Visual Scan of Human Body

Code repository link: https://github.com/jfraszczak/3D-Reconstruction

Jakub Frąszczak 10832319, Image Analysis and Computer Vision 2021/2022

Problem formulation

Goal of the project is a 3D reconstruction of a human shape from images taken from different angles. Due to the presence of specific landmarks like visible rectangular markers and the fact that human body is covered by flexible costume with multiple crossing lines it is possible to exploit geometrical properties in order to reconstruct mentioned shape.

This aspect of computer vision has gained relevance for example in case of autonomous driving, robotics or medical diagnosis where stereo cameras might be used to estimate depth and three dimensional model of a surrounding area.

Outline of the current state of the art methods

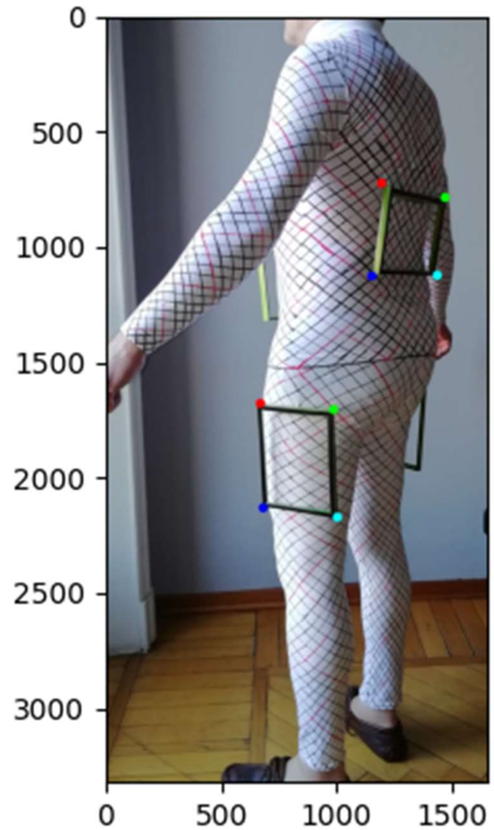
Currently, research which is being conducted puts a significant emphasis on application of deep learning methods in an area of computer vision. These methods, despite the need of data collection and extensive training turn out to be particularly successful and have multiple practical applications in the industry. Plenty of approaches might be used depending on the exact problem formulation, from purely deep learning based methods like use of convolutional neural networks as an encoder decoder architecture mapping images onto the 3D space to the approaches combining deep learning with conventional geometric methods. For example corresponding points from a series of images might be matched automatically with a use of deep learning model in order to then use these points to obtain a dense 3D representation.

Brief summary of approach taken

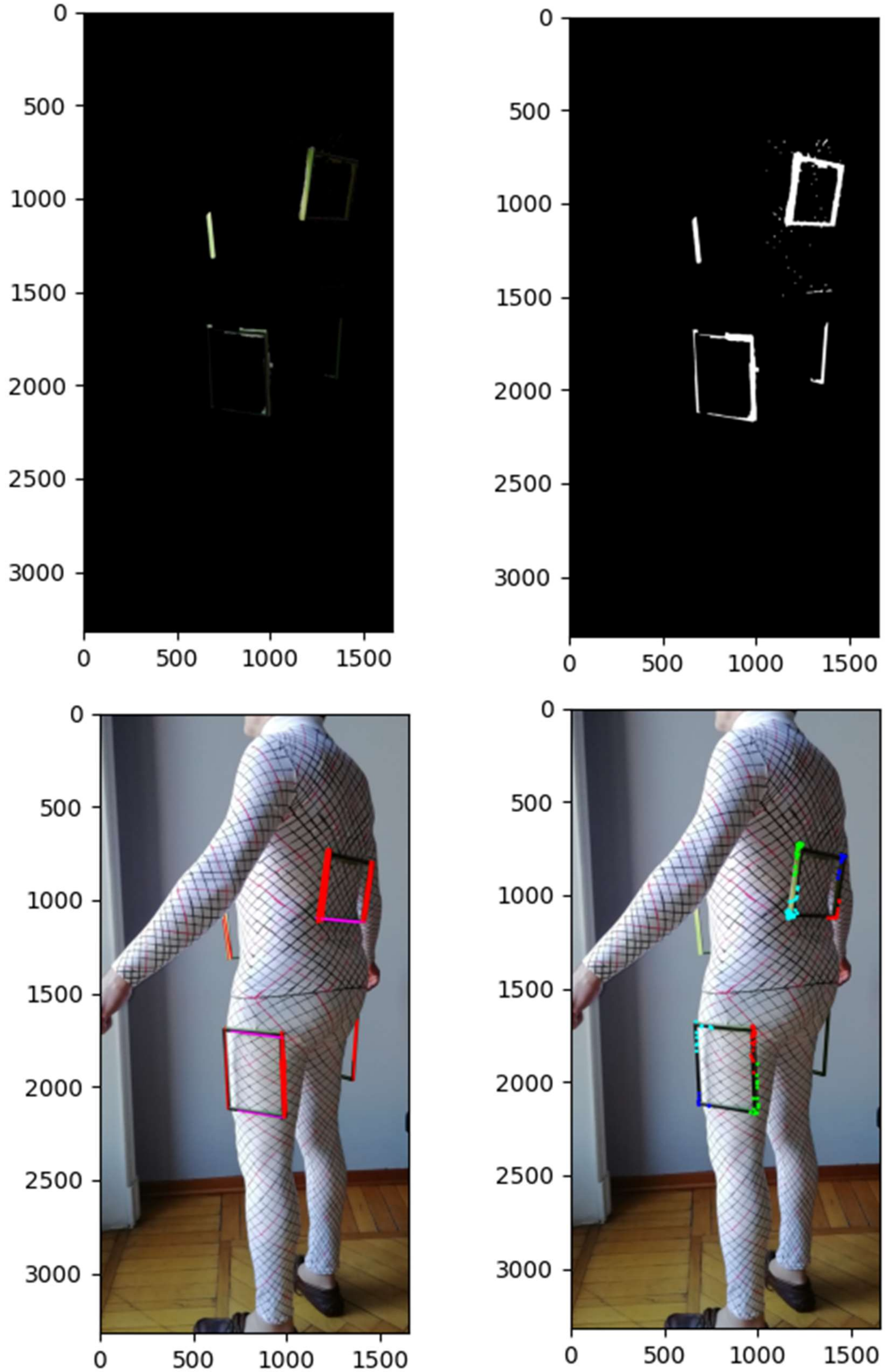
All in all, implementation of the project was strictly based on the approach proposed by the Professor at the very beginning after selection of the topic. Key points are as the following:

1. **Detection of rectangular markers**
   In every picture there are visible markers whose dimensions are known. This step is required in order to perform camera calibration.
2. **Camera calibration**
   Thanks to the information concerning dimensions of the markers it is possible to compute homographies for all the markers collected from available images. Afterwards, computed homographies can be used to calibrate the camera using Zhang algorithm.
3. **Identification of crossings**
   Following step includes detection of curves visible on the human body and identification of crossings between these curves.
4. **Crossings matching**
   In order to be able to perform 3D reconstruction it is essential to find corresponding pairs of points from the two neighbouring images.
5. **3D reconstruction**
   Having computed intrinsic camera matrix and corresponding pairs of points from neigbouring images it is possible to firstly compute essential matrix and then triangulate corresponding pairs in order to obtain 3D reconstruction of extracted points.

Rectangular markers detection



1. **Markers detection**
   Due to the fact that rectangular markers significantly differ in terms of color from the rest of the image it was possible to rather easily extract them. Firstly, images were transformed into HSV color model since it is more intuitive and it enables to conveniently apply filter, selecting only pixels from the specified range of colors.
2. **Masks processing**
   Once only green and dark black pixels were selected morphological opening was applied in order to minimize noise but at the same time preserve shape of the obtained masks.
3. **Corners detection**
   At this step it was already possible to work on the detection of corners. The very first approach which was tried was Harris Corner Detection Algorithm, however, quality of obtained masks turned out to be too poor in order to successfully apply this method. Therefore, Hough Transform was used to find lines these markers consist of. Thickness of extracted edges causes detection of multiple lines along these edges. Lines which were found, were then filtered by selecting either vertical or horizontal lines. Then endpoints of segments were selected and clustered with use of Gaussian Mixture model. At the very end, centroids were calculated from these clusters of points in order to obtain final coordinates of corners.
4. **Identification of markers**
   Knowledge concerning which markers are visible on which image was embedded into the program, for example it was explicitly specified that only markers on the back and on the left leg are visible in the picture taken from the angle of 240 degrees. There was, however, no automatic mechanism verifying whether masks which were extracted represent fully visible marker or not. This aspect undeniably leaves a room for an improvement.


0
500
1000
1500
2000
2500
3000
0
500
1000
1500
0
500
1000
1500
2000
2500
3000
0
500
1000
1500
0
500
1000
1500
2000
2500
3000
0
500
1000
1500
0
500
1000
1500
2000
2500
3000
0
500
1000
1500

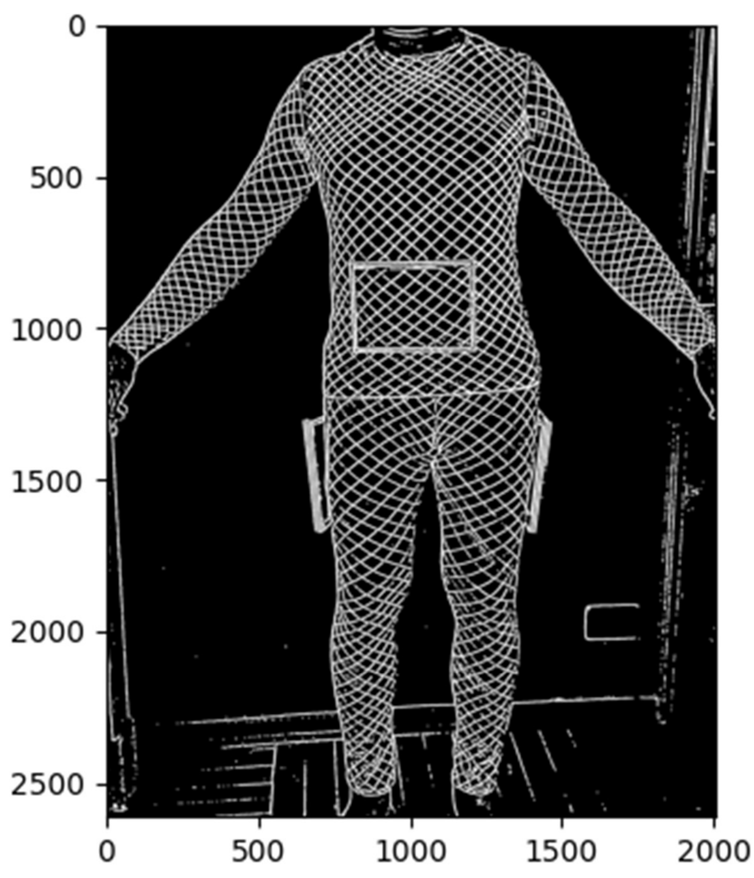
Camera calibration

Once coordinates of markers' corners were known it was possible to compute homographies since dimensions of the markers were known beforehand. All the homographies could be collected from all the pictures since they were taken with the same camera. The more homographies the more accurate results of camera calibration one should expect. After computation of homographies, Zhang algorithm was used to obtain intrinsic camera matrix.

