KU Leuven

Computer Vision

Erasmus Program

Incisor Segmentation

Authors:

Alejandro Rodriguez Salamanca: r0650814
Fernando Collado Egea: r0650586

June 10, 2017

Contents

1 Introduction

Identificating objects in images manually is a tedious process and error prone if performed by a human. Images are inherently noisy, and sometimes indentifying the object we are looking for is not an easy duty. An automatic method to indentify and segmentate objects in images is then needed.

In this projects, a model-based segmentation approach has been implemented based on Active Shape Models, developed by Cootes et al.

2 Active Shape Model

Active shape models are statistical models of the shape of objects which iteratively deform to fit to an example of the object in a new image.

2.1 Load data

For this assignment we are provided by a set of landmarks, each one containing a list of points corresponding with the shape of a tooth. Those landmarks should be used to build an Active shape model. The first step is to load the landmarks. They are stored in list form, following the format $[X_1, Y_1, \ldots, X_n, Y_n]^T$. To increase the number of samples, the mirrored version of the landmarks is also loaded.

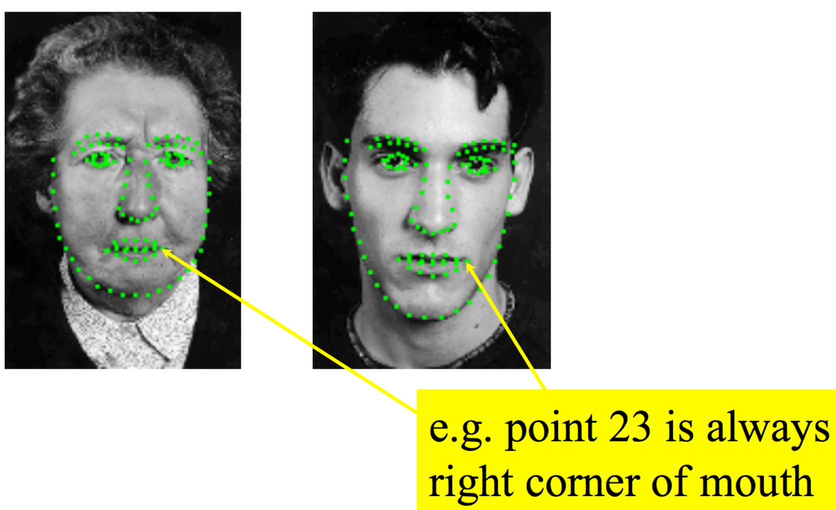

Figure 1: Landmarks on a face

Each landmark indicates points of importance in the image, this is, clear cornes of object boundaries, junctions between boundaries, etc. However, such points are not abundant. For this reason, equally spaced points between landmark points are also used. Is important to point that there should be a correlation between landmarks in different images.

2.2 Align the training set

Once the landmarks are loaded, the training set must be aligned in order to compare equivalent points from different shapes. This alignment is done by scaling, rotating and translating the training shapes using Algorithm 1.

Rotate, scale and translate each shape to align with the first shape in the set;
while *process has not converged* **do**
  Calculate the mean shape from the aligned shapes;
  Normalize the orientation, scale and origin of the current mean to suitable defaults;
  Realign every shape with the current mean;
end

Algorithm 1: Procrustes Analysis

The process converges when the variation of the mean shape between two consecutive interations changes less than a given tolerance.

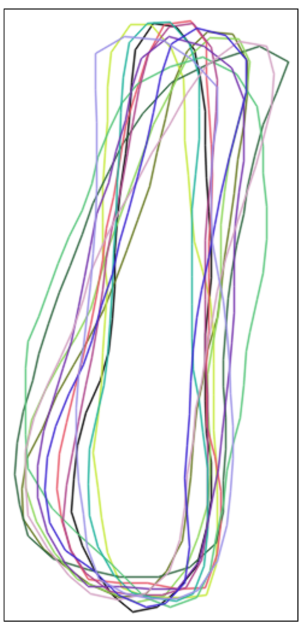

Figure 2: Training shapes

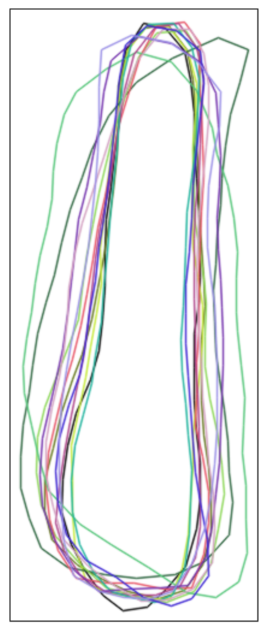

Figure 3: Aligned shapes

Once the shapes are aligned, the mean shape can be calculated. This mean shape will be used to calculate the initial fit of our segmentation.

