TopoGO: Knot Detection of Images

电子与电气工程系
DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Yicheng Zhang[1], Zhengyang Cao[2]

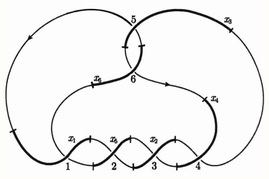
[1, 2]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China. Github: [1]@RUSRUSHB, [2]@drinktoomuchsax

Introduction

Knot theory is an important branch of mathematics that plays a pivotal role in various fields. Knot detection requires tedious manual work, and is very difficult for deep learning methods to handle. We automate its procedure in two approaches with Alexander polynomial which uniquely describes the knot.

Mathematical Procedures

- Construct a planar diagram of the knot.
- Identify and number line segments and crossings.
- Construct the Alexander matrix.
- Calculate the determinant and normalize the polynomial.



(a) Numbered Diagram

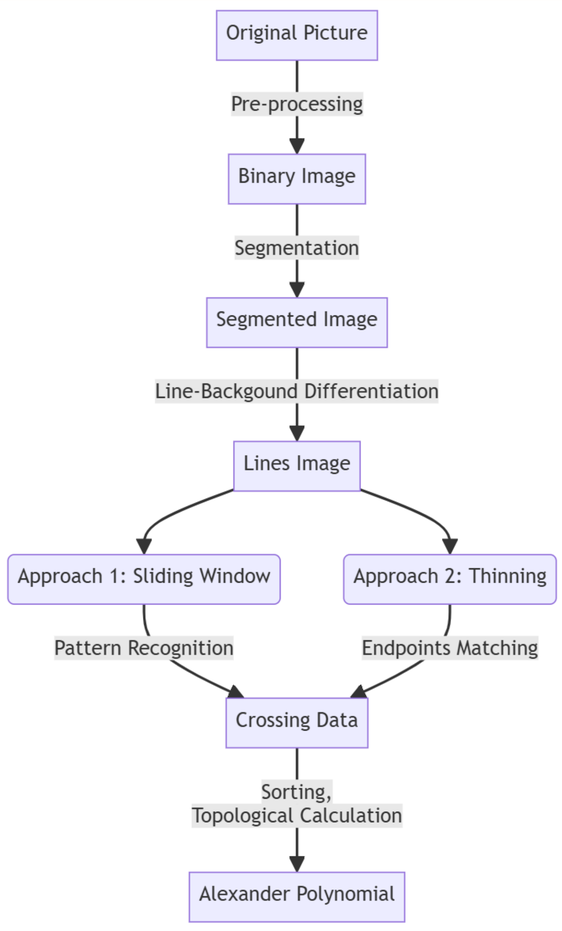
$$\begin{pmatrix} 1-t & -1 & 0 & 0 & -1 & 0 \\ 0 & t & 1-t & -1 & 0 & 0 \\ 0 & 0 & 0 & t & 1-t & t \\ 0 & 0 & -1 & 1-t & 0 & -1 \\ -1 & 1-t & t & 0 & 0 & 0 \\ t & 0 & 0 & 0 & t & 1-t \end{pmatrix}$$

(b) Alexander Matrix

Figure 1: How To Get Alexander Matrix

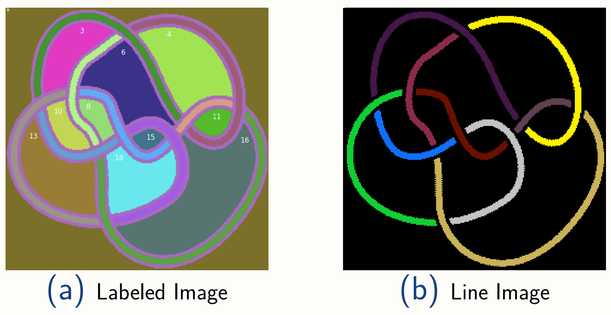
Methodology

We start with pre-processing including binarization. Segment the image to extract lines. Then we have two approaches: sliding window and thinning. Sliding window requires pattern recognition, and thinning requires endpoints matching. Then, we have the information of crossings. After sorting and topological calculation, we will obtain the Alexander polynomial.



Union-Find & Maximum Inscribed Circle

Union-Find Labeling distinguishes different components from the image; Maximum Inscribed Circle is used to differentiate line pixels from the background.



