A New Matching Algorithm between Trees of Shapes and its Application to Brain Tumor Segmentation

Nicolas Boutry[1] and Thierry Géraud[1]

EPITA Research and Development Laboratory (LRDE), EPITA, France

Abstract. Many approaches exist to compute the distance between two trees in pattern recognition. These trees can be structures with or without values on their nodes or edges. However, none of these distances take into account the shapes possibly associated to the nodes of the tree. For this reason, we propose in this paper a new distance between two trees of shapes based on the Hausdorff distance. This distance allows us to make inexact tree matching and to compute what we call residual forests, representing where two trees differ. We will also see that thanks to these residual forests, we can obtain good preliminary results in matter of brain tumor segmentation. This segmentation not only provides a segmentation but also the tree of shapes corresponding to the segmentation and its depth map.

Keywords: Mathematical morphology · tree of shape · brain tumor segmentation

1 Introduction

The tree of shapes (ToS) is a hierarchical representation of the boundaries of the objects in an image (they are sometimes called *level-lines*). For sake of completeness, and because we think that many applications can be derived from it, we propose to introduce the first distance between two ToS based on the distance between their shapes. This distance makes us able to obtain a fast graph inexact matching algorithm, whose complexity is in $O(n_1 \times n_2 \times K + n_1^2 + n_2^2)$ where n_1 and n_2 are the numbers of nodes of the trees T_1 and T_2 respectively and where K is the number of operations needed to compute the distance between two shapes. Our methodology is related to the following topics.

Hausdorff distance: The *Hausdorff distance* (HD) is a very powerful tool used in Pattern Recognition to compute the deformation needed to obtain a curve from another. It is much used in image matching [19]. Sometimes, we can prefer to use the ranked Hausdorff distance [18] (which is more robust), or the Gromov-Hausdorff Distance when we want to compute the distance between two metric trees [26].

Distance between graphs: Among the possible distances between trees, we can find the *tree-edit distances* [3]. When hierarchical structures contain cycles,

they are graph and then specific distances can be used [5]. A co-spectral distance between graphs can also be found in [11], where in brief they compute the Laplacian of the adjacency matrix of two given graphs of same number of nodes; after having computed their respective eigenvalues, they compute the squared sum of the differences of the two spectra which leads to the desired distance. From the computational topology point of view, we can recall the distances between Reeb graphs [2].

Graph matching (GM): The references presented here are not exhaustive since according to Conte *et al.* [12], more than 160 publications are related to GM. There exist several approaches: *exact matching methods* that require a strict correspondence among the two objects or among their subparts, and *inexact matching methods* where a matching can occur even if the two graphs being compared are structurally different. Exact ones can be based on tree search [4] or not [25]. Among them, several flavours exist. From the strongest to the weakest forms: the graph isomorphisms which are bijective, the subgraph isomorphisms, the monomorphisms, and the homomorphisms. An alternative approach is to compute maximal common subgraphs (MCS) [5]. These algorithms are NP-complete, and require exponential time in the worst case [12] except for special kinds of graphs. Concerning the inexact ones, they can be based on tree search [34], on continuous optimization [14], on spectral methods [35], or other techniques [20]. They are considered to be either optimal or approximate depending on the case. Usually, a matching cost is associated to these algorithms (like for the tree-edit distance [3]); the aim is then to find a mapping which minimizes this cost. As explained in [7], *relaxation labeling and probabilistic approaches* [6], *semidefinite relaxations* [32], *replicator equations* [29], and *graduated assignments* [17] can also be used to proceed to graph matching. GM algorithms can be based on *similarity functions* [?] to do for example face recognition. Finally, GM can be based on the tree of shapes (see [28]). However, as we will see later, this approach is not "differential" like ours, since it is deserved to locate patterns that are already known and not for patterns that are unknown.

The tree of shapes: the tree of shapes [16, 10] is a hierarchical representation of the *shapes* in an image. Its origin can be found in [22, 27], and its applications are numerous: grain filtering [9], object detection [13], object retrieval [28], texture analysis [36], image simplification and segmentation [37], and image classification [24]. It is mainly known as being the fusion of the min-tree and the max-tree [30].

The paper is organized as follows: Section 2 gives the mathematical background needed in this paper, Section 3 presents our proposition of distance between two trees, Section 4 introduces our tree-matching algorithm, Section 5 demonstrates that the provided tools can be used to do brain tumor segmentation, Section 6 concludes the paper.

