University of Southern Denmark

Introduction to Robotics and Computer Vision

Vision – Mandatory Exercise 1
Image restoration

by:
Keerthikan Ratnarajah
kerat12@student.sdu.dk
Jes Grydholdt Jepsen
jejep12@student.sdu.dk

Dato: November 8, 2015

Todo list

Contents

1 Introduction

In the real world images are sometimes affected by noise in a way that makes working with them (or looking at them) difficult. The purpose of this project is therefore look at image restoration on a set of images that are affected by different kinds of defects. The overall task is to minimize the impact of the noise and thereby to improve the quality of the image.

To come up with an analysis and a way to remove (or weaken) the defect, the following considerations are done for each image:

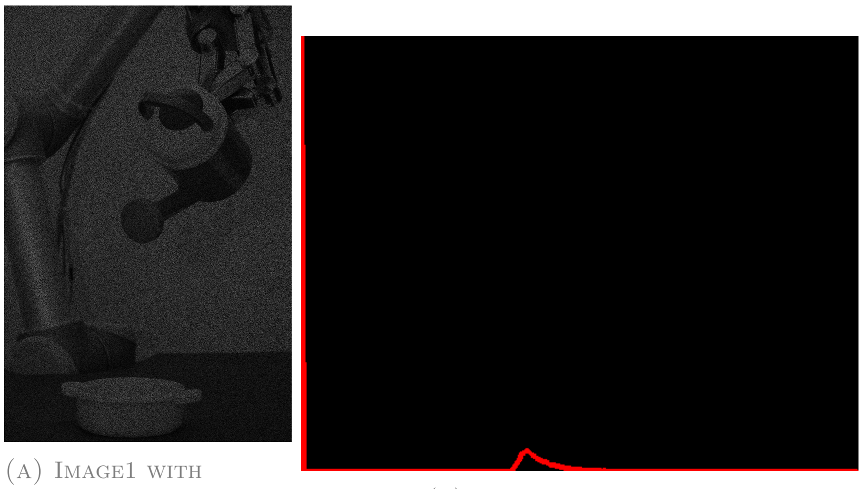
- Investigate the image and identify the defect, for example by using the histogram and/or the frequency spectrum of the image.
- Design a solution that removes or weakens the impact of the defect and investigate the properties of the solution.
- Investigate different solution possibilities.
- Implement and apply the solution(s)

At the end the image restored has to be resemble the original image as best as possible.

Figure 1: The original image

2 Image 1

Figure 2a shows the first image which has to be restored. This image is compared to the original (Figure 1) much darker. Based on the histogram it can be seen that a large amount of black pixels appear to be on the image. The image affected by pepper noise.

