Morphological Object Picking Based on the Color Tree of Shapes

Edwin Carlinet$^{12}$ and Thierry Géraud$^1$
$^1$ EPITA Research and Development Lab. (LRDE)
14-16, rue Voltaire, FR-94276 Le Kremlin-Bicêtre, France
{firstname.lastname}@lrde.epita.fr
$^2$ Université Paris-Est,
Laboratoire d’Informatique Gaspard-Monge (LIGM)
A3SI, ESIEE Paris, Cité Descartes, BP 99
FR-93162 Noisy-le-Grand, France

Abstract—The Tree of Shapes is a self-dual and contrast invariant morphological tree that provides a high-level hierarchical representation of images, suitable for many image processing tasks. Despite its powerfulness and its simplicity, it is still under-exploited in pattern recognition and computer vision. In this paper, we show that both interactive and automatic image segmentation can be achieved with some simple tree processings. To that aim, we rely on the “Color Tree of Shapes”, recently defined. We propose a method for interactive segmentation that does not involve any statistical learning, yet yielding results that compete with state-of-the-art approaches. We further extend this algorithm to unsupervised segmentation and give some results. Although they are preliminary, they highlight the potential of such an approach that works in the shape space.

Keywords—Mathematical Morphology; Tree of Shapes; Segmentation; Interactive segmentation.

I. INTRODUCTION

The Tree of Shapes (ToS) [1, 2] (also known as the level-line tree) has shown recently a regain of interest in the image processing community as it allows to represent efficiently the image as a hierarchical structure easy to manipulate. The abstraction offered by the Tree of Shapes enables to perform advanced image processing tasks in a simple way [3]. For example, image simplification can be performed by selecting or removing some nodes in the tree and image denoising can be computed by pruning some branches. Despite the simplicity of the framework, it has shown its powerfulness by competing with the state-of-art methods in many applications: the Tree-Based Morse Regions (TBMR) [4] achieve as well as the MSER for scene matching; the self-dual attribute profiles computed on the ToS get the state-of-art results for multispectral image classification [5]... The versatility of the structure is no more to demonstrate since it has been applied for visualization [6], filtering [7], object detection [8], image registration [9], and image simplification [10, 11]. It is not such a surprise if the Tree of Shapes achieves such good results but it is rather due to the mathematical properties held by the representation [12]. The Tree of Shapes is the support for self-dual, contrast invariant and morphological connected operators. In other words: 1. It enables to process dark objects over bright backgrounds and the contrary in a symmetric way.

2. It allows for recovering few contrasted objects. 3. It does not move the contours of the objects. Put differently, it features many properties of an image representation that one would like to have for image processing.

Yet, the Tree of Shapes is only defined for grayscale images as it requires a total ordering of the image values. We have shown that imposing a total order on colors leads to “strange” behaviors [13]. Following a former proposal [14], we have formalized in [15] a way to compute a Tree of Shapes on multivariate data that does not require to impose any total order. Just like the Tree of Shapes on scalar images, the Color Tree of Shapes features many invariances on color images that are the invariances to any marginal change and/or inversion...
of contrast. More formally, let an image \( u \) with values in \( \mathbb{R}^3 \),
\( u = (u_1, u_2, u_3) \), and let \( F \) be the family \( F = (F_1, F_2, F_3) \) where \( F_i : \mathbb{R} \to \mathbb{R} \) are monotonic functions (\( \forall x, y \in \mathbb{R}, x < y \implies F(x) < F(y) \) or \( F(x) > F(y) \)). Then, the Color Tree of Shapes \( T(u) \) is invariant by marginal inversion/change of contrast that is \( T(F(u)) = T(u) \).

In this paper, we show the relevance of the Tree of Shapes for image segmentation, first in the context of interactive segmentation and then extending the method for unsupervised segmentation where the seeds are detected automatically without requiring any user intervention.

