CHALMERS

RRY025 Project
Image Processing

Project 9: Portrait mode for image enhancement

Project Group 4

Filip Ronnblad, Julie Noblet-Reverbel
Cheng Chen, Yu-hsi Li

March 26, 2024

Contents

Abstract

Photography is more accesseble and popular than ever with the continous evolution of phone cameras and the image processing software within them. One feature of many cameras today is to create automatic portraits where the face of a person in an image is enhanced and the background is blurred. There is many approaches to creating such a feature and many rely on complicated algorithms. In this report simpler methods is used and evaluated for the purpose of learning basics of image processing. Different methods of denosing, sharpening, face detection and more is implemented and compared to each other in order to get a feeling for what works in creating portrait images and what does not work. Finally some discussion and conclusions is drawn from the analysis and some recommendations for further investigation is given to further be able to understand basic principles of image processing.

1 Introduction

In the age of digital photography, the quest to capture pleasing portraits has become a popular trend among photographers. Portrait mode represents a significant creative revolution in modern cameras, aiding both professionals and amateurs in obtaining photos with distinct focus. The key feature of portrait mode is to isolate and emphasise the subject, typically a face, while simultaneously blurring the background to create the so-called "bokeh" effect.

However, executing this technique can be rather complex, as it involves identifying the main subject within an image, distinguishing it from the background, and applying image enhancement techniques to refine the result.

1.1 Problem definition

Creating a "portrait mode" for pictures means blurring the background of an image while keeping the details of the face/person in it. Different approaches to the problem are possible to use, like:

- Histogram Equalisation
- Sharpening
- Thresholding

The "perfect" portrait mode would be a combination of the different methods divided up into stages such as preprocessing, edge detection and finalisation. The combination of methods from different processing stages is the main focus of this project.

1.2 Limitations

This project can become very hard to handle depending on several factors like the quality of the image or its background. That is why the main limitation of this project is to only use a handful of "easy-looking" images. These images have a quite simple background that seems easy to identify. Mainly two pictures will be analyzed, one with a very smooth background but a light gradient and one with a relatively easy textured background (see fig 3). Ideally, the same implementation should work for both images but realistically some parameters might be tweaked such as thresholds and level of smoothing or sharpening.

1.3 Theoretical Background

Here on follows a theoretical framework for some tools used in the analysis. Certain basic principles of linear systems, such as convolution, Fourier transform and similar, are considered prerequisites and will not be explained but may be used within the methodology or results.

1.3.1 Gaussian filter

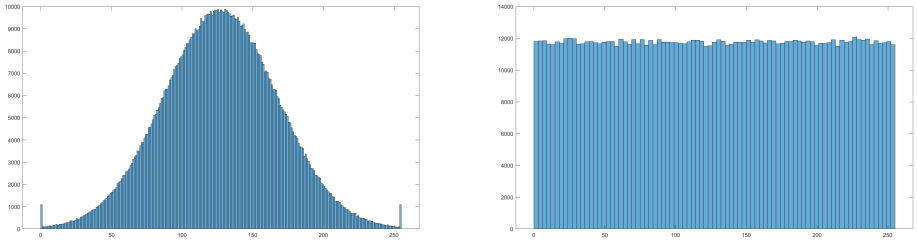
The Gaussian filter is an easy tool mainly used for image smoothing or blurring. The kernel is calculated based on the 2-dimensional function

$$z(x, y) = e^{-\frac{x^2+y^2}{w\sigma^2}}$$

where x and y are coordinates in the filter kernel and σ is a parameter controlling the standard deviation of the filter [1]. Convolving this kernel with an image is a very common way of blurring an image which trivially will be important for blurring the background of a person or for preprocessing to remove noise in an image. Examples of this kernel can be seen further in the methodology section.

1.3.2 Histogram equalisation

Histogram equalisation aims to spread out the intensity of of the image. This, in theory, gives a higher contrasted image where the details can be more visible [6]. A simple example can be seen in fig. 1 where the image on the left can be viewed as the original image with a Gaussian distribution and the image on the right as the equalised image.



(a) Gaussian distribution (b) A flat distribution

Figure 1: An example of histogram equalisation where the pixels distribution on the left has been remapped to the flat distribution on the right.

It is also possible to equalise just a part of an histogram to enhance certain features while leaving other features untouched. This divided equalisation could be useful for example to enhance a object but not the background, or the reverse.

