A Strategy for Boundary Detection Combining Region and Edge Information

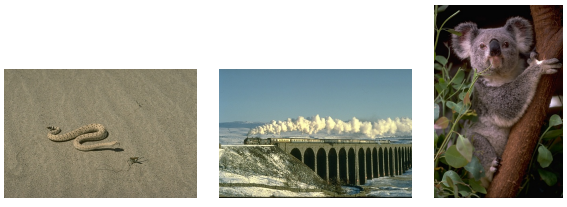
Karin S. Komati, Jorge L. A. Samatelo, Evandro O. T. Salles, Mario Sarcinelli Filho
Graduate Program on Electrical Engineering
Federal University of Espirito Santo
Vitoria, Brazil
{karinkomati,rigvedas}@gmail.com, {evandro,mario.sarcinelli}@ele.ufes.br

***Abstract*—A method to detect boundaries in in natural color images is here proposed, combining edge information and region information. This unsupervised fully automatic process uses edge map information to eliminate false boundaries in the image region map, and region map information to remove noise in the image edge map. Thus, it integrates these two maps into a single one to get the final result. This proposal is extensively compared to the multi-label graph cut approach, since both approaches are unsupervised and fully automatic, as well as receive the same two inputs, although performing different processing. Experiments performed on a large set of natural color images were the base for such comparison. The results show that the approach here proposed is promising, besides allowing interesting interpretations about boundary detection.**

***Keywords*-boundary detection, multilabel graph cut, region growing, edge detection.**

I. Introduction

Boundary detection is a crucial task in computer vision systems. Actually, high level procedures, like object recognition, strongly rely on the quality of boundary information. A fully automated contour detection is a complex issue, since no prior knowledge about what types of regions (uniform, with smooth gradation of color and texture variations) are present in an image, or how many they are. Examples of the variety and complexity of images can be found in Figure 1. Figure 1(a) is the image of a snake in the desert, where the central element and the background have almost the same color, causing an ill-defined border. Figure 1(b) shows an image with a mixture of artificial and natural textures (the bridge has geometric patterns, quite unlike the natural texture of the hill and the trail of smoke from the locomotive). In Figure 1(c) we have a koala in a tree, where, despite the complexity of the texture of its fur, human perception recognizes it as a single element.



(a) (b) (c)

Figure 1. Image examples (extracted from [1]).

Traditionally, boundary-detection techniques are classified as region or edge approaches. Region-based techniques rely on common patterns in intensity values within a cluster of neighboring pixels. The cluster is referred to as the region, and the goal of the segmentation algorithm is to group regions according to their anatomical or functional roles. Edge-based techniques rely on discontinuities in image values between distinct regions, and the goal of the segmentation algorithm is to accurately demarcate the boundary separating these regions. However, there may exist gaps and noisy edges in edge-approach results, whereas region-approach results tend to be over-segmented with inaccurate boundaries.

There are many proposals combining the outputs of region-growing and edge detection methods to improve the quality of their results. Muñoz, Freixenet, Cufí and Martí [2] show seven different strategies for combining similarity (region) and discontinuity (edge) information. They were grouped in two classes: embedded integration and post-processing integration. Embedded integration produces, in general, a single complex algorithm to avoid errors in the results. The post-processing strategy works with a set of many algorithms. This approach accepts faults in the elementary algorithms, and the posteriori integration module tries to correct these errors.

In this paper, we propose a non-supervised and fully automated post-processing integration scheme to combine edge information extracted by a classical method with region information. This structure, called KoSS (for **Ko**mati, **S**alles and **S**arcinelli-Filho), eliminates false boundaries in the region map, guided by the edge map, and the noise in edge map as well, now guided by the region map, thus taking the advantage of their complementary nature. The results obtained with our approach are closer to the human perception than any of the two images used as inputs for the post-processing integration. This proposal is extensively compared to the multi-label graph-cut approach. Both approaches are fully automatic and receive the same input, knowing the region and edge maps, although adopting quite different processing techniques.

However, quantitative performance comparison requires ground truth and well defined metrics. Both requirements can be found in "The Berkeley Segmentation Dataset and Benchmark" (BSDS) [1]. For each image in BSDS, there are at least five hand-labeled segmentations made by human beings, which constitute the ground truth. BSDS operates on a non-

thresholded boundary map. Nevertheless, it does the binarization at many levels. At each level, it computes precision, recall, and F-measure, and thus produces a precision-recall curve for each image, over which it applies the algorithm in [3]. The overall F-measure obtained is the maximum F-measure value along the precision-recall curve, and summarizes the image statistics. Thus, the standard metric of BSDS is F-measure, which determines how well the boundary map obtained approximates the human ground truth boundaries.

Therefore, as the test images are got from the BSDS, the algorithm here proposed is evaluated considering the F-measure metrics, as well as the comparison of its performance with the one of the multi-label graph-cut.