The paper is organized as follows. First, in Section II we remind the basics of the Tree of Shapes and its color extension and present the mainstays of the segmentation method using the tree. Section III shows how our work differs from the other state-of-art object picking methods. Section IV gives an in-depth explanation of the proposed method, as well as a simple extension for non-supervised image segmentation. Section V shows some results using our proposal and we conclude in Section VI.

II. GENERAL OVERVIEW

A. The Color Tree of Shapes

Let an image \( u : \Omega \to E \) defined on a domain \( \Omega \) and taking values on a set \( E \) embedded with an ordering relation \( \leq \). Let, \( [u < \lambda] \) (resp. \( [u > \lambda] \)) with \( \lambda \in \mathbb{R} \) be a threshold set of \( u \) (also called respectively lower cut and upper cut) defined as \( [u < \lambda] = \{ x \in \Omega, u(x) < \lambda \} \). We note \( \mathcal{C}(X), X \in \mathcal{P}(E) \) the set of connected components of \( X \). If \( \leq \) is a total relation, any two connected components \( X, Y \in \mathcal{C}([u < \lambda]) \) are either disjoint or nested. The set \( \mathcal{C}([u < \lambda]) \) endowed with the inclusion relation forms a tree called the min-tree and its dual tree, defined on upper cuts, is called the max-tree. Given the hole-filling operator \( H \), we call a shape any element of the set:

\[
S = \{ H(\Gamma), \Gamma \in \mathcal{C}([u < \lambda]) \cup \mathcal{C}([u > \lambda]) \}_\lambda.
\]

If \( \leq \) is total, any two shapes are either disjoint or nested, hence the cover of \( (S, \subseteq) \) form a tree called the Tree of Shapes (see Fig. 2).

If \( \leq \) is not a total order (e.g., on colors), two shapes may overlap without being nested, and \( (S, \subseteq) \) does not form a tree anymore. Instead of defining a new total order, the Color Tree of Shapes [15] merges some Trees of Shapes computed marginally on each channel of the image into a single structure (a graph) based on the inclusion relation of connected components only. Then, a tree is extracted from the graph based on the graph topology and no more on values. This tree features many properties including the invariance to any marginal inversion/change of contrast.

B. Method Description

The problem of object picking can be summarized as follows: given two sets of points \( F \) and \( B \) in \( \mathcal{P}(\Omega) \) representing the user inputs for the foreground class \( (F) \) and the background \( (B) \), we aim at classifying every point in \( \Omega \) in one of the two classes.

Considering the distance measure between two points \( (p, p') \) in \( \Omega \):

\[
d_{TV}(p, p') = \min_{C_{pp'}} \int_0^1 |\nabla u(C_{pp'}(t)), \dot{C}_{pp'}(t)| dt,
\]

where \( C_{pp'}(t) \) is a path in \( \Omega \) from \( p \) to \( p' \), we can compute the distance between any point \( x \) to the seeds \( F \) and \( B \) and assign the closest class to \( x \). This approach has been used by [16, 17]. However, Dubrovina et al. [18] showed that the proper way to compute such a distance is to have a level set approach and to use the Tree of Shapes.

Thus, a fundamental idea of the method is to use the Color Tree of Shapes representation of the image (instead of working directly on the domain), and to perform the classification on that structure. It applies the same principle as previously (Nearest Neighbor) but uses the tree topology instead of the 2D space topology. The final segmentation is obtained by reconstructing the image from the tree where all nodes have been classified. The method can be summarized in the following steps (see also Fig. 3):

1) Compute the Color Tree of Shapes \( T(u) \) of the image \( u \).
2) Valuate \( T(u) \)’s edges with the distance between nodes (see Section IV for more details).
3) Transpose the user scribbles on \( T(u) \), it gives two seed sets of node for \( F \) and \( B \).
4) Classify every non-seed node as \( F \) or \( B \) by computing its distance to the seed nodes using \( T(u) \)’s topology, and retrieving the label of the closest seed node.
5) Reconstruct the image from the labels of \( T(u) \).
6) Cleanup: keep significant foreground connected components only.

