Evaluation of machine learning applied to the realignment of hierarchies for image segmentation

Milena Menezes Adão [1]
Pontifical Catholic University
of Minas Gerais and
Computer Science Department
Belo Horizonte, Minas Gerais, Brazil
Email: milena.adao@gmail.com

Silvio Jamil F. Guimarães
Pontifical Catholic University
of Minas Gerais and
Computer Science Department
Belo Horizonte, Minas Gerais, Brazil
Email: sjamil@pucminas.br

Zenilton K. G. do Patrocínio Jr.
Pontifical Catholic University
of Minas Gerais and
Computer Science Department
Belo Horizonte, Minas Gerais, Brazil
Email: zenilton@pucminas.br

***Abstract*—A hierarchical image segmentation is a set of image segmentations at different detail levels. However, objects can be located at different scales due to their size differences or to their distinct distances from the camera. In literature, many works have been developed to improve hierarchical image segmentation results. One possible solution is to realign the hierarchy such that every region containing an object (or its parts) is at the same level. In this work, we have explored the use of random forest and artificial neural network as regressors models to predict score values for regions belonging to a hierarchy of partitions, which are used to realign it. We have also proposed a new score calculation witch considering all user-defined segmentations that exist in the *ground-truth*. Experimental results are presented for two different hierarchical segmentation methods. Moreover, an analysis of the adoption of different combination of mid-level features to describe regions and different architectures from random forest and artificial neural network to train regressors models. Experimental results have point out that the use of new proposed score was able to improve final segmentation results.**

I. Introduction

Recently, a huge amount of image data has been stored and made available through the Internet, witch has attracted more attention from the research community to tasks related to image processing and analysis, more specifically, to computer vision.

According to [1], a hierarchical image segmentation is a set of image segmentations at different detail levels in which the segmentations at coarser detail levels can be produced from simple merges of regions from segmentations at finer detail levels. Hierarchical segmentation methods such as [1], [2] could be used as basic tool in many computer vision tasks [3]–[5].

A simple way to understand the concept of hierarchical image segmentation is shown in Figure 1, which represents the result of a hierarchical segmentation by a tree. The image segmentations are produced by cuts in the performed at each level of the hierarchy. It is observed that the first hierarchical level was composed of only one region. In the second hierarchical level the previous region was presented in more detail, so it was possible five regions. In turn, on the third level it was possible to visualize the details of the previous regions.

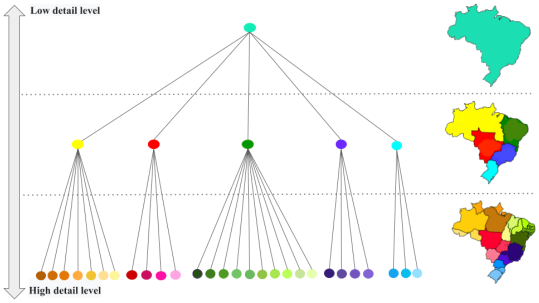
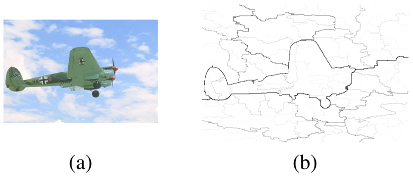


Fig. 1. Example of a result obtained from a hierarchical image segmentation method, represented by a tree, and its segmentations at different levels of detail

The final results of hierarchical image segmentation can be also represented as contour map, e.g., Ultrametric Contour Map – UCM [6], (contour) saliency map [7]). This kind of representation allows a good visibility of results encompassing all hierarchical levels (see Fig. 2), and allows to obtain a particular segmentation (at a certain observation scale) through a simple thresholding.

In literature, many works have been done recently [8]–[11] to demonstrated that scale-awareness seems to be helpful in improving final results in many computer vision tasks. Specifically in hierarchical image segmentation, many challenges still exists, such as, the presence of multiple objects (or part of objects) at different scales in a hierarchy of partitions. In an attempt to solve many challenges, Deep Convolutional Neural



(a) (b)

Fig. 2. Example of result obtained by a hierarchical image segmentation method: (a) original image, (b) representation by UCM.

[1] M.Sc. dissertation

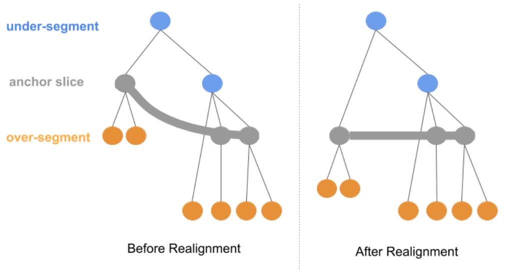


Fig. 3. Example of the realignment of hierarchies process

Networks (DCNNs) have been used to refined computer vision systems [12].

