GEOMETRICAL GRAPH MATCHING USING MONTE CARLO TREE SEARCH

Miguel Amável Pinheiro and Jan Kybic

Center for Machine Perception, Dept. of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic

ABSTRACT

Many medical images contain graph-like geometrical structures such as blood vessels and neuronal networks. We present an algorithm for matching geometrical graphs, in order to quickly and robustly align such images. We use a sampling-based curve descriptor to prune dissimilar edges. The matching is modeled as a single-player game, growing the matching from a random initial correspondence. The coarse global solution is found using a Monte Carlo Tree Search and then refined locally. We show experimentally that our approach finds the correct matching in all tested datasets and is the fastest of all global methods.

***Index Terms*—** graph matching, image registration, Monte Carlo Tree Search, path descriptor

1. INTRODUCTION

Graph and tree shaped structures such as nerve fibers, neuronal networks or blood vessels are commonly found in medical imaging. Our aim is to develop a fast method, which can handle big volumes, very different appearances (see Fig. 1) and large initial transformations. We suggest to match the graphs previously extracted from the images by some automatic method, such as [1]. The matching is performed by a Monte Carlo Tree Search (MCTS) [2], with curve descriptors [3] and a geometrical model for quick pruning of not feasible matches. Once enough graph vertices are matched, fine alignment is performed by solving the assignment problem constrained by the already found matches [4]. The transformation function is then interpolated everywhere.

We test the robustness of our method against the state of the art in terms of speed and alignment quality.

2. RELATED WORK

By ignoring connectivity, graph matching is often reduced to a point cloud matching problem. RANSAC-like techniques sample partial matches between points to find the correspondence [5, 6, 7], however this becomes intractable for a large number of points. Local techniques such as ICP or CPD [8, 9] are faster, however require a good initial estimation of the transformation.

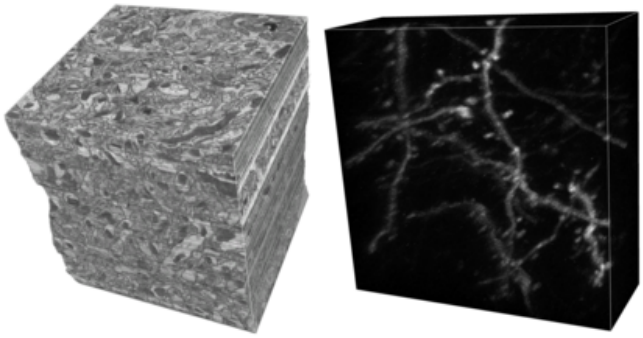

Fig. 1. Brain tissue acquired with electron microscopy (left) and light microscopy (right). The scales are very different and the 3D structure of the volume on the left is contained in the volume on the right.

More generally, the problem of graph matching can be formulated as an optimization problem where affinities are determined from local geometric features between the vertices and edges [10, 11]. So far the best performing is an Active Testing method which builds the match incrementally, guided by quality estimates from partial results [4]. However, neither of the methods is sufficiently fast on larger graphs.

The use of path descriptors was introduced in [3], where edges were modeled as B-splines and the matching was formulated as an integer quadratic program, using affinities between paths. In this paper, we improve the curve descriptors. The key contribution is a novel search algorithm based on the Monte Carlo Tree Search algorithm, which decreases the computational time a factor of more than 50 on some datasets.

3. PROBLEM DEFINITION

Let us have a *geometrical* graph $\mathbf{G}^A = (\mathbf{V}^A, \mathbf{E}^A)$, where each vertex $\mathbf{v}_i^A \in \mathbf{V}^A$ is associated with a point in $\mathbb{R}^D$ and each edge $\mathbf{e}_k^A \in \mathbf{E}^A$ is associated with a continuous path in

This work was supported by the Czech Science Foundation project 14-21421S, the Grant Agency of the Czech Technical University in Prague grant SGS15/156/OHK3/2T/13 and the Fundação para a Ciência e Tecnologia grant SFRH/BD/77134/2011.

