Proceedings of the Workshop on Graphs in Biomedical Image Analysis
MICCAI-GRAIL 2017

Enzo Ferrante, Sarah Parisot, Aristeidis Sotiras
Imperial College London
University of Pennsylvania

14 September, 2017

Editorial

GRAIL 2017 is the first international workshop on GRaphs in biomedicAl Image anaLysis, organized as a satellite event of the 20th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2017) in Quebec, Canada. With this workshop we aim to highlight the potential of using graph-based models for biomedical image analysis. Our goal is to bring together scientists that use and develop graph-based models for the analysis of biomedical images and encourage the exploration of graph-based models for difficult clinical problems within a variety of biomedical imaging contexts.

Graph-based models have been developed for a wide variety of problems in computer vision and biomedical image analysis. Applications ranging from segmentation, registration, classification, and shape modeling, to population analysis have been successfully encoded through graph structures, demonstrating the versatile and principled nature of the graph based approaches. Graphs are powerful mathematical structures which provide a flexible and scalable framework to model objects and their interactions in a readily interpretable fashion. As a consequence, an important body of work has been developed around different methodological aspects of graph including, but not limited to, graphical models, graph-theoretical algorithms, spectral graph analysis, graph dimensionality reduction, and graph-based network analysis. However, new topics are also emerging as the outcome of interdisciplinary studies, shedding light on areas like deep structured models and signal processing on graphs.

The GRAIL proceedings contain 7 high quality papers of 8 to 11 pages that were pre-selected through a rigorous peer review process. All submissions were peer-reviewed through a double-blind process by at least 3 members of the program committee, comprising 18 experts in the field of graphs in biomedical image analysis. The accepted manuscripts cover a wide set of graph based medical image analysis methods and applications, including probabilistic graphical models, neuroimaging using graph representations, machine learning for diagnosis and disease prediction, and shape modeling.

The proceedings of the workshop are published as a joint LNCS volume alongside other satellite events organized in conjunction with MICCAI. In addition to the LNCS volume, to promote transparency, the papers' reviews are publicly available on the workshop website (https://biomedic.doc.ic.ac.uk/miccai17-grail/) alongside their corresponding optional response to reviewers.

We wish to thank all the GRAIL 2017 Authors for their participation and the members of the Program Committee for their detailed feedback and commitment to the workshop. We are very grateful to our sponsors CentraleSupelec and INRIA for their valuable support.

Enzo Ferrante, Sarah Parisot, Aristeidis Sotiras

Organization

Organizing Committee

- Enzo Ferrante, Imperial College London, UK
- Sarah Parisot, Imperial College London, UK
- Aristeidis Sotiras, University of Pennsylvania, USA

Scientific Committee

- Kayhan Batmanghelich, University of Pittsburgh / Carnegie Mellon University, US
- Michael Bronstein, University of Lugano / Tel Aviv University / Intel Perceptual Computing, Switzerland
- Eugene Belilovsky, INRIA / KU Leuven, France
- Christos Davatzikos, University of Pennsylvania, US
- Puneet K. Dokania, Oxford University, UK
- Ben Glocker, Imperial College London, UK
- Ali Gooya, University of Sheffield, UK
- Mattias Heinrich, University of Luebeck, Germany
- Dongjin Kwon, Stanford University, US
- Lisa Koch, ETH Zurich, Switzerland
- Sofia Ira Ktena, Imperial College London, UK
- Georg Langs, University of Vienna / MIT, Austria
- Jose Ignacio Orlando, Conicet / Unicen, Argentina
- Ipek Oguz, University of Pennsylvania, US
- Yangming Ou, Harvard University, US
- Nikos Paragios, CentraleSupelec / INRIA, France
- Mert Sabuncu, Cornell University, US
- Christian Wachinger, LMU München, Germany

Sponsors

Conference Program

Classifying phenotypes based on the community structure of human brain networks

Anvar Kurmukov[1], Marina Ananyeva[2], Yulia Dodonova[1], Boris Gutman[3], Joshua Faskowitz[3], Neda Jahanshad[3], Paul Thompson[3], and Leonid Zhukov[2]

[1] Kharkevich Institute for Information Transmission Problems, Moscow, Russia
[2] National Research University Higher School of Economics, Moscow, Russia
[3] Imaging Genetics Center, Stevens Neuroimaging and Informatics Institute, University of Southern California, Marina del Rey, USA

Abstract. Human anatomical brain networks derived from the analysis of neuroimaging data are known to demonstrate modular organization. Modules, or communities, of cortical brain regions capture information about the structure of connections in the entire network. Hence, anatomical changes in network connectivity (e.g., caused by a certain disease) should translate into changes in the community structure of brain regions. This means that essential structural differences between phenotypes (e.g., healthy and diseased) should be reflected in how brain networks cluster into communities. To test this hypothesis, we propose a pipeline to classify brain networks based on their underlying community structure. We consider network partitionings into both non-overlapping and overlapping communities and introduce a distance between connectomes based on whether or not they cluster into modules similarly. We next construct a classifier that uses partitioning-based kernels to predict a phenotype from brain networks. We demonstrate the performance of the proposed approach in a task of classifying structural connectomes of healthy subjects and those with mild cognitive impairment and Alzheimer‘s disease.

1 Introduction

Understanding disease-related changes in human brains has always been a challenge for neuroscience. A growing field of network science provides a powerful framework to study these changes [5]. This is because any shifts in brain anatomy or functioning are rarely confined to a single locus but rather affect the entire network system.

Human brain networks have been extensively studied in a recent decade. These networks, called connectomes, are constructed from neuroimaging data and represent either anatomical or functional connectivity between cortical brain regions. Several aspects of typical brain network organization have been described, including their modular structure. Modular structure of a network means that its nodes tend to group into modules, or communities, with close within-group connections and sparse between-group connectivity. Meunier et al. [10] discuss why it is reasonable for human brains to be modular, and also review studies on the community structure of human connectomes. Alexander-Bloch et al. [1] demonstrate that brain network community structure differs between phenotypes (healthy subjects and those with childhood-onset schizophrenia).

This suggests that brain network community structure captures enough information about network topology to classify phenotypes associated with certain diseases. To test this hypothesis, one needs a framework to classify networks based on similarity in their partitions into communities. Recently, Kurmukov et al. [8] proposed such an algorithm. Its basic idea was to detect non-overlapping brain network communities, measure pairwise distances between the obtained network partitions and use these distances in a kernel classification framework. However, [8] only considered non-overlapping brain network communities and demonstrated the performance of the proposed method on a small dataset.

Although non-overlapping communities are more commonly studied in network neuroscience, a model of community structure that allows for overlapping offers a more realistic model of brain-network organization [13]. Some cortical areas are known to be heteromodal and to have a role in multiple networks; consistently with this, current theories on brain organization suggest that cognitive functions are organized into widespread, segregated, and overlapping networks. Thus, clarifying the overlapping structure of brain network communities remains a challenging and relatively unexplored research area.

In this study, we generalize the classification approach [8] by considering both non-overlapping and overlapping communities of cortical brain regions. We show how both types of partitions may be used to estimate distances between brain networks and run a kernel classifier on these distances. Based on a large Alzheimer‘s Disease Neuroimaging Initiative dataset, we question whether similarity in brain modular structure can help to differentiate subjects with different diagnoses and tackle this question with the proposed approach.

2 Similarity of brain network community structures

Clustering networks into communities has attracted much attention in graph theory. Here, we only briefly describe the algorithms that we used for partitioning brain networks into communities (both non-overlapping and overlapping), and discuss how community structures of different brain networks may be quantitatively compared.

2.1 Detecting communities in structural brain networks

We use two approaches to detect brain network community structure. Both approaches aim to identify communities, or groups of tightly anatomically connected cortical regions. The major difference is that the first approach separates brain network regions into unique, non-overlapping modules, while the second algorithm allows for nodes belonging to more than one community. Algorithms of the former type are much more common in graph theory, and hence much more widely used in applications including brain network analysis [10]. However, as discussed above, overlapping community structures offer more powerful description of human brain organization, although they are much rarer evaluated [13].

In this study, we use the Louvain method [2] to produce non-overlapping partitions of structural connectomes. Given a graph $G(E,V)$ with a set of edges E, a set of nodes

V, and the adjacency matrix A, the algorithm divides nodes V into groups $\{V_1, V_2, ...V_k\}$ so that $V_1 \cup V_2 \cup ... \cup V_k = V$. Similarly to many other graph partitioning methods, it optimizes the so-called modularity by maximizing the number of intra-community connections and minimizing the number of inter-community links. The Louvain algorithm is a two-step iterative procedure. It starts with each node assigned to a separate cluster. In the first step, it moves each node i to a cluster of one of its neighbors j so that the gain in modularity is maximal. Once there is no such move that improves modularity, the algorithm proceeds to the second step, builds a new graph wherein nodes are clusters from the previous step, and reapplies the first step. Importantly, the Louvain method does not require any a-priori defined number of communities to be detected.

Second, we aim to estimate overlapping communities of structural brain networks. Two types of algorithms can accommodate this, differing in whether they use crisp or fuzzy assignment of nodes into communities. The former means that each node either belongs to each of the possible clusters or not, while the latter allows for a strength of belonging to a community. We detect fuzzy communities based on non-negative matrix factorization (NMF) [7]. Given a non-negative graph adjacency matrix A of size $n \times n$ (n being the number of nodes in brain network), we find its low-rank approximation

$$A \simeq WH, \tag{1}$$

where W is of size $n \times k$ and H is $k \times n$. A parameter k is usually selected to be much smaller than n and stands for a number of communities to be detected. Elements h_{ij} of a normalized matrix H denote probability of a node i being in a community j. Unlike the first method, the NMF algorithm requires specifying the number of communities. In our computational experiments, we show results obtained for different values of k.

2.2 Measuring distance between community structures

We aim to evaluate similarity in community structure of brain networks stemming from different subjects, possibly with different diagnoses. Hence, we need to introduce a measure of distance between two partitions obtained from different brain networks. This becomes possible because nodes in connectomes (i.e., cortical regions) are uniquely labeled, and the set of labels is the same across connectomes obtained with the same parcellation atlas.

To estimate pairwise similarity of partitions of different brain networks we use two modifications of mutual information (MI) score. Let $U = \{U_1, U_2, \cdots U_l\}$ and $V = \{V_1, V_2, \cdots V_k\}$ be partitions of two networks G_U and G_V with the same sets of node labels, l and k be the number of clusters in the partitions U and V, respectively. MI between the partitions U and V is defined by:

$$MI(U,V) = \sum_{i=1}^{l} \sum_{j=1}^{m} P(i,j) \log \frac{P(i,j)}{P(i)P'(j)}, \tag{2}$$

For brain network partitions into non-overlapping communities, we use adjusted mutual information, AMI [12]. We measure similarity between partitions into overlapping communities based on normalized mutual information (NMI, [9]). A property of the latter measure is that it only accepts partitions into overlapping modules with crisp node assignment. To accommodate this, we binarize the community membership matrix H (1) using a threshold parameter; we demonstrate how the results of our computational experiments change depending on this parameter.

Both measures take values in $[0,1]$, with the value of 1 indicating exactly the same partitions. We thus define a distance $\omega(G_U,G_V)$ between the community structures of networks G_U and G_V by:

$$\omega(G_U,G_V) = 1 - I(U,V), \tag{3}$$

where $I(U,V)$ is the index of similarity (AMI or NMI). Networks with the same community structure now have zero distance, and the maximum distance is close to 1.

3 Classifying connectomes based on their community structure

Since we obtained an optimal partition of each brain network into communities and introduced a measure of difference between community structures, we can proceed to the question of whether community structure of cortical brain regions provides enough information for differentiating between phenotypic classes. This question can be addressed in a machine learning framework.

Given a set of brain networks G_i (each with known community structure), class labels y_i, a training set of pairs (G_i,y_i) and the test set of input objects G_j, the task is to make a best possible prediction of the unknown class label y_j. Provided that we already defined a matrix of pairwise distances $\omega(G_U,G_V)$ (3), the most straightforward approach to classification is to convert the obtained distance matrix into a kernel and feed it to a kernel classifier. We accommodate this by exponentiating the obtained distances:

$$K(G_U,G_V) = e^{-\alpha\omega(G_U,G_V)}, \tag{4}$$

and run the support vector machines (SVM) classifier with the obtained kernel.

4 Experiments: network-based Alzheimer's disease classification

We argue that if the community structure of anatomical brain networks is affected by a disease in a certain manner, it should be possible to differentiate between healthy and diseased brain networks solely based on similarity in their community structures. In other words, brain networks stemming from the same class (e.g., obtained for healthy participants) should be more similar in their community structure than brain networks from different phenotypic classes (e.g., normal and diseased brains). Using the approach described in the previous sections, we test this hypothesis in a task of classifying Alzheimers disease (AD), late- and early-stage mild cognitive impairment (LMCI and EMCI), and healthy participants (normal controls, NC).

4.1 Data and network construction

We use the Alzheimer‘s Disease Neuroimaging Initiative (ADNI2) database which comprises a total of 228 individuals (756 scans), with a mean age at baseline visit 72.9±7.4 years, 96 females. Each individual has at least 1 brain scan and at most 6 scans. The data include 47 people with AD (136 AD scans), 40 individuals with LMCI (147 LMCI scans), 80 individuals with EMCI (283 EMCI scans), and 61 healthy participants (190 scans).

Corrected T1-weighted images were processed with Freesurfer‘s [4] recon-all pipeline to obtain a triangle mesh of the grey-white matter boundary registered to a shared spherical space, as well as corresponding vertex labels per subject. We used cortical parcellation based on the Desikan-Killiany (DK) atlas [3] which includes 68 cortical brain regions. T1w images were aligned (6-dof) to the 2mm isotropic MNI 152 template. These were used as the template to register the average b_0 of the DWI images, in order to account for EPI related susceptibility artifacts. DWI images were also corrected for eddy current and motion related distortions. Rotation of b-vectors was performed accordingly. Tractography for ADNI data was then conducted using the distortion corrected DWI in 2-mm isotropic MNI 152 space. Probabilistic streamline tractography was performed using the Dipy [6] LocalTracking module and implementation of constrained spherical deconvolution (CSD) [11] with a spherical harmonics order of 6. Streamlines longer than 5 mm with both ends intersecting the cortical surface were retained.

Edge weights in the original cortical connectivity matrices were proportional to the number of streamlines detected by the algorithm. We binarize these weights by:

$$a_{ij}^{\text{binarized}} = \begin{cases} 1 & \text{if} \quad a_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Thus, we only work with non-weighted graphs throughout the paper.

4.2 Experimental setup

We obtain the best partition of each network into non-overlapping communities using the Louvain algorithm and compute a matrix of pairwise distances between partitions with the AMI metric. In parallel, we cluster each network into overlapping communities based on NMF and produce a matrix of pairwise NMI distances between these clusterings. This second algorithm requires two parameters (the number of communities and the cluster membership threshold), we report how the results of the overall pipeline change depending on their particular values. For purposes of comparison, we also compute pairwise distances between connectomes using the L_2 (Frobenius) norm.

For each of the three distance matrices, we compute a kernel by (4) and run an SVM classifier with this kernel. We vary the values of α in (4) from 0.01 to 10 and the penalty parameter of the classifier from 0.1 to 50; we only report the results obtained for the optimal values of these technical parameters.

We consider four binary classification tasks: AD versus NC, AD versus LMCI, LMCI versus EMCI, EMCI versus NC. We find optimal values for all parameters in the simplest task of classifying AD versus NC and keep them fixed in the remaining tasks. We use 10-fold cross-validation to train SVM on a subsample and make predictions for an unseen part of a sample. As the data include several networks for each subject, we use subjects rather than networks to split data into train and test and put all networks of the same subject into a respective category (thus avoiding data leakage).

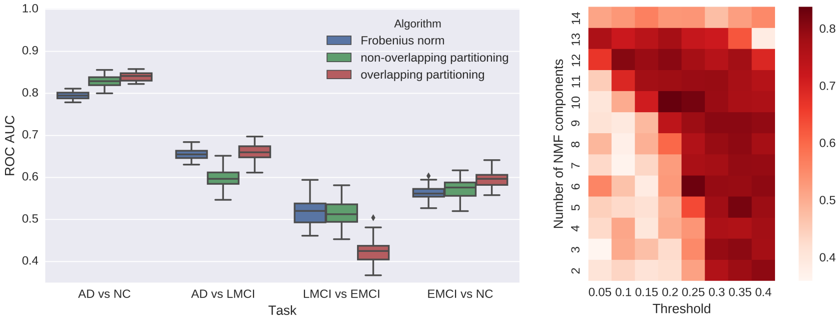


Fig. 1. Left: Classification results. Right: Results of classifying AD versus NC based on the overlapping community detection algorithm, depending on the number of components and the membership threshold; colorbar shows average ROC AUC values.

We train the models on networks and next make a subject-based prediction as an average of predictions obtained for individual networks; this method of evaluation (subject-based rather than network-based) does not affect the reported results in any systematic way. We repeat the procedure 50 times with different data splits and report ROC AUC as a quality metric. All scripts are available at `https://github.com/kurmukovai/GRAIL2017-communities`.

4.3 Results and discussion

Figure 1 (left) shows the results of classifying AD, LMCI, EMCI and healthy controls based on $L2$-distance between the structural connectivity matrices of brain networks and on the distances representing similarity in brain community structures.

As expected, classifying AD versus NC was the simplest task, while for EMCI versus LMCI all algorithms only performed at chance level. For the tasks with reasonable overall classification quality, an algorithm based on overlapping community structures slightly outperformed the other algorithms. For AD versus NC, the model with overlapping communities provides an ROC AUC of 0.840 ± 0.010; the one based on non-overlapping communities gives an ROC AUC 0.828 ± 0.013. For this task, Figure 1 (right) shows how the outcomes of the best-performing algorithm depend on the predefined number of clusters and the threshold of cluster membership used in computing the NMI distance. The best classification results are obtained with the community structure of six overlapping components, with membership probability thresholded at 0.25.

Figure 2 illustrates the obtained community structure based on a single example graph. Figure 3 compares the non-overlapping and the simplified overlapping commu-

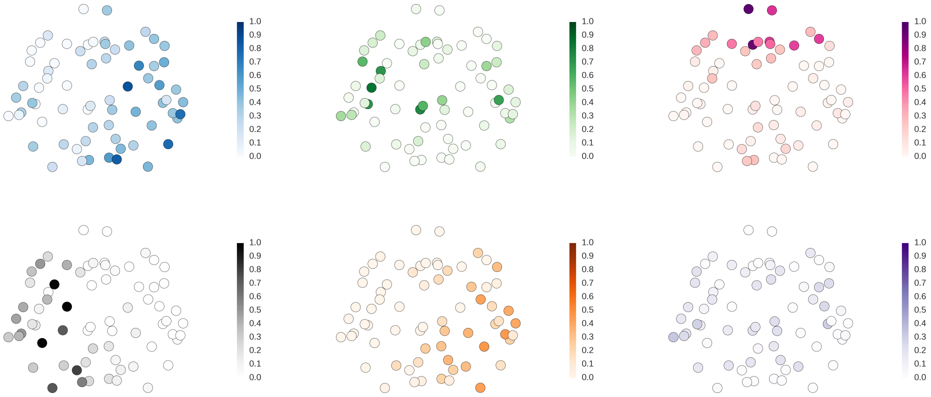


Fig. 2. Six overlapping communities: an example of a single network (healthy subject) with the nodes shown in their original 3D coordinates (axial view); color intensity is proportional to the strength of belonging to the respective community

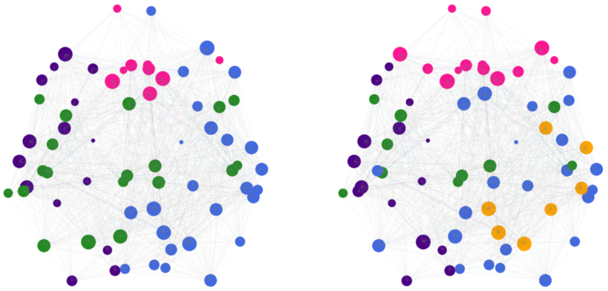


Fig. 3. Comparison of the non-overlapping (left) and overlapping (right) community structures obtained for the same example graph as in Figure 3; node size is proportional to its degree (the number of edges coming from the respective node). Right plot is produced by selecting a single community for each node based on the maximal membership probability.

nity structures obtained for the same graph. The two algorithms seem to identify similar communities, but the outcome of the overlapping community detection algorithm retains more information on the underlying brain network structure.

5 Conclusions

Human brain networks show modular structure which arises based on the entire system of connections between cortical brain regions. Systematic shifts in connectivity patterns, for example those caused by a brain disease, may be expected to induce changes in the community structure of the macroscale brain networks. If true, that would produce similar modular structure in brain networks of individuals with the same phenotype (e.g., Alzheimer‘s disease) and different community structures in brain networks from different phenotypes (e.g., patients versus healthy controls).

In this study, we explored whether the community structure of anatomical human brain networks provides enough information to differentiate phenotypes of the respective individuals. We proposed a framework to compare both overlapping and non-overlapping community structures of brain networks within the machine learning settings. We demonstrated the performance of the proposed pipeline in a task of classifying Alzheimer‘s disease, mild cognitive impairment, and healthy participants. Algorithms based on the distances between partitions of brain networks slightly outperformed the baseline. Models that made full use of overlapping community structures performed slightly better than those based on non-overlapping community structures.

To sum up, the modular structure of anatomical brain networks seems to capture important information about the underlying network structure and can be useful in classifying phenotypes. Further studies are needed to study this idea on other phenotypic categories, and to specifically explore overlapping community structure of cortical regions in human anatomical brain networks.

Acknowledgments

The data used in preparing this paper were obtained from the Alzheimer‘s Disease Neuroimaging Initiative (ADNI) database. A complete listing of ADNI investigators and imaging protocols may be found at `adni.loni.usc.edu`.

The results of sections 2-5 are based on the scientific research conducted at IITP RAS and supported by the Russian Science Foundation under grant 17-11-01390.

References

1. Alexander-Bloch, A.F., Gogtay, N., Meunier, D., Birn, R., et al.: Disrupted modularity and local connectivity of brain functional networks in childhood-onset schizophrenia. Frontiers in Systems Neuroscience, 4 (2010)
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, (2008)

3. Desikan, R.S., Ségonne, F., Fischl, B., Quinn, B.T., et al.: An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. Neuroimage 31(3), 968–980 (2006)
4. Fischl, B.: Freesurfer. Neuroimage 62(2), 774–781 (2012)
5. Fornito, A., Zalesky, A., Breakspear, M.: The connectomics of brain disorders. Nature Reviews. Neuroscience, 16, 159172 (2015)
6. Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., et al.: Dipy, a library for the analysis of diffusion mri data. Frontiers in neuroinformatics, 8, 8 (2014)
7. Kuang, D., Ding, C., Park, H.: Symmetric nonnegative matrix factorization for graph clustering. The 12th SIAM International Conference on Data Mining, pp. 106–117 (2012)
8. Kurmukov, A., Dodonova, Y., Zhukov, L.E.: Classification of normal and pathological brain networks based on similarity in graph partitions. Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference, pp. 107–112 (2016)
9. McDaid, A.F., Greene, D., Hurley, N.: Normalized mutual information to evaluate overlapping community finding algorithms (2011)
10. Meunier, D., Lambiotte, R., Bullmore, E.T.: Modular and hierarchically modular organization of brain networks. Frontiers of Neuroinformatics, 4 (2010)
11. Tax, C.M., Jeurissen, B., Vos, S.B., Viergever, M.A., Leemans, A.: Recursive calibration of the fiber response function for spherical deconvolution of diffusion mri data. Neuroimage 86, 67–80 (2014)
12. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research, pp. 2837–2854 (2010)
13. Wu, K., Taki, Y., Sato, K., et al.: The overlapping community structure of structural brain network in young healthy individuals. PLoS One, 6 (2011)

Autism Spectrum Disorder Diagnosis Using Sparse Graph Embedding of Morphological Brain Networks

Carrie Morris and Islem Rekik*

BASIRA lab, CVIP group, School of Science and Engineering, Computing, University of Dundee, UK

Abstract. Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder involving a complex cognitive impairment that can be difficult to diagnose early enough. Much work has therefore been done investigating the use of machine-learning techniques on functional and structural connectivity networks for ASD diagnosis. However, networks based on the morphology of the brain have yet to be similarly investigated, despite research findings that morphological features, such as cortical thickness, are affected by ASD. In this paper, we first propose modelling morphological brain connectivity (or graph) using a set of cortical attributes, each encoding a unique aspect of cortical morphology. However, it can be difficult to capture for each subject the complex pattern of relationships between morphological brain graphs, where each may be affected simultaneously or independently by ASD. In order to solve this problem, we therefore also propose the use of high-order networks which can better capture these relationships. Further, since ASD and normal control (NC) high-dimensional connectomic data might lie in different manifolds, we aim to find a low-dimensional representation of the data which captures the intrinsic dimensions of the underlying connectomic manifolds, thereby allowing better learning by linear classifiers. Hence, we propose the use of sparse graph embedding (SGE) method, which allows us to distinguish between data points drawn from different manifolds, even when they are too close to one another. SGE learns a similarity matrix of the connectomic data graph, which then is used to embed the high-dimensional connectomic features into a low-dimensional space that preserves the locality of the original data. Our ASD/NC classification results outperformed several state-of-the-art methods including statistical feature selection, and local linear embedding methods.