Crossings detection

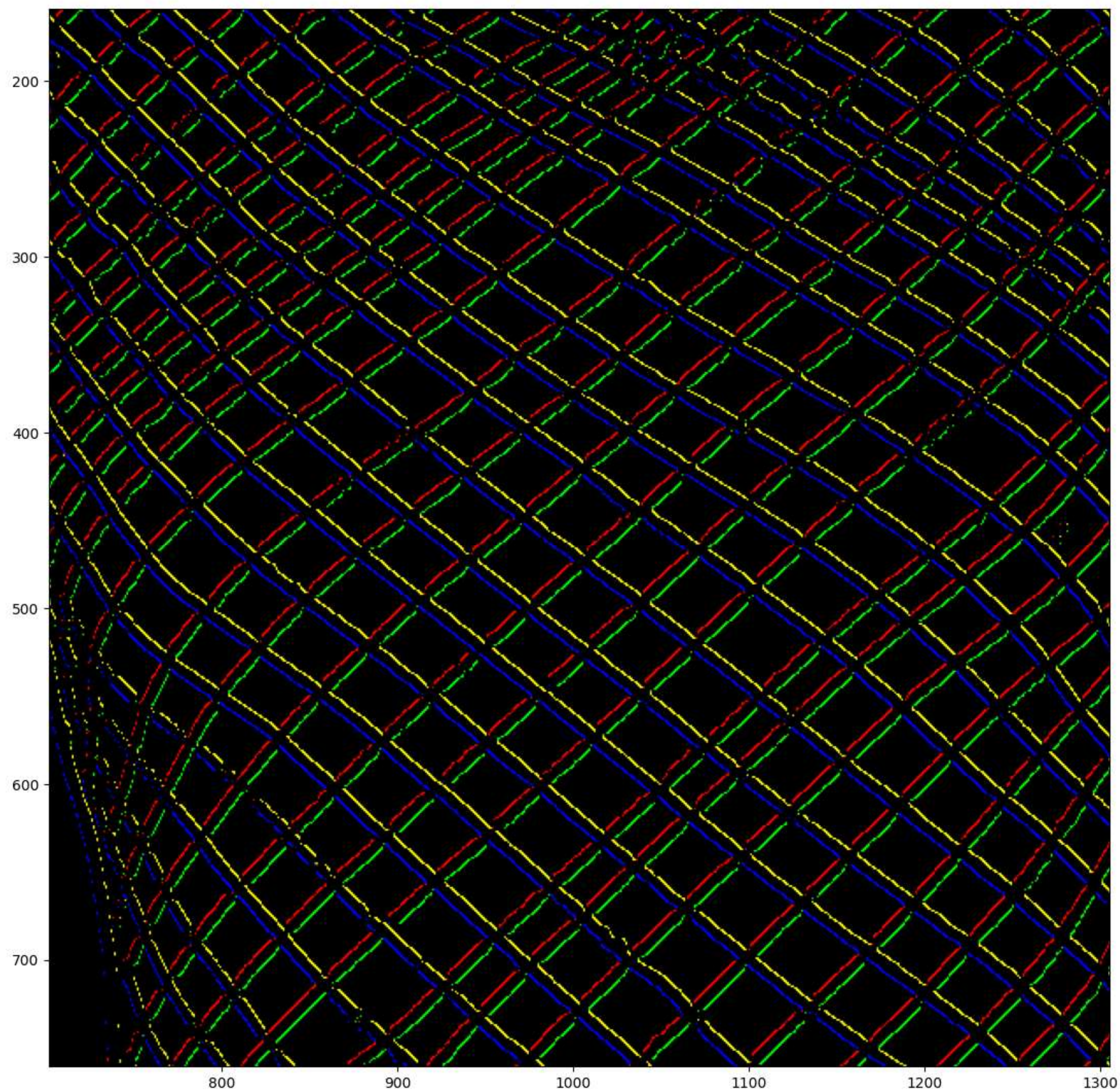
The following subsection goes through all the steps which were taken in order to find crossings visible on the human body.

1. **Image preprocessing**
   In order to extract lines covering the elastic suit following steps were taken:
   1. Image was transformed to grayscale
   2. Gaussian blur was applied in order to reduce potential noise
   3. Adaptive thresholding in order to extract mask of lines
   4. Morphological opening of the obtained mask



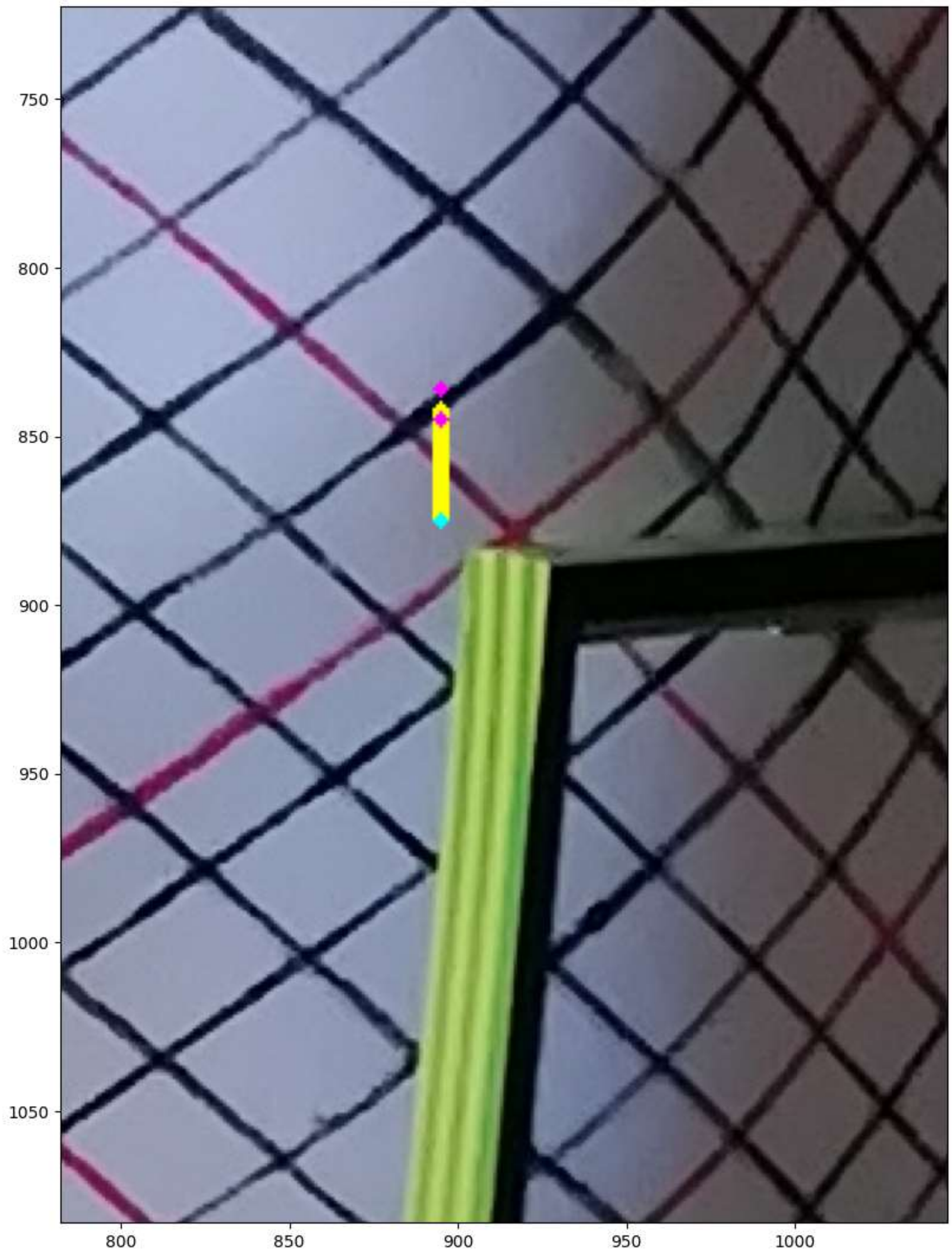
2. **Computation of gradients**

Afterwards, gradients were computed by an operation of convolution and angles of these edges were estimated. Image visible below represents these gradients and their corresponding angles. Different colors represent different range of angles.



3. **Reference starting point**

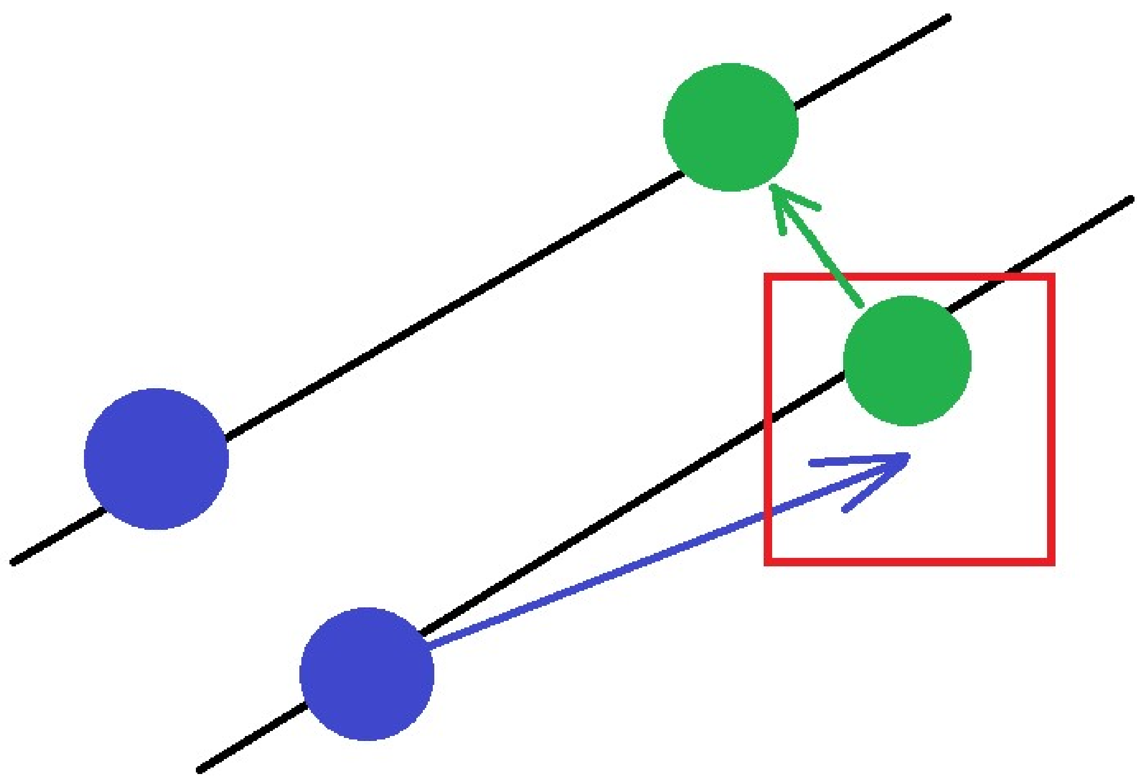
Already extracted corners of markers were used as reference points. From these points the area is being explored in order to find starting points. Potential starting points are being explored going up from the blue reference point along the visible yellow vertical line until gradients directed towards upper right are found. These points are highlighted with pink circles in the image presented below.