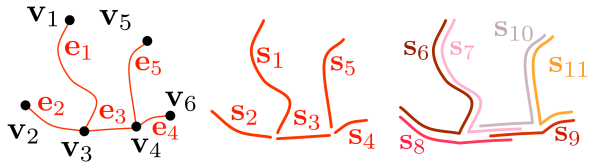

Fig. 2. Example of the superedges (middle and right) of a graph (left), with $K = 2$

$\mathbb{R}^D$ between two vertices. We also define *superedges* $\mathbf{S}^A$ as paths of up to K consecutive edges (see Fig. 2). A second graph $\mathbf{G}^B$ is assumed to be related to $\mathbf{G}^A$ by a geometrical transformation. Our task is to find mappings $M^{\mathbf{V}}\colon \mathbf{V}^A \to \mathbf{V}^B$ and $M^{\mathbf{S}}\colon \mathbf{S}^A \to \mathbf{S}^B$ between vertices and superedges of the two graphs, respectively, where $M^{\mathbf{V}}$ can be inferred from $M^{\mathbf{S}}$. These mappings can be partial if outliers are present.

4. PATH DESCRIPTORS

In biological applications, the geometrical transformation between the images is typically composed of a rigid motion composed with a small nonlinear deformation due to tissue deformation. In particular, we assume that for any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ our transformation T is bi-Lipschitz:

$$\frac{1}{1+\varepsilon_T} d(\mathbf{x},\mathbf{y}) \le d(T(\mathbf{x}),T(\mathbf{y})) \le (1+\varepsilon_T) d(\mathbf{x},\mathbf{y}), \quad (1)$$

where $d(\mathbf{x},\mathbf{y})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.

We propose the use of an improved version of the *path descriptors* [3] to establish a set of compatible superedge pairs $\mathcal{P}_s$.

Let $\zeta_{\mathbf{s}_k}\colon [0,1] \to \mathbb{R}^D$ be a constant speed representation of the curve associated to a superedge $\mathbf{s}_k = (\mathbf{v}_i, \mathbf{v}_j)$. Each edge is described by a series of points instead of a B-spline as in [3]. We therefore find $\zeta_{\mathbf{s}_k}$ using linear interpolation between the points of each edge. We define a path descriptor as $\mathbf{h}_{\boldsymbol{\Omega}}(\mathbf{s}_k) = (h_{\boldsymbol{\omega}_1}, \cdots, h_{\boldsymbol{\omega}_{|\boldsymbol{\Omega}|}})$, where

$$h_{\boldsymbol{\omega}}(\mathbf{s}_k) = \sum_{i=0}^{n_\omega} d\big(\zeta_{\mathbf{s}_k}(t_i), \zeta_{\mathbf{s}_k}(t_{i+1})\big), \quad (2)$$

where $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_{|\boldsymbol{\Omega}|})$ are user defined fixed parameters, and t_i are chosen such that $d\big(\zeta(0), \zeta(t_i)\big) = \omega_i d\big(\zeta(0), \zeta(1)\big)$. Each vector $\boldsymbol{\omega}_j = (\omega_0, \cdots, \omega_{n_\omega+1})$ has elements $0 = \omega_0 < \omega_1 < \cdots < \omega_{n_\omega} < \omega_{n_\omega+1} = 1$, where $\omega_1, \cdots, \omega_{n_\omega}$ are randomly generated for each $\boldsymbol{\omega}_j$ of $\boldsymbol{\Omega}$.

Inspired by (1), we shall consider two superedges $(\mathbf{s}_k^A, \mathbf{s}_l^B)$ from $\mathbf{G}^A$ and $\mathbf{G}^B$, respectively, to be compatible, i.e. $(\mathbf{s}_k^A, \mathbf{s}_l^B) \in \mathcal{P}_s$, if

$$\frac{1}{1+\varepsilon_h} h_{\boldsymbol{\omega}_i}(\mathbf{s}_k^A) \le h_{\boldsymbol{\omega}_i}(\mathbf{s}_l^B) \le (1+\varepsilon_h) h_{\boldsymbol{\omega}_i}(\mathbf{s}_k^A),\ \forall i \in \{1, |\boldsymbol{\Omega}|\}, \quad (3)$$

for some fixed $\varepsilon_h \approx \varepsilon_T$. We have shown [3] that descriptors of this type are discriminative, yet fast to calculate.

