Assignment 3 - Medical Image Analysis

2020/2021

Francesco Corti Julie Mazzella
12002944 12040967

Graz, May 30, 2021

Contact: Francesco Corti Julie Mazzella, corti@student.tugraz.at julie.mazzella@student.tugraz.at

Abstract

The following report aims to illustrate the work done for the third assignment of the Medical Image Analysis. Two image registration methods were implemented through a Python code and multiple trials were applied on two images. The variation of the parameters lead to the analysis of the variation of the error in both algorithms. What was observed is that the choice of the parameters is fundamental to avoid the local minima problem in the achievement of the solution.

1 Introduction

Registration is the process of determining a transformation T between the coordinates in one space and those in another, such that points in the two spaces that correspond to the same anatomical point are mapped to each other.
In particular, the goal is to find a mapping that aligns a moving image with a second fixed image, such that a defined similarity measure is maximized. The fixed image is used as a comparison, while different transformations are applied on the moving image. The best transform is obtained through cost function optimizations and similarity metrics. However, registrations methods are non-convex optimization problems, so an optimal solution is harder to obtain.

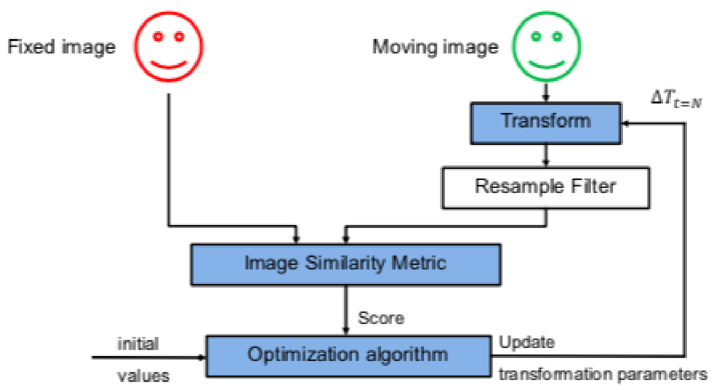

Figure 1: Registration's algorithm

Transformations can be either rigid or non-rigid. In the former, only displacement and rotation functions are applied, thus preserving the length and size of the transformed object.
In the latter, on the other hand, both parametric similarity (affine, piecewise-affine transformations) and non-parametric applications can be done, involving the change of the shape of the original image.

Considering rigid methods, image registration algorithms can be feature-based, surface-based or intensity-based.

1.1 Feature-based Rigid Registration

The most common feature-based registration method is the Point-based algorithm.
Given two sets of corresponding points $P = \{p_i\}$ and $Q = \{q_i\}$, the algorithm finds the rigid-body transformation (the rotation of the matrix R and the translation of vector t) that minimizes the L2 loss between the points:

$$E = \frac{1}{N} \sum_{i=0}^{N} \|Rp_i + t - q_i\|_2^2 \tag{1}$$

where E is the registration error.
The solution to this problem is based on a 3-steps process. First, both sets of points are centered by removing their center value. Then, rotation is determined through the Procrustes Problem: the objective is to find the orthogonal matrix R that rotates the matrix P into matrix Q, such that the Frobenius norm (i.e. L2 norm) of $PR - Q$ is minimized:

$$\min_R \left\{ \|PR - Q\|_F \right\}, s.t. R^T R = 1 \tag{2}$$

This is equivalent to maximizing $tr(R^T P^T Q)$, which is achieved by computing the singular value decomposition (SVD) of $P^T Q$ and taking $R = VU^T$.
Since R is an orthogonal matrix, the multiplication of a matrix by R does not influence its Frobenius norm.
The SVD of $P^T Q$ leads to a decomposition into two orthogonal and a diagonal matrix: $P^T Q = U\Sigma V^T$. Following the trace properties, it is possible to get:

$$tr(R^T P^T Q) = tr(R^T U\Sigma V^T) = tr(V^T R^T U\Sigma) \tag{3}$$

where $V^T RU = I$, so that the solution is: $R^* = VU^T$.
The third and last step is to determine the translation: the center of the set of points P is translated in the center of the set of points Q:

$$t = \bar{q} - R\bar{p} \tag{4}$$

In general, point-based registration uses anatomical landmarks or extrinsic markers, and aligns small number of points. Also, it requires knowledge of point correspondence.