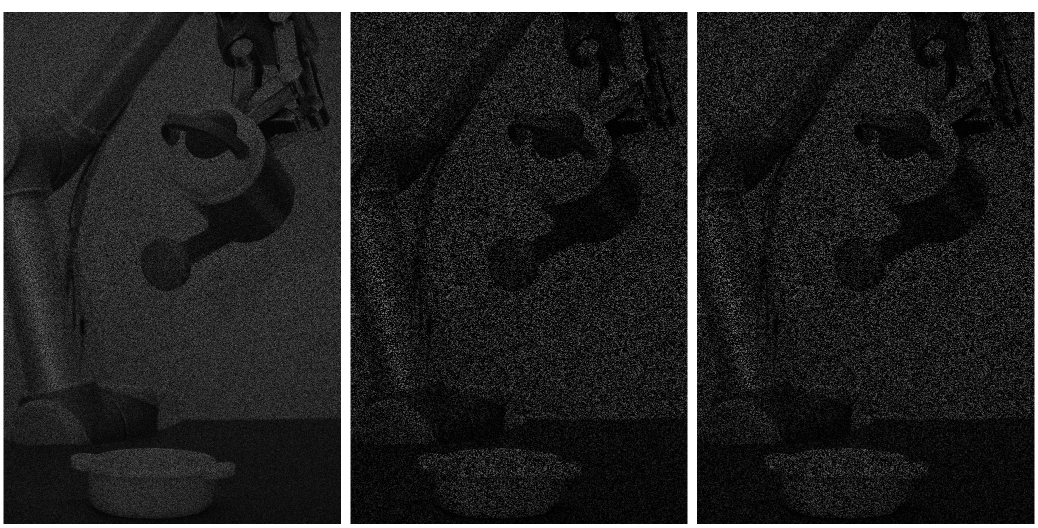


(a) Image1 with no restoration

(b) Histogram of Image1

Figure 2: Analysis of image 1

An effective way of removing salt and pepper noise it to use an median filter, which would take the median value of fixed numbered values and assign it to that pixel position, but as the amount of pepper noise is too large, will an median value mostly lead to a black pixel, thus not provide that much of an improvement as seen in Figure 3.



(a) Image1 with no restoration

(b) Image1 filtered with median filter kernelsize 3

(c) Image1 filtered with median filter kernelsize 9

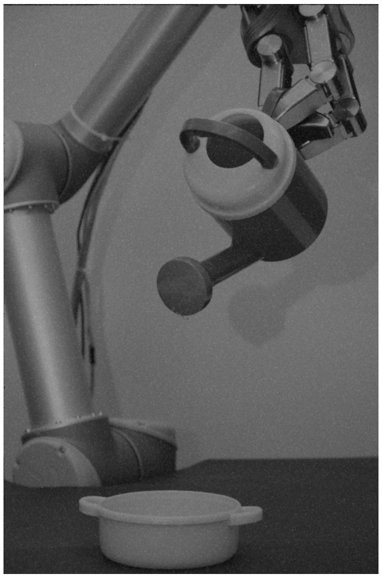
Figure 3: Median filter applied to image1

As the Median filter did not show any form of improvement, was it deemed not unusable for this use case, instead were an contraharmonic mean filter implemented and tried.
An contraharmonic mean filter restores an images using this expression

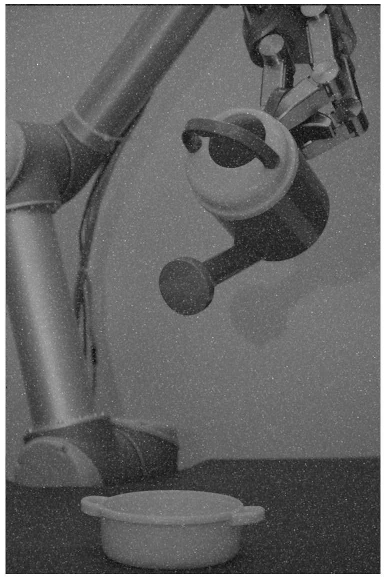
$$\hat{f}(u,v) = \frac{\sum\limits_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum\limits_{(s,t)\in S_{xy}} g(s,t)^{Q}} [\textbf{?}] \tag{1}$$

citation has to be added to the book p. 345

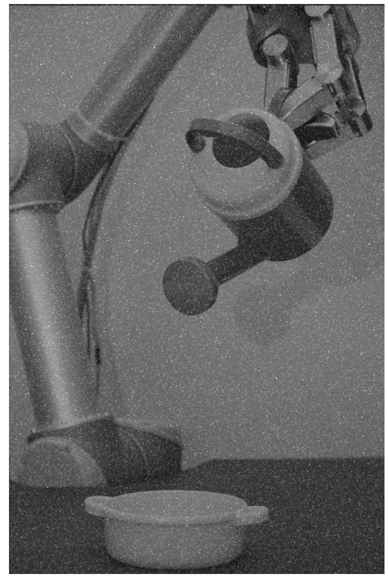
In Equation 1, is S_{xy} a rectangular window with a center pixel. g is the corrupted images, and $\hat{f}$ is the restored image. Q defines the order of the filter.
The filter is suited for removing salt and pepper noises. The filter reduce the effect of pepper noise for positive Q values, and Salt noise for negative Q values. Besides this is this filter also capable of being reduced to an arithmetic filter by setting $Q = 0$ or harmonic filter by setting $Q = -1$. The filter was implemented due to its multiple applicability, and applied on Figure 2a with different positive value which the result can bee seen on Figure 4, to test how well it well it could restore the image to its original condition.



