Computer Vision Final Project

Incisor segmentation in radiographs with Active Shape Model

Lennart Cockx

Jeroen Van De Laer

# Introduction

The program is divided in several modules as seen in Figure 1. The division is made by functionality. This made it easy to test and evaluate the modules individually. A description of each modules will be given with an explanation of the used algorithm and the necessary visualization. In this way it will become clear how the total program works.

Figure 1: Program modules

# File Manager

The purpose of the File Manager is to have a module where the necessary files are loaded, where some basic processing gets done and visualize the loaded items. The available files are dental radiographs, landmarks of the teeth that need to be extracted and the segmentations of these teeth. The goal of the project is to achieve the segmentation the 8 incisors out of the dental radiograph.

The Procrustes analysis gets executed on the landmarks of the teeth. The Procrustes will analyze the given shapes and perform translating, rotating and uniformly scaling of the objects to obtain a similar placement and size between them. This is necessary to have a mutual axis and scaling to be able to do further processing with the landmarks of the teeth. Figure 2 shows the results of the Procrustes analysis of the variations of one tooth.

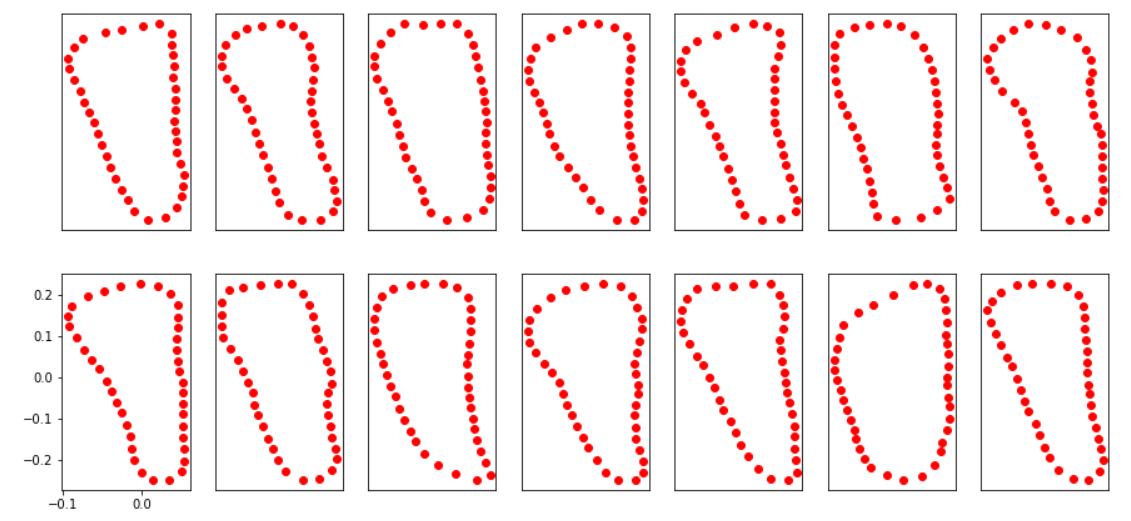


Figure 2: Procusteres of the variations of one tooth

# PCA Analysis

This is a small module where the PCA analysis gets executed. The PCA is implemented via the Scikit learn library. The understanding of the PCA algorithm was already proven with the previous assignment. Therefore, an existing implementation of the PCA was used instead of the manual way. The eigenvectors of a single tooth are visualised in Figure 3.