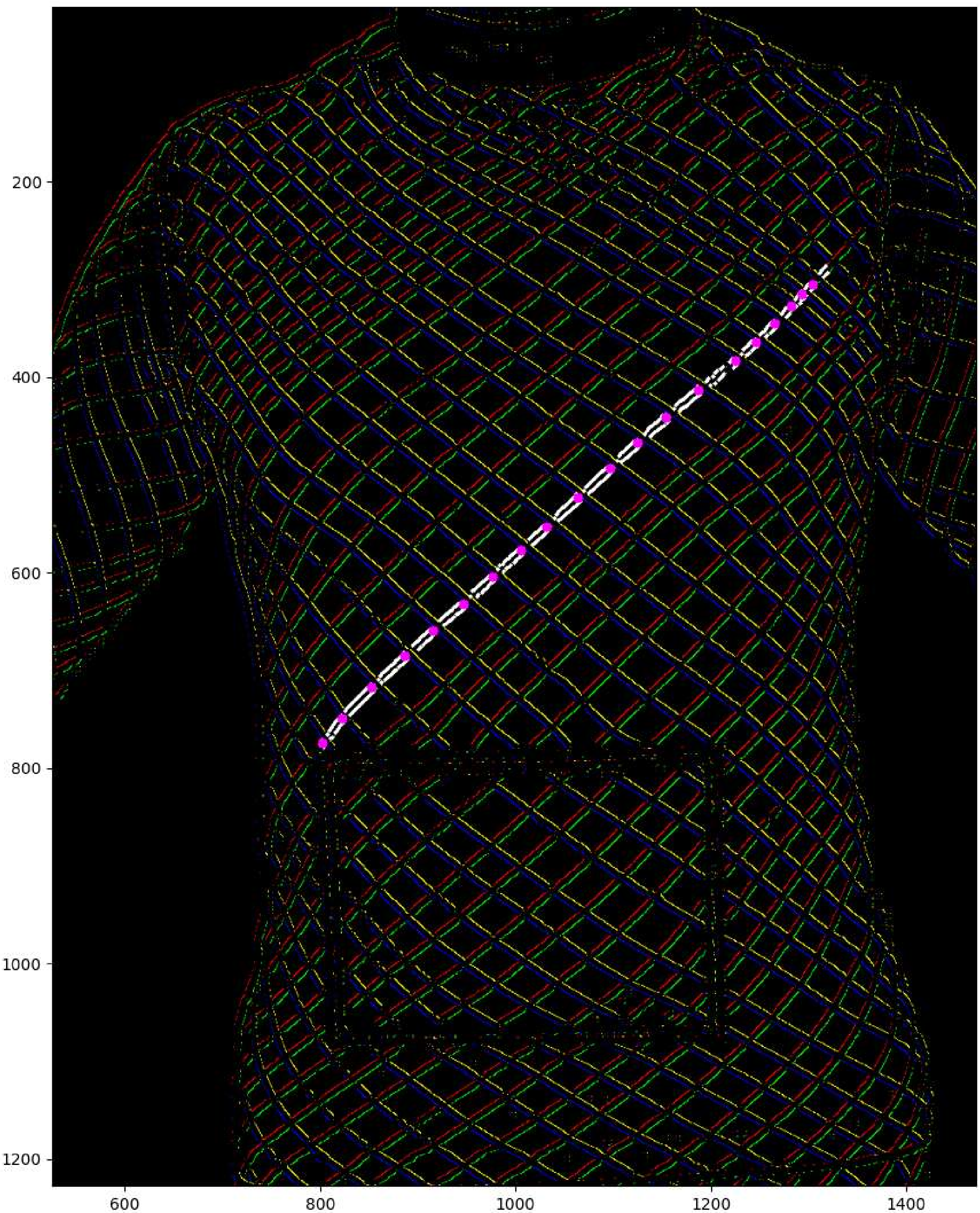
4. **Crossings detection**

First of all, once starting points have been found proposed algorithm keeps going along previously calculated gradients. Angle of a gradient is taken from a current point and new point along this angle is calculated. Estimated angle, however, most likely is not fully accurate that is why newly computed point might not lie on the gradient. Due to this reason nearest area is searched in order to find the closest point lying on the gradient. Next, an algorithm follows a line orthogonal to the first gradient until it crosses with the opposite gradient. In such a way new points are being obtained along the single line.

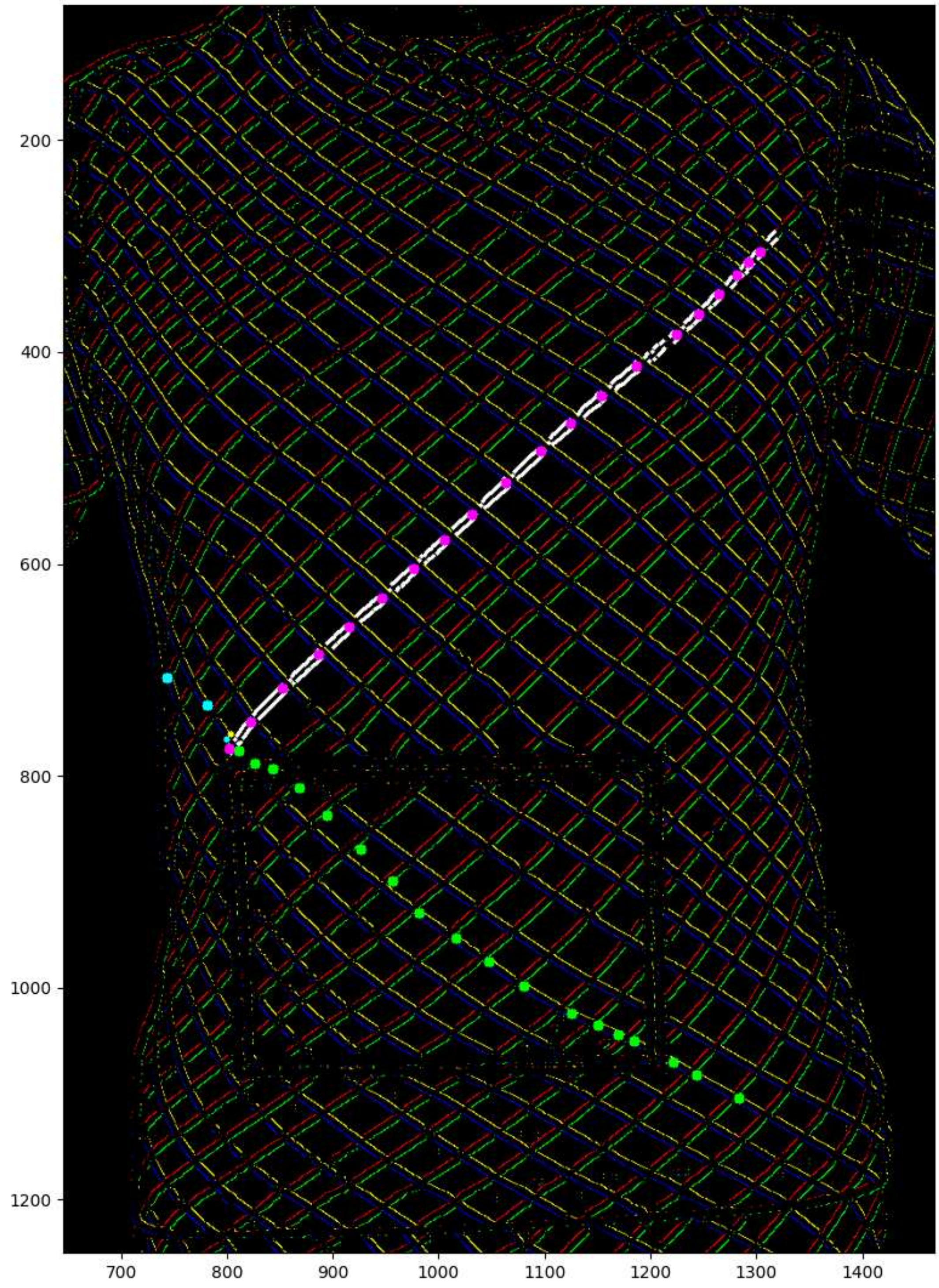
In the picture below blue points represent initial position of points, then new point is computed along an angle of gradient represented by a blue arrow and nearest area (red rectangle) is searched in order to find the closest point actually lying on the gradient. In the end, second point on the opposite gradient is found by following an orthogonal line (green arrow) from a first point.



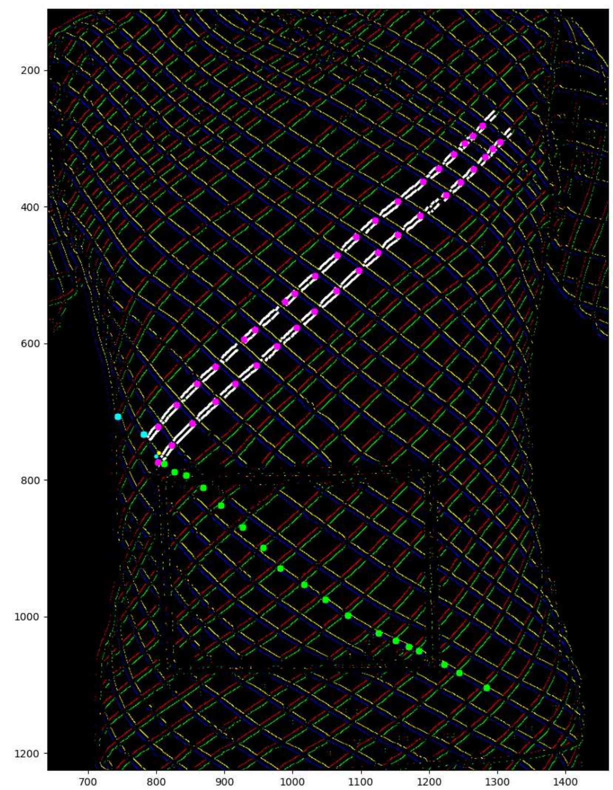
This process is performed iteratively along the whole line. In the meantime, for each point it is checked whether a crossing was found in the nearest area. Verification whether a crossing was found requires only checking whether gradients directed towards upper right and upper left cross each other. White points represent iteratively computed points along the gradients while pink points represent found crossings.

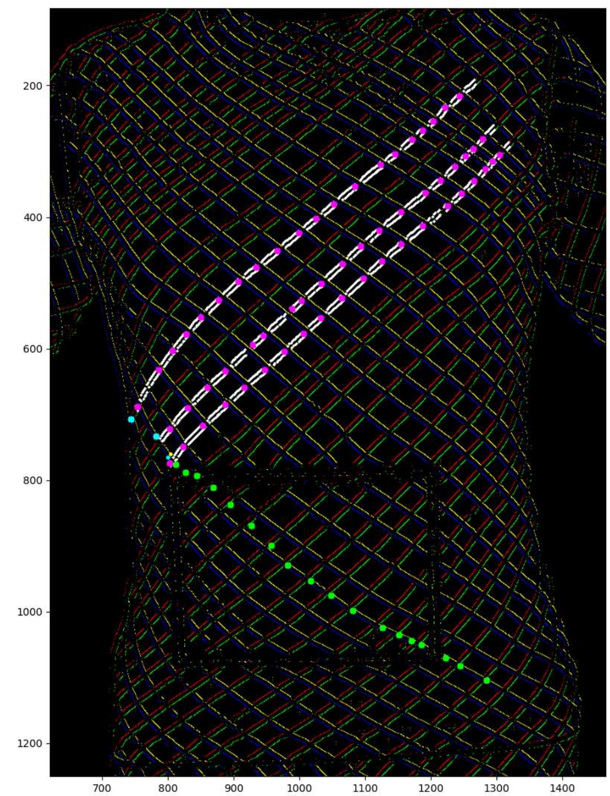


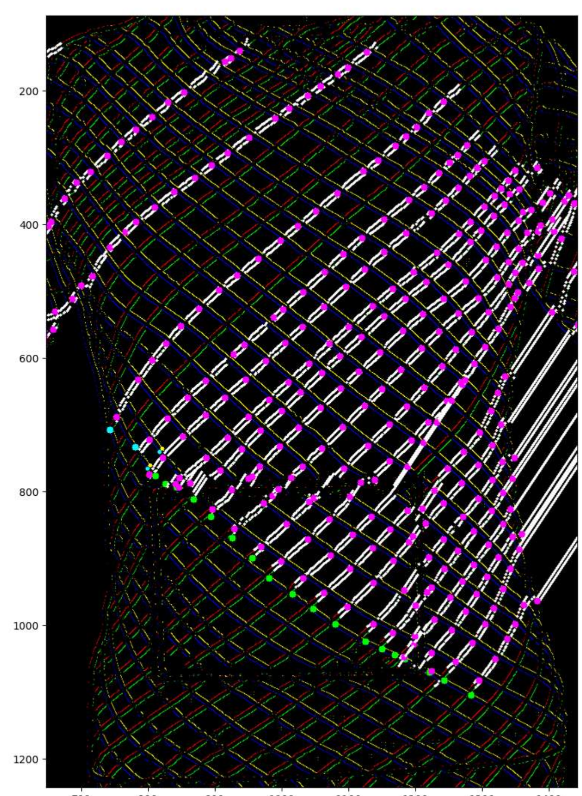
Analogous operation is performed along lines directed towards upper left.