In [13], the authors proposed to improve the final result of a hierarchical algorithm by changing its alignment, in order to put (almost) all objects (and their parts) at the same level (or scale). For that, the authors first trained a random forest with 100 trees as a regressor to predicted the scale of a hierarchy regions using one set of mid-level features. Then, they create a set of regions that better balance between over and under-segmentation, named *anchor slice*. Based on the *anchor slice*, the original hierarchy is realigned adjusting the hierarchy, such that every region in the *anchor slice* is at the same level (or scale), an example is shown in Figure 3.

However, in [13] were used as regressor a random forest with 100 trees and one set of mid-level features, so the following question arose: *"What is the impact on the results of hierarchical segmentation after the process of realigning the hierarchies when training random forests and neural networks with different architectures and with different region descriptors?"*. In order to answer this question in this work, first we have used two different hierarchical image segmentation methods: **gPb-owt-ucm** [2] and **hGB** [1]; and we focus on evaluating: (i) impact analysis of the use of distinct hierarchical methods; and (ii) different combinations of mid-level features to describe regions. After that, we extend that previous study with an *in-depth* analysis of regression model behavior using two distinct approaches: (i) random forest (RF); and (ii) artificial neural network (ANN) varying the size and architecture details for each model, which were trained and evaluated with different combinations of mid-level features (in a total of 660 distinct models). In additional we have also proposed a new score calculation to be used during training step instead of the one proposed by [13]. Then we have compared the resulting hierarchies (after realignment) using both the traditional consensus evaluation strategy [2] and a new proposal considering all user-defined segmentations in the ground-truth.

The main contributions of this work include: (i) analysis of how the process of realigning hierarchies impacts the segmentation of images; (ii) analysis of the quality of the predictions obtained by artificial neural networks and random forests to label if a region is under-, over, or properly-segmented, as well as the analysis of how they act in different architectures and with different combinations of mid-level features; (iii) analysis of how the use of different sets of features in training influences the final segmentations of the process of realigning hierarchies; and (iv) analysis of how the uses of all user-defined segmentations present in the *ground-truth* for the same image in the training process can influence the final results of the process of realigning hierarchies.

II. Theoretical Reference

Interpreting data contained in digital images is considered a complex activity and, therefore, segmentation can be considered as an intermediate process necessary to make possible the application of the interpretation in question [14].

There is a rich literature of hierarchical image segmentation. But in this work, we only describe the hierarchical methods used in: **gPb-owt-ucm** [2] and **hGB** [1], both widely used in the literature.

gPb-owt-ucm [2] method was based on [15] method, witch proposed a function to detect probables boundaries of the image segments, named Pb. In [2], first, a multiscale version of a $Pb(x, y, \theta)$ (mPb) detector is used to predict the probable of a boundary in the θ orientation for a pixel (x, y) by the measure of local brightness, color and texture. The adaptation of the method developed by [2] to the context of multscales, consists in executing the function $Pb(x, y, \theta)$ in different scales for each channel of brightness, color and texture. After that a sequence of Gaussian filters was applied to the resulting mPb image and then a calculation of the eigenvector orientation information of the segments is responsible for normalizing and globalizing (gPb) the initial segmentation of the image. After this point, the algorithm is already able to determine the contours in the image. hen, an UCM is generated from the boundaries of these initial regions.

The **HGB** method [1], [16], starts with simple regions representing single image pixels, than it is able to produce a hierarchy of partitions for the entire image.

For that, an image is transformed into an non-directed graph $G = (V, E)$, where V is a finite set of vertices, which represent the pixels of the image and the edges E represent a measure of dissimilarity between a pair of pixels. A color gradient is calculated by the Euclidean distance in the color space RGB and used as a feature of the pixels. The set E is a subset of $\{x, y\} \subseteq V \mid x \neq y\}$ (which represents the 8-neighbors of each pixel). If w is a function that maps E to R^+, then $w(u)$ is the weight of the edge u measure of dissimilarity between two pixels and (G, w) is a non-directed weighted graph.

Given a finite set V, a partition of V is defined as a set P of non-empty disjoint subsets of V whose union is V. Every element of a P partition is called a P region. Given two partitions P and P' of V, it is said that P' is a refinement of P, denoted by $P' \preceq P$, if any region of P' is included in one of P. A hierarchy of V is a sequence $\mathcal{H} = (P_0, \ldots, P_l)$ of partitions of V, so that $P_{i-1} \preceq P_i$ for any $i \in \ 1, \ldots, l\}$.