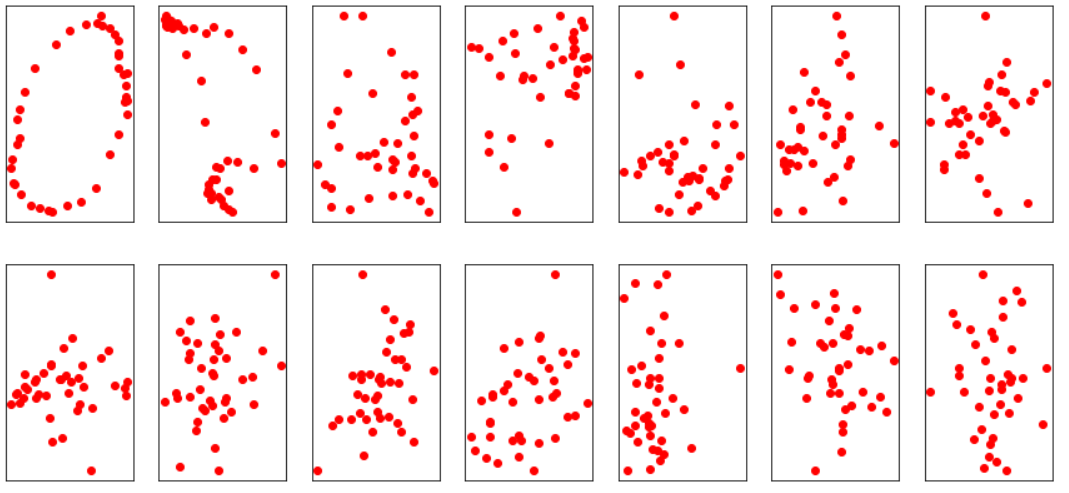


Figure 3: PCA analysis, the eigenvectors of a tooth

# Image Preparation

Some pre-processing algorithms get tested and evaluated in the Image Preparation module. This is done in order to pick the right pre-processing for the given images of the radiographs. The contrast in the picture needs to be stretched as much without getting too much unwanted bright spots. An edge detection is executed to the obtained image. This image will be necessary for other modules to have clear well-defined edges. These edges will be used to detect where the teeth are in the radiograph. The comparison of the different pre-processing techniques is seen in Figure 4.

The median filter is used, because it will filter out the some of the noise in the image. It will even out the image. The contrast stretching will explanation of the different filters. Which filter we are going to use.

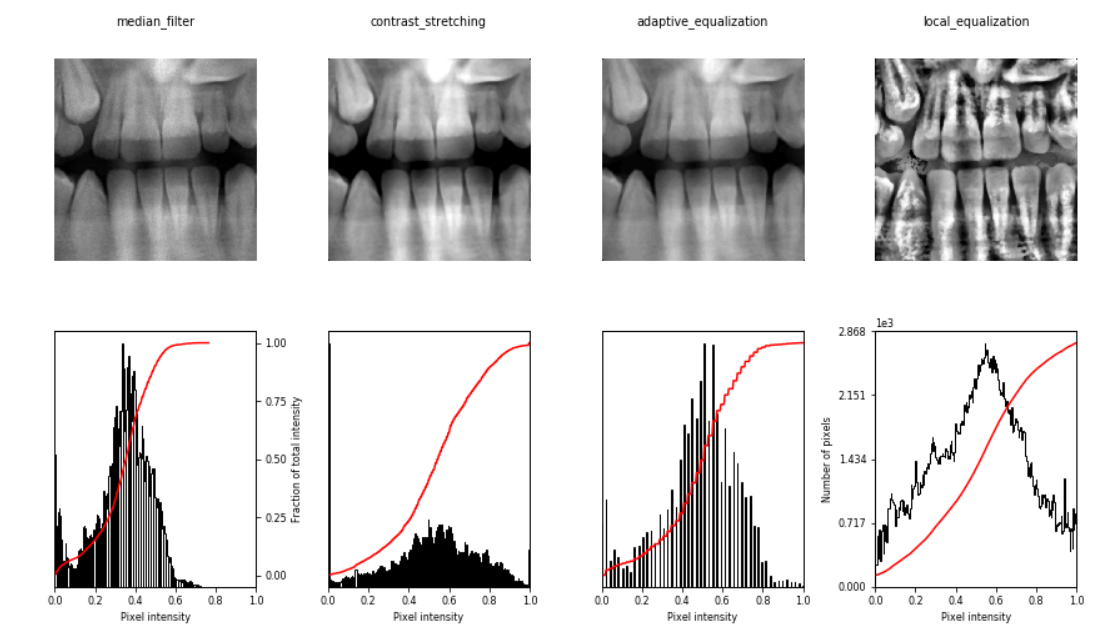


Figure 4: Pre-processing comparison of a) median filter, b) contrast stretching, c) adaptive equalization and d) local equalization

# Initial Pose Estimator

The initial pose estimator is going to give an estimate of the location of the gap between the teeth. This module can be used to automate the estimation of the teeth location. Initial position of the landmarks on the radiograph will be positioned according to this estimate.

The module looks at the intensities of the pixels along the vertical axis. The gap between the teeth have a lower intensity than the teeth themselves. This reduction in intensity is used the locate the gap. Darker spots than the intensity of the gap can occur at the edges of the image or at other location. Therefore, the lowest intensity cannot be used. The solution was to give an initial estimate and the variance of the location of the gap. In this way the unwanted darker spots are ignored. Figure 5 shows the intensity summed along the horizontal axis that is marked by the green area. The radiograph shows where the gaps are located.