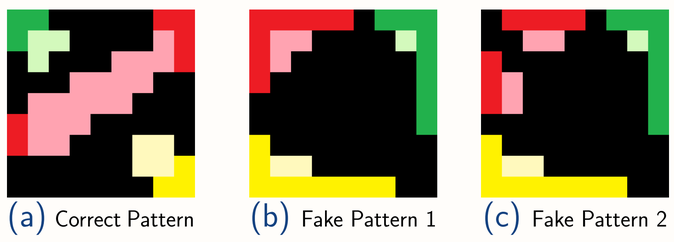
(a) Labeled Image

(b) Line Image

Figure 2: Segmentation of the Binarized Image

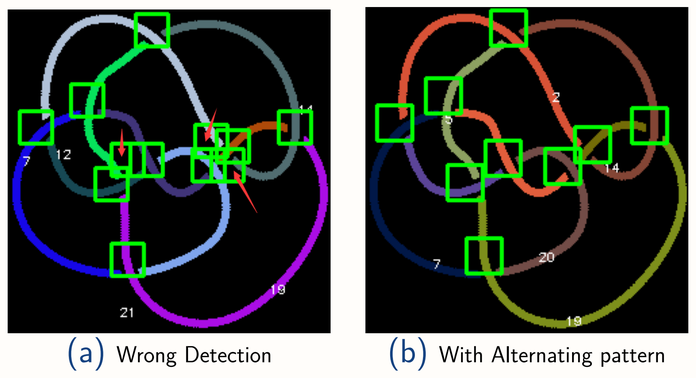
Crossing Detection: Thick Lines

We scan the image with a sliding window and try to identify crossings and their top-bottom relationship. We use a boundary pattern recognition approach to avoid errors.



(a) Correct Pattern (b) Fake Pattern 1 (c) Fake Pattern 2

Figure 3: Crossing Pattern Boundaries: Black and light color region is ignored to accelerate detection

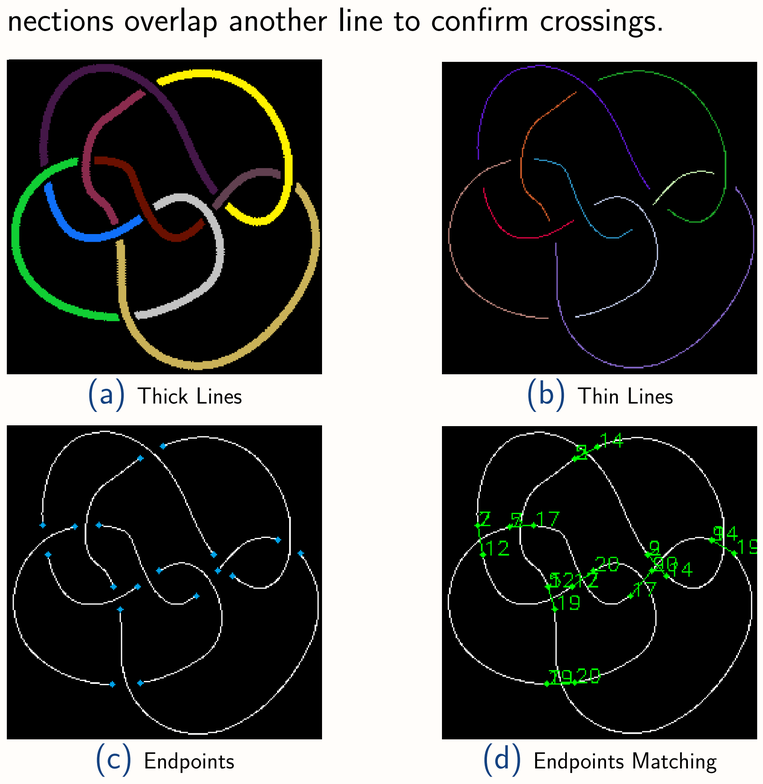


(a) Wrong Detection (b) With Alternating pattern

Figure 4: Crossing Detection Approach 1: sliding window with alternating pattern

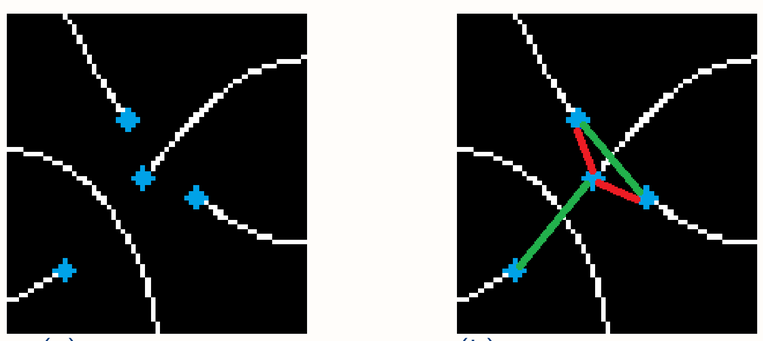
Crossing Detection: Thin Lines

For 1-pixel thin lines, we detect endpoints and match them as near as possible with KDTree, and check whether connections overlap another line to confirm crossings.