1.3.3 Convolutional edge detection

Convolutional edge detection could be executed in many different ways. Most of the established methods are based on the variations in the image, these include first-order derivative kernels like the sobel, roberts and prewitt filters. There are also second-order derivative filters

like the Laplacian kernel which will be the focus onwards. Its operator is

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

An added benefit of the Laplacian filter is the possibility to change the filter size and more importantly, the σ value which works the same way as with any Gaussian filter. Derivative based kernels are generally very noise sensitive and do not work well with noise or irregularities in the background. The Laplacian filter control over the filter size and σ provides the ability to make adjustments to remove some of the noise and unwanted edges in the resulting output [3].

1.3.4 Wavelet Transform

The wavelet transform is a method for decomposing an image into fundamental details. It creates four sub-images, one is a half-smaller approximation of the image, one represents the vertical details of the original image, another represents its horizontal details and the last one represents its diagonal details [4]. The approximation is calculated by averaging the original image. A toy example of the wavelet transform of an image of a box can be seen in fig 2.

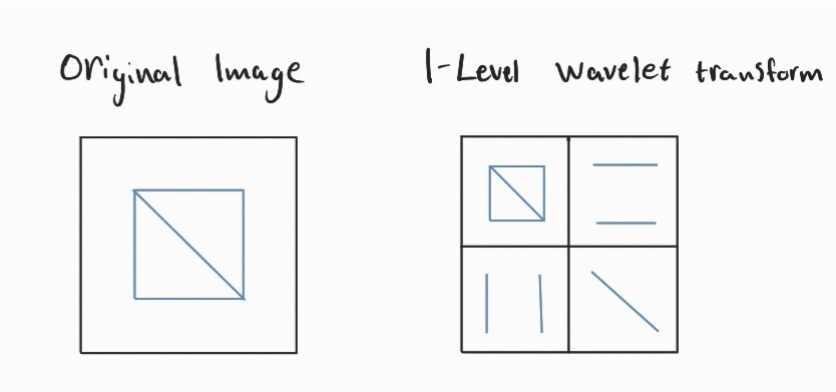


Figure 2: Simple 1-level wavelet transform using the Haar wavelet.

For more complex images a new wavelet transform can be performed on the approximation image to reveal further details in the image. The further down in wavelet layers you go the more "large scale" is the details of the transform, therefore small features such as image noise are usually found in the first couple of layers and larger image noise or shapes can be found in later layers.

The wavelet transform is reversible, meaning that the original image is able to be recreated from the approximated images and the details images. If the coefficients of the transformed image are modified before recalculating the original image, different features of the original image can be enhanced or reduced in the reproduced image. These applications could be useful in

preprocessing an image and performing edge detection.

1.4 Wavelet denoising

Wavelet denoising is a technique used to remove unwanted noise from signals or images. It works by breaking down the image into different smaller images and details and then cleaning up the noisy parts by removing the corresponding details and layers. Empirical Mode Decomposition (EMD) is another way to clean up noisy data. In this specific wavelet denoising method, we use the Empirical Bayes denoising with a process called soft thresholding. Soft thresholding represents a way to gently reduce the noise of an image. Empirical Bayes denoising relies on a rule that considers the characteristics of the noise in the image. It uses this information to make better decisions about which parts of the image are noise and which parts are the actual image that we are looking for. The characteristics of the noise are determined by comparing the actual noisy image with other sharp images that should be very similar. The mixture model refers to the image which is made up of a mixture of noise and information. Empirical Bayes is a way to figure out how much of each there is in the image and clean up the noise accordingly [5].

2 Methodology

Two different images were used in order to develop and test different approaches to creating a portrait image. One picture was chosen with a very simple background and another image had a little bit more noisy background where the background wall was textured.

(a) An easy background image without any texture.

(b) Picture with a textured background.

Figure 3: The two images used in development/tests.

On these two images, different preprocessing, edge detection and finalisation were done in order to achieve a portrait image. In the coming sections come some description on different methods used to get our results.

2.1 Preprocessing

Before an edge detection could be performed, different forms of preprocessing were done to the image in order to either reduce some noise, sharpen up the edges or similar. Below follows the three main strategies that were attempted and tested for preprocessing.

