Differential Segmentation using the Tree of Shapes and the Hausdorff Distance

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 are based on the shapes that are associated to the nodes of the tree. For this reason, we propose in this paper a new distance between two trees of shapes using the Hausdorff distance formula. This distance is an interesting way to make inexact attributed tree matching in quadratic time relatively to the number of vertices. An application related to brain tumor segmentation on the MICCAI BraTS data-set is realized.

1 Introduction

The tree of shapes 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 procedures can be derived from it, we propose to introduce the first distance between trees of shapes. This distance makes us able to compute a fast alternative to graph matching (our complexity is quadratic as a function of the number of nodes).

Let us propose now several states-of-the-art, related to the Hausdorff distance since it is the one we use to compute the distance between two trees, to distance between graphs (and trees) in general, to graph matching, and then to the tree of shapes.

Hausdorff distance: The *Hausdorff distance* (HD) is a very powerful tool used in Pattern Recognition to compute the deformation need 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 [6]. From the computational topology point of view, we can recall the distances between Reeb graphs [2], or the interleaving distance between merge trees [28].

Graph matching: as we will see later, our technique is an alternative to graph matching algorithms. For this reason, we propose a brief state-of-the-art of this

topic. The references presented here are not exhaustive since according to Conte *et al.* [12], more than 160 publications talk about graph matching. There exist several approaches to make *graph matching*: *exact matching methods* that require a strict correspondence among the two objects being matched or at least among their subparts, and *inexact matching methods* where a matching can occur even if the two graphs being compared are structurally different to some extent. Exact ones can be based on tree search [5] or not [25]. They can also be elaborated for special kinds of graphs [1]. 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) [6]. 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 [38], on continuous optimization [14], on spectral methods [39], 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. We will see later that it is not the case in this paper. As explained in [8], *relaxation labeling and probabilistic approaches* [7], *semidefinite relaxations* [33], *replicator equations* [31], and *graduated assignments* [17] can also be used to proceed to graph matching. Graph matching algorithms can be based on *similarity functions* [40] to do for example face recognition. Finally, graph matching can be based on the tree of shapes (see [29]). 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 [30], texture analysis [41], image simplification and segmentation [42], and image classification [24]. It is mainly known as being the fusion of the min-tree and the max-tree [32].

1.1 Plan

The plan is the following: Section 2 expose 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 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 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 is the *cardinality operator* and by A, B two (finite) subsets of Ω. Then, the mapping d_μ from $E \times E$ to R^+:

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

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

$$d_\mu(s_1, T_2) = \min_{\forall s_2 \in T_2} d_\mu(s_1, s_2).$$

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 R^+ :

$$d_\mathcal{T}(T_1, T_2) = \max_{\forall s_1 \in T_1} d_\mu(s_1, T_2),$$

which is not symmetrical. To make it symmetrical, we finally define:

$$D_\mathcal{T}(T_1, T_2) = \max\left(d_\mathcal{T}(T_1, T_2), d_\mathcal{T}(T_2, T_1)\right).$$

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, strongly inspired from [11], that the mapping $D_\mathcal{T}$ is a distance.

Property 1 *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 $d_\mathcal{T}(T_A, T_B)$. A symmetrical reasoning shows that $d_\mathcal{T}(T_B, T_A)$, and then $D_\mathcal{T}(T_A, T_B) = 0$. Conversely, $D_\mathcal{T}(T_A, T_B) = 0$ implies that $d_\mathcal{T}(T_A, T_B) = 0$. Then, for any s_A in T_A, $d_\mu(s_A, T_B) = 0$ and then $\min_{s_B \in T_B} d_\mu(s_A, s_B)$. In other words, there exists for each $s_A \in T_A$ some $s_B \in T_B$ whose distance is equal to 0, they are then equal. Then $T_A \subseteq T_B$. A symmetrical reasoning shows that $T_B \subseteq T_A$, and then $T_A = T_B$.