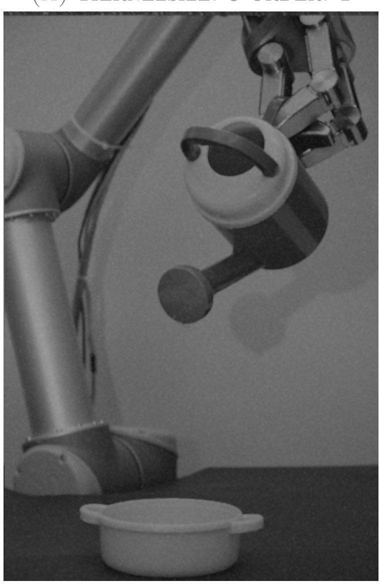
(A) Kernelsize: 5 order: 1



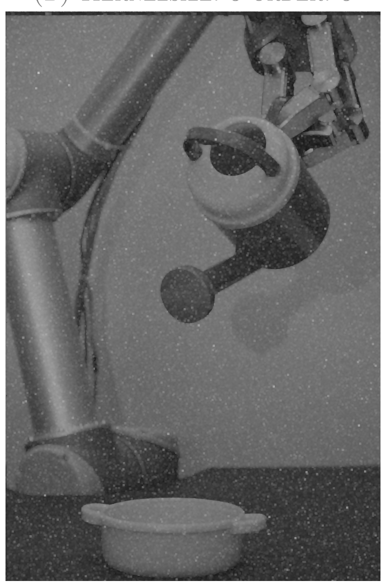
(B) Kernelsize: 5 order: 5



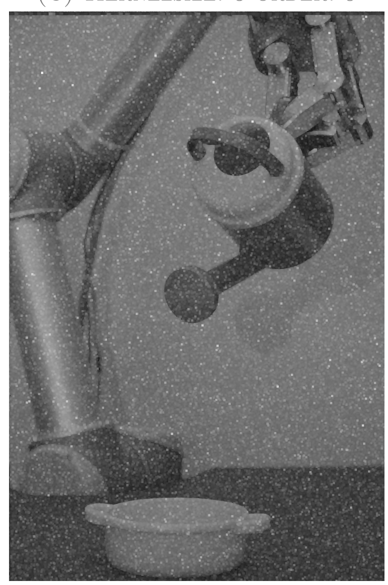
(C) Kernelsize: 5 order: 9



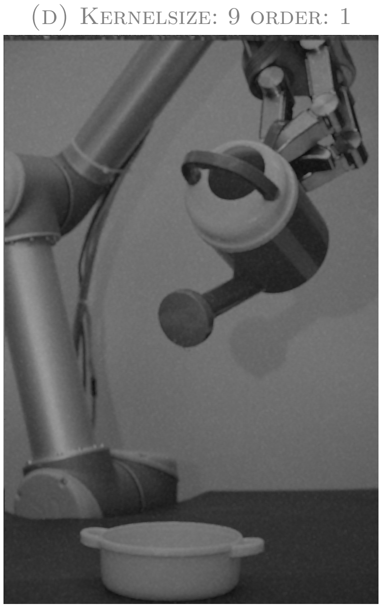
(D) Kernelsize: 9 order: 1



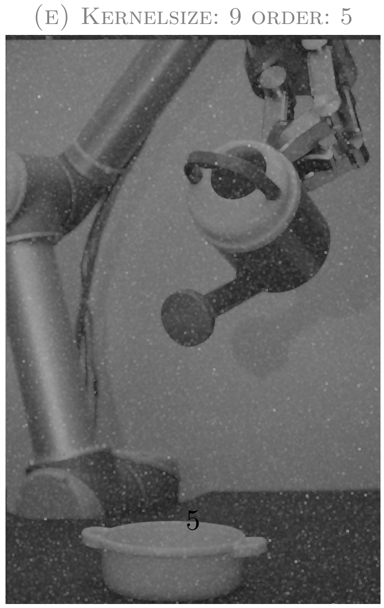
(E) Kernelsize: 9 order: 5



(F) Kernelsize: 9 order: 9



(G) Kernelsize: 11 order: 1



(H) Kernelsize: 11 order: 5

(I) Kernelsize: 11 order: 9

Figure 4: Contraharmonic mean filter applied on Figure 2a with different kernel sizes and orders