Machine learning is a subarea of artificial intelligence witch has arisen to solve difficult problems, since it allows a collection of input and output pairs to learn a function by means of pattern extraction, capable of predicting the output to new input data [17] .

Machine learning methods are divided into two broad categories of learning: (i) unsupervised; and (ii) supervised [18]. [17] defined another category beyond the other two, named semi-supervised (iii).

The inputs to be passed through the learning process are described by a series of features (or attributes) that can be quantitative or qualitative. The branch of machine learning that deals exclusively with qualitative characteristics is called pattern recognition, whereas statistical learning operates exclusively on numerical data [19]. In statistical learning, the characteristics of an input are organized into a vector $[x_1, x_2, \ldots, x_n]^T \in R^n$, constituting the space of characteristics in that each axis corresponds to a specific characteristic. Already qualitative characteristics, must be converted into quantitative.

The set of data used in the learning process is usually in the form of a set of data labeled $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, _N], \mathbf{z}_j \in R^n$. The label indicates the class or value of $\mathbf{z}_j$ that is expected, and is represented by $y_j \in \Omega, j = 1, \ldots,$ N. The data set can be organized as an array with N rows (entries) by n columns (features), with an extra column (or vector) containing the labels.

In this way, a classifier or regressor can be defined as any D function capable of assigning a label to an input data, that is, $D : R^n \mapsto \Omega$. Usually an input $\mathbf{x} \in R^n$ is labeled with the highest scoring class or value.

III. Related Works

Exploiting scaling information has proved useful for many image segmentation tasks, such as segmenting semantic images [12], for contour detection [11] and improving the hierarchies resulting from segmentation methods [13].

In order to achieve the best hierarchical targeting result, [20], proposed flatten the hierarchy in a single segmentation. Commonly, to find the best result of hierarchical segmentation, horizontal cuts are used in hierarchies. In their work, [21], [22] proposed to explore the use of non-horizontal cuts in hierarchies. In this way it is possible to obtain the best partitions regardless of the hierarchical level they are in. Using this same reasoning, finally, [13] proposed to modify the final result of a hierarchical algorithm, in order to modify the scales of observation, that is, to modify the hierarchical levels in which the regions meet, so that (almost) all objects (and their parts) can be at the same level (or scale).

In [2], the authors proposed a benchmark to evaluate segmentations and their boundaries. This benchmark adopts the following evaluation metrics: Segmentation Covering (SC), Probabilistic Rand Index (PRI), Variation of Information (VI) and F-measure. The *ground-truth* used in the calculations are based on a majority voting procedure among the several user-defined segmentations present in the *ground-truth*, creating then, a single segmentation.

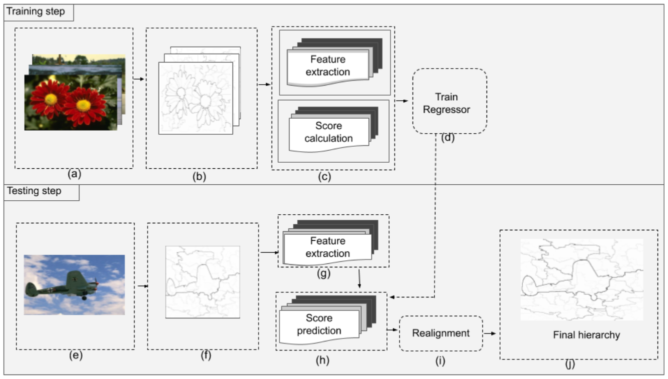


Fig. 4. Realign approach

IV. Methodology

The realignment approach used in this work is illustrated in Fig. 4, which is divided in two steps: training step, in which a regressor is generated; and testing step, when the obtained regressor is used to predict regions scores that are used in realignment process.

During training step, a set of training images is used (4a) to produce the corresponding set of hierarchies (4b). After that, all regions of the hierarchies are described with a set of mid-level features and their scores values are also calculated (4c). We calculated the scores similar to [13], [23] and used a new score calculation proposed in this work in order to compare both. Then, these scores and features are used to train a regression model (4d).

During testing step, for each test image (4e) the corresponding hierarchy is produced (4f). Then, the same set of mid-level features used in training are described for every region of the hierarchy (4g). After that, these features are used with the trained regression model in order to predict the score value for each region (4h). Finally, based on the scores predicated, a *anchor slice* was identified and used in the realignment process (4i) to produces a final hierarchy (4j) (analogously to [13], [23]).

A. Features Extraction

The mid-level features were extracted from all regions of each hierarchy is the same used by [13]: shape features (such as area, perimeters, etc.), graph features (such as cut, ratio cut, etc.) and gestalt features (such as texton similarities, brightness similarities, etc). Moreover, we have also explored features to encode color properties, such as color mean, and color histogram [24].