1 Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder characterized by varied impairments in cognitive function, including difficulties with social communication and interaction, language, and restricted, repetitive behaviours. This has made diagnosing the disorder a challenging task [1]. However, aided by

* Corresponding author: irekik@dundee.ac.uk, www.basira-lab.com

recent technological and methodological advances in neuroimaging tools, there has been growing interest in understanding how ASD can alter the connectivity between different regions within the brain, and how this information may be leveraged to help diagnose the disorder with greater accuracy [2]. The two most widely used measures of brain connectivity used for investigating ASD in the literature are functional connectivity and structural connectivity, derived from functional magnetic resonance imaging (fMRI) and diffusion tensor imaging (DTI) respectively, with literature reviews available for both types of data [3, 4]. For example, all 77 studies discussed in [5]'s review of using machine learning on connectomes (networks of the brain) to predict clinical outcomes used functional and/or structural connectivity networks to do so. Despite this growing body of research on such networks, however, there is still a gap in the literature where morphological networks have not been explored to the same degree. This gap needs to be filled, considering there are studies that indicate morphological features of the brain, such as cortical thickness, can be affected in neurological disorders, including ASD [6, 7]. As such, the use of networks based on morphological data in neurological disorder diagnosis, using machine learning, could prove fruitful. Further, such networks have not been used to investigate ASD in the literature so far. In this study, we will therefore aim to define several networks based on the morphology or geometry of the cortical surface of ASD and NC subjects, and investigate their use in diagnosing ASD using machine learning techniques.

Different morphological views of the cortical surface (e.g. cortical thickness and mean curvature) may also have different relationships between them, where they could be affected simultaneously or independently in different regions of the brain by ASD. As a result, there could be a very complex pattern of how ASD affects the different morphological views of the brain. The easiest and most commonly employed method for exploring such relationships is simply to concatenate the multiple networks together so that the data from each view is included in the overall set of data for each subject, unaltered [5]. However, recent research on Alzheimer's Disease has found better results when using High Order Networks (HONs) [8]. These are constructed from low-order (e.g. functional connectivity) networks by, for each view or network, extracting the correlations between different pairs of brain regions, then calculating the correlation between these values across all views, for each pair of brain regions. This method therefore better allows us to capture the higher-order relationships between different views of the brain. However, it has yet to be applied to ASD data in machine learning research, and so, with this study, we also aim to contribute to the literature on the use of such HONs when investigating ASD. One potential problem with the use of HONs, however, is that the networks produced are very large and, as a result, computationally expensive. To mitigate this, feature selection or dimensionality reduction is necessary. Noting that ASD morphological changes between brain regions might be very subtle, the manifolds where both ASD and healthy connectomic data lie might be very close and challenging to embed into a low-dimensional space. To address this problem, we further propose a classification framework based on a *sparse graph embedding of connectomic data* using the method developed in [9]. Specifically, we use graph embedding of the HONs, which would allow us to (1) explore the high-order relationships without having to deal with overly large networks, (2) learn the features that are most discriminative in classifying and diagnosing ASD, and (3) investigate the effectiveness of SGE as a dimensionality reduction technique on our data, as compared to other state-of-the-art methods.

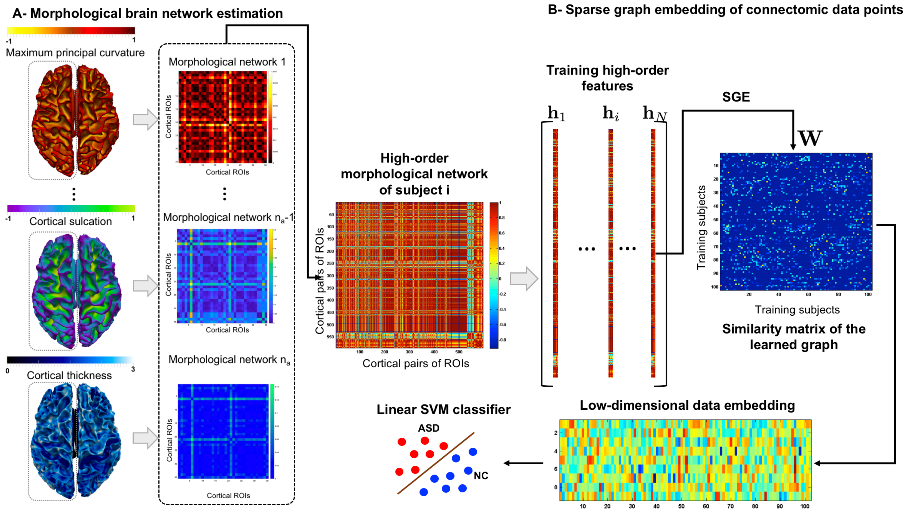

A- Morphological brain network estimation
-1
1
Maximum principal curvature
Morphological network 1
Cortical ROIs
Cortical ROIs
Cortical sulcation
Morphological network n_a-1
Cortical thickness
0
3
Morphological network n_a
High-order morphological network of subject i
Cortical pairs of ROIs
Cortical pairs of ROIs
B- Sparse graph embedding of connectomic data points
Training high-order features
$\mathbf{h}_1$
$\mathbf{h}_i$
$\mathbf{h}_N$
SGE
$\mathbf{W}$
Training subjects
Training subjects
Similarity matrix of the learned graph
Low-dimensional data embedding
Linear SVM classifier
ASD
NC

Fig. 1: *High-order sparse graph embedding (SGE) of high-order brain networks for classifying autism spectrum disorder (ASD) and healthy brains.* (A) For each subject i, we construct n_a low-order morphological networks for each cortical attribute. Then, we merge these into a high-order network represented by a feature vector $\mathbf{h}_i$. (B) Given the high-order feature matrix of all subjects, we use sparse graph embedding [9] to learn a sparse similarity matrix $\mathbf{W}$ of a graph $\mathcal{G}$ which models the relationship between data points lying in different connectomic manifolds. Next, we embed the graph into a low-dimensional space where a linear SVM classifier is trained.

2 Proposed Sparse Graph Embedding of High-Order Morphological Brain Networks for Autism classification

In this section, we present our sparse graph embedding (SGE) of high-order morphological brain networks for ASD classification using the SMCE method proposed in [9]. We denote matrices by boldface capital letters, e.g., $\mathbf{X}$, and scalars are denoted by lowercase letters, e.g., x. For easy reference, we have summarized the major mathematical notations in **Table** 1. **Fig.** 1 illustrates the proposed pipeline for ASD/NC classification in four major steps: (1) construction

Table 1: *Major mathematical notations used in this paper.*

Mathematical notation	Definition
$\mathbf{C}_i^a = (V_C, E_C)$	low-order brain network graph $\mathbb{R}^{n_r \times n_r}$ of a single subject i for cortical attribute a
V_C	nodes or brain ROIs of size n_r
E_C	edges connecting pairs of brain ROIs in a single subject
n_a	number of cortical attributes
$\mathbf{H}_i = (V_H, E_H)$	high-order brain network graph of a single subject i
V_H	a node represents a pair of brain ROIs
E_H	edges connecting two pairs of brain ROIs in a single subject
$\mathbf{h}_i$	high-order connectomic feature vector $\in \mathbb{R}^{\mathbb{D}}$ of subject i derived from brain graph $\mathbf{H}_i$
N	number of training subjects
K	number of manifolds
$\mathcal{M}_l$	manifold where similar connectomic data points lie
d_l	intrinsic dimension of manifold $\mathcal{M}_l$
$\mathcal{G} = (V_{\mathcal{M}}, E_{\mathcal{M}})$	similarity graph of connectomic data points nested in different manifolds $\{\mathcal{M}_l\}_{l=1}^{K}$
$\check{\mathbf{D}}_i$	normalized distance matrix in $\mathbb{R}^{D \times N-1}$ between current data point $\mathbf{h}_i$ and other data points
$\mathbf{Q}_i$	positive-definite diagonal proximity inducing matrix
α_i	a sparse vector whose $d_l + 1$ nonzero elements correspond to $d_l + 1$ neighbours of $\mathbf{h}_i \in \mathcal{M}_l$
$\mathbf{w}_i$	weight vector in $\mathbb{R}^N$ associated with the i-th point
$\mathbf{W}$	similarity matrix in $\mathbb{R}^{N \times N}$ of graph $\mathcal{G}$

of low-order morphological networks, 2) construction of high-order morphological network, 3) connectomic feature extraction, and 4) sparse graph learning and embedding to reduce the dimension of the extracted features for our target classification task.

Low-order morphological network construction. For each subject i and each cortical attribute a (e.g., cortical thickness), we build a brain graph $\mathbf{C}_i^a = (V_C, E_C)$, where each node in V_C represents a cortical region of interest (ROI), and each edge in E_C connecting two ROIs R_p and R_q is defined as $\mathbf{C}_i(p,q) = |\hat{x}_p - \hat{x}_q|$, where $\hat{x}_p$ denotes the average of the cortical attribute across all vertices in region R_p. Given n_a cortical attributes, we generate for each cortical hemisphere n_a morphological brain graphs $\{\mathbf{C}_a\}_{a=1}^{n_a}$.

High-order morphological network construction (HON). We note that ASD might affect not only region-to-region morphological brain connections on a low-order level, but also high-order relationships between pairs of ROIs, where complex interactions between sets of ROIs might be affected. Hence, we propose constructing a high-order morphological network to integrate into a single, larger brain graph $\mathbf{H}_i = (V_H, E_H)$ all low-order brain graphs $\{\mathbf{C}_a\}_{a=1}^{n_a}$ of *both* hemispheres. Each node in V_H denotes a *pair* of ROIs and each edge in E_H connecting two pairs or ROIs (p,q) and (p',q') denotes the Pearson Correlation coefficient between vectors $\mathbf{y_{pq}}$ and $\mathbf{y_{p'q'}}$, where $\mathbf{y_{pq}}$ corresponds to the connectivity strength between the p-th and q-th ROIs across all $2n_a$ brain networks in both hemispheres.

Feature Extraction. We propose two types of features: high-order features (HON), and concatenated low-order features (CON). Noting that all brain graphs are symmetric, for each subject i, we represent its high-order brain graph as a matrix $\mathbf{H}_i$, then concatenate its upper triangle elements into a long feature vector $\mathbf{h}_i$. The weights on the diagonal are set to zero to avoid self-connectedness. For low-order brain graphs $\{\mathbf{C}_a\}_{a=1}^{n_a}$, we simply concatenate the upper triangle elements across all cortical attributes into a feature vector (termed as CON). To address the issue of 'high-dimensional features vs.a low sample size' in classification, we propose embedding our high-dimensional connectomic features into a low-dimensional space where we can efficiently train a linear classifier through learning a sparse graph.

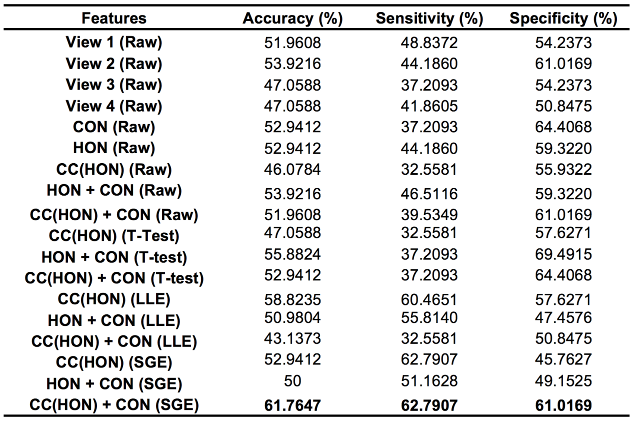
Sparse graph embedding (SGE) using connectomic brain features for ASD classification. Since ASD and NC high-dimensional connectomic data might lie in different manifolds, we aim to find a low-dimensional representation of the data which captures the intrinsic dimensions of the underlying connectomic manifolds, thereby allowing better learning by classifiers. However, since morphological brain changes can be very subtle in autistic subjects compared with healthy brains, their data manifolds can be very close to each other. Hence, estimating a low-dimensional embedding that allows us to distinguish between data points drawn from different manifolds is challenging. To solve this problem, Elhamifar *et al.* proposed a robust algorithm for sparse manifold clustering and embedding (SMCE) that efficiently handles multiple manifolds that are very close to each other [9]. This is achieved through encouraging a sparse selection of nearby connectomic points that lie in the same manifold and spanning a low-dimensional affine subspace. Unlike typical dimensionality reduction methods such as local linear embedding (LLE), which builds a neighbourhood graph by connecting each data point to a *fixed* number of nearest points, SMCE learns a graph neighbourhood automatically, thereby allowing the neighbourhood size on the manifold to vary. This better handles variation in the density of data points on the manifold.

Leveraging the strengths of the SMCE method, we then propose our sparse graph embedding (SGE) framework for the low-dimensional representation of the high-order connectomic brain manifolds of ASD and NC subjects (**Fig.** 1). Given N training high-order feature vectors $\{\mathbf{h}_i \in \mathbb{R}^D\}_{i=1}^N$ lying in K different manifolds $\{\mathcal{M}_{l=1}^K\}$ of intrinsic dimensions $\{d_l\}_{l=1}^K$, we build a similarity graph $\mathcal{G} = (V_\mathcal{M}, E_\mathcal{M})$, where each node in $V_\mathcal{M}$ represents a feature vector $\mathbf{h}$ derived from a brain graph $\mathbf{H}$. Our goal is to learn sparse connections in graph $\mathcal{G}$ through connecting each point to a few neighbouring points with appropriate weights such that the selected neighbouring points are from the same manifold. This is achieved through solving a sparse optimization function that selects for each connectomic point $\mathbf{h}_i \in \mathcal{M}_l$ *a few* neighbouring points that span a low-dimensional affine subspace passing near $\mathbf{h}_i$:

$$\min_{\alpha_i} \lambda ||\mathbf{Q}_i \alpha_i||_1 + \frac{1}{2}||\check{\mathbf{D}}_i \alpha_i||_2^2 \ s.t. \ \mathbf{1}^T \alpha_i = 1, \tag{1}$$

where $\alpha_i^T \triangleq [\alpha_{i1} \dots \alpha_{iN}]$ denotes a solution whose d_l+1 nonzero elements correspond to $d_l + 1$ neighbours of $\mathbf{h}_i \in \mathcal{M}_l$. $\check{\mathbf{D}}_i$ represents the normalized distance matrix between current data point $\mathbf{h}_i$ and other points: $\check{\mathbf{D}}_i \triangleq [\frac{\mathbf{h}_1 - \mathbf{h}_i}{||\mathbf{h}_1 - \mathbf{h}_i||_2} \dots \frac{\mathbf{h}_N - \mathbf{h}_i}{||\mathbf{h}_N - \mathbf{h}_i||_2}] \in \mathbb{R}^{D \times N-1}$. L_1 sparsity penalty constrains points closer to $\mathbf{h}_i$ to be less penalised than points that are further away. $\mathbf{Q}_i$ is a proximity inducing positive-definite diagonal matrix, which favours the selection of close points to the current point $\mathbf{h}_i$ through assigning smaller weights to them. We define its diagonal elements as

Table 1. *ASD/NC classification results using our method and different comparison methods.*



Features	Accuracy (%)	Sensitivity (%)	Specificity (%)
View 1 (Raw)	51.9608	48.8372	54.2373
View 2 (Raw)	53.9216	44.1860	61.0169
View 3 (Raw)	47.0588	37.2093	54.2373
View 4 (Raw)	47.0588	41.8605	50.8475
CON (Raw)	52.9412	37.2093	64.4068
HON (Raw)	52.9412	44.1860	59.3220
CC(HON) (Raw)	46.0784	32.5581	55.9322
HON + CON (Raw)	53.9216	46.5116	59.3220
CC(HON) + CON (Raw)	51.9608	39.5349	61.0169
CC(HON) (T-Test)	47.0588	32.5581	57.6271
HON + CON (T-test)	55.8824	37.2093	69.4915
CC(HON) + CON (T-test)	52.9412	37.2093	64.4068
CC(HON) (LLE)	58.8235	60.4651	57.6271
HON + CON (LLE)	50.9804	55.8140	47.4576
CC(HON) + CON (LLE)	43.1373	32.5581	50.8475
CC(HON) (SGE)	52.9412	62.7907	45.7627
HON + CON (SGE)	50	51.1628	49.1525
CC(HON) + CON (SGE)	**61.7647**	**62.7907**	**61.0169**

$\frac{||\mathbf{h}_j-\mathbf{h}_i||_2}{\sum_{t\neq i}||\mathbf{h}_t-\mathbf{h}_i||_2} \in (0,1])$. The trade-off parameter λ balances the sparsity solution (first term) and the affine reconstruction error (second term).

After solving **Eq.** 1, we define a weight vector $\mathbf{w}_i^T = [w_{i1} \dots w_{iN}] \in \mathbb{R}^N$ associated with the i-th point as: $w_{ii} = 0$ and $w_{ij} \triangleq \frac{\alpha_{ij}/||\mathbf{h}_j-\mathbf{h}_i||_2}{\sum_{t\neq i}\alpha_{it}/||\mathbf{h}_t-\mathbf{h}_i||_2}$, $j \neq i$. Ideally, non-zero elements of $\mathbf{w}_i$ will correspond to sparse neighbours of $\mathbf{h}_i$ which belong to the same manifold. Next, we use these weights to define edges in the similarity graph $\mathcal{G}$ where a node $\mathbf{h}_i$ connects to node $\mathbf{h}_j$ with weight $|w_ij|$. Ideally, points in the same manifold will belong to the same connected component in the learned graph $\mathcal{G}$. Ultimately, we define the similarity matrix $\mathbf{W} \triangleq [|\mathbf{w}_1| \dots |\mathbf{w}_N|]$ in $\mathbb{R}^{N\times N}$ of the manifold graph $\mathcal{G}$, which groups points from the same manifold into a block-by-block matrix structure. We then generate the local embedding of the connectomic features by taking the last eigenvectors of the normalized Laplacian matrix associated with each cluster in $\mathbf{W}$. In the training stage, we learn $\mathbf{W}_{tr}$ for all training subjects. Then we use the produced low-dimensional features to train a linear support vector machine (SVM) classifier. In the testing stage, we map the testing subject to a low-dimensional space (with same dimension) through estimating a new $\mathbf{W}_{ts}$ that includes training and testing samples.

3 Results and Discussion

Evaluation dataset and method parameters. We used leave-one-out cross validation to evaluate the proposed classification framework on 102 subjects (59 ASD and 43 NC) from Autism Brain Imaging Data Exchange (ABIDE I)[1] public dataset, each with structural T1-w MR image [10]. We used FREESURFER to

[1] http://fcon_1000.projects.nitrc.org/indi/abide/

reconstruct both right and left cortical hemispheres for each subject from T1-w MRI. Then we parcellated each cortical hemisphere into 35 cortical regions using Desikan-Killiany Atlas. For each subject, we generated $n_a = 4$ cortical morphological networks: $\mathbf{C}^1$ denotes the maximum principal curvature brain view, $\mathbf{C}^2$ denotes the mean cortical thickness brain view, $\mathbf{C}^3$ denotes the mean sulcal depth brain view, and $\mathbf{C}^4$ denotes the mean of average curvature. For SGE parameters, we set $\lambda = 10$. For both LLE and SGE, we used nested grid-search to estimate the low dimension of the feature embedding (9 for SGE and 50 for LLE).

Method evaluation and comparison methods. We compared our method with three state-of-the-art methods: (RAW) where we directly input the raw connectomic brain features, (t-test) where we perform dimensionality reduction using statistical feature selection, and (LLE) where we perform a local linear embedding of the connectomic features to produce a compact and low-dimensional representation of feature vectors. Since both CON and HON feature vectors are high-dimensional, we propose a preliminary dimensionality reduction step through representing each network by a clustering coefficients (CC) feature vector. This will allow us to benchmark our method against the recent connectomic classification framework proposed in [8] where they first concatenated the clustering coefficients of the functional HON (CC(HON)) and CON features (i.e., CC(HON) + CON), then performed t-test for feature selection to train an SVM classifier for Alzheimer's disease diagnosis. We further evaluated all methods using combinations of different feature types: (1) HON, (2) CON, (3) CC(HON), (4) HON + CON, and (5) CC(HON) + CON. All results are presented in **Table** 1 and **Fig.** 2. Our method produced the best ASD/NC classification accuracy (61.76%) when using (CC(HON) + CON) features, which largely outperformed t-test using (CC(HON) + CON) as in [8].

4 Conclusion

We proposed a sparse graph learning framework for classifying disordered brain connectivities based on the morphology of cortical hemispheres. Specifically, we estimated a local embedding of high-order and low-order morphological brain networks for distinguishing between autistic and healthy brains. Given that morphological brain changes are subtle in ASD patients, our results are promising. Instead of performing the local embedding of data points for each feature type independently, we will further extend our method to jointly embed different feature types nested in multiple views of the same manifold (e.g., ASD data manifold).

References

1. Lord, C., Cook, E.H., Leventhal, B.L., Amaral, D.G.: Autism spectrum disorders. Neuron **28** (2000) 355 – 363

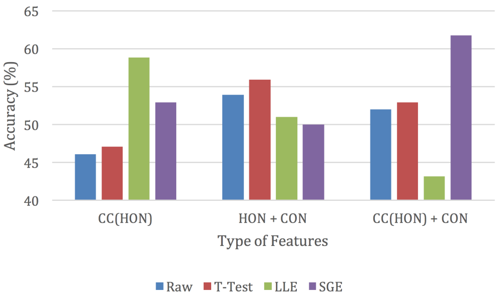


Fig. 2: *ASD/NC classification accuracies for our method (SGE) and other comparison methods using combinations of different connectomic feature types.* We compared our method with three state-of-the-art methods: (RAW) where we directly input the raw connectomic brain features, (t-test) where we perform dimensionality reduction using statistical feature selection, and (LLE) where we perform a local linear embedding of the connectomic features to produce a compact and low-dimensional representation of feature vectors. Our method produced the best ASD/NC classification accuracy when using (CC(HON) + CON) features, which significantly outperformed t-test using (CC(HON) + CON) as in [8].

