UPPER-BOUND ASSESSMENT OF THE SPATIAL ACCURACY OF HIERARCHICAL REGION-BASED IMAGE REPRESENTATIONS

Jordi Pont-Tuset and Ferran Marques

Department of Signal Theory and Communications
Universitat Politècnica de Catalunya (UPC)
{jordi.pont,ferran.marques}@upc.edu

ABSTRACT

Hierarchical region-based image representations are versatile tools for segmentation, filtering, object detection, etc. The evaluation of their spatial accuracy has been usually performed assessing the final result of an algorithm based on this representation. Given its wide applicability, however, a direct supervised assessment, independent of any application, would be desirable and fair.

A brute-force assessment of all the partitions represented in the hierarchical structure would be a correct approach, but as we prove formally, it is computationally unfeasible. This paper presents an efficient algorithm to find the upper-bound performance of the representation and we show that the previous approximations in the literature can fail at finding this bound.

***Index Terms*—** Image segmentation, region-based hierarchy, binary partition tree, supervised assessment

1. INTRODUCTION

Region-based hierarchical image representations have proven its applicability in many fields such as segmentation, filtering, information retrieval [1]; object detection [2, 3, 4], contour detection [5, 6], etc.

The validity of these approaches is usually proven via a supervised task-based or *system-level* assessment [7], that is, comparing the final result against a manually-annotated database known as ground truth, as done in [2]. This way, as contour detectors, the image representations can be assessed by analyzing the quality of the boundary maps obtained in the precision-recall environment presented in [8] on the BSDS300 database [9]. In an object detection environment, the quality of the representation can be analyzed by comparing the object masks obtained against an object ground-truth database as the DCU dataset [4].

As a generic image representation, however, it would be desirable to assess the representations *intrinsically*, independently of the final application the user will give to the representation. A hierarchical region-based image representation is a structured set of image partitions from the most detailed ones (more regions) to the coarsest ones (less regions). This way, an intuitive approach to assess the representation directly could be to compare *all* the partitions represented in the hierarchy and to assess their quality with respect to their number of regions.

Any hierarchy of nested regions based on a set of non-overlapping regions can be represented by a binary tree of regions, which in [1] is referred to as Binary Partition Tree (BPT), so although this work is focused on this type of trees, the results are generalizable to any hierarchy of regions.

Starting from a partition, BPT is constructed by iteratively merging the pairs of regions that are more similar according to a given measure. Each merging produces a new partition with exactly one region less than the previous one until only one region (the single image) remains. Figure 1 illustrates this process on a simple image partition.

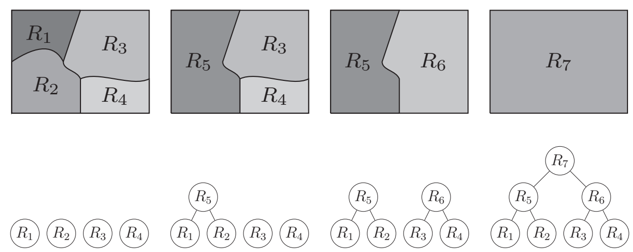

Fig. 1. BPT creation process: From left to right, the two most-similar neighboring regions are merged at each step. Below, the BPT representation depicted by a tree, where the region formed from the merging of two segments is represented as the parent of the two respective nodes

The set of mergings that a BPT produces is known as *merging sequence* and the set of partitions that are iteratively formed in the process is known as *merging-sequence partition set*. In the example of Figure 1, the merging sequence is $\{R1 \leftrightarrow R2, R3 \leftrightarrow R4, R5 \leftrightarrow R6\}$, and the merging-sequence partition set is formed by the four represented partitions.

We first prove (Section 2) that the number of image partitions represented in a BPT can grow exponentially with the number of leaves of the tree (initial number of regions, N), thus it is unfeasible to try to assess the hierarchy by evaluating all the partitions exhaustively by brute force.

An accepted approach to overcome this type of limitation is to analyze the *upper-bound performance* [10] of the representation, that is, to assess the partition that *best* matches the ground truth scanning all possible number of regions. The approach followed in [11, 12] in this direction consists in assessing the N partitions in the *merging sequence*, expecting them to be the *best* representations. There is no guarantee, however, that this set of partitions is indeed the best representation in the hierarchy for each number of regions. For instance, following this strategy, in the example of Figure 1, the partition formed by R_1, R_2, and R_6 would never be analyzed. Examples of looking for partitions outside the merging sequence can be found in [1] and [2].