Here ‘c’ stands form color based features, ‘s’ is used to represent region shape features, ‘g’ stands for graph features, and ‘t’ is used to represent gestalt features.

B. Score Calculations

The calculation of score value S_i^k used in [13], [23] consisting of, for each region R_i^k of hierarchy, its corresponding *ground-truth* region G_i^k is identified and used to compute the overlap with the Intersection over Union (IoU) S_i^k by

Eq. 1. As suggested by [13], [23], only the first user-defined segmentation found for image k in the *ground-truth* was used.

$$S_i^k = \frac{\mid G_i^k \mid - \mid R_i^k \mid}{\max\left(\mid G_i^k \mid, \mid R_i^k \mid\right)} \tag{1}$$

in which $\mid R_i^k \mid$ and $\mid G_i^k \mid$ represent the size of region R_i^k and the size of its corresponding *ground-truth* region G_i^k, respectively for image k. The most-overlapping human-annotated segment is taken as the corresponding *ground-truth* region.

In this work we proposed a new score calculation Υ_i^k by Eq. 2, which considers all user-defined segmentation found in the *ground-truth*.

$$\Upsilon_i^k = \frac{1}{N^k}\left(\sum_{j=1}^{N^k} \frac{\mid G_{ij}^k \mid - \mid R_i^k \mid}{\max\left(\mid G_{ij}^k \mid, \mid R_i^k \mid\right)}\right) \tag{2}$$

in which $\mid G_{ij}^k \mid$ represents the size of j-th user-defined segmentation found in the *ground-truth* region (G_{ij}^k) for image k, N^k stands for the total number of user-defined segmentations for image k; and $\mid R_i^k \mid$ has the same definition of Eq. 1.

Both S_i^k and Υ_i^k return a score value in a interval [-1, 1]; so, when the score is a negative value, it indicates that the region R_i^k is under-segmented, while a positive value stands for over-segmented, and when the score is close to 0 indicates a properly segmented.

C. Regression model generation

Firstly, a random forest (RF) with 100 trees was trained similar to [13], to provide comparability with previous works; after that, aiming to analyzing the impact of the features on the hierarchies produced after the realignment, we trained other RF with 100 trees varying the set of features: 'c+g+s', 'c+s', 'c+t+g+s', and 't+g+s'. It is worth mentioning, that for these firsts experiments we produce the hierarchies using **gPb-owt-ucm** [2] and **hGB** [1].

In this work we have also evaluated the impact of the adoption of different types of regression models to predict regions scores. For that, we have explored random forest (RF) and artificial neural network (ANN) as approaches for generating the regression models. A total of 660 regression models were trained and evaluated. For these experiments we produce the hierarchies using **hGB** [1].

With the purpose of exploring the use of RF to predict regions scores we trained models with the following sizes (in number of trees): 25, 50, 100, 200, and 400, for each model we varied the set of features: 'c', 's', 't', 'g', 'g+t+s', 'g+t+s+c'. Each regression model was trained and tested 10 times, to obtain the average results.

The ANN is a very flexible model that adapts itself to a given data distribution, so it can dynamically pick the best type of regression (*e.g.*, linear, logistic, polynomial) to use. Actually, neural networks are reducible to regression models – a neural network can "pretend" to be any type of regression model. For instance, a very simple neural network with only one input neuron, one hidden neuron, and one output neuron can be trained to be equivalent to a logistic regression. Thus, in order to explore the uses of ANN to train the regions scores, we trained models with two and three layers. For each amount of layers we explore the following number of neurons: 02, 05, and 10. For each model, we varied the set of features: 'c', 's', 't', 'g', 'g+t+s', 'g+t+s+c'. Each regressor model was trained and tested 10 times, to obtain the average results.

After we trained the regressors, we evaluated the quality of each of them.

In this work we have also proposed a new score calculation defined by Eq. 2. Then, we used the same set of features and architecture that have produced the best regressor (of the 660 regressors trained so far) to train and test a new regression model to predict regions scores based in Eq. 2. After, we evaluated the quality of this regressor and compare it with the best regressor obtained before.

D. Hierarchy realignment

After training the regression approach, they are used to make predictions. During testing step, each region of the hierarchy receive a predicted score in $[-1, +1]$ indicating if that region is under-, properly- and over-segmented. The *anchor slice* corresponds to regions that received a score indicating *nearly* properly-segmented. After that, a local linear transform (the same used in [13], [23]) is performed on contour map representing the hierarchy, and the *anchor slice* is aligned to scale value of 0.5 (for the convenience and later use).