II. The Information Flow

In this proposal, the region-growing result and edge information are independently extracted from the same original image (Figure 2). The integration process is here performed through using the KoSS or the multilabel graph cut. Both approaches use the complementary information of edge based and region based information, to reduce the problems that arise in each individual methods.

We are considering that the region-map presents standard characteristics: the image is binary and all regions are bounded by closed contours. For the region growing method, we choose the MM-Frac approach proposed in [4], which is based on the fully-automatic JSEG method [5]. Essentially, the JSEG's homogeneity criterion is mixed with the Multifractal Measurement [6] controlled by an image global statistical property: the shape of the power spectrum of the image being analyzed [7].

It is necessary that the edge-map be a soft map, with each pixel valued from zero to one, where higher values mean greater confidence in the existence of a boundary. As for the edge map, some classical edge detectors (Sobel, Prewitt, Laplacian and morphological gradient) [8] generating an output known as a soft boundary map were tested, and the result is that the morphological gradient presents the overall F-measure slightly better than the other detectors. Therefore, it was chosen as the edge detection method for this work. As for the usual image smoothing to eliminate noise before the edge detection, a classical non-linear edge-preserving smoothing filter, the Kuwahara filter [9], was selected, with 5x5 mask size.

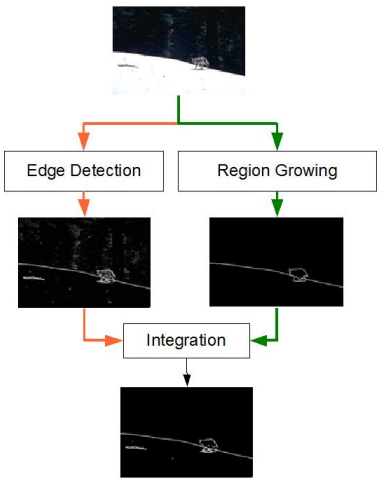


Figure 2. Data flow for the image segmentation process. The integration step receives the edge and region maps as inputs.

Finally, as RGB color images are considered, the three color channels are independently processed, and the results are added into one image.

III. The Multilabel Graph-cut

The boundary detection problem can be understood as the image segmentation considering multiple labels. Some techniques to address this problem are available in the literature [10]. Some of them that naturally combine edge information and region information are based on Markov Random Fields (MRF) models [11]. They assume that the segmentation S is guided by two kinds of constraints: boundary and region, in our case represented by the information provided by the edge detection and region growing technique, respectively. Such constraints can be integrated considering the image segmentation as an optimization problem, whose solution via multiway graph cut is the optimum labeling S.

Let $\mathbf{I}(\mathbf{x}) = \mathbf{z}$ denote an image that maps each pixel $\mathbf{x} \in \mathcal{X}$ to its color value $\mathbf{z}$. Let $S(\mathbf{x}) = f_{\mathbf{x}}$ denote the segmentation, where $f_{\mathbf{x}} = 0, 1, \cdots, l-1$ represents the labeling. Our goal is to solve the segmentation problem that assigns to each pixel $\mathbf{x}$ a label $f_{\mathbf{x}}$, where $f_{\mathbf{x}}$ is locally smooth and consistent with the observed data. Boykov et al. [12] address the image segmentation problem as an energy minimization problem, with the energy function to be minimized has the form

$$E(S) = E_{\text{smooth}}(S) + \lambda E_{\text{data}}(S), \tag{1}$$

where $E(S)$ measures the goodness of the segmentation S. $E_{\text{data}}(S)$, called the data term, measures how many pixels in $\mathcal{X}$ "looks like" the labels that S assigns to them. $E_{\text{smooth}}(S)$, called the smoothness term, measures the extent to which S is not smooth. The parameter λ controls the influence of neighbor pixels and, therefore, the smoothness of the segmentation result. $E_{\text{data}}(S)$ is often formulated as

$$E_{\text{data}}(S) = \sum_{\mathbf{x} \in \mathcal{X}} D(\mathbf{x}, f_{\mathbf{x}}), \tag{2}$$

where $D(\cdot)$ is the penalty for assigning to the pixel $\mathbf{x}$ the label $f_{\mathbf{x}}$. $E_{\text{smooth}}(S)$ is often formulated as

$$E_{\text{smooth}}(S) = \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}} V(\mathbf{x}, \mathbf{y}) \delta(f_{\mathbf{x}} \neq f_{\mathbf{y}}), \tag{3}$$

where $V(\mathbf{x}, \mathbf{y})$ is a penalty imposed by a segmentation boundary between $\mathbf{x}$ and $\mathbf{y}$ (also called smoothness cost), and $\delta(S(\mathbf{x}) \neq S(\mathbf{y})) = 1$ if $\mathbf{x}$ and $\mathbf{y}$ have different labels, otherwise it is 0. The neighborhood system $\mathcal{N}$ is given by the set of all neighbor pairs $\{\mathbf{x}, \mathbf{y}\}$ present in the standard 4-connected grid correspondent to the pixels of the image in the 2D plane.

