RoVi. Vision mini-project 1

Adrián Varga and Javier Javier López-Randulfe

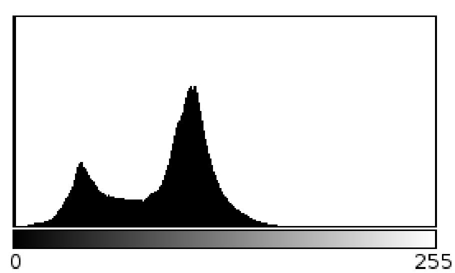
University of Southern Denmark, Odense, Denmark
{advar17, jalop17}@student.sdu.dk,

1 Introduction

This report summarizes the algorithms performed over the provided images for the *Vision mini-project 1* of the RoVi1 course. The goal of those algorithms is to restore the images, and provide an Ad-hoc solution to each one, as they present different artifacts or distortions. The proposed solutions encompass a mixture of processes from the spatial and frequency domain, while the used analysis tools are the histogram, the frequency magnitude, and the human eye.

For obtaining better histograms, an additional software tool called imageJ [2] has been used. In Fig. 1 can be compared the obtained histograms with imageJ and the obtained with the developed C++ code, using the OpenCV library. The results are numerically the same, but imageJ include operations like auto-scale which make it easier to analyze for the human eye. For this reason, the results obtained with imageJ will be used in the rest of the document.

(a)



(b)

Fig. 1. Histograms of the first image, a) with the handmade code in C++, using OpenCV tools, and b) using the imageJ software tool.

2 Image 1

2.1 Detected flaws

The first image presents a high amount of pepper noise. The noisy pixels of this kind of noise are characterized for being reduced to a 0 intensity value. Despite this pattern can be easily noticed in the original image by the human eye, the histogram of the image offer a more objective criteria. Looking at it, it can be observed that a big amount of the image pixels have an intensity value of 0.

Besides the aforementioned noise pattern, the pixels containing useful information also have, in general terms, low intensity values, resulting in the perception that the image is dark. In other words, the image has a low contrast, and the histogram is shrunk to the leftmost bins.

2.2 Restoration algorithms pipeline

Stage 1 - Adaptive max filter: The first operation applied to the image was an adaptive max filter, which is a modification of the max filter. The max filter merely consists in substituting each pixel by the maximum value of its neighbours (including the pixel itself too).

For this image, the max filter has been slightly modified, using a reasoning similar to the adaptive median filter i.e. for a given pixel, the minimal neighbourhood of 3x3 is inspected, and the maximum intensity value is registered. In case that its value is 0, the process is repeated with the next neighbourhood size (always odd kernel dimensions are chosen: 3x3, 5x5, 7x7...). This process is iteratively repeated until a non-zero value is obtained, or the maximum specified kernel size is reached.

Prior to applying the filter, the image was padded with the same amount of rows and columns as the size of the maximum allowed kernel, and were filled with zeros. The value of 0 does not affect the result, as the adaptive max filter will prevent it to be present in the resulting image

Stage 2 - Median filter: After applying the aforementioned adaptive max filter, some sparse pixels with undesired high values appeared. This was expected, as the max filter always takes the highest value of the neighbourhood, and thus the pixels are affected by high intensity neighbours. For minimizing this problem, a median filter with a 3x3 kernel was used.

Stage 3 - Intensity enhancement: After applying both the adaptive max filter and the median filter, most of the original noise was removed from the image. However, the resulting image had a poor contrast, and the dark tones prevailed on it. For solving this problem, several solutions were considered.

The first and more simple solution was to add a constant arbitrary value to all the pixels, based on the histogram information. Despite this alternative improves the image, it has a few drawbacks: the election of the intensity enhancement is arbitrary and there are not very clear objective criteria for choosing one, if this value is high enough, several pixels will saturate to the maximum value, and it is an ad-hoc solution, being difficult to be applied to other images with similar flaws.

Thus, it was opted instead to use a intensity transformation for improving the histogram of the image. The one chosen was the logarithmic transformation, with the c parameter set to 1, whose effect is an expansion of the dark pixel intensities of the image, mapping them to brighter values. Despite there are other popular transformations, this one seemed adequate for the this concrete image.

2.3 Results and discussion

In this section, the images resulting from each applied operation are presented, altogether with a discussion of the chosen parameters and the obtained image.

First of all, the maximum radius for the adaptive max filter had to be specified. From the resulting histograms in Fig. 2, it can be observed that for a maximum radius of 1 (3x3 kernel), the histogram still presents several pixels with an intensity of 0; if the maximum radius is 2 (5x5 kernel), the noisy pixels are almost reduced; and if it is 3 (7x7 kernel), the completely disappear. Hence, this was the chosen value for the maximum radius (Note that for bigger values, the result will be the same, as after a radius of 3, there will no longer be pixels with a value of 0).