2   Mathematical background

The shapes of a real image defined in a finite rectangle Ω in $\mathbb{Z}^2$ are the saturations [10] of the connected components of its (upper and lower) threshold sets. A set T of shapes is then called *tree of shapes* [16] when any two shapes are either nested or disjoint. A *distance* d on a set E is a mapping from $E \times E$ to $\mathbb{R}^+$ which satisfies that for any two elements A, B of E, $d(A, B) = 0$ iff $A = B$, that it is symmetrical, and which satisfies the triangular inequality. Let us denote by μ the *cardinality operator* and by A, B two (finite) subsets of Ω. Then, the mapping d_μ from $E \times E$ to $\mathbb{R}^+$:

$$d_\mu(A, B) = \begin{cases} 0 & \text{if } A \text{ and } B \text{ are empty,} \\ 1 - \frac{\mu(A \cap B)}{\mu(A \cup B)} & \text{otherwise.} \end{cases}$$

is a distance [21] called the *Jaccard distance*. Let (E, d) be some metric space. The *Hausdorff distance* between two finite subsets E_1 and E_2 of E and based on a given distance d is defined as:

$$D_H(E_1, E_2) := \max \left\{ \max_{p_1 \in E_1} \min_{p_2 \in E_2} d(p_1, p_2), \max_{p_2 \in E_2} \min_{p_1 \in E_1} d(p_1, p_2) \right\}.$$

3   A distance between two trees of shapes

Let I_1, I_2 be two images on Ω and T_1, T_2 their respective trees. We define the distance between a shape s_1 of T_1 and T_2 as $d_\mu(s_1, T_2) = \min_{s_2 \in T_2} d_\mu(s_1, s_2)$. Let $\mathcal{T}$ be the set of trees of shapes in Ω. We can define a mapping $d_\mathcal{T}$ from $\mathcal{T} \times \mathcal{T}$ to $\mathbb{R}^+ : d_\mathcal{T}(T_1, T_2) = \max_{s_1 \in T_1} d_\mu(s_1, T_2)$, which is not symmetrical. We finally define $D_\mathcal{T}(T_1, T_2) = \max(d_\mathcal{T}(T_1, T_2), d_\mathcal{T}(T_2, T_1))$. Since Ω is supplied with the distance d_μ, it is metric, and then $D_\mathcal{T}$ is the Hausdorff distance based on the distance d_μ. Let us propose the following proof whose main steps are indicated at `http://www.phys.ens.fr/~chevy/Tutorat/Hausdorff.pdf`, that the mapping $D_\mathcal{T}$ is a distance.

Property 1 *For the Jaccard distance d_μ, the mapping $D_\mathcal{T}$ is a distance.*

Proof: let T_A, T_B, T_C be three elements of $\mathcal{T}$. Then:

1. When $T_A = T_B$, for any $s_A \in T_A$, $\min_{s_B \in T_B} d_\mu(s_A, s_B) = 0$, then for any $s_A \in T_A$, we have $d_\mu(s_A, T_B) = 0$, and then $d_\mathcal{T}(T_A, T_B) = 0$. A symmetrical reasoning shows that $d_\mathcal{T}(T_B, T_A) = 0$, and then $D_\mathcal{T}(T_A, T_B) = 0$. Conversely, $D_\mathcal{T}(T_A, T_B) = 0$ implies that $d_\mathcal{T}(T_A, T_B) = 0$ and $d_\mathcal{T}(T_B, T_A) = 0$. Thanks to $d_\mathcal{T}(T_A, T_B) = 0$, we know that for any $s_A \in T_A$, there exists some $s_B \in T_B$ with $d_\mu(s_A, s_B) = 0$ (and thus $s_A = s_B$). In other words, for any $s_A \in T_A$, $s_A \in T_B$, that is, $T_A \subseteq T_B$. Thanks to $d_\mathcal{T}(T_A, T_B) = 0$, we obtain $T_B \subseteq T_A$. We can conclude with $T_A = T_B$.
2. The symmetry is obtained by construction.