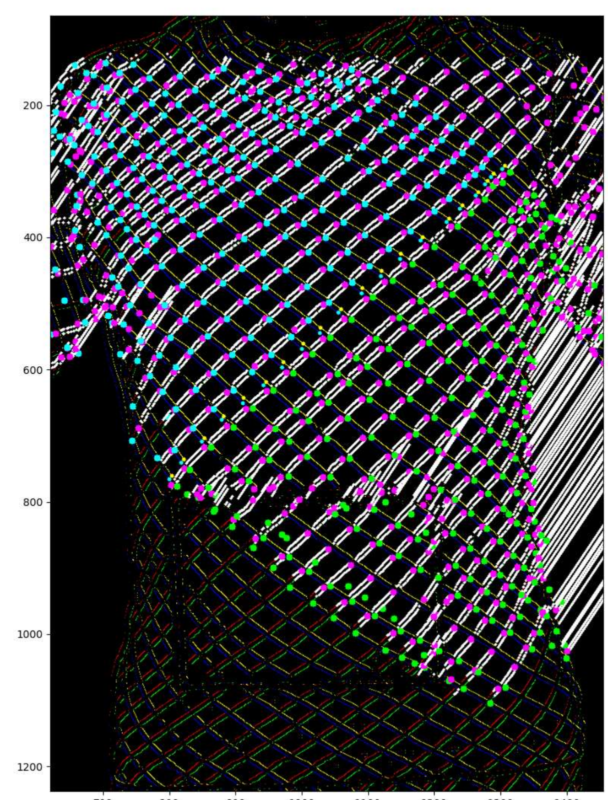


Crossings found along upper left lines are then used as starting points from which other lines are checked. As it can be noticed in the second and third picture, some lines might be missed that is why the whole process is repeated multiple times, starting from the first line directed toward upper left, then starting from the second one, third and so on.

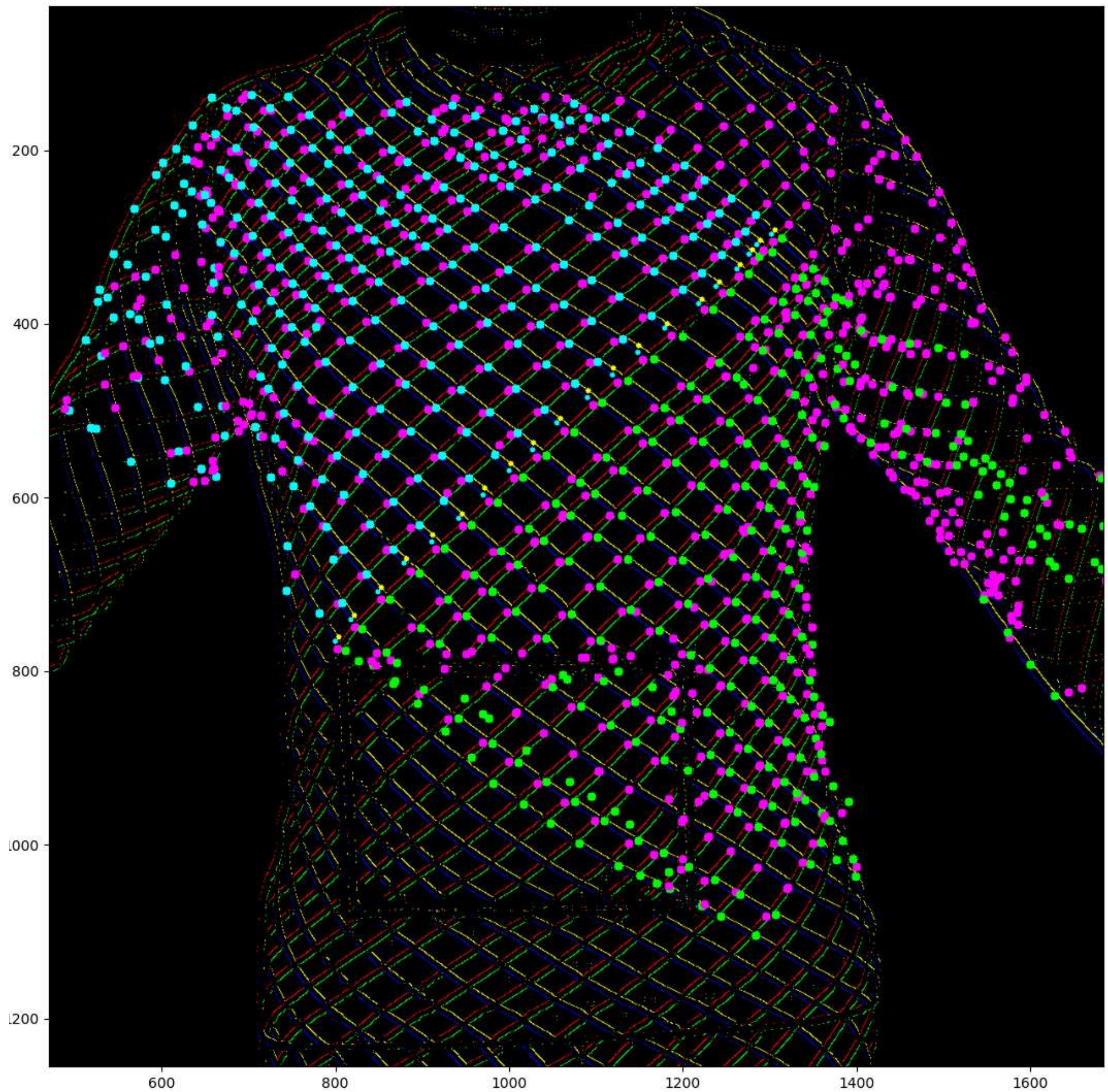


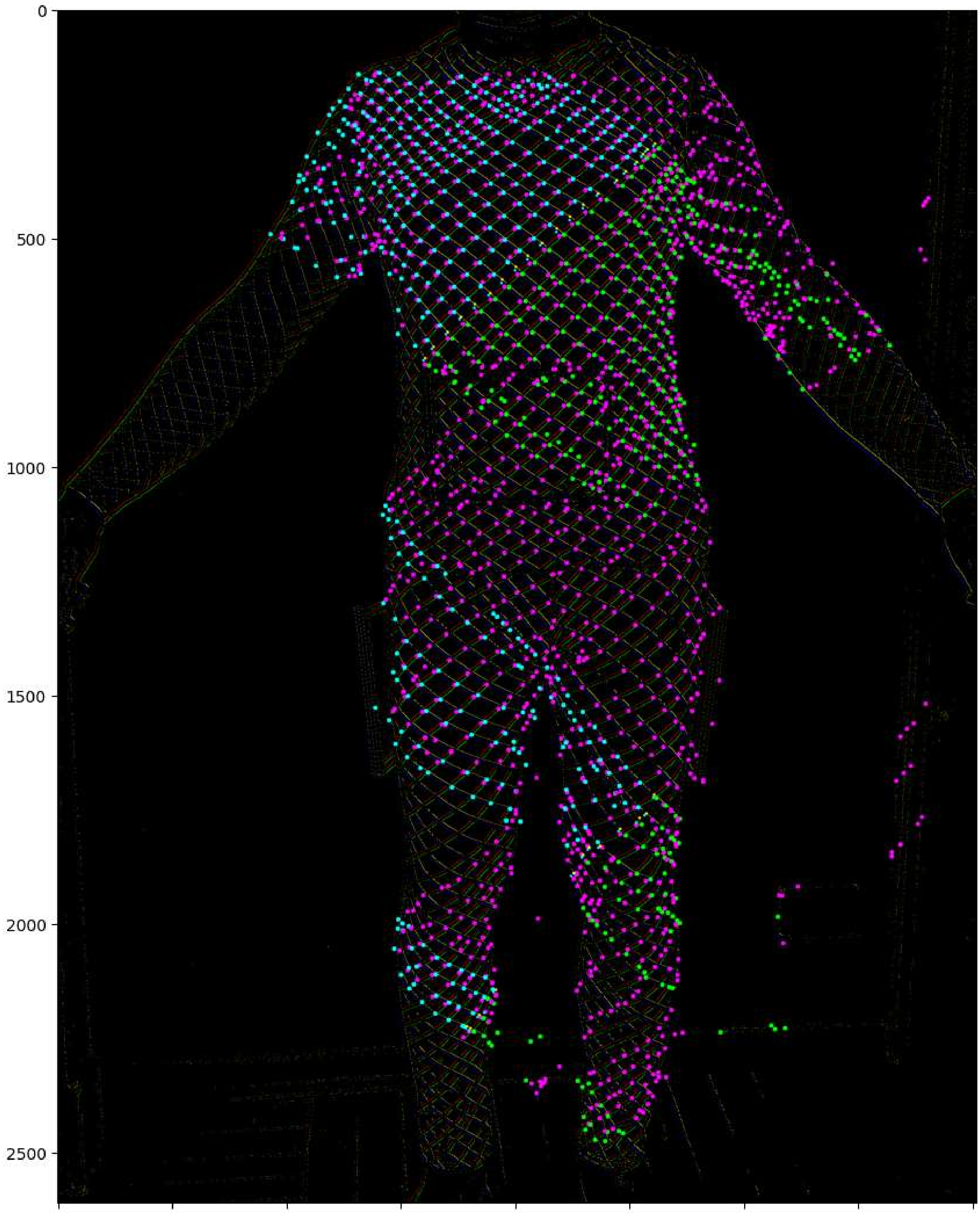






In the end results of crossings detection are rather satisfying however it is computationally expensive process due to multiple redundancies of following the same line multiple number of times.



