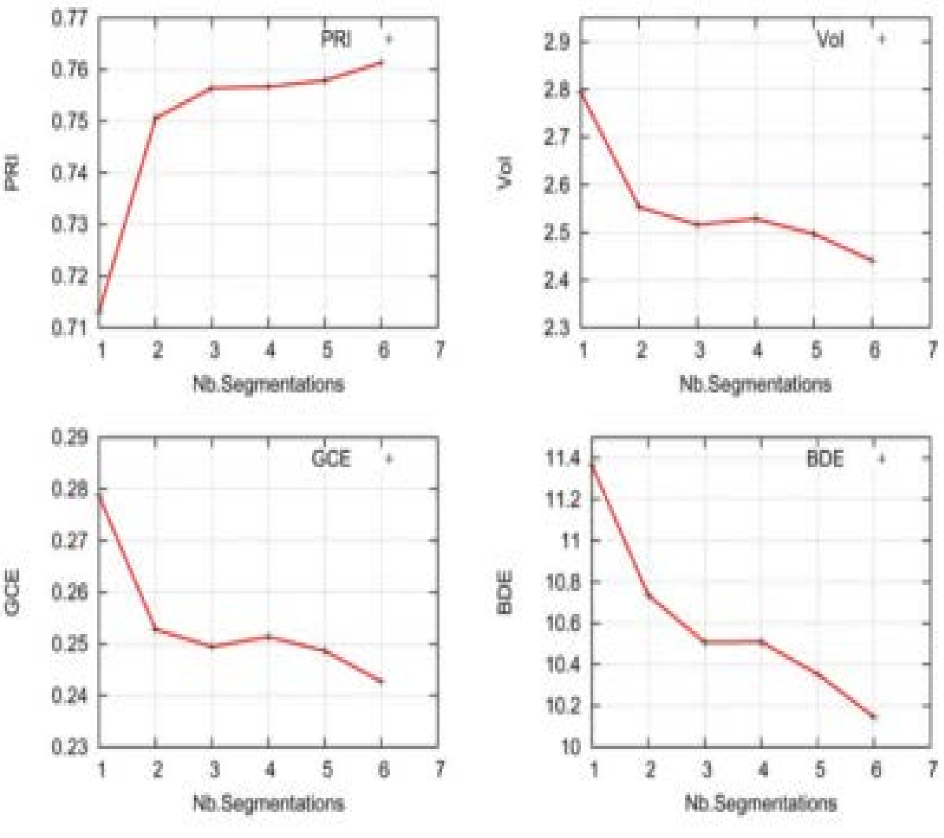


Fig. 5. Evolution of the PRI, VoI, GCE, and BDE measures as a function of the number of segmentations (N_s) to be fused for the FCR algorithm. For $N_s = 1$, i.e., without fusion, the segmentation model is the one described in Section II with K = K = 4.

We have also quantified in Table IV the influence of the size of the window (used to estimate the local color histogram). These tests show that the performance measures are not too much sensitive to this internal parameter.

We can also notice (see Fig. 5) that all the performance measures are all the more better thanN_s (number of segmentation to be fused) is high. This experiment shows the validity of our fusion procedure and also the performance measures obtained by the simple segmentation model presented in Section II.

D. Discussion

Tests have shown that a higher value for k_2 $(k_1 \geq k_2 < 15)$ will induce a lower (consequently better) GCE and BDE performance measure but also a higher (consequently less good) VoI measure. The PRI measure is quite influenced by the value of k whose optimal value seems to be comprised between 0.10 and 0.15. The fusion method is not too much sensitive to the value of N_w (size of the window used to estimate the local histograms), and, finally, the performance measures are all the more better than N_s (number of segmentations to be fused) is high.

The segmentation procedure takes less than one minute for an AMD Athlon 64 Processor 3500+, 2.2 GHz, 4435.67 bogomips and running on Linux. The source code (in C++ language) of our algorithm is available at the followingaddress http://www.iro.umontreal.ca/~mignotte-/ResearchMaterial .

V. CONCLUSION

In this paper, we have presented a new segmentation strategy based on a fusion procedure whose goal is to combine several segmentation maps in order to finally get a more reliable and accurate segmentation result. The initial segmentations to be fused can be the output result of the same initial and simple model used on an input image filtered by a given filter bank, or it can also be provided by different segmentation models or different segmentation results provided by different seeds (or different variation of parameters) of the same stochastic segmentation model. This fusion framework remains simple, fast, easily parallelizable, general enough to be applied to various computer vision applications, and performs competitively among the recently reported state-of-the-art segmentation methods