2.3 Principal Component Analysis

The set of aligned shapes contains as much dimensions as landmarks have each shape. To reduce the dimensionality of the dataset and get rid of the extra dimensions, Principal Component Analysis is applied. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The first principal component has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

The first 8 components hold more than 98% of variance. The dimension of the dataset is reduced from 80 to 8.

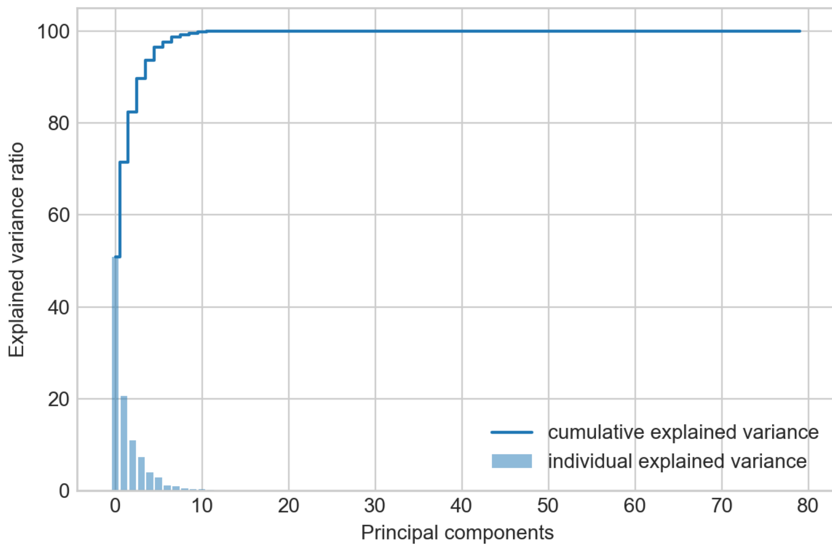

Figure 4: Variance

3 Grey Level Models

We can model the appearence of an object by examining the statistics of grey levels in regions around each of the labelled model points in the training images. As with the shape, the grey-level environment can be modelled by a mean and a number of modes of allowed variation.

For every landmark point j in the image i of the training set, we extract a gray level profile g_{ij} , of length np pixels, centered around the landmark point. To reduce the effect of global intensity changes, we do not use the actual vector g but use the normalized derivative instead.

The gray-level profile of the j^{th} landmark in the i^{th} image is a vector of $2k+1$ element

$$g_{ij} = [g_{ij0}, g_{ij1}, \ldots, g_{ij(2k+1)}]^T$$

And its differential form is of $2k$ length

$$dg_{ij} = [g_{ij1} - g_{ij0}, g_{ij2} - g_{ij1}, \ldots, g_{ij(2k+1)} - g_{ij2k}]^T$$

Finally, the differential form is normalized.

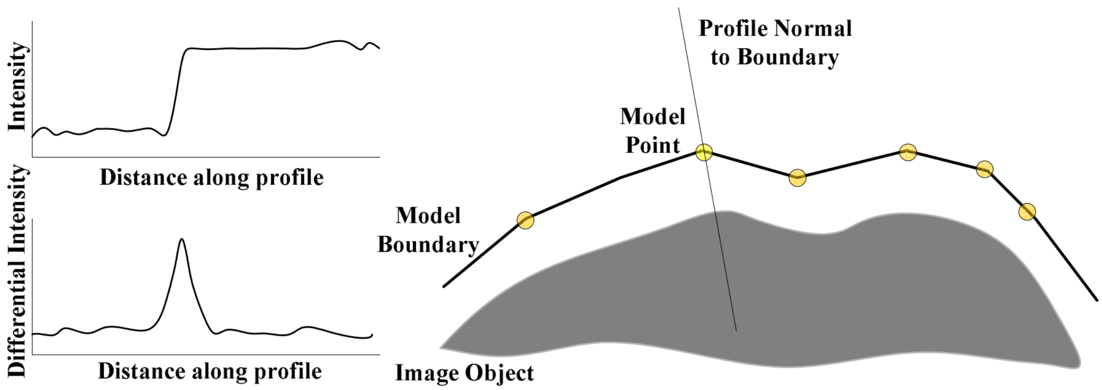

Figure 5: Grey Level Model

4 Preprocess images

Radiographs are images inherently noisy. In order to locate the teeth precisely, the original radiographs should be converted to images as noiseless as posible, but sharp enough to locate the edges of the teeth. For this purpose the following algorithms and transfomations are used:

1. **Non-Local Means Denoising** - This algorithm performs denoising replacing the color of a pixel with an average of the colors of similar pixels. With the correct parameters, the result will have less noise than the original image, maintainig the image sharp.

