A CONTRARIO HIERARCHICAL IMAGE SEGMENTATION

Juan Cardelino, Vicent Caselles, Marcelo Bertalmío

Dept. Tecnologies de la Informació
Universitat Pompeu Fabra
Barcelona, Spain
{juan.cardelino,vicent.caselles}@upf.edu
marcelo.bertalmio@upf.edu

Gregory Randall

Inst. Ingeniería Eléctrica
Universidad de la República
Montevideo, Uruguay
randall@fing.edu.uy

ABSTRACT

Hierarchies are a powerful tool for image segmentation, they produce a multiscale representation which allows to design robust algorithms and can be stored in tree-like structures which provide an efficient implementation. These hierarchies are usually constructed explicitly or implicitly by means of region merging algorithms. These algorithms obtain the segmentation from the hierarchy by either using a greedy merging order or by cutting the hierarchy at a fixed scale.

Our main contribution is to enlarge the search space of these algorithms to the set of all possible partitions spanned by a certain hierarchy, and to cast the segmentation as a selection problem within this space. The importance of this is two-fold. First, we are enlarging the search space of classic greedy algorithms and thus potentially improving the segmentation results. Second, this space is considerably smaller than the space of all possible partitions, thus we are reducing the complexity.

In addition, we embed the selection process on a statistical *a contrario* framework which allows us to reduce the number of free parameters of our algorithm to only one.

***Index Terms*— Image segmentation, Hierarchical systems, Statistics**

I. INTRODUCTION

Image segmentation is one of the oldest and most challenging problems in image processing. Given an image, even for a human observer, it is hard to determine an unique partition of the image, and it is even harder to find consensus between different observers.

A usual approach taken to overcome this difficulty is to find a hierarchy of segmentations rather than an unique partition. These hierarchies are usually constructed in a bottom-up fashion, by using region merging algorithms.

The first merging algorithms presented in the literature (see [1]) aimed to consecutively merge adjacent regions until a stopping criterion was met, thus yielding a single partition. All these algorithms have three basic ingredients: a region model, which tells us how to describe regions; a merging criterion, which tells us if two regions are to be merged or not; and a merging order, which tells us at each step which couple of regions should be merged first. Many of the existing approaches present similar problems: they use only region information (no boundaries are taken into account), they have a considerable number of manually tuned parameters, they use very simple region models, or they use a very simple merging order. Most of the recent work in this area has been directed to improve the merging criterion and the region model, but little effort was carried towards the merging order and the reduction of the number of parameters.

Regarding the merging order, Calderero et al. [2] presented an improved merging algorithm which uses a non-parametric region model and a merging order depending on the scale. However, this work still has some free parameters and only one feature (color) is taken into account in the merging criterion.

On the other hand, Burrus et al. [3] introduced an *a contrario* model to remove the free parameters and a criterion combining different region descriptors. However, they require a training process which is offline but very slow, and has to be done for each image size and for each combination of the parameters of the initialization algorithm. In addition, segmentation results are dependent on those initialization parameters.

Instead of working at a single scale, a more robust approach is to use the intermediate process of merging to build a hierarchy of partitions, which is often represented as a binary tree. After the hierarchy is built, the tree is pruned according to some scale parameter ([4]). However, most algorithms ignore the fact that this hierarchy spans a subset of all the possible partitions of the image, and the segmentation problem then could be cast as the selection of an optimal partition on this subset. In this work we use this approach to overcome some of the usual problems of merging algorithms. In addition, our algorithm uses both boundary and region information within a statistical framework which allows us to select the optimal partition with only one free parameter.

The rest of the paper is organized as follows. Our algorithm is described in section II. Section III shows the experimental results. Finally, in section IV we give some conclusions and future work.

II. OUR ALGORITHM

A hierarchy of partitions $\mathcal{H}$ is a multi-scale representation of many possible segmentations of the image. For our purpose, it is a tree structure where each node represents a region, and the edges represent inclusion between regions. This tree is often constructed by means of a bottom-up merging algorithm. We start with an initial set of regions, often called seeds, which conform the finest possible partition. Then we iteratively merge two or more adjacent regions and represent the resulting region on the graph as a new node which is the father of the merged regions. One of the most popular ways to merge regions is to minimize the well-known Mumford-Shah (M-S) functional [5]. The binary tree resulting of this merging procedure is often called binary partition tree.