A significant advantage of working in the shape space is the ability to recover regions of interest that are not event marked...
by the user. This feature is interesting for objects composed by
several other objects. Because a shape is a component without
holes, it is enough to select the outer region to retrieve the
whole set of objects (see Fig. 4).

On the other hand, the approach does not compute any
statistics about the regions, but only uses level sets that enable
to recover large components with few user scribbles. The
amount of markers required actually depends on the number
of level lines that separate the background and the foreground.

III. RELATED WORKS

Our approach is similar to the one proposed by Dubrovina
et al. [18] since we both use the Tree of Shapes (or in your case
the Color Tree of Shapes) to perform the classification of the
nodes and then reconstruct the image from the labelized tree.
The main difference is that Dubrovina et al. [18] compute the
tree on the likelihood map where each pixel is the confidence
to be a foreground pixel. It implies a statistical modeling of
the user scribbles and so depends on the accuracy of the
modeling of the probability function. Their work is actually
an extension of [17] giving a better accuracy for the geodesic
distance computation between unlabeled pixels and seeds.
In their work, Bai and Sapiro [17] noted how important is
an efficient density modeling method that enables a better
segmentation while requiring less user scribbles than in their
previous work [16]. But still, the quality of the estimation (so
the results) highly depends on the trade-off between the region
complexity and the number of user markers. Actually, most
image editing or matting state-of-the-art algorithms, including
grabcut [19, 20] (GMM modeling) use statistical learning for
a background/foreground estimation, our method does not.

IV. ALGORITHM DETAILS AND EXTENSIONS

A. The ToS-based Object Picking Algorithm

The construction the Color Tree of Shapes is out of the
scope of the current paper, we mainly detail in this section
the way the ToS is used for segmenting (that are the steps
2–6 of Section II-B).

The second step of the algorithm consists in valuating a
distance on tree edges. Contrary to the standard morphological
trees, a node in the Color Tree of Shapes may be associated
with many colors, as a consequence, we consider the average
colors of the points in the node, and the edge between two
nodes is valuated with the distance between their average color
(in La*b* space). Then, we need to label the nodes as
F or B from the user scribbles. In some rare cases, a node may get
contradictory labels if the user tags some points as F and
B that belong to the same node. In that situation, the node gets
labeled with the majority class.

The 4th step consists in computing for every non-labeled
node S (shape) its distance \(d_T\) and \(d_S\) to the closest
foreground and background seed. This can be achieve in
two passes. We note \(par(S)\), the parent of S in T, and
\(d(S, par(S))\) the distance between the average color of S
and \(par(S)\). At initialization, \(d_T(S) = 0\) if the node has
background scribbles, \(+\infty\) otherwise. The forward step goes
from the leaves to the root, and computes: \(d_T(par(S)) =\min\{d_T(par(S)), d_S(par(S))\}\). The backward step
goes from the root to the leaves, and computes: \(d_T(S) =\min\{d_T(S), d_S(par(S)) + d(S, par(S))\}\). The same computa-

![Fig. 3. Scheme of the proposed method for object picking.](image3)