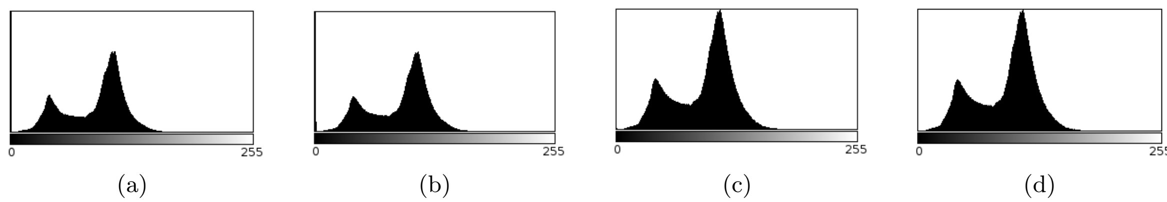


Fig. 2. a) Histogram of the original image, and after being filtered with an adaptive max filter with b) 3x3 maximum kernel size, c) 5x5 maximum kernel size, and d) 7x7 maximum kernel size.

As explained before, a median filter was applied to the image after the adaptive max filter, so some of the sparse bright pixels are removed from the image. Despite no relevant changes are appreciated in the histogram, this operation improved the final result after the whole pipeline, as these sparse pixels get amplified after doing a contrast enhancement.

Finally, the histogram results of the contrast enhancement operation can be observed in Fig. 3, for both the simple intensity enhancement (adding a value of 60 to each pixel), and the log transformation. The images resulting from the second stage and third stage are depicted in Fig. 4. The result of the first stage is not included, as the visual aspect is very similar to the result of the second stage.

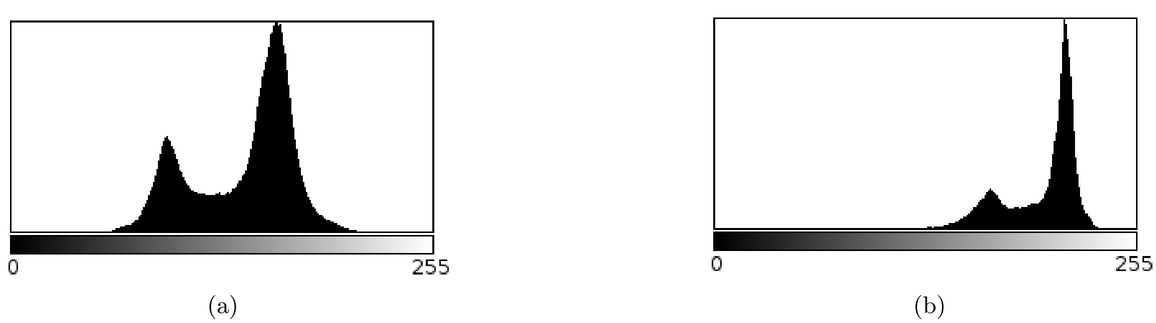
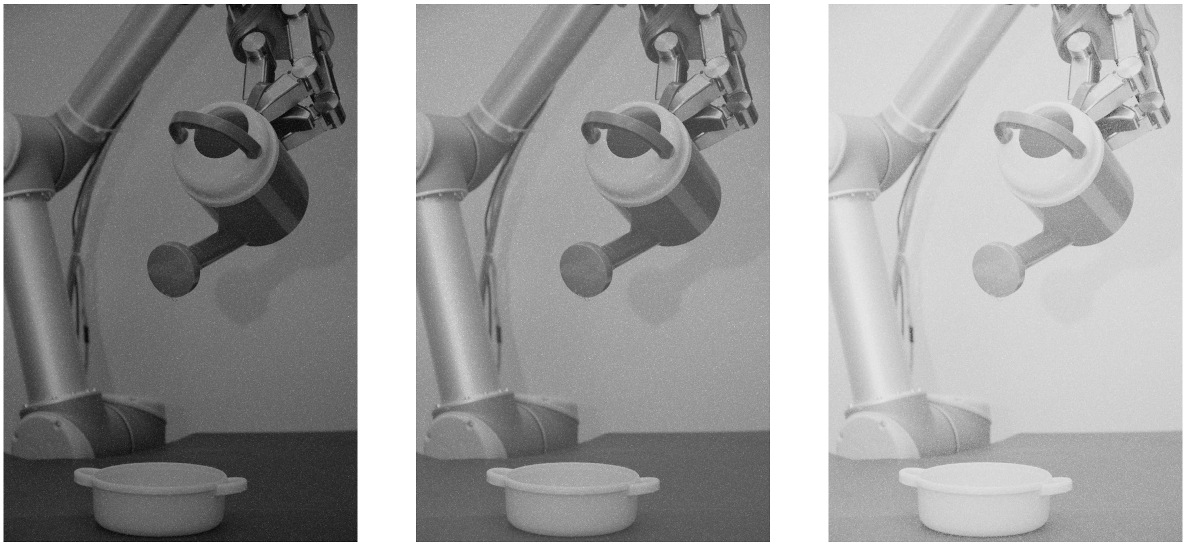


Fig. 3. Histograms of the image, after being enhanced with an a) increase of the intensity of each pixel by 50, and b) after being applied a log transformation.