The energy function defined in (1), considering the terms (2) and (3), is known as the Generalized Potts Model [13]. Boykov et al. [12] have shown how to solve the minimization of the generalized Potts Model by constructing a graph such that the minimum cut is the optimal solution. In addition, the minimum cut can be computed efficiently with an iterative procedure called alpha-expansion [12].

In this work $D(\mathbf{x}, f_\mathbf{x})$ is defined as $D(\mathbf{x}, f_\mathbf{x}) \propto -\log|p(\mathbf{I}(\mathbf{x})|f_\mathbf{x})|$, where $p(\mathbf{I}(\mathbf{x})|f_\mathbf{x})$ is the probability that a certain pixel $\mathbf{x}$ belongs to the label $f_\mathbf{x}$. The probability distribution is modeled using a multivariate normal distribution $\mathcal{N}\left(\mathbf{I}(\mathbf{x})|\boldsymbol{\mu}_{f_\mathbf{x}}, \boldsymbol{\Sigma}_{f_\mathbf{x}}\right)$ and their parameters are fitted using the information provided by region growing technique. That is, for each region, a multivariate normal distribution is defined and tuned with the pixels belonging to that region. Finally $D(\mathbf{x}, f_\mathbf{x})$ is equal to

$$D(\mathbf{x}, f_\mathbf{x}) = \frac{1}{2}(\mathbf{I}(\mathbf{x}) - \boldsymbol{\mu}_{f_\mathbf{x}})^T \boldsymbol{\Sigma}_{f_\mathbf{x}}^{-1}(\mathbf{I}(\mathbf{x}) - \boldsymbol{\mu}_{f_\mathbf{x}}).$$

In (3), the smoothness cost $V(\mathbf{x}, \mathbf{y})$ is defined considering the information produced by the edge detection technique, that is

$$V(\mathbf{x}, \mathbf{y}) = \begin{cases} v(\mathbf{x}), & \text{if } \mathbf{x} = [i, j]^T \text{and } \mathbf{y} = [i+1, j]^T \\ h(\mathbf{y}), & \text{if } \mathbf{x} = [i, j]^T \text{and } \mathbf{y} = [i, j+1]^T \end{cases},$$

where $v(\mathbf{x}) = e^{-\beta I_v(\mathbf{x})}$ and $h(\mathbf{x}) = e^{-\beta I_h(\mathbf{x})}$ when

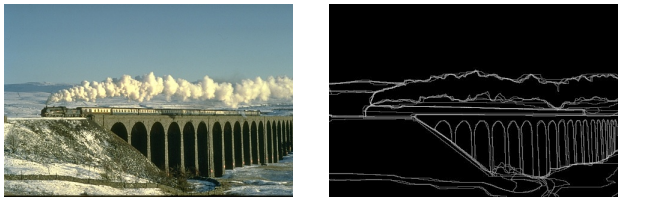
$$\begin{aligned} \nabla\mathbf{I}(\mathbf{x}) &= [I_v(\mathbf{x}), I_h(\mathbf{x})]^T \\ &= [\max_{c=\{R,G,B\}} I_{v_c}(\mathbf{x}), \max_{c=\{R,G,B\}} I_{h_c}(\mathbf{x})]^T \end{aligned}$$

The Sobel masks were used to calculate the gradient vector for each color channel (where the vertical and horizontal components of each color channel of the gradient vector is denoted by $I_{v_R}, I_{v_G}, I_{v_B}$ and $I_{h_R}, I_{h_G}, I_{h_B}$ respectively) and also was considered a $\beta = 5$.

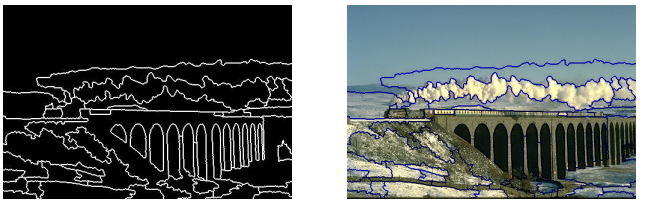
IV. The Proposed Method

The fully automated integration process KoSS is an extension of the KSS one [14]. In general, the conception of the algorithm has not changed, the logic is to eliminate or reduce false information and to emphasize strong edges. The key difference is: KSS performs a pixel-by-pixel analysis of the region-map, while KoSS analyzes each edge-line from the region-map (they deal with the same input images).