2.1.1 Histogram Equalisation

One preprocessing technique used on the original images was the histogram equalisation, in order to increase contrast in the images and to smooth together the darker and lighter parts of the face to enhance. The equalisation was done in two different parts. The part including the face was equalised and spread out on the whole histogram while the other part containing the background remained still. The more difficult of the two pictures had an histogram seen in fig. 4 where lower pixel values would correspond to the person in fig. 3b and the higher values would be the background. The goal of our program was therefore to equalise the lower part of the histogram while keeping the higher part as it is. If the parts of the histogram corresponding to the person or the background could easily be identified, the contrasts of the person were increased and the background should not be enhanced.

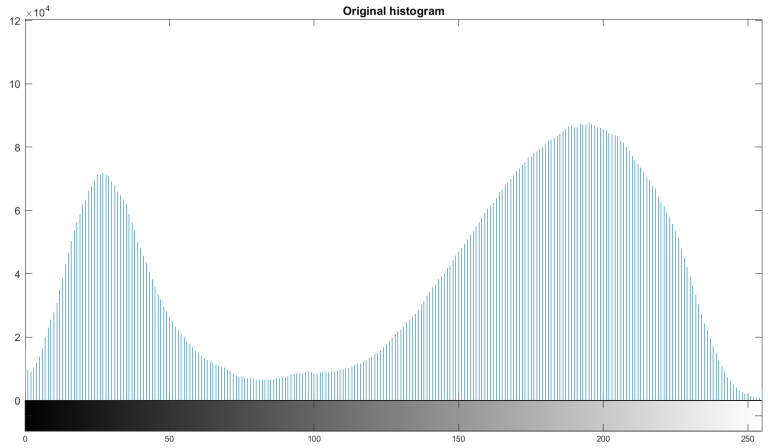


Figure 4: The original histogram of the difficult image

2.1.2 Sharpening

Gaussian blurring was used with the filter in fig. 5 to preprocess an image using sharpening. To create a sharpening filter the smoothed image was removed from the original one. This edge mask that was then added back to the original image with a certain enhancement factor A, set arbitrarily for a ”sharper” image.

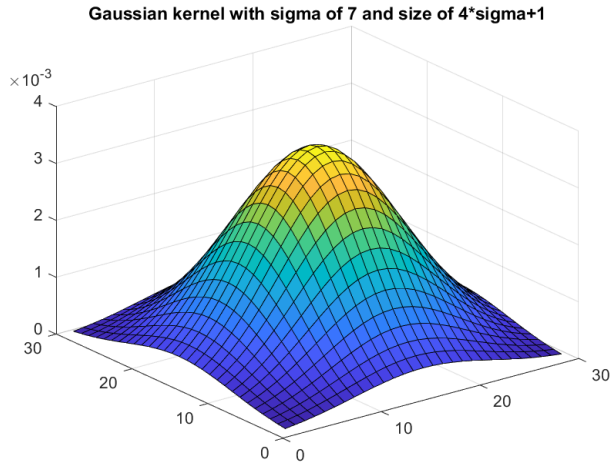


Figure 5: A Gaussian kernel with $\sigma = 7$ and a filter size of $2 \cdot ceil(2\sigma) + 1$ which is the standard for the matlab function imgaussfilt().

2.1.3 Lowpass Filter

Another approach to preprocessing was to use the lowpass filter implemented by Gaussian filtering (imgaussfilt() in matlab) and intensity adjustment. This process smoothed the image and removed some of the high frequency noise.

2.2 Edge Detection

Three different methods were tested for edge detection, a second order derivative Laplacian filter, then wavelet decomposition and finally a binary threshold. The wavelet decomposition picked out relevant details and removed unwanted features. The obtained binary image contained either 1 or 0 depending on the threshold value used on the grayscale values.

2.2.1 Laplacian Filter

To have the better resultas on our images, a Laplacian filter with a filter size of 60 pixels and a σ of 7 could be used. These parameters have been obtained by empirical tests and could vary from image to image. An illustration of the kernel is found in fig. 6.