(a) (b) (c)

Fig. 4. Resulting images at the different stages of the pipeline. a) After performing the adaptive max filter with a 7x7 maximum kernel, plus a median filter, b) After increasing each pixel of the filtered image by a value of 50, and c) After applying a logarithmic transformation to the first filtered image.

3 Image 2

3.1 Detected flaws

In the second image, salt and pepper noise is present within the pixels with useful data. The noisy pixels of this pattern are typically saturated to the maximum intensity value (255 in this case), or set to the minimum (0 in this case). This can be observed at the image histogram (Fig. 5), where 2 big spikes are present at the 0 and 255 intensity values. A particular feature of this noise is that the pixels that are not affected by it keep the original clean value, contrary to other distributions, like Gaussian o Rayleigh.

Moreover, the image contrast is low. In other words, the histogram of the image is shrunk in a narrow area, which makes the image harder to be interpreted by the human eye, as the intensity differences between different regions are then smaller.

3.2 Restoration algorithms pipeline

Stage 1 - Adaptive median filter: The initial task of the proposed pipeline is, as well as with the previous image, to remove the majority of the present noise. However, the max filter can not be used, as the pixels with salt noise would have a very big influence in the filter output. Instead, an adaptive median filter based in the algorithm proposed in [1] has been designed and applied to the image. The main parameter of this filter is the maximum window size that will be used for looking for non-zero and non-saturated neighbours.

Stage 2 - Adaptive max filter: After the first stage, the results obtained still present a small amount of pepper noise. One possible reason for this is the used padding prior to the application of the filter. In order to remove the noise in these pixels, an adaptive max filter, equivalent to the one in the previous section, has been applied to the image. If the parameters for these first two filters are correctly selected, there should be an absence of salt & pepper noise in the image.

Stage 3 - Intensity transformation: While the previous image has mainly dark intensity values (left part of the histogram), the second provided image has a centered histogram. Then, if the same logarithmic transformation were to be applied, the transformed image would be very bright, which is not very desirable. For this reason, a gamma correction algorithm is proposed instead, as the gamma parameter can be tuned in order to perform transformations closer to the identity transformation (when the output and input images are the same).

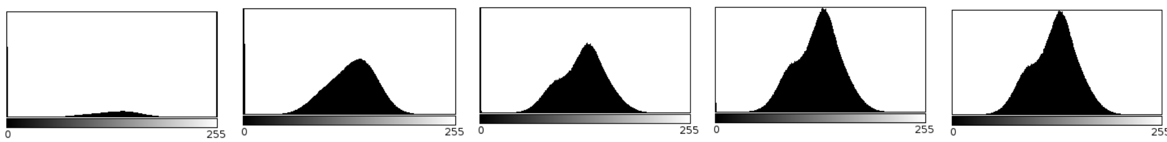
Stage 4 - Smoothing: After applying all the previous stages on the image, it has been observed that the output image contained an undesirable amount of noise. Hence, a smoothing algorithm was applied to the whole image, in order to minimize the impact.

The ones which were tried were the mean and the Gaussian filter, where the tuned parameter was the kernel size.

3.3 Results and discussion

In the following paragraphs, there is a discussion over the results and differences between the different applied algorithms.

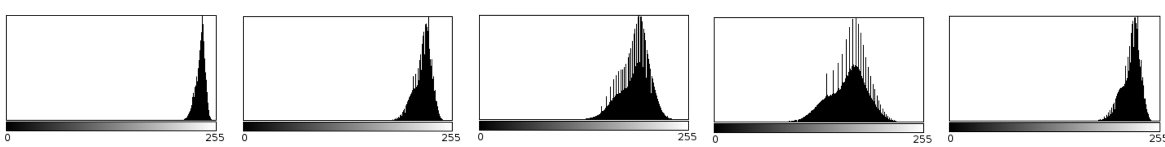
Regarding the adaptive median filter, it has been applied with different maximum window sizes, and the histogram has been plotted afterwards (Fig. 5). It can be concluded that there is no additional improvements for the salt noise if the window maximum size is bigger than 3x3 (Despite pepper noise is still present). As with the max_filter, once it is obtained a window size that allows all the pixels to be filtered, bigger windows will not imply more changes to the pixel intensities. Then, the adaptive max filter is applied to the image with a maximum kernel size of 5x5. For the human eye, the impact is small, but a relevant amount of pepper noise pixels are cleaned with this operation.



(a) (b) (c) (d) (e)

Fig. 5. a) Histogram of the original image, and after being filtered with an adaptive median filter with b) 3x3 maximum kernel size, c) 5x5 maximum kernel size, d) 7x7 maximum kernel size, and e) an adaptive max filter with a 5x5 maximum kernel size on the image resulting from b).

