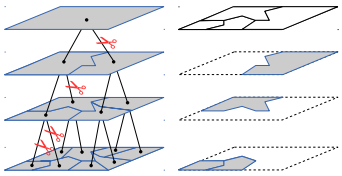
Optimizing Partition Trees for Multi-Object Segmentation with Shape Prior

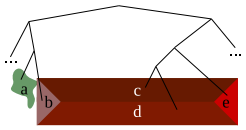
Emmanuel Maggiori[1]
emmanuel.maggiori@inria.fr
Yuliya Tarabalka[1]
yuliya.tarabalka@inria.fr
Guillaume Charpiat[2]
guillaume.charpiat@inria.fr

[1] Inria Sophia Antipolis Méditerranée
Titane team
Sophia Antipolis, France

[2] Inria Saclay
TAO team
Orsay, France



(a) Segmentation as a BPT cut.



(b) Faulty BPT, an object ($bcde$) not represented in a single node.

Figure 1: Binary partition trees.

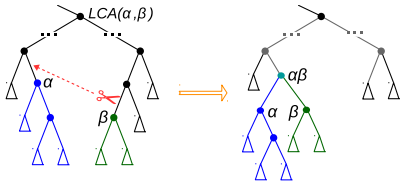
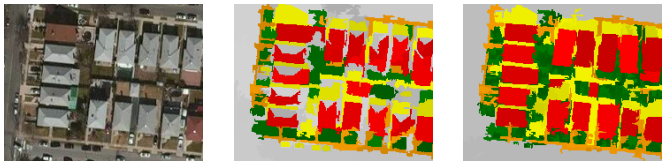
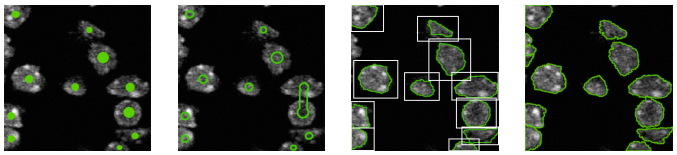


Figure 2: *Prune-and-paste* move to optimize BPTs.



(a) Color image. (b) Unoptimized BPT. (b) Optimized BPT.

Figure 3: Cuts on a BPT with trained color and shape priors.



(a) Input with markers. (b) Method [1]. (c) [1], on user-supplied windows. (d) Optimized BPT.

Figure 4: Convexity prior with [1] and optimized BPTs. [1] must be applied on individual user-supplied windows to account for multiple objects in the scene (b-c). BPT optimization(d) handles multiple objects directly.

The multi-object segmentation of images is one of the great challenges in computer vision. It contemplates the existence of many objects of possibly different classes in the same image. The introduction of **shape descriptors** into segmentation significantly improves its quality. However, it is difficult to optimize energies that involve shape priors because of their non-local nature.

In the last years, hierarchical partitions have been intensively used in different domains. The overall idea is to represent an image as a tree of hierarchical partitions, which is later processed to extract meaningful information. **Binary partition trees** (BPTs) [2] are a particularly efficient structure for this purpose. In our context, multi-object segmentations can be extracted from such trees by performing a cut (see Fig. 1a). Given an energy functional, the optimal cut on a BPT is found in linear time by means of a dynamic programming algorithm.

BPTs can be easily augmented to include shape information, but their traditional bottom-up construction cannot make use of shape constraints, and the tree structure restricts the possible cuts that can be done on them (see e.g., Fig.1b). In this work we exploit the idea of iteratively optimizing the structure of BPTs to produce better partitions with shape priors.

Let $I = (I_j)_{1 \leqslant j \leqslant n}$ be an image containing n pixels. We suppose we are given a set of possible object classes, as well as priors for each class. Multi-label segmentation consists in the partitioning of the pixels into a non-overlapping set of regions $\mathcal{R} = (R_i)$, together with associated class labels $\mathcal{L} = (L_i)$. It can be stated as an optimization problem: minimize

$$E(\mathcal{R},\mathcal{L}) = E_C(I,\mathcal{R},\mathcal{L}) + \sum_{i=1}^{|\mathcal{R}|} E_S(R_i,L_i), \tag{1}$$

where E_C expresses the color prior, and E_S, the shape prior. The color prior is as follows:

$$E_C(I,\mathcal{R},\mathcal{L}) = \sum_{j=1}^{n} -\log P(L_{R(j)}|I_j), \tag{2}$$

where the posteriors $P(L_{R(j)}|I_j)$ can be obtained by training classifiers on pixel color samples. The shape prior term is defined as follows:

$$E_S(R_i,L_i) = -|R_i| \log P(L_i|\mathbf{S}_i), \tag{3}$$

$|R_i|$ being the area of region R_i, and $P(L_i|\mathbf{S}_i)$ being the probability of assigning the label L_i to the region R_i, given a vector $\mathbf{S}_i$ of shape features of that region. The distributions $P(L_i|\mathbf{S}_i)$ are trained from observed data by smoothing feature's histograms.

We wish to enrich the nodes of BPTs by including shape information of the regions. Given that the optimization of the trees will involve recomputing region descriptors, we must design a pool of features that can be computed efficiently from children nodes. We propose to store the **convex hull** of the region at every node. When two regions are merged, the convex hull of the new region can be computed from the children efficiently by using *rotating calipers* [3]. From the convex hull several descriptors can be derived, such as solidity, rectangularity and elongatedness.

To **optimize the BPTs** we follow a local search approach, in which a solution is iteratively modified by performing local transformations on the trees. We propose a move that performs a *prune-and-paste* of a branch into another part of the tree. Fig. 2 illustrates such a move: α is the paste place and β is the pruning place. We denote by $\mathrm{LCA}(\alpha,\beta)$ their lowest common ancestor in the tree. The move creates a new node $\alpha\beta$ in the paste side that comprises α and β. In the pruned side, the tree is collapsed after β is removed. Among the ancestry, only the nodes below $\mathrm{LCA}(\alpha,\beta)$ require to recompute their models (shape and color features), given that further up the regions represented by the nodes do not change.

In our method we simulate all possible moves to compute the impact of each move (ΔC) to the energy in Eq. 1, and construct a priority queue of moves on the values ΔC. In each iteration, we apply the best move, or a number k of independent best moves, prior to reconstructing a new up-to-date queue. When we apply k moves at once, we must verify that the ΔC of a move has not been invalidated by the application of other moves. We have proved theoretical guarantees that allow to reduce the search space of moves, turning the algorithm efficient.

Our experiments validated the effectiveness of our method to make use of shape priors to enhance segmentation, as we show on the satellite image of Fig. 3. We can include different priors simultaneously and the optimization does not depend on which priors were selected. In the case of convexity prior, we compared to [1], as shown on a microscopic image of cell nuclei (Fig. 4). Our method proved then to be an efficient way of introducing shape descriptors in multi-object segmentation.

[1] Lena Gorelick, Olga Veksler, Yuri Boykov, and Claudia Nieuwenhuis. Convexity shape prior for segmentation. In *ECCV*, pages 675–690, 2014.

[2] Philippe Salembier and Luis Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE TIP*, 9(4):561–576, 2000.

[3] Godfried T Toussaint. The rotating calipers: An efficient, multipurpose, computational tool. In *ICCTIM*, pages 215–225, 2014.