This work has been partially supported by the Spanish *Ministerio de Ciencia e Innovación*, under project TEC2010-18094.

The main contribution of this paper is to present a technique to efficiently find the partition coded in the hierarchy that best represents the ground truth for any number of regions, as presented in Section 3. The rationale behind this technique is to take advantage of the tree structure to decompose the assessment recursively, provided that the assessment measure fulfills a locality constraint.

Section 4 presents the experiments to prove that the previous techniques are not capable of always finding the upper-bound performance and that the differences in the resulting assessment are noticeable, while the proposed technique finds the upper-bound performance efficiently.

Finally, we draw the conclusions and sketch the future lines of work in Section 5.

2. UNFEASIBILITY OF A BRUTE-FORCE ASSESSMENT

Let N be the number of leaves of a tree or, in image segmentation terms, the number of regions of the original partition on which the tree is built.

Following the definitions in [13], the *depth* of a node is the number of nodes from it to the root, both inclusive. The *height* of the tree is the maximum depth of its leaves, and it is said to be *height-balanced* (in short, balanced) when the depth of the leaves differ at most by 1. This can be seen as the worst-case scenario in the study of the quality of a hierarchy.

Lemma 1. *The number of nodes of a binary tree with N leaves is $M = 2N - 1$.*

Proof. Let us imagine we mark the nodes of the tree by the following procedure: starting from the N leaves marked, we mark any node with two marked sons and un-mark its sons. We repeat the process until only the root is marked. Each step of this process marks a new region and reduces the number of nodes still to be marked of the tree by one. Since originally there are N marked nodes, and at the end of the process just one, the number of steps from the leaves to the whole image is exactly $N-1$. Consequently, $M = N + (N - 1) = 2N - 1$. □

Lemma 2. *The maximum number of nodes at depth exactly d at the tree is 2^{d-1}.*

Proof. By induction, for $d = 1$, there is only $1 = 2^0$ root. If at level d there are at most 2^{d-1} nodes, at level $j = d + 1$, since the tree is binary, there will be at most $2^{d-1} \cdot 2 = 2^d = 2^{j-1}$. □

Lemma 3. *All the leaves of a binary tree have exactly the same depth d if, and only if, $d = \log_2(N)+1$, or, equivalently $N = 2^{d-1}$.*

Proof. Given that all the leaves have the same depth, each level will be complete, so Lemma 2 applies. Being M the number of nodes in the tree, counting them from the root:

$$M = 2^0 + 2^1 + 2^2 + \cdots + 2^{d-1} = 2^d - 1$$

Applying Lemma 1:

$$2N - 1 = 2^d - 1 \quad \Rightarrow \quad d = \log_2(2N) = \log_2(N) + 1 \quad □$$

Theorem 1. *Let $\mathcal{P}$ be the set of all possible partitions that can be extracted from a height-balanced BPT, and $|\mathcal{P}|$ its cardinality. Being N the number of regions in the original partition (leaves), then: $|\mathcal{P}| \geq 2^{N/2}$.*

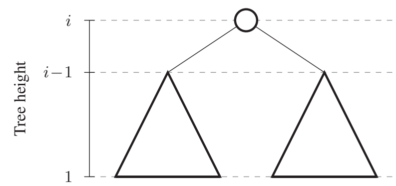

Fig. 2. Recursive decomposition of a BPT. The vertical axis refers to the height of the corresponding tree. When two height-balanced subtrees of height $i - 1$ are merged, the resulting tree has height i

Proof. We will proof the result by induction on the height of the tree h. For $h = 2$, $N = 2$, $|\mathcal{P}| = 2 \geq 2^{2/2} = 2$.

Let us assume that the result is true for $h = i - 1$, i.e., for $N = 2^{i-2}$ (recall Lemma 3). A tree of height $h = i$ can be seen as the merging of the roots of two trees of height $h = i-1$, as depicted in Figure 2.