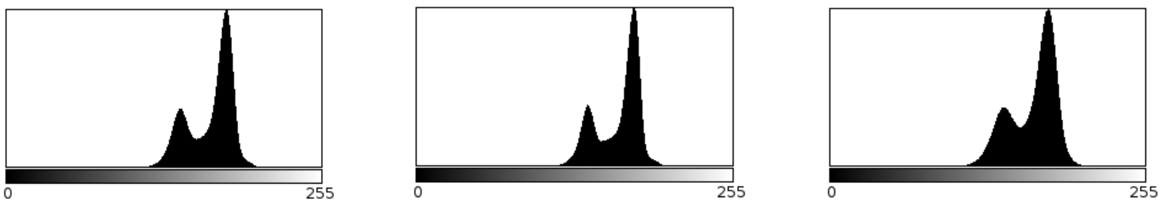
After cleaning most of the noise, different intensity transformations were tested. In Fig. 6 it can be observed that the logarithmic transformation results in very bright images. Hence, gamma transformations with different parameter values were applied, and it was decided to choose the one with $\gamma = 0.6$, as it seemed to be the image with the most adequate brightness to the human eye. Besides the previous subjective criteria, the histogram is moved to the centre-right corner, without being very exaggerated.



(a) (b) (c) (d) (e)

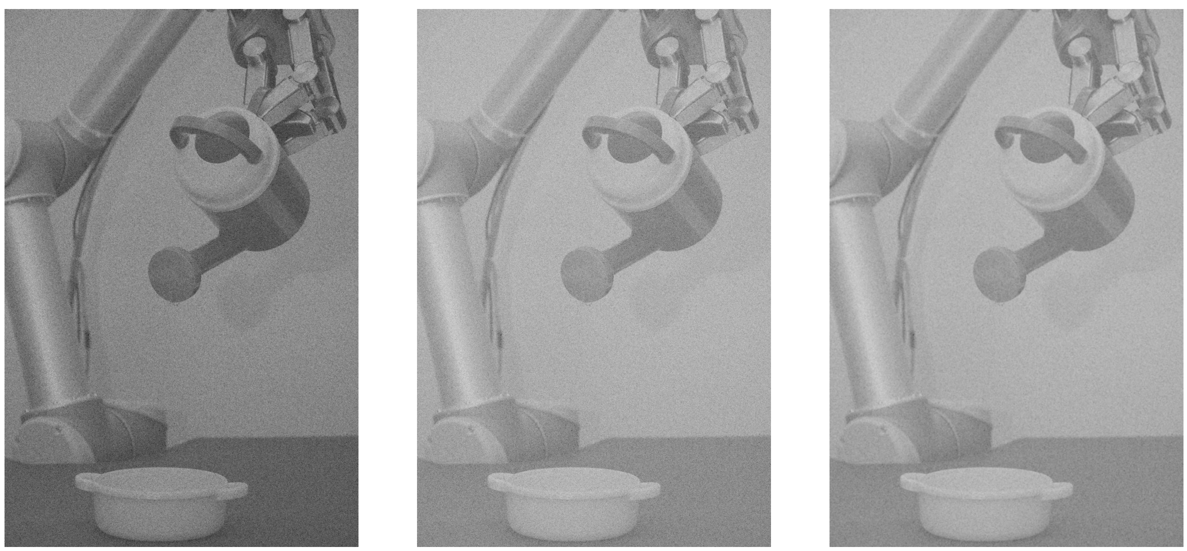
Fig. 6. Histogram of the filtered image after enhancing its histogram with a gamma transformation for a) $\gamma = 0.1$, b) $\gamma = 0.2$, c) $\gamma = 0.4$, d) $\gamma = 0.6$, and e) after using the logarithmic transformation.

Finally, different smoothing algorithms have been applied to the transformed image. Namely, the mean filter and the Gauss filter, trying different kernel sizes. From Fig. 7 can be concluded that the histogram of the Gauss filtered image with a 5x5 kernel seems to be the best, so it was the chosen one. This resulting images after the second, third and fourth stage of the pipeline can be observed at Fig. 8.



(a) (b) (c)

Fig. 7. Histogram of the image after smoothing it with a) a Gauss filter with 5x5 kernel size, b) a Gauss filter with 7x7 kernel size, c) a Mean filter with 3x3 kernel size.



(a) (b) (c)

Fig. 8. Original image after the different steps of the applied pipeline a) After adaptive median and max filters, b) After enhancing the histogram with a gamma correction, setting $\gamma = 0.6$, c) After applying the 5x5 Gauss filter.

4 Image 3

4.1 Detected Flaws

The third image has flaws that can be seen through visual inspection, and also the histogram. From first glance it seems obvious that there is some kind of additive noise that is neither salt nor pepper. It is not easy to determine which type of noise it is, but a first guess could be that it is Gaussian.

Looking at the histogram (Fig. 9), we can observe that its distribution is somewhat stretched compared to the others (possibly a result of Gaussian noise), and that there is a big spike of pixels at maximum intensity. Therefore, we can hypothesize that the image has either received some kind of positive additive noise or received a noise and then was shifted to higher intensities, or maybe the camera was overexposed and noisy. Either way there is definitely a loss of information from the high number of maxed out pixels.