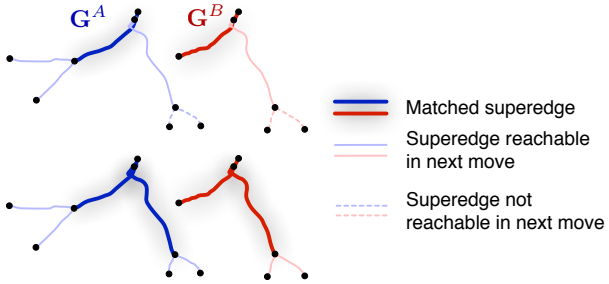

Fig. 3. Example for possible next moves (the top pair is state n and the bottom pair is state $n+1$) for a simplified case of superedges with $K = 1$. The superedges reachable in each state are connected with already matched superedge pairs.

5. MONTE CARLO TREE SEARCH

5.1. Rules of the game

We propose to find $M^{\mathbf{S}}$ (and $M^{\mathbf{V}}$) using a search algorithm inspired by the Upper Confidence Bound on Trees (UCT), a variant of the Monte Carlo Tree Search (MCTS) [2]. We formulate the problem as a single-player game. The objective is to maximize the length of matched edges plus a reward for the number of matched vertices:

$$\begin{aligned} Q(M^{\mathbf{V}}, M^{\mathbf{S}}) &= \sum_{(\mathbf{s}_k^A, \mathbf{s}_l^B) \in M^{\mathbf{S}}} \frac{l(\mathbf{s}_k^A) + l(\mathbf{s}_l^B)}{2} \\ &+ \kappa\, \overline{l(\mathbf{S}^A, \mathbf{S}^B)}\, |M^{\mathbf{V}}|, \end{aligned} \quad (4)$$

where $l(\mathbf{s}_k)$ is the superedge length, $\overline{l(\mathbf{S}^A, \mathbf{S}^B)}$ is the average length of the superedges of both graphs, and κ is a user-defined parameter. Note that our superedge formulation allows up to $K-1$ nodes to be skipped; the $|M^{\mathbf{V}}|$ term serves to discourage skipping if nodes are present in both images.

The player maximizes Q by adding pairs of superedges one by one to the matching starting from an empty set, obeying the following rules:

- The pair of superedges $(\mathbf{s}_k^A, \mathbf{s}_l^B)$ must be compatible (in $\mathcal{P}_s$, i.e. obeys (3)) and there must be no conflict with the existing matches or the geometrical model (1).
- If it exists, a pair incident with existing matches must be chosen first.

These rules are intended to explore the graphs by local growth, with previous matches quickly constraining future ones (see Fig. 3).

5.2. Algorithm

Each node ν of the search tree $\mathcal{T}$ represents one state (partial matching) and contains the criterion value Q_ν and (unlike

standard UCT [2]) Q_ν^+, which is the maximum reward Q_ν (4) for the subtree rooted in ν.

The algorithm repeatedly performs the following four steps until the computational budget is exhausted:

1. **Selection** - The *expandable* node with the highest urgency $\tilde{Q}_\nu$ is selected by a greedy top-down tree search, with

$$\tilde{Q}_\nu = \frac{Q_\nu^+}{Q_{\text{norm}}} + \gamma\sqrt{\frac{2\log n}{n_\nu}}. \quad (5)$$

   The second term is the *upper confidence bound* (UCB) [2] and balances between exploration of yet unvisited branches and exploitation of known good branches; γ is a user-defined parameter, Q_{norm} is the maximum possible Q_ν, n is the current iteration number and n_ν is the number of times the node ν has been visited.