A contrario models where introduced by Desolneux et al. in [6], within the framework of the so-called Computational Gestalt Theory (CGT). Given a set of objects the aim is to find the *meaningful* ones, according to the so-called Helmholtz principle, which states that: “we immediately perceive whatever could not happen by chance”. To apply this principle, CGT algorithms rely on a *background* or noise model, and then define the interesting objects as large deviations from this model. To measure this deviation they compute the so-called *Number of False Alarms (NFA)*, which is an upper bound on the expected number of occurrences of a certain event under the background model. If this expectation is low, it means that the considered event is meaningful and could not arise by chance. Given an event $\mathcal{O}$, NFAs are computed as $NFA(\mathcal{O}) = N_{tests}P(\mathcal{O})$, where N_{tests} is the number of possible events in the set. Defined in this way, it can be proven that if $NFA(\mathcal{O}) < \epsilon$, then the expected number of detections in an image of noise generated by the background model is lower than ϵ.

The main idea of our algorithm is to use region and boundary information to select the optimal partition from the set of all partitions spanned by the given hierarchy. This selection is embedded into an *a contrario* model which states that a region is meaningful if its gray level is homogeneous *enough* and its boundary is contrasted *enough*.

II-A. Region term

To define this term, we first need to define what a *meaningful region* is. As we are working with the Mumford-Shah model, we will say that a region is meaningful when its *error* is small, in a similar way to [7]. To start, lets us review the data term in the M-S model for a single region:

$$E_R = \sum_{x \in R} (I(x) - \mu_R)^2 \tag{1}$$

where μ_R is the mean gray value of region R. This can be seen as the L_2 error when we approximate each pixel of the region by μ_R. We can also define the pixel-wise error as $e_R(x) = (I(x) - \mu_R)^2$, so with this notation the error becomes $E_R = \sum_{x \in R} e_R$.

If we consider that E_R is a random variable generated by the background model and we define $\hat{E}_R$ as the observed region error we can define the number of false alarms of a region as

$$NFA_r(R) = N.P(E_R < \hat{E}_R) \tag{2}$$

where N is the number of regions to consider. This NFA measures the goodness of fit of the region pixels to the model given by the M-S mean μ_R. If the probability P is very low, it means that the error is extraordinarily small and could not arise by chance. Thus, the NFA is small and the region will be marked as *meaningful*.

In order to complete the definition of (2) we need to compute the probability $P(.)$ of the error in each region. We do this in two steps, first we estimate the density function $p(e)$ of the pixel-wise error $e(x)$. After that, we compute the probability that region R has error less or equal than $\hat{E}_R$.

Note that the error $\hat{e}_R(x)$ depends on μ_R which in turn depends on the region the pixel belongs to. Before the segmentation starts, we don't know to which region the pixel will be assigned, so we can't compute this quantity. To overcome this, we can compute the error with respect to all the possible regions the pixel x could possibly be assigned to. In this way, we are not computing the error with respect to a single partition, but to all the possible partitions spanned by $\mathcal{H}$.

To compute $P(E_R < \hat{E}_R)$ we make the assumption of independence between pixels. Thus, looking at equation (2), the random variable E_R is a sum of n independent and identically distributed random variables e_R. The probability can be then approximated (for large n values) by a normally distributed random variable, using the Central Limit Theorem (CLT). In practice, with $n > 20$ the gaussian approximation of the CLT is very accurate.

II-B. Meaningful regions vs meaningful mergings

From our definition of meaningful region, it can be seen that the NFA associated with each region depends on the given initial partition. Given a certain tree, it is a common practice to remove the leaves with small area, to reduce the computational burden of the algorithm. Usually, leaves have small errors, so removing one of them will decrease the probability of observing a small error. If we remove a big number of small leaves we will make the small errors less probable, thus making all nodes more meaningful. This is not a desirable behavior, because we want the result of our algorithm to be independent of the pre-processing performed.