2. Anagnostou, E., Taylor, M.J.: Review of neuroimaging in autism spectrum disorders: what have we learned and where we go from here. Molecular Autism **2** (2011) 4
3. Philip, R.C., Dauvermann, M.R., Whalley, H.C., Baynham, K., Lawrie, S.M., Stanfield, A.C.: A systematic review and meta-analysis of the fmri investigation of autism spectrum disorders. Neuroscience Biobehavioral Reviews **36** (2012) 901 – 942
4. Stanfield, A.C., McIntosh, A.M., Spencer, M.D., Philip, R., Gaur, S., Lawrie, S.M.: Towards a neuroanatomy of autism: A systematic review and meta-analysis of structural magnetic resonance imaging studies. European Psychiatry **23** (2008) 289 – 299 Neuroimaging.
5. Brown, C., Hamarneh, G.: Machine learning on human connectome data from MRI. arXiv:1611.08699v1 (2016)
6. Cauda, F., Costa, T., Nani, A., Fava, L., Palermo, S., Bianco, F., Duca, S., Tatu, K., Keller, R.: Are schizophrenia, autistic, and obsessive spectrum disorders dissociable on the basis of neuroimaging morphological findings?: A voxel-based meta-analysis. Autism Research (2017)
7. Khundrakpam, B.S., Lewis, J.D., Kostopoulos, P., Carbonell, F., Evans, A.C.: Cortical thickness abnormalities in autism spectrum disorders through late childhood, adolescence, and adulthood: A large-scale mri study. Cerebral Cortex **27** (2017) 1721
8. Chen, X., Zhang, H., Gao, Y., Wee, C.Y., Li, G., Shen, D., the Alzheimer's Disease Neuroimaging Initiative: High-order resting-state functional connectivity network for mci classification. Human Brain Mapping **37** (2016) 3282–3296
9. Elhamifar, E., Vidal, R.: Sparse manifold clustering and embedding. (2011) 55–63
10. Mueller, S.G., Weiner, M.W., Thal, L.J., Petersen, R.C., Jack, C., Jagust, W., Trojanowski, J.Q., Toga, A.W., Beckett, L.: The Alzheimer's Disease Neuroimaging Initiative. Neuroimaging Clinics of North America **10** (2005) 869–877

Topology of surface displacement shape feature in subcortical structures

Amanmeet Garg, Donghuan Lu, Karteek Popuri, and Mirza Faisal Beg

Simon Fraser University, 8888 University Drive, Burnaby, Canada

Abstract. The shape of anatomical structures in the brain has been adversely influenced by neurodegenerative disorders. However, the shape feature covariation between regions (e.g. subfields) of the structure and its change with disease remains unclear. In this paper, we present a first work to study the topology of the *surface displacement* shape feature via its persistence homology timeline features and model the polyadic interactions between the shape across the subfields of subcortical structures. Specifically, we study the caudate and pallidum structures for *Shape Topology* change with Parkinson's disease. The shape topology features show statistically significant group level difference and good prediction performance in a repeated hold out stratified training experiment. These features show promise in their potential application to other neurological conditions and in clinical settings with further testing on larger data cohorts.

Keywords: Topological data analysis, shape topology, Parkinson's disease, caudate, pallidum

1 Introduction

Neurodegeneration in Parkinson's disease (PD) has shown morphology change in subcortical (deep gray matter, substantia nigral area) structures where shape [4, 2] of the structures was found to be adversely affected in Parkinson's patients. Additionally, topology of functional and structural networks has shown strong differences in Parkinson's patients in comparison to healthy individuals. However, little is known for the covariation of shape across the subfields of anatomical structures in the brain. This highlights the need to study the topology of the shape networks in brain subcortical structures. In this work, we study the topology via a novel topology data analysis method and test the features for their ability to differentiate between groups, and utility as a disease marker. Additionally, we compare these novel features with classical network features commonly studied in the scientific literature. On the lines of our previous work [2] we study the caudate and pallidum structures in both hemispheres.

The shape of an object is the geometry information retained after removal of position, orientation, and scaling (size) of an object. The change in shape of anatomical structures has been previously observed in our work with Parkinson's disease [2]. The structures inside the brain are closely located and often have touching boundaries. The shape change or deformation on one surface (e.g. medial boundary of caudate) is expected to influence the other surfaces (e.g. lateral boundary of the caudate) of the structure. This leads us to question, *Is there an interaction of shape change between the surface subfields of a structure ?*. We model this question as a topological data analysis problem where we study the topology of the surface displacement (SurfDisp) shape feature indexed on the geometry (nodes, group of vertices) on the structure. It is important to note that we do not develop a new shape feature, rather, our focus is to study the co-variation of shape and derive topological features for this information.

In this work, we address this question by studying the inter-regional co-variation of shape in the subcortical (deep gray matter) structures in the brain due to neurodegeneration. To this end, we apply the previously developed shape analysis framework and further model the topology of the network of interaction between subfields on the structure. The surface deformation based SurfDisp shape feature is our signal of interest where the difference in SurfDisp is a measure of covariation (similarity) between two regions within the structure. We compute the multiscale homology feature (persistence homology) of the induced Vietoris-Rips simplicial complex on the underlying topology of the SurfDisp network. We present experiments to test the ability of these features to differentiate between the disease and healthy groups on a group level and correctly classify previously unseen subjects.

2 Methods

The central aim of this work is to quantify the inter-regional covariation of shape between the surface subfields within the subcortical structures via the shape topology features. These features utilize the SurfDisp data as their signal of interest and model the topology of shape in brain structures. Here we describe each module of our pipeline followed by the statistical analysis and classification experiments.

2.1 Shape feature

The SurfDisp shape feature is obtained via a template injection approach where a population average template for each structure is injected into the surface of corresponding structure in each subject via non linear registration. This results in vertex wise correspondence between the surfaces of the template and target subjects. Further, the vector between the vertices of the reference (template) and the target surface is projected along the template surface normal to obtain the SurfDisp feature. The complete mathematical details of the SurfDisp method are available in our prior work in analysis of shape change in subcortical structures [2]. Specifically, for a structure with m vertices we obtain a surface displacement s_i at each vertex for a total of m values per subject.

2.2 Shape topology

The shape topology models the polyadic (many-to-many) interaction between the subfields on the surface of subcortical structures for covariation of shape (SurfDisp). To obtain such information we first obtain a reduced dimensionality representation of the shape data via adaptive parcellation, followed by a network filtration computed for each structure to obtain the homology of the shape topology space. The workflow is visually represented in the figure 1.

Adaptive Parcellation: The SurfDisp data inherently resides in a high dimensional space with large number of vertices per structure in comparison to the number of subjects in the data cohort. In order to mitigate the effects of curse of dimensionality and retain the computational tractability of the persistent homology features we perform adaptive parcellation by computing a patchwise averaged representation of the surface. In this, we begin by computing n patches (represented by their centroids) on the surface of the structure of the template as clusters of neighbouring vertices where 3D coordinates of a vertex form the input. The averaging is based on the assumption that neighbouring vertices exhibit similar form of shape change which differs from far and distant vertices. For each patch we compute the average SurfDisp value s_i resulting in n values per subject. As a result, m SurfDisp values (one per vertex) are converted into $n << m$ value (one per patch). The resultant matrix (subject × features) acts as an input to the persistent homology computation pipeline as outlined below. For each subject with n patches, we compute the distance $d_{ij} = s_i - s_j, i, j = 1, 2, \cdots, n$ resulting in a $n \times n$ square symmetric distance matrix D yielding a weighted undirected graph G.

Network Filtration: In complex network analysis, the threshold to obtain a binary graph from a weighted graph is a parameter often selected based on the suitability to the application at hand, thus, limiting its generalizability. To circumvent this issue, we construct a network filtration with a monotonically increasing set of threshold values to obtain a set of binary undirected networks with different levels of network sparsity. This approach has the advantage of providing a complete set of network topology characteristics from a fully disconnected network to a fully connected network.

A weighted undirected brain graph G is thresholded at a value ε_k to yield a binary undirected graph G_k. Upon changing the threshold $\varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_k < \cdots < \varepsilon_n$ we get a hierarchical sequence of n binary undirected graphs $G_1 \subseteq G_2 \subseteq \cdots \subseteq G_k \subseteq \cdots \subseteq G_n$ termed as a *Network Filtration.* A graph originates from a distance matrix D where, each entry d_{ij} is the connection strength between the nodes i and j. Each D is converted into an adjacency matrix A_k where, $\{a_{ij} = 1 | d_{ij} < \varepsilon_k, 0 \; otherwise\}$ for a chosen threshold ε_k giving the graph G_k. We compute the features of nodal degree, clustering coefficient and local efficiency for each network in the filtration. For a detailed mathematical description of these features, the reader is guided to the seminal work by Rubinov and Sporns [7].

The inter-patch distance for each subject varies depending on the brain size rendering the graph filtration generated based on the raw distance values influenced by the scale of the overall brain size in addition to the relative inter-regional distances whose alterations with disease are of interest. Thus, in order to overcome such scale-related differences, we normalize the values of the inter-patch distance for each individual to the range of $[0, 1]$ prior to the generation of a graph filtration. This enables a comparison of the network and topology features across subjects and groups by potentially reducing the affect of scale variation of the brain size.

2.3 Persistent Homology

The central idea behind the theory of persistent homology (PH) is to build a sequence of nested subsets on a space of simplicial complexes, studied at different resolutions. For our work, the Vietoris-Rips (VR) complex completely defined by the underlying 1-skeleton is induced on a symmetric distance matrix of pairwise distances between points in a point cloud.

A VR complex is defined on a metric space M for a specific distance value γ by forming a k-simplex for every finite set of $k+1$ points that has diameter at most γ. For a set of k nodes in the point cloud, the VR complex has at most $(k-1)$ simplices, enabling the geometry networks to obtain higher dimensional interactions limited in binary networks to 1-dimensional simplices (edges). Monotonically increasing values of the scale parameter ε_k lead to a VR filtration where $VR_{\varepsilon_1} \subseteq VR_{\varepsilon_2} \cdots \subseteq VR_{\varepsilon_k} \cdots \subseteq VR_{\varepsilon_n}$. For each filtered persistence module of the VR complex we obtain the tuples (b_i, d_i), with $b_i < d_i$ commonly known as a *birth-death pairs* of a k-dimensional simplex in the filtration. The length $d_i - b_i$ provides information of persistence of a k-simplex where long persistence times are suggestive of signal and short persistence times indicate towards noise.

Persistence diagrams: A *persistence diagram (PDia)* is a two dimensional representation of the birth and death times of the k-simplices in a given point cloud, where the horizontal axis is the birth time b_i and the vertical axis is the death time $d_i > b_i$. Each tuple (b_i, d_i) for a simplicial complexes is represented as a point in the 2-dimensional space. An overlay of persistence diagrams from two different point clouds enables comparison of two point clouds where a strong topological difference will be visible as a segregation of points in the PD space, and a strong overlap of points would suggest a topological similarity between the point clouds. Informally, a PD is a scatter plot of the persistence timelines where the x-axis is the birth time and the y-axis is the death time.

Persistence Landscapes: A persistence landscape (PL) for each $\{(b_i, d_i)\}_{i=1}^{n}$ is a sequence of functions $\lambda_k : \mathbb{R} \to [0, \infty], k = 1, 2, 3, \ldots$ where $\lambda_k(x)$ is the k-th largest value of $\{f_{b_i,d_i}(x)\}_{i=1}^{n}$ [1]. For every birth-death pair (b, d) we define a piecewise linear function $f_{(b,d)} : \mathbb{R} \to [0, \infty]$ such as:

$$f_{(b,d)} = \begin{cases} 0, & if x \notin (b,d), \\ x-b & if x \in (b, \frac{b+d}{2}], \\ -x+d & if x \in (\frac{b+d}{2}, b). \end{cases}$$

For a set of persistence landscapes $\lambda^1, \ldots, \lambda^N$ we compute the average landscape as $\bar{\lambda} = \sum_{j=1}^{N} \frac{1}{N}\lambda^j$.

Persistence Landscape Kernel: The distance between two persistence landscapes $\mathbb{L} = \{\mathbb{L}_k\}$ and $\mathbb{L}' = \{\mathbb{L}'_k\}$ can be obtained as the L^p norms for $1 < p < \infty$ which is defined as,

$$\|\mathbb{L}_k - \mathbb{L}'_k\|_p = \left[\sum_{k=1}^{K} \int \|\mathbb{L}_k - \mathbb{L}'_k\|_p^p\right]^{\frac{1}{p}} \tag{1}$$

and for $p = 2$, the L_2 distance between two persistence landscapes acts as a kernel metric between them named as a PL kernel [1].

Persistence Scale Space Kernel: The persistence scale space kernel (PSSK) [6] represents the multiset of points in a persistence diagram as a sum of dirac delta functions centered at each point. This enables the representation of points in persistence diagrams in a Hilbert space thereby supporting computation of a kernel between two point. Briefly, for two persistence diagrams F and G we compute the PSSK kernel ($k_\sigma(F, G)$) as:

$$k_\sigma(F,G) = \frac{1}{8\pi\sigma} \sum_{p \in F, q \in G} \exp^{-\frac{\|p-q\|^2}{8\sigma}} - \exp^{-\frac{\|p-\bar{q}\|^2}{8\sigma}} \tag{2}$$

where each $p = (b_i, d_i)$, $q = (b_j, d_j)$ and $\bar{q} = (d_j, b_j)$. For two persistence timelines represented as persistence diagrams we can compute the kernel matrix between all data groups.

2.4 Experiments

We tested the classical network features and persistence homology (PL & PDia) features for group level difference and their ability to predict previously unseen subjects in experiments as outlined below.

Group difference analysis: For the two groups of persistence landscapes $\mathbb{L}^1, \ldots, \mathbb{L}^N$ and $\mathbb{L}^1, \ldots, \mathbb{L}^M$, let δ be the true L_p distance between their average PLs, $\overline{\mathbb{L}_N}$ and $\overline{\mathbb{L}_M}$. We permute the group labels and compute the group level average landscapes $\overline{\mathbb{L}_N}$ and $\overline{\mathbb{L}_M}$ and find the corresponding L_p distance between them. The p-value of the statistical test equals the proportion of random permutations in which the distance between $\overline{\mathbb{L}_N}$ and $\overline{\mathbb{L}_M}$ is greater than the true difference.

For the classical network features of nodal degree, local efficiency and clustering coefficient for the graphs in the filtration for each subject, we test the group level difference between the features in the two groups in a Hotelling's T2

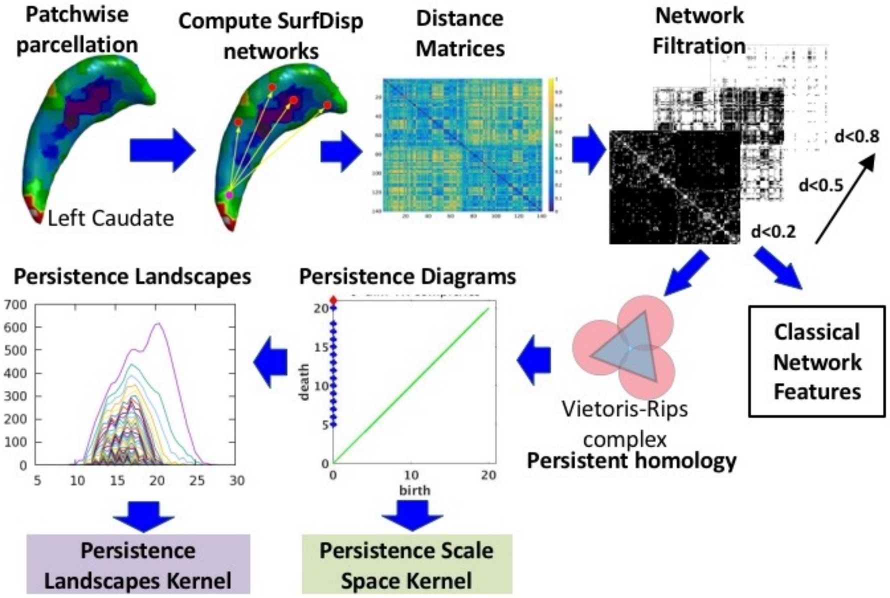


Fig. 1. Method workflow for the shape topology networks.

test in a permutation testing experiment (2000 permutations) after PCA dimensionality reduction into a reduced dimensionality space. The experiment tests for the hypothesis that the feature values originate from two distributions with same mean value, rejecting the hypothesis at $\alpha = 0.05$ and p-value < 0.05.

Classification experiment: To test the ability of the network and PH features to correctly classify unseen subjects, we trained a kernel support vector machine classifier with a radial basis function (RBF) kernel for the complex network features, the PL kernel for the PL features and the PSSK kernel for the PDia features. The classifier was trained in a random and repeated holdout stratified training experiment with parameter tuning for the RBF and PSSK kernels. Results for the accuracy, sensitivity, specificity and F1-measure are reported for each classification experiment.

Computational tools: We input the distance matrices for the geometry networks in the package *Perseus* with the parameter (*distmat*) to compute the PH of the Vietoris-Rips complex for each brain point cloud from the inter-patch distance matrices [5]. We obtain birth-death pairs (b_i, d_i) for the k-dimensional simplices for $k = 0, 1, 2, 3$. A recent *persistence landscapes* toolbox was released for research use by [1] enabling computation and statistical inference of the persistence landscapes. The number of landscapes λ^i varies dependent upon the underlying persistence diagrams which inherently depend on the birth-death pairs (b_i, d_i).

Further, we perform permutation testing on the persistence landscapes in the two groups for 2000 random group assignments.

2.5 Imaging and Demographics

Imaging data for this work was obtained from the publicly available database provided under the Parkinson's Progressive Markers Initiative (*PPMI*). Detailed protocol for image acquisition and quality control for the study is available at the website *www.ppmi-info.org*. The two groups with De Novo PD patients (n = 189, age = 68.02±4.77, 115M/74F) and healthy controls (CN) (n = 137, age = 63.85±7.46, 75M/62F) were selected and analysed through the method as described above. The original T1 MRI images were first preprocessed to obtain the segmentation labels for the subcortical structures via a multi-template registration based segmentation method (FS+LDDMM [3]). The segmentation outlines and surfaces were quality controlled prior to the computation of SurfDisp shape feature.

3 Results

The homology of the SurfDisp feature only had 0-dimensional PH features, higher dimensional homology features were not present in the SurfDisp networks for all structures. The persistence landscapes showed statistically significant difference between the two groups for all structures (table 1). The local efficiency (L-caud,L-Pall), clustering coefficient (R-caud) and nodal degree (L-pall) were significantly different between the two groups. The PL kernel showed poor performance in classification experiments, on the contrary, the PSSK kernel showed good performance (ACC=74.9, 75.1) for left and right pallidum (table 2). The classical network features were unable to correctly classify subjects in the two groups (table 3).

Table 1. Group level difference performance of the surface displacement network features in the permutation testing experiment to differentiate Parkinson's disease and healthy groups.

Feature	Caudate		Pallidum	
	L	R	L	R
Persistence Landscape β_0	0*	0*	0*	0*
Local Effficiency	0.030*	0.333	0.0064*	0.482
Clustering Coefficient	0.80	0.028*	0.121	0.261
Nodal Degree	0.231	0.003	0.049*	0.5829

Table 2. Classification performance of persistence homology features of surface displacement networks in Parkinson's disease.

	PL			
	Sens	Spec	F1	Accuracy
lcaud	0.495	0.510	0.615	49.753
rcaud	0.503	0.471	0.618	49.727
lpall	0.495	0.494	0.613	49.506
rpall	0.496	0.527	0.617	50.156

	PSSK			
	Sens	Spec	F1	Accuracy
lcaud	0.533	0.473	0.642	52.195
rcaud	0.465	0.480	0.585	46.766
lpall	0.883	0.145	0.847	**74.91**
rpall	0.886	0.141	0.852	**75.01**

caud: caudate, pall: pallidum
PL: persistence landscape kernel,
PSSK: persistence scale space kernel,
Sens: sensitivity, Spec: specificity,
F1: F1-measure, Acc: accuracy.

Table 3. Classification performance of classical network features of surface displacement networks in Parkinson's disease.

		Sens	Spec	F1	nPCs	Acc
lcaud	2	0.603	0.425	0.661	63.920	55.524
	3	0.568	0.466	0.611	8.420	54.095
	4	0.567	0.553	0.653	50.850	56.333
rcaud	1	0.471	0.678	0.590	82.560	52.610
	2	0.571	0.381	0.599	11.850	52.067
	3	0.528	0.547	0.619	61.190	53.333
lpall	1	0.595	0.466	0.662	49.030	56.029
	2	0.686	0.331	0.675	5.900	59.105
	3	0.544	0.519	0.630	47.010	53.724
rpall	1	0.499	0.508	0.592	36.740	50.143
	2	0.372	0.650	0.433	3	44.571
	3	0.530	0.514	0.619	34.640	52.552

1) Clustering Coefficient, 2) Nodal Degree,
3) Local Efficiency
Sens: sensitivity, Spec: specificity,
F1: F1-measure, Acc: accuracy.

4 Discussion & Conclusion

In this work, we aimed to quantify the inter-regional covariation of shape between subfields of subcortical structures. To this end, we computed the persistence homology and classical network analysis features for SurfDisp data indexed on the surface of the structures. It is interesting to note that the SurfDisp data on the subcortical structures only showed a 0-dimensional homology, whereas, higher dimensional homology was not present with the Vietoris-Rips complex. This can be attributed to the small distribution of values in the surface displacement data, where the 0-dimensional homology is present and the 1,2 &3 dimensional homology components are not present. Additionally, we can infer that the SurfDisp point cloud connectivity grows through the filtration in a single large connected component, possibly attributable to small spread of SurfDisp values in the data space.

The focus of this work was to test performance of the PH features in comparison to classical network features. In the statistical experiments, significant group level difference was found in the persistence landscapes and some network features for the structures. However, the classification performance was subpar and was unable to correctly predict previously unseen subjects. However, the PSSK kernel for the right and left pallidum showed good performance to correctly classify subjects. This suggests that the PL features contain information that is distinguishable on a group level, however, share a strong overlap for it to identify individual subjects. Thus, suggesting that the approximation of the persistence diagrams to persistence landscapes potentially leads to loss of information, which is otherwise captured by the PSSK kernel.

In this work our goal was to quantify and study the inter-regional co-variation of shape change in subcortical structures with brain abnormalities. The topology of the underlying data space was quantified in the persistence homology features and studied for their strengths to identify group level and subject level differences due to brain abnormalities. The results suggest a robust ability of the method and its derived features to differentiate on a group level. The features did not show a consistent and strong performance to predict individual subjects suggestive of wide variability between subjects overpowering the differences between subjects. Future work on bigger data cohorts is expected to enhance the subject level prediction of disease conditions. The feature is sensitive to disease and brain abnormalities as it is able to successfully differentiate between groups, where, on average, large changes can be observed. However, the prediction of disease via correct classification of individuals depends upon the sensitivity of the feature to changes within a subject.

The PH features computed in the current work showed moderate performance to predict individual subjects in a machine learning model. This can be associated with the averaging of features into small number of patches to obtain connectivity between SubCortical surface ROIs. Further, the surface displacement feature has both outward (positive) and inward (negative) deformation of the surface. Thus, smaller patches are needed to avoid the averaging affect on large patches potentially reducing the sensitivity of the SurfDisp data. In the current work, we limited to large patch size due to the limits of tractability of the persistence homology computation. Further development of computationally efficient algorithms would greatly solve this limitation is expected to yield state-of-the-art performance in prediction of disease.

This is a first work to study the persistence homology of a shape feature for subcortical structures and can potentially benefit from newer methodological extensions. We studied the SurfDisp shape feature, however, the general nature of the Shape topology method enables its applicability to other shape features such as spherical harmonics, initial momentum an the like. Further extensions to include more complex distance functions or better homology features can potentially improve it application in clinical settings.

Acknowledgment Data used in the preparation of this article were obtained from the Parkinsons Progression Markers Initiative (PPMI) database (http://www.ppmi-info.org/data). For up-to-date information on the study, visit www.ppmi-info.org. The authors acknowledge and thank the Natural Sciences and Engineering Research Council (NSERC), Michael Smith Foundation for Health Research (MSFHR), Canadian Institute of Health Research (CIHR), Brain Canada and MITACS Canada for their generous funding support.