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:

$$\begin{aligned} d_\mu(s_A, T_C) &\leq min_{s_C 2 T_C} d_\mu(s_A, s_C) \\ &\leq d_\mu(s_A, s_B) + \min_{s_C 2 T_C} d_\mu(s_B, s_C) \\ &\leq d_\mu(s_A, s_B) + d_\mu(s_B, T_C), \end{aligned}$$

   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 $d_\mu(s_A, T_C) \leq d_\mu(s_A, s_B) + d_\mu(s_B, T_C)$, then thanks to the min operator, we obtain:

$$\begin{aligned} d_\mu(s_A, T_C) &= \min_{s_B 2 T_B} d_\mu(s_A, T_C), \\ &\leq \min_{s_B 2 T_B} d_\mu(s_A, s_B) + \min_{s_B 2 T_B} d_\mu(s_B, T_C), \\ &\leq d_\mu(s_A, T_B) + \max_{s_B 2 T_B} d_\mu(s_B, T_C), \\ &\leq d_\mu(s_A, T_B) + d_\mu(s_B, T_C), \\ &\leq d_\mu(s_A, T_B) + D_\mathcal{T}(T_B, T_C), \end{aligned}$$

   which concludes this part of the proof.
   (d) Obviously, we have:

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

   since:

$$d_\mu(s_A, T_B) \leq \max_{s_A 2 T_A} d_\mu(s_A, T_B) \leq d_\mathcal{T}(T_A, T_B) \leq D_\mathcal{T}(T_A, T_B).$$

(e) We can then conclude by grouping the last two inequalities, and 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:

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

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. □

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

$$\begin{aligned} T_1^\emptyset &= \{s \in T_1 \mid d_\mu(s, T_2) \leq \lambda\}, \\ T_2^\emptyset &= \{s \in T_2 \mid d_\mu(s, T_1) \leq \lambda\}, \end{aligned}$$

Then the subtrees $T_1^\emptyset$ of T_1 and $T_2^\emptyset$ of T_2 satisfy:

$$D_\mathcal{T}(T_1^\emptyset, T_2^\emptyset) \leq \lambda.$$

Proof:: Let us prove first that:

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

When (P) is false, there exists some $s_1^\emptyset \in T_1^\emptyset$ such that:

$$\min_{s_2' \in T_2'} d_\mu(s_1^\emptyset, s_2^\emptyset) > \lambda,$$

that is, for any $s_2^\emptyset \in T_2^\emptyset$, we have $d_\mu(s_1^\emptyset, s_2^\emptyset) > \lambda$. However, because $s_1^\emptyset$ belongs to $T_1^\emptyset$, $d_\mu(s_1^\emptyset, T_2) \leq \lambda$, then there exists $s_2 \in T_2$ such that $d_\mu(s_1^\emptyset, s_2) \leq \lambda$, and $s_2 \notin T_2^\emptyset$. By symmetry of d_μ, we have that $d_\mu(s_2, s_1^\emptyset) \leq \lambda$, then:

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

since $s_1^\emptyset \in T_1^\emptyset \subseteq T_1$, then $s_2 \in T_2^\emptyset$. We obtain a contradiction, then (P) is true. By symmetry, we obtain:

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

then for any $s_1^\emptyset \in T_1^\emptyset$ and for any $s_2^\emptyset \in T_2^\emptyset$, $d_\mu(s^\emptyset 1, T_2^\emptyset) \leq \lambda$ and $d_\mu(s_2^\emptyset, T_1^\emptyset) \leq \lambda$, which leads to $D_\mathcal{T}(T_1^\emptyset, T_2^\emptyset) \leq \lambda$. □

4 Tree-matching and residual trees

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 explain the concepts of residual forests and trees.

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 $\lambda \in \mathsf{R}^+$ when their Hausdorff distance $D_{\mathcal{T}}(T_1, T_2)$ is lower than or equal to a given threshold $\lambda \geq 0$.

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 $T_1^\emptyset$ of T_1 and $T_2^\emptyset$ of T_2 satisfying: $D_{\mathcal{T}}(T_1^\emptyset, T_2^\emptyset) \leq \lambda$ for some $\lambda \in \mathsf{R}^+$. For this aim, it is sufficient to compute: $T_1^\emptyset = \{s_1 \in T_1 \ ; \ d_\mu(s_1, T_2) \leq \lambda\}$ and $T_2^\emptyset = \{s_2 \in T_2 \ ; \ d_\mu(s_2, T_1) \leq \lambda\}$. We are ensured that $T_1^\emptyset$ and $T_2^\emptyset$ 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 $T_1^\emptyset$ and $T_2^\emptyset$ satisfies:

$$D_{\mathcal{T}}(T_1^\emptyset, T_2^\emptyset) \leq \lambda,$$

and then we obtain subtrees of T_1 and T_2 which are as much similar as we want.

4.3 Residual trees

Assuming we have computed $T_1^\emptyset$ and $T_2^\emptyset$ for a given $\lambda \in \mathsf{R}^+$, we can then remove from T_1 the elements of $T_1^\emptyset$ (we obtain the forest F_1) and from T_2 the elements of $T_2^\emptyset$ (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 will be the residual forests.