V. Experimental Results

BSDS500 dataset [2] was used, which includes 500 images (200 for training, 100 for validation, and 200 for testing).

As segmentation evaluation measures, we adopted: (i) Segmentation Covering (SC); (ii) Probabilistic Rand Index (PRI); (iii) Variation of Information (VI); and (iv) F-measure for boundary (F_b); all four computed at Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS) see [25] for a review of these measures and scales. Note that for all measures a large value is better, except for VI.

In the fist experiment, we evaluated the behavior of the realignment of hierarchy method proposed by [13] using the result of different hierarchical image segmentation methods and different set of features. Average results obtained for regression made with random forest are shown in Table I.

In case of gPb-owt-ucm method, it was possible to observe that the realignment of hierarchies was performed using the color, graph and shape characteristics, that the average of

TABLE I
AVERAGE RESULTS FOR REGRESSION WITH RANDOM FOREST.

	SC ↑		PRI ↑		VI ↓		F_b ↑	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
gPb-owt-ucm no realignment	0.59	0.65	0.83	0.86	1.69	1.48	0.73	0.76
gPb-owt-ucmalignment(c+gr+s)	0.58	0.64	0.82	0.85	1.75	1.49	0.69	0.76
gPb-owt-ucm alignment(c+s)	0.55	0.61	0.79	0.83	1.81	1.58	0.69	0.74
hGB no realignment	0.43	0.62	0.74	0.80	2.34	1.88	0.49	0.50
hGB alignment(c+ge+gr+s)	0.38	0.53	0.75	0.81	2.45	1.89	0.50	0.50
hGB alignment(ge+gr+s)	0.39	0.53	0.74	0.81	2.45	1.89	0.50	0.50
hGB alignment(c+s)	0.43	0.62	0.74	0.80	2.34	1.88	0.49	0.50

F_b comparing with the segmentations that did not undergo the realignment process. In case of HGB method, it was possible to observe that the average values of F_b OIS were maintained in all cases and in the tests that the color, texture, graph, shape and color characteristics were used, and the characteristics graph, and shape, there was an improvement of the average value of F_b ODS and the average value of PRI. The average value of VI remained the same in the case with the characteristics of color and shape.

In order to better understand the impact of the use of different machine learning methods for training and prediction the scores, in the second experiment RF and ANN were trained, totaling 660 regressors with different sets of features and different architecture (quantity of trees in the random forests, and amount of neurons and layers in the neural networks). For this experiment, the training database was divided as follows: 150 images for training and 50 images for the predictions (tests). The hierarchical segmentations used were obtained by the HGB [1] method. In total, in each execution, 361,218 regions were used during training, while 121,271 regions were used for predictions. Then, the quality of each regressor was evaluated by mean squared error (MSE). Average results obtained are shown in Table II.

For RF, the best results are related to models with a large number of trees (400); while, for ANN, the best results are related to models with two and three layers containing only 02 or 05 neurons per layer. Top results for both regressors (RF and ANN) were obtained with mid-level feature combination ‘g+t+s’; and best overall MSE was equal to 17.4% – achieved by one of the ANNs with three layers containing 02 neurons per layer (ANN/3L/2N using ‘g+t+s’).

After, another experiment was carried out in order to evaluate the realignment results using the regressor that obtained the best result on the second experiment (*i.e.*, ANN/3L/2N). Therefore, the best regressor obtained was used to predict the scores of regions beloging to hierarchical segmentations produced for the BSDS500 test set and then these hierarchies were realigned. Finally, to better understand the results obtained from the realignment of the hierarchies, some individual comparisons were made between the segmentations before and after the realignment. Average results obtained are shown in Table III.

The F_b measure presented an improvement of 8∼10% for ODS value and an improvement of 16% for OIS, while PRI measure presented an improvement of only 1.25% for OIS value. The other measures did not present any improvement (and some even got worst, such as ODS values for SC and for VI). This could be explained by the fact that, although some individual results have improved a lot after realignment process, others got worsened (or did not change at all). Fig. 5 exhibits an example that has improved after realignment process, along with one that has remained unmodified and another which has worsened.

TABLE II
AVERAGE MSE FOR PREDICTIONS WITH RF AND ANN MODELS.