References

1. Bubenik, P.: Statistical topological data analysis using persistence landscapes. Journal of Machine Learning Research 16, 25 (2015)
2. Garg, A., Appel-Cresswell, S., Popuri, K., McKeown, M.J., Beg, M.F.: Morphological alterations in the caudate, putamen, pallidum, and thalamus in Parkinson's disease. Frontiers in Neuroscience 9(March), 1–14 (2015)
3. Khan, A., Wang, L., Beg, M.: FreeSurfer-Initiated Fully-Automated Subcortical Brain Segmentation in MRI Using Large Deformation Diffeomorphic Metric Mapping. NeuroImage 41(3), 735–746 (2008)
4. McKeown, M.J., Uthama, A., Abugharbieh, R., Palmer, S., Lewis, M., Huang, X.: Shape (but not volume) changes in the thalami in Parkinson disease. BMC neurology 8, 8 (jan 2008)
5. Mischaikow, K., Nanda, V.: Morse Theory for Filtrations and Efficient Computation of Persistent Homology. Discrete and Computational Geometry 50(2), 330–353 (2013)
6. Reininghaus, J., Huber, S., Bauer, U., Kwitt, R.: A stable multi-scale kernel for topological machine learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 07-12-June, pp. 4741–4748 (2015)
7. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: Uses and interpretations. NeuroImage 52(3), 1059–1069 (sep 2010)

Graph Geodesics to Find Progressively Similar Skin Lesion Images

Jeremy Kawahara, Kathleen P. Moriarty, and Ghassan Hamarneh

Medical Image Analysis Lab, Simon Fraser University, Burnaby, Canada
{jkawahar, kmoriart, hamarneh}@sfu.ca

Abstract. Skin conditions represent an enormous health care burden worldwide, and as datasets of skin images grow, there is continued interest in computerized approaches to analyze skin images. In order to explore and gain insights into datasets of skin images, we propose a graph based approach to visualize a progression of similar skin images between pairs of images. In our graph, a node represents both a clinical and dermoscopic image of the same lesion, and an edge between nodes captures the visual dissimilarity between lesions, where dissimilarity is computed by comparing the image responses of a pretrained convolutional neural network. We compute the geodesic/shortest path between nodes to determine a path of progressively visually similar skin lesions. To quantitatively evaluate the quality of the returned path, we propose metrics to measure the number of transitions with respect to the lesion diagnosis, and the progression with respect to the clinical 7-point checklist. Compared to baseline experiments, our approach shows improvements to the quality of the returned paths.

Keywords: Graph geodesics, skin lesions, visualizing similar images

1 Introduction

Globally, skin conditions are the fourth most common cause of healthy years lost due to disability [6], and represent the most common reason for a patient to visit their general practitioner in studied populations [16]. Skin conditions such as malignant melanoma, a common cancer, can be fatal [14]. Many groups recognize the potential for computerized systems to analyze skin lesions and help reduce the burden on health care, and much work has gone into developing computerized systems to diagnose skin disorders [13]. Typically, such systems take as input a skin lesion image, and output either a discrete label or the probability that this lesion has a particular diagnosis. For example, Estevan et al. [5] used a human designed taxonomy to partition clinically similar images into classes that have a similar number of samples. They fine-tuned a Convolutional Neural Network pretrained over ImageNet [15] to classify skin lesions, and achieved results comparable to human dermatologists. While knowing the probability that the image contains a particular type of skin lesion is a worthwhile goal, a disadvantage to this approach is that it is a "black box", where the user gains no insights into the automated diagnosis or of the underlying dataset of skin images.

A different approach from classification that offers some insights into the dataset or diagnosis is to adopt an image retrieval based approach. For example, Bunte et al. [3] extracted colour features from clinical skin images, learned a supervised transformation of these features, and retrieved images in a dataset based on the k nearest neighbours to these features. These returned images can be displayed to the user, giving insights into the appearance of similarly diseased images and allow the diagnosis to be inferred. Kawahara et al. [10] displayed a network graph visualization based on the nearest neighbours to a single query image, which allow users to efficiently search the space of similar lesion images.

Another approach to visualize *general images* was proposed by Hegde et al. [7], where rather than retrieving the k nearest images to a single query image, their approach uses two query images (a source and target) to retrieve a list of images that progress in visual similarity between the two images. They accomplish this by representing images as nodes in a graph, where the edges between nodes indicate their pair-wise distance, and the geodesic (shortest path) between source and target nodes represents a visually smooth progression of images. A similar approach for *general images* was recently implemented online [12], which is based on an experimental visualization tool as part of Google Arts and Culture [11]. In other works, representing images as nodes to find an optimal path between nodes has been used to guide subject-template brain registration in MR images [8].

In this work, we apply a similar method to find images of skin lesions that visually progress between a source and target lesion. This visualization approach may be useful for clinicians who wish to find reference images of hard to classify, visually challenging "borderline" cases across types of skin diseases (e.g., note the visually challenging aspects in distinguishing clark nevus from melanoma in Fig. 1 *bottom row*). Another use may be to show or predict the visual progression over time between a low-risk benign lesion to a malignant lesion (e.g., progression in Fig. 3 *bottom row*). This may give insights into the progression of the disease (e.g., Clark/Dysplastic nevi is potentially a precursor to melanoma and studies estimate that 20-30% of melanomas come from nevi [4]), or serve as a useful reference for patients to monitor and compare the progression of their own lesion. In these potential applications, the target images could be from either a set of predefined reference images, or the geodesics to each of the nearest unique diseases could be automatically shown.

To the best of our knowledge, this is the first work that has applied geodesic paths to visualize skin lesion images. In contrast to previous work [7,8,11,12], we propose to let each node in our graph represent images from two modalities (a dermoscopic and a clinical image), where the edge weights are influenced by both types of images. We apply an exponential function to the pair-wise dissimilarity measures, and show how this results in longer paths of higher quality without risking disconnected graphs. Finally, we propose measures to quantitatively evaluate the quality of our paths, which is lacking in prior work. These proposed quality measures are particularly important as without them, we would need to qualitatively inspect each path.

2 Methods

A skin lesion can be captured by both a dermatoscope (producing a *dermoscopic* image x_d), and a photo camera (producing a *clinical* image x_c), where the dermoscopic images show a more standardized view of the lesion, and the clinical images are non-standardized and often show additional contextual information (e.g., the body part the lesion is on) not available in the dermoscopic images. Given a dataset of skin lesions, the i-th skin lesion is represented by a dermoscopic and clinical pair of images $(x_d^{(i)}, x_c^{(i)})$. We create a graph where each pair of images $(x_d^{(i)}, x_c^{(i)})$ are represented by a single node $v^{(i)}$, and an edge $e^{(ij)}$ encodes the dissimilarity between nodes i and j. Our goal is to find a set of nodes $(v_0^{(s)}, v_1^{(i)}, \ldots, v_{R-1}^{(j)}, v_R^{(t)})$ of an unknown length R such that the initial node $v_0^{(s)}$ is a given source node (the superscript identifies the lesion, and the subscript indicates the position in the returned path), the R-th node is a given target node $v_R^{(t)}$, and the intermediate nodes $(v_1^{(i)}, \ldots, v_{R-1}^{(j)})$ represent lesions that visually progress between the source and target nodes. We find these intermediate nodes using Dijkstra's algorithm, which computes the geodesic between the source and target and returns a path of nodes representing a progression of visually similar lesions.

The key components that we now examine in detail are how to: extract image features that capture the salient properties of skin images, compute local dissimilarity between pairs of skin lesion images, weigh and connect the node edges using multi-modal images, and quantitatively evaluate the quality of the returned paths.

Skin Images as Deep Pretrained Neural Nets Responses. The responses of skin images with deep convolutional neural networks pretrained over ImageNet [15] have shown to be effective feature vectors for skin lesion classification despite the differences in appearance between skin lesions and natural images [9]. We use a similar approach to compute feature vectors as in [9], and for a particular image, extract the responses from the first fully connected layer of VGG16 [17], and average the responses over a set of predefined image augmentations,

$$\Phi(x)_m = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \phi(\pi(x - \mu))_m \tag{1}$$

where π is a function to augment an image (e.g., left-right flip); Π is the set of $|\Pi|$ number of image augmentations; $\phi(\cdot)_m$ extracts the m-th response of the first fully connected layer of VGG16; and, μ represent the mean pixel over the training data from ImageNet, which is subtracted from the skin lesion image x. The resulting feature vector $\Phi(x)$ represents a single lesion image by averaging the augmented responses over a single image, without increasing the dimensions of the feature vector.

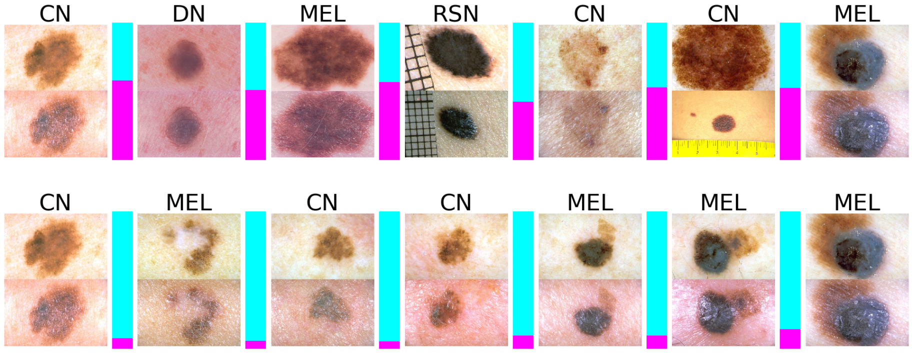


Fig. 1: An example random path (*top*) and geodesic returned from the proposed method (*bottom*), where the *leftmost* and *rightmost* image represent the source and target nodes, respectively. The dermoscopic image is shown above the clinical image in each row. The *magenta bar* indicates the dissimilarity between adjacent images, where a higher bar indicates that they are more dissimilar.

Local Image Dissimilarity. Given two feature vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^M$ (which represent the responses of two skin images), we compute the dissimilarity between them as the cosine distance raised to the p-th power,

$$\mathcal{D}(\boldsymbol{u}, \boldsymbol{v}) = \left(1 - \frac{\sum_i^M u_i v_i}{\sqrt{\sum_i^M u_i^2}\sqrt{\sum_i^M v_i^2}}\right)^p \tag{2}$$

where setting $p \neq 1$ non-linearly changes the dissimilarity between vectors. By using a high p (e.g., $p = 4$), we assign very low values to edges connecting similar images, thus encouraging geodesics to pass through nearby nodes of similar images, avoiding very short paths even in the case of complete graphs, i.e., fully connected graphs (further discussed in the Results section). Other distance measures are possible (e.g., L_1, L_2), and we found them to give empirically similar results. Fig. 1 shows the dissimilarity between pairs of images (dissimilarity is displayed in magenta using $p = 1$ for clarity).

Multi-modal Edge Weights. We define the edge weight $e^{(ij)}$ between nodes i and j as a weighted sum based on both the dermoscopic and clinical images,

$$e^{(ij)} = \alpha\mathcal{D}(\Phi(x_d^{(i)}), \Phi(x_d^{(j)})) + (1 - \alpha)\mathcal{D}(\Phi(x_c^{(i)}), \Phi(x_c^{(j)})) \tag{3}$$

where $\mathcal{D}(\cdot)$ is a function that computes the dissimilarity (Eq. 2) between the feature vectors $\Phi(\cdot)$ computed in Eq. 1; and α weighs the influence of the dermoscopic and clinical images ($0 \leq \alpha \leq 1$). Increasing α causes an edge to be more influenced by the dermoscopic image than the clinical image, which may be desired as dermoscopic images contain more salient lesion properties.

Node Connectivity. To form the graph, we must decide on the connectivity of nodes. This can be done by connecting the k nearest neighbours (where nearest is defined via Eq. 2) to each node with an edge. However, choosing k is challenging as a large k (e.g., a complete graph) increases computational complexity and can lead to very short paths being returned when a direct edge exists between any pair of source and target nodes. Too small a k can lead to disconnected graphs, where no path exists between the source and target nodes. In the Results, we experiment with different values of k and show that by setting a high value of p in Eq. 3, the returned paths remain longer even in the case of complete graphs.

Surrogate Measures of Path Quality. While we provide qualitative results through visualizing the returned paths (Fig. 3), we also propose the following measures to quantitatively evaluate the quality of the returned paths. We define a *quality path* as a smooth visual progression of images. However, this definition is hard to precisely define and directly measure. Thus we propose a surrogate measure that uses the diagnoses of the lesions, as skin lesion datasets are often accompanied with a corresponding clinical diagnosis $y^{(i)}$ (e.g., melanoma, nevus), indicating the disease type of the i-th lesion $x^{(i)}$, where $y^{(i)}$ is an attribute of node $v^{(i)}$. Our assumption is that lesions with the same diagnosis will likely be visually similar, and that a high quality path will have a smooth progression with respect to the lesion diagnosis. In order to give a high cost to paths that frequently change neighbouring labels, we define the *transition cost* as,

$$\mathrm{trans}(v_0^{(s)}, v_1^{(i)}, \ldots, v_{R-1}^{(j)}, v_R^{(t)}) = \frac{1}{R-1} \sum_{r=1}^{R} \Big(1 - \delta(y_r^{(a)} - y_{r-1}^{(b)})\Big) \tag{4}$$

where R is the number of nodes in the returned path; and $y_r^{(i)}$ indicates the skin lesion diagnosis for the r-th returned path node corresponding to node $v_r^{(i)}$ (e.g., $y_0^{(s)}$ and $y_R^{(t)}$ correspond to the labels of the source and target nodes $v_0^{(s)}, v_R^{(t)}$ respectively). The Dirac delta function $\delta(\cdot)$ returns 1 if the two labels have the same class, and 0 otherwise.

Our second surrogate quality measure quantifies the progression of the 7-point score between the source and target nodes. The 7-point score is a clinical measure of melanoma based on the visual presence of seven criteria (e.g., irregular streaks) within a lesion. The weighted sum of these seven criteria form the 7-point score $\hat{y} \in \mathbb{Z}$ [1], where $\hat{y}^{(i)}$ is an attribute of node $v^{(i)}$. We assume that a quality path will have 7-point scores that smoothly progress from a low to high score, as higher scores indicate the presence of lesion more indicative of melanoma (and vice versa). We define the *progression cost* as,

$$\mathrm{progress}(v_0^{(s)}, \ldots, v_R^{(t)}) = \frac{1}{R} \sum_{r=1}^{R} \Big(\max\Big[\Big(\mathrm{sgn}(\hat{y}_0^{(s)} - \hat{y}_R^{(t)})(\hat{y}_r^{(i)} - \hat{y}_{r-1}^{(j)})\Big), 0\Big]\Big) \tag{5}$$

where sgn(z) returns the sign of the difference between the source and target node scores,

$$\mathrm{sgn}(z) = \begin{cases} 1, & \text{if } z = 0 \\ \frac{z}{|z|}, & \text{otherwise.} \end{cases} \tag{6}$$

This measure returns a cost of 0 if the 7-point score consistently decreases, increases, or remains constant along the path between the source and target nodes, and penalizes by the magnitude of the change otherwise. This approach, however, will always compute a 0 cost if the path only consists of the source and target nodes. As this is a degenerate case, we ignore the progression costs for paths of length two when computing results, and note that this measure is biased to return lower costs for shorter paths, and is thus most informative when comparing paths with the same number of nodes.

3 Results

Data. We test our proposed approach and surrogate measures using the Interactive Atlas of Dermoscopy [2] skin dataset. This dataset contains 1011 cases of skin lesions, where all but four cases are captured by both a clinical x_c and dermoscopic x_d image (in the four cases missing x_c, we set $x_c = x_d$). Each case has a class label y that represents a known lesion diagnosis, and a 7-point score $\hat{y}$. The diagnosis y can take on one of the 15 class labels: basal cell carcinoma (BCC), blue nevus (BN), clark nevus (CN), combined nevus (CBN), congenital nevus (CGN), dermal nevus (DN), dermatofibroma (DF), lentigo (LT), melanoma (MEL), melanosis (MLS), miscellaneous (MISC), recurrent nevus (RN), reed or spitz nevus (RSN), seborrheic keratosis (SK), and vascular lesion (VL). The 7-point score $\hat{y} \in \mathbb{Z}$ ranges between 0 and 7 (in this dataset), where a higher score indicates the lesion has visual properties more indicative of melanoma. The lesion diagnosis and the 7-point score are only used to quantify the quality of the returned paths, and are not used to form the graph. We randomly select a set of 1000 pairs of source and target nodes which are used across all experiments.

Recovering Synthetic Paths. We start by testing if our proposed approach can recover the path of images created by a progressive synthetic transformation. To do this, we crop the image by removing 15% of the pixels at the borders of the images, and repeat this five times. This progressively enlarges the lesion over a series of five images. We added these five synthetic images to our dataset, select the original image as the source and the final synthetic image as the target ($p = 4$ and $k = 30$). We find our approach not only recovers all synthetic images, but it recovers the correct sequence of synthetic images, i.e. in the order they were synthesized (Fig. 2), indicating that this approach and the feature vectors are sensitive to scale despite the CNN being trained on images at multiple scales.

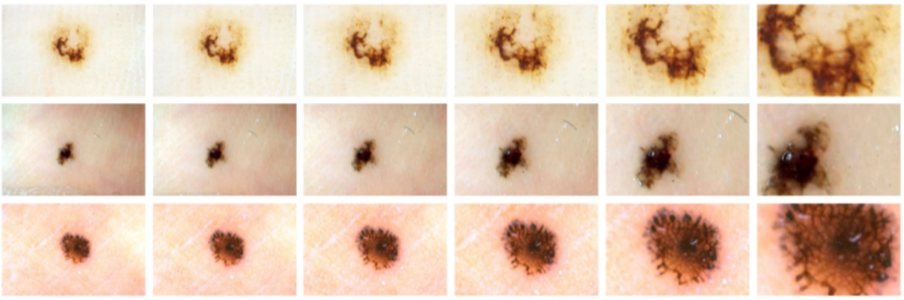


Fig. 2: Synthetic examples: Here the *leftmost* images represent the source nodes, which belong to the original (non-enlarged) dermoscopic images in the dataset. The *rightmost* images represent target nodes, which were the last of the progressively enlarged images. The returned geodesic path is represented by the images in between. Note that the returned geodesic included all five synthetic images, in proper order of increasing enlargement.

Retrieving Paths from a Complete and Non-Complete Graphs. For our first experiment, in Table 1 **row 1.1** (*complete graph with* $p = 1$) we report results using a complete graph (i.e., $k = 1011$) using only the dermoscopic images (i.e., $\alpha = 1$ in Eq. 1) and setting $p = 1$ in Eq. 3. We observe that when using a complete graph, the returned paths often consist of only the source and target nodes as their shared edge yields the shortest path. This experiment highlights the need to either prune the edges in the graph or modify the edge weights. Following the approach of [12], we form a new graph where each node is connected to its $k = 30$ neighbours. **Row 1.2** (*non-complete graph with* $p = 1$) shows that restricting the node connectivity increases the number of nodes in the returned path and improves the transition cost (note that the progression cost performs worse as it is biased towards paths with fewer nodes, and is thus most informative when comparing paths with a similar number of nodes).

Paths with Exponential Edge Weights. While decreasing node connectivity (i.e., lowering k) results in longer paths, care must be taken when choosing k, as reducing k increases the risk of forming disconnected graphs where no path exists between a source and target node. Thus instead of pruning edges, our next experiment (**row 1.3** *complete graph with* $p = 4$) shows how applying an exponential function (i.e., $p = 4$ in Eq. 3) to the dissimilarity function results in longer paths of higher quality even in a complete graph. By removing the need to prune graphs (i.e., choose k), we guarantee a path to exist, while still preventing short paths. If we are not concerned with disconnected graphs, we can combine edge pruning using k neighbours with the increased p, to match the computational efficiency of a pruned graph without penalty to quality (**row 1.4** *non-complete graph with* $p = 4$). For the remaining experiments, we use $p = 4$ and non-complete graphs with $k = 30$, as our graphs remained connected.

Table 1: Quantitative results of the returned paths using the proposed surrogate quality measures. The *Img.* column indicates if the input was a dermoscopic image x_d, clinical image x_c, or included both. k represents the number of nearest neighbours used to form edges that connect nodes. *Aug.* indicates if the image was augmented or not when forming the image feature vector. *Trans., Progress.,* indicates the average and standard deviation transition and progression cost as defined in the text. *Num. Path* shows the average and standard deviation number of nodes in the computed path.

Exp.	Img.	Aug.	p	k	Ordered	Trans.	Progress.	Num. Path
1.1	x_d	✗	1	1011	min-path	0.76 ± 0.42	0.10 ± 0.19	2.02 ± 0.13
1.2 [12]	x_d	✗	1	30	min-path	0.64 ± 0.34	0.23 ± 0.26	3.59 ± 0.85
1.3	x_d	✗	4	1011	min-path	0.56 ± 0.26	0.37 ± 0.20	8.11 ± 2.87
1.4	x_d	✗	4	30	min-path	0.56 ± 0.26	0.37 ± 0.20	8.12 ± 2.87
1.5	x_d	✓	4	30	min-path	0.55 ± 0.25	0.35 ± 0.17	9.16 ± 3.62
1.6	-	-	-	-	random	0.76 ± 0.19	0.54 ± 0.24	9.16 ± 3.62
1.7	x_d	✓	-	-	linear	0.58 ± 0.25	0.44 ± 0.21	9.16 ± 3.62
1.8	x_c	✗	4	30	min-path	0.65 ± 0.18	0.46 ± 0.20	10.64 ± 5.08
1.9	x_d, x_c	✗	4	30	min-path	0.45 ± 0.24	0.34 ± 0.19	7.90 ± 3.27
1.10	x_d, x_c	✓	4	30	min-path	0.45 ± 0.23	0.34 ± 0.17	8.86 ± 3.73

Comparing Random and Linearly Interpolated Path. In **row 1.5** (*augmented images*) we augment the feature vector with left-right image flips (Eq. 1), which results in longer geodesics paths and minor improvements to the path quality. We form a path with an equal number of nodes as those returned in the geodesic path in the previous experiment (from row 1.5) by randomly sampling nodes (without replacement). As the labels in our dataset are highly imbalanced, these random paths give us a baseline quality score (**row 1.6** *random paths*). We also compare our method by ignoring the graph, and instead using linearly interpolated feature vectors between the source and target feature vectors. These interpolated feature vectors are uniformly separated to match the number of returned nodes in row 1.5. The nearest unique neighbour to this interpolated feature vector is used to form the path. **Row 1.7** (*linear paths*) shows that this approach yields paths of worse quality when compared to using graph geodesics. We highlight that the graph geodesic approach has the additional advantage of automatically determining the number of nodes in the path, whereas the linearly interpolated approach requires this to be specified (we set it equal to the length of the geodesic path).

Using Clinical Image Features. In **row 1.8** (*clinical images*) we use only the clinical image (i.e., $\alpha = 0$ in Eq. 3) and notice a marked decrease in the quality of the paths when compared to dermoscopic images. This is expected since dermoscopic images are more standardized and focused on the lesion, while clinical images have a non-standard field of view and can capture background artifacts.