1.2 Surface-based Rigid Registration

The 3D boundary of an object is an easily characterized geometrical feature that can be used for registration. It generally aligns a large number of points, with unknown correspondence. Surface-based methods imply the determination of corresponding surfaces in different images and the finding of a transformation that best aligns these surfaces. To do so, segmentation is necessary for surface extraction.
Given a set N_P of surface points $\{p_i\}$ and a surface Q, the problem wants to find the rigid-body transformation T (rotation matrix R and translation vector $\vec{t}$) that minimizes the mean squared distance between the points and the surface:

$$d(T) = \frac{1}{N_P} \sum_{i=1}^{N} {}_P \|T(p_i) - q_i\|_2^2 \tag{5}$$

where the points $\{q_i\}$ are the corresponding points to the points $\{p_i\}$ that lie on the surface Q:

$$q_i = C(T(P_i), Q), C = correspondencefunction \tag{6}$$

The Iterative Closest Point Algorithm is the most common surface-based registration technique and consists of a 2 stage iterative algorithm that looks for the closest points on a surface.
To register the data shape P to the model shape Q: first it is necessary to decompose P into the point set $\{p_i\}$, then to establish the initial registration T_0 as $p'_i = T_0(p_i)$. Afterward, the set of closest points $\{q_i\}$ on Q is computed as the points that minimize:

$$d(P'_i, Q) = \min_q \{\|q - P'_i\|_2\} \tag{7}$$

Then, the transformation T_j is obtained by computing the optimal registration of the corresponding point sets $\{p_i\}$ and $\{q_i\}$. Finally, the transformation T_j is applied to transform the point set $p_i = T_j(p_i)$. The process is repeated until convergence is reached.
This algorithm is representation independent, meaning that the data shape P is converted to a point set and the model shape Q can be an arbitrary surface, with the only restriction that the distance of a point to the surface must be efficiently computable.
One big disadvantage of this algorithm is that segmentation is an important issue, that increases complexity. Also, since it is not a convex problem there might be some local minima: this can be solved by trying random initial guesses. Finally, the least-squares norm is not robust enough to avoid outliers, so the L1 norm can also be considered. However, by doing so, the solution of Procrustes alignment fails.

2 Methods

The aim of the assignment is to implement two image registration algorithms: the Procrustes and the Iterative Closest Point methods. To do that a Python code was implemented, using Numpy as scientific computing library.

2.1 Procrustes algorithm

As stated in Section 1.1, the implementation of the Procrustes method works in 3 steps. First, the two sets of points were centered by removing their mean. Then, they were normalized by dividing them for their Frobenius norm.
Afterward, the SVD was applied on the matrix $H = P^T Q$, to obtain the matrices U, Σ and V^T. The matrix D was also computed as the diagonal matrix containing the sign of the determinant of UV^T. Finally, the rotation matrix R was obtained by multiplying V, D, and U^T. Following this second step, the computation of the scaling factor was achieved by dividing the Frobenius norms of both sets of points. Then, the transformation matrices for P and Q were defined, together with the scaling matrix and the rotation matrix in the homogeneous coordinate system.
The last step is to compute the transformation matrix T as the product of the previously defined elements. The transformation T was then applied on the sets of point P in the homogeneous coordinate system as follows: $P' = (TP^T)^T$. At last, the P' points were transformed in the Cartesian coordinate system.