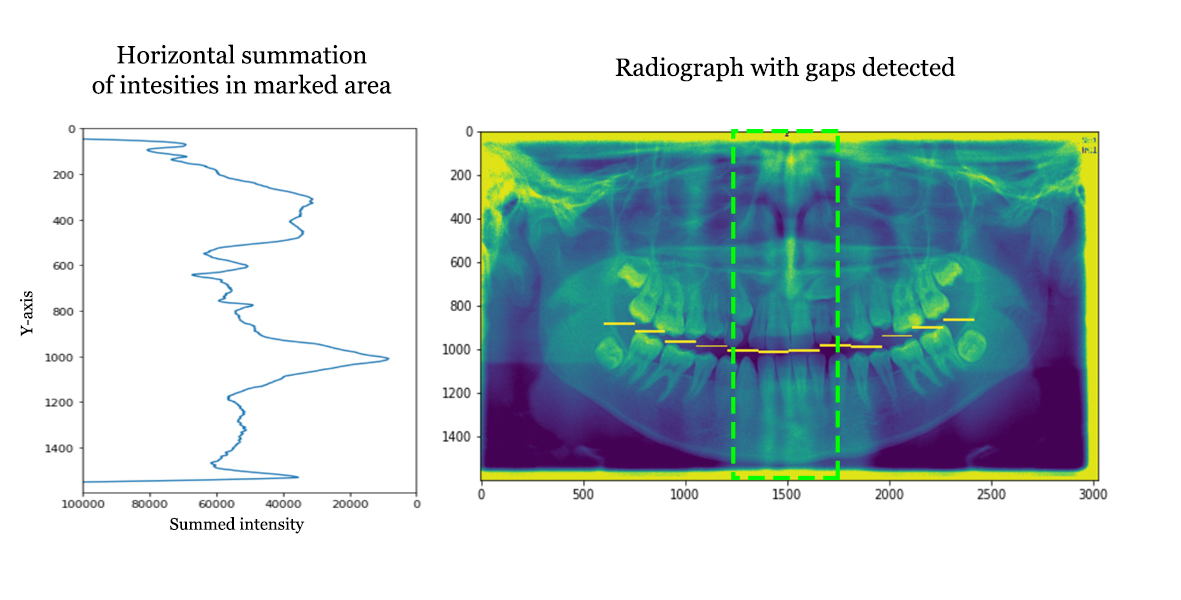


Figure 5: a) Intensity along vertical axis and b) radiograph with gap detection

# Interface

The interface module allows for a user-friendly way to interact with the different parts of the tooth segmentation process. Upon opening, the first radiograph is displayed with the active shape model overlaid on top. The user can switch between the various available radiographs and test the manual/automatic initialisation and segmentation.

|  |  |
| --- | --- |
| Figure 6: Interface controls | Figure 7: Radiograph viewer with overlaid with the Active shape model |

By moving the mouse, the position of the Active Shape Model can be chosen manually. The model can be fixed in place by clicking. Optionally, the shape and size of the model can be adjusted to fit the current radiograph. The chosen initial position can be saved to a numpy array file to test the fitting function. If an automatic initialisation is preferred, this is also possible. The initial pose estimator from the previous section will be used to determine a good y-position for the currently shown radiograph. The x-position is taken in the centre of the image, as most are symmetrical.

Finally, if the Active Shape model is in the desired position, segmentation can be done. To do this, a polygon is drawn between the landmarks of the model. The obtained polygons are used as a mask to extract the desired teeth, shown in Figure 8.

|  |  |
| --- | --- |
| Figure 8: Example segmentation |  |

# Fit Function

The fit function module is going to look for the biggest edge on the normal of the given points for a certain distance. The landmarks of the teeth are given as input. New points are expected as output that indicate the new shape. This shape is sought by looking at the nearby edges on the normal to the boundary. This procedure will not maintain the shape of the tooth. It only cares about the edge strength.

In Figure 9a, the radiograph is seen with a few landmarks which represent the boundary of an estimation of the tooth. The normal on this boundary is drawn a certain range of view. The edge strength on this line is plotted on Figure 9b It is seen that there is a nearby edge on one side. The point with the maximum edge strength will be chosen as a next estimate. This is done for all landmarks and we get a new estimation of the tooth of interest.