2. **Expansion** - N_{exp} children of the most urgent node are added by performing valid moves. The priority is to add a superedge pair with the least number of edges. If equal, longer superedges are preferred.

3. **Simulation** - We estimate the reward Q_ν^+ of the most promising added node by greedily adding N_{sim} children in a depth-first search manner. The same heuristic is used in the selection of the superedge pair as in the Expansion.

4. **Backpropagation** - We update Q_ν^+ along the path from the current node back to the root.

The algorithm stops after reaching a predefined depth $\mathcal{D}$ or maximum iterations N. The result is a matching for a part of the vertices of the graphs, allowing coarse registration.

To align the structures locally and along the edges, we use a method [4] which solves an assignment problem constrained on the found matches.

6. EXPERIMENTS AND RESULTS

6.1. Experiments

The description of the datasets is shown together with the results in Table 1. In the *EM/LM* and *Brain vessels* datasets, two different modalities are used. In other datasets, the images to be registered are acquired at two different time points. If a graph has disconnected components, virtual edges are created between vertices within a short distance, which can only be matched with other virtual edges.

We test our approach against the graph matching using integer quadratic programming (PDQP) [3], Active Testing Search (ATS) [4], Coherent Point Drift (CPD) [9], Integer Projected Fixed Point (IPFP) [11] and a Path following algorithm (PATH) [12]. In all methods except of CPD, the same fine alignment algorithm [4] is used to align the structures based on their correspondence output. For each method, we set the parameters as suggested in the corresponding papers.

To test the robustness of the methods with respect to initial position, we randomized the initial position of the graphs by applying a rigid transformation with random rotation and translation to the graph $\mathbf{G}^A$. This procedure was repeated 50 times for each dataset. We present results for both with and without random initialization for CPD. All other methods are invariant to rigid transformations.

The parameters for the proposed method used throughout the experiments were $n_\omega = 5$, $|\mathbf{\Omega}| = 50$, $K = 3$, $N_{exp} = 2$, $N_{sim} = 25$, $\gamma = 0.01$ and $\kappa = 0.8$. The remaining parameters ε_h (ranging from 0.02 to 0.35) and termination criteria (typically $\mathcal{D} = 10$ and $N = 10000$) were set specifically for each dataset. The graphs were normalized such that $\mathbf{V}^A, \mathbf{V}^B \in [-1, 1]^D$.

6.2. Results

In Table 1, we show the geometrical error (the average distance between true correspondences of the two aligned structures), processing time and percentage of correct alignments (we consider an alignment incorrect if the error is above 0.15) after fine alignment. Local methods such as CPD work well when the initial position of the graphs is close to the true match. Otherwise, a global method must be used. The method proposed here, PDQP and ATS present the lowest errors of the global methods, but our method is faster than PDQP and ATS while returning more correspondences.

7. CONCLUSIONS

We presented a method which uses a Monte Carlo tree search strategy to quickly obtain a set of matches between two geometric graphs. Dissimilar superedges are pruned by path descriptors. We experimentally compared our approach to state of the art methods, showing that the proposed method can successfully register all tested datasets with the smallest error in spite of unknown initial orientation. It is faster than all other global methods tested.

8. REFERENCES

[1] E. Türetken, F. Benmansour, and P. Fua, "Automated reconstruction of tree structures using path classifiers and mixed integer programming," *IEEE ICCV*, pp. 566–573, 2012.