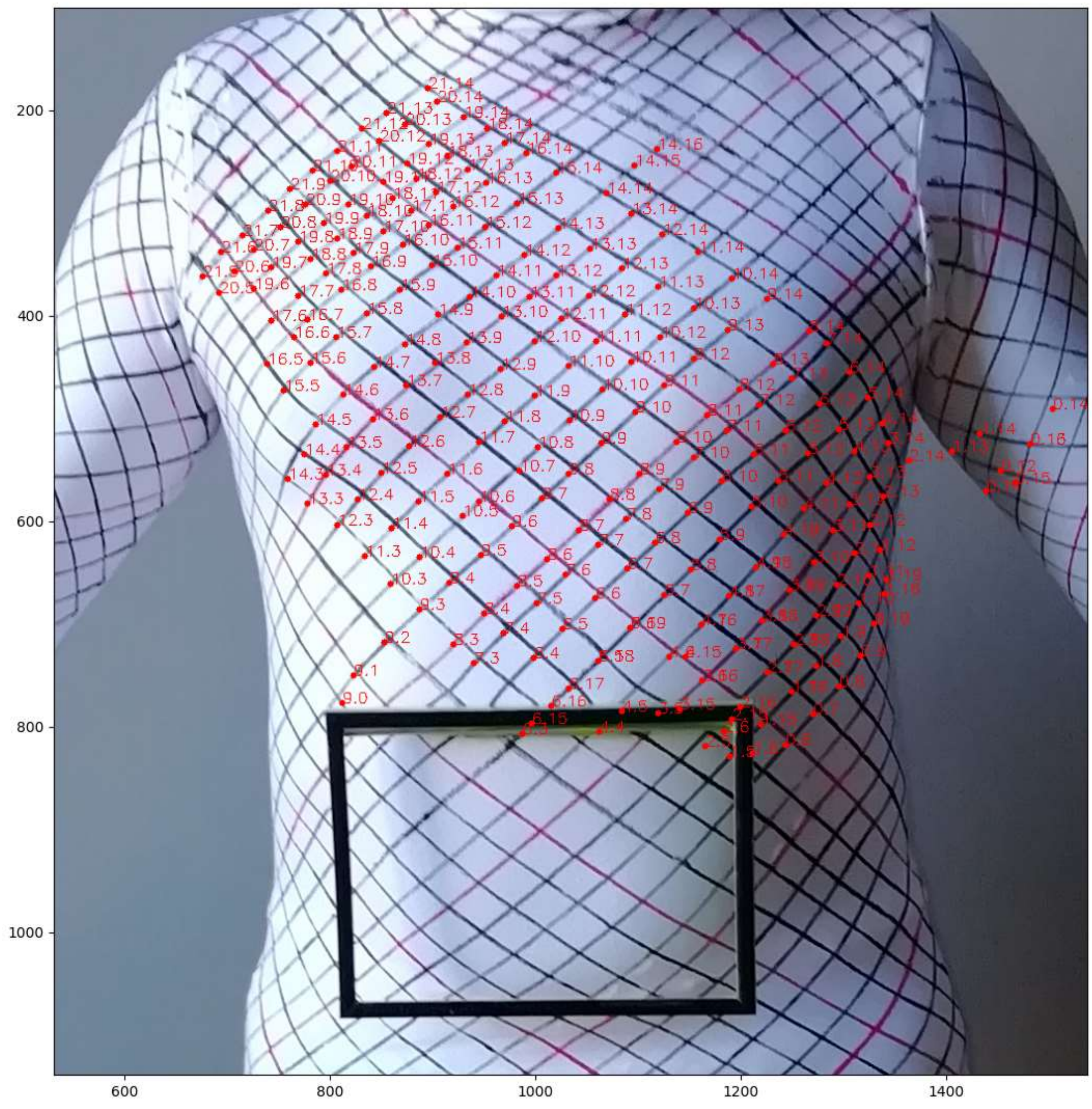

0
500
1000
1500
2000
2500
0
250
500
750
1000
1250
1500
1750
200

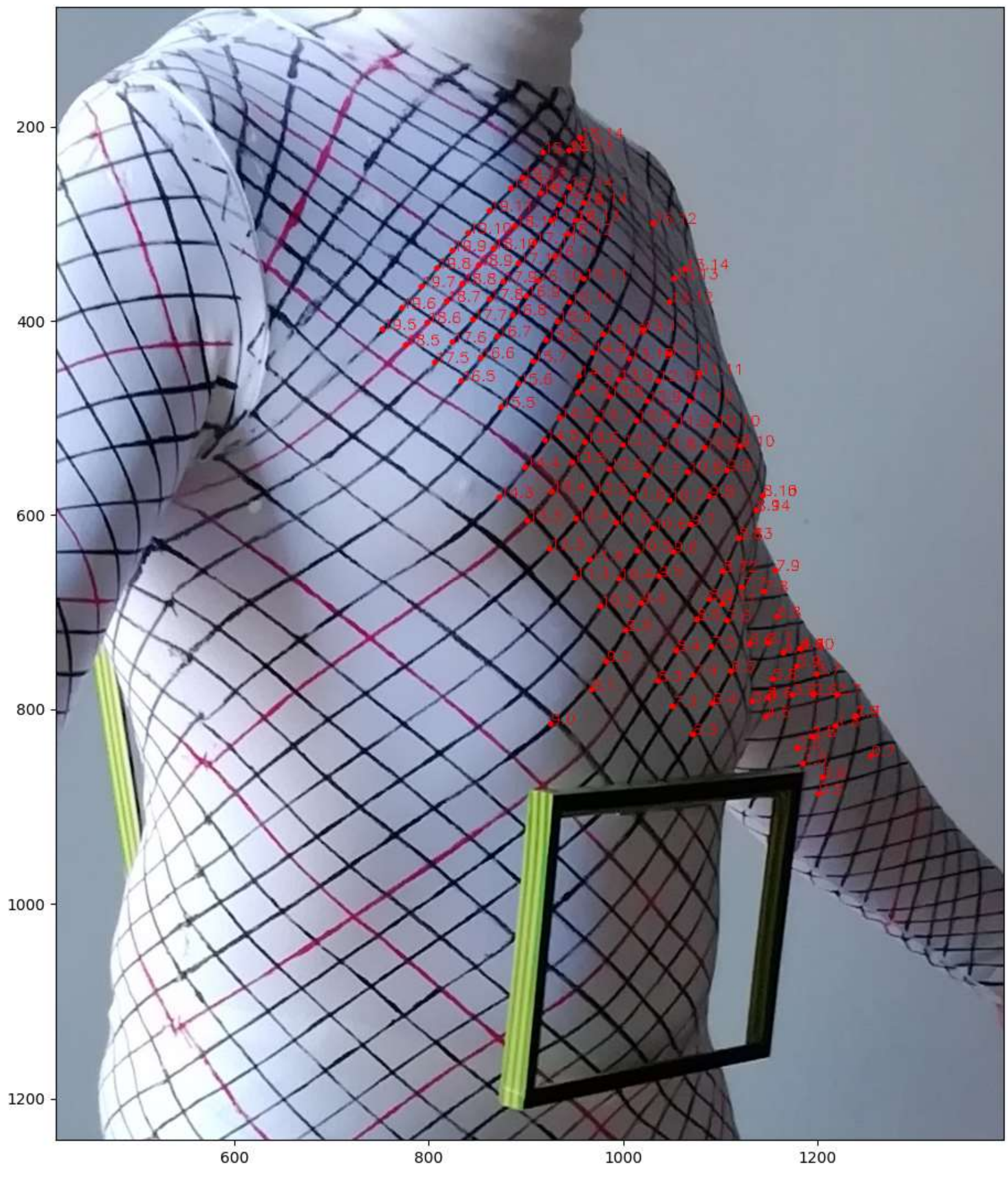
Crossings identification

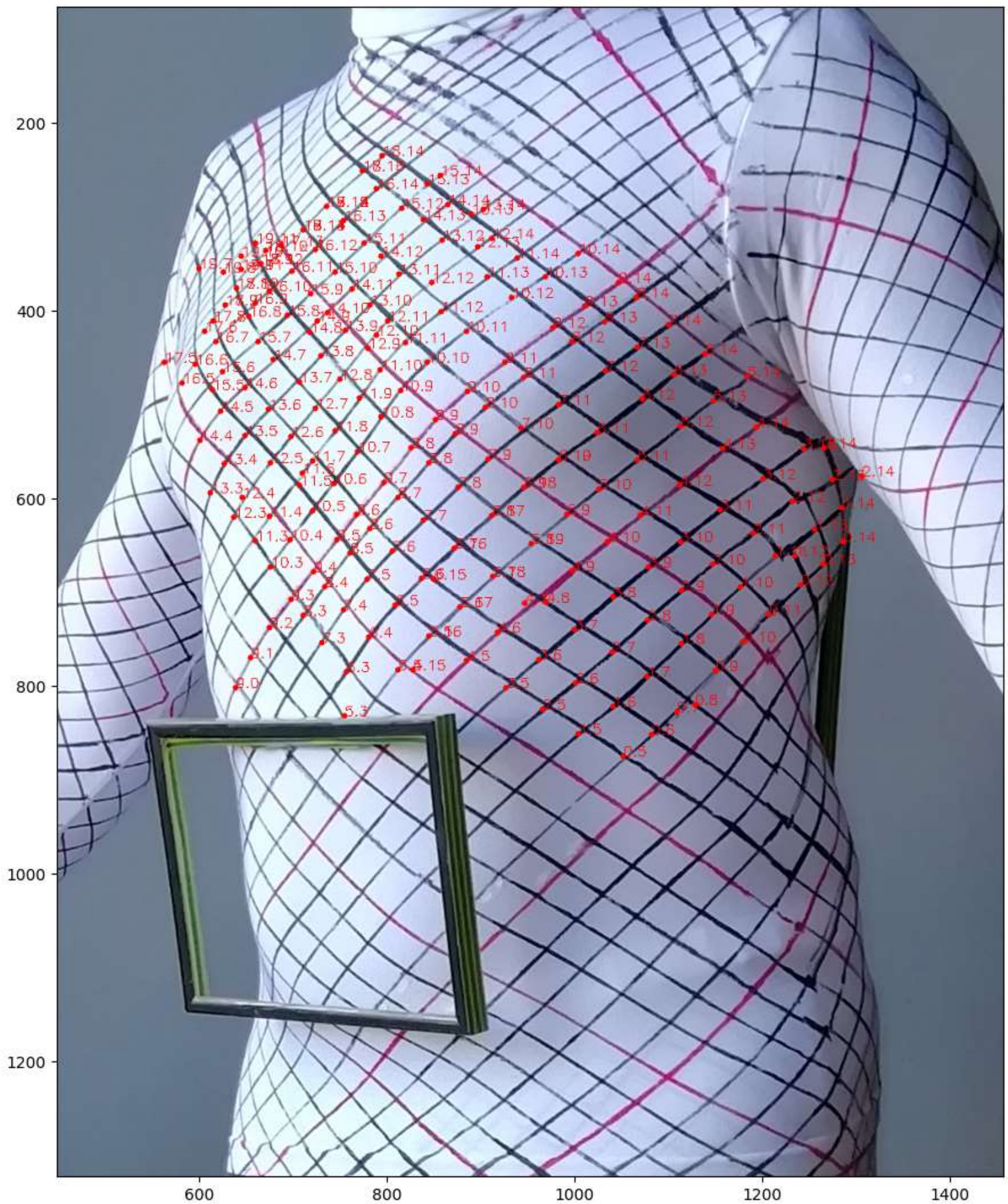
This aspect of the project turned out to be the most problematic one since even if only one line or few crossings are not found it immediately makes it impossible to correctly match corresponding pairs of points and therefore significantly impacts final reconstruction.

Procedure of identifying appropriate crossings is nearly the same as detection of them but it was necessary to additionally implement logic of indication which crossing is in which row and column of created crossing matrix. In order to do so, algorithm keeps going along curves and saves crossings into the crossing matrix. Once it detects that potential crossing can't belong to the for example 5th row since it's y coordinate is bigger than rest of points in current row, new row is inserted into the crossing matrix and remaining crossings are repositioned accordingly. Such an approach works relatively well for the surface of central body since it assumes that all the crossings are evenly located across the whole surface, however, it fails for less regular areas like legs and arms since in these areas there are more irregularities. Surely, this aspect leaves plenty of room for an improvement.

Final results can be observed in the pictures below.