2. **Top hat** - Is the difference between an input image and its opening. It extracts small elements and details from given images. The result of this transform is substracted to the original image to remove objects that are brighter than the surroundings.

3. **Black hat** - Is the difference between an input image and its closing. It performs the opposite effect than top hat. The result of this transformation is substracted to the original image to remove objects that are darker than the surroundings.

4. **CLAHE** - This name stands for Contrast Limited Adaptive Histogram Equalization. In this, image is divided into small blocks called "tiles". Then each of these blocks are histogram equalized as usual. If any histogram bin is above the specified contrast limit those pixels are clipped and distributed uniformly to other bins before applying histogram equalization.

5. **Edge detection with Sobel and Laplacian** - Sobel edge detector is a gradient based method based on the first order derivatives. It calculates the first derivatives of the image separately for the X and Y axes. The Laplacian of an image highlights regions of rapid intensity change.

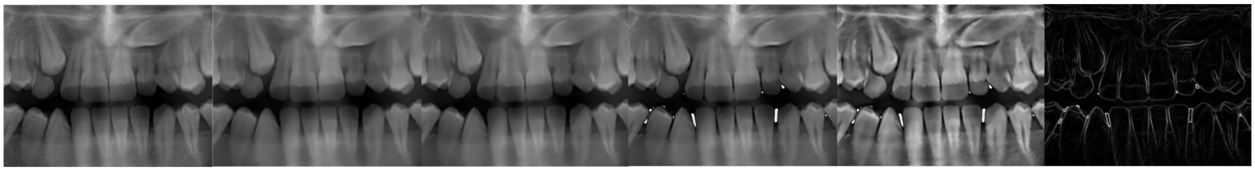

Figure 6: Image after applying the alogorithms

5 Initialization

5.1 Manual

As suggested in the project statement, manual initialization was implemented first to evaluate the fitting algorithm and the automatic initialization separately.

The implementation is pretty straightforward. Using different OpenCV routines, the mean shape can be dragged to its initial position, and one it has been located, the fitting algorithm starts.

Nevertheless, this procedure presents a problem. The rotation of the shape can't be changed, which causes that the initial estimation is not as accurate as it could be.

5.2 Automatic

What really would make a difference in object detection is to make it automatic, and not having to drag a shape, for each object to be detected, in each image. Our first approach to make an automatic model was using histogram of oriented gradients (HOG), combined with training a support vector machine (SVM), to detect the shape of the 8 incisives altogether. But we stumbled upon an obstacle. Whereas with other images (much more clear, less noise, with highly detailed edges) this method is feasible, with our images it was not as good, even after treating the image with techniques (such as denoising, white and black hat...) The following images represent an example with a normal image, and one cropped section of incisives, from radiograph1.