Up to this point, our definition of meaningful region is absolute, which makes it strongly dependent on the histogram of the error. To overcome this problem we introduce a similar concept but applied to mergings. We will say that a certain merging is meaningful if it *improves* the previous representation. That is, if the meaningfulness of the merged region is greater than the one of the two separate regions.

To compute the meaningfulness of a merging we will need to compute two quantities: the NFA of the union of two regions, and the NFA of the separate existence of the two original regions. To compute the first quantity, we can apply the definition of the previous sections. However, we need to adjust the number of tests, because now we are not testing all possible regions, but only those created as a result of a union. Let R_1 and R_2 be the regions to be merged and $R_u = R_1 \cup R_2$. Thus, the NFA of the union is:

$$NFA_r(R_1 \cup R_2) = N_u.P(E_{R_u} < \hat{E}_{R_u}) \tag{3}$$

where N_u is the number of possible unions in the tree, which is exactly $\frac{N}{2}$. Here, as we modeled the union as a single region, the error $\hat{E}_{R_u}$ is computed using the mean μ_u of the union.

Now we need to consider the existence of two separate and independent regions R_1 and R_2. As they are different regions, we have a different model for each region, which are the means μ_1 and μ_2, and the corresponding errors $\hat{E}_{R_1}$ and $\hat{E}_{R_2}$. As the M-S error is additive, we can consider that the total error of approximating both regions by their means is $\hat{E}_{R_1;R_2} = \hat{E}_{R_1} + \hat{E}_{R_2}$. Thus we can define the NFA as

$$NFA_r(R_1; R_2) = N_c.P(E_{R_1;R_2} < \hat{E}_{R_1;R_2}) \tag{4}$$

where N_c is the number of possible couples (R_1, R_2) which is also $\frac{N}{2}$. As the involved quantities are probabilities which could take very small values, it is usual to work with the logarithm of the $NFAs$. Thus, we will say that a merging is meaningful if

$$\mathcal{S}_r = \log NFA_r(R_1 \cup R_2) - \log NFA_r(R_1; R_2) < 0 \tag{5}$$

II-C. Boundary term

Regarding region boundaries, we propose to merge two regions when the boundary between them is not contrasted enough. In addition, we want to obtain the regularizing effect of the boundary term in the M-S functional, which favors short curves. For this reason, we use a definition similar to the one by Cao et al. [8], in the sense that we say a curve is meaningful if it has an *extraordinarily* high contrast. However, we also penalize long curves by using the accumulated contrast along the curve instead of the minimum as in Cao's model. Thus, we define *meaningful regions* as those having a short and contrasted boundary. To measure the length of the curves we use the geodesic curve length

$$L(\Gamma) = \int_{\Gamma} l(x(s))ds \tag{6}$$

where $l(x) = g(|\nabla I(x)|)$ is a pixel-wise contrast detection function. Here $g(x)$ yields small (near 0) values in well contrasted pixels and values near 1 in the low contrasted regions. For the background model, we use the histogram of the gradient as proposed in [8]. From the image, we can obtain the empirical distribution of the gradient norm, or the distribution of $l(x)$ which is a function of $|\nabla I|$. However, we need to compute the distribution of the sum over all the pixels of the curve, so our new random variable will be $\mathcal{L} = \sum_{x \in \Gamma} l(x)$. As we did in section II-A we can compute the probability by means of the CLT from the distribution of l, and define the NFA of a curve Γ as

$$NFA_b(\Gamma) = N_{curves}.P(L < \hat{L}) \tag{7}$$

where N_{curves} is the number of curves tested.

