Real-time precipitation suppression in video streams

Hoa Truong
ada09htr@student.lu.se

Adam Nilsson
ada09an1@student.lu.se

June 24, 2014

Master's thesis work carried out at Axis Communications AB.

Supervisor: Kalle Åström, kalle@maths.lth.se

Examiner: Jacek Malec, Jacek.Malec@cs.lth.se

Abstract

In surveillance cameras rain and snow can introduce an unwelcome noise to the video stream. The resulting effect of the rain becomes bright streaks in the frames of the video. These streaks can disturb human viewers and image processing algorithms.

Rain streaks can be hard to detect and remove as they are a very dynamic phenomenon dependent on camera settings and weather conditions. This thesis aims to research some already invented rain removal algorithms and compare and evaluate them. Surveillance cameras supply video in real-time so it is not possible to access the whole video and perform heavy computations relying on information from the future.

Qualities such as level of streak suppression and time required to perform necessary calculations are weighed against each other.

Keywords: rain-removal, image enhancement, rain-detection

Acknowledgements

We would like to thank Axis Communications AB for allowing us to perform this master thesis in cooperation with them. We would also like to thank our examiner Jacek Malec and supervisors Kalle Åström, Mazdak Farzone and Dennis Nilsson for their valuable feedback on our work progress.

Contents

1 Introduction **7**
- 1.1 Background . . . 7
- 1.2 Goal . . . 7
- 1.3 Problem formulation . . . 8
  - 1.3.1 Challenges . . . 8
  - 1.3.2 Constraints . . . 9
- 1.4 Previous work . . . 9
- 1.5 Work division . . . 10
- 1.6 Report structure . . . 11

2 Theory **13**
- 2.1 Rain model & the frequency domain . . . 13
- 2.2 Skewness . . . 18
- 2.3 Intensity . . . 19
- 2.4 Chromatic properties . . . 21
- 2.5 Morphological properties . . . 21

3 Method **23**
- 3.1 Tools . . . 23
- 3.2 Detection algorithms . . . 24
  - 3.2.1 Frequency . . . 24
  - 3.2.2 Skewness . . . 25
  - 3.2.3 Intensity . . . 26
  - 3.2.4 Chromatic . . . 28
  - 3.2.5 Morphological . . . 29
  - 3.2.6 Blurring . . . 29
- 3.3 Restoration Algorithms . . . 30
  - 3.3.1 Median . . . 30
  - 3.3.2 Mean . . . 31
  - 3.3.3 Subtraction . . . 31

3.4 Data gathering . . . 31
3.4.1 Images . . . 31
3.4.2 Video . . . 32
3.5 Priorities . . . 33
3.6 Example . . . 34

4 Results & evaluation **37**
4.1 Time measurement . . . 37
4.2 Evaluation of Detection Algorithms . . . 38
4.2.1 Frequency . . . 40
4.2.2 Skewness . . . 41
4.2.3 Intensity . . . 41
4.2.4 Chromatic . . . 45
4.2.5 Morphological . . . 48
4.3 Evaluation of Restoration Algorithms . . . 50
4.3.1 False detection . . . 51
4.3.2 Comparison metrics . . . 52
4.4 Algorithm results . . . 53
4.4.1 Detection Algorithms . . . 53
4.4.2 Restoration algorithms . . . 56

5 Discussion & conclusions **59**
5.1 Detection quality . . . 59
5.2 Real-time constraint . . . 60
5.2.1 Detection frame - color or gray scale . . . 61
5.2.2 Reducing false detections . . . 62
5.3 Restoration quality . . . 62
5.3.1 Limitations with taking the mean as the correct value . . . 62
5.4 Conclusions . . . 63
5.5 Improvements & Future work . . . 63

Bibliography **65**

Chapter 1
Introduction

1.1 Background

Axis communications AB is a company which develops surveillance cameras. These cameras are used in a range of circumstances, some of which are outdoors. The cameras are used to supply video streams of scenes such as parking lots, roads or restricted areas.

A video camera which captures video in a rainy environment will produce artifacts in the resulting frames in form of streaks.

The raindrop refracts and reflects light from its surroundings. This make them under some circumstances very visible to an observer. The resulting effect on the image appear discontinuous and places constraints on the algorithms used to detect and remove the streaks. The properties of rain may vary heavily from storm to storm. Factors such as wind, alter the direction of the streaks. In the case of snow the wind plays a bigger role as snow particles are lighter and may flow around in the scene with less clear of a direction than rain would.

Precipitation is caused by the weather and have most impact on outdoor scenes. Therefore this thesis will mainly treat outdoor scenes.

1.2 Goal

The thesis is done in the context of surveillance cameras and real-time modification of the frames from a camera recording a scene with rain- or snowfall.

The streaks produced by rain or snow can be disturbing for both human observers and image analysis algorithms which might be operating on the video stream from the surveillance camera. A surveillance camera does not want its motion detection to trigger on different weather phenomena. The streaks may also cause failure in face recognition algorithms when raindrops occlude parts of the face.

If a solution should be viable in a surveillance camera context the algorithm removing the streaks should only utilize the images from the past and the present. Time complexity of the work performed becomes another important factor as well as the quality of the processed images.

Before starting the thesis it was concluded from limited research that complete removal of rain is very challenging. The aim was to implement and evaluate some existing algorithms. If possible, combinations of algorithms would be tested to improve the results. The resulting algorithm should be fitted for Axis cameras and be useful in real-time and use no information unavailable to a typical surveillance camera.

Apart from implementing deprecipitation for video streams, the computational time required for the different algorithms are also calculated and compared. Furthermore, the results of the deprecipitation algorithms are also evaluated. A method to evaluate the detection and replacement algorithms are also proposed and used.

1.3 Problem formulation

To remove/suppress the streaks from a video three things will have to be ascertained about the frames in the video stream.

1. Are there any rain streaks in the frame?
2. Which pixels are occupied by rain streaks?
3. What will the replacement value for these pixels be?

The first item in the list is included as a step if one would like to progress differently depending on the accuracy of the rain detection. It may be solved, or partly solved with item two. The second and third steps are two separate things which both have great effect on the result.

Accurate rain detection is important in order to preserve image quality. Too many false detections will lead to non-rain pixels being detected as rain and erroneously replaced. Depending on the replacement algorithm the consequences of this could either be that nothing happens or new artifacts being introduced to the images. Too many missed detections will lead to less rain or no rain being removed.

When the pixels containing rain are detected, their current value will be replaced another value disguising the rain. Ideally the observer should not suspect that there had ever been a rain streak at all at the pixels which previously were covered by rain. Methods to determine what these replacement values are are commonly referred to as inpainting.

Even though rain and snow are not the same, in some cases they can be treated equally with the same rain removing algorithm, and still be removed or suppressed.

1.3.1 Challenges

There are many specific challenges to overcome when removing precipitation from video. Some are specific to the algorithm being used. Other are from the characteristics of the video, such as brightness or colors.

The camera which films the scene may be static or moving. A moving camera makes the whole scene move and all the intensities of all the pixels will change from frame to frame. In the surveillance camera context this can be more or less critical as there are cameras which pan over an area and those who are fixed. If the camera would be moving then image stabilization can be used in order to make the scene more stable.

A scene can contain moving objects apart from the raindrops. Without this property the task of removing rain becomes almost trivial as it reduces the problem to removing all movement in the video. Therefore it is important to be able to differentiate between rain and moving objects such as people or vehicles.

A scene's lighting can have a large effect on how the streaks get imaged in a frame. A light source near the camera causes the nearby streaks to become a lot brighter, as they reflect the light.

Rain or snow comes in all shapes and sizes and the distance from the particle to the camera varies. Weather as a whole is dynamic and its effect is often seemingly erratic.

Rain streaks are not opaque and have different levels of transparency. Algorithms have to cope with this and decide whether or not a pixel warrants being assigned as part of a raindrop. Raindrops out of focus are often almost invisible and therefore it is not necessary to treat them.

1.3.2 Constraints

The thesis primarily focuses on stationary surveillance cameras. This constraint was added halfway through the thesis, and was a result of some algorithms struggling with the problem that arises when all the pixel values in the image changes at the same time. Another factor that played a part in this decision was that the majority of the surveillance cameras are stationary.

1.4 Previous work

In [2] the properties of rain and snow in the frequency plane is analysed. It is found that the visual effect of rain in an image is consistent in the Fourier transform of the image. This can later be used to create a model of rain and snow which is used to remove frequencies similar to the rain or snow frequencies in an image. The model uses several parameters such as exposure time, focal length, droplet size, brightness and orientation of the rain. All these parameters need to be adjusted for the video and rain at hand, so that the model can be accurate. To improve the result, the temporal aspect of the video is utilized and pixels which are both rain like in the frequency space and are changing rapidly in the time domain are marked as rain. After the rain is detected, an alpha blending between the image containing rain and an estimated rain-free image is made in order to remove the rain. This rain-free estimate is made by using a median filter on the same image.

A camera has many different parameters and many of them can be adjusted. In [5] it is proposed that these parameters can be set so that the rain effects are minimized or removed. Exposure time and field of vision, among others, are used to lower the effects of rain.

The chromatic and temporal properties of rain are investigated in [4]. The temporal property states that a pixel is not always covered by rain throughout the entire video. With this fact in mind a histogram of pixel intensity is created for every pixel and two intensity peaks are identified; the intensity peak for the background and the intensity peak for the rain. It is stated that for this to work the camera has to be stationary or video stabilization has to be performed. The chromatic property says that the change of the red, green and blue values of a pixel are roughly the same from frame to frame if it was covered by a raindrop.

In [3], the typical shape of rain is used to find regions similar to raindrops. The approach uses few frames to achieve rain removal. First the intensity of three consecutive frames are examined and used, in order to get some initial estimations of regions containing rain pixels. These regions' sizes and aspect ratios are then examined and those which do not fit the characteristics of a raindrop are discarded.