	Feature combination					
	g+t+s	g+t+s+c	g	t	s	c
RF/25T	0.185	0.185	0.218	0.187	0.199	0.203
RF/50T	0.181	0.182	0.217	0.184	0.196	0.199
RF/100T	0.179	0.180	0.215	0.182	0.194	0.198
RF/200T	0.179	0.179	0.214	0.181	0.193	0.197
RF/400T	**0.178**	**0.179**	**0.214**	**0.181**	**0.193**	**0.197**
ANN/2L/2N	0.197	**0.179**	0.248	0.180	0.220	**0.193**
ANN/2L/5N	0.186	0.198	**0.195**	**0.178**	0.262	0.196
ANN/2L/10N	0.260	0.247	0.355	0.189	0.248	0.241
ANN/3L/2N	**0.176**	0.182	0.195	0.189	**0.205**	0.193
ANN/3L/5N	0.200	0.209	0.195	0.179	0.563	0.196
ANN/3L/10N	0.236	0.279	0.236	0.212	0.289	0.218

TABLE III
SEGMENTATION EVALUATION MEASURES BEFORE AND AFTER REALIGNMENT WITH DIFFERENT FEATURE COMBINATIONS.

	SC ↑		PRI ↑		VI ↓		F_b ↑	
	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS
No realignment	**0.43**	**0.62**	**0.74**	0.80	**2.34**	**1.88**	0.49	0.50
Realigned (g+t+f)	0.37	0.53	**0.74**	**0.81**	2.45	1.89	0.53	**0.58**
Realigned (g+t+f+c)	0.37	0.53	**0.74**	**0.81**	2.45	1.89	0.53	**0.58**
Realigned (g)	0.42	0.53	**0.74**	**0.81**	2.45	1.89	0.54	**0.58**
Realigned (t)	0.36	0.53	**0.74**	**0.81**	2.45	1.89	0.53	**0.58**
Realigned (f)	0.37	0.53	**0.74**	**0.81**	2.45	1.89	0.53	**0.58**
Realigned (c)	0.39	0.53	**0.74**	**0.81**	2.44	1.89	0.53	**0.58**

Using the score calculation proposed in this work by Eq. 2, in order to assess the impact of using Υ_i^k (instead of S_i^k), another regressor with the same characteristics and architecture of the best regressor obtained on the second experiment (ANN/3L/2N with ‘g+t+s’) was trained. For that, the scores of the test dataset was calculated. Then, the quality of this new regressor was evaluated, and the test hierarchies were later realigned. Finally, an evaluation of the realigned segmentations that were generated was done. In Fig. 6 and Fig. 7, one can see that images, with unmodified and worsened results presented in Fig. 5, have now been improved. The new regression model achieved a MSE of 10.5% (which represents a 40% reduction

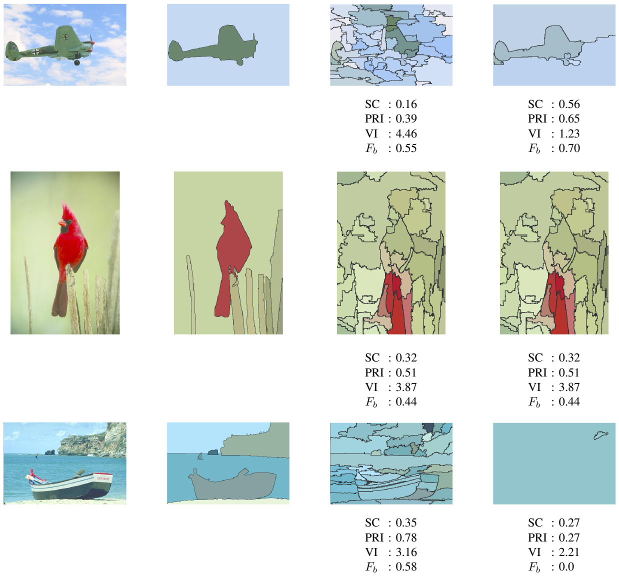
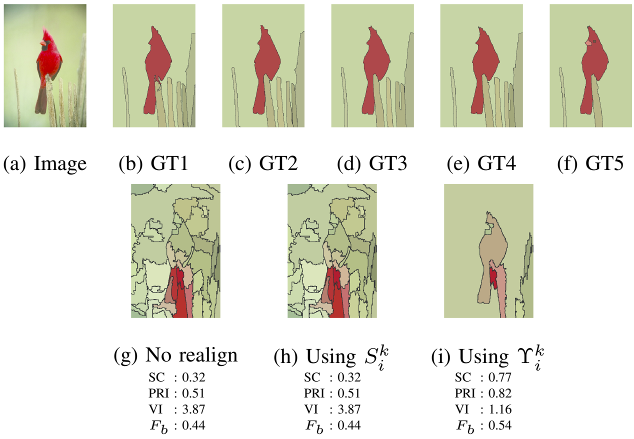


Fig. 5. Examples of results: an improved result (first line); an unmodified result (second line); and a worsened result (third line). From left to right: (i) original image; (ii) first segmentation in the *ground-truth*; (iii) segmentation before realignment; and (iv) segmentation after realignment. Both segmentations were obtained through a horizontal cut a scale value 0.5.