5 An application: brain tumor segmentation

We propose to present an algorithm making unsupervised brain tumor segmentation in MRI images [37, 34, 4, 35, 36]: we compare the tree of shape of a brain

which possibly has a tumor with the tree of shapes of a brain without tumor. Necessarily, both brains must look like as each other. By fixing some threshold, we are able to consider which residual trees of the ill brain which may be tumors: *if a shape of the ill brain is too much different from the shapes of the tree of a sane brain similar to it, it may be that a tumor is present.* Note that the following experiment is in 2D, but it can easily be extended to n-D [16, 15]. For this aim, let us propose the following algorithm:

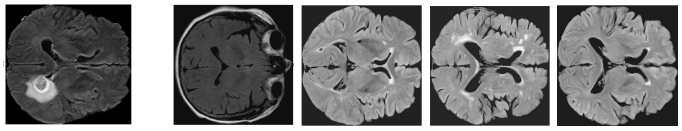

Fig. 1. From left to right, the initial FLAIR image where we want to locate the tumor, then OASIS-3 brain slices of similarities equal to 0.396337, 0.537613, 0.558739 and 0.604324 respectively with to the tumored brain slice. The last one is the best matching brain in the database.

1. We choose one of the 335 brains of size $240 \times 240 \times 155$ in the BraTS 2019 database 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 $IntensitySim$.
   - 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 between these two signals, we name this last value $GradientSim$.
   - We compute on the FLAIR image the mask $MaskFluidsBraTS$ corresponding to the pixels whose distance to the center of coordinates $(119, 119)$ is below a fixed value $R = \sqrt{sx^2 + sy^2}/4$ where $sx = sy = 240$ and whose value is lower than $\Xi - 0.5 * \sigma$ (Ξ and σ are the statistical mean and standard deviation of the intensity of this slice). We proceed this way for $MaskFluidsOASIS$. We deduce the cross-correlation $FluidsSim$ (normalized by the L2 norm) of these two masks.
   - We finally compute the similarity as the weighted sum:

$$1/3 * IntensitySim + 1/3 * GradientSim + 1/3 * FluidsSim.$$

3. We choose in the database of 749 OASIS-3 [23] FLAIR images (with no tumors) the slice of the brain which best matches with the slice coming from the BraTS database (see Figure 1). Note that we assume that the tumor does not introduce too much bias in the similarity computation.

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). We use a uniform quantification so that the value space is $[\![0, 10]\!]$.

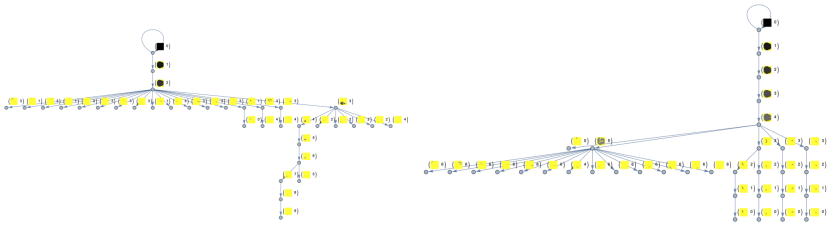

Fig. 2. Filtered trees of the slices of the ill brain and its best-matching sane brain (the background is in yellow, the shapes are in black).

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 n^{th} greater component in each tree. This way we obtain $T^{sane}_{simp} = \{\mathcal{S}^{sane}_i\}_i$ and $T^{tum}_{simp} = \{\mathcal{S}^{tum}_i\}_i$ (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: $M_{i,j} = d_\mu(\mathcal{S}^{tum}_i, \mathcal{S}^{sane}_j)$.

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

8. For T^{tum}_{simp}, we keep only the nodes whose corresponding shape has a distance lower than λ to the other tree T^{sane}_{simp}; we obtain then T^{tum}_{match} (see Figure 3); we can do the same thing for the other tree to ensure that the final distance is lower than or equal to λ but here we will not use the sane tree anymore.
9. We compute the residual trees $\{T^{res}_i\}_i$ by removing to T^{tum}_{simp} the elements of T^{tum}_{match}: 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^{res}_i whose amplitude is too low because low amplitudes are rarely tumors in FLAIR images, whatever their position

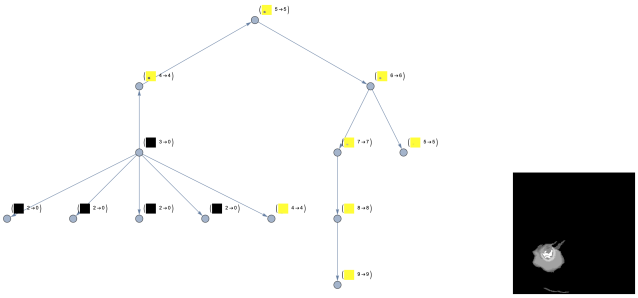

Fig. 4. First residual tree extracted from the grain-filtered tree of the tumored brain and the corresponding depth map. The other residual trees, not shown here, depict only small differences between the two images.