2.2 Iterative Closest Point algorithm

As described in Section 1.2, the Iterative Closest Point algorithm consists of 2 iterative stages. First, P is decomposed into the set of points $\{p_i\}$. The initial registration $T_0 : p_i' = T_0(p_i)$ is established.
Secondly, the set of closest points $\{q_i\}$ on Q is obtained by using the euclidean distance, until the error obtained is not under a predefined threshold, formally $q_i = \underset{q \in Q}{argmin} \|q - p_i^{'}\|_2^2$. After that, by using the Procrustes algorithm, the optimal registration of corresponding point sets $\{p_i\}$ and $\{q_i\}$ is retrieved, giving a transformation matrix T_j. In the end, the transformation matrix is used to transform the point set $p_i^{'} = T_j(p_i)$. This stage is repeated until a low error is obtained.

3 Results

The two image registration algorithms were both applied to the following images:

Figure 2: First image

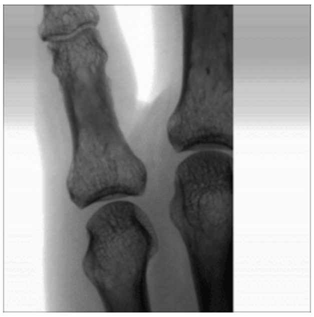

Figure 3: Second image

The result of the Procrustes registration algorithm is:

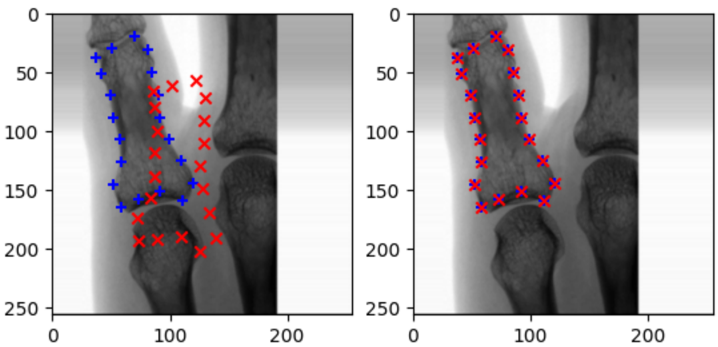

Figure 4: Result of the Procrustes registration algorithm

While the results of the Iterative Closest Point registration algorithm are:

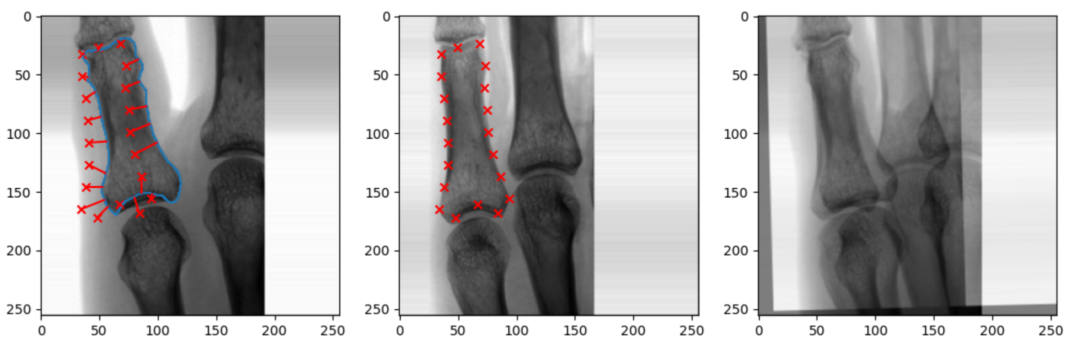

Figure 5: ICP algorithm result after 1 iteration

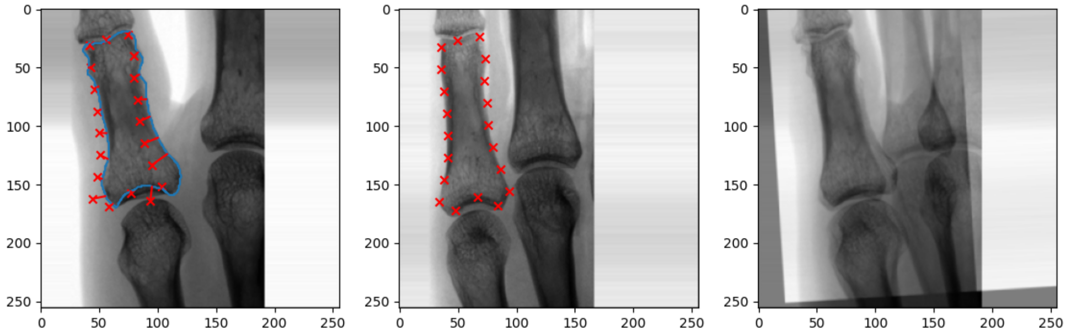

Figure 6: ICP algorithm result after 2 iterations

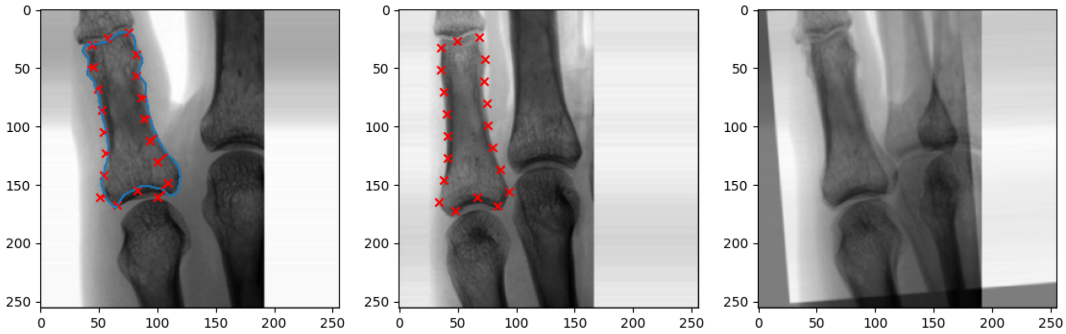

Figure 7: ICP algorithm result after 3 iterations

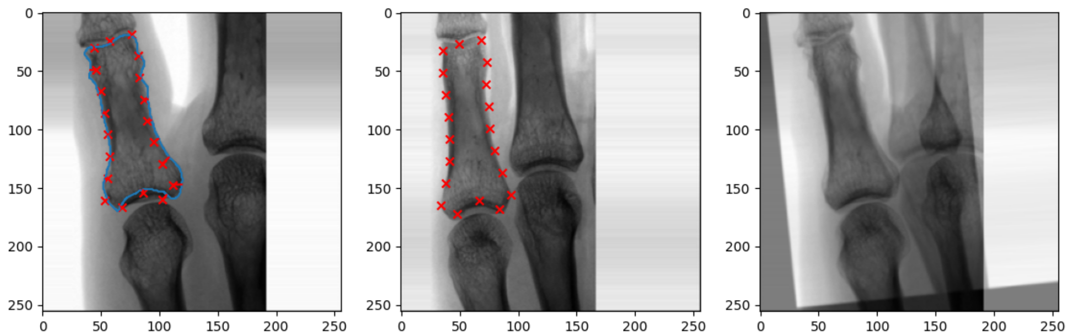

Figure 8: ICP algorithm result after 4 iterations