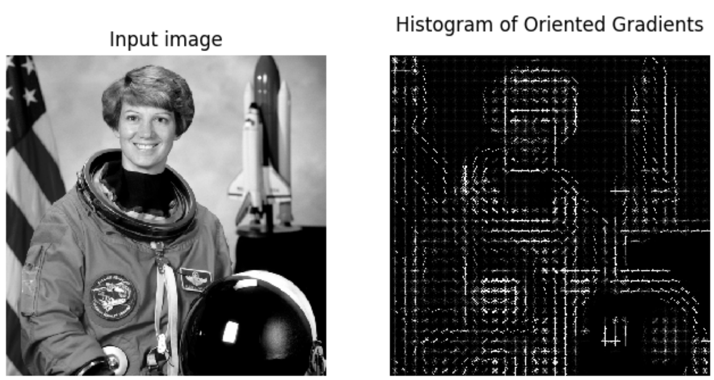

Figure 7: Normal image and its HOG

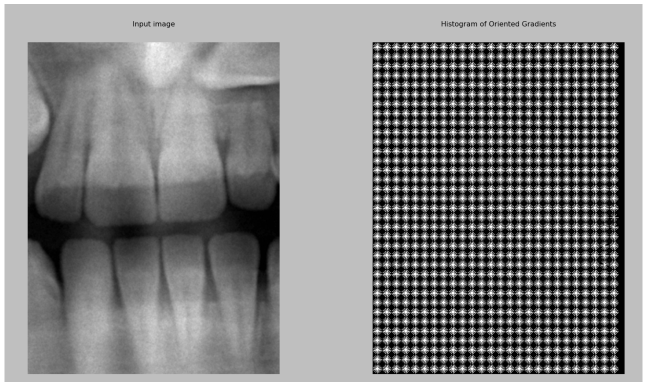

Figure 8: Cropped teeth section

After that, we opted for using the Viola-Jones detection technique. With this technique, some "basic" features are used to train a cascade classifier, which consist of several "weak" classifiers, boosted by the appending of one afther the other.

In order to train the classifiers, sections such as the one shown in the figure above were cropped from each of the 30 radiographs, for the positive data, and, as for the negative data, 150 random images were gathered. Each image (regardless positive or negative), was denoised and greyed before being used for training.

As for how the training was made, first, a *out.vec* was created with the tool *opencv_createsamples*, and the set of images, together with a list of anotations for the positive samples. Once having the out.vec, the next step is to use the tool *opencv_traincascade*, in order to create the corresponding .xml file with the classifier. Due to the ammount of data not being too large, we choose to make it a 3-stage classifier (more would have been not efficient, due to the time to train it, and not improving too much, due to having a low quantity of samples)

The results obtained are good enough, but not better. The gist of the Viola-Jones technique is that it is quite fast, but it is not flawless, and it found quite some false positives, as shown in the image below

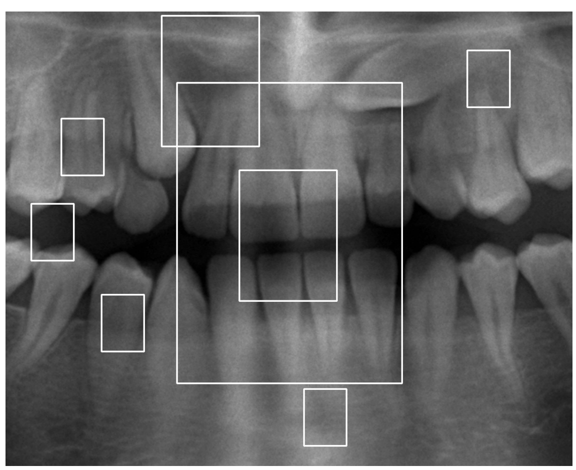

Figure 9: Initial results of Viola-Janes detection

As can be seen, there is one very good match, and several bad matches. For this matter, what we did in order to correct the results are two things, first, detection-wise, we tinkered with the detection parameters, changing the scale at which the image would be resized for further detections, and the minimum number of neighbours, which, if changed to a high value, it would result in few detections, but of higher quality, which is what we are interested on. Result-wise changes made to improve the detection were, first delete rectangles that are inside of others, and then proceed to ignore those that are not sufficiently large enough. With these two techniques, we have a result such as this

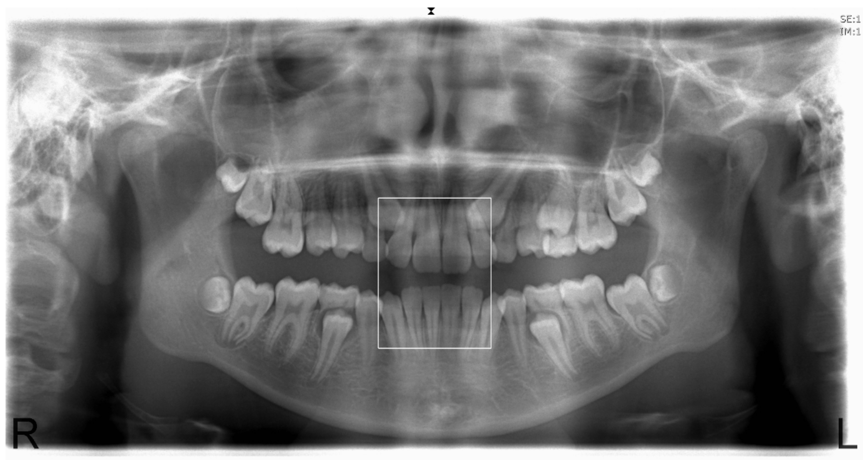

Figure 10: Improved results

Once we have a proper automatic detector mechanism, all that remains is to place the shape in its correct position. For this matter, a small choosing technique was developed:

- If the incisive is one of the upper four, the y position is located at the first third of the rectangle. For the x centroid, the width is divided in 4, and the centroid will be the center of one of them (corresponding to the incisive)

- On the other hand, if the incisive is one of the lower four, the y centroid is located slightly below the second third of the height. The x position in this case is trickier. The lower incisives are usually smaller (altogether) than the upper ones, so a different mechanism has to be made. For this matter, instead of dividing the width by 4, it was dividived by 8, and the first two, and the last two areas are not going to be selected. Meaning, the incisives will be placed at one of the middle 4 areas, out of the division of 8 made.

6 Fit the model

The algorithm used can be found in the paper of Cootes et al. *An introduction to Active Shape Models.*

Starting with the initial estimate computed previously as X_i

- **Find the best nearby match** — First, we sample a profile m pixels either side of the current point (with $m > k$, being k the number of pixels used to sample either side

1. Examine a region of the image around each point X_i to find the best nearby match for the point $X_i{}'$
2. Update the parameters (X_t, Y_t, s, θ, b) to best fit the new found points X.
3. Apply constraint to the parameters to ensure plausible shapes.
4. Repeat until convergence.

Algorithm 2: Fitting algorithm

of each landmark when building the Grey Shape Models). Then, we test the quality of fit of the corresponding grey-level model at each of the $2(m-k)+1$ possible positions along the sample and choose the one which gives the best match. The best match is calculated using the Mahalanobis distance.

- **Update the parameters** — The parameters (`b`, translation, scale and rotation) must be updated to generate a new estimation of the object, using Algorithm 3, as explained in *An Introduction to Active Shape Models.*
- **Apply constraints** — To ensure that the generated shapes are valid, some constraints must be applied to the different paramenters. The shape parameter `b` is allowed to vary at most three times the standard deviation. Scaling is also limited to the range `0.8 - 1.2`, and rotation is allowed in the range $\pm\pi/6$ with respect to the mean shape.

1. Initialise the shape parameters, b, to zero (the mean shape).
2. Generate the model point positions using $x = \bar{x} + Pb$
3. Find the pose parameters (X_t, Y_t, s, θ, b) which best align the model points x to the current found points Y.
4. Project Y into the model co-ordinate frame by inverting the transformation T.
5. Project y into the tangent plane to $\bar{x}$ by scaling: $y' = y/(y \times \bar{x})$
6. Update the model parameters to match to y'
7. If not converged, return to step **2**.

Algorithm 3: Fitting algorithm

7 Results

For the evaluation of the segmentation, as proposed in the project statement, leave-one-out has been used. In this technique, the data of the tooth we are trying to locate and segmentate is left out of the trainign set, and it is used to test the obtained result.

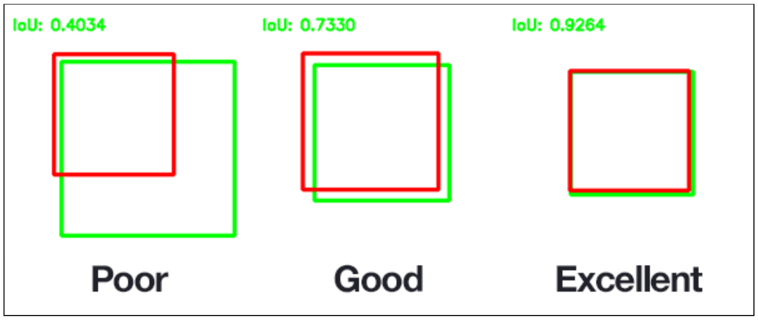

Figure 11: Index of Union

But we need to know how good we have segmentated a given tooth with respect to the ground truth. For this, the Jaccard index has been used. The Jaccard Index or Jaccard coefficient is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

The result will be a number between 0 (no overlap) and 1 (perfect segmentation).