Then we will have that each partition in the left tree can be combined with any partition of the right tree forming a new partition. Adding the new root, it follows that:

$$|\mathcal{P}_i| = |\mathcal{P}_{i-1}|^2 + 1 \geq |\mathcal{P}_{i-1}|^2$$

Applying the induction hypothesis:

$$|\mathcal{P}_i| \geq |P_{i-1}|^2 \geq \left(2^{\frac{2^{i-2}}{2}}\right)^2 = 2^{\left(2^{i-2}\right)} = 2^{\frac{2^{i-1}}{2}}$$

By induction it follows that:$|\mathcal{P}| \geq 2^{N/2}$. □

In other words, the number of partitions that can be extracted from a balanced BPT **grows exponentially** with the number of regions of the original partition. For a generic BPT the brute-force assessment would therefore be unfeasible in the worst-case scenario. To get a flavor of the dimensionality, for a balanced tree constructed on only 64 regions, the number of different partitions that can be extracted is higher than four thousand millions.

3. UPPER-BOUND PERFORMANCE ASSESSMENT

Intuitively, recalling Figure 2, the main idea behind our approach is to compute the best representation of k regions at height i as combinations of j and $k-j$ regions on each subtree of height $i-1$. Starting at height 1, only a representation with $k = 1$ regions is possible, and then at each increase in height, only the best representations of the subtrees have to be explored, avoiding all the suboptimal representations. Following, we describe the algorithm formally.

Definition 1. *Given a partition $P = \{R_j\}$, and a ground truth partition GT, an assessment measure $m(P, GT)$ is said to be* ***local*** *when there exists another measure m^l such that:*

$$m(P, GT) = \sum_j m^l(R_j, GT)$$

Let $t(R_j)$ be the height of the subtree rooted at node R_j. Let us denote the regions of a tree as R_1,..., R_N, ..., R_{2N-1}, numbered in increasing order of t, i.e.:

$$1 = t(R_1) = t(R_2) = \cdots = t(R_N) \leq \cdots \leq t(R_{2N-1})$$

Note that, under this condition, R_1, ..., R_N are the leaves and R_{2N-1} is the root. Note also that, in the case of the BPT, the merging order fulfills this requirement.

Definition 2. *We define s_k^i as the value of the addition of m^l on the best representation by means of exactly k regions from the subtree below R_i.*

Following this definition, our objective is to compute s_k^{2N-1}, that is, the upper-bound performance of the whole tree for $k = 1 \ldots N$ regions. Formally, Algorithm 1 describes the procedure to find the upper-bound performance of a tree with respect to a local measure m, where il and ir are the indices of the *left* and *right* sons of the node R_i, respectively; and $t'(R_i)$ is the number of leaves in the subtree below R_i, so we have that $t'(R_j) = t'(R_{il}) + t'(R_{ir})$.

Algorithm 1: Upper-bound tree assessment

```
for i = 1, ..., 2N−1 do
    s_1^i ⟵ m^l(R_i, GT)
    if i > N then
        for k = 2, ..., t'(R_il) + t'(R_ir) do
            s_k^i ⟵ +∞
        end
        for p = 1, ..., t'(R_il)−1 do
            for q = 1, ..., t'(R_ir)−1 do
                s_{p+q}^i ⟵ min{ s_{p+q}^i, s_p^il + s_q^ir }
            end
        end
    end
end
```

Note that we are assuming that the assessment measure is an *error* measure or a *distance*, in the sense that the lower the better, since we compute the minimum of the measure in the leaves of each node. We should change to the maximum if the measure is a *similarity*, i.e., the higher the measure, the better.

Lemma 4. *The number of regions at height h in a balanced tree is exactly $N2^{1-h}$.*

Proof. At height 1 there are N leaves and, at each height increase, the number of regions is divided by 2. □

