
Morphological Graph Opening

Release 0.00

David Doria

June 13, 2011

Army Research Laboratory, Aberdeen MD

Abstract

This document presents an implementation of an algorithm to perform a morphological opening on a graph. The intent is to remove short branches in a graph while preserving the large scale structure.

This implementation is based on the algorithm described in [1]. We have used the data structures from Boost Graph Library (BGL).

The code is available here: <https://github.com/daviddoria/GraphOpening>

Latest version available at the [Insight Journal](http://hdl.handle.net/10380/3250) [<http://hdl.handle.net/10380/3250>]
Distributed under [Creative Commons Attribution License](#)

Contents

1	Introduction	2
2	Graph Erosion	2
3	Graph Dilation	2
4	Algorithm	2
4.1	Naive Approach	2
4.2	Speedup	2
5	Notes and Behavior	2
6	Demonstration	3
7	Code Snippet	3

1 Introduction

This document presents an implementation of an algorithm to perform a morphological opening on a graph. This opening consists of a series of erosions, followed by a series of dilations. This is often done as a first step in contour closure algorithms.

This implementation is based on the algorithm described in [1].

2 Graph Erosion

Remove all edges attached to EndPoints (leaf nodes). After multiple iterations of this, small branches will be "absorbed" into a main branch.

3 Graph Dilation

Add back edges to current EndPoints. This is NOT simply going to grow back the same tree. If we ran the erosion enough times to absorb a branch, the branch will not grow back. If we did not, then the branch will indeed grow back.

4 Algorithm

The algorithm proceeds as follow:

- Erode the graph until a stopping criteria is met
- Dilate the graph the same number of times as the graph was eroded

4.1 Naive Approach

At each iteration, erode or dilate the graph while considering nothing about the previous step. In both the erosion and dilation steps, this means finding all of the end points at every iteration, which is the only costly procedure in the algorithm.

4.2 Speedup

A list of the remaining vertices after every edge removal is created. These are the only potential candidates which must be determined to be end points or not at the next iteration.

5 Notes and Behavior

- If an edge was removed in the erosion process, as long as it belonged to the child-most branch that was entirely removed, it will not be regrown in the dilation process.

6 Demonstration

7 Code Snippet

References

- [1] Sappa, A, *Efficient Closed Contour Extraction from Range Image's Edge Points*. Proceedings of the 2005 IEEE International Conference on Robotics and Automation ([document](#)), 1