The improvement of KoSS over KSS is the elimination of some artifacts generated by the edge-map input. Figure 3 presents results of the segmentation of an Image got from the BSDS, for which the difference in the results of KSS and KoSS can be perceived. Such figure shows the original image (a), the human benchmark (b), the MM-Frac result (c), the KSS result (f) and the KoSS result (i). The false boundaries elimination can be observed in the edge-line in the sky over the smoke of Figure 3(d), the MM-Frac result over the original image, which is fully erased in the KoSS result shown in Figure 3(j),

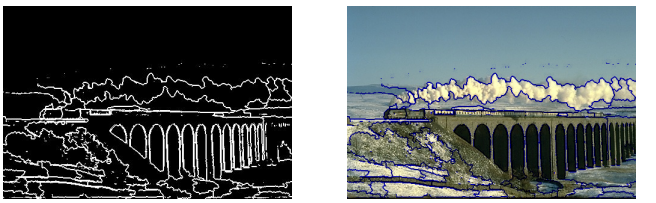


(a) Image 182053 of BSDS (b) Human benchmark (0.74)



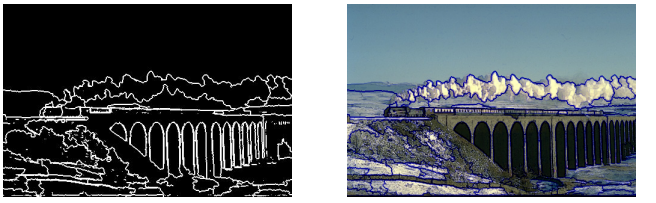
(c) MM-Frac result (0.69) (d) MM-Frac result over original image

(e) Zoomed part of MM-Frac result



(f) KSS result (0.70) (g) KSS result over original image

(h) Zoomed part of KSS result



(i) KoSS result (0.71) (j) KoSS result over original image

(k) Zoomed part of KoSS result

Figure 3. KSS versus KoSS: artifacts arise in KSS result, and not in KoSS.

the KoSS result over the original image. This result is more compatible with the human perception, shown in Figure 3(b) (in the KSS result, Figures 3(f) and 3(g), it is still possible to see some lost pixels in the sky zone). Figure 3(e) presents a zoomed part of the image in Figure 3(c), thus making easier to see this particular area. Figure 3(h) is a zoomed part of the image in Figure 3(f), showing exactly the same area shown in

Figure 3(e), whereas Figure 3(k) is a zoomed part of the image in Figure 3(i). Figure 3(h) shows the artifacts due to pixel-by-pixel processing, while Figure 3(k) does not present any pixel associated to the false edge. The number in parentheses is the F-measure value for the segmented image.

In the sequel, the KoSS algorithm is presented as a pseudo-code (see Algorithm 1), and after it is detailed. The post-processing integration method, the KoSS system, is independent of how the edge-map and the region-map are got. However, it is necessary that the region-map be a binary image and the edge-map be a soft map.

Algorithm 1 KoSS

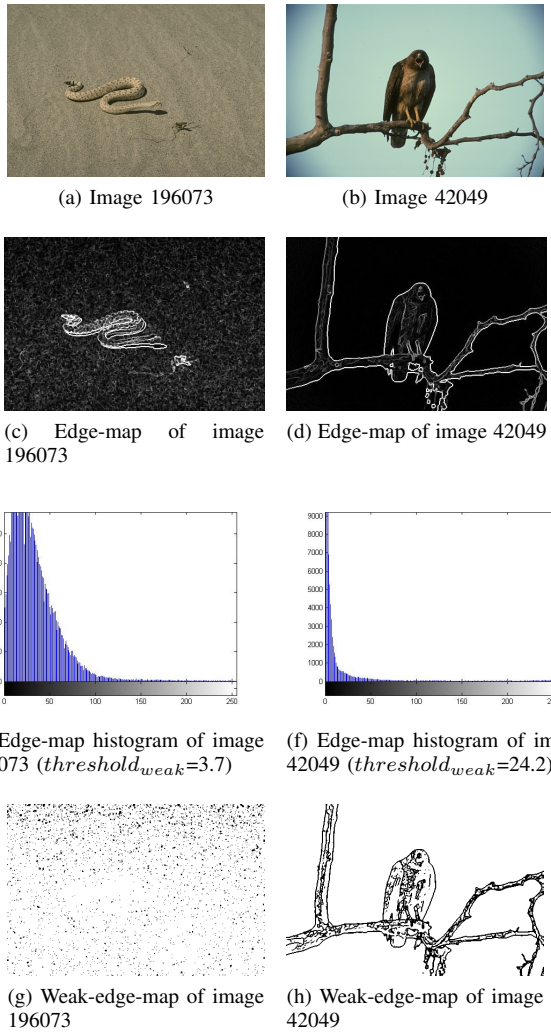
```
Inputs: edge-map and region-map
Build a weak-edge-map from edge-map
Build a list-of-edge-lines from region-map
{Part I}
for each edge-line in the list-of-edge-lines do
    set count-weak-edge = 0
    for each edge-unit in the edge-line do
        if (majority of neighborhood of edge-unit position in
        weak-edge-map) is marked then
            increment count-weak-edge
        end if
    end for
    if (count-weak-edge > length(edge-line)/3) then
        erase edge-line from region-map
    end if
end for
{Part II}
for each weak-edge-unit on weak-edge-map do
    if (weak-edge-unit position is not marked on region-
    map) then
        erase weak-edge-unit from edge-map
    end if
end for
Set image-result = adjustLimit(edge-map + region-map)
```

In step 2, we should detect the weak edges from the edge map. This step is basically a binarization process in the edge map, where each pixel with a low gray level value corresponds to a weak edge pixel. To automate the threshold value, we use the results of [15], where the threshold value is based on the histogram h of the edge-map, given by

$$threshold_{weak} = \frac{\sum_{i=0}^{50} h_i}{\sum_{i=50}^{200} h_i}, \quad (4)$$

where i = [0, 255] is the value of a pixel in a gray-scale image.