Theorem 2. *Algorithm 1 time complexity is $O\big(N(\log_2 N)^2\big)$.*

Proof. The number of relevant s_k^i updates in Algorithm 1 are:

$$T(N) = 2N + 1 + \sum_{i=N+1}^{2N-1} t'(R_{il}) \cdot t'(R_{ir})$$

In the worst-case scenario, a height-balanced tree, the number of leaves below each region at height h is exactly h, that is, $t'(R_{il})$ is equal to the height of R_{il}. Let us rewrite the summation by levels of height, that is, from $h = 2$ to $\log_2 N + 1$. The number of regions exactly at height h is $N2^{1-h}$ (Lemma 4), so the total number of operations is:

$$\begin{aligned} T(N) &= 2N + 1 + \sum_{h=2}^{\log_2 N+1} h \cdot h \cdot N2^{1-h} \leq \\ &\leq 2N + 1 + N \sum_{h=2}^{\log_2 N+1} h \\ &= 3N + \frac{1}{2}(\log_2 N + 1)(\log_2 N + 2) = O\big(N(\log_2 N)^2\big) \end{aligned}$$ □

That is, Algorithm 1 takes advantage of the hierarchical structure to find the upper-bound performance efficiently, which was not feasible by a brute-force analysis.

4. EXPERIMENTS

As an assessment measure, we will use the **asymmetric partition distance** d_{asym} [14] (as done in [11, 12]), defined as:

$$d_{asym}(P, GT) = \sum_{R \in P} \max_{R' \in GT} |R \cap R'|$$

This definition directly proves that the measure is local, since letting:

$$d^l_{asym}(R_j, GT) = \max_{R' \in GT} |R_j \cap R'|$$

we have that $d_{asym}(P, GT) = \sum_j d^l_{asym}(R_j, GT)$, thus fulfilling Definition 1.

As baseline, we will compare the values obtained by the upper-bound assessment against only assessing the partitions of the merging sequence of a BPT, as done in [11, 12] and presented in Section 1.

The BPTs are built as in [11], starting at 300 regions (leaves). The 500 images from the segmentation ground-truth database BSDS500 [5] are segmented and evaluated using the merging sequence technique and the upper-bound performance assessment.

First, to illustrate how a typical result looks like, Figure 3 plots the evolution of the error on the trees built on three example images. As expected by definition, the quality obtained via the upper-bound performance technique is always better than the merging sequence analysis. Qualitatively, the evolution of the merging-sequence curve is stepped, which shows that, at some points, although increasing the number of regions, the technique is not capable of finding a better representation.

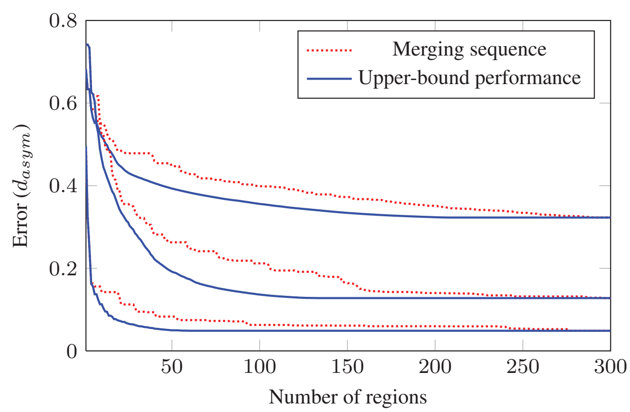

Fig. 3. Merging sequence versus upper-bound-performance error for three example trees

To make the results statistically significant, Figure 4 shows the mean of the error for the whole set of 500 images of BSDS500, compared against the 2696 ground-truth partitions defined on these images. The differences between the strategies are indeed kept in mean, thus proving that measuring the quality on the merging sequence does not provide the upper-bound performance as expected.

To get a better quantitative idea of the relevance of the differences, Figure 5 plots the percentage increase of the error of the merging sequence analysis with respect to the upper-bound performance.