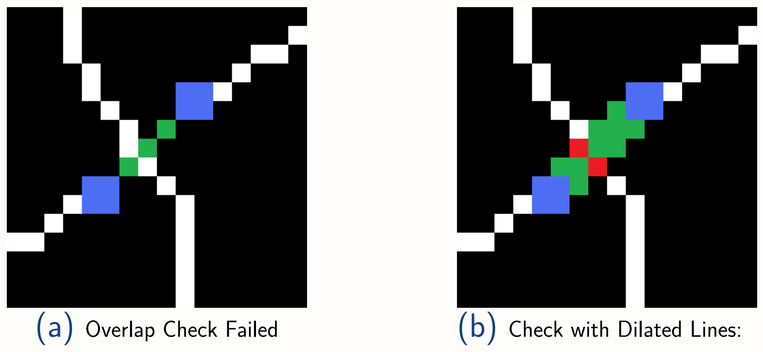
(a) Thick Lines (b) Thin Lines (c) Endpoints (d) Endpoints Matching

Figure 5: Crossing Detection Approach 2: Thinning



(a) Endpoints: Regional (b) Endpoints Matching: Red is wrong, green is right

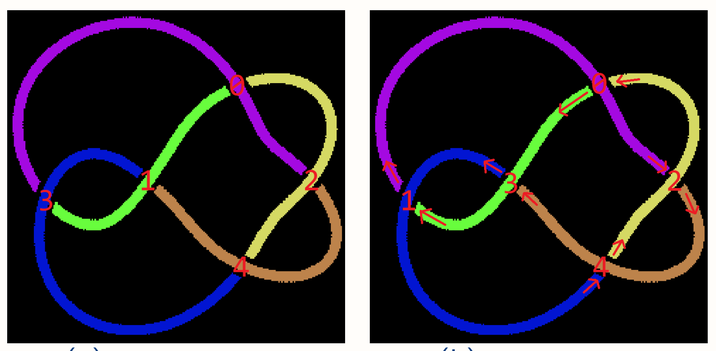
Figure 6: Endpoints and Matching: Regional



(a) Overlap Check Failed (b) Check with Dilated Lines: Red part is overlapping

Figure 7: Overlap Check

The crossings and lines are identified, but not in the correct sequence. We use $O(1)$ array manipulation, instead of $O(n^2)$ flood fill action "to move along the line".



(a) Unsorted Crossings (b) Sorted Crossings

Figure 8: Crossing Sort

Results and Discussion

We tested 151 images from the Rolfsen Knot Table. Lower methods are always superior in accuracy.

Table 1: Comparison of Knot Detection Methods

Method	Accuracy	Runtime	Fail Cases
Three-Color	37.1%	110s	(1)(2)(3)(4)
Alternating Pattern	42.4%	217s	(2)(3)(4)
Adaptive Window	66.2%	356s	(3)(4)
Adap. Win. & Alt. Pat.	95.4%	552s	(3)(4)
Thinning	96.7%	207s	(4)

(1) Close crossings (2) Very close crossings
(3) Large crossings (4) Background errors

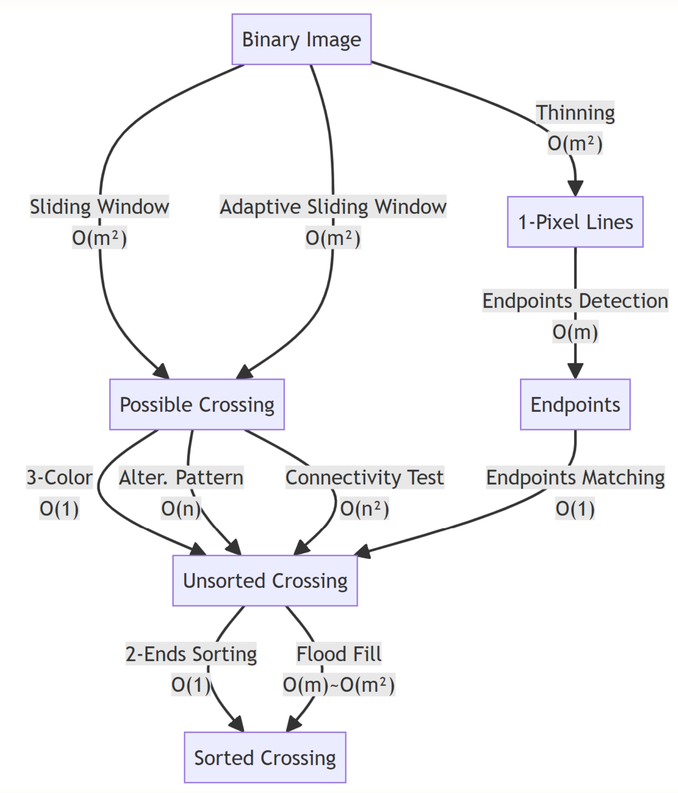


Figure 9: Complexity of Approaches

(m: size of the picture. n: size of the sliding window.) Three-Color method requires $O(m^2)$ time and space complexity, and Thinning method requires $O(m^3)$.

Conclusion

We presented two approaches for knot detection: sliding window and thinning. The Thinning method achieved the best accuracy at 96.7%, avoiding background errors and too-large-crossing errors. The Three-Color method is the fastest though not robust against unregulated inputs. It could be improved with regulated images or restrictions like Alternating Pattern test. Our algorithm addresses problems that are difficult for deep learning to handle, demonstrating the powerful capabilities of mathematical and graphical analysis in the field of image processing.