II-D. Combination of terms

Using the two previous definitions of meaningful regions, we would like to develop an unified notion which is able to take into account both definitions at the same time. To achieve this we propose to compute the following quantity:

$$NFA_j(R) = N.P\left(E_R < \hat{E}_R; L_{\partial R} < \hat{L}_{\partial R}\right) \tag{8}$$

where N is the number of tested regions and P is the joint probability of the region having a small error and a contrasted boundary at the same time. A way to make this model computable is to make the (strong) assumption of independence between boundary and region terms. This allows us to factorize P and write the new NFA as

$$NFA_j(R) = N.P(E_R < \hat{E}_R).P(L_{\partial R} < \hat{L}_{\partial R}) \tag{9}$$

Taking this into account, we can define a meaningful merging, using region and boundary information, as

$$\mathcal{S}_j = \log NFA_j(R_1 \cup R_2) - \log NFA_j(R_1; R_2) < 0 \tag{10}$$

Definition (10) allows us to construct a parameterless algorithm; however, in practice we verified that our algorithm tends to oversegment images. The explanation of this phenomenon relies on our estimation of the number of tests. This estimation is very hard to compute analytically, so we used a very rough estimate ($N_u \approx N_c$). To overcome this problem, we consider an alternate formulation based on the following observation:

$$\mathcal{S}_j = \log N_u + \log P(R_1 \cup R_2) - \log N_c - \log P(R_1; R_2) \tag{11}$$

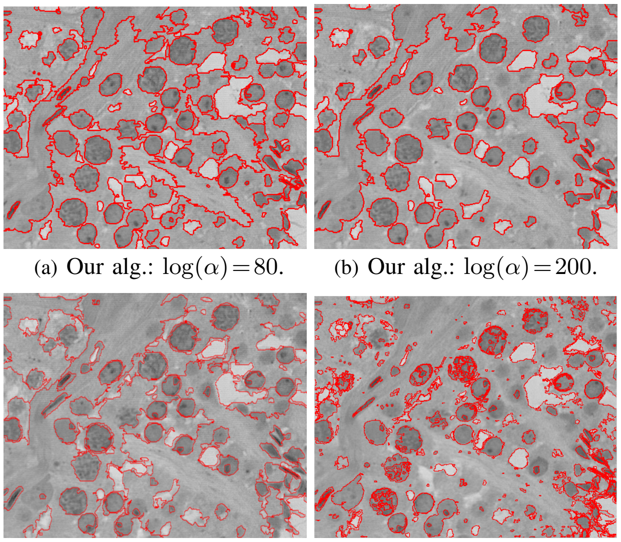
where we expanded equation (10) to explicitly show the number of tests. If we do not want to estimate both numbers of tests, we can merge them into a single variable called α. From this point of view, our definition of meaningful merging becomes:

$$\mathcal{S}_j = \log P(R_1 \cup R_2) - \log P(R_1; R_2) < \log \alpha(R_1; R_2) \tag{12}$$

where $\alpha(R_1; R_2) = \log(\frac{N_c}{N_u})$. For the sake of simplicity, we assume that α is a constant value for every couple of regions (R_1, R_2). At the moment this parameter is set manually, but it could be estimated as in [3].

II-E. Implementation

The computational cost of the algorithm can be roughly divided in two parts: computing the tree and selecting the meaningful regions on it. A usual way to construct the tree is to use all pixels as the initial partition. So, for a N pixel image, the tree will have $2N$ nodes. The computational cost of the second part is proportional to the number of nodes on the tree, so we can reduce the computational cost by pruning the tree. In practice we rarely segment regions of small area, thus we remove the lower nodes of the tree corresponding to the smaller regions. This is performed by pruning the tree with a fixed (and small) value of λ which ensures that no important regions are lost.


(a) Our alg.: $\log(\alpha) = 80$. (b) Our alg.: $\log(\alpha) = 200$.

(c) Burrus: $k = 150, \sigma = 0.8$. (d) Cao: $\log(\epsilon) = -3$.

Fig. 1. Comparison of results over the *Ram* image. (a), (b) results of our algorithm with different parameter values. (c) Burrus et al. [3]. (d) Cao et. al [8].

For example, in the *Ram* image (fig. 1) the total number of pixels is 200901, but using $\lambda = 50$ we have to process only 22397 regions, which results on a 90% reduction on the computational cost. In spite of this pruning, the resulting image still retains enough level of detail and no important objects are lost.