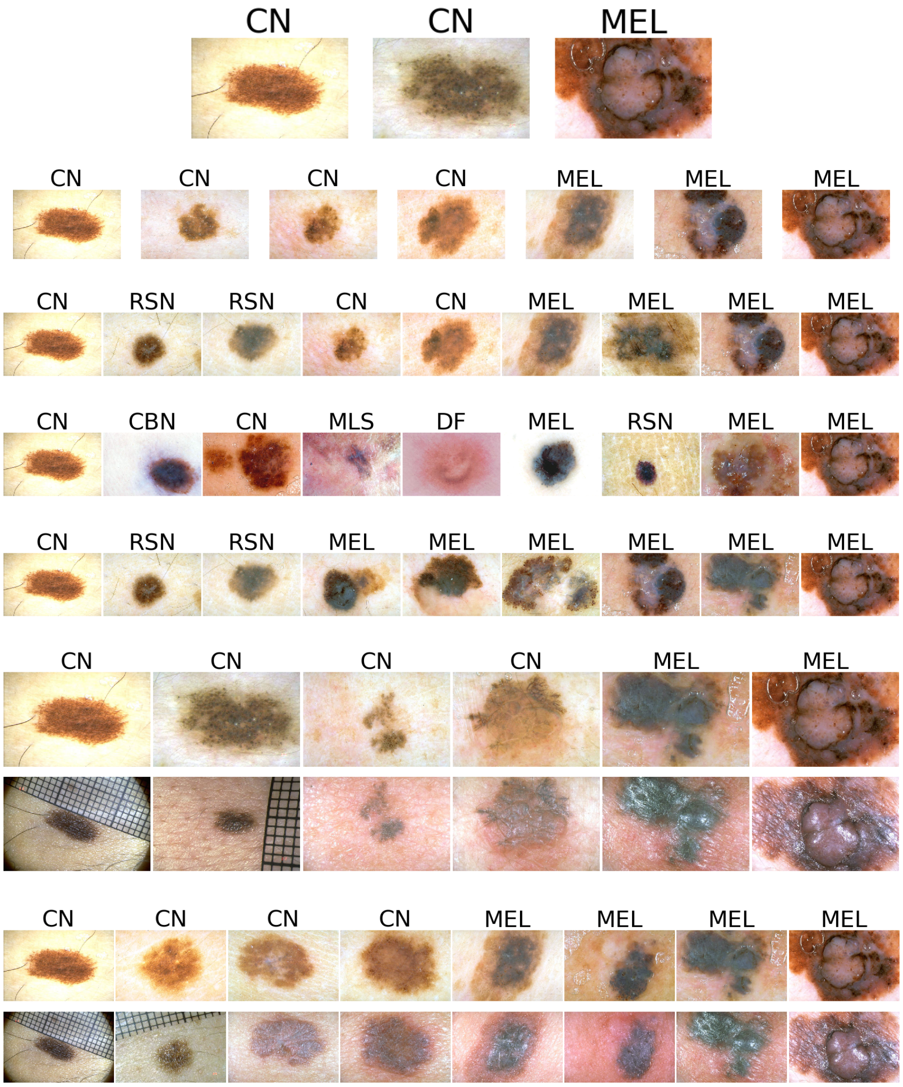

CN CN MEL
CN CN CN CN MEL MEL MEL
CN RSN RSN CN CN MEL MEL MEL MEL
CN CBN CN MLS DF MEL RSN MEL MEL
CN RSN RSN MEL MEL MEL MEL MEL MEL
CN CN CN CN MEL MEL
CN CN CN CN MEL MEL MEL MEL

Fig. 3: Visualizing Paths. The *leftmost* and *rightmost* dermoscopic images are the given source (clark nevus) and target (melanoma) node, where the images in each row in between them correspond to the computed geodesic/minimal path. Each row, starting from the *top* to *bottom* row, correspond to the following experiments in Table 1: 1.2 (*non-complete graph with* $p = 1$), 1.4 (*non-complete graph with* $p = 4$), 1.5 (*augmented images*), 1.6 (*random paths*), 1.7 (*linear paths*), 1.9 (*dermoscopic and clinical images*), and 1.10 (*full approach*). The geodesic of Experiments 1.9 and 1.10 incorporates clinical images, shown directly below the dermoscopic images.

Combining Dermoscopic and Clinical Image Features. In **row 1.9** (*dermoscopic and clinical images*) we include both the clinical and dermoscopic images, weighting the dermoscopic images higher (i.e., $\alpha = 0.8$ in Eq. 3) as the dermoscopic images better capture the salient lesion features and avoid irrelevant background artifacts. The returned paths now respect both imaging modalities, yielding improvements to the quality of the paths, most noticeable with transition costs. Finally, in **row 1.10** (*full approach*) we show the full proposed approach, which uses augmented images from both modalities with the dissimilarity measure raised to the power of $p = 4$ on a non-complete graph. While the path quality measures remain similar to the previous experiment, the total path length increases.

4 Conclusions

We proposed a method to visualize a smooth progression of similar skin lesion images between two skin lesions. Our graph geodesic based approach applies an exponential dissimilarity function and considers information from multiple modalities (clinical and dermoscopic images) to form the graph edges, leading to longer paths of higher quality. We proposed surrogate measures of path quality based on the diagnostic labels of the skin lesions to quantitatively assess the resulting paths. Future work would explore how to improve the feature vectors that represent the skin images (e.g., fine-tuning the CNN over a skin dataset), and examine how to make the progression quality measure less sensitive to the length of the path.

Acknowledgments. Thanks to the Natural Sciences and Engineering Research Council (NSERC) of Canada for funding and to the NVIDIA Corporation for the donation of a Titan X GPU used in this research. Thanks to Sara Daneshvar for preparing the data used in this work.

References

1. Argenziano, G., Fabbrocini, G., Carli, P., Vincenzo, D.G., Sammarco, E., Delfino, M.: Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions. Comparison of the ABCD rule of dermatoscopy and a new 7-point checklist based on pattern analysis. Arch.Dermatol. 134(12), 1563–1570 (1998)
2. Argenziano, G., Soyer, H.P., Giorgio, V.D., Piccolo, D., Carli, P., Delfino, M., Ferrari, A., Hofmann-Wellenhof, R., Massi, D., Mazzocchetti, G., Scal-venzi, M., Wolf, I.H.: Interactive atlas of dermoscopy: a tutorial (Book and CD-ROM) (2000)
3. Bunte, K., Biehl, M., Jonkman, M.F., Petkov, N.: Learning effective color features for content based image retrieval in dermatology. Pattern Recognition 44(9), 1892–1902 (2011)
4. Duffy, K., Grossman, D.: The dysplastic nevus: From historical perspective to management in the modern era: Part I. Historical, histologic, and clinical aspects. Journal of the American Academy of Dermatology 67(1), 1–27 (2012)

5. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639), 115–118 (2017)
6. Hay, R.J., Johns, N.E., Williams, H.C., Bolliger, I.W., Dellavalle, R.P., Margolis, D.J., Marks, R., Naldi, L., Weinstock, M.a., Wulf, S.K., Michaud, C., J L Murray, C., Naghavi, M.: The Global Burden of Skin Disease in 2010: An Analysis of the Prevalence and Impact of Skin Conditions. The Journal of Investigative Dermatology 134, 1527–1534 (2014)
7. Hegde, C., Sankaranarayanan, A.C., Baraniuk, R.G.: Learning Manifolds in the Wild. Journal of Machine Learning Research 5037 (2012)
8. Jia, H., Wu, G., Wang, Q., Wang, Y., Kim, M., Shen, D.: Directed Graph Based Image Registration. MLMI MICCAI Workshop 7009, 175–183 (2011)
9. Kawahara, J., BenTaieb, A., Hamarneh, G.: Deep features to classify skin lesions. In: IEEE ISBI. pp. 1397–1400 (2016)
10. Kawahara, J., Hamarneh, G.: Image Content-Based Navigation of Skin Conditions. In: World Congress of Dermatology (2015)
11. Klingemann, M., Doury, S.: X Degrees of Separation (2016), `https://artsexperiments.withgoogle.com/xdegrees/`
12. Kogan, G.: Shortest path between images (2017), `https://github.com/ml4a/ml4a-guides/blob/master/notebooks/image-path.ipynb`
13. Korotkov, K., Garcia, R.: Computerized analysis of pigmented skin lesions: A review. Artificial Intelligence in Medicine 56(2), 69–90 (2012)
14. Markovic, S., Erickson, L.A., Rao, R., Creagan, E.T., et al.: Malignant Melanoma in the 21st Century, Part 1: Epidemiology, Risk Factors, Screening, Prevention, and Diagnosis. Mayo Clinic Proceedings 82(3), 364–380 (2007)
15. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015)
16. Schofield, J.K., Fleming, D., Grindlay, D., Williams, H.: Skin conditions are the commonest new reason people present to general practitioners in England and Wales. British Journal of Dermatology 165, 1044–1050 (2011)
17. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. Intl. Conf. on Learning Representations (ICLR) (2015)

Uncertainty Estimation in Vascular Networks

Markus Rempfler[1,2], Bjoern Andres[3], and Bjoern H. Menze[1,2]

[1] Institute for Advanced Study, Technical University of Munich, Germany
[2] Department of Informatics, Technical University of Munich, Germany
[3] Bosch Center for Artificial Intelligence (BCAI)

Abstract. Reconstructing vascular networks is a challenging task in medical image processing as automated methods have to deal with large variations in vessel shape and image quality. Recent methods have addressed this problem as constrained maximum a posteriori (MAP) inference in a graphical model, formulated over an overcomplete network graph. Manual control and adjustments are often desired in practice and strongly benefit from indicating the uncertainties in the reconstruction or presenting alternative solutions. In this paper, we examine two different methods to sample vessel network graphs, a perturbation and a Gibbs sampler, and thereby estimate marginals. We quantitatively validate the accuracy of the approximated marginals using true marginals, computed by enumeration.

1 Introduction

Vessel segmentation and centerline extraction is a longstanding problem in computer vision [1]. From a medical perspective, segmenting and tracking vessels is crucial for planning and guiding several types of interventions. Several recent methods, however, have focussed on reconstructing vessel network graphs [2, 3, 4, 5, 6]. Analysing vascular graphs is expected to give insights into various biological properties, e.g. the relation between vascular remodeling processes and neurological diseases or pharmaceutical treatments [7]. These methods formulate the task as MAP inference in a constrained probabilistic model over a (super-)graph of candidate vasculature, where the solution encodes the subgraph that is most likely to represent the underlying vasculature. Variations of this approach include joint-tasks such as anatomical labeling of vasculature [6] or artery-vein separation [5].

As in many applications, exploring multiple solutions or even marginal distributions would be preferable over mere point estimates – either to present local uncertainty to the end user or to pass it over to the next stage of the processing pipeline. An automated reconstruction can be inspected and, if needed, edited by an expert. In such a workflow, the controlling expert benefits from an indication of the uncertainty in the presented reconstruction (cf. Fig. 1). To this end, recent work investigated how to find the m-best diverse solutions to the MAP problem in conditional random fields (CRFs) to explore a variety of highly probable assignments [8, 9]. This approach, however, increases the computational complexity of the discrete optimization further. Alternatively, Markov

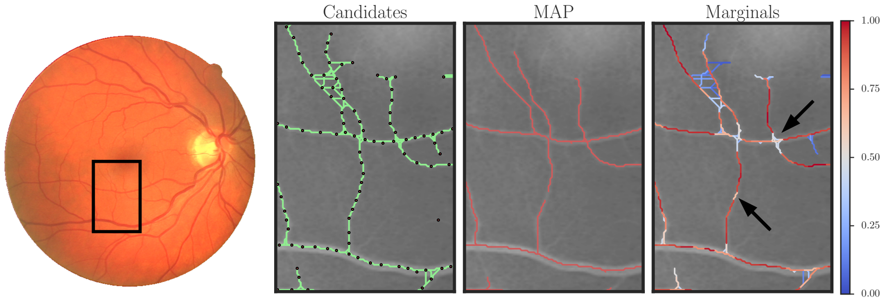


Fig. 1. Illustration of the uncertainty quantification in vasculature graphs from a 2D retinal image (**left**). Recent methods reconstruct the network from an overcomplete graph of candidate vessels (**second**, graph in green) by calculating the MAP state (**third**, graph in red) in a probabilistic model. Approximating marginal distributions (**right**) enables us to quantify the uncertainty in the network graph, which is valuable information for manual inspection and correction. Two examples are indicated with black arrows: In the first, the model is uncertain whether it is a furcation or a crossing, while in the second, a connection is not contained in the MAP but still has a high marginal probability.

chain Monte Carlo (MCMC) methods can be used to sample from probabilistic models [10, 11]. While being well established for many statistical inference tasks, they are often considered expensive and difficult to parametrize for typical problems in computer vision. Papandreou and Youille [12] presented the idea to introduce local perturbations and solve for the MAP estimate of the perturbed model repeatedly to generate samples. They identify a perturbation distribution which allows to estimate marginal densities of the original Gibbs distribution while leveraging the computational efficiency of available discrete solvers. This idea was extended to a broader problem class in [13], while the theoretical framework was further developed in [14, 15, 16, 17]. A few empirical studies investigated the effectiveness of such perturbation models in typical segmentation problems [15, 18, 19].

In this paper, we extend recent graph-based methods for reconstructing vascular networks that rely on integer progamming. We adapt two sampling approaches for the underlying probabilistic model, a perturbation sampler based on [12, 14, 15, 13] and a Gibbs sampler based on [10, 20]. They enable estimates of marginal distributions and a straight-forward way to quantify uncertainty in properties calculated from the resulting network graphs. To deal with the difficulty of validating the quality of the approximated marginals, we compare the approximated marginals to the true marginals, calculated by enumeration.

2 Background

Several recent methods for vessel network reconstruction pose the problem as MAP inference in a (constrained) probabilistic model over a supergraph composed of candidate vessels [2, 3, 4, 5, 6]. In short, such a candidate supergraph is typically constructed by detecting points that are likely to lie on a vessel centerline, composing the nodes $v \in V$ of the graph, and then inserting an edge $e \in E$ for each path that connects two nodes in close proximity. The MAP state then encodes a subgraph and thereby represents which parts of the candidate supergraph are present in the reconstruction. Calculating this MAP state can be formulated as an integer linear program (ILP) and solved by a branch-and-cut procedure. In the remainder of this section, we first describe such probabilistic model for vessel graphs and its MAP estimator. Details on the particular choice of candidate graph construction used in this study can be found in Sec. 4.

Probabilistic Model. Given a (directed) candidate graph $G = (V, E)$, we define a measure of probability $P\left(\mathbf{X} = \mathbf{x} | \Omega, I, \mathbf{\Theta}\right)$ over possible vessel networks within G, encoded by $\mathbf{x} \in \{0, 1\}^E$. These indicator variables then encode whether an edge e is present in the solution ($x_e = 1$) or not ($x_e = 0$). We denote the set of *feasible* solutions as Ω, the image evidence as I and the model parameters as $\mathbf{\Theta}$. The measure of probability can be defined as:

$$P\left(\mathbf{x} | \Omega, I, \mathbf{\Theta}\right) \propto P\left(\Omega | \mathbf{x}\right) \prod_{ij \in E} P\left(x_{ij} | I, \mathbf{\Theta}\right) \prod_{C \in \mathcal{C}(G)} P\left(x_C | \mathbf{\Theta}\right), \tag{1}$$

$$\text{where } \; P\left(\Omega | \mathbf{x}\right) \propto \begin{cases} 1 & \text{if } \; \mathbf{x} \in \Omega, \\ 0 & \text{otherwise} \end{cases} . \tag{2}$$

We identify three parts: First, $P\left(\Omega | \mathbf{x}\right)$ is the uniform prior over all feasible solutions. Second, $P\left(x_e | I, \mathbf{\Theta}\right)$ is the local evidence for an edge, i.e. the *unaries*. Third, $P\left(x_C | \mathbf{\Theta}\right)$ corresponds to joint-events C that form higher-level potentials, and $\mathcal{C}(G)$ denotes the set of all events at any possible location within G. $x_C = 1$ indicates that the particular event C occurred.

In [2, 3, 4, 5, 6], these different parts have been chosen depending on the particular image datasets and target application of the reconstructed vasculature. For this study, we will impose the following constraints: each node can have at most one incoming edge and at most two outgoing edges. Furthermore, we do not allow the solution to contain circles. These three types of constraints define our Ω. As higher-level events x_C, we consider *appearance*, *termination* and *bifurcation* in each node, leaving us with at most $3|V|$ possible events in $\mathcal{C}(G)$. These events can be represented with binary indicator variables x_C and a set of $3|V|$ auxiliary constraints that tie their state to the original edge variables $\mathbf{x}$ upon which they depend. Note that the number of involved edge variables of a particular type of event varies with its location within G: For example, a bifurcation event at node v involves all x_e of potential outgoing edges $e \in \delta^-(v)$. We denote the set of auxiliary constraints necessary for higher-level events as Ω_A in the remainder of this section. The description of both Ω and Ω_A in terms of linear inequalities can be found in the supplement.

MAP Estimator. Using the bilinear representation of the pseudo-boolean probability functions $P\left(x_{ij}|I,\mathbf{\Theta}\right)$ and $P\left(x_C|\mathbf{\Theta}\right)$, we can formulate the MAP estimator to (1) as ILP:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} w_{ij}x_{ij} + \sum_{C\in\mathcal{C}(G)} w_C x_C & (3) \\ \text{s.t.} \quad & \mathbf{x}\in\varOmega,\ \ [\mathbf{x},\mathbf{x}_C]\in\varOmega_A,\ \ x\in\{0,1\}\ \ , & (4) \end{aligned}$$

where $w_{ij} = -\log\frac{P(x_{ij}=1|I,\mathbf{\Theta})}{1-P(x_{ij}=1|I,\mathbf{\Theta})}$ and $w_C = -\log\frac{P(x_C=1|\mathbf{\Theta})}{1-P(x_C=1|\mathbf{\Theta})}$. The constraint $\mathbf{x}\in\varOmega$ is due to $P\left(\varOmega|\mathbf{x}\right)$ and $[\mathbf{x},\mathbf{x}_C]\in\varOmega_A$ ties auxiliary variables for the events to the edge variables $\mathbf{x}$. Finally, all variables are binary. This ILP can be optimized with the branch-and-cut algorithm. Certain types of constraints contained in $\varOmega$ may consist of an extensive number of inequalities (e.g. the cycle-free constraint). In this case, we employ a lazy constraint generation strategy: Whenever the solver arrives at an integral solution $\mathbf{x}'$, we check for violated constraints in the corresponding solution, add them if required and reject $\mathbf{x}'$. If no violation is found, i.e. $\mathbf{x}'$ is already a feasible solution, then it is accepted as new current solution $\mathbf{x}^*$. For our set of constraints $\varOmega$, we use this scheme for the cycle constraints, where we identify strongly connected components efficiently with [21] and add the violated constraints for the cycles within them. All other constraints for incoming and outgoing edges, as well as auxiliaries can be added to the optimization model from the start.

3 Uncertainty Estimation by Means of Sampling

3.1 Perturbation Sampler

Following the work of [12, 14, 15], a perturbation model is induced by perturbing the energy function of a random field and solving for its (perturbed) MAP state:

$$P\left(\hat{\mathbf{x}}|I,\mathbf{\Theta}\right) = P_\gamma\big(\hat{\mathbf{x}}\in\operatorname*{arg\,min}_{\mathbf{x}\in\varOmega} E(\mathbf{x};I,\mathbf{\Theta})+\gamma(\mathbf{x})\big)\ \ , \tag{5}$$

where $E(\mathbf{x},I,\mathbf{\Theta})$ is the energy function of the random field and $\gamma\left(\mathbf{x}\right)$ is the perturbation. It was shown that if the full potential table is perturbed with IID Gumbel-distributed samples of zero mean, then the perturbation model and the Gibbs model coincide [12]. In practice, this is not feasible. The full potential table may be too large and it destroys local Markov structure, rendering optimization difficult. However, it was shown in several studies that even first order Gumbel perturbations yield sufficiently good approximations [12, 15]. In this case, only the unary potentials are perturbed and hence, the perturbation $\gamma\left(\mathbf{x}\right)$ becomes:

$$\gamma\left(\mathbf{x}\right) = \sum_{i=1}^{N}\sum_{l\in\mathcal{L}}\gamma_i^l\mathbb{1}(x_i=l)\ \ , \tag{6}$$

with γ_i^k being IID samples from the Gumbel distribution [22] with zero mean and variance $\frac{\pi^2}{6}$, and $\mathbb{1}(.)$ is the indicator function. Sampling from the perturbation model then boils down to drawing a new perturbation $\gamma(\mathbf{x})$ and determining the new MAP state. Having a procedure to sample efficiently from the model enables us to estimate marginal distributions of variables (and variable subsets) as well as derived measures of uncertainty. We refer the interested reader to [12, 13, 14, 15, 16] for further information on perturbation models.

We next derive the first-order perturbed objective for the MAP estimator in (3). First, we note that two states will need two independent gumbel samples $\gamma^1_{ij}, \gamma^0_{ij}$ according to (6). Our MAP estimator, however, uses only one binary variable to encode both states. We use again the bilinear representation of the pseudo-boolean functions to find that perturbing the unaries adds a difference of the two independent gumbel samples, i.e. $\Delta\gamma_{ij} = (\gamma^1_{ij} - \gamma^0_{ij})$, to the original weight w_{ij}. The first-order perturbed objective of (3) is thus:

$$\sum_{(i,j)\in E} (w_{ij} + \Delta\gamma_{ij})x_{ij} + \sum_{C\in\mathcal{C}(G)} w_C x_C \quad . \tag{7}$$

Drawing a sample from our probabilistic model therefore boils down to constructing a new perturbed objective (with a new set of $\Delta\gamma_{ij}$) and optimizing the according ILP with the original constraints (4) and (7) instead of (3). This can be implemented by changing the coefficients of the optimization problem for each new perturbation. We note that we can warm-start the optimization with the previous solution and that we can keep previously generated constraints since they are not depending on the weights but only on the structure of G and thus, remain valid.

3.2 Gibbs Sampler

As alternative to the perturbation sampling, we employ a Gibbs sampler [10], a method of the MCMC family. We apply the following two modifications described in [20] to obtain a *metropolized* variant of the Gibbs sampler, which is expected to be more efficient for discrete problems. 1) variables are sampled in random-scan fashion within each sweep, and 2) the acceptance probability is replaced with the Metropolis-Hastings acceptance probability

$$\alpha = \min\left(1, \frac{1 - \pi(x_e|\mathbf{x}_{\setminus e})}{1 - \pi(x'_e|\mathbf{x}_{\setminus e})}\right) \quad , \tag{8}$$

where $\pi(x_e|\mathbf{x}_{\setminus e})$ and $\pi(x'_e|\mathbf{x}_{\setminus e})$ are the conditional probabilities of current and proposed state. To cope with the extra constraints of Ω, we can employ the same procedures to identify violated constraints as within the branch-and-cut algorithm. In this case, however, it suffices to check only those constraints which involve the changed variable(s). Changes that render the state infeasible with respect to Ω have a zero probability and will thus always be rejected. Auxiliary variables x_C for higher-level events need not to be sampled but can be determined directly from the current state $\mathbf{x}$ using the relationship encoded by the auxiliary constraints Ω_A. After a burn-in period of 1000 sweeps, we run one sweep for each sample.

4 Experiments & Results

We conduct our experiments on retinal images [23]. In the first part of this section, we detail on the preprocessing, i.e. the candidate vessel graph construction. In the second part, we then present both quantitative and qualitative results of the two sampling approaches. We address the difficulty of validating marginal distribution estimates by computing exact marginals on smaller problem instances, where brute-force enumeration of all states is computationally possible.

Candidate Graph Construction. As a first step, we need to propose vasculature in terms of an overcomplete candidate graph $G = (V, E)$. We rely on the following scheme to achieve this, which is mainly based on [2, 3, 24]:

1. *Centerline detection.* We compute a centerline score $f_{\text{cl}}(I)$ for the entire image using a regression approach based on [24]. High centerline scores indicate the presence of the centerline of the vessel.
2. *Candidate node selection.* We construct a collection of candidate nodes V by iteratively selecting the locations with the highest value in the centerline measure map and suppressing its neighbourhood within a radius r_{sup} until no more locations with a value larger than θ_T are left.
3. *Connection of candidates.* Next, we reconnect previously selected candidate nodes to its N closest neighbours using Dijkstra's algorithm on the centerline score map. A connection between two nodes $i, j \in V$ then forms an edge $(i, j) \in E$ in the vessel candidate graph. Connections that pass through a third candidate node are discarded as they would introduce unnecessary redundance. To save computation time, we limit the maximum search radius to r_s.