3. Triangular inequality: let us proceed in five steps:
   (a) For any $s_A \in T_A$ and any $s_B \in T_B$, let us prove that:

   $$d_\mu(s_A, T_C) \leq d_\mu(s_A, s_B) + d_\mu(s_B, T_C).$$

   Since d_μ is a distance, for any $s_C \in T_C$:

   $$d_\mu(s_A, s_C) \leq d_\mu(s_A, s_B) + d_\mu(s_B, s_C),$$

   which implies by applying the increasing min operator:

   $$\begin{aligned} d_\mu(s_A, T_C) &= \min_{s_C \in T_C} d_\mu(s_A, s_C) \\ &\leq d_\mu(s_A, s_B) + \min_{s_C \in T_C} d_\mu(s_B, s_C) \\ &\leq d_\mu(s_A, s_B) + d_\mu(s_B, T_C), \end{aligned}$$

   which proves the inequality.
   (b) Now, let us prove that for any $s_A \in T_A$ and any $s_B \in T_B$:

   $$d_\mu(s_A, s_B) + d_\mu(s_B, T_C) \leq d_\mu(s_A, s_B) + D_\mathcal{T}(T_B, T_C).$$

   This property is due to $d_\mu(s_B, T_C) \leq d_\mathcal{T}(T_B, T_C) \leq D_\mathcal{T}(T_B, T_C)$.
   (c) For any s_A in T_A, let us prove that:

   $$d_\mu(s_A, T_C) \leq d_\mu(s_A, T_B) + D_\mathcal{T}(T_B, T_C).$$

   We already know that $d_\mu(s_A, T_C) \leq d_\mu(s_A, s_B) + D_\mathcal{T}(T_B, T_C)$, then thanks to the min operator, we obtain:

   $$\begin{aligned} d_\mu(s_A, T_C) &= \min_{s_B \in T_B} d_\mu(s_A, T_C), \\ &\leq \min_{s_B \in T_B} d_\mu(s_A, s_B) + D_\mathcal{T}(T_B, T_C), \\ &\leq d_\mu(s_A, T_B) + D_\mathcal{T}(T_B, T_C), \end{aligned}$$

   which concludes this part of the proof.
   (d) Thus we obtain:

   $$d_\mu(s_A, T_C) \leq d_\mu(s_A, T_B) + D_\mathcal{T}(T_B, T_C) \leq D_\mathcal{T}(T_A, T_B) + D_\mathcal{T}(T_B, T_C),$$

   which leads to:

   $$\begin{aligned} d_\mathcal{T}(T_A, T_C) &= \max_{s_A \in T_A} d_\mu(s_A, T_C) \\ &\leq \max_{s_A \in T_A} \left(d_\mu(s_A, T_B) + D_\mathcal{T}(T_B, T_C) \right), \\ &\leq D_\mathcal{T}(T_A, T_B) + D_\mathcal{T}(T_B, T_C), \end{aligned}$$

   then with a similar reasoning, we obtain that:

   $$d_\mathcal{T}(T_C, T_A) \leq D_\mathcal{T}(T_A, T_B) + D_\mathcal{T}(T_B, T_C),$$

   and then $D_\mathcal{T}(T_A, T_C) \leq D_\mathcal{T}(T_A, T_B) + D_\mathcal{T}(T_B, T_C)$, which concludes the proof. $\qquad\square$

Property 2 *Let T_1 and T_2 be two trees of shapes defined on the same domain. Let us compute the subsets T_1' and T_2' with $\lambda \geq 0$ a given threshold:*

$$T_1' = \{s \in T_1 \mid d_\mu(s, T_2) \leq \lambda\},$$
$$T_2' = \{s \in T_2 \mid d_\mu(s, T_1) \leq \lambda\}.$$

Then the subtrees T_1' of T_1 and T_2' of T_2 satisfy $D_{\mathcal{T}}(T_1', T_2') \leq \lambda$.

Proof: Let us prove first that:

$$\forall s_1' \in T_1', \ \min_{\forall s_2' \in T_2'} d_\mu(s_1', s_2') \leq \lambda. \quad (P)$$

When (P) is false, there exists some $s_1' \in T_1'$ such that: $\min_{s_2' \in T_2'} d_\mu(s_1', s_2') > \lambda$, that is, for any $s_2' \in T_2'$, we have $d_\mu(s_1', s_2') > \lambda$. However, s_1' belongs to T_1', then $d_\mu(s_1', T_2) \leq \lambda$, then there exists $s_2 \in T_2 \setminus T_2'$ such that $d_\mu(s_1', s_2) \leq \lambda$. By symmetry of d_μ, we have that $d_\mu(s_2, s_1') \leq \lambda$, then $\min_{s_1 \in T_1} d_\mu(s_2, s_1) \leq \lambda$, then $s_2 \in T_2'$. We obtain a contradiction, then (P) is true. By symmetry, we obtain:

$$\forall s_2' \in T_2', \ \min_{s_1' \in T_1'} d_\mu(s_2', s_1') \leq \lambda,$$

thus for any $s_1' \in T_1'$ and for any $s_2' \in T_2'$, $d_\mu(s_1', T_2') \leq \lambda$ and $d_\mu(s_2', T_1') \leq \lambda$, which leads to $D_{\mathcal{T}}(T_1', T_2') \leq \lambda$. □

4   Tree-matching and residual forests

In this section, we present our definition of tree-matching, we explain how we are able to ensure that the Hausdorff distance between two subtrees is lower than a given threshold, and then we introduce our *residual forests*.