![Fig. 5. Illustration of the automatic segmentation. Left: Image simplification with the \(\alpha\)-tree (\(\omega = 200\)). Blue squares stand for the centers of the candidate regions (\(\lambda = 3000\)). Middle: The markers computed over the distance map. Right: The multi-classes segmentation based on the seeds found previously. The segmented components are shown with the average color over the region.](image5)
The objective is now to extend this algorithm for automatic segmentation without any user intervention. Instead of a binary segmentation, we first extend the algorithm to a multi-classes classification without any effort. Given a set of marked points \( \{M_1, \ldots, M_n\} \) labeled respectively with classes \( C_1, \ldots, C_n \), we still aim at classifying any other non-labeled data in \( \Omega \). We still use the Nearest Neighbor approach using the distance defined by Eq. (1). The binary to multi-classes classification is thus straightforward. The real challenge is to get automatically the initial set of seeds. For this purpose, we can rely on any initialization method used by other segmentation algorithms (e.g. the minima of the gradient function as for the Watershed Transform, the modes of the density function as for the Mean Shift algorithm...). We arbitrary have chosen to initialize our algorithm from the \( (\alpha, \omega) \)-flat zones of the image that are the set of connected components \( \Gamma \) in the image where the magnitude of the gradient inside \( \Gamma \) cannot exceed \( \alpha \) and the amplitude of the range cannot exceed \( \omega \) [23]. More formally, the initial set of seeds comes from:

\[
\{ \Gamma_1 \} = \{ \Gamma \in \bigcup_{\alpha} CC_{\alpha}(u) \mid \max(u(\Gamma)) - \min(u(\Gamma)) < \omega \}
\]

where \( CC_{\alpha}(u) \) denotes the set of \( \alpha \)-connected components of \( u \). The set \( \{ \Gamma_1 \} \) forms a partition of the image but leads to an over-segmentation (especially on object boundaries with high gradients), thus we only keep the subset \( \{ \Gamma_1^\prime \} \) of \( \{ \Gamma_1 \} \) whose size is above a given threshold. Finally, we compute a thick skeleton of the candidate regions that will be the final markers. The process can be summarized as follows (see also Fig. 5):

1. Compute the \( \alpha \)-tree of \( u \) and retrieve the largest \( \alpha \)-connected components such that their range does not exceed \( \omega \) and their size is above a given threshold \( \lambda \).
2. For each component, compute a distance map from the background and threshold this distance map. It yields thick skeletons of the components that are the automatic markers.
3. Perform the multi-classes segmentation extension of the algorithm seen in Section IV.

### V. Experiments and Discussion

Figures 1 and 6 show some results of the interactive segmentation and automatic segmentation using our approach (implemented using our platform Olena [24, 25]). At this point, and despite its simplicity, our approach for interactive segmentation competes with the state-of-the-art methods such as Grabcut from a qualitative point of view. However it is still premature to have a real comparison (quantitative) with those approaches as we have to tackle some problems first. A common failure case of our method appears when the object of interest is traversed by some level-lines of the background. In that case, we cannot separate correctly the foreground and the background. We have observed in recent experiments that this problem can be partially solved by a different rooting of the Color Tree of Shapes (by choosing the point at infinity in the foreground or background scribbles when constructing the tree). The automatic segmentation extension of this work is still in development and is outperformed by the current state-of-art segmentation methods. The method widely depends on the way the seeds are chosen and those given by the \( \alpha \)-tree.
mainly appear in the background. We have to first improve the seed computation process before any further comparison with other methods. However, regarding the quality of the current results, it tends to show the potential of a direct classification in the shape space and that the Color Tree of Shapes is an adequate structure for segmentation.

VI. CONCLUSION

We have shown the versatility and the potential of the Tree of Shapes for segmentation purposes. We have proposed a marker-based classification using the Color Tree of Shapes which is free of statistical learning and despite its simplicity, yields results similar to the ones obtained by the state-of-the-art methods. We have also presented an extension that enables an automatic segmentation without requiring any user intervention. The automatic segmentation, still at a development process, provides interesting results that show the potential of the Color Tree of Shapes and the advantages of working in the shape space. As a further work, since our approach is already robust without statistical learning, we plan to introduce more sophisticated classification strategies to improve the interactive segmentation that would enable to further compare our work with other state-of-the-art methods. For the unsupervised segmentation part of this work, since the bottleneck lies in the automatic detection of the seeds, we plan to experiment other pre-processing methods to get better markers.

REFERENCES

[19] C. Rother, V. Kolmogorov, and A. Blake, “grabcut”:


