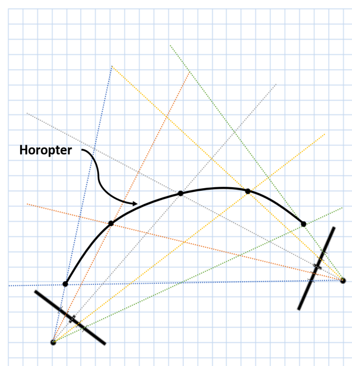
ICBV HW #5

Dor Litvak Gal Elgavish

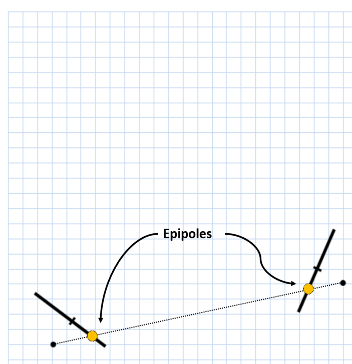
Question 1 - Stereopsis

Section A

1 - Drawing the Horopter

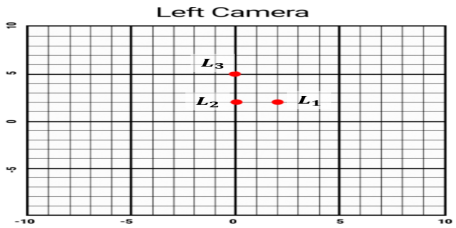


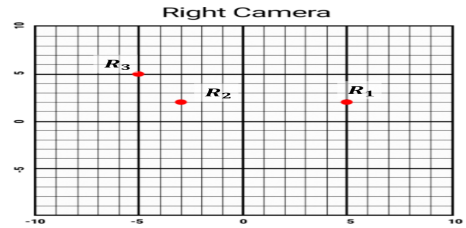
2 - Marking the Epipoles



Section B

We have marked the points so it will be easier to describe:





1 - How many different world configurations can correspond to the pair of images received in the system?

For a rectified calibrated stereo system, the y values in each image should be the same.
So L_3, R_3 are going together.
For the remaining points, since it was said in the course forum that the cameras can "see" behind them, there are 2 world configurations possible:

- $(L_3, R_3), (L_2, R_2), (L_1, R_1)$
- $(L_3, R_3), (L_2, R_1), (L_1, R_2)$

In each configuration, the ball corresponds to R_1 is behind the cameras (i.e. $z < 0$), since there is no point in the left camera with a larger x value.

2 - Find the distance of the 3 ping-pong balls from the cameras

We will use:

$$z = \frac{b \cdot f}{D} = \frac{50}{D}$$

For configuration $(L_3, R_3), (L_2, R_2), (L_1, R_1)$:

- (L_3, R_3): $z = \frac{50}{5} = 10$
- (L_2, R_2): $z = \frac{50}{3}$
- (L_1, R_1): $z = -\frac{50}{3}$

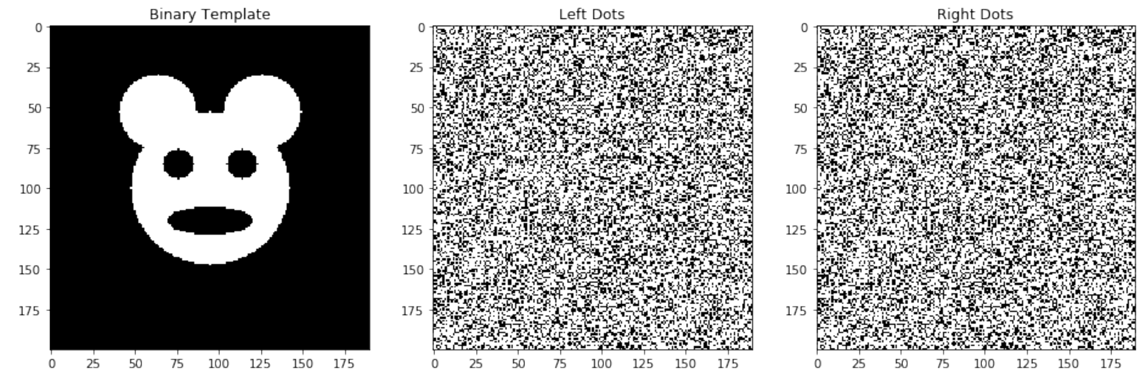
For configuration $(L_3, R_3), (L_2, R_1), (L_1, R_2)$:

- (L_3, R_3): $z = \frac{50}{5} = 10$
- (L_2, R_1): $z = -\frac{50}{5} = -10$
- (L_1, R_2): $z = \frac{50}{5} = 10$

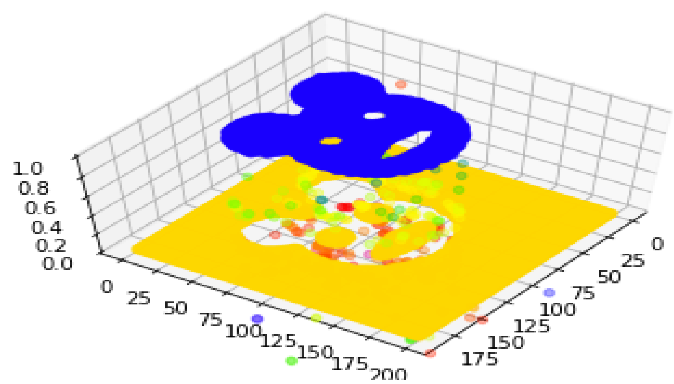
Section C - Random Dot Stereograms

The code is in the Jupyter Notebook
Here are some plots (w/o the animation):

binary image, left and right dots images



3D reconstructed shape



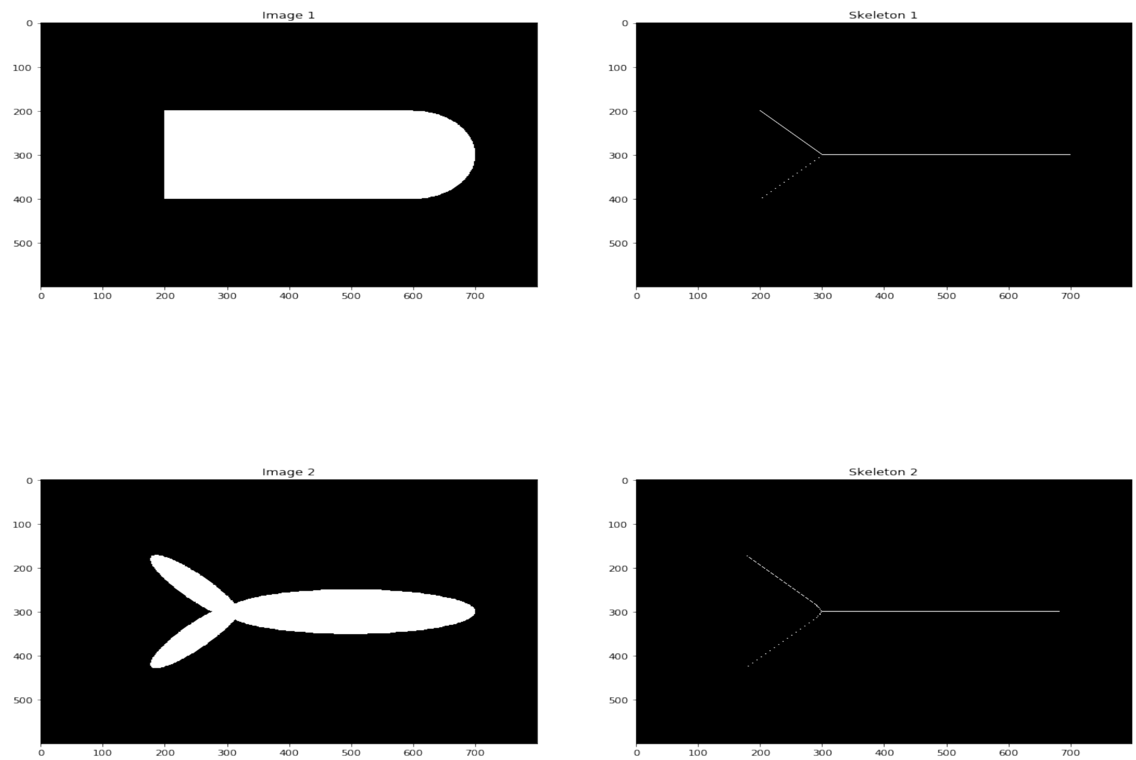
Question 2

Section A – Skeleton representation

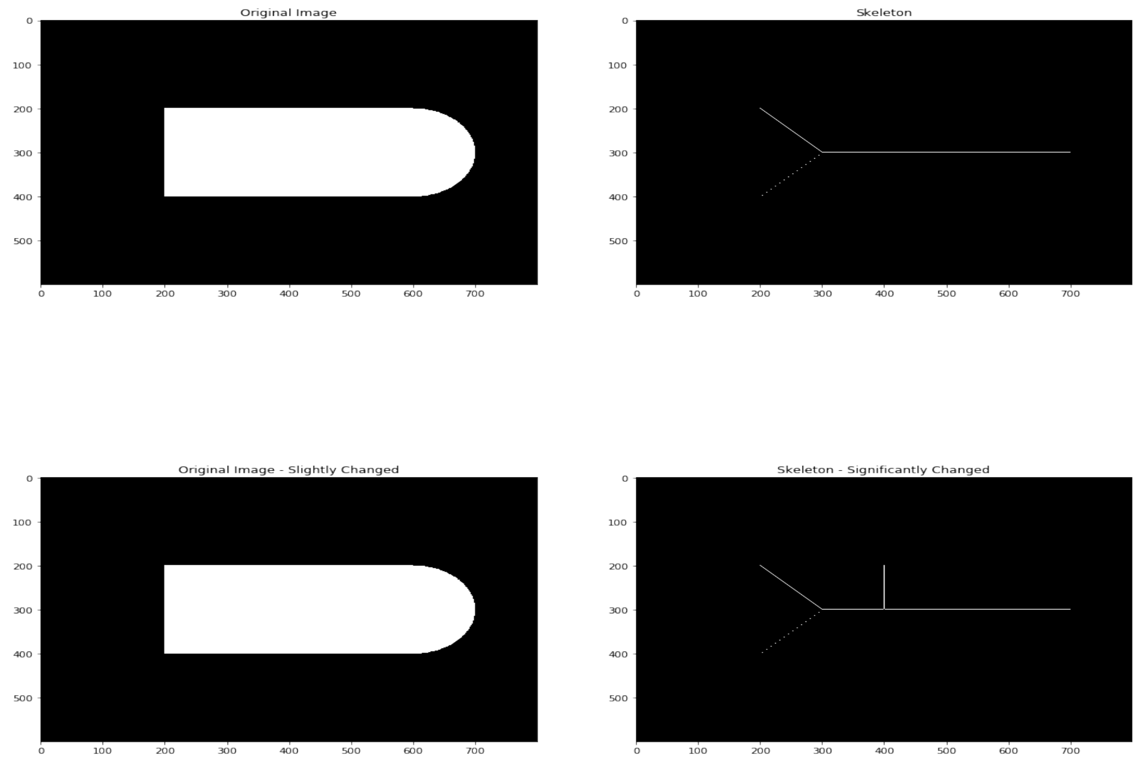
1) The skeleton does not represent a unique shape.
Straight lines can be the skeletons of ellipses as well as part of the skeletons of rectangles, and thus, we can build different shapes that will have the same skeleton as in the question.

The code for the next images in the Jupyter Notebook

2) Different shapes, same skeleton:

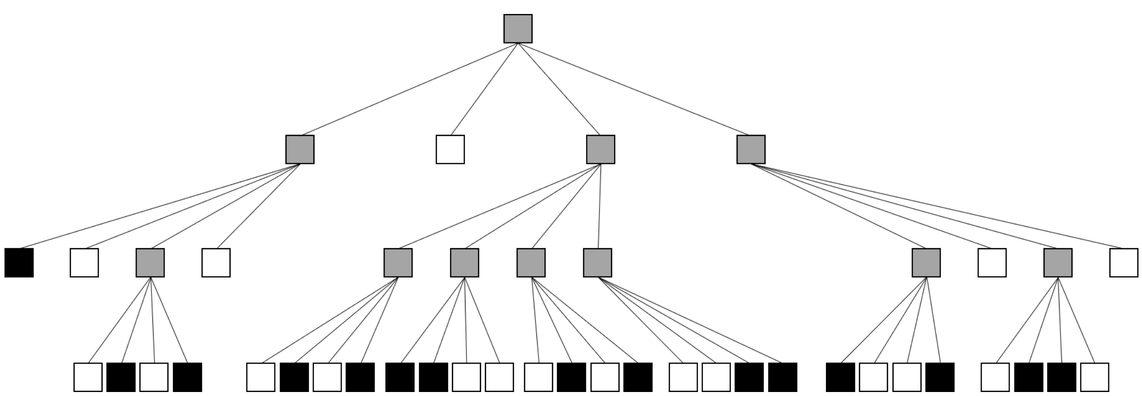


3) Slightly changed images, produce very different skeleton:

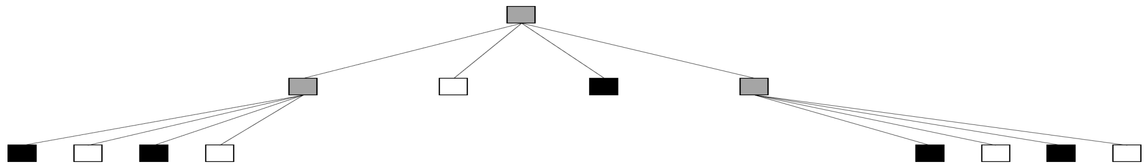


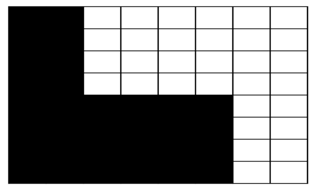
Section B – Hierarchical representation

The Quad Tree for the Letter 'b'