7.1 Manual initialization

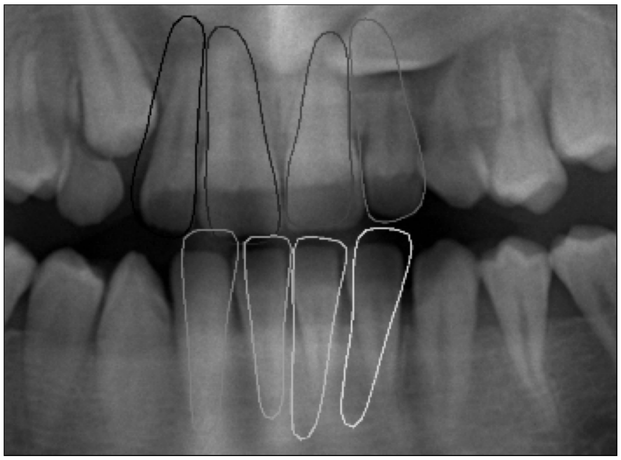

Figure 12: Result (manual initialization)

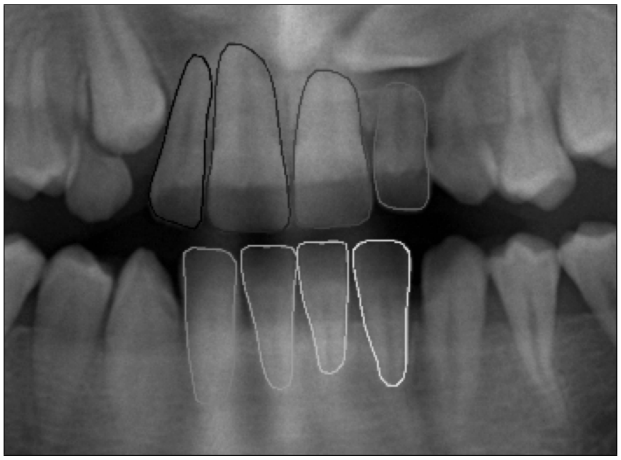

Figure 13: Ground truth

It can be seen that, in general, the crown of each tooth is approximated well, but the root is not. This is caused by how the fitting is done. As we use Grey Level Models, in the enhanced images the contour of the tooth is much more defined in the crown.