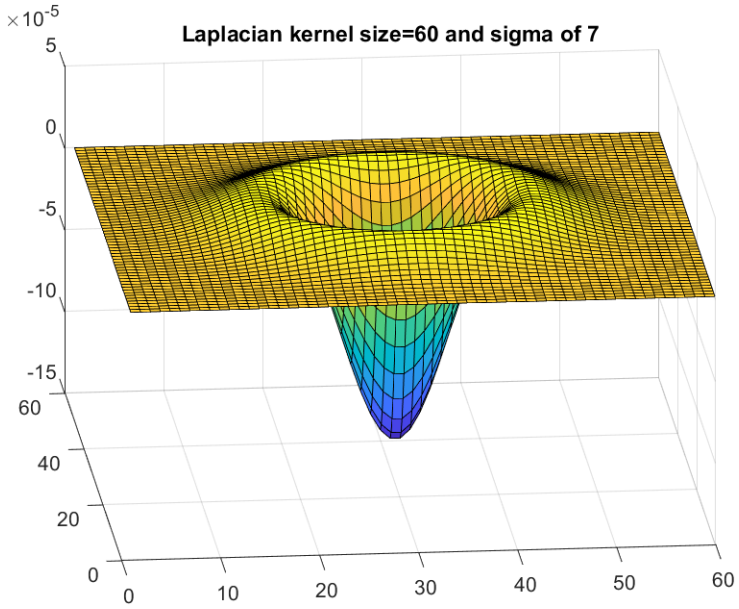


Figure 6: A Laplacian kernel with size 60 and sigma of 7.

The output of the filter was a mask of the original image with nonzero coefficients highlighting the edges in the images. All nonzero coefficients could be set to 1 and thus creating a binary image that could be further manipulated in the finalisation steps.

2.2.2 Wavelet Decomposition

To apply wavelet decomposition to our image, one wavelet must first be chosen. Haar wavelet was the simplest and most fundamental wavelet used in wavelet analysis. It was especially useful for detecting abrupt changes in data, which is why it is frequently employed in applications like edge detection in image processing or step detection in time-series data. Wavelet decomposition with the Haar wavelet was then applied to the image and its approximation to reach the fourth level of decomposition of the image. This step was executed on the three coloured layers of the images. This decomposition produced four sets of wavelet coefficients for each colour layer. Wavelet-based denoising was then applied on these coefficients, using techniques such as Bayesian denoising and soft thresholding. Finally, the image was reconstructed from these wavelet coefficients, and the reconstructed channels were merged into a single RGB image.

2.2.3 Thresholding

To apply thresholding on the images, the magnitude spectrum of the Fourier transform must be transformed into a binary matrix. That means that all the values of the coefficients over the chosen threshold value were put to 1 and all others are put to 0. An illustration of this is found in fig. 7 where the threshold is set to 0.7 of the maximum value.

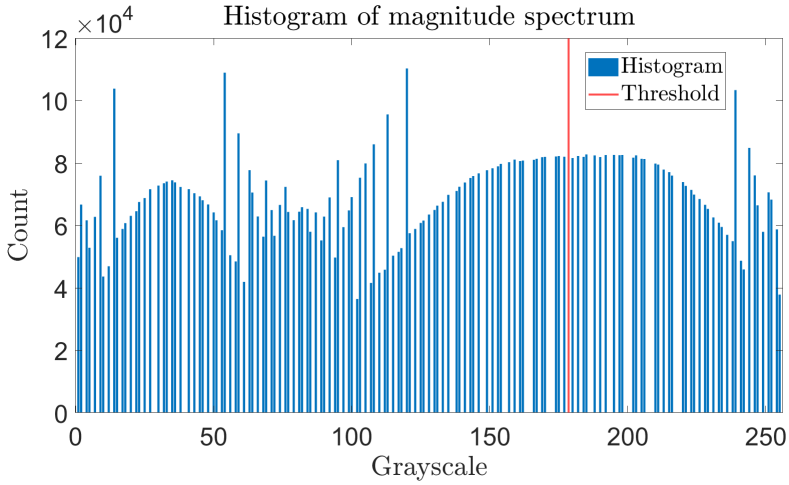


Figure 7: Thresholding of the magnitude spectrum

With a fixed threshold the output would be an image similar to what can be seen in fig. 8 where you can easily identify a person. The value of the threshold was tested and set empirically to minimise the pixels of the background taken but also to get as much pixels as possible of the person inside the mask. In fig. 8 three cases could be found. From left to right, the cases represented when the threshold is too low, too high or just right.

(a) Low threshold (0.7). (b) High threshold (0.9). (c) Best threshold (0.85).