Sajitha Krishnan and D.Venkataraman [7] use the symmetric temporal property of rain in order to discern rain pixels from other pixels. The symmetry comes in form of the intensity changes of a pixel when rain passes over it. Compared to other motion, an intensity change due to rain is more symmetric across multiple frames than if it was caused by movement of other objects. The intensity of a pixel will be brighter when the rain is falling over it and darker when the rain has passed it. This rising of intensity, followed by falling of intensity is more symmetric for rain. This is used to distinguish pixels containing rain from pixels that do not contain rain. Since the skewness is a measure of asymmetry, the lower the skewness are for a set of pixels, the higher the probability that these pixels are affected by rain.

1.5 Work division

At some stages the work behind this thesis has been done together, at other the work was divided.

Roughly the division of the work was the following.

Adam:

- Implemented frequency, chromatic and morphological detection.
- Implemented a program that enabled the modification of streaming videos.
- Found Axis-cameras that could give good rain videos.
- Evaluated precipitation detection.
- Wrote the theory and method section in the report.

Hoa:

- Implemented skewness and intensity detection.
- Implemented mean, median and subtraction restorations.
- Implemented blurring.

- Evaluated precipitation restoration.
- Implemented color neutralization as a way to remove false detection.
- Wrote the result and conclusion section in the report.

The rest of the work was carried out together. Some examples are pair programming, code evaluation, reading and correcting the report, discussion of algorithm development and so on.

1.6 Report structure

The report is structured in the following way. Chapter one gives a brief introduction and background to the problem formulation. Chapter two explains the theory behind this thesis. Chapter three explains the methods used, data gathering and what is important to consider when removing precipitation from videos. Chapter four presents the results, and the evaluation. Lastly the conclusions of this work and its discussions are presented in the fifth chapter.

Chapter 2
Theory

This chapter summarizes the theory about the properties of rain and snow, which has been used in the algorithms in the previous research and then used in the algorithms in this work. It also describes the problem formulation in further detail.

2.1 Rain model & the frequency domain

Barnum and Narasimhan [2] build a model of rain appearance , when captured by a camera. This model model utilize mathematical properties in order the describe the concept of the rain streaks in images or videos. The model is then transferred into the frequency domain with a two-dimensional discrete Fourier transform and used to predict rain frequencies.

The model of a single rain streak is referred to as the streak model. The model of rain consisting of many rain streaks in the frequency plane of the whole video sequence is referred to as the rain model.

Streak model

The rain model in [2] approximates the rain streaks with a motion-blurred Gaussian and utilizes several approximations of rain characteristics. The streaks of rain are imaged when the camera captures light from a falling raindrop were its speed is an important factor. Research has shown that a falling raindrop with the diameter a has a velocity v were

$$v(a) = -0.2 + 5.0a - 0.9a^2 + 0.1a^3. \tag{2.1}$$

Raindrops are not always spherical as shown in [6]. They become more deformed the larger they get. This is not taken into account in this model because its impact would have been negligible with respect to the simplicity of the model. For their model they use rain and snow particles of sizes between $0.1 - 3mm$ while real particles can range up to $10mm$

according to [6]. The simplified value seems to serve them well for the model and are in line with the assumed average size of $1mm$ in [5].

The typical rain streak in a video appear brighter than the background (discussed further in 2.3) and is blurred along the edges. The model aims to catch this property and avoids complex lighting properties such as light passing through and being refracted in the raindrops. A streak's breadth b is considered to be relative to the cameras focal length f and the distance z from the camera to the raindrop as

$$b(a,z) = a\frac{f}{z}. \tag{2.2}$$

The length l of the streak of the model depends on the exposure time e of the camera according to

$$l(a,z) = (a + v(a) \cdot e)\frac{f}{z}. \tag{2.3}$$

This reflects how far the raindrop falls during each intake of light for the image sensor of the camera.

With these three equations a simple rain streak is defined. A simple illustration of a raindrop falling can be seen in figure 2.1.

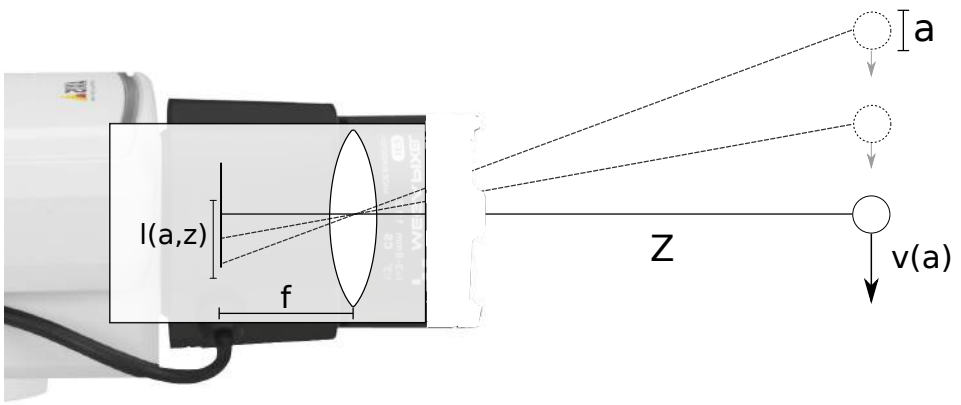

Figure 2.1: A streak is imaged when a raindrop falls. The faster it falls the longer it is stretched out over the sensor.

Two additional parameters are added to control the streak's position. The entire purpose of the model is to get a rain like effect which can later be used to find the appropriate frequencies of the rain. The individual raindrop is represented as a Gaussian g as shown in equation 2.2. As wind blows the raindrops change direction which needs to be reflected in their continued model. Due to the change of the rain streak direction, the orientation θ is introduced and the position of the streak is given as $\mu = [\mu_x \mu_y]$

$$g(x,y,a,z,\theta,\mu) = \int_0^{l(a,z)} exp(-\frac{(x - cos(\theta)\gamma - \mu_x)^2 + (y - sin(\theta)\gamma - \mu_y)^2}{b(a,z)^2})d\gamma. \tag{2.4}$$

Integrating over the length of the streak gives an image with coordinates x and y. This streak can liken many real streaks but it is a simplified version so it cannot change its

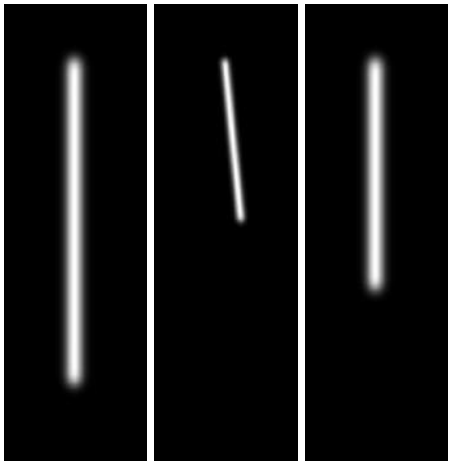

Figure 2.2: Gaussian streaks created by the streak model in [2]. The leftmost image is the basic streak and in the middle a streak with increased θ which is further from the camera can be seen. To the right a streak with a shorter exposure time can be seen.

characteristics in form of i.e. being curved and it is symmetrical where real streaks may not be. Three examples of streaks can be seen in figure 2.2.

With equation 2.4 the initial part of the model is finished. Barnum and Narasimhan's premise is that the chaotic behaviour of rain in videos is more well behaved to an extent in frequency space which can be seen in figure 2.3.

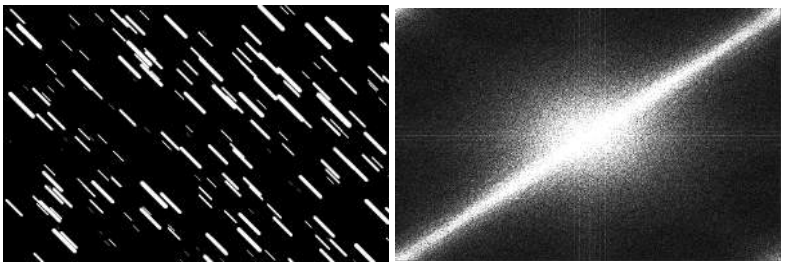

Figure 2.3: To the left there is an image with simulated rain. To the right there is the magnitude of the corresponding discrete two dimensinal Fourier transform. Notice that the Fourier transform of the rain is a belt perpendicular to the actual rain.

Rain model

To get a perfect representation in frequency space one would need to apply the Fourier transform to all the streaks in a scene. This can be done by using

$$\left\| \mathcal{F} \left\{ \sum_{d=1}^{N} g(x, y, a_d, z_d, \theta_d, \mu_d) \right\} \right\|. \tag{2.5}$$

The sum represents the addition of all streaks in the image. This can be simplified to the case where the Fourier transform is done for each streak individually,

$$\left\| \sum_{d=1}^{N} G(x, y, a_d, z_d, \theta_d, \mu_d) \right\|. \tag{2.6}$$

Now a model of the streaks are achieved but it is not yet useful as the correct number of streaks in the image and the correct parameters for all the streaks need to be identified. Barnum and Narasimhan studied the behaviour of rain in frequency space and make three key observations of the appearance of rain in frequency space:

Observation 1: The shape of the magnitude does not depend strongly on streaks' locations in an image.

Observation 2: The shape of the magnitude is similar for different number of streaks.

Observation 3: The magnitude is approximately constant across the temporal frequencies.

These observations eliminate the need to know both the number of streaks in an image and where the streaks are located, to be able to create a rain representation in frequency space. Equation 2.4 describes how one rain streak may look, after the removal of spatial parameter μ equation 2.6 becomes

$$\left\| \sum_{d=1}^{N} G(x, y, a_d, z_d, \theta_d) \right\|. \tag{2.7}$$

The problem of the sum persists since the different raindrops in the scene are different and contain many frequencies. To remove the sum Barnum and Narasimhan uses statistical analysis of rain to construct the final model. It is enough for the model to capture enough frequencies accurately to be useful for the purpose of rain removal. Instead of using the sum of all streaks they use three assumptions to simplify the formula.