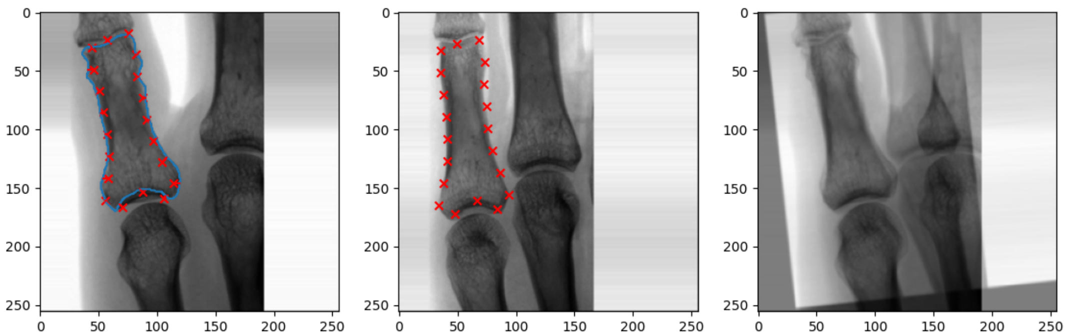

Figure 9: ICP algorithm result after 5 iterations

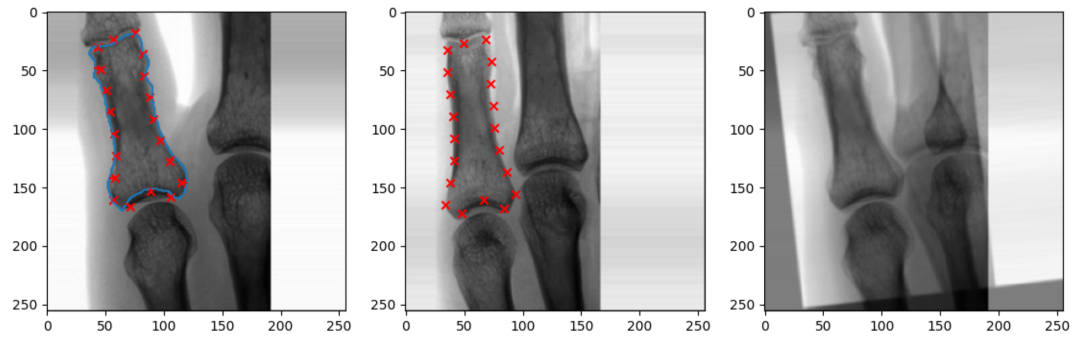

Figure 10: ICP algorithm result after 6 iterations

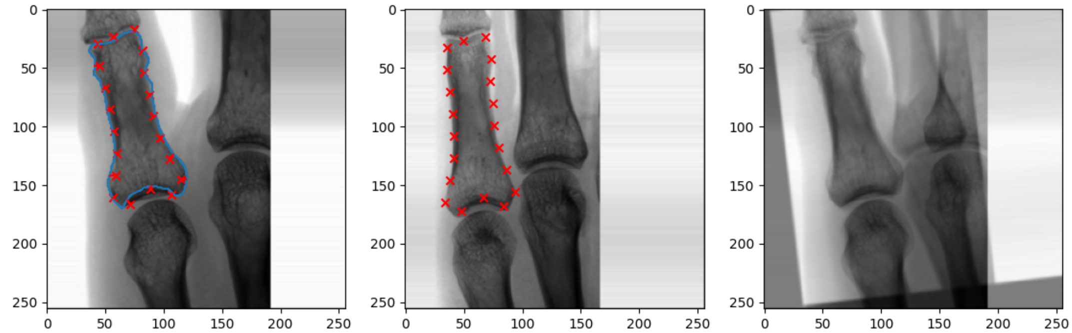

Figure 11: ICP algorithm result after 7 iterations

The errors plots obtained for the ICP algorithm with different translate and scale values are shown below:

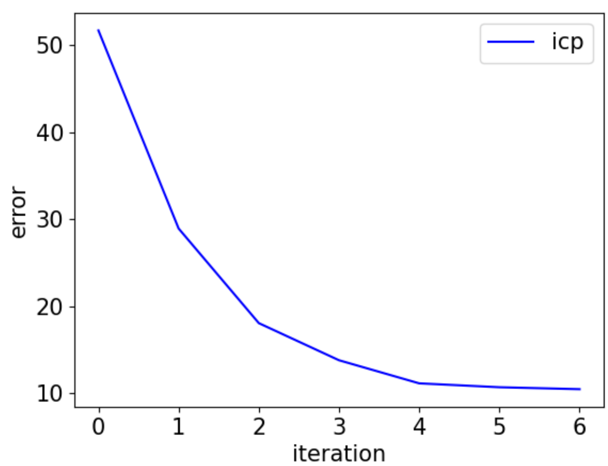

Figure 12: ICP error, translation equal 0 and scale equal 1

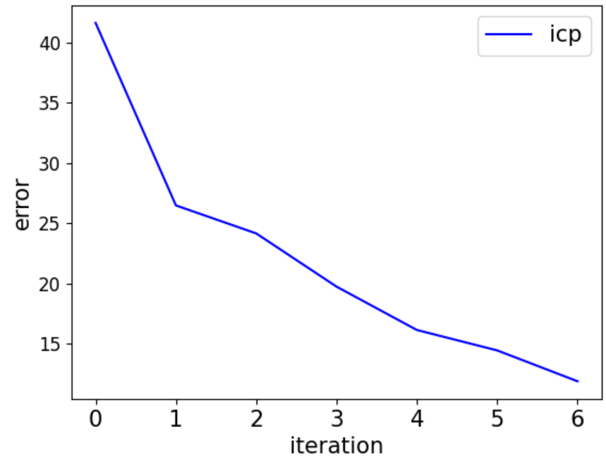

Figure 13: ICP error, translation equal 10 and scale equal 1

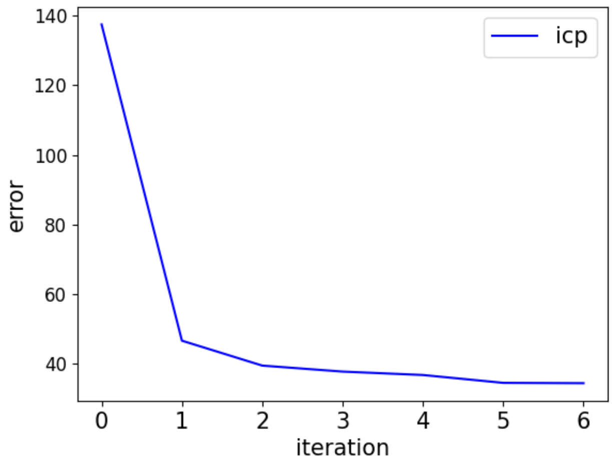

Figure 14: ICP error, translation equal 50 and scale equal 1

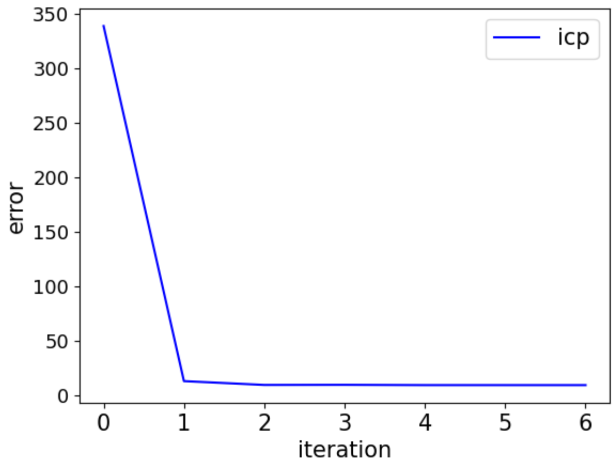

Figure 15: ICP error, translation equal 100 and scale equal 1

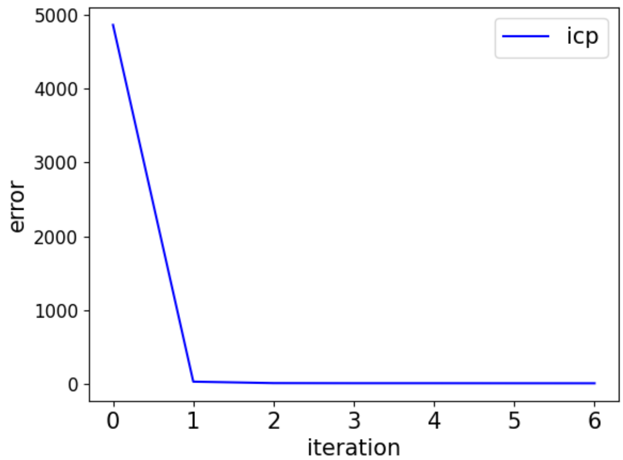

Figure 16: ICP error, translation equal 0 and scale equal 10

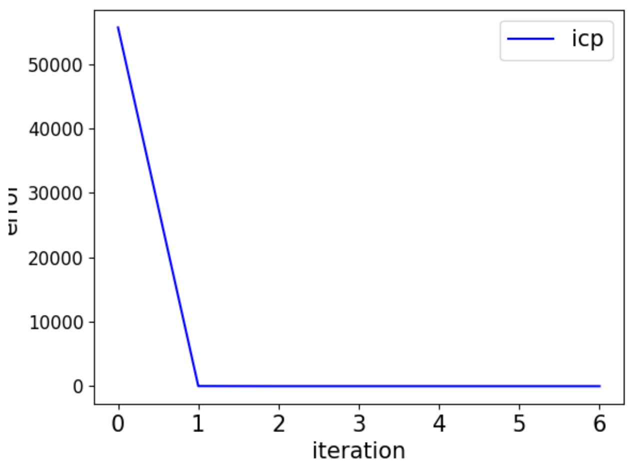

Figure 17: ICP error, translation equal 0 and scale equal 100

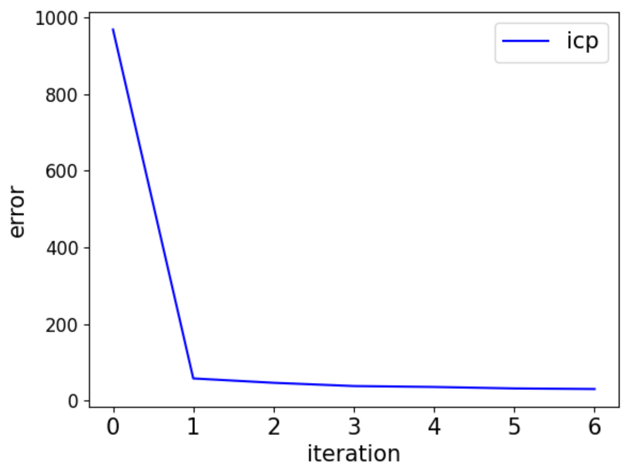

Figure 18: ICP error, translation equal 0 and scale equal 3

4 Discussion

The difference between the Procrustes Alignment and the Constrained Procrustes Alignment is that the Procrustes Alignment finds an orthogonal matrix R that rotates matrix P into matrix Q. In the Constrained Procrustes Alignment, on the other hand, only a rotation matrix R is accepted as a solution to the alignment problem, meaning that its determinant must be one.

In the second task, the Iterative Closest Point algorithm is used because a small set of points is not passed as input like in the previous task. Instead, a surface Q that contains 749 points is considered. Hence, the algorithm for each iteration has to compute the closest points from the data points P to the model shape Q and then use the Procrustes method to find the transformation matrix that is applied iteratively, to find a solution of the alignment problem.

As it can be seen from Figure 12 and Figure 13, the more the translation parameter increases, the more the error decreases faster to a lower value. In fact, when the translation parameter is equal to 10, at the first iteration the error reaches a value of more or less 25, compared to the value of 30 of the behavior with null translation.
However, increasing more the translation parameter, the error converges to a higher value, as it can be seen in Figure 14 and Figure 15.

Regarding the scaling factor, what can be observed is that for values between 2 and 5 the ICP algorithm fails, converging to a high error value. When the scaling factor is higher, on the other hand, the error converges to a value around 10. The reason for this behavior can be that there is no correspondence between points, and the best fitting is chosen iteratively by the Procrustes method. Hence, the algorithm probably converges to local minima, making the alignment wrong.

5 Conclusion

Image registration is an issue that depends on the initial conditions of the problem. If there is a large number of points available and there is no previous knowledge about point correspondence, the ICP algorithm is the best solution to face the issue. However, when the number of points is not large enough and previous knowledge about point correspondence is available, the Procrustes method achieves better results.
Since the optimization problem is a non-convex one, the scaling and the translation parameters have to be chosen wisely, because a wrong choice can lead to local minima and thus to the failure of the algorithm.