Figure 8: Comparison of the different thresholds on the easy image.

2.3 Black-White Manipulation

In our group, many different approaches have been tested on the obtained binary images. The general process was to; connect the binary points surrounding the face to create a full mask; remove remnant pixels of the background/noise; fill in the area enclosed by the face.

To close the image borders, the functions bwmorph('thicken'), imclose() among others were helpful in matlab. Removing noise and background artifacts was possible with bwareaopen()

and filling in the mask was done using imfill(). Examples of outputs of these functions can be seen in the results section of the report.

2.3.1 Image Finalisation

After creating the image of the mask, the finalisation was trivial for all previous methods. The image of the face only was stored in a temporary variable by multiplying the original image with the binary mask. The original image was then blurred with a Gaussian filter with a large arbitrary sigma chosen to make the background "looking good". Finally the temporary image of the face was added back into the blurred image to make the face looks sharp again.

3 Results

Below follows the results for the different methods tested. Similar to the method structure, this part is divided between preprocessing, edge detection and finalisation.

3.1 Preprocessing

The results for equalising, sharpening and lowpass filtering can be seen in fig.9. The images obtained with histogram equalisation or sharpening have more details compared to the original image, the edges seem indeed to be sharper. On the other side, the image obtained with a lowpass filter looks smoother, reducing the noise in the background. The difference between enhancing and lowpass filtering will be explored more with the edge detection results.

(a) Equalised image

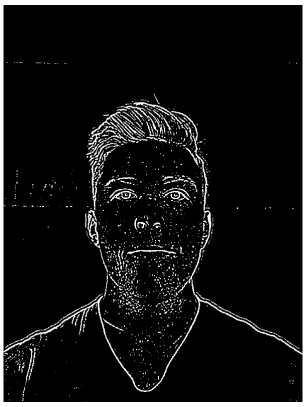
(b) Sharpened image

(c) Lowpassed image

Figure 9: Different ways in which preprocessing was applied to an image.

3.2 Edge Detection

The results obtained after using the Laplacian filter for edge detection on our image can be seen in fig. 10. For these specific images, our original image had been first preprocessed by sharpening. On the easy image we get a sharp and clear border around the person but on the more difficult image, some noise from the background remains. On both of these images however the border around the person is quite accurate. Denoising the difficult image using any other preprocessing method before applying edge detection to it reduces the noise in the final image but it also blur the edges around the person. The edges around the person become then less accurate and its mask looses a lot of accuracy.

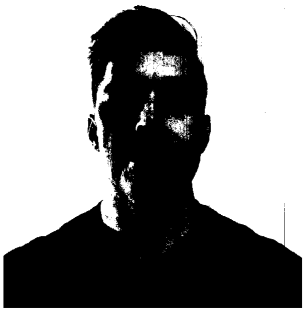


(a) Edge detection easy image.

(b) Edge detection difficult image

Figure 10: Edge detection using the Laplacian filter with both images.

The results from the binary threshold edge detection can be seen in fig. 11. This was one of the methods that worked the best on both the easy and difficult image. As well as for the Laplacian filter on the easy image, we get a very clear outline of the person without many background noise. For the more difficult image, we get background noise on the final mask once again. However, the black dots corresponding to the noise in the background is in this case much smaller in size compared to the large black outline of the person. This made the process of removing noise after edge detection much easier without increasing risk of removing important details in the image.



(a) Threshold detection easy image

(b) Threshold detection difficult image

Figure 11: Edge detection using binary thresholding.

The final edge detection method using wavelet decomposition can be seen in fig. 12. Similar to the two previous methods, the edge detection works well on the easy image but we get more noise on the difficult image. The difficult image in the wavelet method suffers from a similar problem to the Laplacian filter where the size of the noise and the size of the persons border is of similar size, this makes the Black and White(BW) image manipulation difficult and the risk

of removing important features increases.