(see the black thumbnails in Figure 4); we chose empirically the threshold $(\Xi + \sigma)$.

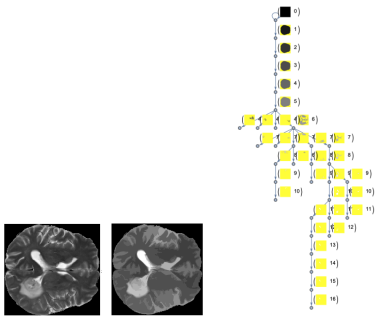

Fig. 5. From left to right, the T2-weighted slice of the tumored brain, the depth image of the grain-filtered T2 slice, and then the corresponding tree.

11. Then we compute the tree of shapes T_{T2} of the quantized T2 modality, 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).
12. Using the mask computed before, we deduce the image:

$$\text{depth}^{\emptyset}_{T_{T2}} = (1 - MaskFluidsBraTS) * \text{depth}_{T_{T2}}$$

which represents the T2-weighted structures in the brain minus the fluids.
13. By thresholding this depth map at $\beta = 0.5 * \max(\text{depth}^{\emptyset}_{T_{T2}})$, we obtain the location(s) in the image where tumors should be (see Figure 6).
14. We can then apply this mask to each residual tree to obtain the predictions of our method (see Figure 6).

Fig. 6. From left to right, the T2-depth mask, the segmentation of the tumor using the first residual tree filtered by the T2-depth mask, and the ground truth.

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 has been proposed. In the future, we plan to find other applications of this distance.

7 Acknowledgements

Data were provided by OASIS-3 (Principal Investigators: T. Benzinger, D. Marcus, J. Morris; NIH P50AG00561, P30NS09857781, P01AG026276, P01AG003991, R01AG043434, UL1TR000448, R01EB009352. AV-45 doses were provided by Avid Radiopharmaceuticals, a wholly owned subsidiary of Eli Lilly).

References

1. Alfred V Aho and John E Hopcroft. *The design and analysis of computer algorithms*. Pearson Education India, 1974.
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. Philip Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.
4. Bjoern Menze et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.
5. Coenraad Bron and Joep Kerbosch. Finding all cliques of an undirected graph (algorithm 457). *Commun. ACM*, 16(9):575–576, 1973.
6. Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, 19(3-4):255–259, 1998.
7. 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, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.
8. 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.

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. Frédéric Chevy. Ensembles fractals.
12. Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and arti cial 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*, 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*, 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, Sarah Keefe, Wallace Lauren, Chengjie Xiong, Elizabeth A Grant, Krista L Moulder, John C Morris, Tammie LS Benzinger, and Daniel S Marcus. Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer's disease. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, 14(7):P1097, 2018.
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 Mémoli. On the use of Gromov-Hausdorff distances for shape comparison. 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. Dmitriy Morozov, Kenes Beketayev, and Gunther Weber. Interleaving distance between merge trees. *Discrete and Computational Geometry*, 49(22-45):52, 2013.
29. Yongsheng Pan, J Douglas Birdwell, and Seddik M Djouadi. Preferential image segmentation using trees of shapes. *IEEE Transactions on Image Processing*, 18(4):854–866, 2009.
30. 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.
31. Marcello Pelillo. Replicator equations, maximal cliques, and graph isomorphism. In *Advances in Neural Information Processing Systems*, pages 550–556, 1999.
32. 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.
33. Christian Schellewald. *Convex mathematical programs for relational matching of object views*. PhD thesis, Universität Mannheim, 2004.
34. Spyridon Bakas et al. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scienti c data*, 4:170117, 2017.
35. Spyridon Bakas et al. Segmentation labels and radiomic features for the preoperative scans of the TCGA-GBM collection. *The Cancer Imaging Archive*, 286, 2017.
36. Spyridon Bakas et al. Segmentation labels and radiomic features for the preoperative scans of the TCGA-LGG collection. *The Cancer Imaging Archive*, 286, 2017.
37. 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.
38. 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.
39. Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence*, 10(5):695–703, 1988.
40. Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. 1996.
41. Gui-Song Xia, Julie Delon, and Yann Gousseau. Shape-based invariant texture indexing. *International Journal of Computer Vision*, 88(3):382–403, 2010.
42. 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.