By visual inspection it can be seen that the contraharmonic mean filter has removed most of the black pepper noise which was visible in Figure 2a. As the kernel and order size increases it can be seen that output image becomes more blurred and more of the pepper noise get removed but also introduces a salt noise as the order increases, for which order sizes above 1 was deemed not usable for the restoration process. The only filter which was capable of removing most of the pepper noise, and still keep the image minimally blurred is the output from the filter with an kernelsize of 5, and order 1 Figure 4a. Which was used for further processing.

The image is still a bit dark compared to the original image, to reduce its poor contrast was a method called contrast stretching implemented and applied to that image. This method tries to expand the range of intensity levels the image contains, such that the full intensity range of the image will be used.

ciite http://www

This is done by modifying each pixel in the image using Equation 2

$$\hat{f}(x,y) = \frac{g(x,y) - minP}{maxP - minP} \cdot 2^{bpp} - 1 \tag{2}$$

$\hat{f}$ is the new image, (x,y) is a pixel position, g is the original image, $minP$ is the lowest pixel value in image g, $maxP$ is the max pixel value in image g andbpp, is the bit depth of the image.
The result of using this method can be seen here.

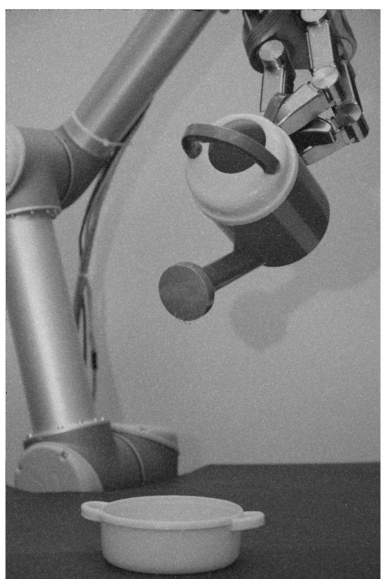
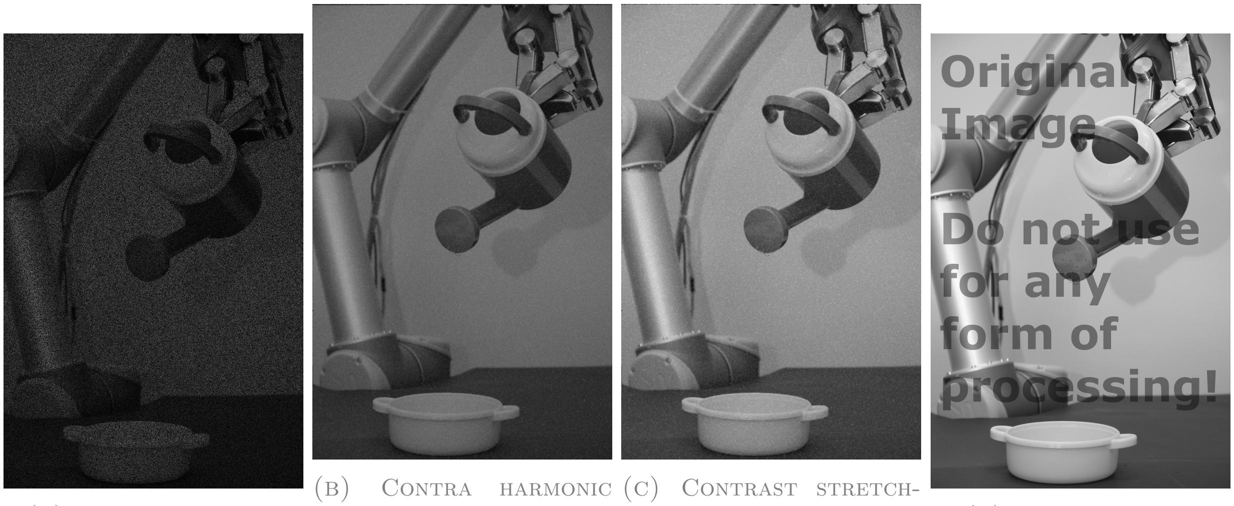