A noisy edge map will result in low $threshold_{weak}$ values, while a strongly defined edge map will result in high $threshold_{weak}$ values. Figure 4 presents two examples with images got from BSDS: the first line shows the original image, the second line shows the associated edge-maps and the third line shows the histogram associated to the edge-maps. Using equation (4), $threshold_{weak}$ for the edge-map of image 196073 is 3.7 and $threshold_{weak}$ for the edge map of image 42049 is 24.2. The weak edge maps are presented in the fourth line in Figure 4.



(a) Image 196073 (b) Image 42049

(c) Edge-map of image 196073 (d) Edge-map of image 42049

(e) Edge-map histogram of image 196073 ($threshold_{weak}$=3.7) (f) Edge-map histogram of image 42049 ($threshold_{weak}$=24.2)

(g) Weak-edge-map of image 196073 (h) Weak-edge-map of image 42049

Figure 4. Examples of low $threshold_{weak}$ and high $threshold_{weak}$.

As Figure 4(h) shows, for higher $threshold_{weak}$ values there are more weak edges (white pixels in an image), and thus more information from region map will be eliminated in Step 1 of KoSS. For images with a low $threshold_{weak}$, there are a few weak edge pixels, resulting in images with many black pixels, as in Figure 4(g). So, when the image is noisy, most information from the region map is preserved. Therefore, equation (4) represents the degree of confidence of the edge detection result, pointing when edge map information is more reliable than region map information.

In Step 3 of Algorithm 1, the region map is divided in a list of edge lines. Actually, the region map can be viewed as an skeleton, whose elements can be classified as end points, normal points and branch points [16]. In a 3x3 neighborhood, end points have only one neighbor element, normal points have exactly two and branch points have more than two. An edge-line (or skeleton branch) is a subset of the skeleton entirely

consisting of normal points except for the extremes, that are end points or branch points.

Algorithm 2 shows the code correspondent to such step, in Matlab®. In line 1, the goal is to find all branch points in an image, and erase all of them (see step 2 of Algorithm 2), thus resulting an image having a set of edge-lines with only end points as extremes. The result of line 3 is the variable **lineStats**, containing the list of edge-lines.

Algorithm 2 Step 3 of Algorithm 1 in Matlab®

```
1: Ipoints = bwmorph(MMFracImage, 'branchpoints', 1);
2: MMFracImage(Ipoints) = 0;
3: lineStats = regionprops(MMFracImage, 'PixelList', 'Pix-
   elIdxList');
```

In the rest of Algorithm 1, the logic is to eliminate or to reduce false information. Part I (lines 4-14) eliminates edge-lines which are considered weak by the edge detection. The condition of line 11 was empirically defined after analyzing the results of several experiments. In Part II (lines 15-19), the weak information of the edge map is eliminated. All pixels not belonging to an edge line on region-map and considered weak edge on weak-edge-map are erased. In step 20, the sum operation will enhance all boundary pixels that match in the two different processed maps.

Thus, KoSS erases some edge information of the region map and does not preserve weak information for the edge map. The result seems cleaner, preserving and enhancing only the strong edges of both maps.

V. Experimental Results

We tested both methods, KoSS and multilabel graph cut, with natural colored images provided by the BSDS image data set, applying it to all one hundred (100) images of the test data set of BSDS. As discussed so far, our experiments do not include any parameter-tuning for individual images.

Quantitatively speaking, the metrics recall, precision and F-measure of each method computed by the BSDS method are shown in Table I. KoSS and multilabel graph cut improves the recall metric when comparing with input metrics (MM-Frac and edge detection metrics). Moreover, KoSS performs better then multilabel graph cut. On the other hand, multilabel graph cut is better in terms of precision. Edge detection presents the lowest value, because of the noisy pixels. KoSS and multilabel graph cut present the same F-measure values. In addition, Table II shows the average value of the same three metrics, considering all the one hundred (100) images available in the test dataset of BSDS. From such average values, the advantage of the KoSS method becomes clearer: its average F-measure value is 0.63, which is the closest value, considering the average F-measure correspondent to the human perception.

In terms of qualitative comparison, Figure 5 shows some images for which the multilabel graph cut method outperforms the KoSS method (its results are closer to the human perception), while Figure 6 shows examples in which the results for the KoSS method are closer to the human perception. The column order in both figures is maintained: (a) shows the input image, (b) the human benchmark and the segmentation result for (c) JSEG, (d) MM-Frac, (e) edge detection, (f) multilabel graph cut and (g) KoSS approach, already binarized using the best threshold computed by BSDS. Each result has its computed F-measure metric.