Assumption 1: The probability of finding a raindrop along the depth of the scene is equally distributed.

Assumption 2: A streak has equal chance of having an orientation of anywhere between θ_{min} and θ_{max}.

Assumption 3: A particle has equal chance of being of any size between $a_{min} = 0.1mm$ and $a_{max} = 3mm$.

With these assumptions about the parameters they construct the model to contain frequencies in proportion to the mean streak and introduce a scalar Λ to scale to a correct brightness. The z^2 in the following formula come from the first assumption since the volume imaged at a depth is relative to the depth squared. The further away a raindrop is imaged the more weight it is given.

$$R^*(u, v, \Lambda, \theta_{min}, \theta_{max}) = \Lambda \int_{\theta_{min}}^{\theta_{max}} \int_{a_{min}}^{a_{max}} \int_{z_{min}}^{z_{max}} z^2 \|G(u, v, a, z, \theta)\| dz da d\theta \tag{2.8}$$

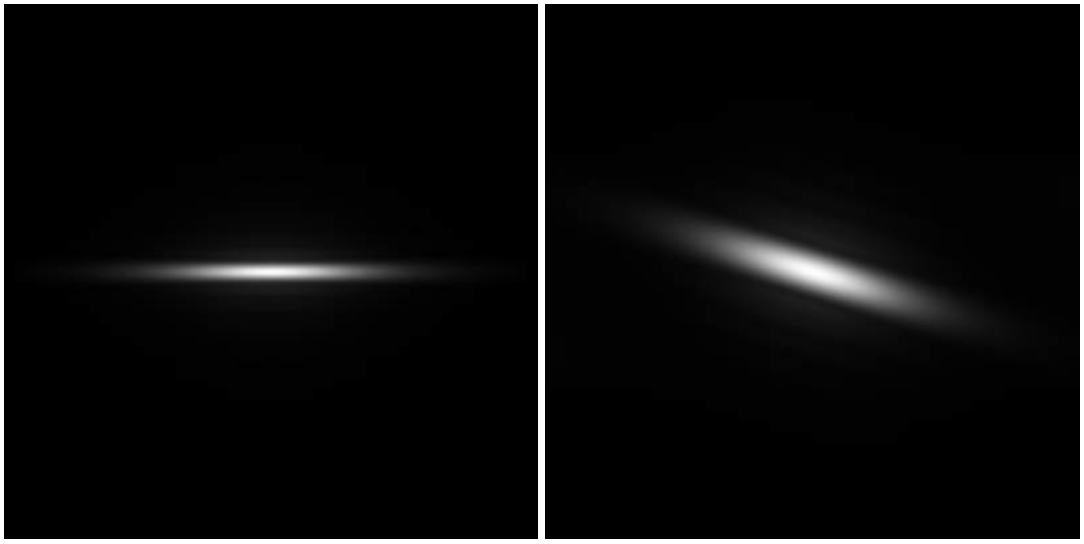

Figure 2.4: The rain model of frequencies for rainfall, to left a model of rain falling straight down and to the right where the rain is slightly angled. As the model is simplified, less frequencies compared to figure 2.3 can be seen. Also the frequencies are more focused in a belt.

And now it can be seen that the model no longer depends on the individual streaks in the image in any way. The integrals does not need to be computed exactly and an approximation by sampling across θ, a and z can be used.

The model can be extended with the third observation, which states that the contribution of rain to frequency along the temporal dimension are roughly the same.

$$R^*(u, v, w, \Lambda, \theta_{min}, \theta_{max}) \approx R^*(u, v, \Lambda, \theta_{min}, \theta_{max}) \tag{2.9}$$

Fitting model to video

The input parameters θ and Λ need to be fitted to the particular movie being processed. Barnum and Narasimhan reasons that because of the fact that most objects in a scene are contributing mainly to the lower frequencies, therefor a frequency chosen at random is likely to be large part rain. This property is amplified if only the temporal frequencies are considered, which represents everything that moves in the scene. Λ is suggested to be estimated once per frame. To estimate the temporal neighbourhood of frames are needed and the following ratio becomes an approximate of the brightness, where $M(u, v, w)$ is a Fourier transformed frames and R^* is the model with Λ equal to 1. The ratio is defined as

$$\Lambda \approx \frac{median(\|M(u, v, w)\|)}{median(R^*(u, v, w, 1, \theta_{min}, \theta_{max}))}. \tag{2.10}$$

In the above equation 2.10 the current temporal frequency are disregarded. The current temporal frequency is the current image and it equals to $w = 0$. The previous image and temporal frequency is denoted as $w = -1$ and the next image and temporal frequency is denoted as $w = 1$.

To come up with a good value for the orientation of the streaks continuous frames needs to be analyzed. It is not necessary to update the range of orientation every frame since the orientation does not change that quickly. Barnum and Narasimhan also simplify the range to a single value for rain since it generally fall in the same direction while snow is more chaotic.

Using the standard deviation across time as an estimate of important frequencies

$$\tilde{R}(u,v) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\|M(u,v,t)\| - \|\bar{M}(u,v)\|)^2} \tag{2.11}$$

an appropriate θ can then be found by minimizing the difference between the estimate and the model,

$$\arg\min_{\theta} \int\int (\|R^*(a,v,\Lambda,\theta_{min},\theta_{max})\| - \tilde{R}(u,v))^2 dvdu. \tag{2.12}$$

Detection using model

The next step is to use the model to actually detect the rain in the scene. The frequencies in the model R^* are meant to represent rain. The real frequencies of the input image is in the Fourier transformed image $M(u,v)$. The ratio $R^*/M(u,v)$ will give the amount of frequency being rain. The spatial information of the raindrops in the real image are stored in the phase of the image's Fourier transform. This can be used to position the rain frequencies.

A rain removed image back in the spatial domain can be achieved by calculating p_2 according to

$$p_2(x,y,t) = \mathcal{F}^{-1}\left\{\frac{R^*(u,v,\Lambda,\theta_{min},\theta_{max})}{\|M(u,v)\|} \times exp(i\phi M(u,v))\right\}. \tag{2.13}$$

If $R* > M(u,v)$, for a frequency, the model have predicted more rain than there actually are rain and background objects combined. This means the model is wrong for that frequency and Barnum and Narasimhan cap the ratio at one, saying that rain is contributing everything at that frequency.

Barnum and Narasimhan sets out to correct false detections by using information from more than one frame. They perform a three-dimensional Fourier transform on equation 2.13 and use that in a ratio the same way they did in equation 2.13. The reasoning behind equation 2.9 provides justification to the comparison of the model to the three-dimensional transform.

2.2 Skewness

In the temporal domain, pixels affected by rain will exhibit a symmetrical change of intensities. Krishnan and Venkataraman, [7], finds that pixels where rain is responsible for movement in a picture the intensity values above and below the mean are more predictable than for other movement. This symmetry around the mean value of the pixel intensity is represented by its skewness.

The method does not make any assumptions about direction of the rain or any other characteristics rain may have and focuses on the symmetric change of intensity of pixels. The skewness of a sample is defined as

$$skew(x_1, ..., x_n) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}}{s} \right)^3 \tag{2.14}$$

where $\bar{x}$ is the mean and s_x is the standard deviation of the sample.

The closer the value of skew is to zero the more symmetric the input are. The skewness function can output three different kinds of values, negative, zero or positive. A negative skew value states that a majority of the values are larger than the mean and a positive skew states that the majority of the values are less than the mean as is demonstrated in figure 2.5.

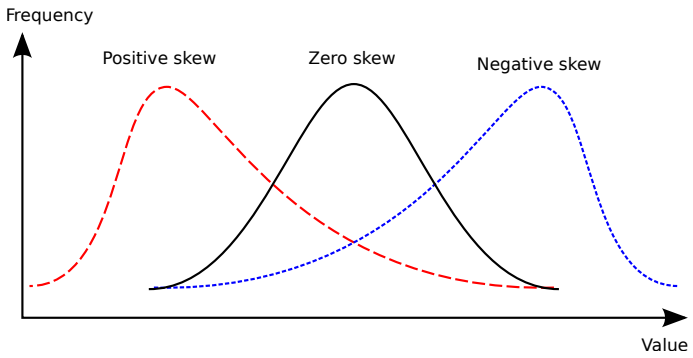

Figure 2.5: Three distributions with different skewnesses. The skewness is a value which shows where the values lie in relation to the mean of the values.

The skewness value of some consecutive frames are calculated and detection of a rain pixel is achieved through introducing a threshold. Pixels with a skewness value beneath the threshold are identified as containing rain.

A major aspect to notice about this method is that stationary objects will be detected as well as rain. This is a clear disadvantage but if the algorithm can distinguish moving objects from everything else it could be a good complement-algorithm. To detect moving objects in a scene excluding rain solves many of the tricky parts of the problem. Depending on the restoration algorithm it may not even matter if stationary objects are detected as rain.

2.3 Intensity

The effect of rain on a pixel, as stated in the skewness chapter, is that it makes the pixel brighter. This is explored in [6]. A raindrop acts as a lens refracting light from a wide range of the background. Garg and Nayar performed experiment with drops and concluded that a raindrop refracts light from a 165 ° angle behind it as seen in figure 2.7, seen from the camera.