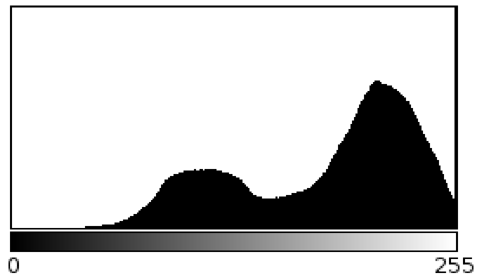


Fig. 9. histogram of the original image, with more than 80000 pixels on max intensity

4.2 Restoration algorithms

Due to the somewhat unknown nature of the flaws It is useful to test out multiple approaches. Each used spatial filter will be elaborated upon in this section with varying extent, and in the next section their results will be compared to each other.

Approach 1: Adaptive Noise Reduction (modified):

The Adaptive Noise Reduction filter works on the basis of a global and a local variance. By comparing the variance of the noise with the variance of the local kernel, it is possible to gain insight into differentiating between the added variance from the noise, and the variance in the actual image without the noise. Using this to our advantage the simple smoothing filter can be applied adaptively, while also preserving edges within the actual image. The equation for this filter is the following:

$$\hat{f}(u,v) = g(u,v) - \frac{\sigma^2_{glob}}{\sigma^2_{loc}}(g(u,v) - m_{loc}) \tag{1}$$

where $g(u,v)$ is the intensity of pixel with coordinates u,v, the sigmas are the deviations within the two respective kernels, and m_{loc} is the mean of the local kernel. There is also a condition, requiring that $\sigma^2_{glob} \leq \sigma^2_{loc}$

Since in our case, the noise is uniform throughout the image, noise variance can be determined as a constant for all pixels. This value could be determined automatically by running a global noise kernel throughout the image, and taking the most common lowest variance. At first, we have also ran a smaller noise kernel in parallel with the local kernel, to determine the noise for each pixel separately. The results were similar, but changing it to a constant improved the restoration quality.

We also exchanged the variance to deviation, to get a higher level of blurring. The final parameters used for the algorithm were 21 for the local kernel size, and 7.5 for σ_{glob}. The result can be seen on Fig. 10

Approach 2: Gaussian filter:

Since it is plausible to assume that the image suffers from Gaussian noise, a Gaussian filter could be a generally fast, easy and valuable improvement. The resulting image presented is from a kernel size of 9.

Approach 3: Median filter:

Due to the fact that the histogram contains a lot of max intensity pixels, a median filter could be worth considering. In this way, those pixels might benefit from being replaced by some of their neighbors, and would it generally serve as a way to average out the noise. The resulting image presented is from a kernel size of 9.

Approach 4: Non-Local Means Denoising:

OpenCV comes by default comes with a filtering algorithm that is useful for noises with similar distributions to this. It is similar to the Adaptive Noise Reduction filter in that it uses larger kernels to gain more information about the noise, but is otherwise a lot more complex [3]. The resulting image was produced with a filter strength of 10.

4.3 Results and discussion

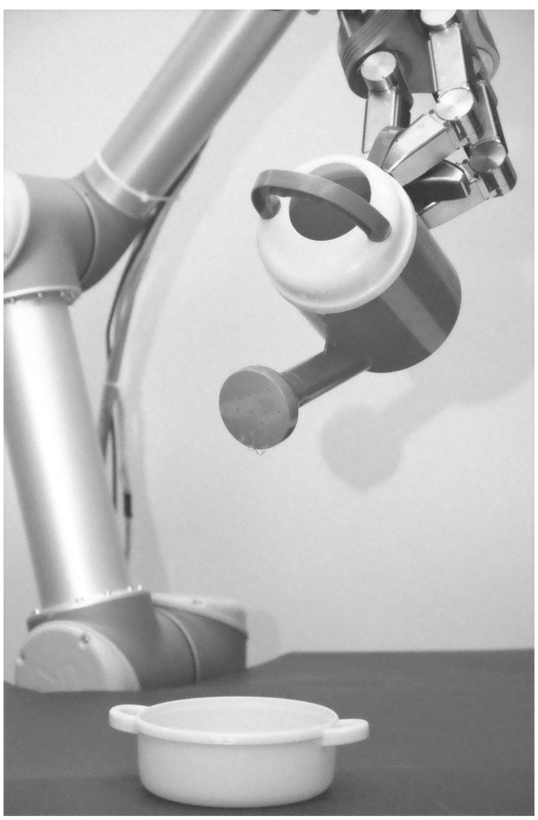


Fig. 10. resulting image of the Adaptive Noise Reduction filter, with $\sigma_g lob = 7.5$, and local kernel size of 21