An important characteristic of multilabel graph cut is its capacity to change the initial labeling considering the edge detection information. That is, the labeling generated by the region growing technique is the initial state of solution in a discrete optimization process, in special in the images 304034, 160068 and 101085 of Figure 5. Additionally, KoSS method eliminates false information and enhances edge matching in both maps, but it does not change the boundaries positions, as multilabel graph cut does.

In all the images of Figure 5, KoSS flaws were over-segmentation. In the fifth column, the results of edge detection are presented. The results are very noisy and this is mainly due to the fact that edge detection techniques rely entirely on the local information available in the image. So, KoSS does not erase false edge lines from region map, as the grass in the first line, for instance, or the lines caused by wall texture in the background in the third line.

At the same time, the method of edge detection is responsible for highlighting details such as the stick in the left, in the snow area of the fourth line of Figure 6 (image 167062), the insect near to the snake in the second image of Figure 6 (image 196073) and the lines on the ground of image 37073 (first line of Figure 6). Details detected using edge detection method are kept, but the noise was attenuated and disappears after the binarization computed by BSDS. The noise caused by sand in image 196073, the koala's fur (image 69015) and

Table I
Precision, Recall and F-measure metrics calculated by BSDS.

	Recall	Precision	F-measure
JSEG	0.61	0.56	0.59
MM-Frac	0.63	0.56	0.59
Edge Detection	0.65	0.49	0.56
KoSS	0.69	0.54	0.61
Multilabel GraphCut	0.66	0.58	0.61
Human	0.70	0.89	0.79

Table II
Precision, Recall and F-measure metrics calculated by simple average.

	Recall	Precision	F-measure
JSEG	0.61	0.57	0.57
MM-Frac	0.63	0.57	0.59
Edge Detection	0.69	0.55	0.59
KoSS	0.73	0.57	0.63
Multilabel GraphCut	0.65	0.57	0.60
Human	0.70	0.89	0.78