If compared to other noise, raindrops are much more dependent on the background from which they gain their intensity. They also find that if the background behind the

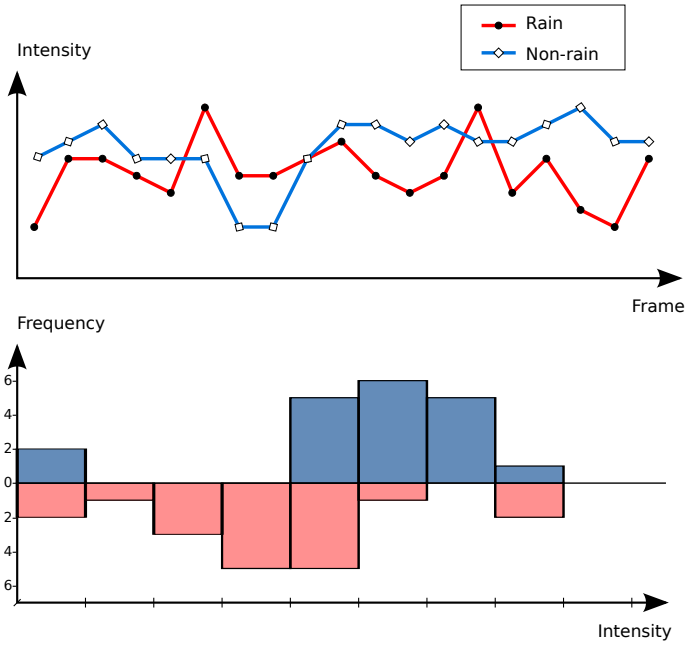

Figure 2.6: In the upper graph an example of two pixel intensity values over 19 frames can be seen. In the lower graph the distribution of the values of these two graphs are shown. The example pixel being covered by rain has a distribution of intensities that are more symmetrical than its counterpart. The rain pixel has a skewness of 0.287 while the non-rain has a skewness of -1.51.

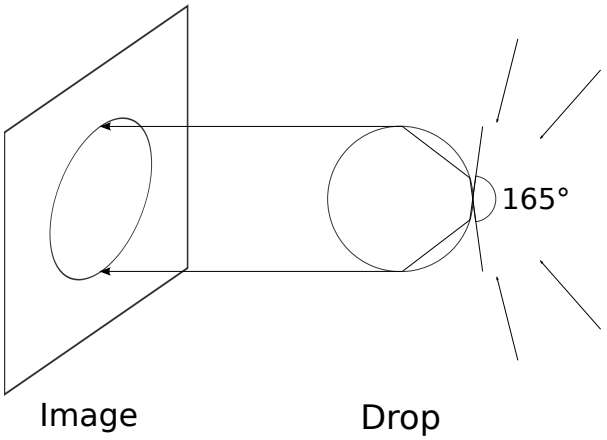

Figure 2.7: A raindrop refracting light from the background. The wide angle of ingoing light is what makes the rain streaks in the videos become bright.

raindrop is very bright the change is relatively less than if a raindrop is in front of a darker background.

The result is that a pixel in an image exhibits a short intensity spike when a raindrop passes over it, this can be seen in figure 2.8. It is short because of the speed of the raindrop; the magnitude depends on the scene. Variables that affect the behaviour of the intensity increase are exposure time and amounts of precipitation. In heavy rainfall a pixel can be

covered by rain streaks in multiple frames resulting in the spike lasting longer. Slower falling particles such as snow lingering long enough to occupy partially the same pixels in continuous frames were also observed.

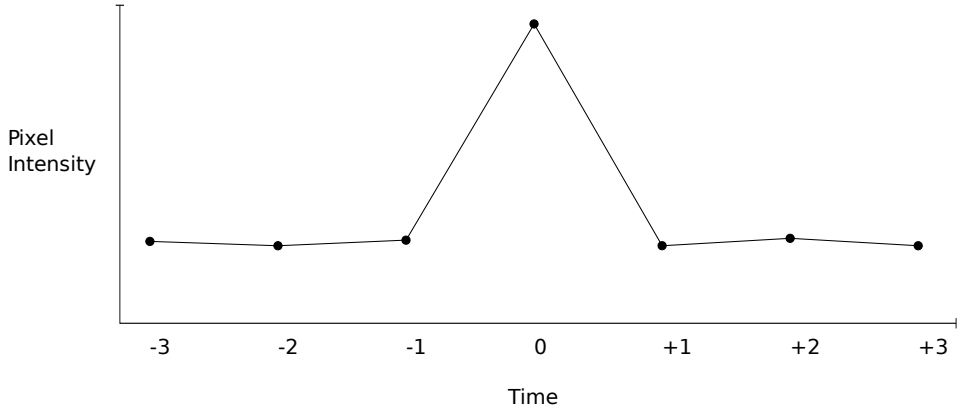

Figure 2.8: When a raindrop passes over a pixel a sharp spike in intensity is created. The height of this spike depends on the lighting in the scene.

2.4 Chromatic properties

Zhang et al [4] builds on [6] and investigate how light of different wavelengths behave in relation to raindrops. Different colored light have different refraction indices and therefore raindrops should refract different colors differently. In the end the fact that different colors of light refracts differently is not significant enough to be used in their rain detection. Their research leads to the revelation that raindrops cause an intensity change of a pixel, which is similar across the three color channels red, green and blue. For non-rain pixels the change of the intensities in the red, green and blue channels will not be the same. This is used in order to differentiate between rain movement and non-rain movement.

This means that a raindrop falling in front of a red background will be tinted red to a significant degree. Whilst another non-transparent moving object, such as a car, will reveal no color information of the pixels that are occluded by it.

A limitation is identified where an object is gray and moves. A gray object lacks color so the change across the channels when it moves can become rain-like.

2.5 Morphological properties

Brewer and Liu investigated the shape of rain in [3]. As a starting point they use intensity characteristics of rain to select probable rain pixels. Neighbouring marked pixels are treated as connected regions and their shape is analyzed. Equations 2.1, 2.2 and 2.3 are used as a model of the shape of a raindrop. The aspect ratio of a raindrop is

$$AR(a) = \frac{l(a,z)}{b(a,z)}, \tag{2.15}$$

which can be simplified as

$$AR(a) = \frac{v(a) \cdot e}{a} + 1. \tag{2.16}$$

The aspect ratio only depends on the diameter of the drop and the exposure time of the camera. The regions found by the intensity spike filter can be tested against a likely range of aspect ratios if the exposure time of the camera is known. This assumes a range of sizes for the rain particles. Brewer and Liu uses the same sizes as in section 2.1 which is between $0.1 - 3mm$.

The regions found can be decreased to ease the computation if one observes that very small regions will not contribute to the result meaningfully. A region which consists of too few pixels is removed on the basis that it is not large enough to be an obstructing raindrop and is likely a false detection.

To calculate the aspect ratios of the remaining regions one is faced with obstacles. A region is not a precise fit over a rain streak and the intensity detection gives deformed results and holes. To solve the issue with holes a morphological fill operation can be performed. The aspect ratio is found through taking the second central moment of the region and finding the corresponding ellipse to fit over the region. The aspect ratio is approximated as the quote of the major axis and the minor axis of the ellipse.

Brewer and Liu also further improves the result by proposing to calculate the orientation based on the tilt of the ellipse. Rain falls in approximately the same direction so this can be used to filter away the ellipses with a rotation which is far off the general mean ellipse in the frame. This only works when there are low amounts of non-rain motion in the scene.

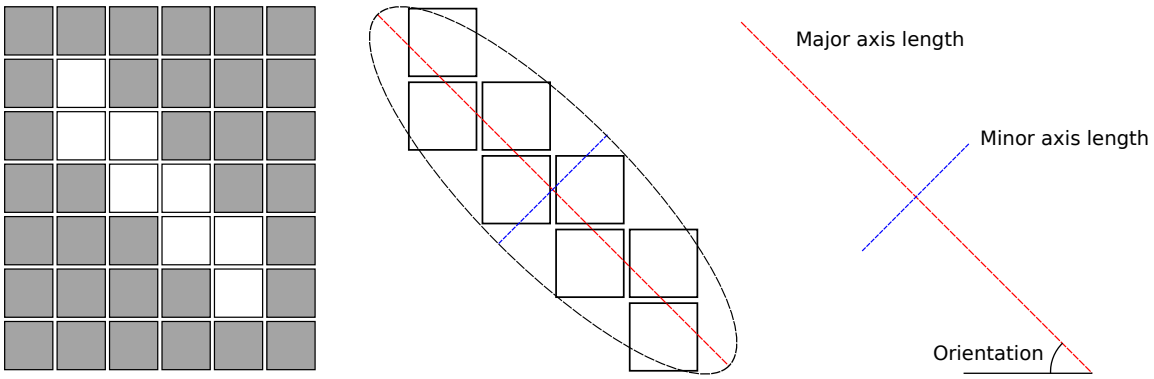

Figure 2.9: Ellipse approximation of a region of pixels marked as rain. To the left a detected region of connected pixels is shown. In the middle the corresponding ellipse approximation is outlined, from this approximation we get the information in the right graph. The characteristics of the regions are measured from this approximation and since there can be many regions to check this can be very time consuming.

Chapter 3
Method

This chapter will describe the working process employed during the thesis.

Previous research in the area of rain removal was studied first. Several papers discussed the topic, some more extensively than others. Barnum and Narasimhan's research about frequency space [2] showed promise so that was where the thesis started.

After an algorithm was familiarized with, a running implementation was developed. The development time for the different algorithms varied greatly.

During the development of the algorithms some evaluation occurred all the time as the end results could be seen directly and be analysed. The complete evaluation is presented in section 4 .

3.1 Tools

This section will provide a brief summary of the tools used during the thesis.

For the image analysis part, Octave was the main development tool used in this master's thesis. Octave is an open-source alternative to the similar proprietary software called Matlab, and shares many functions. Since Octave provides very efficient ways to handle matrices, Octave was the program of choice.

Matlab was also utilized for the morphological algorithms where Octave lacked some useful functions.

Java was used to load videos which could not be loaded in Octave. The Java implementation was closely intertwined with Octave and the software package JavaOctave was also used as a bridge between the two.

The camera used to record the evaluation video presented at section 4 is an AXIS P3367 Network Camera. It supports 5 megapixels or HDTV 1080p (2592x1944) quality. A lot of footage from other cameras supplied by Axis were also inspected.

3.2 Detection algorithms

Five algorithms where implemented and compared in order to find the most suitable algorithm.

They are presented below.

3.2.1 Frequency

The frequency domain algorithm was investigated first out of the five algorithms. Unlike some of the other algorithms, the process of removing rain with information regarding the representation of the frequency of rain consist of several steps.