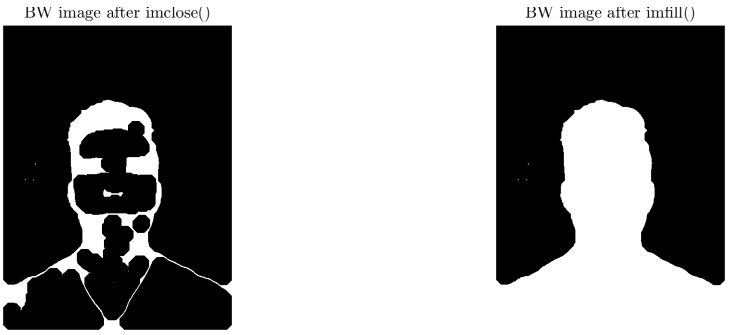
(a) Wavelet detection easy image (b) Wavelet detection difficult image

Figure 12: Edge detection using wavelet decomposition.

3.3 Finalisation

During the finalisation of our images, many different methods and matlab functions were used, some example results are shown below and some more detailed procedures can be found in the appendix of this document.

Imclose() is a really useful function to connect borders together and close the edges around the person if they were not perfect. An example result of this is shown in fig. 13a where the edges around the person are much more defined compared to fig. 10a corresponding to the image before using the imclose() function. If all pixels in the borders around the person are correctly closed and tracing a closed figure, the imfill() function can be used to fill this figure with the value 1 and the result would be similar to fig. 13b.



(a) Black and white image after imclose(). (b) Black and white image after imfill().

Figure 13: Black and white edge masks in different stages of mask manipulation.

One of the most useful finalisation tools was the function bwareopen() in matlab, which removes clusters of pixels with a certain size. A good example of the usefulness of this function can be seen in fig. 14 which is after using the function on the noisy image shown in fig. 11b.

Figure 14: Example using bwareopen() to remove noise.

In appendix A1 more examples of finalisation methods can be found if interested. Some examples of transformations from the first binary image of the original image with only one edge to a full image mask can also be found there.

3.4 Final images

When a functioning image mask had been created, the finalisation of the image can be seen in fig. 15. These images were created from the same binary thresholding algorithm. The binary threshold ended up being the best solution out of the ones that were tested when working with both the easy and the difficult image. All edge detection methods used on the easy image were able to create an image with a blurred background similar to fig. 15a. Due to the background noise, the most of these methods failed to create something similar to fig. 15b for the difficult image.

(a) Final result with easy background

(b) Final result with difficult background

Figure 15: The two final portrait images with two different background.

4 Discussion

Below follows thoughts on different processes and methods used in the previous sections and the effect on the final images.

4.1 Choice of Prepossessing

Two different philosophies have been tested in the preprocessing. The first approach aimed to smooth/denoise the image to make the effect on the background as small as possible. The second involved enhancing existing features, sush as sharpening, to get a better edge around the person in the images. For smoothing/denoising both lowpass filtering and thresholding wavelet coefficients were used. Both these methods work in removing some of the noise from a more complex background but generally the edges of the person in the images became less clear which also showed after edge detection. It was much more difficult to get a clear BW outline of a person if excessive low pass filtering/wavelet thresholding had been done before.

On the other hand enhancing the existing features in the image had the opposite effect. It usually made edge detection easier when simple sharpening had been done to the image but in the case of background noise the features of the background might have also been enhanced creating problems that could be seen in fig. 10b. Similar behaviour could be seen with histogram equalisation both face features but also background noise could become more visible.

The trade off between removing noise and enhancing edges becomes important when discussing the edge detection methods in the next section.

4.2 Choice of edge detection methods

When it comes to edge detection, both the Laplacian filter and the wavelet filter exhibited similar behavior in the sense that they effectively detected the borders. However, the Laplacian filter outperformed in terms of delineating the person's edges, but it also introduced more background noise, as observed in fig. 10b and the wavelet version with worse border but also less noise is seen in fig. 12b. In both of these cases, some form of sharpening was applied during the preprocessing progress. Using any form of denoising/smoothing as preprocessing usually made the resulting edges worse and the finalisation difficult. Another difference between the Laplacian and the wavelet approach was that because of the more detailed edges less steps had to be carried out to finalise the image mask. In the case of Laplacian edge detection the finalisation could be done in two steps, imclose() and imfill(), for the easy image. But with the wavelet method a total of 6 steps had to be done in order to finalise the mask which is found in appendix A1.

Another challenge with the wavelet method was the inconsistency in determining the appropriate wavelet layer where the correct details could be found. Deciding which layer to cancel coefficients and where to enhance proved to be a tedious manual process. Automating this process in some way could prove to yield useful in both denoising and enhancing the face, for example an algorithm that goes through each layer and from the standard deviation of the histogram estimates if it is a layer with a lot of noise or important details could be one approach.