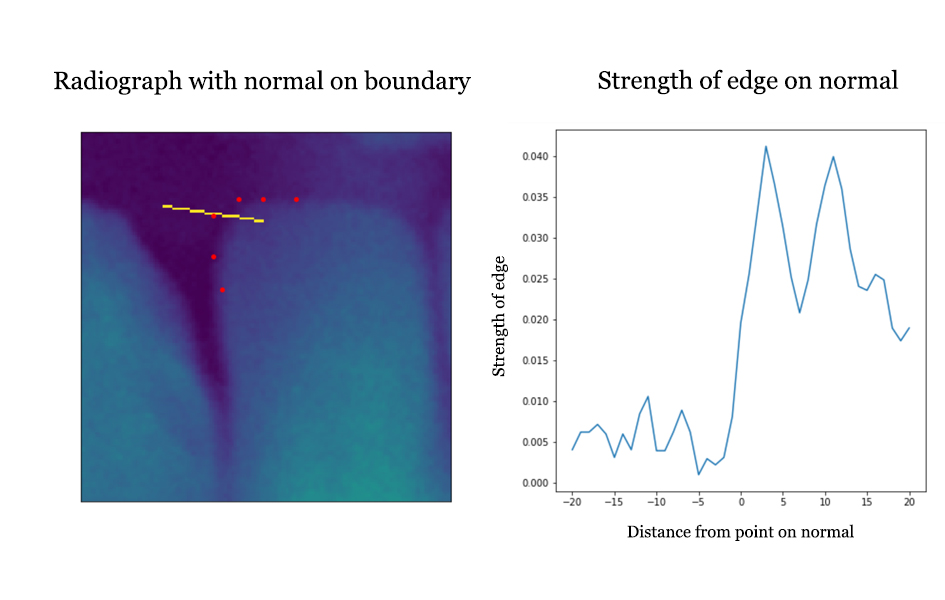


Figure 9: Edge strength normal on boundary

Eventueel nog uitleggen hoe we de hoek berekenen.

# Active Contour

An implementation of an active contour algorithm was implemented in the Active Contour module. This module was not used in the final result due to bad point estimation. The algorithm is based on the Viterbi algorithm with the use of an energy function for the likelihood of a point.

The energy function consisted of internal and external energy. Internal energy takes the structure of the object into account. Whereas the external energy which keeps the edges into account so there is a need to converge to an edge. The internal energy looks at the distances between consecutive points. The energy will increase when the points lay farther apart. The external energy will decrease when there is a strong edge. The goal is to minimize the energy function.

The Viterbi algorithm tries to find the optimal combination of points to minimize the total energy function. The points get a window with certain size to look for new possible points. The algorithm tries to find the best predecessor point for each possible position of the current point. It stores those results and we can backtrack from the last point to get the optimal configuration.

Some problems occurred when using this algorithm. Points where converging to each other because the internal energy was the lowest here. The shapes became unrecognizable, because there was not restriction on how to manoeuvre to other point states. The points were convergence to the wrong elements. Upscaling the energy function from considering two points to three points did not improve performance.

Misschien nog foto van slechte resultaten hier.

# Active Fit Contour

The Active Fit Contour module is a combination of the Active Contour and Fit Function module. This is done to overcome the problems of the normal the normal Active Contour module. It restricts the new possible point location to only the points on the normal of the boundary. This will keep the shape of the tooth in a more consistent way. The problem of points converging to the same location is excluded with this approach as well.

The internal function is also updated. Here the mean distance between each point is calculated. The new distance between the points is divided by the mean distance. In this way, the measure is more normalised. The same goes for the external energy which will be divided by the maximum absolute value in the picture.

Hier ook wat afbeeldingen

# Matching Model Points

In this module the given points get matched to the model points. This module is an implementation of the protocol described in Figure 10. The model is generated by the PCA analysis. Here a limit is placed on the amount of variance from the mean shape. The absolute value of the shape parameter b may only variate till to ensure plausible shapes. The transformation gets calculated to be able to compare the model points to the target points.

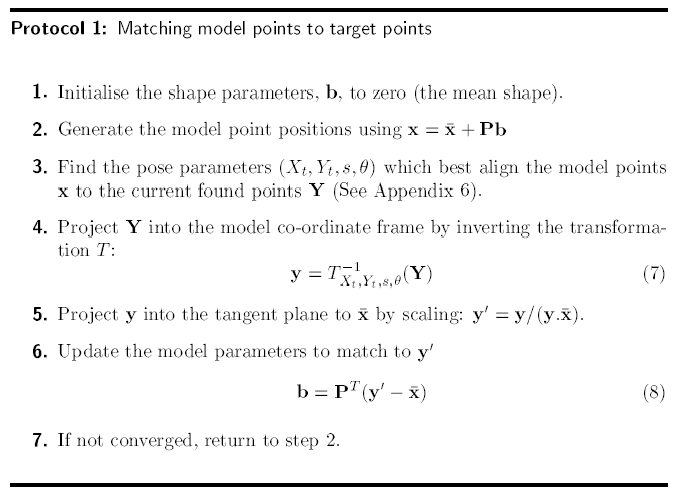


Figure 10: Protocol for matching model points to target points

Hier kunnen ook wat afbeeldingen.

There is still some room for improvement in this module. The pose parameters must be calculated in step 3 of the protocol seen in Figure 10. The purpose of the pose parameters is for the transformation from the given points to the original model size, location and rotation. The calculation is performed by calculating the transformation of 3 points chosen from the first point list to the 3 points at the same location on the second point list, which are maximally spaced from each other. In this way an error can occur when the chosen points are not representable to calculate the transformation.

# Active Shape Model

The last module provides the active shape model. This module executes the pieces of the other modules and puts them together to have an active shape model. The pre-processing gets executed beforehand. Then an iterative process occurs of finding new points possible points and matching those model points to the shape model. This process continues until there is a convergence. The process can be observed after every iteration to visualize how the active shape model evolves from state to state.

Visualisation of the last steps

# Conclusion