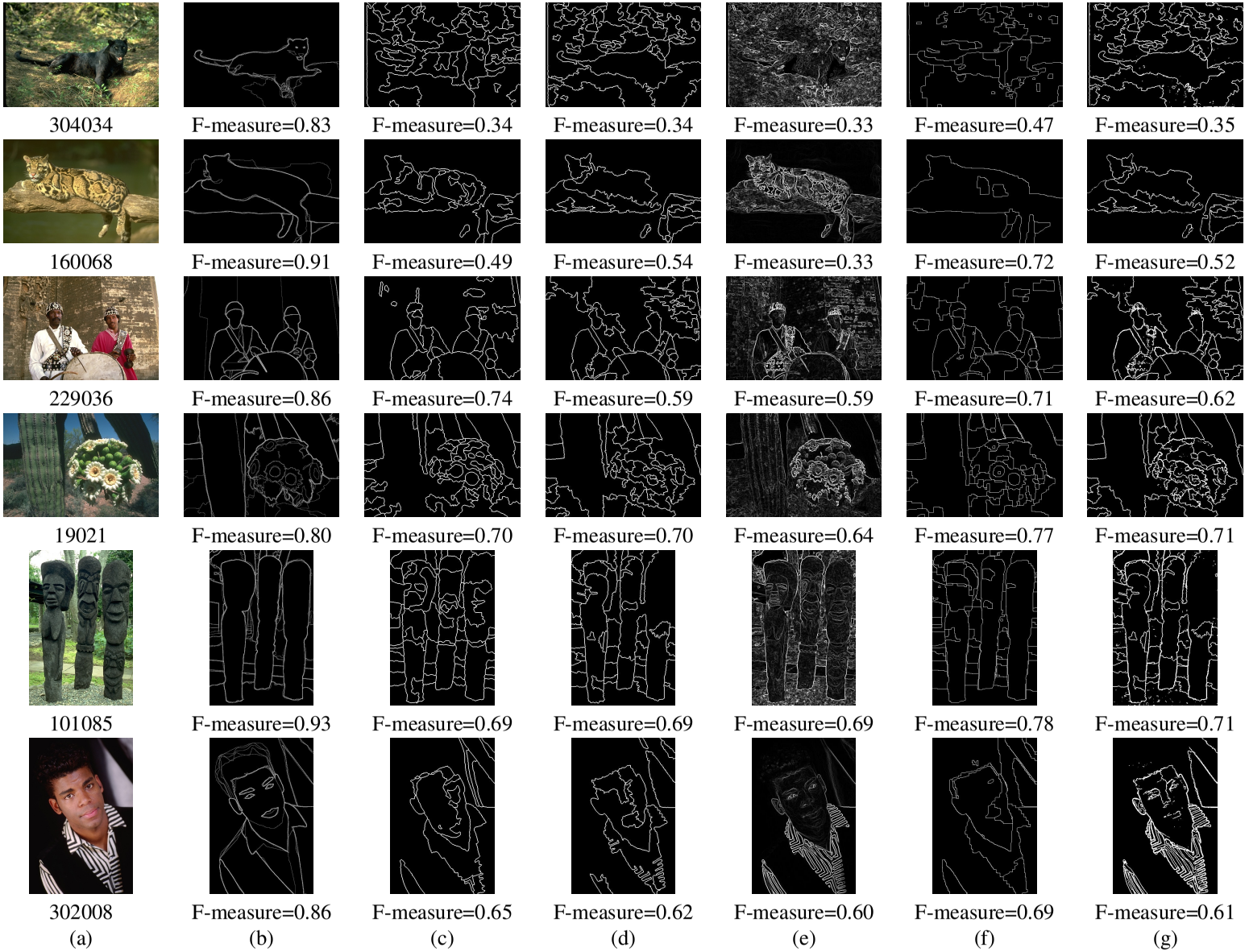


Figure 5. (a) Original image (b) Human benchmark (c) Results of JSEG method (d) Results of the MM-Frac method (e) Results of the edge detection method (f) result of multilabel graph cut and (g) result of KoSS

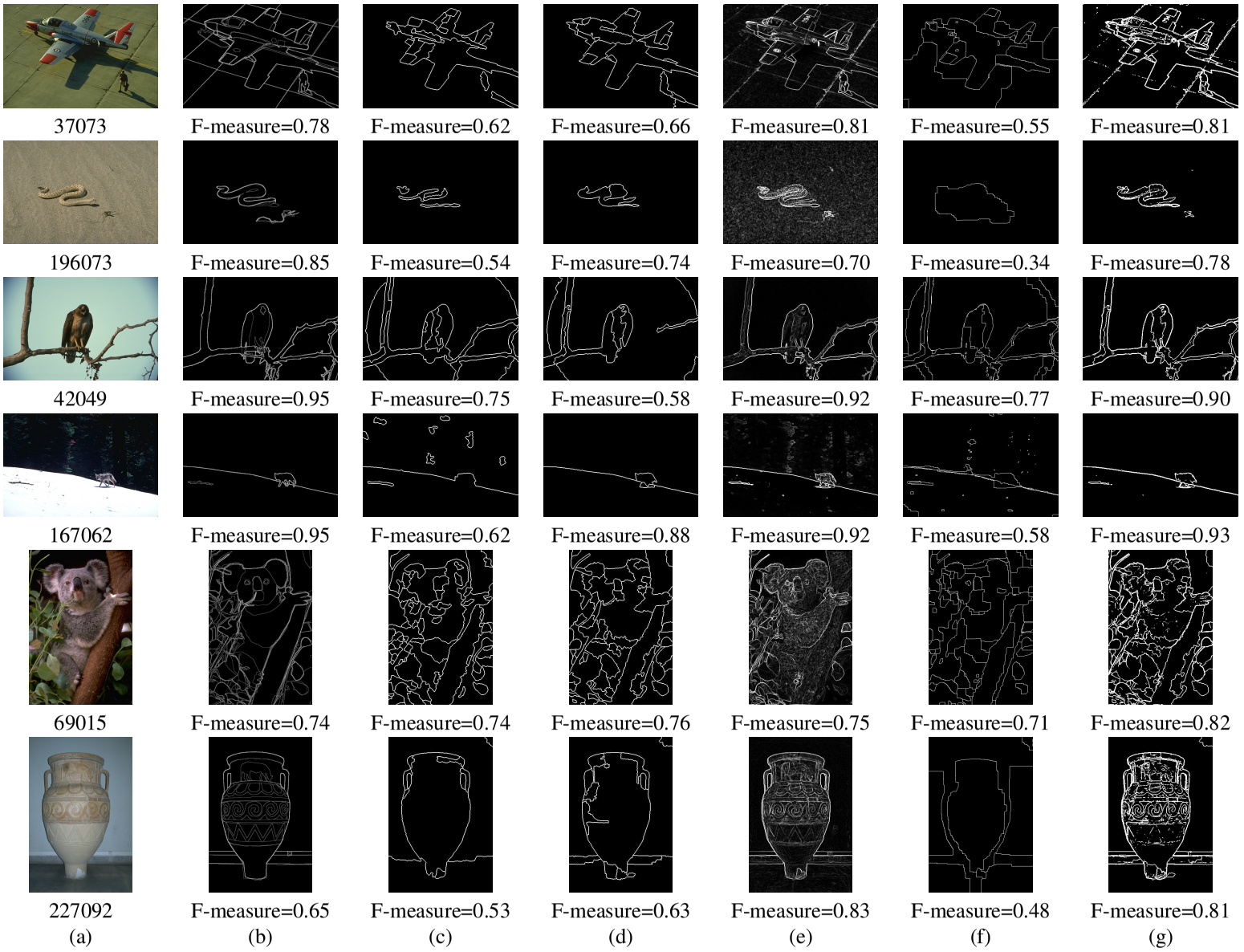


Figure 6. (a) Original image (b) Human benchmark (c) Results of JSEG method (d) Results of the MM-Frac method (e) Results of the edge detection method (f) result of multilabel graph cut and (g) result of KoSS

the trees in the background of image 167062 are erased in the final KoSS result. Thus, the boundaries are more accurate and are closer to the human perception.