The binary thresholding for the two chosen images proved to be the best performing approach, it was the only approach that could automatically work on a new image without changing much of the parameters, only possibly tweaking the threshold. What made this approach particularly effective, unlike the other two, was the distinction between pixel clusters representing noise and those depicting the person. All noise was located inside small clusters around the background, but the pixels corresponding to the person remained one big cluster. This made it quite easy to use BW image manipulation to remove the background noise. As for the example the difference between fig. 11b which is the binary thresholding image before BW manipulation and fig. 14 which is after some BW manipulation aimed at reducing noise. If the same procedure to denoise would be used on any of the other methods, i.e, fig. 10b or 12b large portions of the important details would also be removed.

A challenge that is somewhat visible in the final images is that there seems to be a tiny "sticker" effect that we could not fully remove. In certain places parts of the background comes with the image mask and creates a tiny border around the person. This effect could be minimised by thinning the borders of the image mask. But if the border was thinned to much however details of the person was being removed so some of the sticker effect was left just to not remove to much of the person. This is an aspect that could be tweaked in the future in order to find what would be the most pleasing to the viewer and might also depend on the image.

4.3 Future works

One of the main bottlenecks with creating a portrait mode have been the preprocessing of the image, the tradeoff between denoising and keeping important details was difficult and most of the time resulted in weird artifacts in the edge detection image. It is reasonable to assume that much more work can be done to improve the preprocessing, thus far only quite simple techniques had been utilised and it would be interesting to test more advanced methods that could both denoise effectively while keeping important details. As an example an adaptive histogram equalisation similar to what was done in [6] by Z. Hameed and C. Wang would be interesting to use instead. In this approach they did a local type of equalisation where they took into account local signal level in order to enhance important gradients in the image but not simultaneously enhance noise. This is exactly the problem that was encountered in this projects preprocessing

and such an approach might have proven to be useful.

Similar ideas of localised detection would be interesting to investigate for edge detection as well. Under the assumption that the person in the image always is in the center removing certain areas of the image where we know with high probability that there will be no people would make it much easier to deal with noise. Giving emphasis to central image pixels when doing edge detection could also prove to reduce the effects of the background while also resulting in better edges.

Finally to address the BW manipulation. The main problem with the finalisation was that matlabs built in functions for BW manipulation was not always suited for the type of manipulations that we wanted to do. Functions such as imfill() proved to be quite sensitive, particularly when dealing with isolated pixels, which can pose challenges when working with larger images. If time would allow, writing some functions from scratch could prove to make the finalisation both easier and more consistent. For example, developing a function that initiates from a random position in the central region of the image, where a person is most likely to be located, and then systematically fills in pixels around it until the periphery of the person is reached could be a valuable addition. If specialised functions were designed for outlining a person, they could be more tailored and less generic compared to matlabs built-in functions.

5 Conclusion

Creating a functioning portrait mode is a difficult task. In the above report different theories, ideas and methods related to domain analysis has been done to explore and evaluate methods mentioned in this document. Different kind of denoising and sharpening methods have been implemented and used on our images before trying to detect the face on them via edge detection. Finally, the final image was completed by BW image manipulation and background blurring with a Gaussian filter. For the tested images, binary thresholding was an effective way of getting a good edge detection and being able to keep the noise level low. The most challenging tasks during the investigation was preprocessing the images in order to get a good edge detection. As the edges detection seems already pretty efficient in matlab, we believe that the face detection in images can be improved a lot by finding better preprocessing methods. In our preprocessing treatments, we used either sharpening or smoothing methods but ideally using a combination of these methods could improve the results.