Streak model: Uses theory to construct a rain streak similar to the streaks falling rain make in an image.

Rain model: A Fourier transform of the streak model created to imitate rainfall.

Fitting model to video: A big part and a big obstacle is the various parameters of the models which needs to be adjusted to fit a particular rainfall.

Detection Use the model to identify the frequencies of rain in the image and remove them.

Streak Model

Firstly the model was built with the formulas described in section 2.1. The streak model is handled as a matrix containing the values for the streak. The streak can easily be visualized as seen in figure 2.2.

The dimensions of the matrix, which contains the streak are the same as the intended matrix dimension of the input image.

Rain model

To get an approximation of the mean streak as in equation 2.8 a number of streaks were generated and added together. These streaks were created with parameters drawn from different distributions suggested by Barnum and Narasimhan [2]. This introduced random elements and was an implementation choice made for the sake of simplicity. The model then corresponds to the band of frequencies where rain is likely to contribute, see figure 2.4.

Fitting model to the video

In order to get an accurate model that can be used to remove rain from videos the model needs to be fitted to the video. Two parameters are to be fitted. The first one is the brightness of the frames in the video and the other one is the orientation of the rain streaks.

The brightness of the rain and orientation of the rain was fitted manually. Manually fitting the orientation could be done by looking at the falling raindrops. Fitting the brightness can be done by watching the transforms and make sure they have similar in brightness. In

order to fit it accurately some values were tested and the result was observed. If the result was good enough then that value for the brightness was kept.

Detection

By applying equation 2.13 a gray scale matrix is constructed. This gray scale matrix indicates where rain exists in the picture. One model per frame is created, but with manual fitting of the model parameters the model is rarely updated. The aim was to go further and implement a way to automatically estimate the parameters needed for the model. The automatic orientation estimation was implemented, but since the algorithm did not show enough promise this approach was not continued.

3.2.2 Skewness

In accordance with the theory in section 2.2 it was found that the skewness value somewhat highlights the rain streaks. With the combination of skewness and thresholds one could find pixels likely to be rain. This could be done with as few as three frames and it was found that a lower threshold could be introduced to remove pixels of stationary objects with lower skewness value than of the rain pixels.

However only taking the skewness of a sequence and using a threshold to differentiate the rain from other motion gives a detection matrix which contains the accumulated rain streaks of all the frames. This is unfortunate since there is no way to distinguish the rain from the current frame using only the skewness value of a single sequence of frames.

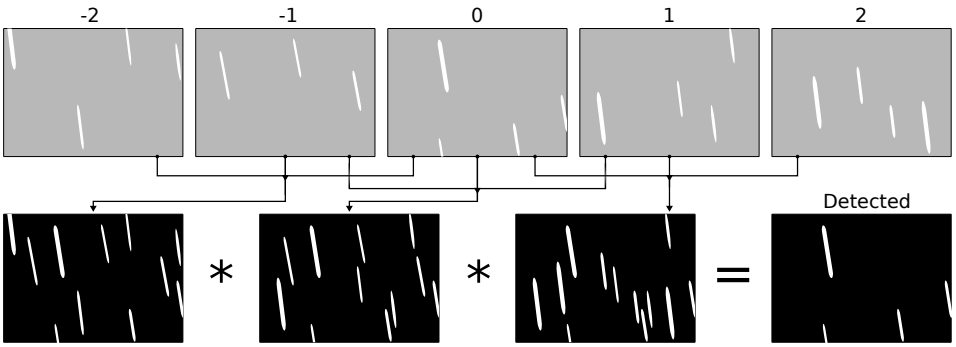

Figure 3.1: Illustration of how skewness can be used to detect the rain streaks from a single frame. The skewness is calculated for three sequences of three frames and a threshold is filtering out the rain. The results are converted to binary matrices and then multiplied together to get the streaks from frame 0 only.

This was solved by calculating the skewness for N sequences of N frames. The sequences are offset by one frame for each new calculation of the skewness value requiring a total of $2N - 1$ continuous frames. This algorithms is shown in figure 3.1 with skewness of sequences of three frames.

Skewness values for a larger number of frames (15 in this case) were calculated. With an upper threshold a more reliable highlighting of moving objects in the scene could be

made. The goal with the algorithm at this stage was to highlight moving objects and group stationary parts of the scene along with pixels with the rain.

3.2.3 Intensity

The intensity property of rain is a very central property of the phenomena and as such it is a powerful tool. The approach is straightforward but can be varied by introducing different thresholds. Several restrictions on the intensity spike characterizing the behaviour of precipitation were investigated.

The main property of the intensity change can be checked by measuring the height of the spike. In the visualization in figure 3.2 this is the threshold A. If time equal to zero is a rain pixel then the temporal neighbours have to have an intensity below A. This is the main idea of detection based on intensity changes.

It can also be noted that in general raindrops have the highest brightness in the center. The farther from the center, the smaller the brightness. So when using a threshold to differentiate rain from non-rain one has the choice of removing a lot of rain; this will result in an increase in false detection as the threshold is set lower.

Experiments were carried out using a single averaged intensity for a pixel and treating each color channel separately. In section 3.2.3 using the three color channels is expanded upon.

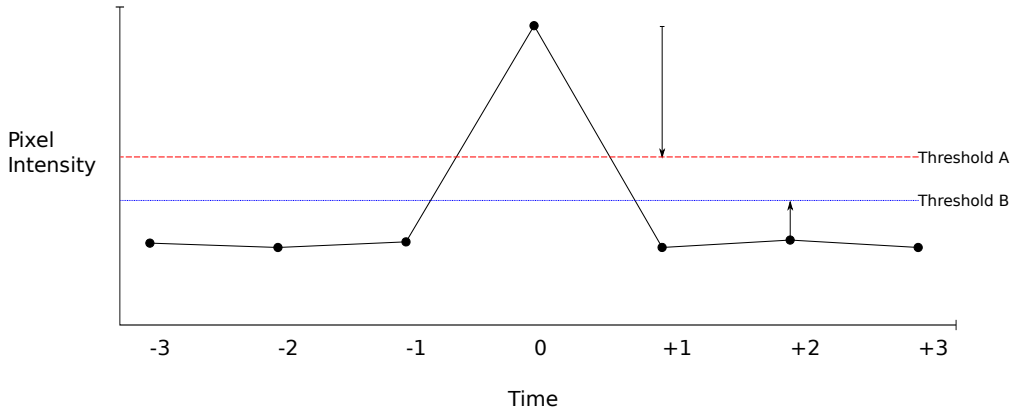

Figure 3.2: Intensity spike with thresholds determining if the spike is rain-like. Temporal neighbours to the pixel at time 0 have to be on the correct side of the thresholds if the pixel at time 0 shall be counted as a rain pixel.

Several variants of this approach were investigated:

Two frames: Utilizes two frames and only measures intensity increase and ignores any following behaviour of the pixel. The intensity increase must be higher than a threshold. An advantage of this method is that it does not use 'future' frames.

Three frames: The main idea in its original form. Compare the same pixels in three consecutive frames. The middle frame's pixel has to be higher value than the following and the previous by a specified threshold.

Five frames: The same as with three frames but with two extra frames. This condition follows $fiveframes <= threeframes$ with respect to pixels detected. The aim is to reduce false detections by making random noise less likely to trigger an intensity spike. By using five frames the threshold can be set wider.

Five frame + gradual increase threshold A continuation of the five frames variation to further decrease false detections. With threshold B as seen in figure 3.2 the gradual increase of the pixel intensity in time can be judged. Threshold B is used to prevent -1 from being higher than -2 by a set value, $+1$ and $+2$ are the same respectively. This further suppresses noise in the output but it also makes the algorithm more likely to fail to detect streaks when this noise is prevalent.

One disadvantage with using the intensity detection algorithm over three frames is that if raindrops appears at exactly on the same pixels in more than one frame, the difference in intensity will not be detected by the intensity detection algorithm. But the event that raindrops would appear exactly over the same pixels on two or more consecutive frames are unlikely [8].

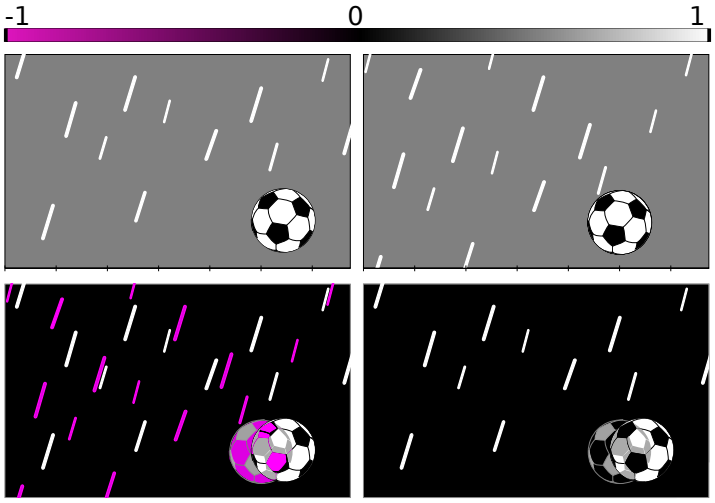

Figure 3.3: Image illustrating how working with a 2 frame intensity threshold works. The top right frame is subtracted from the top left frame giving the result in the bottom left frame. The negative values is then capped to zero giving the final frame located at bottom right. Highlighting all motion which is going from darker to brighter.

With this detection approach the pixels were treated as either rain or not rain. It is possible to make this more complex with a gray scale detection which marks pixels between $0 - 100\%$ rain. The benefits of having a continuous detection frame is that it can tell not only whether or not a pixel is affected by rain but also how much the intensity has changed due to the rain. The continuous detection approach when using intensity was not investigated in this thesis though.

Color neutralization

When the precipitation detection algorithms erroneously detects moving objects whose color is not white, this can be discovered in the detection matrix. It can be observed that those regions have a color which is not white.