Figure 5: Kernelsize: 11 order: 1

At this point it was determined that image has been restored enough to resemble the original image.

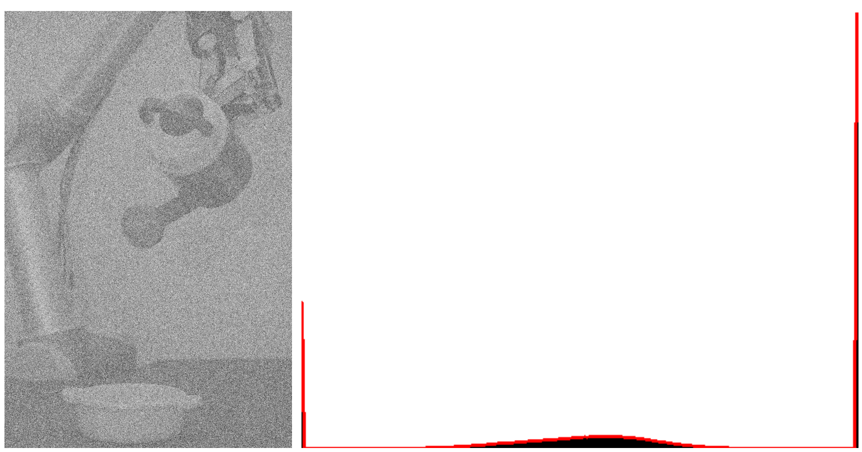


(A) Image with noise (B) Contra harmonic mean filter applied (C) Contrast stretching applied (D) Original image

Figure 6: Image restoration process

3 Image 2 – Salt and Pepper noise

Figure 11q shows the original image, which is filled with black and white pixels. This kind of noise is known as salt–and–pepper noise, and from the image's histogram (figure 11r) can the amount of salt–and–pepper noise be seen; salt noise on the right and pepper noise on the left.