[2] C. Browne, E.J. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intellig. and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[3] M.A. Pinheiro and J. Kybic, "Path descriptors for geometric graph matching and registration," *International*

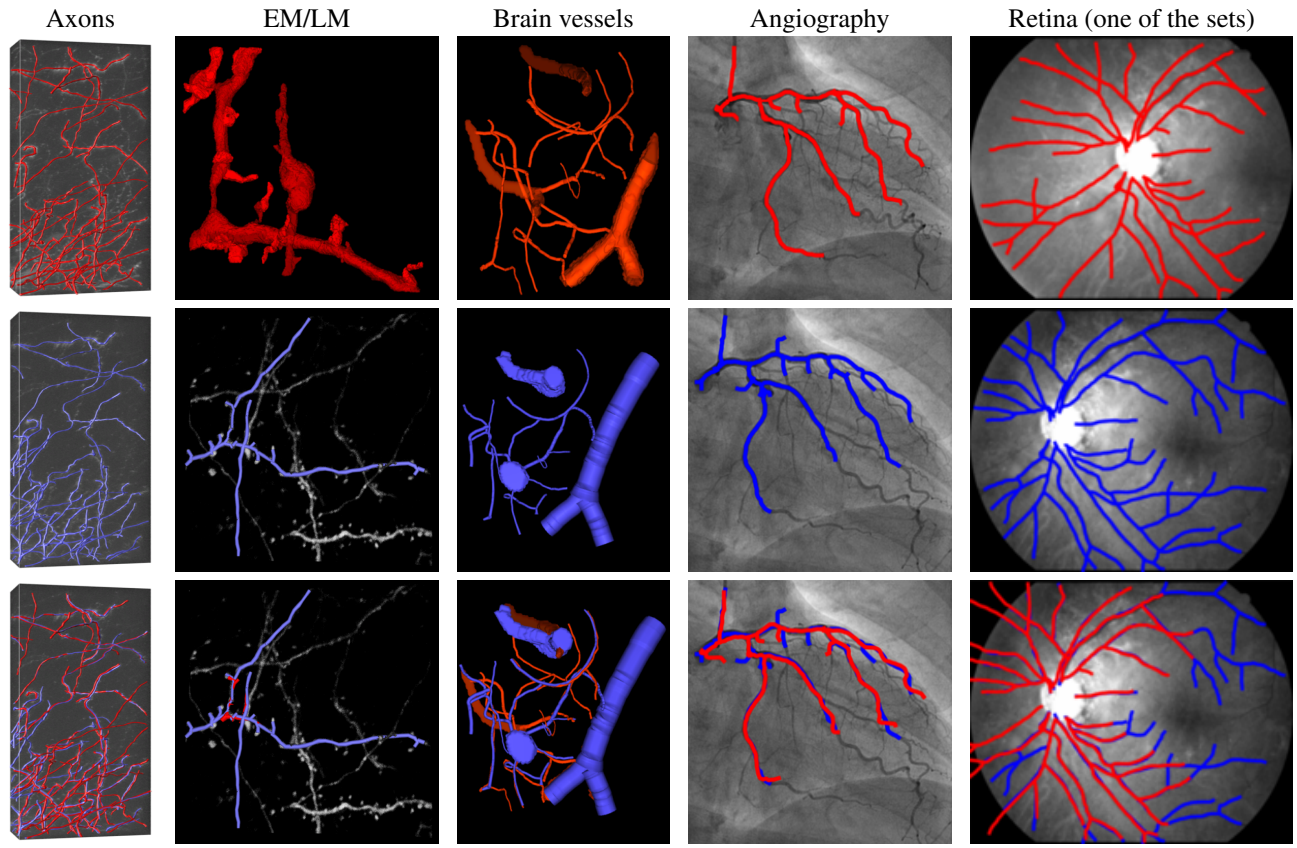

Fig. 4. Visualization of the datasets (first two rows) and result of aligning the structures with the proposed algorithm (bottom row). Best viewed in color.