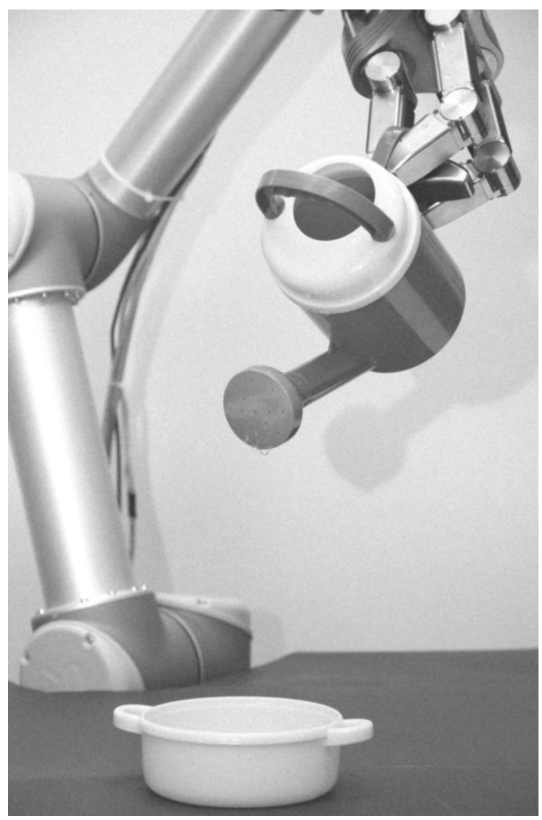


Fig. 11. resulting image from Gaussian filter with kernel size of 9

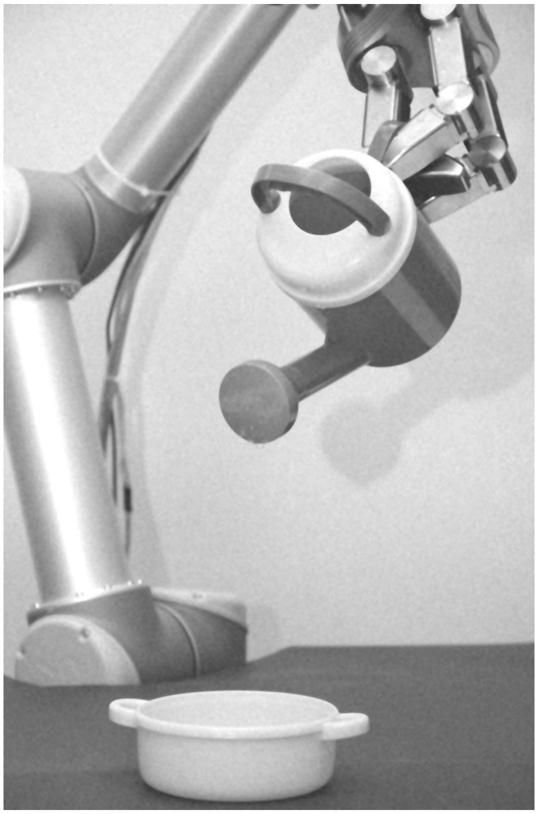


Fig. 12. resulting image from Median filter with kernel size of 9

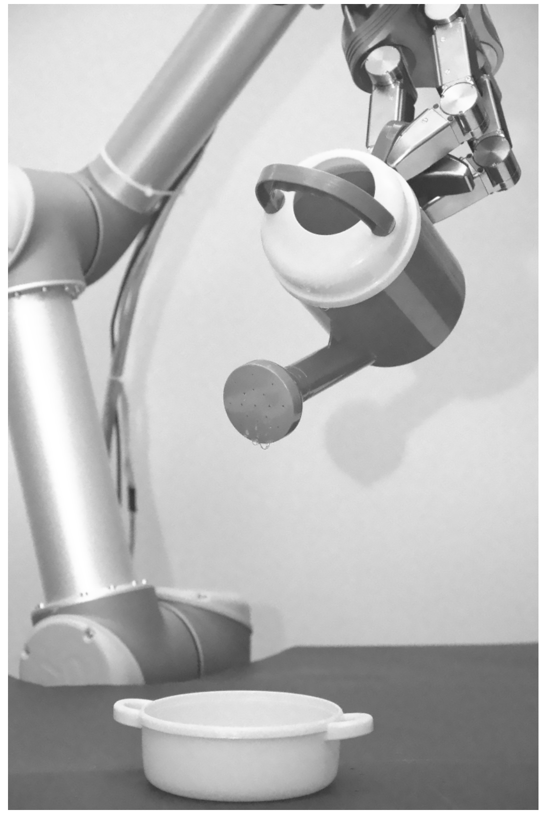


Fig. 13. resulting image form the Non-Local Means Denoising filter, with a filter strength of 10

After presenting all the individual filters, the results can now be compared. The Gaussian (Fig. 11) and the median (Fig. 12) filters produced similar results, with the Gaussian filter being more life-like, as expected. The noise was significantly reduced in both cases.

Our implemented Adaptive Noise Reduction filter (Fig 10) also produced a significant reduction in noise, with the added benefit of preserving the contours, resulting in a crisper image. The differences can be more easily observed by scaling up the images. The handle of the watering can, and the tool at the tip of the robot manipulator are much more authentic from a close-up.

The Non-Local Means Denoising algorithm (Fig. 13) produced the best result by far with even clearer edges and bigger noise suppression. This can be attributed to the complexity of the algorithm.