4.1   Our definition of tree-matching

In this paper, we consider that two trees T_1 and T_2 computed on the images I_1 and I_2 defined on Ω *match* relatively to a given $\lambda \in \mathbb{R}^+$ when their Hausdorff distance $D_{\mathcal{T}}(T_1, T_2)$ is lower than or equal to the threshold λ. A strong property is that when T_1 and T_2 match relatively to 0, they are identical sets of shapes, since it means that for any shape s_1 in T_1, there exists some shape s_2 in T_2 equal to s_1, and conversely (thanks to the symmetry of $D_{\mathcal{T}}$).

4.2   Subtrees extraction

Now let us assume that we have two trees T_1 and T_2 corresponding to two images I_1 and I_2 respectively, both defined on Ω. We want to find two subtrees T_1' of T_1 and T_2' of T_2 satisfying: $D_{\mathcal{T}}(T_1', T_2') \leq \lambda$ for some $\lambda \in \mathbb{R}^+$. For this aim, it is sufficient to compute: $T_1' = \{s_1 \in T_1 \ ; \ d_\mu(s_1, T_2) \leq \lambda\}$ and $T_2' = \{s_2 \in T_2 \ ; \ d_\mu(s_2, T_1) \leq \lambda\}$. We are ensured that T_1' and T_2' are trees: they are both sets of shapes which are disjoint or nested and they both contain the maximal element Ω. Furthermore, by Property 2, we ensure that the Hausdorff distance between T_1' and T_2' satisfies: $D_{\mathcal{T}}(T_1', T_2') \leq \lambda$, and then we obtain subtrees of T_1 and T_2 which are as much similar as we want.

4.3   Residual forests

Assuming we have computed T_1' and T_2' for a given $\lambda \in \mathbb{R}^+$, we can then remove from T_1 the elements of T_1' (we obtain the forest F_1) and from T_2 the elements of T_2' (we obtain the forest F_2). We call then F_1 and F_2 *residual forests* of T_1 (relatively to T_2) and of T_2 (relatively to T_1) respectively. The connected components of F_1 and F_2, called *residual trees* of I_1 and I_2 respectively, will then represent where I_1 and I_2 differ from each other. Obviously, the lower λ, the bigger the residual forests.

4.4   Complexity

Let us assume that some threshold λ is given. For two given trees of shapes T_1, T_2 of numbers of nodes n_1, n_2 respectively, we can compute a distance matrix M whose element $M_{i,j}$ is equal to $d_\mu(s_i^1, s_j^2)$ where $\{s_i^1\}_i$ are the shapes of T_1 and $\{s_j^2\}_j$ are the shapes of T_2. This is done in $O(n_1 \times n_2 \times K)$ where K is the time needed to compute the distance between two shapes. From this matrix, we can deduce in linear time each term $d_\mu(s_i^1, T_2)$ for each i and $d_\mu(s_j^2, T_1)$ for each j using the min operator. This decision step is then in $O(n_1 \times n_2)$. Now, for each shape that we want to remove from T_1 to finally obtain T_1', we have to remove the corresponding node in T_1 structure, which is linear time. At most we have to do this $(n_1 - 1)$ times, this part is then in $O(n_1^2)$. Obtaining T_2' from T_2 is then in $O(n_2^2)$. The total complexity of our algorithm is then in $O(n_1 \times n_2 \times K + n_1^2 + n_2^2)$.

5   An application: brain tumor segmentation

To show an application of our distance, we propose an algorithm able to do unsupervised brain tumor segmentation [31]. The key idea is the following: assuming that two brains look like each other, except that the first has a tumor and the second has not, the tumor should appear in one of the residual trees of the tumored brain since the residual trees encode the difference between two images. In this experiment, we use the MICCAI BraTS multi-modal[1] dataset [33] to obtain tumored brains, and the OASIS-3 dataset [23] for brains with no tumor. We assume that the brains are aligned. Note that the following experiment is in 2D, but it can easily be extended to n-D [16, 15].

Details of the algorithm: For practical reasons, we normalize the FLAIR and T2 modalities using Gauss normalization (we remove the mean and divide by 5 times the standard deviation), we clip the values between -1 and 1, and we uniformly quantify so that the value space becomes $[\![0, 10]\!]$ for the two FLAIR modalities (tumored and sane brains) and $[\![0, 20]\!]$ for the T2 modality (tumored brain).