In these experiments, we set $r_{\text{sup}} = 5\,\text{px}$ and $\theta_T = 0.3 \max f_{\text{cl}}(I)$ for the candidate selection, and $N = 4$ and $r_s = 30\,\text{px}$ for the edge construction. We use a discriminative path classifier to estimate $P\left(x_{ij} = 1|I\right)$, i.e. how likely edge $ij \in E$ belongs to the graph or not, which is then used to calculate the weights w_{ij}. To this end, we use gradient boosted decision trees with 5 features calculated along the path: length, tortuosity, cumulative f_{cl}, $\min f_{\text{cl}}$ and standard deviation of f_{cl}. Additional details on both centerline regressor and path classifier can be found in the supplement. For each class of events, appearance, termination and bifurcation, we introduce one parameter θ^a, θ^t and θ^b as constant weight for the respective event happening at a given node, and set them to $\theta^a = 0.5$, $\theta^t = 0.1$ and $\theta^b = 0.1$.

Comparison. In order to quantitatively validate the marginals that we approximate by using the perturbation sampler, we set up a series of 15 small test graphs with $|E| \leq 20$ from the test images of [23], such that we are able to enumerate all feasible states and thereby obtain exact marginals by brute force. We then compare these exact marginals to the approximate marginals obtained

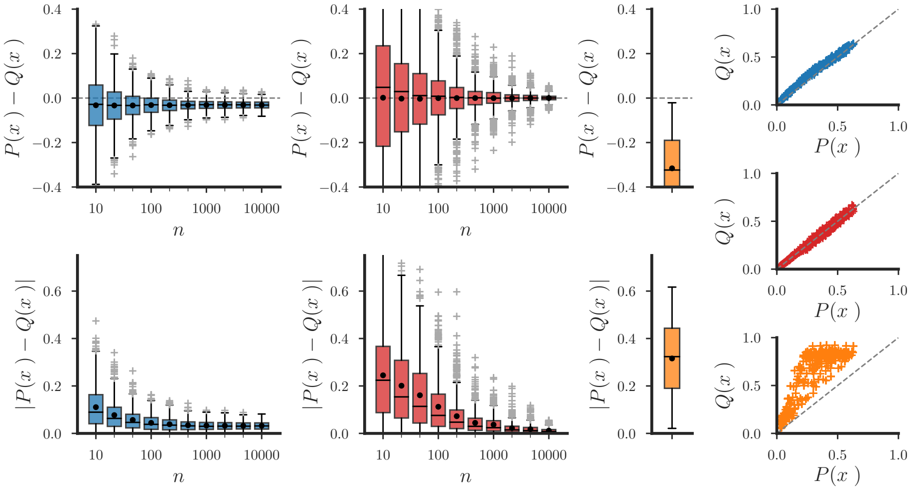


Fig. 2. Comparison of the approximated marginals $Q(x_i)$ with the exact marginals $P(x_i)$ calculated by brute force enumeration of all states. Approximates are obtained from the perturbation sampler (**blue**), from the Gibbs sampler (**red**), and from the raw classifier probability (**orange**). The figure shows deviation $P(x_i) - Q(x_i)$ (**top row**) and absolute deviation $|P(x_i) - Q(x_i)|$ (**bottom row**) of the marginal estimate with increasing number of samples n. Boxplots denote the median with a black bar, the mean value with a black dot and outliers with a grey cross. **Right column:** Scatter plot of exact marginal probabilities $P(x_i)$ versus approximated marginal probabilities $Q(x_i)$. We observe that the perturbation sampler converges to an absolute bias of about 0.032 on average and has the tendency to overestimate the marginal probabilites slightly. The Gibbs sampler does not exhibit such a systematic bias, but needs more samples to reduce its variance. Using the probabilistic output of the local classifier as an approximate to the marginals is considerably less accurate than both sampling approaches.

by both perturbation and Gibbs sampler. We solve the ILP of our MAP estimator by the branch-and-cut algorithm of [25] and implement the lazy constraint generation as callback. We use the default relative optimality gap of 10^{-4}.

In Fig. 2, we compare the approximated marginals from our perturbation sampler with exact marginals. We sample 10000 samples per case in total and repeat the experiment 5 times. We observe that the absolute deviation of the approximated from the exact marginals converges already at about 1000 samples to an absolute error of $|P(x_i) - Q(x_i)| \approx 0.032$ on average and the perturbation sampler shows a tendency to overestimate the marginal probabilities. Such a systematic bias is to be expected, as we apply a low-order perturbation instead of the (intractable) full perturbation. The Gibbs sampler does not exhibit such systematic bias, yet shows a larger variance when fewer samples are aquired. With 10000 samples, its mean absolute approximation error is 0.012 and therefore better than the perturbation sampler. Wilcoxon signed-rank tests for each

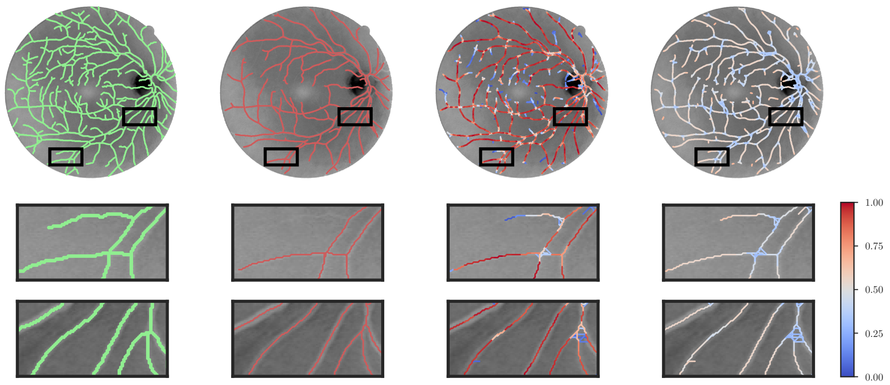


Fig. 3. Visualisation of the vascular network graphs overlaid on the (grey-scale) input image. **From left to right:** Pixel-based centerline obtained by skeletonizing the ground truth segmentation, MAP reconstruction, approximated marginals using the perturbation sampler, and the Gibbs sampler. The colorbar applies only to the marginals of two right columns, where we show the marginal $P(x_{ij} = 1 \vee x_{ji} = 1)$, i.e. the probability of either edge being active, for better visibility. We find that the marginals of the perturbation sampler indicate uncertainty in small bifurcations and point out the (possible) presence of weak terminal branches, which would be discarded if we only consider the MAP solution. The Gibbs sampler displays overall a higher uncertainty on such large graphs.

fixed number of samples n indicate that the approximation errors of perturbation and Gibbs sampler are significantly different ($p < 0.001$), with the exception of $n = 1000$ where both show similar errors. Using the probabilities of the path classifier directly as an approximate marginal is considerably worse than both sampling approaches. Note that the exact marginals for our test cases do not exhibit very high values (cf. Fig. 2, right column) due to the fact that for these small graphs, often no direction is strongly dominating and thus, several solutions that contain similar physical paths but in different orientations are competing.

A qualitative visualisation of the approximated marginals on complete graphs is given in Fig. 3. We draw 100 perturbation samples, which we found a reasonable trade-off between computation time and informativeness of the marginals, and slightly increase the relative optimality gap to $5 \cdot 10^{-3}$ to prevent the branch-and-cut solver from spending too much time proofing optimality. From the Gibbs sampler, we draw 10000 samples after a burn-in period of 1000. We find that the marginals from the Gibbs sampler display overall a higher uncertainty in the graph than the perturbation samples, which could be due to more difficult transitions between different modi of the distribution and would likely require adapted sampling parametrization or even an extension of the set of allowed transformations. In both cases, thresholding the marginal distributions $P(x_e)$ has no guarantee to satisfy all constraints and is therefore not recommended for obtaining a single reconstruction. To improve a reconstruction, an interactive procedure using the uncertainties (and individual samples) would be advisable, and for downstream analysis, metrics of interest should be calculated on each sample. Regarding computation time, the average runtime per sample is 7.85 s for the perturbation and approximately 0.01 s for the Gibbs sampler (not including any additional overhead caused by the burn in period). The perturbation sampler spends on average 0.5 % of its runtime in the lazy constraint generation where violated cycle inequalities are identified.

5 Conclusion

We adapted two sampling approaches for vascular network graph reconstruction models, a perturbation sampler and a Gibbs sampler. Our experiments confirm the expected systematic bias of the perturbation sampler due to the computationally cheaper low-order perturbations. The Gibbs sampler, on the other hand, exhibits an unbiased behaviour but instances with varying properties might require an appropriately adapted parametrization. The perturbation approach benefits from not having a burn in period, which renders it considerably easier to use on large instances. Both approaches were shown to be more informative than the predictive probabilities of local classifier and can be used to approximate marginals or determine the uncertainty in network graph properties. Beyond this, the two sampling procedures could be employed within a Bayesian model selection framework or for maximum-likelihood hyperparameter estimation.

Acknowledgements. With the support of the Technische Universität München – Institute for Advanced Study, funded by the German Excellence Initiative (and the European Union Seventh Framework Programme under grant agreement n° 291763).

References

1. Lesage, D., Angelini, E., Bloch, I., Funka-Lea, G.: A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. Medical Image Analysis 13(6), 819–845 (2009)
2. Türetken, E., Benmansour, F., Andres, B., Glowacki, P., Pfister, H., Fua, P.: Reconstructing curvilinear networks using path classifiers and integer programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(12), 2515–2530 (2016)
3. Rempfler, M., Schneider, M., Ielacqua, G.D., Xiao, X., Stock, S.R., Klohs, J., Székely, G., Andres, B., Menze, B.H.: Reconstructing cerebrovascular networks under local physiological constraints by integer programming. Medical Image Analysis 25(1), 86 – 94 (2015)
4. Rempfler, M., Andres, B., Menze, B.H.: The minimum cost connected subgraph problem in medical image analysis. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III. pp. 397–405. Springer International Publishing, Cham (2016)

5. Payer, C., Pienn, M., Blint, Z., Shekhovtsov, A., Talakic, E., Nagy, E., Olschewski, A., Olschewski, H., Urschler, M.: Automated integer programming based separation of arteries and veins from thoracic ct images. Medical Image Analysis 34, 109 – 122 (2016)
6. Robben, D., Türetken, E., Sunaert, S., Thijs, V., Wilms, G., Fua, P., Maes, F., Suetens, P.: Simultaneous segmentation and anatomical labeling of the cerebral vasculature. Medical Image Analysis 32, 201–215 (2016)
7. Klohs, J., Baltes, C., Princz-Kranz, F., Ratering, D., Nitsch, R.M., Knuesel, I., Rudin, M.: Contrast-enhanced magnetic resonance microangiography reveals remodeling of the cerebral microvasculature in transgenic arcaβ mice. Journal of Neuroscience 32(5), 1705–1713 (2012)
8. Batra, D., Yadollahpour, P., Guzman-Rivera, A., Shakhnarovich, G.: Diverse m-best solutions in Markov Random Fields. In: ECCV 2012, LNCS, vol. 7576, pp. 1–16. Springer Berlin Heidelberg (2012)
9. Kirillov, A., Savchynskyy, B., Schlesinger, D., Vetrov, D., Rother, C.: Inferring m-best diverse labelings in a single one. In: IEEE International Conference on Computer Vision (ICCV). pp. 1814–1822 (2015)
10. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6(6), 721–741 (1984)
11. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
12. Papandreou, G., Yuille, A.L.: Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In: International Conference on Computer Vision 2011. pp. 193–200 (2011)
13. Tarlow, D., Adams, R.P., Zemel, R.S.: Randomized optimum models for structured prediction. In: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, vol. 22, pp. 1221–1229 (2012)
14. Hazan, T., Jaakkola, T.: On the partition function and random maximum a-posteriori perturbations. In: Proceedings of the 29th International Conference on Machine Learning (ICML-12), pp. 991–998 (2012)
15. Hazan, T., Maji, S., Jaakkola, T.: On sampling from the gibbs distribution with random maximum a-posteriori perturbations. Advances in Neural Information Processing Systems pp. 1268–1276 (2013)
16. Orabona, F., Hazan, T., Sarwate, A., Jaakkola, T.: On measure concentration of random maximum a-posteriori perturbations. In: International Conference on Machine Learning. pp. 432–440 (2014)
17. Gane, A., Hazan, T., Jaakkola, T.: Learning with maximum a-posteriori perturbation models. In: Artificial Intelligence and Statistics. pp. 247–256 (2014)
18. Alberts, E., Rempfler, M., Alber, G., Huber, T., Kirschke, J., Zimmer, C., Menze, B.H.: Uncertainty quantification in brain tumor segmentation using CRFs and random perturbation models. In: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). pp. 428–431 (2016)
19. Meier, R., Knecht, U., Jungo, A., Wiest, R., Reyes, M.: Perturb-and-MPM: Quantifying segmentation uncertainty in dense multi-label CRFs. CoRR abs/1703.00312 (2017), http://arxiv.org/abs/1703.00312
20. Liu, J.S.: Monte Carlo strategies in scientific computing. New York: Springer (2001)
21. Mehlhorn, K., Näher, S., Sanders, P.: Engineering DFS-based graph algorithms. CoRR abs/1703.10023 (2017), http://arxiv.org/abs/1703.10023
22. Gumbel, E.J.: Statistical theory of extreme values and some practical applications: a series of lectures. No. 33, US Govt. Print. Office (1954)

23. Staal, J.J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging 23(4), 501–509 (2004)
24. Sironi, A., Türetken, E., Lepetit, V., Fua, P.: Multiscale centerline detection. IEEE Transactions on Pattern Analysis & Machine Intelligence 1, 1–14 (2015)
25. Gurobi Optimization, I.: Gurobi Optimizer Reference Manual (2017), http://www.gurobi.com

Extraction of Airways with Probabilistic State-space Models and Bayesian Smoothing

Raghavendra Selvan[1], Jens Petersen[1], Jesper H. Pedersen[2] and Marleen de Bruijne[1,3]

[1] Department of Computer Science, University of Copenhagen, Denmark
[2] Department of Cardio-Thoracic Surgery RT, Rigshospitalet, University Hospital of Copenhagen, Denmark
[3] Departments of Medical Informatics and Radiology, Erasmus MC, The Netherlands
raghav@di.ku.dk

Abstract. Segmenting tree structures is common in several image processing applications. In medical image analysis, reliable segmentations of airways, vessels, neurons and other tree structures can enable important clinical applications. We present a framework for tracking tree structures comprising of elongated branches using probabilistic state-space models and Bayesian smoothing. Unlike most existing methods that proceed with sequential tracking of branches, we present an exploratory method, that is less sensitive to local anomalies in the data due to acquisition noise and/or interfering structures. The evolution of individual branches is modelled using a process model and the observed data is incorporated into the update step of the Bayesian smoother using a measurement model that is based on a multi-scale blob detector. Bayesian smoothing is performed using the RTS (Rauch-Tung-Striebel) smoother, which provides Gaussian density estimates of branch states at each tracking step. We select likely branch seed points automatically based on the response of the blob detection and track from all such seed points using the RTS smoother. We use covariance of the marginal posterior density estimated for each branch to discriminate false positive and true positive branches. The method is evaluated on 3D chest CT scans to track airways. We show that the presented method results in additional branches compared to a baseline method based on region growing on probability images.

Keywords: Probabilistic state-space, Bayesian Smoothing, Tree Segmentation, Airways, CT

1 Introduction

Segmentation of tree structures comprising of vessels, neurons, airways etc. are useful in extraction of clinically relevant biomarkers [1,2]. The task of extracting trees, mainly in relation to vessel segmentation, has been studied widely using different methods. A successful class of these methods are based on techniques from target tracking. Perhaps the most used tracking strategy is to proceed from an initial seed point, make local-model fits to track individual branches in a sequential manner and perform regular branching checks [3,4]. Such methods are prone to local anomalies and can prematurely terminate if occlusions are encountered. The method in [3] can overcome such problems to a certain extent using a deterministic multiple hypothesis testing approach; however, it is a semi-automatic method requiring extensive manual intervention and can be computationally expensive. In [4], vessel tracking on 2D retinal scans is performed using a Kalman filter. They propose an automatic seed point detection strategy using a matched filter. From each of these seed points vessel branches are progressively tracked using measurements that are derived from the image data. A gradient based measurement function is employed which fails in low-contrast regions of the image, which are predominantly regions with thin vessels. Another major class of tracking algorithms are based on a stochastic formulation of tracking [5,6] using some variation of particle filtering. Particle filter-based methods are known to scale poorly with dimensions of the state space [1].

In spirit, we propose an exploratory method like particle filter-based methods, with a salient distinction that the proposed method can track branches from several seed points across the volume. We use linear Bayesian smoothing to estimate branch states, described using Gaussian densities. Thus, the method inherently provides an uncertainty measure, which we use to discriminate true and false positive branches. Further, unlike particle filter-based methods, the proposed method is fast, as Bayesian smoothing is implemented using the RTS (Rauch-Tung-Striebel) smoother [7] involving only a set of linear equations.

2 Method

We formulate tracking of branches in tree structures using probabilistic state-space models, commonly used in target tracking and control theory [7]. The proposed method takes image data as input and outputs a collection of disconnected branches that taken together forms the tree structure of interest. We first process the image data to obtain a sequence of measurements and track all possible branches individually using Bayesian smoothing. We then use covariance estimates of individual branches to output a subset of the most likely branches yielding the tree structure of interest. Details of this process are described below.

2.1 Tracking individual branches

We assume the tree structure of interest, $\mathbf{X}$, to be a collection of T independent random variables $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_T\}$, where individual branches are denoted $\mathbf{X}_i$. Each branch $\mathbf{X}_i$ of length L_i is treated as a sequence of states, $\mathbf{X}_i = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{L_i}]$. These states are assumed to obey a first-order Markov assumption, i.e.,

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \dots, \mathbf{x}_0) = p(\mathbf{x}_k|\mathbf{x}_{k-1}). \quad (1)$$

The state vector has seven random variables,

$$\mathbf{x}_k = [x, y, z, r, v_x, v_y, v_z]^T, \quad (2)$$

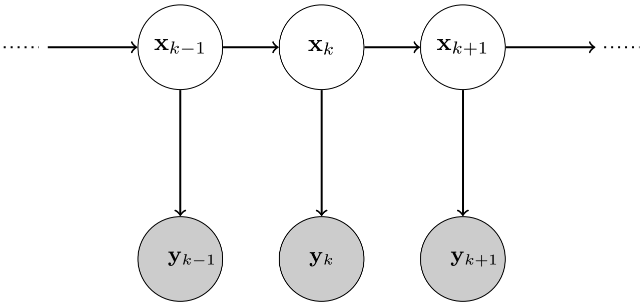


Fig. 1: Bayesian network view of the relation between the underlying true states, $\mathbf{x}_i$, and the measurements, $\mathbf{y}_i$, for a single branch.

describing a tubular segment centered at Euclidean coordinates $[x, y, z]$, along an axis given by the direction vector $[v_x, v_y, v_z]$ with radius r.

The observed data, image $\mathbf{I}$, is processed to be available as a sequence of vectors. We model the measurements as four dimensional state vectors consisting only of position and radius. This is accomplished using a multi-scale blob detector [8]. The input image $\mathbf{I}$ with N_v voxels is transformed into a sequence of N measurements, with position and radius information, denoted $\mathbf{Y} = [\mathbf{y}_0, \ldots, \mathbf{y}_N]$, where each $\mathbf{y}_i = [x, y, z, r]^T$. This procedure applied to the application of tracking airway trees is described in Section 2.5.

2.2 Process and Measurement Models

Transition from one tracking step to another within a branch is modelled using the process model. We use a process model that captures our understanding of how individual branches evolve between tracking steps and has similarities with the model used in [4]. We assume first-order Markov independence in state transitions from (1), captured in the process model below:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{q} = \begin{bmatrix} 1 & 0 & 0 & 0 & \mathit{\Delta} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \mathit{\Delta} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \mathit{\Delta} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \\ r_{k-1} \\ v_{x\,k-1} \\ v_{y\,k-1} \\ v_{z\,k-1} \end{bmatrix} + \mathbf{q} \tag{3}$$

where $\mathbf{F}$ is the process model function and $\mathbf{q}$ is the process noise. $\mathbf{q}$ is assumed to be a zero mean Gaussian density, i.e, $\mathbf{q} \sim N(\mathbf{0}, \mathbf{Q})$, with process covariance, $\mathbf{Q}_{7\times7}$, acting only on direction and radius components of the state vector,

$$\mathbf{Q}_{[4:7,4:7]} = \sigma_q^2 \mathit{\Delta} \times \mathbf{I}_{4\times4}, \tag{4}$$

where only the non-zero part of the matrix is shown and σ_q^2 is the process variance. The parameter $\mathit{\Delta}$ can be seen as step size between tracking steps. As (3) is a recursion, the initial point (seed point), $\mathbf{x}_0$, comprising of position, scale and orientation information is provided to the model. Seed points are assumed to be described by Gaussian densities, $\mathbf{x}_0 \sim N(\hat{\mathbf{x}_0}, \mathbf{P}_0)$, with mean $\hat{\mathbf{x}_0}$ and covariance $\mathbf{P}_0$. We present an automatic strategy to detect such initial seed points in 2.5.

The measurement model describes the relation between each of the 4-D measurements, $\mathbf{y}_k$ in the sequence, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, and the state vector, $\mathbf{x}_k$, as shown in Figure 1. A simple linear measurement model captures this relation,

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{m} = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \\ r_k \\ v_{x_k} \\ v_{y_k} \\ v_{z_k} \end{bmatrix} + \mathbf{m} \tag{5}$$

where $\mathbf{y}_k$ are observations generated by true states of the underlying branch at step k, $\mathbf{H}$ is the measurement function. $\mathbf{m} \sim N(\mathbf{0}, \mathbf{R})$ is the measurement noise with covariance $\mathbf{R}$ that is a diagonal matrix with entries, $[\sigma^2_{m_x}, \sigma^2_{m_y}, \sigma^2_{m_z}, \sigma^2_{m_r}]$, which correspond to variance in the observed position and radius, respectively. All possible measurement vectors obtained from the image are aggregated into the measurement variable $\mathbf{Y}$.

2.3 Bayesian Smoothing

The state-space models presented above enable us to estimate branches using the posterior distributions, $p(\mathbf{X}_i|\mathbf{Y})\forall i = [0, \ldots, T]$, using standard Bayesian methods. We employ Bayesian smoothing as all the measurements are available at once, when compared to sequential observations that are more common in object tracking applications. Due to a linear, Gaussian process and measurement models, Bayesian smoothing can be optimally performed using the RTS smoother [7]. RTS smoother uses two Bayesian filters to perform forward filtering and backward smoothing. Forward filtering is identical to performing Kalman filtering and consists of sequential prediction and update with observed information of the state variable. Once a branch is estimated using forward filtering, the saved states are used to perform backward smoothing using a Kalman-like filter which improves state estimates by incorporating additional information from future steps. Standard equations for an RTS smoother are presented below [7].

Forward Filtering Equations in the first column of Table 1 are used to perform prediction and update steps of the forward filtering. In the prediction step, process model is used to predict states at the next step. Mean $\hat{\mathbf{x}}_{k|k-1}$ and covariance $\mathbf{P}_{k|k-1}$ estimates of the predicted Gaussian density, i.e, of state k conditioned on the previous state, denoted with subscript $k|k-1$, are computed in (6),(7). In the update step, described in (8) − (12), predicted density is associated with a measurement vector to obtain posterior density. First, the new information from

Table 1: Standard RTS Smoother Equations

Forward Filtering

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} \tag{6}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \tag{7}$$

$$\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \tag{8}$$

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \tag{9}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1} \tag{10}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k \tag{11}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{12}$$

Backward Smoothing

$$\mathbf{G}_k = \mathbf{P}_{k|k}\mathbf{F}^T\mathbf{P}_{k+1|k}^{-1} \tag{13}$$

$$\hat{\mathbf{x}}_{k|L} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\hat{\mathbf{x}}_{k+1|L} - \hat{\mathbf{x}}_{k+1|k}) \tag{14}$$

$$\mathbf{P}_{k|L} = \mathbf{P}_{k|k} - \mathbf{G}_k(\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|L})\mathbf{G}^T \tag{15}$$

measurement $\mathbf{y}_k$ is computed using (8) and is aptly called the "innovation", denoted as $\mathbf{v}_k$. Uncertainty in the new information, innovation covariance $\mathbf{S}_k$, is computed in (9). Then, predicted mean is adjusted with weighted innovation and predicted covariance is adjusted with weighted innovation covariance to obtain the posterior mean and covariances, in (11) and (12), respectively. The weighting computed in (10), denoted as $\mathbf{K}_k$, is the Kalman gain which controls the extent of information fusion from process and measurement models.