References

[1] Rafael C. Gonzalez, Richard E. Woods, Chapter 3 - Smoothing (Lowpass) Spatial Filters, Digital Image Processing, Fourth Edition, 2018, Pages 166-167, ISBN 978-1-292-22304-9

[2] Rafael C. Gonzalez, Richard E. Woods, Chapter 3.3 - Histogram Processing, Digital Image Processing, Fourth Edition, 2018, Pages 133-136, ISBN 978-1-292-22304-9

[3] Phillip A. Mlsna, Jeffrey J. Rodríguez, Chapter 19 - Gradient and Laplacian Edge Detection, Editor(s): Al Bovik, The Essential Guide to Image Processing, Academic Press, 2009, Pages 495-524, ISBN 9780123744579, https://doi.org/10.1016/B978-0-12-374457-9.00019-6. (https://www.sciencedirect.com/science/article/pii/B9780123744579000196)

[4] Rafael C. Gonzalez, Richard E. Woods, Chapter 7 - Wavelets and Other Image Transforms, Digital Image Processing, Fourth Edition, 2018, Pages 463-538, ISBN 978-1-292-22304-9

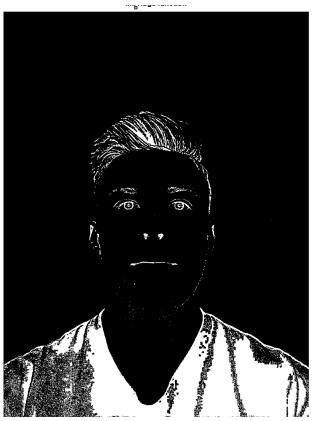
[5] The Mathworks Inc "Wavelet Toolbox Documentation", mathworks.com. Accessed: October 19, 2023. [Online]. Available: https://se.mathworks.com/help/wavelet/ref/wdenoise2.html

[6] Z. Hameed and C. Wang, "Edge detection using histogram equalization and multi-filtering process," 2011 IEEE International Symposium of Circuits and Systems (ISCAS), Rio de Janeiro, Brazil, 2011, pp. 1077-1080, doi: 10.1109/ISCAS.2011.5937756.

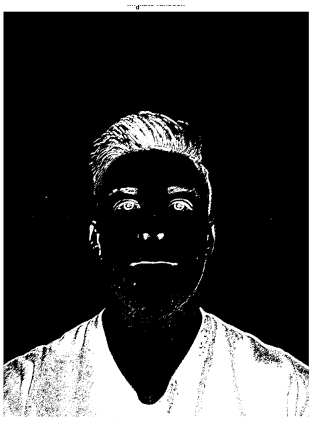
6 Appendix

6.1 A1

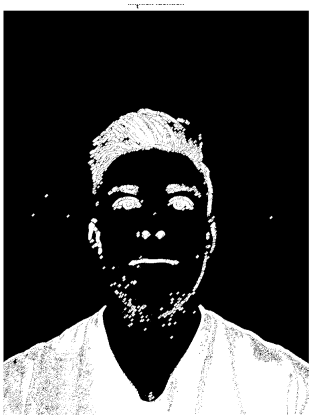
6.1.1 A1: Finalisation after wavelet edge detection



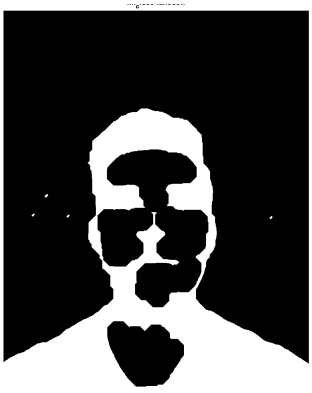
(a) after bwmorph('bridge').



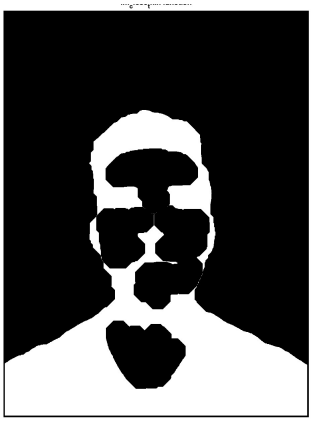
(b) after imdilate().



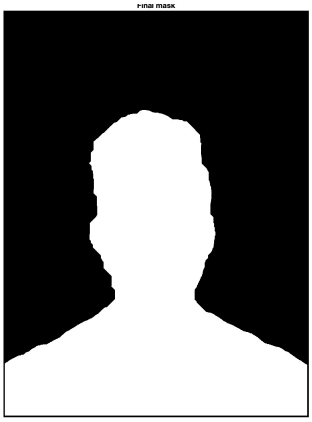
(c) after bwmorph('thicken').



(d) after imclose().



(e) after bwmorph('thin') and bwareaopen



(f) final mask.

Figure 16: Black and white edge masks in different stages of mask manipulation.