(A) The original image (B) Histogram of the original image

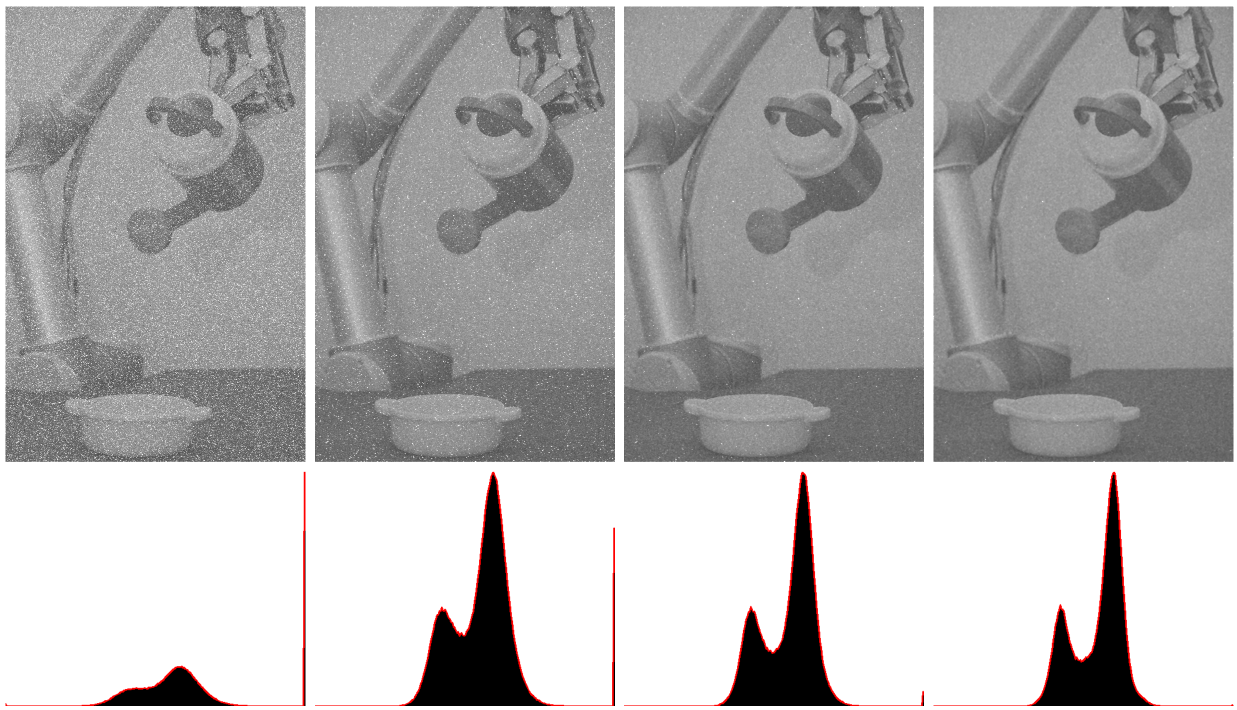
Figure 7: Analysis of image 2

Approximately are there three times more salt noise compared to the pepper noise. The values in the middle of the histogram is what is left of the original image, and in order to restore as much as possible, and median filter is applied on the image. The median filter is chosen because it is very effective against salt-and-pepper noise in the images, and the OpenCV function

```
void medianBlur(InputArray src, OutputArray dst, int ksize)
```
[1]

is used for applying the filter to the image. Basically the image blurred by using the median filter, and depending on the size of `ksize`, the more blurred will the image become.

The `ksize` is the size of the kernel filter applied to each pixel in the image, and therefore must the value of `ksize` be odd and greater than 1, so in order to test if the noise is removed, a kernel of `3` is applied on the image 2 and checked if all the noise is removed. If not all the noise is removed, the kernel size is increased with two, and checked again, and so on.