5 Image 4

5.1 Detected Flaws

The issue with this image can be easily suspected from visual inspection. There is a very clear and periodic distortion overlaid upon the original image (or what we would cognitively expect from a scene as such). Zooming in, diagonal waves can be observed that span from the bottom-left towards the top-right (Fig 14). Zooming in even further reveals diagonal waves, this time spanning from the bottom-right towards the top-left (Fig 15). These two distortions can be interpreted as sinusoidal wavefronts that are propagating orthogonally in respect to the previously mentioned lines. Therefore, we should expect 2 outlying peaks inside of the frequency domain of the image.

The histogram decomposition of the image resulted in a histogram that appeared balanced, and so it can be assumed that the intensities have otherwise not been tempered with apart from this periodic distortion.

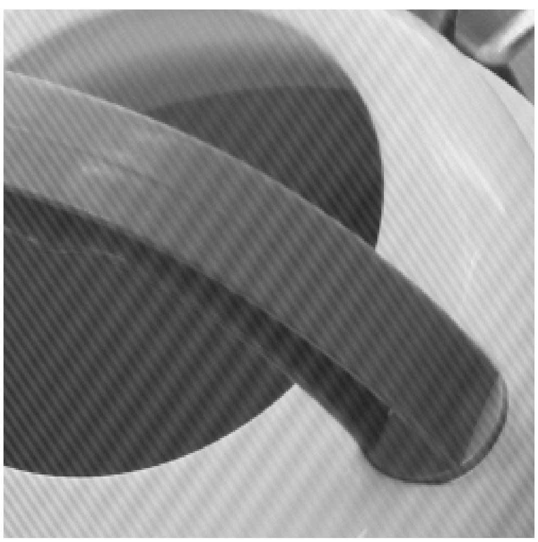


Fig. 14. a scaled region of the original image, showing diagonal lines

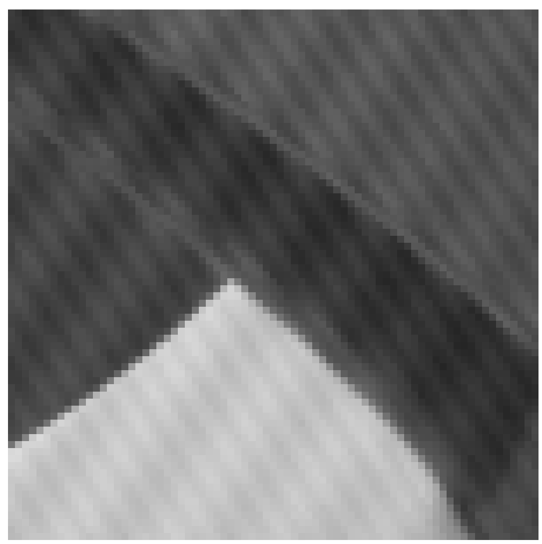


Fig. 15. a scaled region of the original image with an even higher zoom, showing diagonal lines in a different direction

After we have determined that the issue should be easy to interpret inside of the frequency domain, we have performed Discrete Fourier Transformation (and a quadrant shift) on the image. The result can be seen on Fig. 16. The result is what was expected, and the 2 white peaks are clearly visible (marked with a red highlight for clarity). Of course, since Fourier Transforms are symmetric for the positive and negative range (similar to an even function), both peaks show up twice, but the left half of the image contains no extra information.

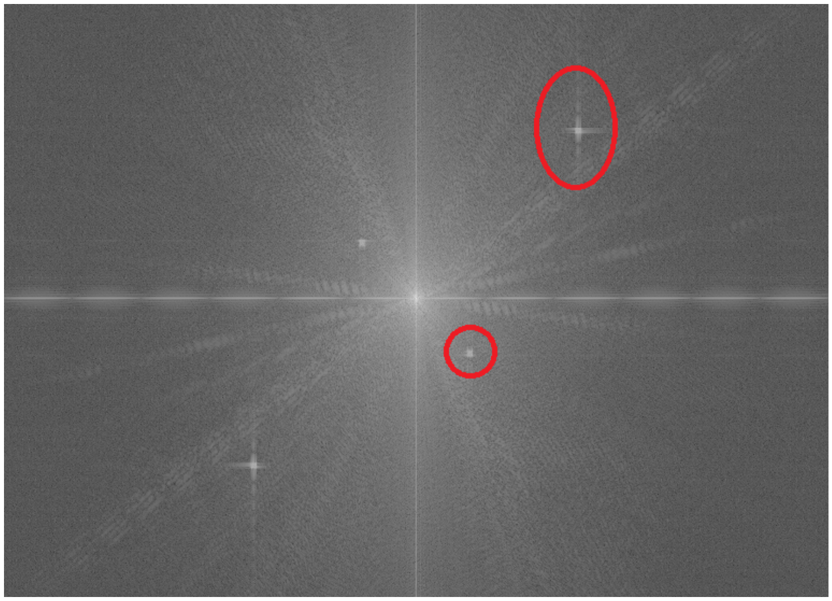


Fig. 16. the magnitude from the DFT transform of the distortion ridden image, with manually added red highlights

5.2 Restoration algorithm pipeline

There are many filters that can be used in the frequency domain, mostly focused on magnitude. A Notch filter, is a type of filter that can be used for lowering values in a specified frequency band (frequency vales are in pixels here) and direction range (since images are 2D). Therefore, a Notch filter should be an ideal tool for removing the previously discussed peaks.