Color neutralization attempts to reduce or remove these false detections. Colors other than white in the detection matrix means that the red, green and blue channel are not even. By uneven it is meant that the pixel intensities in these channels deviate from each other; they do not have the same values. The more the channels differ, the larger the deviation, the further away from the color white will the resulting pixel be. A threshold can be set in order to differentiate between acceptable and unacceptable deviation.

If the pixel values in the channels deviate too far away from each other, then those pixels can be considered as a non-precipitation pixel. In that case they can be removed from the detection frame by being set to zero.

One drawback of color neutralization is that it cannot handle false detections that yields even color channels. Examples are moving grayscale objects.

Whether color neutralization is needed or not is largely determined by the color quality of the processed image and ultimately the people wanting that image.

For an example demonstrating the color neutralization effect see the Example section 3.6.

3.2.4 Chromatic

To utilize the chromatic property of rain streaks as described in section 2.4 the different color channels of the images are examined.

If $\Delta R \approx \Delta G \approx \Delta B$ is true then it is a candidate rain pixel, otherwise it is not. To retrieve the difference a simple subtraction was performed on two following frames. Both absolute and percentage based restrictions were placed on the deviations between the channels and the results were explored.

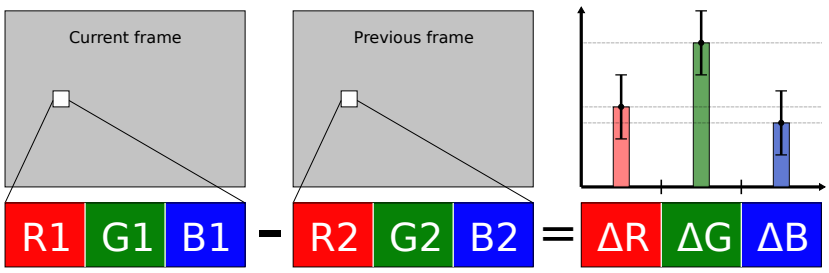

Figure 3.4: When comparing two frames the difference across each color channel needs to be in range of each other. It can be seen in this figure how the green color channel is outside the bounds of both red and blue whilst red and blue are of similar magnitude.

After extensive investigations it was found that the chromatic property of the streaks is best used in combination with other algorithms. The algorithm especially detects the edges of moving objects. The reason is because this is where the most drastic color change occurs. This is beneficial since it is here the replacement algorithms have the most problems as well.

If the moving object is brighter than the background, then the pixel intensity will increase when the object passes over it. As the object is passing the pixel intensity might decrease due to noise. This increase and decrease in pixel intensity could coincide with the intensity spike used in order to detect rain. Therefore the moving object might be detected. Whether or not it truly coincides with the intensity spike depends on how much the intensity will decrease.

Similarly if the moving object is darker than the background, then the pixel intensity will decrease when the object passes it. But afterwards the pixel intensity might increase due to noise. This increase and decrease in intensity could again coincide with the intensity spike characterising precipitation. Therefore the edges of the object might be detected.

3.2.5 Morphological

The morphological algorithm consists of three steps.

Size: Assess connected regions of potential rain pixels based on their size. If a region is to small, reject it, it is not large enough to be a rain streak.

Aspect Ratio: Calculate a region's approximate aspect ratio and reject the region if it does not conform to the aspect ratio of a rain streak.

Orientation (optional): Reject regions based on their orientation.

To be able to use a rain streak's aspect ratio to remove false detections one have to extract regions containing rain streaks. Any detection approach can be used to get a detection matrix containing rain streaks. By using the aspect ratio, which depends on the camera's exposure time, false detection can be removed.

To calculate the aspect ratio of a region the same approximation is made as in figure 2.9.

A general problem is that rain streaks can be broken into segments. This can be seen in figure 3.5. These segments do not conform to the aspect ratio or orientation of the whole streak. This mainly happens on large out of focus raindrops where the intensity spike is somewhat lower. A way to solve the problem with segmented rain streaks is to connect them together. The problem with connecting segments is that it is also possible to connect non-rain segment so that it appears as rain. This problem will lead to an increase in false detection.

Orientation estimation algorithms based on the detected regions were not created. Instead, a large threshold of acceptable values for the orientation of the regions were experimented with. Orientation is dependent on weather conditions and needs to be estimated for an accurate detection.

3.2.6 Blurring

Sometimes the regions in the image being treated might look a bit rugged. This is because the detection algorithm might only detect parts of the raindrops. It could also be that some pixels are falsely detected around the raindrop. One way to remove this rugged effect is to

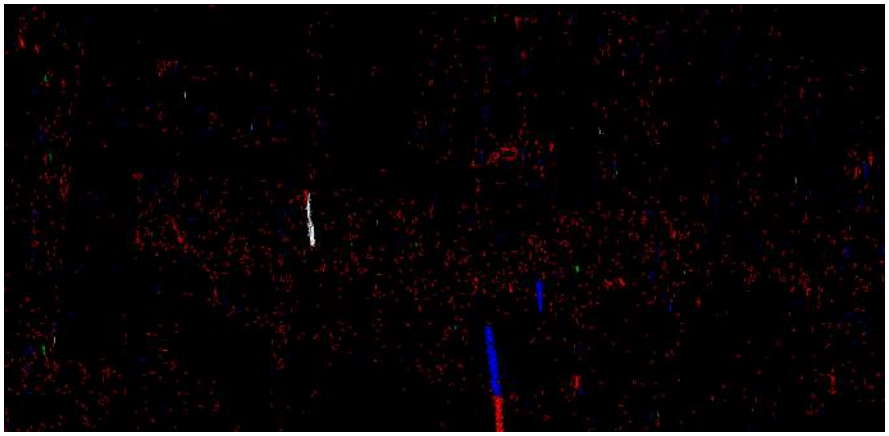

Figure 3.5: An illustrative detection matrix for a frame with rain, the original frame is in figure 4.1 which can be seen in section 4. In the image everything white is classified as a rainstreak, green is rejected due to aspect ratio, blue is rejected due to orientation and red is rejected due to both. The big raindrop at the bottom is detected but is split into three big regions and is misclassified. The used thresholds for aspect ratio and orientation are unforgiving in this example.

apply Gaussian blur on the rain detection frame before using it to replace the rain with the background color.

Example demonstrating the effect of rugged pixels can be seen in section 3.6.

3.3 Restoration Algorithms

Once the pixels have been detected as rain they need to be modified and the color need to be replaced by an estimation of what is behind the rain streak. This approximation can be done via several methods as will be described in this section.

A detection algorithm may output a binary detection matrix or a gradual detection matrix. If a pixel can be marked as partially a raindrop this needs to be handled by the restoration algorithm. To handle this $\alpha - blending$ is performed between the restoration source H and the original image M,

$$pixel = \alpha H + (1 - \alpha)M. \tag{3.1}$$

3.3.1 Median

Taking the median across a number of frames will return the pixel with the middlemost intensity of the pixels. The expectation with this method is that less than half of the frames have raindrops at the replacement pixel. This method uses the fact that rain streaks are brighter than their background and will therefore all lie on the same side of the median value.

The median over three frames were taken and the frame where the pixels were marked as a rain were excluded. If two of these pixels are covered by a rain streak the restoration will fail, otherwise it will replace the pixel with the median of the color from the previous three frames.

3.3.2 Mean

In this replacement algorithm the mean of a number of images is taken in order to obtain an average color of the pixel. This average color is then used to replace the detected rain pixel. It should be noted that taking the mean of too many images at once might introduce additional unwanted colors if the scene is non-static due to other factors than rain.

3.3.3 Subtraction

In this approach the intensity difference between the rain-free pixel and the rain-affected pixel are subtracted away from the rain pixel. The reason is because rain will increase the pixel intensity by a certain value. By subtracting away this value, the original rain-free value is achieved for the pixel. In other words, the additional intensity increment caused by the rain is removed by being subtracted away.

3.4 Data gathering

Both images and videos containing precipitation were gathered and used during this thesis. Some examples are shown in section 4.4. Authentic rain was not available the majority of the time so this was at times challenging.

The difference between testing on ‘images’ and ‘video’ was not significant. The distinction is made because of differences in acquiring material and the small practical usage differences.

3.4.1 Images

Image testing was an simplified first step to test algorithms. Some algorithms or part of algorithms did not use the temporal information provided by a stream of video thus testing on images became a first natural step. Even algorithms which used adjacent frames was frequently tested on images first.

Images used for algorithm testing consisted of both artificial rain images and images with real rain. In the case of artificial rain, the artificial rain was introduced to the rain-free image in order to study how rain affected the frequency representation of an image. The raindrop itself is modelled by a blurred Gaussian with a pre-defined size, brightness and orientation.

Rain generation

The artificial rain generation code, has its roots in the implementation of Barnum and Narasimhan’s [2] frequency space algorithm. The model the algorithm uses could with

some modifications be used to generate rain streaks. The rain streaks are Gaussian blurred gradients which can be adapted with a set of parameters. The streaks are then created from distributions and spread randomly over the image.

Variable	Description
Theta	Changes the direction of the rain streaks.
NumberOfStreaks	Amount of rain.
f	Focal length of thought camera. An increased value gives a wider and longer streak as can be deduced from the equations 2.3 and 2.2
z	Depth of raindrop. An increased value gives a smaller streak.
e	Exposure time of thought camera. An increased value gives a longer streak
A	A scalar which can be used to adjust the brightness of the streaks. Useful to get a brightness in accordance with the scene. An example would be a scene with a street lamp illuminating the rain; to compensate for this in the model the A value would have to be changed.

Advantages with the artificially created rain was the high level of control and easy access to different types of rain. Another useful aspect was that the correct values of the rain parameters were known. An example when this came in handy was during the development of a program that estimates the orientation of the rain. The problem with using artificial rain is that since it is based on the same model used in rain removal, the correlation between the artificial rain and the model used in the rain removal are unrealistically high.