---

[1] The BraTS dataset provides FLAIR, T1, T1CE, and T2 modalities for each brain but we will limit us to the FLAIR and T2 modalities.
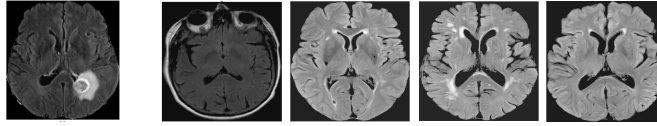
Fig. 1. From left to right, the initial tumored slice where we want to locate the tumor, then the sane slices of similarities equal to 0.396337, 0.537613, 0.558739 and 0.604324 (relatively to the tumored slice). The last one is the best matching brain in the dataset.

The algorithm:

1. We choose one of the 335 brains of size $240 \times 240 \times 155$ in the BraTS 2020 dataset and we extract the slice corresponding to $z = 77$ in the FLAIR modality file (see Figure 1).
2. The similarity between two slices is computed this way:
   - We compute the cross-correlation between the intensities of the two slices (each one has been normalized by its L2 norm), that we name $\mathcal{I}_{\text{sim}}$.
   - We compute the norms of the gradient of both slices in a pixel-wise manner, we normalize by their L2 norms each of these images, and we deduce the cross-correlation $\mathcal{G}_{\text{sim}}$ between these two signals.
   - We compute on the FLAIR images the masks $\text{Fluids}_{\text{BraTS}}$[2] and $\text{Fluids}_{\text{OASIS}}$ (see Section A for the details). We deduce their cross-correlation $\mathcal{F}_{\text{sim}}$ (normalized by the L2 norm).
   - We finally compute the similarity as the weighted sum:

$$1/3 * \mathcal{I}_{\text{sim}} + 1/3 * \mathcal{G}_{\text{sim}} + 1/3 * \mathcal{F}_{\text{sim}}.$$

3. We choose in the database of 749 OASIS-3 FLAIR images (with no tumors) the slice of the brain which best matches with the slice coming from the BraTS database (see Figure 1).
4. We compute the trees of shapes T^{sane} and T^{tum} of the sane brain and of the tumored brain respectively on quantified slices (to limit the number of components in the computed trees).
5. Using grain filtering, we keep in each computed ToS a maximal number of $n = 35$ nodes to obtain the most representative structures in the image. The grain filtering removes all the shapes in the two trees whose area is lower than the one of the nth greater component in each tree. This way we obtain $T_{simp}^{sane} = \{\mathcal{S}_i^{sane}\}_i$ and $T_{simp}^{tum} = \{\mathcal{S}_i^{tum}\}_i$ (see Figure 2).
6. We fix a threshold $\lambda = 0.6$ (empirically chosen) which determines when two shapes will be considered as sufficiently similar.
7. We compute in a matrix M the distances between each shape of the first tree with each shape of the second tree: $M_{i,j} = d_\mu(\mathcal{S}_i^{tum}, \mathcal{S}_j^{sane})$.
8. For T_{simp}^{tum}, we keep only the nodes whose corresponding shape has a distance lower than λ to the other tree T_{simp}^{sane}; we obtain then T_{match}^{tum} (see Figure 3).

---

[2] We call *fluids* the cerebro-spinal fluids, which appears in dark gray in a FLAIR image and which are generally located at the center of the brain.
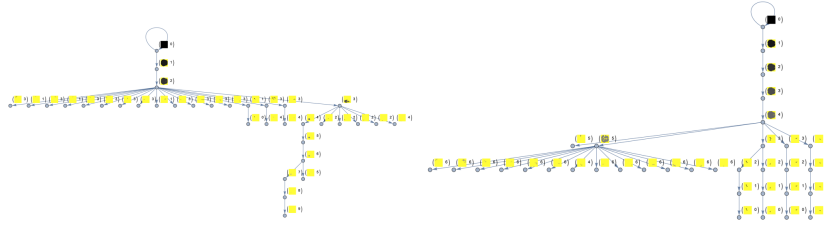
Fig. 2. Filtered trees of the slices of the ill brain and its best-matching sane brain (the background is in yellow, the shapes are in black).