In it's simplest form the notch filter could draw constant 0 values in a circle or a rectangle. However, in order to avoid the artifacts resulting from such fast changes in the magnitude, we have decided to implement equation of the Butterworth filter into the notch filter. The Butterworth filter has the following equation for images:

$$f(u,v) = \frac{1}{1 + [\frac{\sqrt{u^2+v^2}}{D_0}]^{2n}} \tag{2}$$

where u and v are pixel coordinates, D_0 is the kernel size, and n is the order. Increasing the order results in faster decline, converging towards drawing a solid circle of zeros.

By replacing u and v with $u - u_0$ and $v - v_0$ respectively in eq. 2, an offset can be applied and the equation becomes usable as a notch filter. Applying this filter requires us to know, or either somehow automatically determined the size and position of the peaks. Nonetheless, we have used this filter in our approach.

5.3 Results and discussion

After determining the peak positions and experimenting with the parameters of the Butterworth filter, and finally applying the filter to the magnitude of the image as a multiplication, we have arrived at the magnitude image seen on Fig. 17.

For the peak closer to the center we applied a lower order "n" so as to avoid sudden changes in the more sensitive lower frequency region. The peak was also smaller in size, so the used D_0 could also be lowered. The opposite applies to the higher frequency peak.

The result of the inverse DFT of the filtered magnitude, and unchanged phase, can be seen on Fig 18. The periodic distortions are virtually gone, and the loss in data arguably does not introduce significant artifacts.

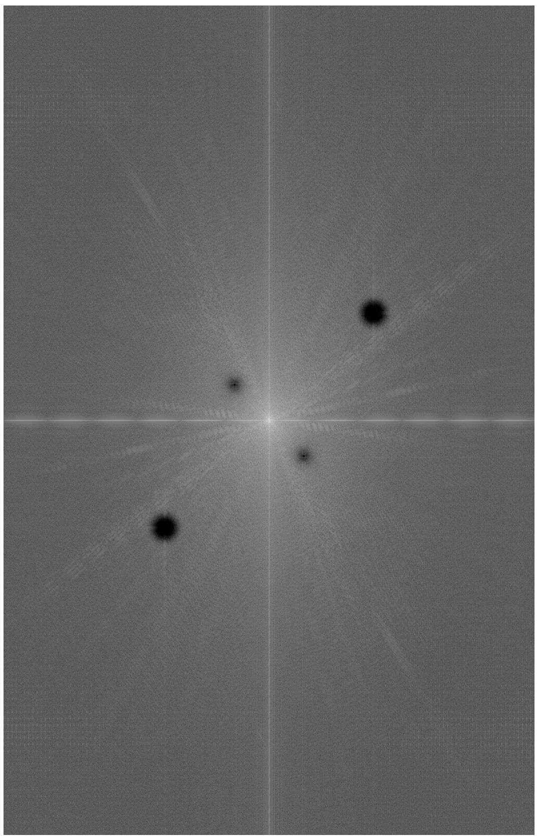


Fig. 17. the new magnitude of the restored frequency domain data

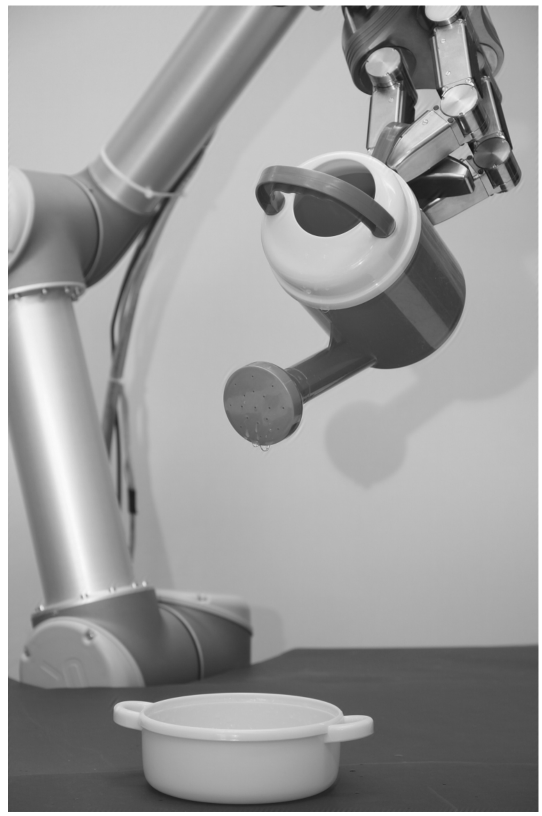


Fig. 18. the final result of the image restoration

References

1. Rovi1 lecture 5 slides. image restoration.
2. M. D. Abràmoff, P. J. Magalhães, and S. J. Ram. Image processing with imagej. *Biophotonics international*, 11(7):36–42, 2004.
3. A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.