We continue estimation of the posterior density (described by posterior mean and covariance) in a sequential manner for the branch until no new measurements exist for updating. After the final update step, a sequence of posterior mean estimates $[\hat{\mathbf{x}}_{0|0}, \ldots, \hat{\mathbf{x}}_{L_i|L_i}]$ and posterior covariance estimates $[\mathbf{P}_{0|0}, \ldots, \mathbf{P}_{L_i|L_i}]$, obtained from the forward filter are saved, for further use by the backward smoother.

Backward smoothing The smoothed estimates are obtained by running a backward filter starting from the final tracked state of the forward filter. The intuition behind backward smoothing is that the uncertainty in making predictions in the forward filtering can be alleviated using information from future steps. It is implemented using the equations in the second column of Table 1.

Gating When performing the RTS smoother recursions, the forward filter expects a single measurement vector for the update step. We employ rectangular and ellipsoidal gating to reduce the number of measurements handled during the update step [9].

First, we perform simple rectangular gating which is based on excluding measurements that are outside a rectangular region around the predicted measurement $\mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ in equation (8) using the following condition:

$$|\mathbf{y}_i - \mathbf{H}\mathbf{x}_{k|k-1}| \leq \kappa \times \operatorname{diag}(\mathbf{S}_k), \forall \mathbf{y}_i \in \mathbf{Y} \tag{16}$$

where $\mathbf{S}_k$ is the covariance of the predicted measurement in equation (9). The rectangular gating coefficient, κ, is usually set to a value ≥ 3 [9]. Rectangular gating localises the number of candidate measurements relevant to the current tracking step. To further narrow down on the best candidate measurement for update, we follow rectangular gating with ellipsoidal gating [9]. With ellipsoidal gating we accept the measurements within the ellipsoidal region of the predicated covariance, using the following rule:

$$(\mathbf{H}\mathbf{x}_{k|k-1} - \mathbf{y}_i)^T \mathbf{S}_k^{-1} (\mathbf{H}\mathbf{x}_{k|k-1} - \mathbf{y}_i) \leq G \tag{17}$$

where G is the rectangular gating threshold, obtained from the gating probability P_g, which is the probability of observing the measurement within the ellipsoidal gate,

$$P_g = 1 - \exp\Big(-\frac{G}{2}\Big). \tag{18}$$

2.4 Tree as a Collection of Branches

Once a branch is smoothed and saved using Bayesian smoothing described previously, we process new seed points and start tracking branches until no further seed points remain to track from. This procedure yields a collection of disconnected branches. The next task is to obtain a subset of likely branches that represent the tree structure of interest by discarding false positive branches.

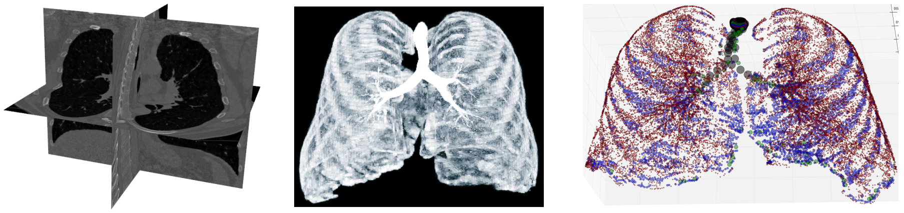
Validation of Tracked Branches An advantage of using Bayesian smoothing to track individual branches is that apart from estimating the branch states from the image data (using the smoothed posterior mean estimates), we can also quantify the uncertainty of the estimation at each tracking step (using the smoothed posterior covariance estimates). Thus, we have the possibility of aggregating this uncertainty over the entire branch to validate them. We explore this notion to create a criterion for accepting or rejecting branches.

By aggregating variance for all tracking steps in each branch, we obtain a measure of the quality of branches. A straightforward approach is to use total variance, obtained using the trace of each of the smoothed posterior covariance matrices. We average the sum total variance over the length of each branch, l_i, to obtain a score, μ_i, which is then thresholded by a cut-off μ_c to qualify the branches,

$$\mu_i = \frac{\sum_{k=1}^{l_i} \mathrm{Tr}(\mathbf{P}_{k|k})}{l_i}. \tag{19}$$

2.5 Application to Airways

The proposed method for tracking tree structures can be applied to track airways, vessels or other tree structures encountered in image processing applications. We focus on tracking airways from lung CT data and present the specific strategies used to implement the proposed method.



(a) Intensity image (b) Probability image (c) Multi-scale blob image

Fig. 2: The pipeline of image representations, ultimately showing the multi-scale representation.

Multi-scale representation The measurement model discussed in Section 2.2 assumes a 4-D state vector as measurements to the RTS smoother. This is achieved by first computing an airway probability image using a k-Nearest Neighbour voxel classifier trained to discriminate between airway and background, described in [11]. Blob detection with automatic scale selection [8] for different scales, $\sigma_s = (1, 2, 4, 8, 12)mm$, is performed on the probability image to obtain the 4D state measurements as blob position and radius. Indistinct blobs are removed if the absolute value of the normalized response at the selected scale, σ_s^*, is less than a threshold [8]. This makes the representation sparse, $N << N_v$, and the tracking more efficient than if performed at voxel-level. An example of the sparse representation can be found in Figure 2.

Initialisation of Branches The multi-scale representation of the image data discussed above also provides a response corresponding to the best scale. As this response is normalised for scales, we incorporate this information in selecting the initial seed point for every branch. We start tracking from the seed point with the largest scale and the largest response. The initial direction information is obtained from eigen value analysis of the Hessian matrix computed at the corresponding scale provided in the measurement vector. Once a branch is tracked along the initial direction, we track from the same seed point but in the opposite direction. Thus, if a seed point is obtained from the middle of a branch we can track it bidirectionally. After tracking in both directions, all the involved measurements including the seed point are removed from the measurement vector, and the next best candidate seed point is chosen and tracking commences from there. The tracking procedure on the entire image is complete when no more seed points are available.

3 Experiments and Results

3.1 Data

The evaluation was carried out on 32 low-dose CT chest scans from a lung cancer screening trial [10]. Training and test sets comprising of 16 images each were randomly obtained from the data set. All scans have a resolution of approximately

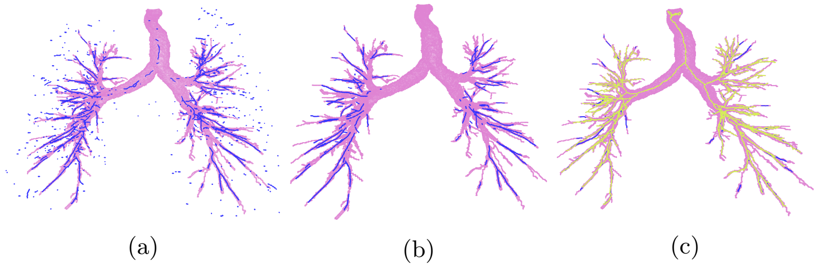


Fig. 3: Visualisation of the centerlines extracted using the proposed method before and after thresholding to discard false positive branches overlaid on the reference segmentation, shown in (a) and (b) respectively. The combined results from the proposed method and region growing on probability is shown as the blue centerline in (c).

1mm × 0.78mm × 0.78mm. The reference segmentations consist of expert verified union over the results of two previous methods [11,12]. The proposed method is compared with region growing on the probability images.

3.2 Error Measure, Initial Parameters and Tuning

We use an error measure defined as the average of two distances, $d_{err} = (d_{FP} + d_{FN})/2$. The first distance, d_{FP}, captures the false positive error and is the average minimum Euclidean distance from segmentation centerline points to reference centerline points. d_{FN} similarly defines the false negative error, as the average minimum Euclidean distance from reference centerlines points to segmentation centerline points.

There are several parameters related to the RTS smoother that need to be initialised. These parameters were tuned using the training set and fixed for the evaluation on the test set to: standard deviations of the process noise, $\sigma_q = 0.3$, measurement noise on radius $\sigma_{m_r} = 1$ mm and measurement noise on position $(\sigma_{m_x}, \sigma_{m_y}, \sigma_{m_z}) = 2$ mm. The initial covariance, $\mathbf{P}_0$ across branches was set to $\mathbf{I}_{7\times 7}$. The most crucial parameter in the proposed method is the threshold parameter μ_c presented in Section 2.4. The threshold to validate branches is tuned to be $\mu_c = 2.0$. The gating probability was set to a high value, $P_g = 0.99$ [9].

3.3 Results

Figure 3 illustrates features of the proposed method by visualising centerlines overlaid on the reference segmentation. Influence of the threshold parameter μ_c is illustrated with the segmentation results for a single volume without any threshold (seen in Figure 3a) and after applying the tuned threshold (seen in Figure 3b). Evidently, thresholding the average total variance of a branch eliminates false positive branches.

The final output obtained from the method is a collection of disconnected branches. While such collection of branches are still useful in extracting biomarkers, for evaluation purposes we merge the results obtained with the segmentations from region growing on probability images and extract centerlines from the merged segmentation using 3D thinning, as seen in Figure 3a and 3b. This also allows us to demonstrate the improvement our method provides by extracting peripheral airway branches, which are typically the challenging ones. One such combined result is shown in Figure 3c, where the yellow centerlines correspond to region growing and blue one is the combined result.

Table 2: Performance comparison on the test set

Method	d_{FP}(mm)	d_{FN}(mm)	d_{err} (mm)	Std.Dev. (mm)
RG	0.423	3.579	2.001	0.208
(RTS+RG)$_1$	0.449	2.102	1.276	0.187
(RTS+RG)$_2$	0.401	2.658	1.529	0.165

Performance on the test set for two different scenarios of the proposed method is reported in Table 2 along with the numbers for region growing on probability images. The result for the best performing region growing on probability images is denoted with RG and those obtained by combining the proposed method with region growing are denoted as RTS+RG. We first combine the proposed method with the best performing region growing case (with minimum d_{err}) results and it is denoted as (RG+RTS)$_1$. We observe an improvement of about 36% on d_{err}. It is to be noted, there is substantial reduction in d_{FN}, indicating that many branches missed by region growing are now segmented. There is a very small increase in false positives which could also be due to the missing branches in the reference segmentation; however, the net result is a large improvement. To test whether the proposed method can simultaneously reduce the number of false positives and false negatives compared to region growing, we merge the proposed method with the region growing result that yields non-optimal d_{err}, and do observe a reduction in both d_{FP} and d_{FN} when compared to the best performing RG as seen in the entries for (RG+RTS)$_2$.

The computational expense for running the proposed method is small. The largest chunk of it is used in generating the multi-scale representation of the images, which is in the range of 10-15s per volume. Tracking using the RTS smoother and obtaining the segmentation takes about 4s on a laptop with 8 cores and 32 GB memory running Debian operating system.

4 Discussion and Conclusions

We presented an automatic method for tracking tree structures, in particular airways, using probabilistic state-space models and Bayesian smoothing. We demonstrated that branches can be tracked individually from across the volume, starting from several seed points. This approach of tracking branches from across the volume has the advantage that even in the presence of occlusions, such as mucous plugging or image acquisition noise, the chances of detecting branches beyond the occlusions are higher. An inherent measure of uncertainty in the branch estimates has been presented due to the Bayesian nature of the method. We demonstrated the use of thresholding this uncertainty measure to discriminate detected branches. The use of sparse representation of voxels in the image using blob detection makes the method computationally efficient.

A possible limitation with the proposed method is that it yields a disconnected tree structure. For applications where this is an issue, one can enforce a global connectivity constraint on the disconnected set of branches to obtain fully connected tree as done in [13] or similar. It is also possible to derive biomarkers directly from the disconnected branches, as shown in [14].

We performed an evaluation of the results obtained from the proposed method by combining it with the results from region growing on probability images. We showed that there is substantial improvement in the segmentation results, indicating that the exploratory approach taken up in our method has potential in improving tree segmentations.

5 Acknowledgements

This work was funded by the Independent Research Fund Denmark (DFF) and Netherlands Organisation for Scientific Research (NWO).

References

1. Lesage D, et.al. A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. Medical image analysis. 2009 Dec 31;13(6):819-45.
2. Lo P, et.al. Extraction of airways from CT (EXACT'09). IEEE Transactions on Medical Imaging. 2012 Nov;31(11):2093-107
3. Friman O, et.al. Multiple hypothesis template tracking of small 3D vessel structures. Medical image analysis. 2010 Apr 30;14(2):160-71.
4. Yedidya T, et.al. Tracking of blood vessels in retinal images using Kalman filter. InComputing: Techniques and Applications. Digital Image 2008 (pp. 52-58).
5. Florin, C., et.al. Particle filters, a quasi-monte carlo solution for segmentation of coronaries. MICCAI. 2005 (pp. 246-253). Springer Berlin Heidelberg.
6. Lesage, D., et.al. Adaptive particle filtering for coronary artery segmentation from 3D CT angiograms. Computer Vision and Image Understanding. 2016.151, pp.29-46
7. Särkkä, Simo. Bayesian filtering and smoothing. Cambridge University Press, 2013.
8. Lindeberg, Tony. Feature detection with automatic scale selection. International journal of computer vision 30.2. 1998.
9. Bar-Shalom Y, Willett PK, Tian X: Tracking and data fusion. YBS publishing; 2011
10. Pedersen, Jesper H et. al. The Danish randomized lung cancer CT screening trial-overall design and results of the prevalence round, Journal of Thoracic Oncology, 2009.
11. Lo, Pechin, et.al. Vessel-guided airway segmentation based on voxel classification. First International Workshop on Pulmonary Image Analysis. MICCAI. 2008
12. Lo, Pechin, et.al. Airway tree extraction with locally optimal paths. MICCAI. 2009
13. Graham, Michael W., et al. Robust 3-D airway tree segmentation for image-guided peripheral bronchoscopy. IEEE transactions on Medical Imaging (2010)
14. Sørensen, Lauge, et al. Dissimilarity-based classification of anatomical tree structures. Information Processing in Medical Imaging. Springer Berlin/Heidelberg, 2011.

Detection and Localization of Landmarks in the Lower Extremities Using an Automatically Learned Conditional Random Field

Alexander Oliver Mader,[1,2,3] Cristian Lorenz,[3] Martin Bergtholdt,[3]
Jens von Berg,[3] Hauke Schramm,[1,2] Jan Modersitzki,[4] and Carsten Meyer[1,2,3]

[1] Institute of Computer Science, Kiel University of Applied Sciences, Germany
[2] Department of Computer Science, Faculty of Engineering, Kiel University, Germany
[3] Department of Digital Imaging, Philips Research Hamburg, Germany
[4] Institute of Mathematics and Image Computing, Lübeck University, Germany
alexander.o.mader@fh-kiel.de

Abstract. The detection and localization of single or multiple landmarks is a crucial task in medical imaging. It is often required as initialization for other tasks like segmentation or registration. A common approach to localize multiple landmarks is to exploit their spatial correlations, e.g., by using a conditional random field (CRF) to incorporate geometric information between landmark pairs. This CRF is usually applied to resolve ambiguities of a localizer, e.g., a random forest or a deep neural network. In this paper, we apply a random forest / CRF combination to the task of jointly detecting and localizing 6 landmarks in the lower extremities, taken from a dataset of 660 X-ray images. The dataset is challenging since a significant number of images does not show all the landmarks. Furthermore, 11.3 % of the target landmarks are altered by prostheses or pathologies.

To account for this, we introduce a "missing" label for each landmark (represented by a node in the CRF). Moreover, instead of manually specifying the CRF model by selecting suitable potential functions and the graph topology, we suggest to automatically optimize both in a learning framework. Specifically, we define a pool of potential functions and learn their CRF weights (relative contributions), in addition to the potential values in case of missing landmarks. Potentials with a low weight are removed, thus optimizing the graph topology. Detailed evaluations on our database show the feasibility of our approach. Our algorithm removed on average 23 of the initial 51 CRF potentials, and correctly detected and localized (within 10 mm tolerance) on average 92.8 % of the landmarks, with individual rates ranging from 90.0 % to 97.4 %.

1 Introduction

The automatic localization of landmarks in medical images is a crucial task. It is clinically required, inter alia, for the purposes of diagnosis, surgical planning, and post-operative assessment. Because of the large amount of variability and outliers in medical data, the automatic and accurate localization of landmarks is comparably hard. It becomes harder, when a landmark's presence is not guaranteed (e.g., due to a restricted field of view). In this case, each landmark has to be detected before it can be localized.

Many approaches have been proposed to solve the task of localizing spatially correlated landmarks. Often, first a “landmark localizer” is used to generate a (pseudo) probability map for each landmark over the image domain. To this end, e.g., random forests [6, 7, 14, 18], decision trees [2], deep convolutional neural networks [15], and the discriminative generalized Hough transform [16] have been used. Then, a conditional random field (CRF) is often applied to select the globally optimal configuration for all landmarks, characterized by the largest joint posterior probability [2,6,14,18]. The posterior probability of the CRF is generally expressed by an energy, which is parameterized by potential functions (often unary or binary). The unary potentials of the CRF are related to the locations of individual landmarks and are defined based on the localizer output. Binary potentials model the spatial relations between two landmarks, assuming a specific topology (i.e., graph connectivity). Both the potentials (e.g., distance [2], vector [6], vector field profiles [5], etc.) and the topology are often selected in a heuristic manner. Potentials of higher arity are possible, but seldomly used due to computational complexity [12,19]. Only few papers explicitly learn the weights of the CRF potentials, i.e., their relative contribution to the joint posterior probability, let alone address the possibility of missing landmarks due to, e.g., a restricted field of view. Among them [2], which includes a heuristic penalty for a false miss. To compute the globally optimal landmark configuration based on the CRF model, various inference algorithms [19] can be applied.

In this paper, we automatically learn essential components of a CRF – including the possibility of missing landmarks – to automatically detect (i.e., determine whether a landmark is present in the current image) and localize (i.e., specify the position of a landmark present in the current image) six landmarks of the lower extremities in a database of 660 X-ray images with significant fractions of missing landmarks due to restricted field of view. Specifically, we define a pool of potential functions, the weights of which are – together with the values of potentials in case of missing landmarks – automatically learned. Starting from a fully connected graph, potentials with low weights are removed. In this way, the graph topology can be automatically optimized. Applying our method, on average 23 of the initially 51 CRF potentials were removed, and (on average) 92.8 % of the landmarks were correctly detected and (if present) localized within 10 mm tolerance.

2 Related Work

Various approaches have been proposed to detect and localize a set of landmarks. Here, we briefly summarize the contributions that are closest to our work. Random forests have been used to generate landmark localization hypotheses, e.g., in [6, 14, 18]. Donner et al. [6] use a random forest / Hough forest combination to first classify the image into candidate regions for each landmark, which are then aggregated by the Hough forest to generate precise location hypotheses. In contrast, [14, 18] use the random forest to directly regress (pseudo) probability maps for the location of each landmark, based on local and global [18] or only local [14] image features, followed by a non-maximum suppression (NMS). Other approaches include decision trees based on a set of image features [2], deep convolutional networks [15] and the discriminative generalized Hough transform [16]. The CRF is generally based on unary potentials (based on the localizer output) and heuristically motivated binary potentials, e.g., distance [2], vector [6], vector field profiles [5], etc. Bergtholdt et al. [2] associate the CRF potentials with weights which are automatically learned using maximum likelihood (ML) based on the posterior probabilities of the training data. They also account for missing landmarks by assigning heuristically motivated values for the corresponding potentials, the weights of which are also learned (involving a heuristic parameter for false misses). However, the ML criterion may stress the influence of outliers, requiring corresponding weightings in case of a large amount of incorrect localization hypotheses. Moreover, a ML approach quickly becomes infeasible with increasing number of combinations in terms of computational complexity. Bergtholdt et al. [2] uses a fully connected graph, and thus does not exploit the potential of simplifying the graph topology for a reduced computational complexity. In contrast, [6] defined the CRF graph topology heuristically based on the differential entropy of the distribution of relative landmark distances calculated on the training data. This may not be optimal if other features than the relative distance are used to characterize landmark pairs.

In this work, we define a pool of CRF potential functions (currently unary and binary, but generally of any arity) and associate a weight with each potential function and each landmark pair (generally each landmark subset). Starting from a fully connected graph, we automatically learn the potential weights together with the values of the potentials in case of missing landmarks. Potentials with low weights are removed, thus optimizing the CRF graph topology. In contrast to [2], we use a max-margin approach (considering only the best incorrect configuration of all landmarks in addition to the correct configuration) in an energy-based formulation [13]. For efficiency reasons (short training and test times, moderate number of annotated training images required), our landmark localizer is based on regression trees [14]. However, any other localizer generating (pseudo) probability maps for each landmark can be used instead (including a deep neural network).

The task of localizing six landmarks of the lower extremities has been addressed before in [8,16,17] using the discriminative generalized Hough transform. However, they only addressed the localization task, i.e., only considering landmarks known to be contained in the image. Thus, they are not able to cope with missing landmarks.

3 Methods

The task is to detect and localize – if present – up to N different landmarks in an image. We solve this problem in two steps: First, landmark-specific regression tree ensembles rating local image features are used to generate n localization hypotheses $\hat{\mathcal{X}}_i = \{\hat{\mathbf{x}}_{i,1}, \ldots, \hat{\mathbf{x}}_{i,n}\}$ for each landmark $i \in \{1, \ldots, N\}$. Second, the unary information of the localizer is combined with binary information rating spatial features between landmarks and jointly modeled in a CRF. An additional "missing" state is introduced to solve the detection problem and all required parameters are automatically learned in a gradient descent optimization. Finally, a common CRF inference technique is applied to find the best selection $\hat{\mathbf{S}} \in \{0, 1, \ldots, n\}^N$ out of all possible selections $\mathcal{S}$. For each landmark, one or no localization hypothesis is selected, effectively solving the detection and localization in one inference step.

Section 3.1 introduces the regression-tree-ensemble-based localizer, followed by the joint formulation of weighted knowledge sources in a CRF in Section 3.2. Finally, the optimization step used to learn all CRF parameters and to reduce the number of necessary potential functions is illustrated in Section 3.3.

3.1 Landmark Localization using Regression Tree Ensembles

The goal of the first step is to predict (as accurately as possible) candidate positions for each landmark based on local context only. At this stage we tolerate confusions as long as any (not necessarily the first) of the $n = 15$ best localization hypotheses is correct, since they will be resolved in the second step. The basic idea is to transform an image $\mathbf{I} : \mathbb{R}^2 \rightarrow \mathbb{R}$ into a pseudo (not normalized) probability map $\widetilde{\mathbf{P}}_i : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ in which the location of the highest value $\hat{\mathbf{x}}_{i,1} = \arg\max_{\mathbf{x}} \widetilde{\mathbf{P}}_i(\mathbf{x})$ corresponds to the most likely predicted position of the target landmark i. For efficiency reasons, we use random forests, which only need a small or moderate number of annotated training images. As in [14], for each landmark i, an ensemble of $K = 96$ decision tree regressors [4] is used to transform feature vectors $\mathbf{f}_i^k(\mathbf{x})$, computed for a certain position $\mathbf{x}$ in image $\mathbf{I}$ for the k-th regression tree, into pseudo probabilities $\widetilde{p}_i^k(\mathbf{x})$. This is done for all pixels in the image and averaged over all trees k to form the pseudo probability map $\widetilde{\mathbf{P}}_i$. Finally, NMS with a minimal distance between peaks of 3 pixels is applied to find local maxima. The n best local maxima are used as localization hypotheses $\hat{\mathcal{X}}_i = \{\hat{\mathbf{x}}_{i,1}, \ldots, \hat{\mathbf{x}}_{i,n}\}$ for each landmark i.