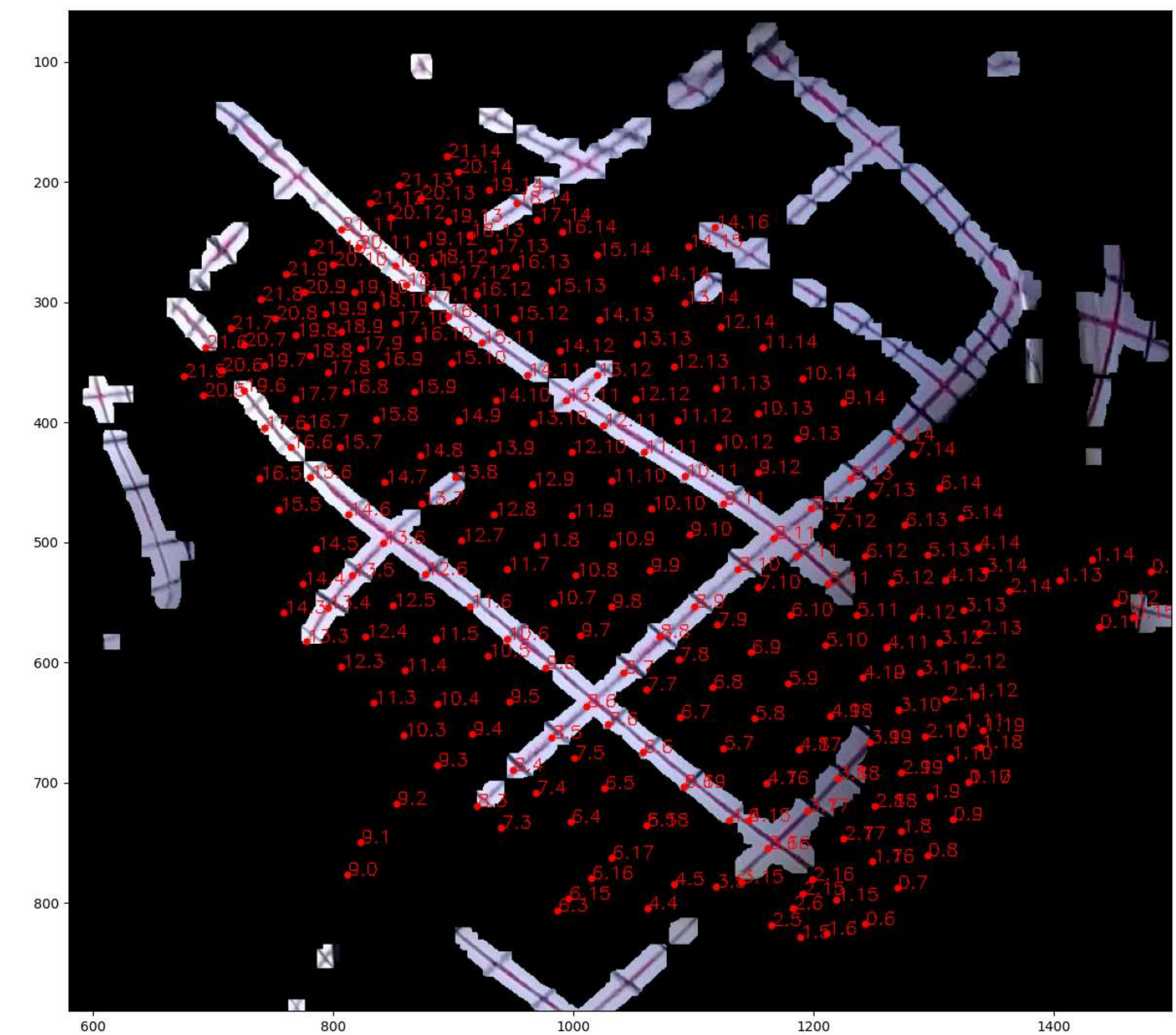


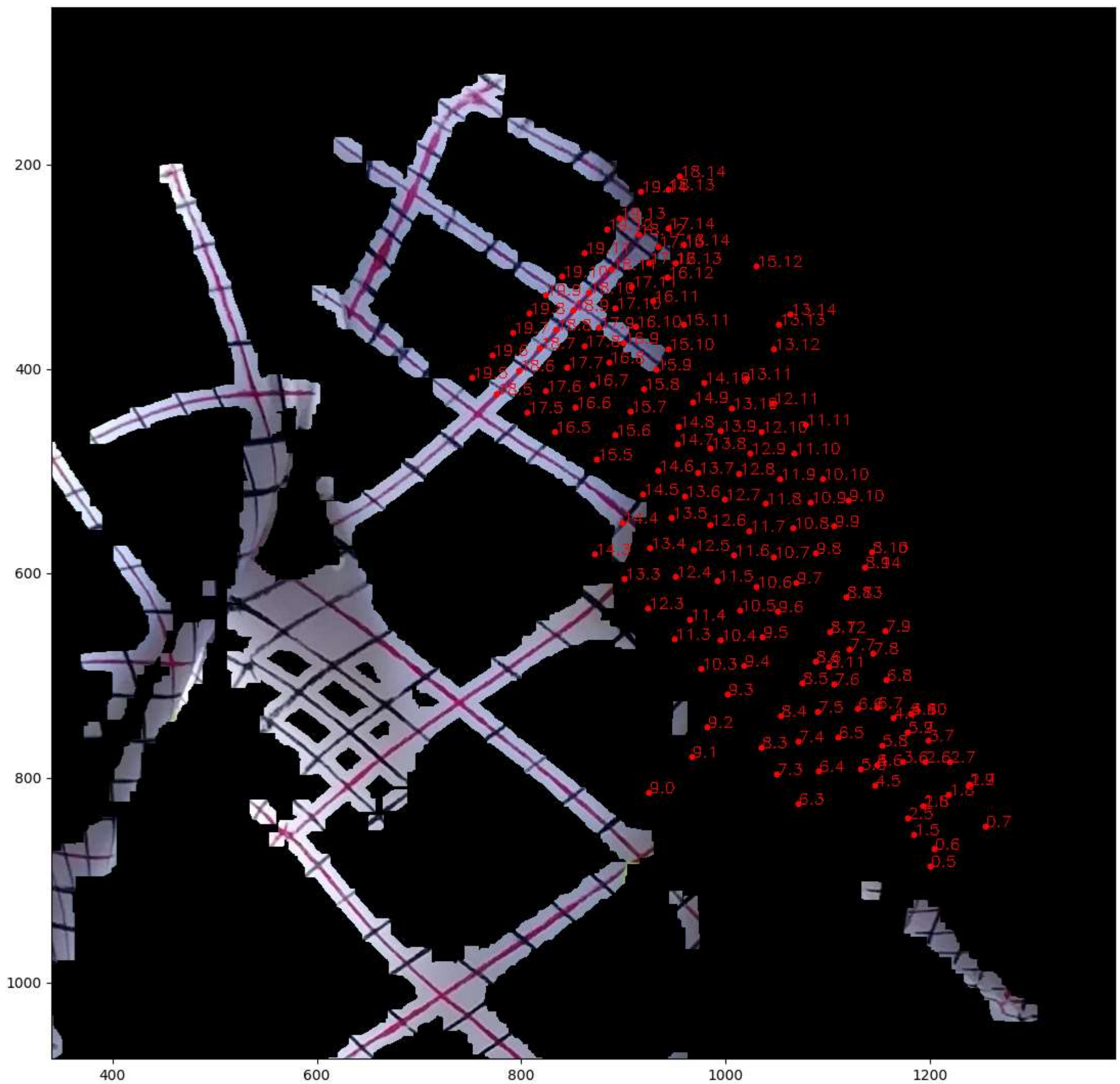

200
400
600
800
1000
1200
600
800
1000
1200

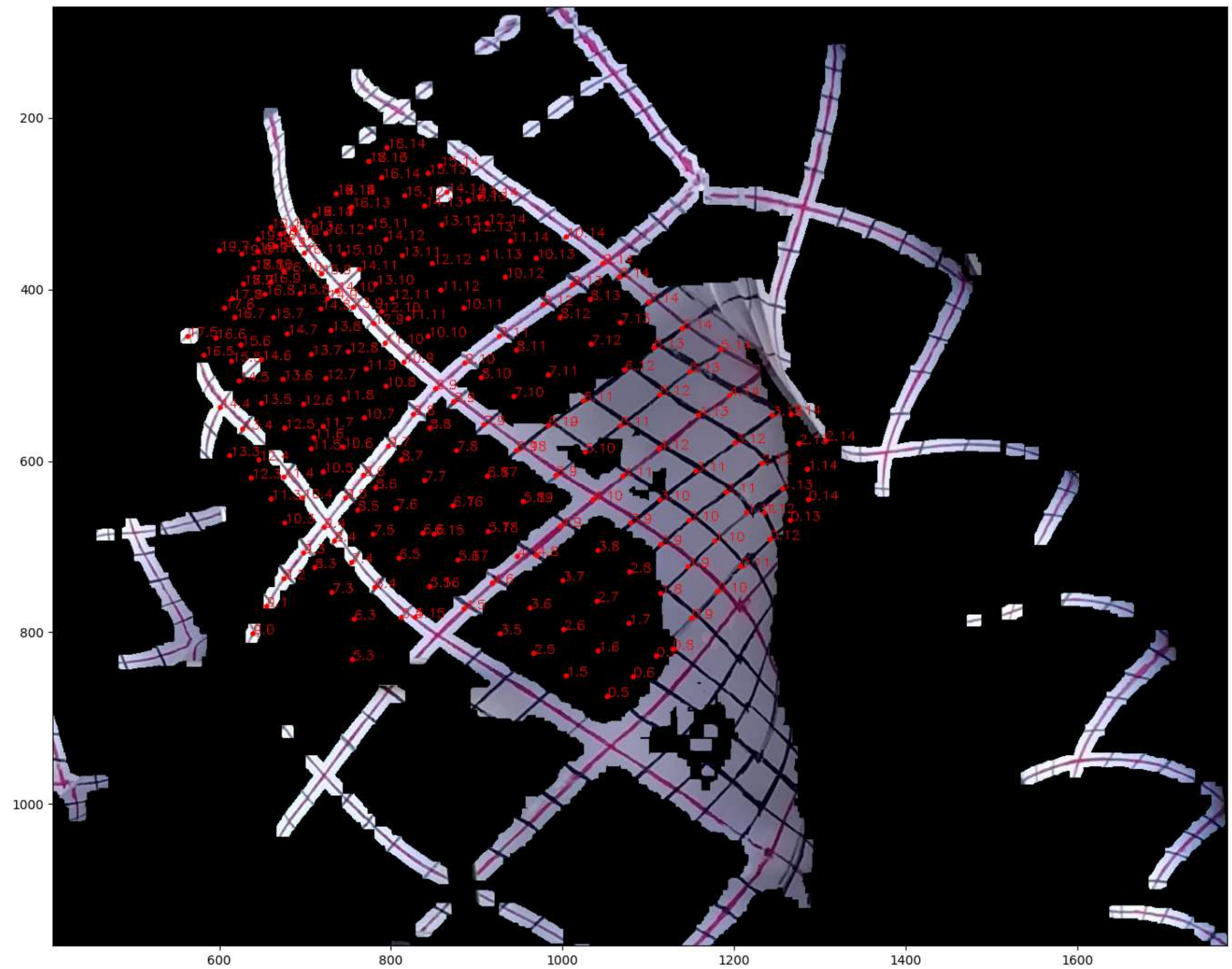

200
400
600
800
1000
1200
600
800
1000
1200
1400

Finding pink lines

In order to find pink lines pixels being in pink color were extracted. Then obtained mask was morphologically dilatated. It can be however noticed that detection of these pink lines isn't fully successful since different areas of human body are in different light and some pink parts of curves are indistinguishable. Once mask for pink lines was ready all the crossings in crossing matrix were iterated through and it was checked whether certain crossing lies on the pink line or not. Lines for which the biggest amount of crossings were lying on pink mask were decided to be pink. This way indications which lines are pink were obtained. This way crossing matrices could have been rearranged accordingly.




200
400
600
800
1000
400
600
800
1000
1200


200
400
600
800
1000
600
800
1000
1200
1400
1600

3D reconstruction

Having obtained intrinsic camera matrix, coordinates of corners of rectangular markers as well as corresponding pairs of crossings it was already possible to reconstruct given shape.

In order to estimate essential matrix both corners of rectangular markers as well as identified crossings were used. Once essential matrix was calculated it was possible to compute relative camera projection matrices and triangulate corresponding pairs of points in order to get coordinates of these points in the three dimensional space.