A drawback with using generated rain is that, whilst different types of streaks can be generated, nature offer so many more varieties of rain. The artificial rain always has some of the same base characteristics to it while authentic rain is more chaotic. Another drawback of the implementation was the fact that it only allowed a single orientation for all streaks in the entire image. Real streaks are more dynamic and do not fall in the same direction. This problem could be solved by running several passes over the same image with different orientation and saving it in between. And finally, the artificially generated rain does not contain noise. Real rain being filmed might contain noise due to the camera settings.

Different scenes were used as images in which rain were applied. The images were taken from various web sites.

3.4.2 Video

Most of the validation was done by applying the algorithms developed to the streams of frames. The videos were mainly fetched from various websites, including websites which stores the results of previous research. Especially the videos supplied by Barnum and used in their work [2] were used. These videos had an unaffected rain/snow part and a rain/snow suppressed part which was useful for comparison.

The supervisors at Axis also supplied some footage of snow and rain. In addition to that, real-life videos were also filmed with an Axis camera. The scene in the real-life video consisted of a road, a parking lot and several structures. Pedestrians and vehicles on the

Figure 3.6: Example image with artificial rain. The artificial rain is much more ordered than real authentic rain.

road served as moving objects. The algorithms in this thesis has been applied on all the materials mentioned and the most promising results are presented.

A third source of footage was the Internet which offered a wide range of different scenes in different movies and types of rain.

3.5 Priorities

With respect to the variables involved when removing rain, a prioritization fitting to the context of video surveillance were decided. During the early testing and the development-of-algorithms phase the time aspect was less focused on in favor of visual results. The final relative prioritization was as follows:

1. Avoid false detection of moving objects as rain
2. Correct detection of in-focus raindrops
3. Speed of calculations
4. Rain replacement quality
5. Avoid false detection of static objects as rain
6. Handle moving camera
7. Correct detection of out-of-focus raindrops

The priorities were not entirely static during the project and changed due to debate and results. One aspect which was valued more highly in the beginning was the ability for the algorithm to handle a moving camera. It was deprecated as a result of discussions with supervisors and its relative impact on the rain removal result.

3.6 Example

This section demonstrates the one way of using the precipitation removal algorithm for snow step by step.

By using the intensity method as the only detection algorithm the snow is correctly detected. A problem is that the walking people and umbrellas are also falsely detected which can be seen clearly in figure 3.8. In order to remove the false detections, color neutralization is applied. Most of the false detections are thus removed but the result looks a bit pixelated, especially at the top of the red umbrella and at the black shoes as can be seen in figure 3.7. In order to remove this artifact Gaussian blur is applied.

Figure 3.7: Going from top to down, left to right. Image 1 is the original snow covered frame. Image 2 is image 1 with the snow removed. Notice the piece of the red umbrella being removed by mistake. Image 3 is image 1 with the snow removed. Notice unaffected umbrella. This is because color neutralization has been applied to the detection frame. But some very small areas are a bit pixelated and this can be removed with blurring. Image 4 is image 1 with the snow removed. Color neutralization and blurring has been applied. See figure 3.8 for the difference in the detection frames between each step.

Figure 3.8: Going from top to down, left to right. Image 1 is the detection frame. Image 2 is image 1 with color neutralization applied. Image 3 is image 2 but with blurring applied. Image 3 is also the final detection frame. As can be seen, in every step the false detection (peoples, umbrellas) are reduced.

Chapter 4
Results & evaluation

This chapter will present how the algorithms perform. Later in the chapter in section 4.4 images processed by each algorithm is displayed.

The images in figure 4.1 were taken from a video stream recorded on Monday 2014-04-14 in Lund, Sweden. The camera used to record this video was an AXIS P3367 Network Camera. It supports 5 megapixels or HDTV 1080p (2592x1944) quality. The sunny image is taken with the same camera the next day.

Disturbances in the video include rain streaks, a flying bird and raindrops, which lands on the dome of the camera. The raindrops on the camera's dome appear as smudges and is not in the scope of this thesis to remove as their behaviour is entirely different from falling rain.

4.1 Time measurement

To measure the time of the algorithms Octave's command cputime() was used. The command has the following documentation:

> Return the CPU time used by your Octave session. The first output is the total time spent executing your process and is equal to the sum of the second and third outputs, which are the number of CPU seconds spent executing in 'user mode 'and the number of CPU seconds spent executing in 'system mode ', respectively.

The time estimation is dependent on the specific implementation and is mostly useful for comparison between the implemented algorithms. To measure the time the mean of the algorithms' performance over 30 frames was calculated. The frames itself consists of 1650x800 pixels. This was done on a stationary computer with a Intel(R) Core(TM) i7-3770 CPU 3.40GHz processor.

Figure 4.1: The first picture is a rainy scene with a flying bird taken in Lund, Sweden. The second picture is a rainy scene with two cars. The third picture is the same scene taken under good weather conditions the next day

4.2 Evaluation of Detection Algorithms

The detection algorithms' usefulness is evaluated with detection quality and time complexity in mind. To measure the results of the algorithms two rough detection matrices were constructed for the selected frames. All rain streaks were manually marked in one frame to test rain detection. In another other all moving objects was marked to measure false detections in those objects. The two matrices can be seen in figure 4.2

To evaluate detection figure 4.2 wand compare it to the algorithms' outputs. Then each pixel can be categorized into the categories $true\_positive$, $false\_positive$, $true\_negative$ and $false\_negative$.

The category $true\_positive$ denotes all the rain pixels being correctly detected as rain. Similar to the category $false\_positive$ denotes all the non-rain pixels falsely detected as rain pixels. The category $true\_negative$ denotes all the non-rain pixels correctly not being detected as rain. Finally the category $false\_negative$ denotes all the non-rain pixels falsely

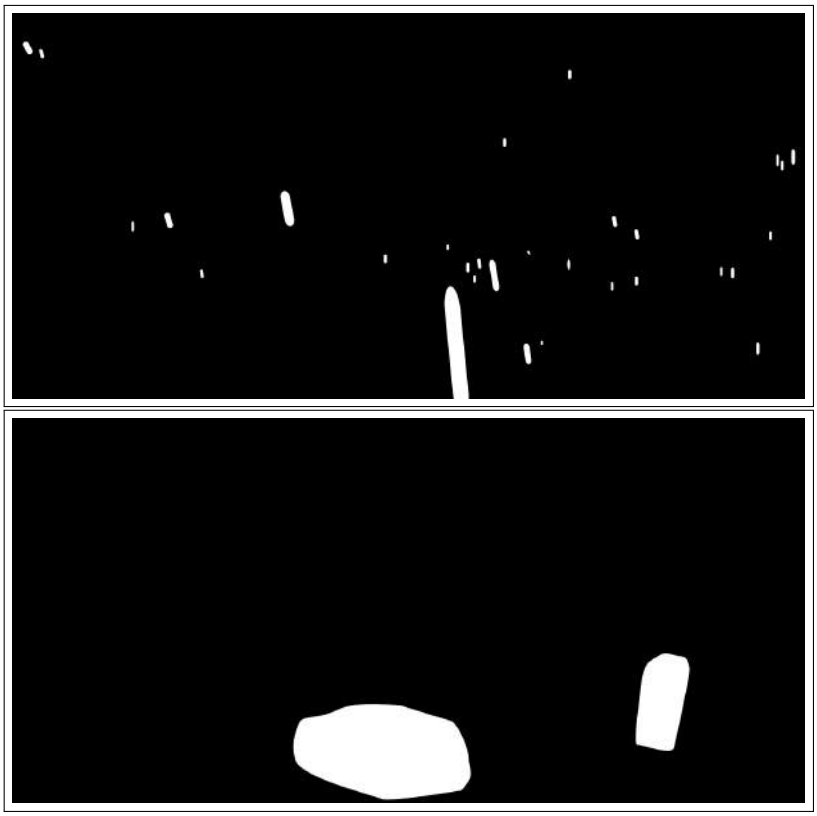

Figure 4.2: Two manual detection matrices for the frames in figure 4.1 where individual streaks and moving objects have been marked respectively.

being detected as rain.

Using these values the following metrics the *precision* of an algorithm can be calculated as

$$precision = \frac{\sum true\_positive}{\sum true\_positive + \sum false\_positive}, \tag{4.1}$$

which gives a precision value which says how often the algorithm is correct when it outputs that a pixel is rain.

The *sensitivity* of an algorithm is a measure of how many of the actual rain pixels are correctly identified as such. It can be seen as how much of the rain in the frame that is covered by the algorithm,

$$sensitivity = \frac{\sum true\_positives}{\sum true\_positives + \sum false\_negative}. \tag{4.2}$$

Finally the *accuracy* metric is measured by

$$accuracy = \frac{\sum true\_positive + true\_negative}{\sum true\_negative + true\_positives + \sum false\_negative + \sum false\_positives}, \tag{4.3}$$

and describes how correct the outcome of the algorithm is overall.

Frequency		
Precision (%)	Sensitivity (%)	Accuracy (%)
<2	>90	<10

Table 4.1: This table present the general performance of the frequency algorithm.

This evaluation scheme depends greatly on the evaluation matrix in figure 4.2. This matrix roughly aligns with the rain streaks but it has been intentionally constructed with a good margin so that the highlighted areas cover the entire raindrop and a bit of its surroundings. This adds an error where the manual detection is wrong. This error is somewhat contained, since the algorithm produces a lot of random false detections. These random false detections will be distributed across the frame, both inside and outside the manual detection. A specific problem where algorithms falsely detects pixels near streaks has not been found.

Due to the fact that falsely detecting moving objects are more severe, this was also measured with the second manual detection frame in figure 4.2. The only relevant value is considered to be the coverage of the moving objects, the sensitivity.

The metrics negative predictive value and specificity where also examined. These values are similar to the precision and sensitivity but looks at an algorithm's ability to predict negative outcomes correctly. These values were omitted from the report since they provided little usefulness and did relate to our priorities less strongly.

4.2.1 Frequency

A complete evaluation of this algorithm's true potential was not possible since the final step was left out. However the developed algorithm was evaluated.