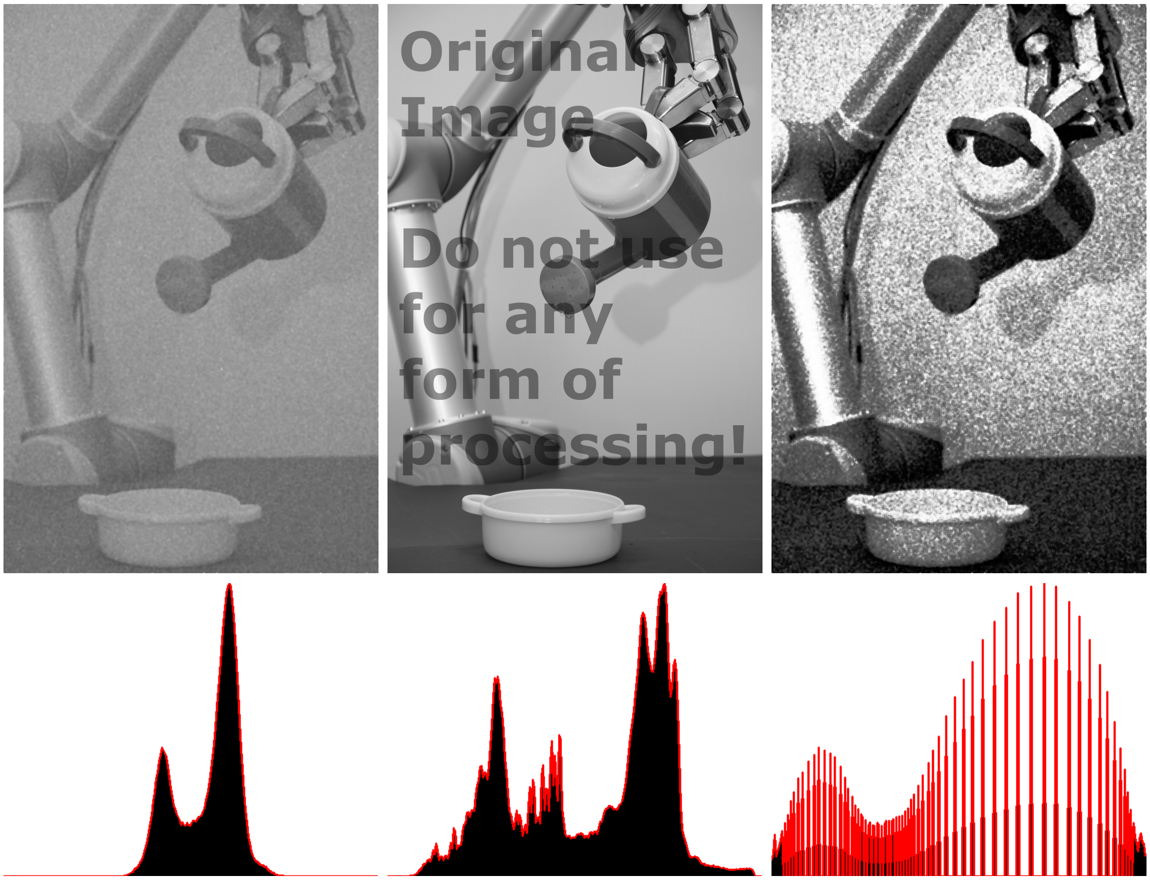


(A) KSIZE = 3 (B) KSIZE = 5 (C) KSIZE = 7 (D) KSIZE = 9

FIGURE 8: ANALYSIS OF IMAGE 2

Figure 8a, 8b, 8c and 8d shows the process of finding the right `ksize`, but all four kernel sizes still does not remove all the noise, especially the salt noise. All the salt–and–pepper noise is removed with a `ksize=11`, but the disadvantage of this approach is that the details in the image are reduced. For restoring as much as possible of the these details, a histogram equalization is applied on the noise reduced image. Histogram equalization restores the details by improves the contrast in the image by stretching out the intensity range of the image. On the left and right side of image 9a's histogram are there underpopulated intensities, which is why an histogram equalization is an optimal choice. The result is shown on figure 9c.

[1] `http://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html#voidmedianBlur(InputArraysrc, OutputArraydst,intksize)`



(a) ksize = 11 (b) Equalized (c) Equalized

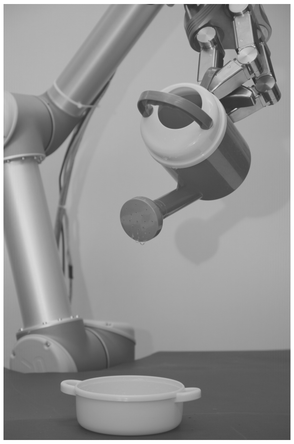
Figure 9: Analysis of image 2

Which one is best?

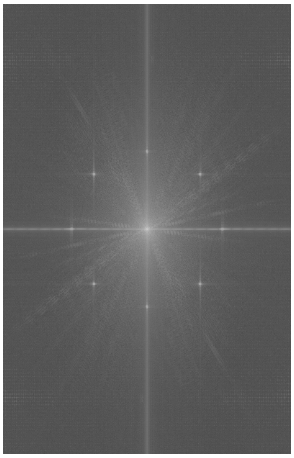
4 Image 3

5 Image 4

Figure 10a shows Image4_2 which has to be restored. This image compared to the original (Figure 1) has some both horizontal and vertical stripes. Based on the frequency spectrum of this image, it can be seen that some frequency component exist around the center frequency creating this effect. The purpose of this restoration will be to reduce the effect of these component, and reconstruct it so it resembles Figure 1



(A) Image4_2 with no restoration



(B) Frequency spectrum of Image4_2

Figure 10: Analysis of image 1

The frequency components creating the circle are reason why the the stripes are occuring in the image. One way of resolving this issue is to create a low pass filter, which let everything below a certain frequency pass, and the filter frequencies above away.