Fig. 3. Extraction of the part of the tree of the tumored brain which best matches with the sane brain. The root of this tree is the only node which loops.
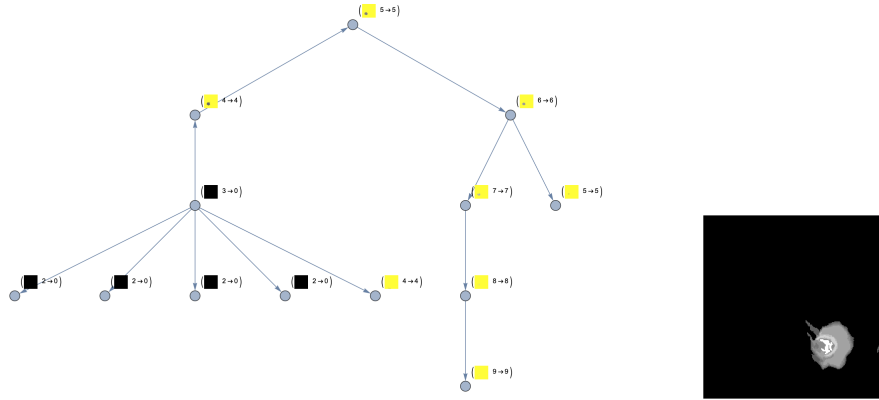

Fig. 4. From left to right, the most relevant residual tree extracted from T_{simp}^{tum} and its depth map. Other trees are filtered out because of their negligible corresponding area in the image.

9. We compute the residual trees $\{T_i^{res}\}_i$ by removing to T_{simp}^{tum} the elements of T_{match}^{tum}: these residual trees correspond to the tumor(s) or to small differences between the two brains (see Figure 4).

10. We set at zero the components of T_i^{res} whose amplitude is too low because low amplitudes are rarely tumors in FLAIR images, at whatever their position (see the black thumbnails in Figure 4); we chose empirically the threshold $(\Xi + \sigma)$ of the BraTS FLAIR image (see Section A for the definitions of Ξ and σ).
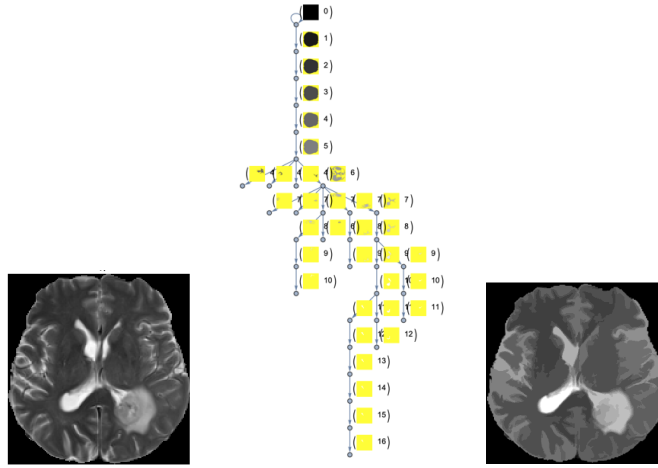
Fig. 5. From left to right, the T2-weighted slice of the tumored brain, its corresponding tree, and then its depth map.

11. Then we compute the tree of shapes T_{T2} of the T2 modality of the same tumored brain as before, we simplify it as usually using a grain filter keeping only the n greatest components, and we deduce the corresponding depth map $\text{depth}_{T_{T2}}$ (see Figure 5) representing the minimal number of level-lines we have to cross to reach a pixel in an image.

12. Using the mask computed before, we deduce the image:

$$\text{depth}'_{T_{T2}} = (1 - \text{Fluids}_{\text{BraTS}}) * \text{depth}_{T_{T2}}$$

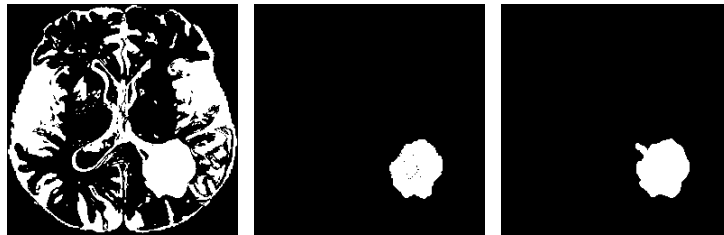which represents the T2-weighted structures in the brain minus the fluids.

Fig. 6. From left to right, the T2-depth mask when $\beta = 0.35 * \max(\text{depth}'_{T_{T2}})$, the binary prediction (the union of the shapes of the residual tree) filtered by the T2-depth mask, and the binary ground truth. We finally obtain a Dice score of 0.953 on this slice.