(a) Image (b) GT1 (c) GT2 (d) GT3 (e) GT4 (f) GT5

(g) No realign
SC : 0.32
PRI : 0.51
VI : 3.87
F_b : 0.44

(h) Using S_i^k
SC : 0.32
PRI : 0.51
VI : 3.87
F_b : 0.44

(i) Using Υ_i^k
SC : 0.77
PRI : 0.82
VI : 1.16
F_b : 0.54

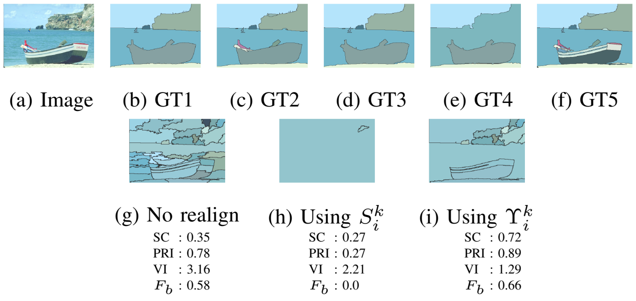
Fig. 6. Example of segmentation results before and after the realignments using both S_i^k and Υ_i^k.

in relation to the previous best MSE value of 17.4% obtained using S_i^k).

Finally, we also analyzed the resulting hierarchies after realignment with the newly trained regressor, using both the traditional consensus evaluation strategy [2] and an approach in which all user-defined segmentations in the *ground-truth* are used.

The evaluation strategy proposed by [2] obtain a single segmentation based on a majority voting procedure among the several user-defined segmentations present in the *ground-truth*. Again, this could favor over- or under-segmented results depending on the number and quality of those segmentations.

In this work, we have also proposed to evaluate the final segmentation results which is the one obtained through a horizontal cut a scale value 0.5 (which corresponds to the *anchor slice*) using each one of the segmentations present in the *ground-truth* separately. At the end, we have taken an average of each segmentation measure for each image. The results are shown at Table IV which also presents the results when the maximum value of each segmentation measure for each image is considered (instead of the average). As one can see, except for F_b, all other measures have shown improvements.


(a) Image (b) GT1 (c) GT2 (d) GT3 (e) GT4 (f) GT5

(g) No realign
SC : 0.35
PRI : 0.78
VI : 3.16
F_b : 0.58

(h) Using S_i^k
SC : 0.27
PRI : 0.27
VI : 2.21
F_b : 0.0

(i) Using Υ_i^k
SC : 0.72
PRI : 0.89
VI : 1.29
F_b : 0.66

Fig. 7. Another example of segmentation results before and after the realignments using both S_i^k and Υ_i^k.

TABLE IV
SEGMENTATION MEASURES FOR DIFFERENT EVALUATION STRATEGIES AFTER REALIGNMENT PROCESS USING Υ_i^k.

Evaluation strategy	SC ↑	PRI ↑	VI ↓	F_b ↑
Consensus evaluation [2]	0.53	0.81	1.89	**0.58**
Evaluation using all GT segmentations (avg)	0.56	0.83	1.81	0.46
Evaluation using all GT segmentations (max)	**0.63**	**0.88**	**1.45**	0.53

VI. Conclusion

In this work, we explored the use of random forests to predict the best scale value for given region, which is then used to realign the entire hierarchy.

Experimental results are presented for two different segmentation methods (gPb-owt-ucm and hGB) and with an analysis of the impact when different combination of mid-level features to describe regions are used.

In order to improve and better understand our results, we have explored different regression approaches to predict score values from a hierarchy of partitions generated from an image by a hierarchical image segmentation method. Those score values should be near 0 when the regions is properly segmented, while score values should be near to -1 or +1 for regions that are under- or over-segmented, respectively. Thus, predicted scores could be used to created a partition containing regions properly segmented that can be used to realign the hierarchy, , to make adjustments to the hierarchy representation such that every properly segmented region ends up at the same final predefined level (or scale).

In this work, we have presented an analysis of regression model behavior using random forest and artificial neural network, using distinct architecture and 06 different combinations of mid-level features.

We have also proposed and evaluated a new score calculation, witch considering all user-defined segmentations that exist in the *ground-truth*. Experimental results have demonstrated that its adoption was able to improve final segmentation results.

Finally, we have evaluated the resulting hierarchies (after realignment) using a new proposal which considers all user-defined segmentations that exist in the *ground-truth*. This new assessment strategy could be seen as an alternative to the traditional one which is based on a consensus generated by a majority voting procedure among the several user-defined segmentations present in the *ground-truth*.

