Bioimage informatics

DiversePathsJ: Diverse Shortest Paths for Bioimage Analysis

Virginie Uhlmann 1, Carsten Haubold 2, Fred A. Hamprecht 2,* and Michael Unser 1,*

1 École Polytechnique Fédérale de Lausanne (EPFL), Biomedical Imaging Group, 1015 Lausanne, Switzerland and
2 Heidelberg University, IWR/HCI, 69120 Heidelberg, Germany.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: We introduce a formulation for the general task of finding diverse shortest paths between two end-points. Our approach is not linked to a specific biological problem and can be applied to a large variety of images thanks to its generic implementation as a user-friendly ImageJ/Fiji plugin. It relies on the introduction of additional layers in a Viterbi path graph, which requires slight modifications to the standard Viterbi algorithm rules. This layered graph construction allows for the specification of various constraints imposing diversity between solutions.

Results: The software allows obtaining a collection of diverse shortest paths under some user-defined constraints through a convenient and user-friendly interface. It can be used alone or be integrated into larger image analysis pipelines.

Availability and implementation: http://bigwww.epfl.ch/algorithms/diversepathsj

Contact: michael.unser@epfl.ch, fred.hamprecht@iwr.uni-heidelberg.de

Supplementary information: Supplementary material is available at Bioinformatics online.

1 Introduction

A large number of problems in image analysis and computer vision can be formulated as search for the shortest path. Given an edge-weighted graph, the optimal shortest path between two nodes can be found efficiently using dynamic programming (Bellman, 1952). The Viterbi algorithm is a popular example of the latter. A solution is, however, always only optimal with respect to its associated cost function or energy. In practice, due to discrepancies between the model associated with an objective function and reality, it is common for the optimal solution to differ from what would be the best one from a human perspective. For this reason, obtaining a collection of good solutions instead of a single one is often desirable.

The design of algorithms for obtaining the M most probable solutions to combinatorial problems has been extensively studied in the context of probabilistic random field models and is known as the M-best maximum a posteriori (MAP) problem (Lawler, 1972; Batra, 2012). Further works aimed at introducing supplementary constraints to ensure that the M-best solutions are sufficiently diverse (Batra et al., 2012; Kirillov et al., 2015).

In Straehle et al. (2013), these very general formulations were adapted for bioimage segmentation in the context of exploring various watershed solutions by searching for the K-smallest spanning trees. Most similar to our approach is the one from Fujita et al. (2003) in the context of robot motion planning, which could also be viewed as a two-layers-graph construction.

We make two contributions: first, we propose a more general formulation for the diversity constraint and second, we provide the first freely available, readily-usable, and general purpose open source implementation of a diverse shortest path tool for bioimage analysis.

2 Method

The search for a diverse collection of M-diverse best solutions using a layered-graph construction is implemented efficiently by modifying the rules of the standard Viterbi algorithm. The mathematical formulation of our approach is inspired from our more general work (Haubold et al., 2017)
...and is provided in the Supplementary Material. Here, we give an intuitive description of the algorithm.

To find the shortest path, the standard Viterbi algorithm is performed from a source point \( \sigma \) to a target point \( \tau \) by computing the cost to all intermediate nodes. We refer to this collection of costs as the forward derived cost map (Figure 1a). The shortest path is found by backtracking the minimal cost from \( \tau \) to \( \sigma \) (Figure 1b). For the second shortest path, another pass of Viterbi algorithm is first performed, this time from the target \( \tau \) to the source \( \sigma \), to obtain the backward derived cost map (Figure 1c). The forward and backward derived cost map are then summed (Figure 1d) and all nodes laying within an exclusion corridor around the shortest path are assigned an infinitely large cost (represented as white area in Figure 1e). The exclusion corridor aims at enforcing that the second shortest path differs from the first one. The node corresponding to the new minimal cost in the summed map, denoted as (\( \nu^* \), \( z^*_\nu \)), is identified. Finally, the second shortest path is obtained by joining the two sub-paths obtained by backtracking from (\( \nu^* \), \( z^*_\nu \)) to \( \sigma \) and from (\( \nu^* \), \( z^*_\nu \)) to \( \tau \). This amounts to performing the Viterbi algorithm on a two-layers graph constructed by copying the original graph and introducing layer-jump-edges (Figure 1f).