13. By thresholding this depth map at $\beta = 0.35 * \max(\text{depth}'_{T_{T2}})^3$, we obtain the location(s) in the image where tumors should be (see Figure 6 on its left side).
14. We apply this mask to the segmentation computed from the residual tree, we apply on it some morphological operator (here an opening of radius 2) to obtain the final (smooth) result depicted in Figure 6. Note that the morphological operator is optional and generally does not change much the final Dice score.

The set of parameters presented in this paper has been established based on many tests made on 50 test images.

6   Conclusion

In this paper, we have presented the first distance between two trees of shapes which is computed based on its shape-valued nodes. We have also seen that this distance can be used to compute residual trees representing hierarchies of the locations where two images differ. An application related to brain tumor segmentation is proposed with promising preliminary results, but furthermore it provides a tree of shapes of the segmentation and the corresponding depth map. In the future, we plan to study if our segmentation method can be applied to multimodal signals using the multi-variate tree of shapes [8], to optimize/automatize the choice of the parameters which is crucial in such a methodology, to see if we can increase the Dices thanks to some contrast enhancement algorithm [1], and more generally, we plan to find other applications of this distance. An exploration of tree-traversing methods will also be done to fasten our implementation.

7   Acknowledgements

References

1. Tarik Arici, Salih Dikbas, and Yucel Altunbasak. A histogram modification framework and its application for image contrast enhancement. *IEEE Transactions on Image Pprocessing*, 18(9):1921–1935, 2009.
2. Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring distance between reeb graphs. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 464–473, 2014.

---

[3] We will see later that the value of β can be changed when the contrast around the tumor is very low.

3. Philip Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.
4. Coenraad Bron and Joep Kerbosch. Finding all cliques of an undirected graph (algorithm 457). *Commun. ACM*, 16(9):575–576, 1973.
5. Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.
6. Tibério S Caetano, Terry Caelli, and Dante Augusto Couto Barone. Graphical models for graph matching. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2004.
7. Tibério S Caetano, Julian J McAuley, Li Cheng, Quoc V Le, and Alex J Smola. Learning graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1048–1058, 2009.
8. Edwin Carlinet and Thierry Géraud. Getting a morphological tree of shapes for multivariate images: Paths, traps, and pitfalls. In *Proceedings of the IEEE International Conference on Image Processing*, Paris, France, October 2014.
9. Vicent Caselles and Pascal Monasse. Grain filters. *Journal of Mathematical Imaging and Vision*, 17(3):249–270, 2002.
10. Vicent Caselles and Pascal Monasse. *Geometric description of images as topographic maps*. Springer, 2009.
11. Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Society, 1997.
12. Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.
13. Agnès Desolneux, Loinel Moisan, and Jean-Michel Morel. Edge detection by helmholtz principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.
14. Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 100(1):67–92, 1973.
15. Thierry Géraud, Edwin Carlinet, and Sébastien Crozet. Self-duality and digital topology: links between the morphological tree of shapes and well-composed gray-level images. In *Proceedings of the International Symposium on Mathematical Morphology*, volume 9082 of *Lecture Notes in Computer Science Series*, pages 573–584. Springer, 2015.
16. Thierry Géraud, Edwin Carlinet, Sébastien Crozet, and Laurent Najman. A quasi-linear algorithm to compute the tree of shapes of n-D images. In *Proceedings of the International Symposium on Mathematical Morphology*, volume 7883 of *Lecture Notes in Computer Science Series*, pages 98–110. Springer Berlin Heidelberg, 2013.
17. Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.
18. Daniel P Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
19. Daniel P Huttenlocher, Michael E Leventon, and William J Rucklidge. *Visually-guided navigation by comparing two-dimensional edge images*. Cornell University, Department of Computer Science, 1994.
20. Les Kitchen. Discrete relaxation for matching relational structures. Technical report, Maryland Univ College Park Computer Science Center, 1978.
21. Sven Kosub. A note on the triangle inequality for the Jaccard distance. *Pattern Recognition Letters*, 120:36–38, 2019.

22. A.S. Kronrod. On functions of two variables. *Uspehi Mathematical Sciences*, 5:24–134, 1950. In Russian.

23. Pamela J LaMontagne, Tammie LS Benzinger, John C Morris, Sarah Keefe, Russ Hornbeck, Chengjie Xiong, Elizabeth Grant, Jason Hassenstab, Krista Moulder, Andrei Vlassenko, et al. Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *medRxiv*, 2019.

24. Bin Luo and Liangpei Zhang. Robust autodual morphological profiles for the classification of high-resolution satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1451–1462, 2014.

25. Brendan D McKay et al. *Practical graph isomorphism*. Department of Computer Science, Vanderbilt University Tennessee, USA, 1981.

