TKDetection: a software to detect and segment wood knots

Adrien Krähenbühl Bertrand Kerautret Isabelle Debled-Rennesson

Université de Lorraine, LORIA, ADAGIo team, UMR 7503, 54506, Nancy, France.
adrien.krahenbuhl@loria.fr, bertrand.kerautret@loria.fr, isabelle.debled-rennesson@loria.fr

Abstract TKDetection is a software proposing to segment the wood knots obtained from X-Ray Computed Tomography (CT) scanners. It implements algorithms combining tools of image analysis and discrete geometry, like connected component extraction, contour extraction or dominant point detection. TKDetection is the first free and open source software for the automatic knot segmentation. It is available on Github platform [2].

Keywords: X-Ray CT images, segmentation, discrete geometry

1 Introduction

The non-destructive analysis of the internal structure of trees is an actual challenge for biologists and sawmills. The first try to understand the details of tree growth mechanisms, while the second ones hope to improve their productivity and the wooden plank quality. The both currently use X-Ray CT scanners to obtain precise information and statistics on wood stems. TKDetection is a software allowing to manipulate images of tree scanned by X-Ray CT scanners. Its goal is to detect and segment the tree knots before computing their characteristics. It has been used to experiment and validate the algorithms proposed in [4] and [3]. TKDetection uses the concepts of different domains and especially the discrete geometry to perform the segmentation step. We can quote discrete tools like discrete circles, discrete lines, blurred segments or dominant points. They allow pixel, voxel and primitive manipulate with an exact integer position.

2 Software overview

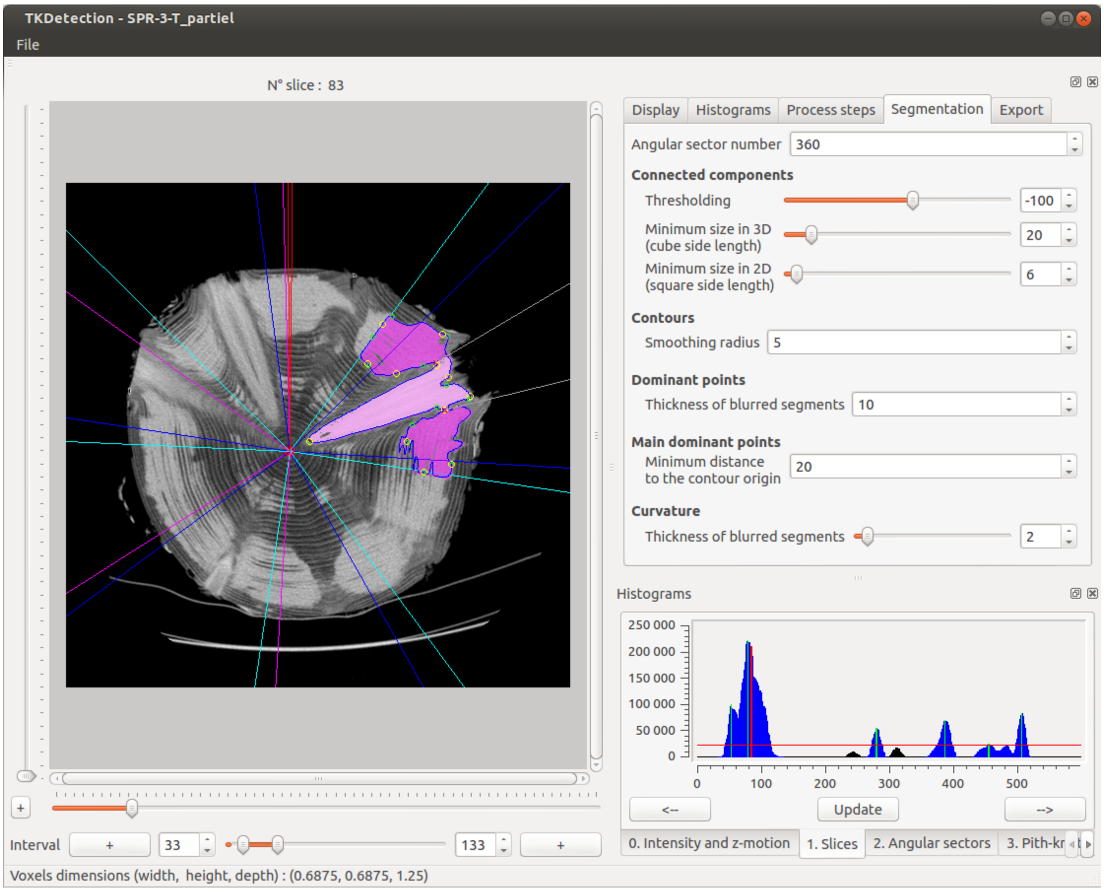
2.1 Interface and functionalities

TKDetection is a single windowed software (see Fig. 1). The central part is dedicated to the image display and proposes to zoom and slide between images. Two movable and floatable panels complete the interface. The first panel proposes to set parameters related to the display panel and to all steps of knot segmentation that appear in [3]. The parameters are grouped by tabs. A first tab offers several display modes, in addition to original display. It corresponds to low level image processing like contour detection or optical flow. It also shows the z-motion display, a notion introduced in [4]. There are four other tabs concerning the histograms, the steps of global process, the segmentation step and the result exports. The different data sets processed in the algorithms (like the intensity histogram, z-motion histogram or contour histogram) are dynamically displayed in the second panel according to the user selection.

2.2 Development

TKDetection is developed in C++ with the QT framework. The latter mainly provides graphical components and the signals/slots mechanism exploited in the user interface.