To extract the feature vector $\mathbf{f}_i^k(\mathbf{x})$ for a certain pixel $\mathbf{x}$ we use a BRIEF-like [3] approach. Each tree in the ensemble is associated with an individual sampling mask to extract $F = 128$ pixel intensity values from a local patch. The mask is obtained by sampling locations from $\mathbf{X} \sim$ i.i.d. $\mathcal{N}\left(\mathbf{0}, \frac{1}{25}\left(\begin{smallmatrix} A_1^2 & 0 \\ 0 & A_2^2 \end{smallmatrix}\right)\right)$ with $\mathbf{A} = (a_1 \ a_2)$ being the patch size; in our experiments $\mathbf{A} = (351 \ 351)$ to capture the target object's size. Finally, the masks origin is placed at $\mathbf{x}$ and the intensity value at $\mathbf{x}$ is subtracted from the marked pixel intensities, resulting in our F-dimensional feature vector $\mathbf{f}_i^k(\mathbf{x})$.

Boostrapping is used to train the regression trees in a discriminative fashion by iteratively growing a set $\mathcal{O}_i^k \subseteq \mathbb{R}^F \times \mathbb{R}$ of feature vectors and corresponding target values over all training images. We start out by collecting "positive" samples for each training image by computing feature vectors $\mathbf{f}_i^k(\mathbf{x})$ for all $M = 317$ pixels within a circle with radius $R = 10$ (corresponding to the localization criterion) around the respective annotated landmark position $\mathbf{x}_i^*$. We allow for some ambiguity by introducing a Gaussian distribution $\mathcal{N}_i\left(\mathbf{x}_i^*, \frac{1}{9}R^2\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)\right)$ around $\mathbf{x}_i^*$ and use the density values computed at position $\mathbf{x}$ as regression targets. All "positive" samples are added to the set $\mathcal{O}_i^k$ and an intermediate tree is trained on them. After that, "negative" samples are generated by iterating over all training images and applying the intermediate tree to the training image in order to find the most offending responses. NMS is used to select the M pixels with the largest pseudo probabilities outside the circle located at $\mathbf{x}_i^*$. For those M "negative" pixels, feature vectors are computed and added – with a target regression value 0 – to the growing set of samples $\mathcal{O}_i^k$. After each iteration, a new intermediate and more discriminative tree is trained on the larger set of samples $\mathcal{O}_i^k$ and used in the next iteration. The final tree is then added to the ensemble.

All parameters of the regression tree ensemble have been optimized on a previous task [14] and were adapted to the current one. Some parameters (like the sampling mask) were intuitively chosen to match the dataset, while others (e.g., the number of trees) were chosen to match the hardware constraints.

3.2 CRF with Pool of Potential Functions and "Missing" Label

To compensate for incorrect first best localization hypotheses $\hat{\mathbf{x}}_{i,1}$ for arbitrary landmarks i, we use a CRF to model geometric relationships between landmarks. For notational simplicity, we introduce an index $s_i \in \{0, 1, \ldots, n\}$ for each landmark i to denote the "missing" label $s_i = 0$ and the selection $s_i > 0$ of one of the localization hypotheses $\hat{\mathcal{X}}_i$. For instance, $s_i = 2$ means that the second localization hypothesis $\hat{\mathbf{x}}_{i,2}$ is assigned to the i-th landmark in the CRF. We apply an energy-based formulation [13], where a low energy $E(\mathbf{S})$ of a configuration $\mathbf{S} = (s_1, \ldots, s_N)$ of localization hypotheses over all landmarks implies a large posterior probability. The energy $E(\mathbf{S})$ of the CRF is parameterized by a set of T potential functions $\mathit{\Phi} = \{\phi_1(\cdot), \ldots, \phi_T(\cdot)\}$ (of arbitrary arity) with corresponding weights $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_T)$ scaling each term, and missing potential values $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_T)$:

$$E(\mathbf{S}) = \sum_{j=1}^{T} \lambda_j \cdot \begin{cases} \beta_j & \text{if } s_i = 0 \text{ for any } i \in \text{Scope}(\phi_j) \\ \phi_j(\mathbf{S}) & \text{else} \end{cases} . \tag{1}$$

The explicit inclusion of the missing potential values $\boldsymbol{\beta}$ is necessary to allow computation of $E(\mathbf{S})$ in case of missing landmarks and to automatically learn their values. In inference, the task is to find the selection $\hat{\mathbf{S}}$ amongst all $(n+1)^N$ possible selections $\mathcal{S}$ that minimizes the energy from Eq. (1):

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S} \in \mathcal{S}} E(\mathbf{S}) . \tag{2}$$

The search problem depicted in Eq. (2) becomes intractable very fast with a growing number of states and landmarks, which might require the usage of approximate inference. However, in our case we can still use exact inference in form of the A* search algorithm by Bergtholdt et al. [1], which uses an admissible heuristic to find the global optimum.

The idea of our approach is to define a "pool" $\mathit{\Phi}$ of potential functions $\phi_j(\mathbf{S})$ (motivated clinically, anatomically, by geometric considerations or by "helpful" image features) and to automatically learn their weights λ_j w.r.t. the detection and localization criterion. Potentials with a low weight can then be removed. To illustrate this principle, we define one unary potential for each landmark and three – in this work purely geometrically motivated – binary potentials per landmark pair in $\mathit{\Phi}$.

Unary localizer potential Let $\mathbf{U}_i = (u_{i,1}, \ldots, u_{i,n})$ be the regressed scores for the localization hypotheses $\hat{\mathcal{X}}_i$. We define the unary localizer potential for the i-th landmark as

$$\phi_i^{\mathrm{loc}}(\mathbf{S}) = -\log(u_{i,s_i}) \,. \tag{3}$$

Binary distance potential The first binary potential uses a Gaussian distribution to model the distance between two landmarks i and j. Assuming we estimated the empirical mean $\mu_{i,j}^{\mathrm{dist}}$ and variance $\sigma_{i,j}^2$ of distances on training annotations, and that $f(\cdot)$ is the probability density function of a normal distribution, we define the binary distance potential as

$$\phi_{i,j}^{\mathrm{dist}}(\mathbf{S}) = -\log\left(f(\|\hat{\mathbf{x}}_{i,s_i} - \hat{\mathbf{x}}_{j,s_j}\| \mid \mu_{i,j}^{\mathrm{dist}},\, \sigma_{i,j}^2) \right) \,. \tag{4}$$

Binary angle potential The second binary potential uses a von Mises distribution to model the angle of the line spanned between two landmarks i and j in relation to the x-axis. Similar to the previous distribution, we estimated the distribution's parameters $\mu_{i,j}^{\mathrm{ang}}$ and $\kappa_{i,j}$ using training annotations. Finally, with $g(\cdot)$ being the distribution's probability density function and $\alpha(\mathbf{x})$ a function computing the angle between the vector $\mathbf{x}$ and the x-axis, we define the potential as

$$\phi_{i,j}^{\mathrm{ang}}(\mathbf{S}) = -\log\left(g(\alpha(\hat{\mathbf{x}}_{i,s_i} - \hat{\mathbf{x}}_{j,s_j}) \mid \mu_{i,j}^{\mathrm{ang}},\, \kappa_{i,j}) \right) \,. \tag{5}$$

Binary vector potential For the third binary potential, we use a multivariate Gaussian distribution to model the vector between two landmarks i and j. This includes distance and orientation. However, the vector potential is neither scaling nor rotation invariant, whereas the distance and angle potentials are rotation and scaling invariant, respectively. We include the vector potential to illustrate the concept of a "pool" of potential functions. Again, we estimate the necessary parameters $\boldsymbol{\mu}_{i,j}^{\mathrm{vec}}$ and $\boldsymbol{\Sigma}_{i,j}$ on training annotations. Finally, with $h(\cdot)$ being the probability density function of a multivariate normal distribution, we define this potential as

$$\phi_{i,j}^{\mathrm{vec}}(\mathbf{S}) = -\log\left(h(\hat{\mathbf{x}}_{i,s_i} - \hat{\mathbf{x}}_{j,s_j} \mid \boldsymbol{\mu}_{i,j}^{\mathrm{vec}},\, \boldsymbol{\Sigma}_{i,j}) \right) \,. \tag{6}$$

Pool of potentials With these definitions, we define our pool of potential functions for a fully connected graph as

$$\begin{aligned} \mathit{\Phi} =& \{\phi_i^{\mathrm{loc}}(\cdot) \mid i = 1\,..\,N\} \cup \\ & \{\phi_{i,j}^{\mathrm{dist}}(\cdot),\, \phi_{i,j}^{\mathrm{ang}}(\cdot),\, \phi_{i,j}^{\mathrm{vec}}(\cdot) \mid i = 1\,..\,N,\, j = i+1\,..\,N\} \,. \end{aligned} \tag{7}$$

The remaining tasks are to weight each potential ($\mathbf{\Lambda}$), estimate the energies when an involved landmark is missing ($\boldsymbol{\beta}$) and to remove unnecessary potentials. Note that in principle our approach works with potentials of arbitrary arity.

3.3 Learning of Parameters and Removing Potentials

There exist heuristics [2] to estimate the potential weights as well as the missing energies, but a more common approach is to learn those parameters from data. We follow the latter path by defining an appropriate loss function over data $\mathcal{D}$ and use a gradient descent scheme to optimize it. The probabilistic approach is to use maximum likelihood, which has the drawbacks that one must compute the partition function, which gets intractable quickly, and that it stresses the influence of outliers. Thus, we follow a max-margin approach [12, 13] and try to increase the margin between the correct selection $\mathbf{S}^*$ and the best (lowest energy) incorrect selection $\mathbf{S}^-$. This requires appropriate inference for which we again use the A* algorithm.

A well known loss function is the hinge loss, which tries to increase the energy gap between $\mathbf{S}^*$ and $\mathbf{S}^-$ until a certain margin $m = 1$ is satisfied. The intuition is that a margin m improves generalization and that only samples not satisfying the margin continue to contribute to the parameter updates. Let our loss function be defined as

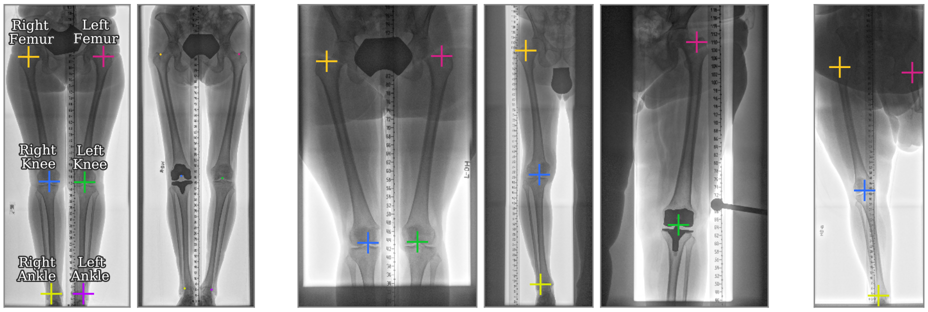
$$L(\mathbf{\Lambda}, \boldsymbol{\beta}) = \frac{1}{K} \sum_{k=1}^{K} \max\left(0, m + E(\mathbf{S}_k^*) - E(\mathbf{S}_k^-)\right) + \theta \cdot \sum_{j=1}^{T} |\lambda_j| \, . \tag{8}$$

In addition to the data term over all K training samples, we added a θ-weighted L1 regularization term w.r.t. $\mathbf{\Lambda}$ to further accelerate the sparsification of terms. I.e., instead of defining a topology and manually selecting appropriate potential functions, our idea is to start with a fully connected graph and a pool of different potentials Φ and to learn which of those potentials are meaningful. Once we optimized $\mathbf{\Lambda}$, we can simply remove all zero-weighted ($\lambda_j = 0$) potentials. This solves the problem of defining a topology as well as selecting meaningful potentials.

To optimize the loss function from Eq. (8), we apply a variant of stochastic gradient descent in form of the Adam algorithm by Kingma and Ba [11]. We use a global step-size of $\alpha = 0.01$ and leave all remaining parameters as proposed in [11]. Furthermore, we use a mini-batch size of $K = 40$ samples per iteration, which greatly improves the time until convergence, which is usually reached after ~ 200 iterations. To improve generalization, we optimize the potential weights $\mathbf{\Lambda}$ and missing energies $\boldsymbol{\beta}$ on a different portion of training examples than used to train the potential functions themselves (i.e., probability distribution parameters, localizers, etc.). Once all parameters are estimated, we remove all unnecessary potentials where $\lambda_j = 0$ to reduce the runtime and complexity of the system.

4 Results

We evaluated the proposed approach on an in-house dataset of 660 images showing the lower extremities of 606 patients with an age in the range of 19 to 100 years.

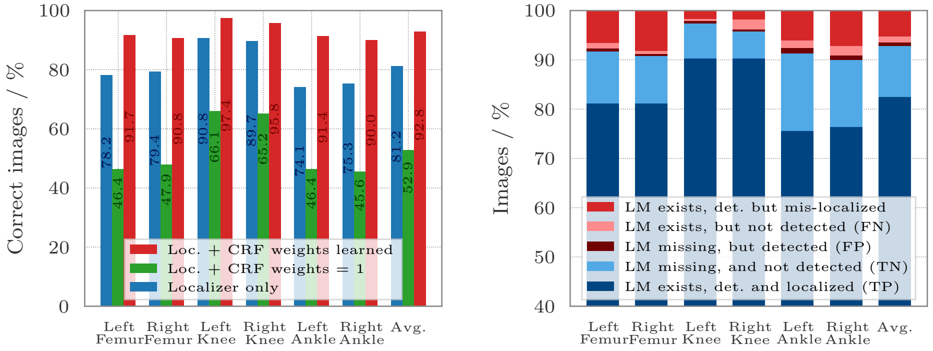


(a) Full lower body (b) Restricted field of view (c) Missing

Fig. 1: A few samples including annotations of the 660 images showing (a) full lower bodies, (b) a restricted field of view and (c) a full lower body with missing limbs. Note the two knee prostheses. The small circle annotations in the second image correspond to the area in which a localization is assumed correct.

The task is to detect and localize (if present) up to 6 different landmarks, namely the femur, knee and ankle of both legs. A few sample images are shown in Fig. 1. We downsampled the images to an isotropic resolution of 1 mm/px to speed up the processing. Due to a restricted field of view and missing limbs in a subset of images, not all landmarks are present in all images. Only 73.4 % of the images contain all landmarks, while 8, 78, 77, and 10 images only contain 5, 4, 3, and 2 landmarks, respectively. Hence, the task is to detect whether a landmark is present in conjunction with the task to localize it, if present. We consider two kinds of results to be correct. First, the landmark is missing and the algorithm predicted it to be missing. Second, the landmark is not missing and the algorithm detected it and predicted a position with an Euclidean distance to the true position below 10 mm. The tolerance of 10 mm has been chosen by Ruppertshofen et al. [17] and is illustrated in the third image in Fig. 3a.

We used patient-grouped 5-fold cross validation in our experiments, which provided us with, on average, 530 training images per fold. 30 % of the training images of each fold were used to train the localizer (Section 3.1) and to estimate the parameters of the probability distributions (Section 3.2), while the remaining 70 % were used to learn the weights and missing potential values (Section 3.3). Note that we exclusively used 15 % of the latter training images as validation set to properly select a regressor weight θ. The final results over all folds in terms of correct detection and localization (as described above) are shown in Fig. 2a. The localizer itself, i.e., always using the first best localization hypothesis, shows mediocre performance with on average 81.2 %. First, it assumes a landmark is always present and thus the numbers are biased. Second, it performs significantly worse when the landmarks are close to the image's border, i.e., for the femur and ankle landmarks due to only partly available information. In contrast to previous works [6,14,18], we have to properly estimate the CRF weights. Without learning the parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}$, just setting them to 1, we obtain an accuracy of only



(a) Performance of different systems

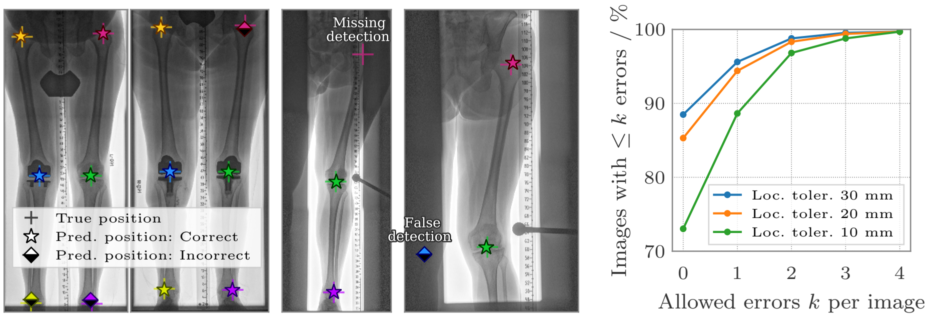
(b) Error rates of weighted CRF

Fig. 2: (a) Amount of correct images in percent w.r.t. the different landmarks; 100 % corresponds to 660 images. (b) Distribution of errors across detection and localization over all images in percent for the localizer with learned CRF weights. The two bottom-most bars correspond to the rates of our approach depicted in (a), followed by three bars for the three different sources of errors.

52.9 %. In contrast, after learning all parameters we handle 92.8 % of the samples correctly, averaged over the different landmarks. Furthermore, our approach is also very robust against altered target objects in form of prostheses. 97.5 % of the 319 prostheses were properly detected and localized. The performance of our approach is broken down in Fig. 2b. We see that the detection task was solved on average in 82.5 % TP + 10.3 % TN + 5.3 % mis-loc. = 98.1 % of the images. The largest amount of errors is due to mis-localization with 5.3 %, in contrast to mis-detection with only 0.7 % FP + 1.3 % FN = 2.0 %.

Looking at resulting images (see examples in Fig. 3a), the localization tolerance of 10 mm appears to be quite strict, which is also illustrated in the second image in Fig. 1. This is addressed in Fig. 3b, where the amount of correct images w.r.t. a certain number of errors per image in relation to the localization tolerance is plotted. Increasing the tolerance from 10 mm to 20 mm, the percentage of images where all 6 landmarks are handled correctly increases by 12.3 percent points to 85.3 %; the average detection and localization rate for a single landmark increases to 96.2 %. Depending on the application, e.g., if the localization hypothesis is further refined by post-processing on a small crop of the image, a less strict localization tolerance might be sufficient.

A quantitative comparison to [16] is difficult due to different evaluation setups (cross-validation in our work and a single unknown training and test split in [16]) and a different objective (namely localization only, not detection). However, if we only consider cases where an existing landmark was detected (true positives), we can quantify the localization performance of our approach. Note, due to the above reasons we refrain from drawing any conclusions. Using the same tolerance of 10 mm as used by Ruppertshofen et al., we achieved to correctly localize 91.7 %, 98.1 % and 92.0 % of the femur, knee and ankle landmarks, respectively,



(a) Illustration of the three different error types

(b) Influence loc. tolerance

Fig. 3: (a) Illustration of the three different kinds of errors: The first two images illustrate mis-localization due to the error tolerance of 10 mm. However, note the accurate localization despite the prostheses. The third image shows a landmark (left femur) not being detected. The fourth image illustrates a falsely detected landmark. (b) Distribution function of landmark errors per image (between 0 and 6) for different localization tolerance levels.

averaged over both legs. In contrast, Ruppertshofen et al. achieved a respective performance of 73.9 %, 93.7 % and 86.6 %.

By dropping all zero-weighted potentials, we were able to remove on average 45.5 % of $6 + 3 \cdot \frac{6 \cdot 5}{2} = 51$ CRF potentials: All unary potentials remained, while 3, 7, and 13 of the angle, distance and vector potentials were removed, respectively. This reduced the inference time on average by 20.1 %.

5 Discussion and Conclusions

In this paper, we proposed an automatic approach for learning the weights of potentials as well as the values of the potentials for missing landmarks in a conditional random field using a max-margin hinge loss and gradient descent. In particular, we suggested to define a pool of potential functions for the CRF, learn their weights and remove all potentials which were assigned a weight of 0 by employing an L1 sparsity prior. This allows to automatically select the most appropriate potential functions and to define the CRF graph topology, starting from a fully connected graph, in a single optimization framework. We investigated our approach to localize six landmarks of the lower extremities on a dataset of 660 X-ray images with significant fractions of missing landmarks due to restricted field of view. Although on average 45.5 % of the CRF potentials have been removed, we detected and localized (within 10 mm) on average 92.8 % of the different landmarks while being very robust against prostheses. Increasing the localization tolerance to 20 mm further improved the performance to 96.2 %. Our approach can be extended to use different (and additional) landmark localizers (e.g., deep convolutional neural networks), further binary potentials (e.g., incorporating gray value profiles along edges [5]) or potentials of higher arity (e.g., the relative position of landmark triples), where higher order clique reduction techniques [10] seem promising. Also, a zooming approach [9] could be added to further refine the landmark positions. Since our approach is fairly general, we can apply it to different landmark localization tasks with limited manual effort.

Acknowledgements

The authors thank the Diagnosezentrum Urania, Vienna and the Dartmouth Hitchcock Medical Center, Lebanon for providing the radiographs that served as training and test sets; Gooßen [8] for the annotations. This work has been financially supported by the Federal Ministry of Education and Research under the grant 03FH013IX5. The liability for the content of this work lies with the authors.

References

1. Bergtholdt, M., Kappes, J.H., Schnörr, C.: Learning of graphical models and efficient inference for object class recognition. In: JPRS. pp. 273–283 (2006)
2. Bergtholdt et al.: A study of parts-based object class detection using complete graphs. IJCV 87(1), 93–117 (2010)
3. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010 pp. 778–792 (2010)
4. Criminisi et al.: Regression forests for efficient anatomy detection and localization in computed tomography scans. Medical image analysis 17(8), 1293–1303 (2013)
5. Donner et al.: Sparse MRF Appearance Models for Fast Anatomical Structure Localisation. BMVC (2007)
6. Donner et al.: Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization. Medical image analysis 17(8), 1304–1314 (2013)
7. Glocker et al.: Vertebrae localization in pathological spine ct via dense classification from sparse annotations. In: MICCAI. pp. 262–270. Springer (2013)
8. Gooßen, A.: Computational Imaging in Orthopaedic Radiography. BoD (2012)
9. Hahmann et al.: Model interpolation for eye localization using the Discriminative Generalized Hough Transform. BIOSIG (2012)
10. Ishikawa, H.: Higher-order clique reduction in binary graph cut. In: CVPR. pp. 2993–3000. IEEE (2009)
11. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. ICLR (2014)
12. Komodakis, N., Xiang, B., Paragios, N.: A framework for efficient structured max-margin learning of high-order mrf models. IEEE TPAMI 37(7) (2015)
13. LeCun, Y., Chopra, S., Hadsell, R.: A tutorial on energy-based learning. Predicting Structured Data (2006)
14. Mader, A.O., Schramm, H., Meyer, C.: Efficient epiphyses localization using regression tree ensembles and a conditional random field. In: BVM (2017)
15. Payer et al.: Regressing heatmaps for multiple landmark localization using cnns. In: MICCAI. pp. 230–238. Springer (2016)
16. Ruppertshofen et al.: Discriminative generalized hough transform for localization of joints in the lower extremities. CSRD 26(1), 97–105 (2011)
17. Ruppertshofen et al.: Shape model training for concurrent localization of the left and right knee. In: SPIE Medical Imaging (2011)
18. Štern, D., Ebner, T., Urschler, M.: From local to global random regression forests: Exploring anatomical landmark localization. In: MICCAI. pp. 221–229 (2016)
19. Wang, C., Komodakis, N., Paragios, N.: Markov random field modeling, inference & learning in computer vision & image understanding: A survey. CVIU (2013)

Index of Authors