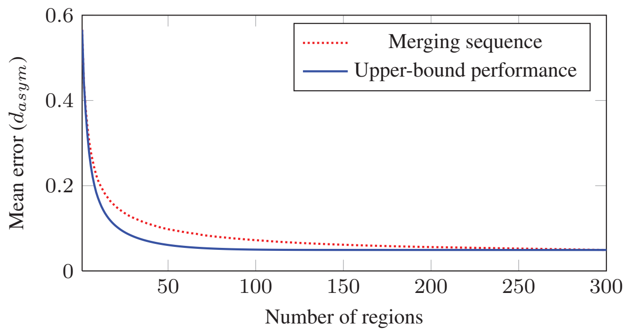

Fig. 4. Merging sequence versus upper-bound-performance error mean in BSDS500

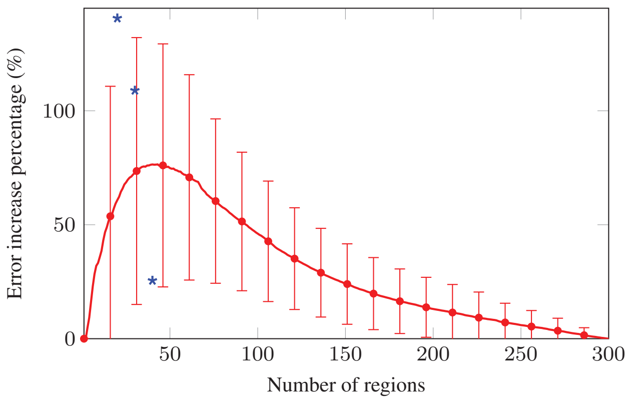

Fig. 5. Error increase percentage of merging sequence with respect to upper-bound-performance error. Blue stars show the values for the three example images of Figure 6

The error percentage increase is maximum at around 40 regions, where it reaches 76%. This value is clearly significant and could mislead the assessment performed when comparing different techniques or tuning a segmentation algorithm.

The high variance observed in the error percentage increase makes it even more crucial to use the upper-bound performance, since it means that the merging sequence result is not consistent between images, and therefore it cannot be considered as a bias that does not affect the comparison of the results.

Finally, Figure 6 shows three examples of partitions from the ground truth, the merging sequence, and the upper-bound performance analysis. In it, we graphically corroborate that the differences can indeed be significant. The error increase percentage for these three examples is plotted in blue stars in Figure 5.

5. CONCLUSIONS

This paper presents a novel and efficient algorithm to find the upper-bound performance of a hierarchical region-based image representation in a supervised environment. It takes advantage of the structure of the hierarchies in form of a tree to avoid a brute-force evaluation of all the partitions, which we prove to be unfeasible.

We prove that previous approaches were not capable of finding the upper-bound performance, and that the differences can mislead the result of the assessment.

Fig. 6. Three example partitions with 20, 30, and 40 regions, respectively. First column: ground truth partition from BSDS500, second column: partition obtained from the merging sequence, and third column: partition reaching the upper-bound performance

6. REFERENCES

[1] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Trans. on IP*, vol. 9, no. 4, pp. 561–576, 2000.

[2] L. Huihai, J. C. Woods, and M. Ghanbari, "Binary partition tree analysis based on region evolution and its application to tree simplification," *IEEE Transactions on IP*, vol. 16, no. 4, pp. 1131–1138, 2007.

[3] J. Cardelino, V. Caselles, M. Bertalmio, and G. Randall, "A contrario hierarchical image segmentation," in *IEEE ICIP*, 2009, pp. 4041 –4044.

[4] K. McGuinness and N. E. O'connor, "A comparative evaluation of interactive segmentation algorithms," *PR*, vol. 43, pp. 434–444, 2010.

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

[6] J. Pont-Tuset and F. Marques, "Contour detection using binary partition trees," in *IEEE ICIP*, 2010, pp. 1609–1612.

[7] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *CVIU*, vol. 110, no. 2, pp. 260–280, 2008.

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

[9] 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 *IEEE ICCV*, 2001, vol. 2, pp. 416–423.

[10] F. Ge, S. Wang, and T. Liu, "Image-segmentation evaluation from the perspective of salient object extraction," in *IEEE Conf. on CVPR*, 2006.

[11] V. Vilaplana, F. Marques, and P. Salembier, "Binary partition trees for object detection," *IEEE Trans. on IP*, vol. 17, no. 11, pp. 2201–2216, 2008.

[12] F. Calderero and F. Marques, "Region merging techniques using information theory statistical measures," *IEEE Trans. on IP*, vol. 19, no. 6, pp. 1567–1586, 2010.

[13] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 2001.

[14] J. S. Cardoso, P. Carvalho, L. F. Teixeira, and L. Corte-Real, "Partition-distance methods for assessing spatial segmentations of images and videos," *CVIU*, vol. 113, no. 7, pp. 811 – 823, 2009.