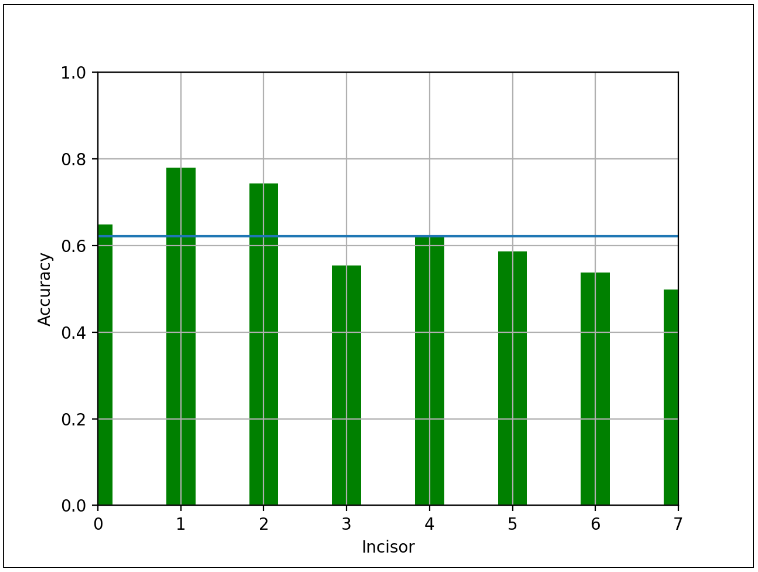

Figure 14: Accuracy of the segmentation

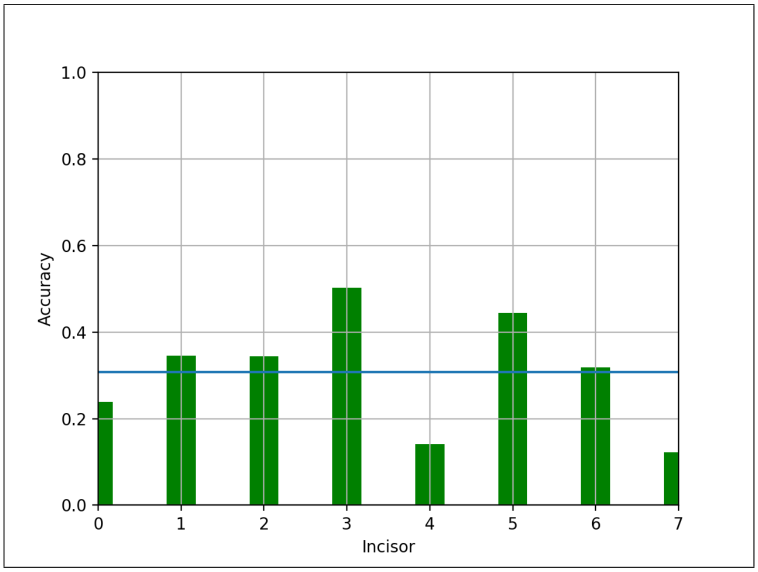

Figure 15: Accuracy of the segmentation

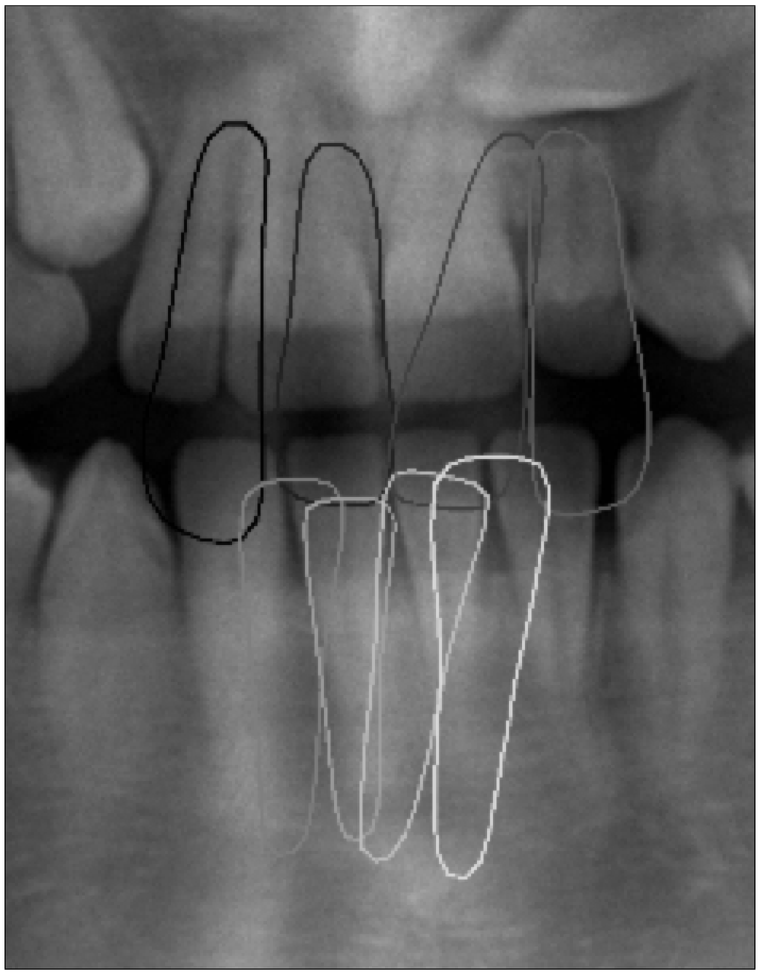

Figure 16: Result (automatic initialization)

7.2 Automatic initialization

8 Future Improvements

8.1 Multi-resolution Active Shape Model

To improve the efficiency and robustness of the detection, a multi-resolution model can be used.

At the base of the pyramid it is the original image and the level is the lowest (level 0). The image in the higher level is formed by subsampling the former image then we obtain a lower resolution version of the image with half number of the pixels along each dimension. Subsequent levels are obtained by further subsampling.

With this method, the initial fit of the model to approximate will vary faster in position and orientation, and this fit will be improved and refinated in each level of the pyramid.

8.2 Width of Search Profile

In the implemented algorithm, search profile is only along a single line when searching forthe suggested movements of landmarks. But this search technique could be affected by noise in the image. To avoid this problem, the search space could be expanded adding some width to it.

8.3 Landmarks Grouping

As we have seen new allowed shapes should comply with the constraint that the range of b must be in the inverval $-3\sqrt{\lambda} \leq b \leq 3\sqrt{\lambda}$, where λ is the eigenvalue result of the principal component analysis. But some of the shape variation isnt decided by a single principal component, and unexpected shapes could still apear even with the range of b limited. This problem can be avoided grouping ladmarks that tend to move together.