III. RESULTS

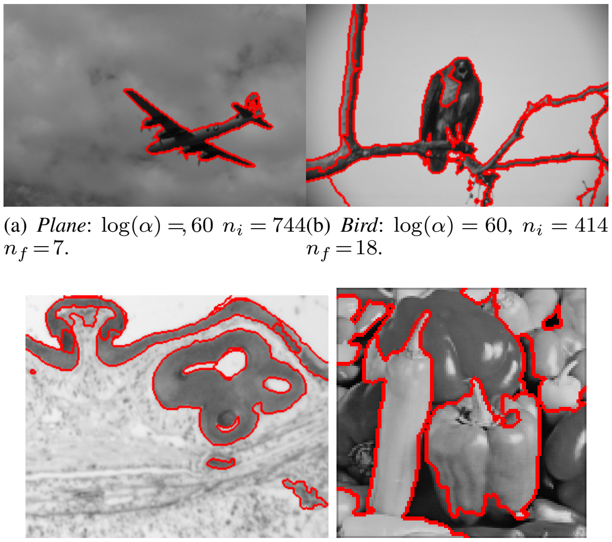
The only free parameter to be set is the threshold α on the significance. There is also the initial λ but for small values ($<= 50$) the results are independent of this parameter, so it is not tuned at all.

In figure 2 we show the results in four examples[1]. The results are quite good in general, but there are some exceptions that we discuss in the following. In the *Plane* image the tail is slightly oversegmented; and in the case of *Bird*, the objects are even more oversegmented because of the non-uniform illumination in the objects and the background. In the case of *Spider*, all objects are correctly detected, but some regions are slightly oversegmented. On the *Peppers* image, the segmentation is coarse with some of the dark regions between the peppers missing. However, our algorithm is able to perform quite well in all the examples without changing the parameter.

To show the effect of the parameter α, we ran two examples varying its value. As figures 1(a) and 1(b) show, α controls the quantity of objects detected. A large value implies that the algorithm will merge many objects, so it will tend to have less false positives. When α is reduced we are able to detect more objects but also more false positives appear. In general we observed that our parameter presents less variability than the λ of M-S. For instance, to obtain a good segmentation of the *Church* image a $\lambda \approx 1000$ is required, whereas to segment *Plane* a value of $\lambda \geq 5000$ is required. On the other hand, with our algorithm the value is unchanged.

In figures 1 and 3 we compare our results with some related approaches. In 3(a) we show the result of a region merging

[1] An extensive evaluation and quantitative results can be found at: http://iie.fing.edu.uy/rosaluna/wiki/ImageSegmentationAlgorithms



(a) *Plane*: $\log(\alpha)$ =, 60 $n_i = 744$ $n_f = 7$.

(b) *Bird*: $\log(\alpha) = 60$, $n_i = 414$ $n_f = 18$.

(c) *Spider*: $\log(\alpha) = 60$, $n_i = 5858$ $n_f = 31$.

(d) *Peppers*: $\log(\alpha) = 60$, $n_i = 2737$ $n_f = 9$.

Fig. 2. Results of our algorithm over some test images. n_i and n_f are the initial and final number of regions.

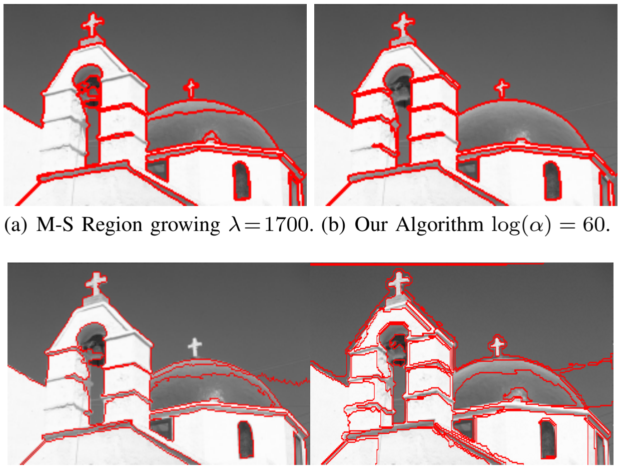
(RM) algorithm with a M-S model. Here we can see that our algorithm is able to work across scales giving in general a good segmentation of objects present in different scales of the image. In the RM approach, to detect the cross (on the dome) we need to fix a small λ, which in turn oversegments the dome; but if we increase λ to correct this, we miss both the dome and the cross. In 1(c) we show the result of Burrus et. al [3]; in this case the results are quite good but again restricted to one scale. Many of the light objects are lost while some small dark spots are detected inside the blobs. In addition, in figure 3(d) we show another result of Burrus et al. where again, no objects are missing but many are oversegmented.