The next \( M - 2 \) diverse best solutions can be computed efficiently by updating the exclusion corridor to be the union of the exclusion corridors built from previously found solutions, and searching for the next minimum in the remaining available nodes of the summed cost map. The \( M \) shortest paths are thus computed at the cost of two standard Viterbi algorithm passes plus \( M \) backtraclings. A possible refinement is to enforce in addition that the new solution accumulates a given amount of diversity. Formally, this requires the introduction of a supplementary input cost map on the lower graph layer encoding the discrepancy between the new solution and previous ones, referred to as the diversity map. It can be combined with the exclusion corridor to enhance diversity in the set of shortest paths. Layer-jump-edges are then allowed only when the integrated diversity over the path exceeds a threshold.

Our approach is implemented as a Java-based plug-in for ImageJ and Fiji. The plug-in is designed to be generic and readily usable. A full description of the GUI and all available settings is provided in the Supplementary Material. The required input consists in two end-points provided by the user or by custom automated pre-processing steps. The collection of resulting (diverse) paths can then either be swiftly browsed and displayed on the image using the arrow keyboard keys, or exported in generic formats for further processing.

3 Results

Bioimages often feature several instances of the same object in a single image (e.g., cells). Due to small differences between the appearance and neighborhood of these individual objects, it is practically impossible to find a cost function whose minimum always corresponds to the desired output. Exploring several shortest paths, however, allows to analyze multiple instances of the same objects exhibiting slight variations without the need for fine-tuning of the cost function.

We illustrate the usefulness of searching for several shortest paths in bioimage analysis with identification of the medial axis of mycobacteria (phase contrast microscopy images) and of \( C. \) elegans nematodes (brightfield microscopy images). The original images, their characteristic, experimental protocols and plugin settings can be found in the Supplementary Material. We use our approach with the same parameters to analyze two similar objects in the same image. For one of the objects (Figures 1g and 1j), the shortest path is the desired medial axis. Conversely, for the second one, the shortest path does not correspond to a correct medial axis (Figures 1b and 1k). The desired solution can, however, be found in the set of additional diverse shortest path (Figures 1i and 1l), the exploration of which is made easy through the interface of our ImageJ/Fiji plugin. A full description of the software is provided in the Supplementary Materials.

Medial axis outline serves as the first step for quantitative estimation of shape features such as length and bending. The proposed approach is however not restricted to this use-case and covers every problem in which a path is to be searched between two end-points. Our implementation only requires the specification of the source and target locations as ImageJ points and of the input cost map specifying the cost associated to each pixel in the image, in addition to data-independent parameters (number of desired shortest paths and minimum amount of diversity between paths).

Ridge detectors and algorithms estimating probability distributions for the identification of principal curves such as subspace constrained mean shift (Ozertem et al., 2011) are good examples of relevant input cost maps. DiversePathJ can be used alone in an interactive way or integrated as a module in larger automated image analysis pipelines.

Funding

This work was supported by the Swiss National Science Foundation under Grant 200020_162143/1 (VU and MU); and by DFG SFB1129, DFG HA4364/9 – 1 and IGS MathComp (CH and FAH).

Conflict of Interest:

none declared.

References


Fig. 1. Algorithm for finding the second-best path. (a) Forward derived cost map (from source node \( s \)). (b) Shortest path. (c) Backward derived cost map (from target node \( t \)). (d) Summed cost map. (e) Exclusion corridor mask and node \( (u^*, x_n^*) \) with minimal cost, and (f) resulting second-shortest path obtained by backtracking from \( (u^*, x_n^*) \) to \( s \) and \( t \). The underlying input image, source, and target locations are shown in (k). Application examples in images of mucobacteria (phase-contrast) and neutrophils (brightfield). For some objects, the shortest paths correspond to the desired one (g) and (i), but for others in the same image (h) and (j), only the 3rd shortest is correct (j) and (l).