Dataset		**Proposed** Random	**PDQP** [3] Random	**ATS** [4] Random	**CPD** [9]	**CPD** [9] Random	**IPFP** [11] Random	**PATH** [12] Random	**Description**
Axons	Error	**0.015**	0.081	0.090	**0.015**	0.833	0.713	0.958	2-photon microscopy volumes of an axon network [13]
1 set	t (s)	57.1	122.7	4414.4	16.5	42.9	93.1	25.0	
$\lvert\bar{\mathbf{V}}\rvert$: 124	$\mathcal{C}$ (%)	100	100	100	-	12	0	0	
EM/LM	Error	**0.016**	0.055	0.035	0.449	0.302	0.180	0.213	Brain microscopy images acquired using light and electron microscopy
1 set	t (s)	3.5	6.0	67.4	0.1	0.0	0.8	1.6	
$\lvert\bar{\mathbf{V}}\rvert$: 22	$\mathcal{C}$ (%)	100	100	100	-	0	0	0	
Brain vessels	Error	**0.028**	**0.028**	0.046	0.108	0.711	0.500	0.856	Brain blood vessels by optical and 2-photon microscopy
1 set	t (s)	7.7	28.1	644.1	1.4	2.7	3.0	1.1	
$\lvert\bar{\mathbf{V}}\rvert$: 28	$\mathcal{C}$ (%)	100	100	100	-	12	0	0	
Angiography	Error	**0.033**	0.084	0.036	0.065	0.729	0.113	0.552	2D x-ray heart angiograms
1 set	t (s)	1.1	2.4	392.0	0.8	0.9	1.3	1.5	
$\lvert\bar{\mathbf{V}}\rvert$: 22	$\mathcal{C}$ (%)	100	100	100	-	20	100	0	
Retina	Error	**0.004**	0.007	0.044	0.026	0.659	0.171	0.255	Retinal fundus images [14, 15]
14 sets	t (s)	62.3	6127.1	2226.8	10.0	28.2	562.1	27.1	
$\lvert\bar{\mathbf{V}}\rvert$: 142	$\mathcal{C}$ (%)	100	100	100	93	28	63	43	

Table 1. Average distance between true matches of aligned graphs (Error), processing time t in seconds and percentage of correct alignments $\mathcal{C}$ out of 50 pairs with random initial position for each experiment. The label *Random* identifies the experiments where the initial position of the structures were randomized. For each dataset, we present the number of sets in each of them and the average number of vertices $|\bar{\mathbf{V}}|$. The visualization of each dataset and alignment result for the proposed method is presented in Fig. 4. Graphs were normalized s. t. $\mathbf{V}^A, \mathbf{V}^B \in [-1, 1]^D$.

Conference on Image Analysis and Recognition, pp. 3–11, 2014.

[4] E. Serradell, M.A. Pinheiro, R. Sznitman, J. Kybic, F. Moreno-Noguer, and P. Fua, "Non-rigid graph registration using active testing search," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2014 (early access).

[5] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[6] O. Chum and J. Matas, "Matching with PROSAC – progressive sample consensus," *IEEE CVPR*, pp. 220–226, 2005.

[7] S. Choi, T. Kim, and W. Yu, "Performance Evaluation of RANSAC Family," *British Machine Vision Conference*, pp. 1–12, 2009.

[8] P.J. Besl and N.D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[9] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[10] M. Leordeanu and M. Hebert, "A Spectral Technique for Correspondence Problems Using Pairwise Constraints," *IEEE ICCV*, vol. 2, pp. 1482–1489, 2005.

[11] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and MAP inference," *NIPS*, pp. 1114–1122, 2009.

[12] M. Zaslavskiy, F. Bach, and J.P. Vert, "A path following algorithm for the graph matching problem.," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2227–2242, 2009.

[13] A. Holtmaat, J. Randall, and M. Cane, "Optical imaging of structural and functional synaptic plasticity in vivo," *European Journal of Pharmacology*, vol. 719, no. 1–3, pp. 128 – 136, 2013.

[14] K. Deng, J. Tian, J. Zheng, X. Zhang, X. Dai, and M. Xu, "Retinal fundus image registration via vascular structure graph matching," *Journal of Biomedical Imaging*, 2010.

[15] R. Kolář, V. Harabiš, and J. Odstrčilík, "Hybrid retinal image registration using phase correlation," *Imaging Science Journal*, vol. 61, no. 4, pp. 369–384, 2013.