In all images, a detail that is more noticeable in Figure 6, the multilabel graph cut results present blocking artifacts, because each image is represented by a 4-connected lattice [17]. On the other hand, the image 37073 show over-segmentation problems, because the edge information (the boundary between airplane and its shadow, for instance) is not sufficiently defined for that algorithm, which is not well-suited to segment thin objects (see image 196073).

Deng and Manjunath [5] pointed out that the major problem they observed in JSEG result is caused by the varying shades due to the illumination. Even integrating with multifractal measurement, this problem still remains [18]. For instance, the color of the sky can vary in a very smooth transition, as in the image 42049, the third line of Figure 6, where there is no clear boundary in the sky. Indeed, the human perception does not consider this smooth color variation as a different region. The result of KoSS also does not present this false boundary. The smooth variation is not perceived by the edge detection, and then the boundary is erased by the KoSS method. On the other hand, the result of the Multilabel Graph-cut method shows the circular boundary.

VI. Conclusion and Future Work

This work proposes a post-processing integration method, for which the main goal is to integrate the region-growing result and edge information, in an image segmentation procedure. Our strategy, called KoSS, is to put together the two maps, eliminating the false boundaries in region-map, based on edge information, and eliminating the noisy edges in the edge-map, based on region information. The KoSS algorithm works well and solves the problem of false boundaries pointed out in other works. Furthermore, all strong edges of both input maps are held, improving the boundary detection. Unfortunately, the KoSS results still present broken edges, not keeping the contour closed.

As our results show, the KoSS approach improves the sensitivity to boundary regions, thus providing boundary detection results that match human perception better than the results associated with the methods used to generate its two input images.

A second goal of the paper is to compare our results with the results of other method with different philosophy, dealing with the same information input. Even when the human beings who generated the ground truth for the database used agree in terms of edges, they often disagree when precisely defining which pixels in an image correspond to such edges. Boundary detection is a hard problem and the comparison with other methods is even harder. Although BSDS metrics assist you when comparing the approaches, sometimes a single number is not enough to give all information about the performance of a method. For instance, despite the overall F-measure of multilabel graph cut and KoSS were equal, in some cases, the multilabel graph cut showed some outliers, which is not good in terms of segmentation.

Although the goal is not computational performance, it is worth noting that when we implemented KoSS in Matlab it was fast, requiring approximately 0.5 seconds for each BSDS image. Moreover, KoSS is easy to reproduce, fully automated, and does not require a training stage (is an unsupervised method).

Acknowledgments

The authors thank CAPES and CNPq, respectively, for the scholarships of Mrs. Komati and Mr. Samatelo. They also thank PPGEE (UFES), for supporting research development.

References

[1] 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. of the 8th IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423.

[2] X. Muñoz, J. Freixenet, X. Cufí, and J. Martí, "Strategies for image segmentation combining region and boundary information," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 375–392, 2003.

[3] D. R. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, 2004.

[4] K. S. Komati, E. O. Salles, and M. Sarcinelli-Filho, "Two-level strategy for image boundary detection," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP2011)*, March 2011, pp. 181–186.

[5] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, August 2001.

[6] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 1, pp. 72–77, 1995.

[7] A. Torralba and A. Oliva, "Statistics of natural image categories." *Network: Computation in Neural Systems*, vol. 14, no. 3, pp. 391–412, 2003.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., 2001.

[9] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita, *Digital processing of biomedical images*. Plenum Press, 1976, pp. 187–203.

[10] H. Zhang, J. Fritts, and S. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, May 2008.

[11] S. Z. Li, *Markov random field modeling in image analysis*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[12] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[13] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 2, pp. 147–59, February 2004.

[14] K. S. Komati, E. O. Salles, and M. Sarcinelli-Filho, "Kss: Using region and edge maps to detect image boundaries," *Computing in Science and Engineering*, vol. 13, pp. 46–52, 2011.

[15] O. Rotem, H. Greenspan, and J. Goldberger, "Combining region and edge cues for image segmentation in a probabilistic gaussian mixture framework," in *CVPR*, 2007.

[16] D. Attali, G. Sanniti di Baja, and E. Thiel, "Skeleton simplification through non significant branch removal," *Image Processing and Communications*, vol. 3, no. 3-4, pp. 63–72, 1997.

[17] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03)*, vol. 2, 2003, pp. 26–33.

[18] K. S. Komati, E. O. T. Salles, and M. S. Filho, "Fractal-jseg: Jseg using an homogeneity measurement based on local fractal descriptor," in *Proceedings of SIBGRAPI*, 2009, pp. 253–260.