One type of lowpass filter is the butterworth lowpass filter which is defined as

$$H(u,v) = \frac{1}{1+[\frac{D(u,v)}{D_0}]^{2n}} [?] \tag{3}$$

D_0 is the cutoff frequency given as the distance from the origin, n is the order, and D(u,v) is distance between a point (u,v) in the frequency domain and the center frequency rectangle, it is calculated as such.

Cite til bogen side 295

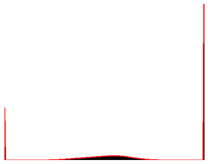
$$D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{\frac{1}{2}} \tag{4}$$

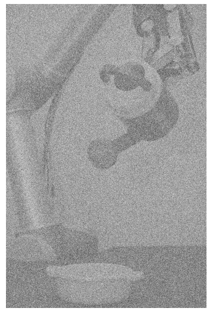
P and Q is the padded size of the rectangle.

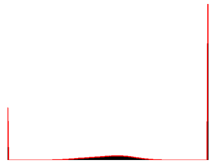
The cutoff frequency can easily by found, by computing the distance between the origin and the corresponding frequency component. The origin i as the pixel position $(1536, 2408)$, and one of the undesired frequency component lies as pixel position $(2336, 2400)$

This distance is computed to be 800, which means that D_0 below 800, should be able to filter out the frequency components causing the stripes.

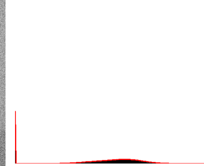

(a) The original image


(b) Histogram of the original image

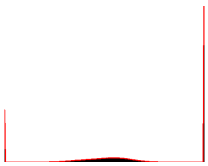

(c) The original image

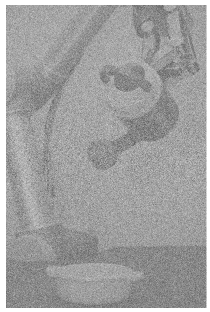

(d) Histogram of the original image

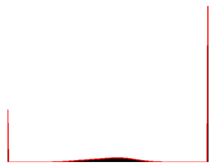

(e) The original image


(f) Histogram of the original image


(g) The original image

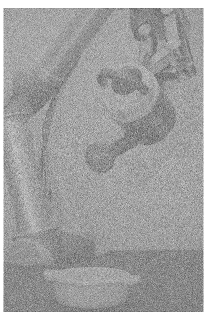

(h) Histogram of the original image

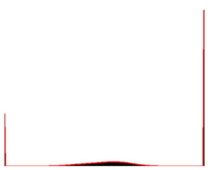

(i) The original image

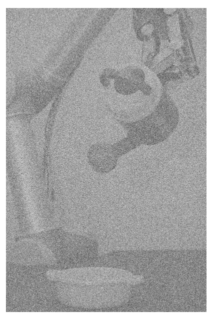

(j) Histogram of the original image

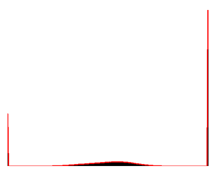

(k) The original image


(l) Histogram of the original image


(m) The original image


(n) Histogram of the original image


(o) The original image


(p) Histogram of the original image


(q) The original image


(r) Histogram of the original image

From the figure (comming) it can be seen that mostly weakens the effect of the frequency components, an interesting fact which also can be seen from the plot, is that as the order of the filter increases, so increases the appearance of ringing noise. This is due to filter becoming more ideal thus having the transition occur more abrupt as shown in 12 .

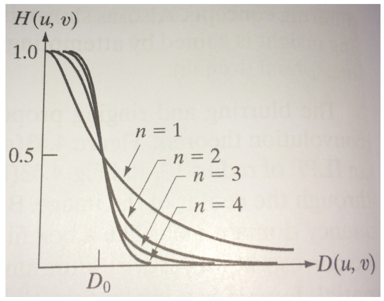


Figure 12: Filter becomes more ideal as the order increases.

6 Image 5