The Compressed Quad Tree for the Letter ’b’





Question 3 – PCA

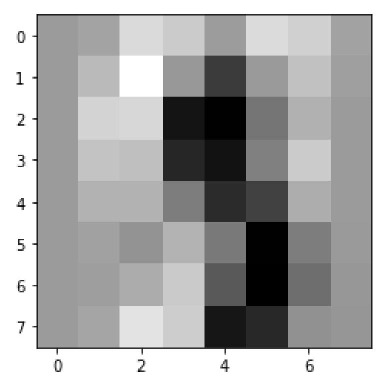
Section 1

The first 10 principal components present in the next figure contains the main features of the dataset. They are approximate catches 80% of the information of each image. Using linear combinations, we can ”build” and present an approximation of each one of the numbers in this dataset. We can also see that the information catches on these images is the main information of the dataset, for example, image PS10 catches the information of half a circle in the lower image, that can be found in number 3, 8, 6, 9, 5.

Another example is an attempt to build one of the numbers in the dataset using a linear combination of the 10 first principal components. $6 * PC4 - PC6 + 2 * PC10 \approx 9$.

This is not exactly the 9 digit, but given that those 10 PC gives us only 80% of the information this is close enough and can be said with confidence that this image contains digit 9.

Approximation of digit 9.



Section 2

For the first 10 principal components, we have approximately 0.8 of the cumulative variance which means those components contain 80% of the information on the digits. This information is correlated with the insights we got from visually inspecting them. Those are the main features, but they are not containing the full information on the digits (the other 20This is a good compression of the data but may not catch the full information and may cause mistakes between similar digits.

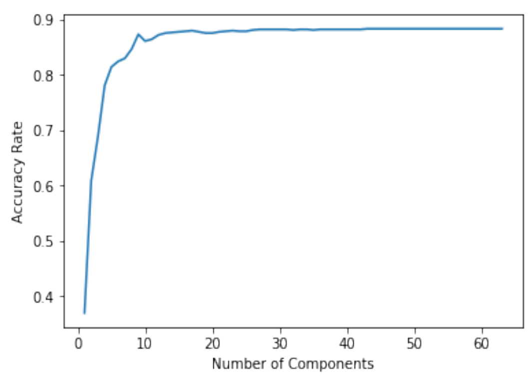
Section 3

The best number of components depends on the graph on figure 3 slope and on our task. On the one hand, we don't want too many components that contain too much information, because some of them would only contain local information that can decrease our classification. On the other hand, a small number of components can cause a mix between clusters. The best number of components is the smallest number containing the majority of information on the images. This number depends on the slope of the cumulative variance vs the number of components graph, And it is approximately 15 components contains approx 90% of the data's features.

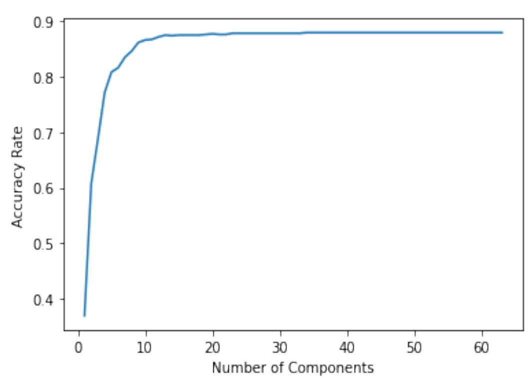
Section 4

The code is provided in the Jupyter notebook.

The graph of accuracy rate vs number of components, using norm 2 distance measure:



The graph of accuracy rate vs number of components, using minkowski distance measure with p=3:



Section 5

a)

Using norm 2 distance measure:
Max accuracy: 0.8832035595105673
Using 42 components

Using minkowski p=3 distance measure:
Max accuracy: 0.8798665183537263
Using 33 components

b)

What is the relation between figure 3 and the results you got in this section? There is a consistency between figure 3 and the results we got. In both cases the first 10 features gave us good approximation about that data 86% accuracy which is very close to the max accuracy 88%, and 80% of the data variance. In both figures the slope of the graph flattened at about 10-15 components . Most of the features were catched and there was nothing to learn more from the data.

c)

The number of components I would choose to use is 13 in the norm 2 distance measure which give us 87.5% accuracy (comparing to 88.3% for 42 components.) And using minkowski p=3 distance measure also 13 components 87.5% accuracy out of 87.9% accuracy. (If I need to maximize the accuracy and don't have number of components limitation I could also can use 26 components for norm2 which gives me 88.2% out of 88.3%.)