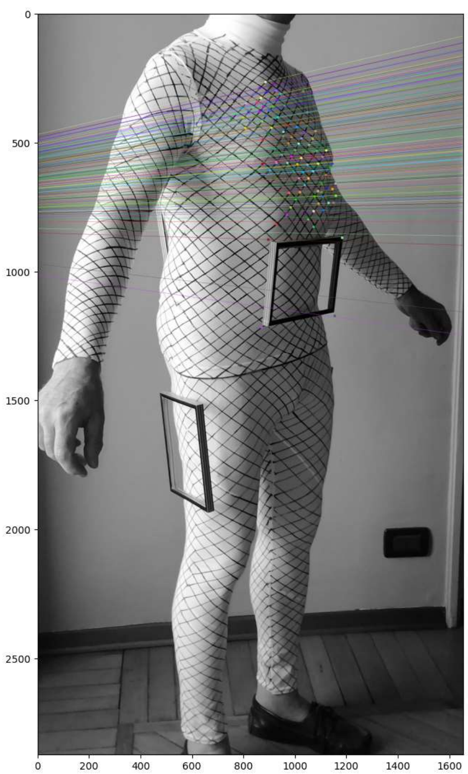
Relative Camera Projection Matrices

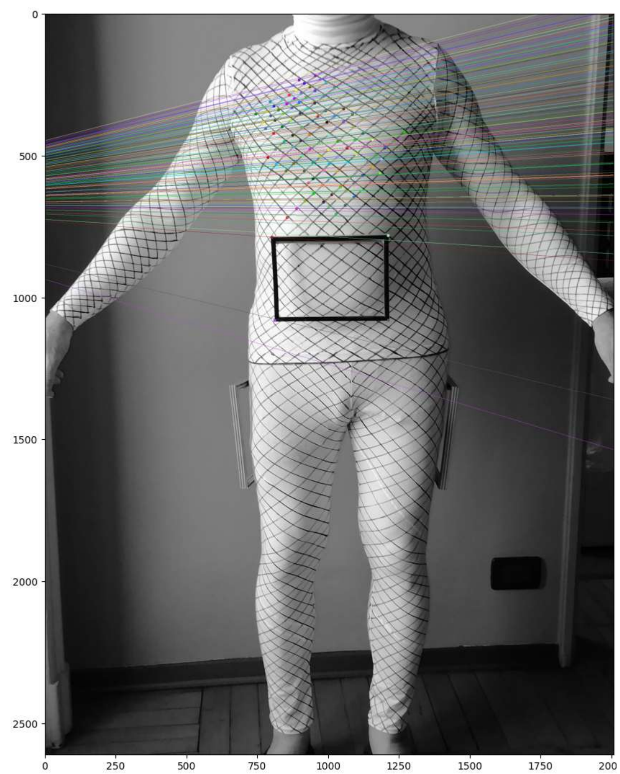
*P1 = K * [I|0]*

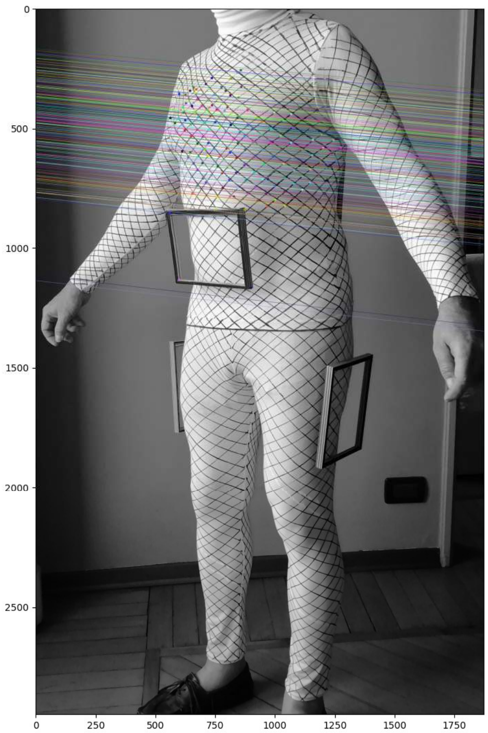
*P2 = K * [R|t]*

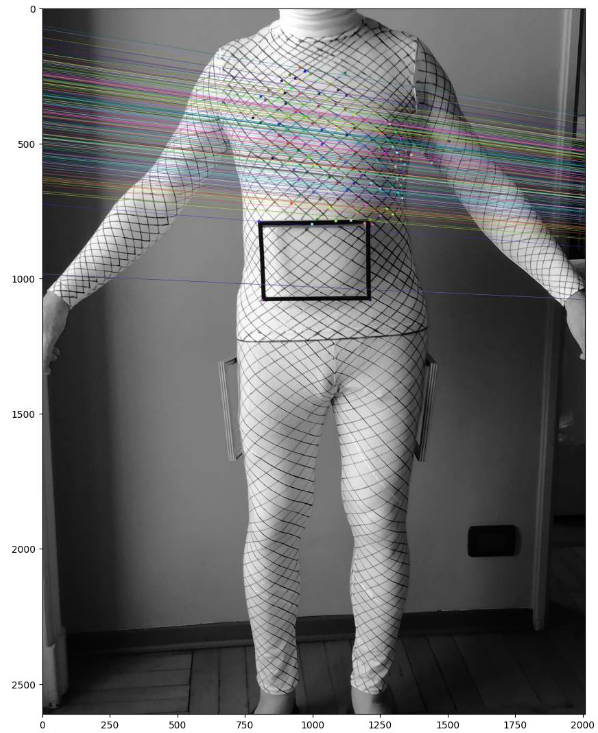
K – intrinsic matrix, R – rotation, t - translation

Images below visualizes epipolar lines going through corresponding pairs of points.