TKDetection uses some external libraries:



Angular sector number 360

Connected components

Thresholding -100

Minimum size in 3D (cube side length) 20

Minimum size in 2D (square side length) 6

Contours

Smoothing radius 5

Dominant points

Thickness of blurred segments 10

Main dominant points

Minimum distance to the contour origin 20

Curvature

Thickness of blurred segments 2

Interval 33 133

Voxels dimensions (width, height, depth) : (0.6875, 0.6875, 1.25)

Figure 1: Overview of TKDetection interface.

- Armadillo to manipulate matrix [6],
- Insight Toolkit (ITK) to open DICOM and TIFF images [7],
- Qxt, Qwt and QwtPolar that provide additional graphical elements to Qt,
- DGLib for the dominant points detection [5],
- ImaGene, used by DGLib.

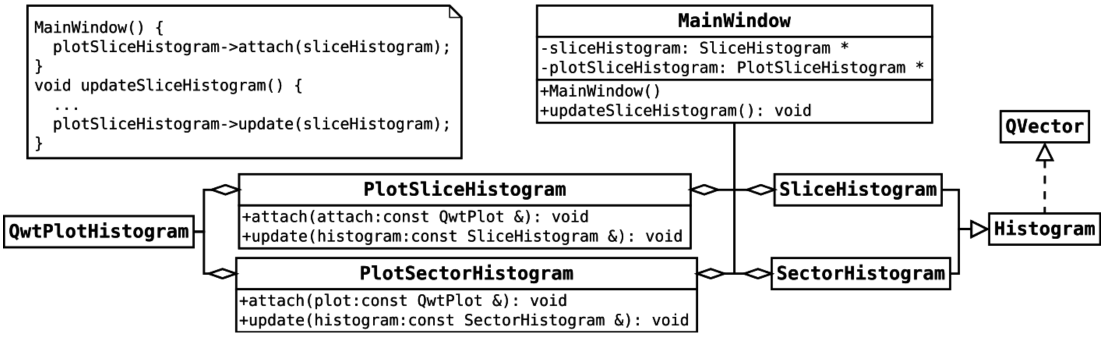
A simple viewer *DGtalBranchVisu*, given in TKDetection repository, offers a 3D visualization of the export results. It uses the DGTal library and the associated DGtal-tools [1]. TKDetection constantly and easily evolves through a permanent design focus. We can see by example in Fig. 2 the class diagram of histogram design.

3 Global process details

A complete process consists in segmenting all wood knots from a set of X-Ray CT images by using the method detailed in [3]. However, TKDetection is mainly developed to test, experiment and visualize algorithms. Accordingly, it is not optimised to segment a complete stem ; it is designed to quickly test the settings for one knot.

A basic example of single knot segmentation can be performed by following these steps:

1. Open a DICOM image of tree,
2. Eventually modify the default parameters,
3. Extract the pith position (required for the knot interval computation),



```
MainWindow() {
  plotSliceHistogram->attach(sliceHistogram);
}
void updateSliceHistogram() {
  ...
  plotSliceHistogram->update(sliceHistogram);
}
```

MainWindow
-sliceHistogram: SliceHistogram *
-plotSliceHistogram: PlotSliceHistogram *
+MainWindow()
+updateSliceHistogram(): void
QVector
PlotSliceHistogram
+attach(attach:const QwtPlot &): void
+update(histogram:const SliceHistogram &): void
SliceHistogram
QwtPlotHistogram
Histogram
PlotSectorHistogram
+attach(plot:const QwtPlot &): void
+update(histogram:const SectorHistogram &): void
SectorHistogram

Figure 2: Class diagram of histogram design.

4. Select a knot interval (required for the corresponding angular sector computation),
5. Select an angular sector (required for the corresponding knot area segmentation).

At each step, the user can visualize the final result superimposed to the original image. For example in Fig. 1, the segmented knot is shown in light pink while the contour is marked in blue. The yellow and green circles represent the dominant points of this contour. It can be seen on the bottom right panel, the histogram of knot intervals. A vertical red line represents the current displayed slice. The most important feature is the ability to restart the process from any step. The selection of steps 4 and 5 can quickly be changed or recomputed by changing the settings. This allows to make intensive setting tests. At the end of the process the tool provides a segmented knot that can be exported. The supported export formats are PGM3D, OFS, DAT and others. The histograms can be also exported to any image format (JPEG, PNG, PDF, etc.).

Acknowledgements

We would like to thank T. P. Nguyen for the dominant point detection code [5].

References

[1] DGtal: Digital geometry tools & algorithms library. `http://libdgtal.org`.

[2] Adrien Krähenbühl and Bertrand Kerautret. TKDetection. `https://github.com/adrien057/TKDetection`.

[3] Adrien Krähenbühl, Bertrand Kerautret, and Isabelle Debled-Rennesson. Knot segmentation in noisy 3d images of wood. In *DGCI 2013, Sevilla, Spain*, March 2013.

[4] Adrien Krähenbühl, Bertrand Kerautret, Isabelle Debled-Rennesson, Fleur Longuetaud, and Frédéric Mothe. Knot detection in x-ray ct images of wood. In *ISVC 2012, Rethymnon, Crete, Greece*, volume 7432 of *LNCS*, pages 209–218, July 2012.

[5] Thanh Phuong Nguyen and Isabelle Debled-Rennesson. A discrete geometry approach for dominant point detection. *Pattern Recognition*, 44(1):32–44, 2011.

[6] Conrad Sanderson. Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA, 2010.

[7] T. S. Yoo, M. J. Ackerman, and W. E. Lorensen. Engineering and algorithm design for an image processing api: A technical report on itk - the insight toolkit. *Proc. of Medicine Meets Virtual Reality*, pages 586–592, 2002.