26. Facundo Memoli. On the use of Gromov-Hausdorff Distances for Shape Comparison. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007.

27. Pascal Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Transactions Image Processing*, 9(5):860–872, 2000.

28. Yongsheng Pan, J. Douglas Birdwell, and Seddik M. Djouadi. Preferential image segmentation using trees of shapes. *IEEE Transactions Image Processing*, 18(4):854–866, April 2009.

29. Marcello Pelillo. Replicator equations, maximal cliques, and graph isomorphism. In *Advances in Neural Information Processing Systems*, volume 11, pages 1933–1955, 1999.

30. Philippe Salembier, Albert Oliveras, and Luis Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

31. Nicolas Sauwen, M Acou, S Van Cauter, DM Sima, J Veraart, Frederik Maes, Uwe Himmelreich, E Achten, and Sabine Van Huffel. Comparison of unsupervised classification methods for brain tumor segmentation using multi-parametric mri. *NeuroImage: Clinical*, 12:753–764, 2016.

32. Christian Schellewald. *Convex mathematical programs for relational matching of object views*. PhD thesis, Universität Mannheim, 2004.

33. Spyridon Bakas et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

34. Wen-Hsiang Tsai and King-Sun Fu. Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(12):757–768, 1979.

35. Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.

36. Gui-Song Xia, Julie Delon, and Yann Gousseau. Shape-based invariant texture indexing. *International Journal of Computer Vision*, 88(3):382–403, 2010.

37. Yongchao Xu, Thierry Géraud, and Laurent Najman. Salient level lines selection using the Mumford-Shah functional. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1–5, 2013.

A   Supplementary materials

Formal computation of Fluids$_{\text{BraTS}}$ and Fluids$_{\text{OASIS}}$ For a given FLAIR image $\mathscr{F}$ taken in the BraTS (resp. in the OASIS3) dataset, we compute the mask $\mathscr{M}$ of fluids this way. For $(x, y) \in [\![1, sx]\!] \times [\![1, sy]\!]$ with $sx = sy = 240$:

$$\mathscr{M}(x, y) := \begin{cases} 1 \text{ if } \sqrt{(x - x_c)^2 + (y - y_c)^2} \leq R \text{ and } \mathscr{F}(x, y) \leq \Xi - 0.5 * \sigma, \\ 0 \text{ otherwise,} \end{cases}$$

with $(x_c, y_c) = (119, 119)$ the coordinates of the center of the image $\mathscr{F}$, $R = \sqrt{sx^2 + sy^2}/4$, and Ξ and σ respectively the statistical mean and standard deviation of the intensity of $\mathscr{F}$. It will lead to Fluids$_{\text{BraTS}}$ in the first case and to Fluids$_{\text{OASIS}}$ in the second case.

Table 1. Dice scores with $\beta = \gamma * \max(\text{depth}'_{T_{T2}})$.

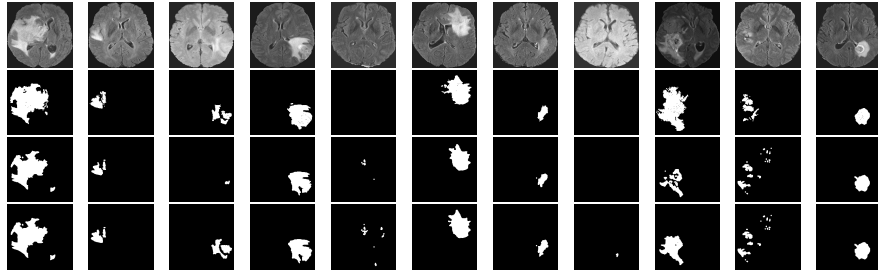| Patient Number	1	2	3	4	5	6	7	8	9	10	11
Dice score ($\gamma = 0.35$)	0.873	0.821	0.142	0.913	N/A	0.859	0.875	N/A	0.442	0.444	**0.953**
Dice score ($\gamma = 0.2$)	**0.873**	**0.871**	**0.860**	**0.923**	N/A	**0.895**	**0.923**	N/A	**0.642**	**0.474**	0.944
Tumor? ($z = 77$)	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes
Contrast	High	Low	Low	High	N/A	High	Low	N/A	Low	Low	High

Fig. 7. From top to bottom, the FLAIR slices of the eleven studied patients, the ground truths, the predictions with γ set at 0.35 and the predictions with γ set at 0.2.

Dice scores on the 10 first patients on BraTS dataset Table 1 and Figure 7 show the results obtained with β set at different values. We can see that, depending on the contrast around the tumor, the optimal choice of β is not the same.