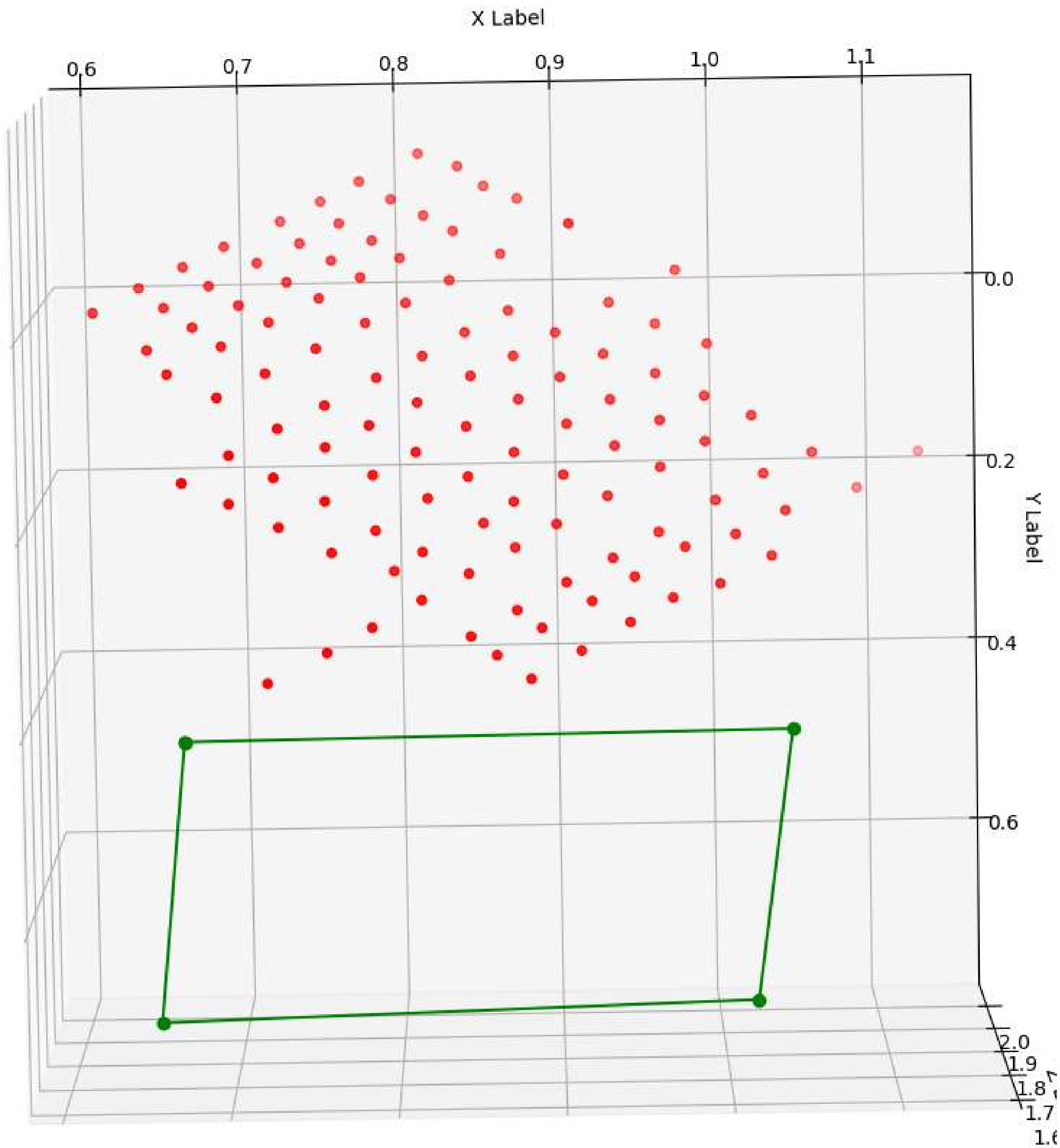




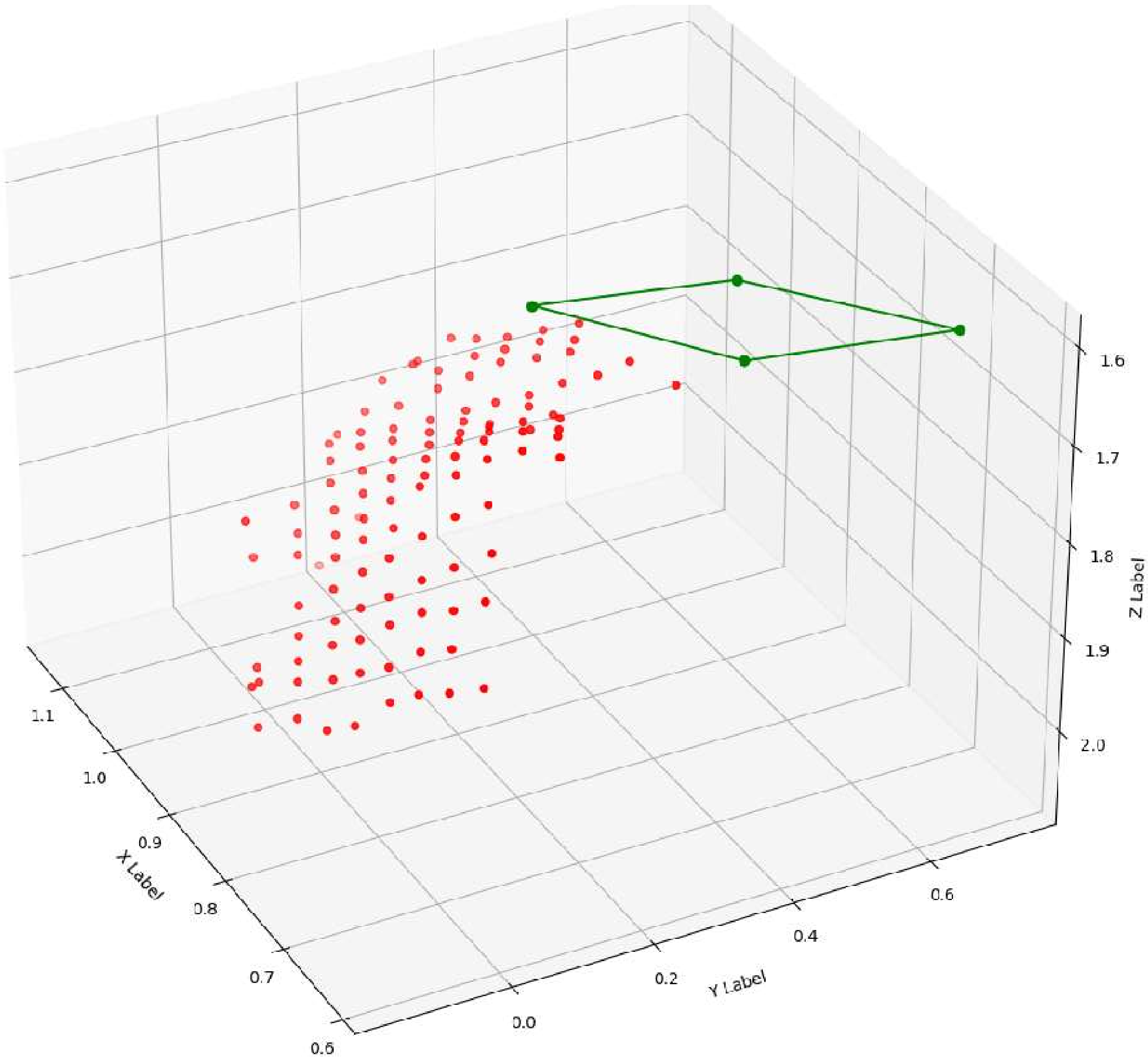




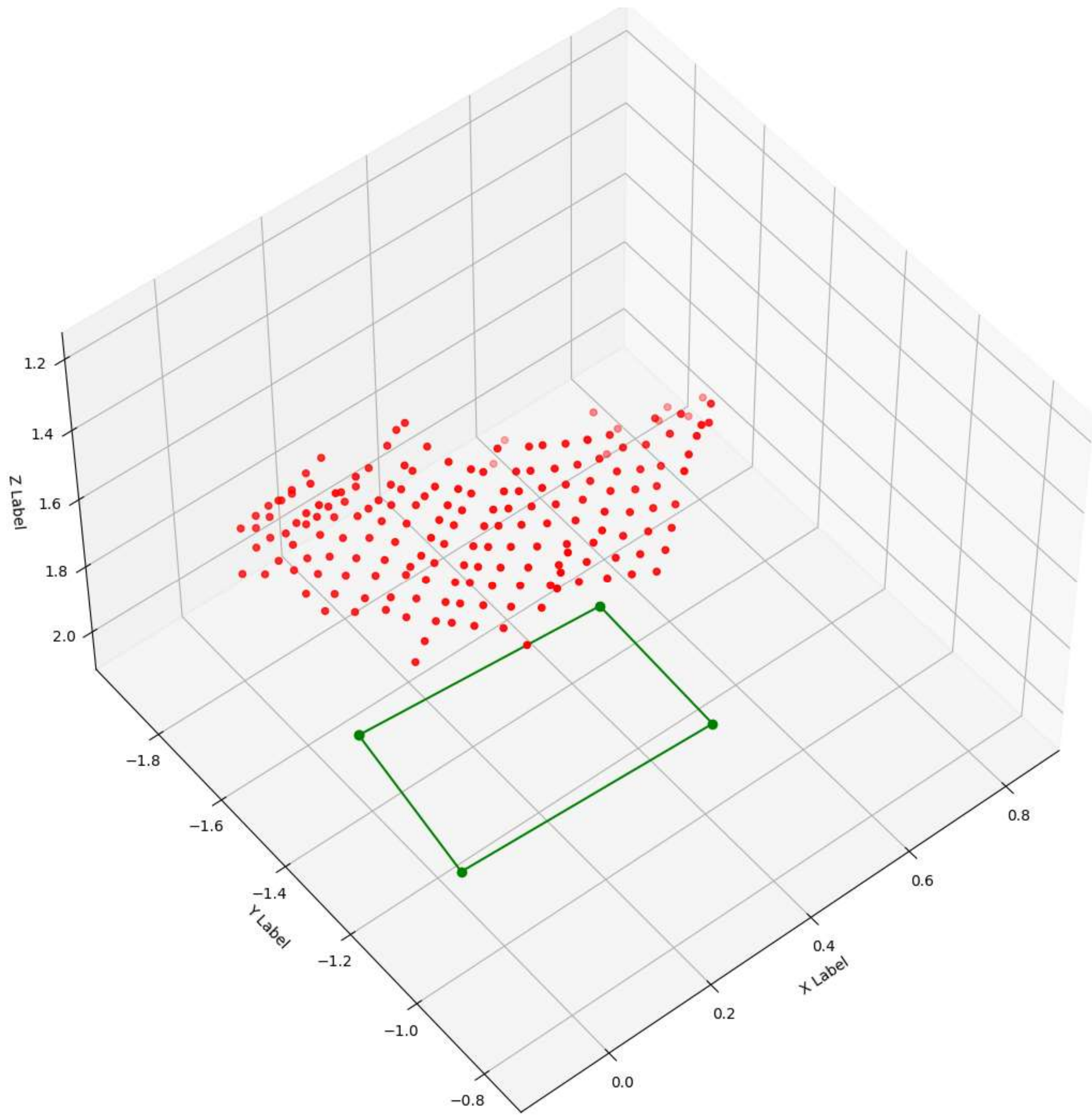
Final results of the reconstruction can be seen on the following pages. Frontal rectangular marker was highlighted in order to provide kind of a reference for all the visible points. It is noticeable that the front part of chest has been successfully reconstructed.



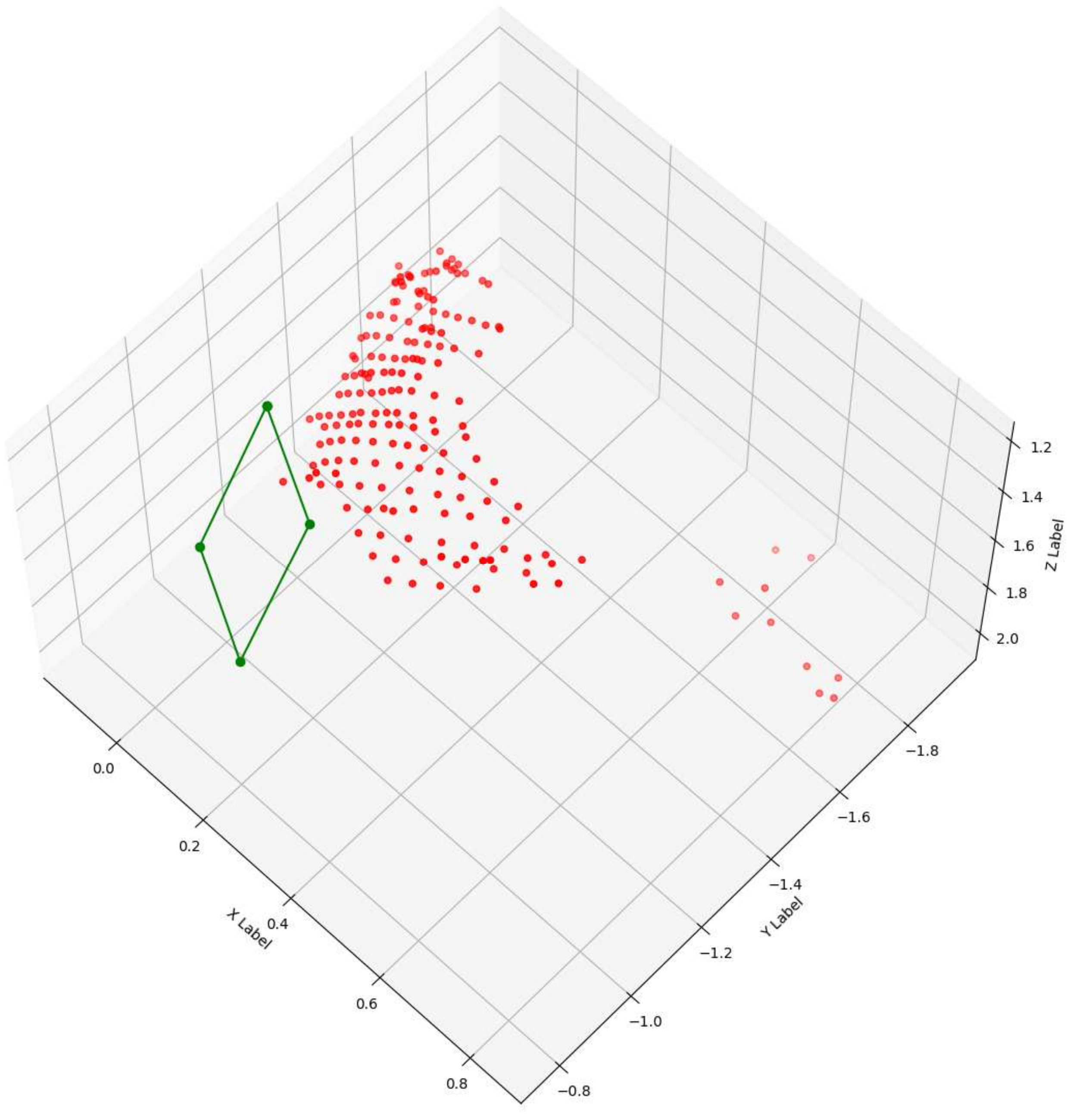
X Label
0.6
0.7
0.8
0.9
1.0
1.1
Y Label
0.0
0.2
0.4
0.6
2.0
1.9
1.8
1.7



1.1
1.0
0.9
X Label
0.8
0.7
0.6
0.0
0.2
Y Label
0.4
0.6
1.6
1.7
1.8
Z Label
1.9
2.0



1.2
1.4
1.6
1.8
2.0
Z Label
−1.8
−1.6
−1.4
−1.2
−1.0
−0.8
Y Label
0.0
0.2
0.4
0.6
0.8
X Label



0.0
0.2
0.4
0.6
0.8
X Label
−1.8
−1.6
−1.4
−1.2
−1.0
−0.8
Y Label
1.2
1.4
1.6
1.8
2.0
Z Label

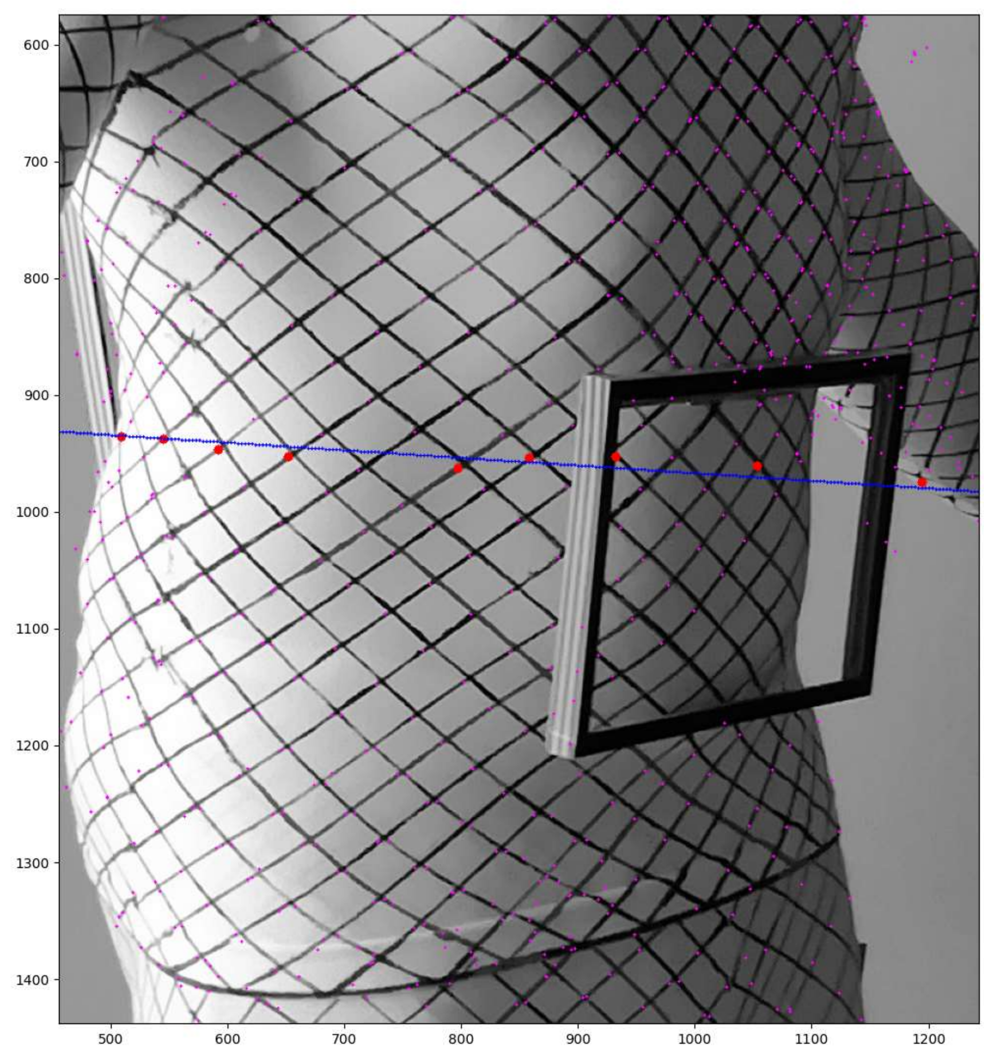
Post-epipolar matching considerations

Due to problems with identification of corresponding points on remaining areas of body like legs and arms using epipolar constraints seemed to be a promising solution to this problem. Unfortunately, initial analysis has proven that it might be impossible to successfully apply this idea. In fact, this approach turned out to successfully enable finding corresponding points lying on the line Fx’, where F is fundamental matrix and x’ is a point in the second image. However, multiple other crossings also are located on this line and could be potential corresponding points. There seems to be no solution to select the correct one among found points or at least there isn’t any I am aware of.

Epipolar constraint

xFx’ = 0

Image below represents this issue. There are multiple red crossings lying on the epipolar line. Among them there is a desired one but there is no way to correctly identify it.



Conclusions

In conclusion, effort put into the project provided me with valuable knowledge and better understanding of the topic of 3D reconstruction as well as made me more comfortable working on processing of computer images.

The main goal of the project has been achieved, nevertheless, there is plenty of apsects which could be improved. Among these which I consider to be most crucial are:

- Automatical verification whether detected marker is fully visible in the picture and which marker it is instead of strictly embedding knowledge which markers are visible in which images.
- Optimization of procedure of detection of crossings, since there are plenty of redundant iterations performed to increase number of detected crossings. In real time application it would be definitely too time demanding algorithm.
- Work on accurate identification of crossings located on less regular surfaces like legs and arms.

Furthermore, if pictures were taken from smaller angles between each other then they would share more common points and therefore shape reconstructed from two neighboring pictures would consist of more points and would represent bigger area of the shape. Moreover, robustness of the implemented solution might be an issue since it was only tested on available set of pictures instead of wider range of images.

The project leaves also plenty of room for extended approaches like full reconstruction of the shape from all the sides or even deep learning approaches could be used to obtain dense 3D representation instead of the one based only on identified crossings.