In the case of Calderero et al. [2], shown in figure 3(c), as they penalize out-of-scale objects, they tend to lose small structures like the cross on top of the dome, and oversegment the dome. The last approach shown (fig. 1(d)) is that by Cao et al. [8]. As they only use the minimum contrast along the boundaries, many shapes which have low contrasted boundaries are lost. This confirms the importance of using region based information.

Regarding the computational cost, our algorithm takes 3 seconds to compute the tree and 10 seconds to process the *Ram* image (400x500) on a 2.0GHz Athlon 64 CPU with C code and optimization flags turned on. However this could be further improved with a careful implementation. To complete this evaluation, we compared execution times on the *Ram* image with the rest of the discussed approaches: Burrus takes 117 minutes to train and 10 seconds to segment; Cao takes 2.5 seconds; and the M-S region merging algorithm takes 6.5 seconds.

IV. CONCLUSION AND FUTURE WORK

In this work we presented a novel segmentation approach based on a hierarchy of partitions of an image. We introduced an *a contrario* statistical model based on the combination of region and boundary based descriptors, which allows us to validate the mergings present in the hierarchy. In this way we reduce the number of free parameters and alleviate the problem of local minima in greedy algorithms. The results obtained by our algorithm are comparable to those presented in the literature. In addition, our parameter has a clear meaning and reduces the variability compared with the λ of the M-S functional.



(a) M-S Region growing $\lambda = 1700$.

(b) Our Algorithm $\log(\alpha) = 60$.

(c) Calderero: $\alpha = 0.15$ $nbin = 10$.

(d) Burrus: $k = 150, \sigma = 0.8$.

Fig. 3. Comparison of results over the *Church* image.

However, our approach has two main drawbacks. First, it suffers from some locality, due to the fact that we validate one couple of regions at a time. This could be solved by validating complete partitions instead of pairs of regions. Second, we still have a free parameter coming from the difficulties of analytically computing the number of tests. One possible way to remove this parameter is to use a simulation procedure like the one presented by Burrus et al. [3].

V. ACKNOWLEDGMENTS

The authors acknowledge partial support by PNPGC project, ref. MTM2006-14836. V. Caselles also acknowledges partial support by "ICREA Acadèmia" for excellence in research funded by the Generalitat de Catalunya. We thank N. Burrus for kindly sharing and explaining the code of his algorithm to us; and we also thank F. Calderero for sharing his results with us.

VI. REFERENCES

[1] J. M. Morel; S. Solimini, *Variational Methods in Image Segmentation*, Birkhäuser, 1995.

[2] F. Calderero and F. Marqués, "General region merging approaches based on information theory statistical measures," *ICIP*, 2008.

[3] Nicolas Burrus; Thierry M. Bernard; Jean-Michel Jolion, "Image segmentation by a contrario simulation," *Pattern Recognition journal (to appear)*, 2009.

[4] G. Koepfler, C. Lopez, and J. M. Morel, "A multiscale algorithm for image segmentation by variational method," *SIAM J. Numer. Anal.*, vol. 31, pp. 282–299, 1994.

[5] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and variational methods," *Comm. on Pure and Applied Math.*, vol. XLII, pp. 577–685, August 1988.

[6] A. Desolneux; L. Moisan; J. M. Morel;, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, Springer, 2008.

[7] L. Igual; J. Preciozzi; L. Garrido; A. Almansa; V. Caselles; B. Rouge, "Automatic low baseline stereo in urban areas,"

Inverse Problems and Imaging, vol. 1, no. 2, pp. 319–348, October 2007.

[8] F. Cao; P. Musé; F. Sur, “Extracting meaningful curves from images,” *J. of Math. Imaging And Vision*, Dec. 2005.