As future works, we intend to study a technique that allows to represent all user-defined segmentations that exist in the *ground-truth* in a single annotation. In addition, we intend to apply the technique of realigning hierarchies in video segmentation.

Acknowledgment

The authors are grateful to PUC Minas, CAPES (PVE 125000/2014-00), FAPEMIG (PPM 00006-16), and CNPq (Universal 421521/2016-3 and PQ 307062/2016-3) for the financial support to this work. As result of this M.Sc. dissertation, a paper was published [23] and a periodical article was submitted, by which we were waiting for the acceptance.

References

[1] S. J. F. Guimarães, Y. Kenmochi, J. Cousty, Z. K. G. P. Jr, and L. Najman, "Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the Felzenszwalb-Huttenlocher method," *Mathematical Morphology - Theory and Applications*, vol. 2, pp. 55–75, 2017.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE TPAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[3] L. S. Belo, C. A. Caetano, Z. K. G. Patrocnio Jr, and S. J. F. Guimares, "Summarizing video sequence using a graph-based hierarchical approach," *Neurocomputing*, vol. 173, pp. 1001–1016, 2016.

[4] F. Rodrigues, P. Leal, Y. Kenmochi, J. Cousty, L. Najman, S. Guimarães, and Z. Patrocnio Jr, "Graph-based hierarchical video cosegmentation," in *Image Analysis and Processing - ICIAP 2017*, S. Battiato, G. Gallo, R. Schettini, and F. Stanco, Eds., 2017, pp. 15–26.

[5] K. J. F. Souza, A. A. Arajo, Z. K. G. Patrocnio Jr, and S. J. F. Guimares, "Graph-based hierarchical video segmentation based on a simple dissimilarity measure," *PRL*, vol. 47, pp. 85–92, 2014.

[6] P. Arbelaez, "Boundary extraction in natural images using ultrametric contour maps," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE, 2006, pp. 182–182.

[7] J. Cousty, L. Najman, Y. Kenmochi, and S. J. F. Guimarães, "Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 4, pp. 479–502, May 2018.

[8] L. C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *IEEE CVPR 2016*, 2016, pp. 3640–3649.

[9] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, and X. Hu, "Scale-aware face detection," in *IEEE CVPR 2017*, 2017, pp. 1913–1922.

[10] Z. Jie, X. Liang, J. Feng, W. F. Lu, E. H. F. Tay, and S. Yan, "Scale-aware pixelwise object proposal networks," *IEEE TIP*, vol. 25, no. 10, pp. 4525–4539, 2016.

[11] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2018.

[12] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE TPAMI*, vol. 40, no. 4, pp. 834–848, 2018.

[13] Y. Chen, D. Dai, J. Pont-Tuset, and L. Van Gool, "Scale-aware alignment of hierarchical image segmentation," in *IEEE CVPR 2016*, 2016, pp. 364–372.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice-Hall, Incg, 2001.

[15] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, May 2004.

[16] S. J. F. Guimarães, J. Cousty, Y. Kenmochi, and L. Najman, "A hierarchical image segmentation algorithm based on an observation scale," in *SSPR/SPR*. Kluwer Academic Publishers, 2012, pp. 116–125.

[17] S. Russell, S. Russell, P. Norvig, and E. Davis, *Artificial Intelligence*, ser. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[18] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, 2nd ed. Wiley Publishing, 2014.

[19] K. Fu, *Syntactic pattern recognition and applications*, ser. Prentice-Hall advanced reference series: Computer science. Prentice-Hall, 1982.

[20] C. Xu, S. Whitt, and J. J. Corso, "Flattening supervoxel hierarchies by the uniform entropy slice," in *IEEE ICCV 2013*, 2013, pp. 2240–2247.

[21] J. Cousty and L. Najman, "Morphological floodings and optimal cuts in hierarchies," in *IEEE ICIP 2014*, 2014, pp. 4462–4466.

[22] L. Guiges, J. Cocquerez, and H. L. Men, "Scale-sets image analysis." *IJCV*, vol. 68, no. 3, pp. 289–317, 2006.

[23] M. M. Adão, S. J. F. Guimarães, and Z. K. G. Patrocínio, "Evaluation of scale-aware realignments of hierarchical image segmentation," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications – CIARP 2018*, R. Vera-Rodriguez, J. Fierrez, and A. Morales, Eds. Springer International Publishing, 2019, pp. 141–149.

[24] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *IEEE CVPR 2010*. IEEE, 2010, pp. 3241–3248.

[25] J. Pont-Tuset and F. Marques, "Supervised evaluation of image segmentation and object proposal techniques," *IEEE TPAMI*, vol. 38, no. 7, pp. 1465–1478, 2016.