ACKNOWLEDGMENT

The author would like to thank the 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. S. Banks, Signal Processing, Image Processing and Pattern Recognition. Englewood Cliffs, NJ: Prentice-Hall, 1990.
2. S.P.Lloyd, “Least squares quantization in PCM,” IEEE Trans. Inf. Theory, vol. IT-28, no. 2, pp. 129–136, Mar. 1982.
3. P.Berkhin, “Survey of clustering data mining techniques,” Accrue Software, San Jose, CA, 2002.
4. J.Besag, “On the statistical analysis of dirty pictures,” J. Roy. Statist. Soc. B, vol. 48, pp. 259–302, 1986.
5. D.Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no. 5, pp. 603–619, May 2002.
6. J.Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905, Aug.2000.
7. P.Felzenszwalb and D.Huttenlocher, “Efficient graph-based image segmentation,” Int. J. Comput. Vis., vol. 59, pp. 167–181, 2004.

8. S.Zhu and A.Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 9, pp. 884–900, Sep.1996.
9. M.Mignotte, C.Collet, P.Pérez, and P.Bouthemy, "Sonar image segmentation using a hierarchical MRF model," IEEE Trans. Image Process., vol. 9, no. 7, pp. 1216–1231, Jul. 2000.
10. M.Mignotte, C.Collet, P.Pérez, and P.Bouthemy, "Three-class Markovian segmentation of high resolution sonar images," Comput. Vis.Image Understand., vol. 76, no. 3, pp. 191–204, 1999.
11. F.Destrempes, J.-F.Angers, and M.Mignotte, "Fusion of hidden Markov random field models and its Bayesian estimation," IEEE Trans. Image Process., vol. 15, no. 10, pp. 2920–2935, Oct. 2006.
12. Z.Kato, T.C.Pong, and G.Q.Song, "Unsupervised segmentation of color textured images using a multi-layer MRF model," in Proc. Int.Conf. Image Processing, Barcelona, Spain, Sep. 2003, pp. 961–964.
13. P.Pérez, C.Hue, J.Vermaak, and M.Gangnet, "Color-based probabilistic tracking," in Proc. Eur. Conf. Computer Vision, Copenhagen, Denmark, Jun. 2002, pp. 661–675.
14. J.B.Martinkauppi, M.N.Soriano, and M.H.Laaksonen, "Behavior of skin color under varying illumination seen by different cameras at different color spaces," in Proc. SPIE, Machine Vision Applications in Industrial Inspection IX, San Jose, CA, Jan. 2001, pp. 102–113.
15. J.-P.Braquelaire and L. Brun, "Comparison and optimization of methods of color image quantization," IEEE Trans. Image Process., vol. 6, no. 7, pp. 1048–1952, Jul. 1997.
16. H.Stokman and T.Gevers, "Selection and fusion of color models for image feature detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 3, pp. 371–381, Mar. 2007.
17. Z.Kato, "A Markov random field image segmentation model for color textured images," Image Vis. Comput., vol. 24, no. 10, pp. 1103–1114, 2006.
18. E. Maggio and A. Cavallaro, "Multi-part target representation for color tracking," in Proc. Int. Conf. Image Processing, Italy, Genova, Sep.2005, pp. 729–732.
19. R. Unnikrishnan, C. Pantofaru, and M. Hebert, "A measure for objective evaluation of image segmentation algorithms," in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop on Empirical Evaluation Methods in Computer Vision, Jun. 2005, vol. 3, pp.34–41.
20. M. Meila, "Comparing clusterings—An axiomatic view," in Proc. 22nd Int. Conf. Machine Learning, 2005, pp. 577–584.
21. D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in Proc. 8th Int. Conf.Computer Vision, Jul. 2001, vol. 2, pp. 416–423.
22. J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi, "Yet another survey on image segmentation: Region and boundary information integration," in Proc. 7th Eur. Conf. Computer Vision Part III, Copenhagen, Denmark, May 2002, pp. 408–422, LNCS.
23. A. Y. Yang, J. Wright, S. Sastry, and Y. Ma, "Unsupervised segmentation of natural images via lossy data compression," Comput. Vis. Image Understand., 2007, submitted for publication.
24. Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy coding and compression," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, pp. 1546–1562, Sep. 2007.
25. A. Y. Yang, J. Wright, S. Sastry, and Y. Ma, "Unsupervised segmentation of natural images via lossy data compression," Elect. Eng. Comput. Sci. Dept. Univ. California, Berkeley, 2006 [Online]. http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-195.html
26. R.Unnikrishnan, C.Pantofaru, and M.Hebert, "Toward objective evaluation of image segmentation algorithms," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 929–944, Jun. 2007.
27. S.-H.Cha and S.N.Srihari, "On measuring the distance between histograms," Pattern Recognit., vol. 35, pp. 1355–1370, 20

Article received: 2011-07-12