The frequencies detected contain much noise in the environment. This is undesirable since the noise ruins moving objects to a great degree by giving too much false detections.

One advantage with the frequency approach is that the rain streaks are detected with a ratio giving a value of how much of the color at the pixel is due to rain. This prevents jagged edges in the resulting picture after the pixels have been restored.

In table 4.1 extremely lacking in the precision metric and also very low values for the accuracy metric can be seen. The model was not able to be fitted to the particular rainfall in the frames, the rain was not detected at all. The rain streaks are varied in size and are not very pronounced which makes it hard for the model. Better rain detection have been achieved under some circumstances but there is always significant false detections prevalent.

The time consumption of creating the model is also a disadvantage compared to other algorithms. It takes around 8 seconds per frame to compute the model and use it to compute the detection matrix. Additional computational time would be required if the orientation and scalar would be estimated as well. If the frequency model is to be viable, then it needs to be created in a more efficient way

4.2.2 Skewness

The skewness property has been investigated in two ways. The theory in [7] states that the skewness of rain pixels is different from non-rain pixels. It has been found that scenes are too dynamic and a threshold cannot be reliably set to differentiate rain from moving objects.

A lot of fine-grained noise decreases the precision a lot.

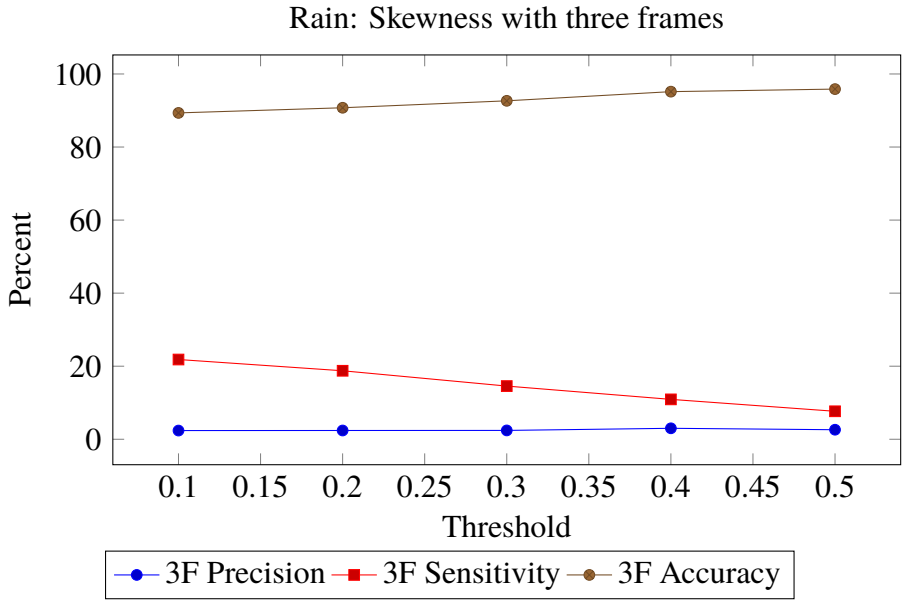

Figure 4.3: The precision is abysmal which comes from an evenly distributed noise in the detection. The rain streaks are outlined by the algorithm but are drowned out by the noise. Movements are also not distinguised well.

In figure 4.3 and 4.4 it can be seen how the skewness is weak when detecting the streaks. Using 15 frames could potentially be useful if it were not for the fact that all raindrops in the 15 frames get detected. They could be removed with the same method as in the case with three frames but this would require using 15 frames into the future.

In plot 4.5 it can be seen that the skewness algorithm avoids detecting the car and detects as little as five% of it. This needs to be compared to the value at the same threshold in plot 4.3 and it can be observed that the rain detection at this threshold is poor.

The time consumption of the algorithm varies around 380 ms when using three frames and multiplying them together; keep in mind that the skewness is taken for three sets of three frames. The time consumption for 15 frames with no further modification besides thresholds are around 375 ms. Variation in the thresholds has a small impact on the time required to produce the detection matrix.

4.2.3 Intensity

The intensity algorithm has several parameters which change the outcome in a way which presents a trade off between false detections and detected rain. As stated the priority is

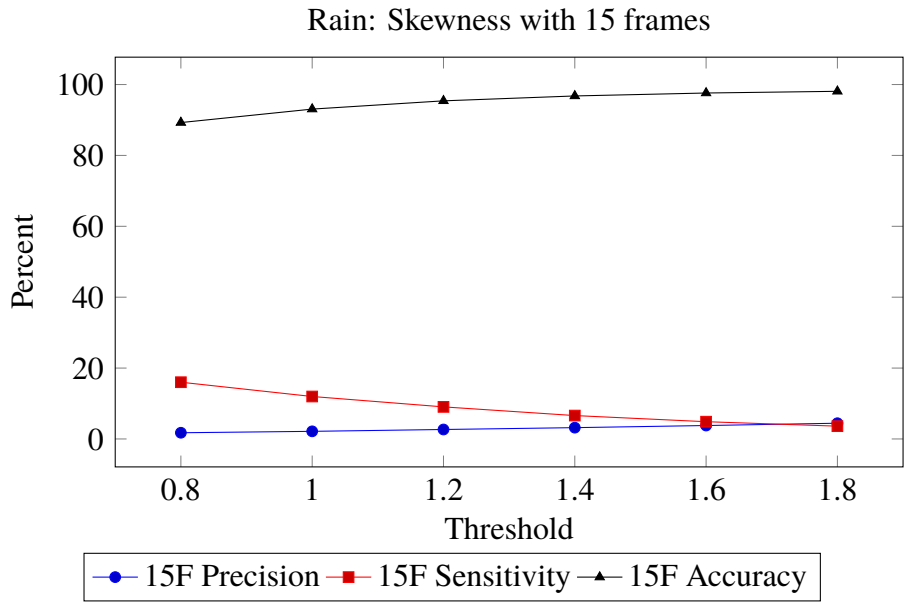

Figure 4.4: Similar to the plot 4.3, lacking values can be seen. When using 15 frames, rain streaks from all the 15 frames are kept. Using 15 frames actually has significantly less noise.

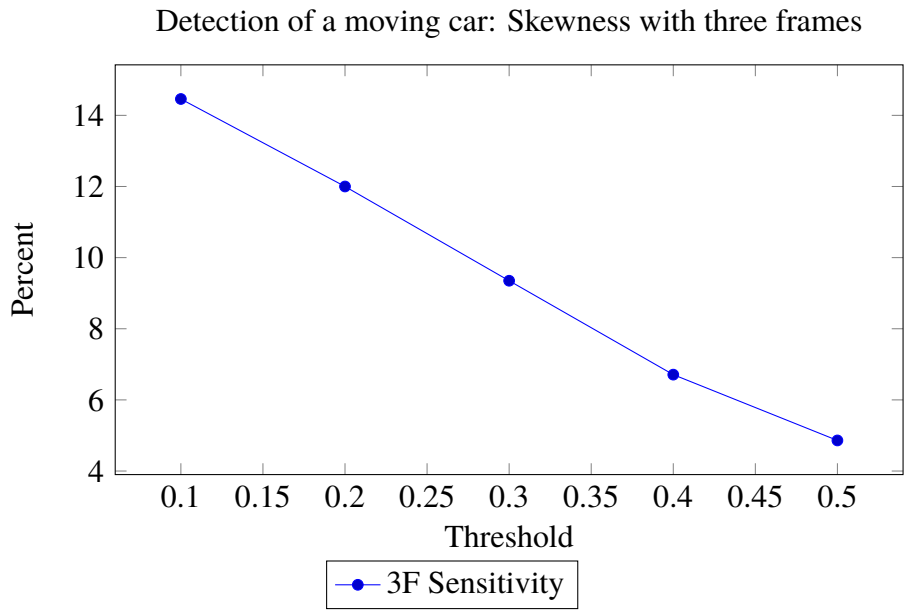

Figure 4.5: Here it can be seen that the skewness algorithm detects less and less of the car when increasing the threshold, which is expected.

to not introduce false detections in moving objects and this is taken into account in the evaluation when using the second evaluation image where the moving objects in the scene are marked.

The intensity is measured as a value between 0 and 255, the threshold is compared to a value in this range.

Rain: Intensity - three frames

Threshold	Precision (%)	Sensitivity (%)	Accuracy (%)
5	5.36	25.55	95.00
7	7.30	20.01	97.19
10	10.69	14.51	98.66

Table 4.2: A table of how well the intensity algorithm performs on the rain frame in 4.1 when using three frames with different thresholds.

Rain: Intensity - five frames

Threshold	Precision (%)	Sensitivity (%)	Accuracy (%)
5	8.81	20.31	96.83
7	12.21	15.59	97.85
10	18.08	11.01	98.48

Table 4.3: A table of how well the intensity algorithm performs on the rain frame in 4.1 when using five frames with different thresholds.

In the tables 4.2 and 4.3 it can be seen that as the thresholds are becoming more strict, the precision gets better while the sensitivity gets worse. It can be seen that using five frames seem to perform slightly better and allows the threshold to be set lower for the same results. Comparing three frames and seven in threshold to five frames and five in threshold it can be seen how they share similar sensitivity while the five frame case has slightly better precision.

In table 4.4 the variation of sensitivity can be seen. The sensitivity in this case describes how much of the moving objects that are classified as rain. As expected it can be seen that with smaller thresholds, the worse the algorithms perform and detect larger parts of the cars.

The second threshold as proposed in 3.2 has the aim of reducing false detections. The results can be seen in figure 4.8 and 4.9 which uses an additional threshold of 3. The rain removed is almost halved but the precision is improved. The benefit can be seen in

Moving car: Intensity - three and five frames

Frames	Threshold	Sensitivity (%)
3	5	14.56
3	7	12.35
3	10	10.21
5	5	8.37
5	7	7.34
5	10	6.28

Table 4.4: A table of how well the intensity algorithm perform on the moving car frame in 4.1 In this table high values of sensitivity are disadvantageous.

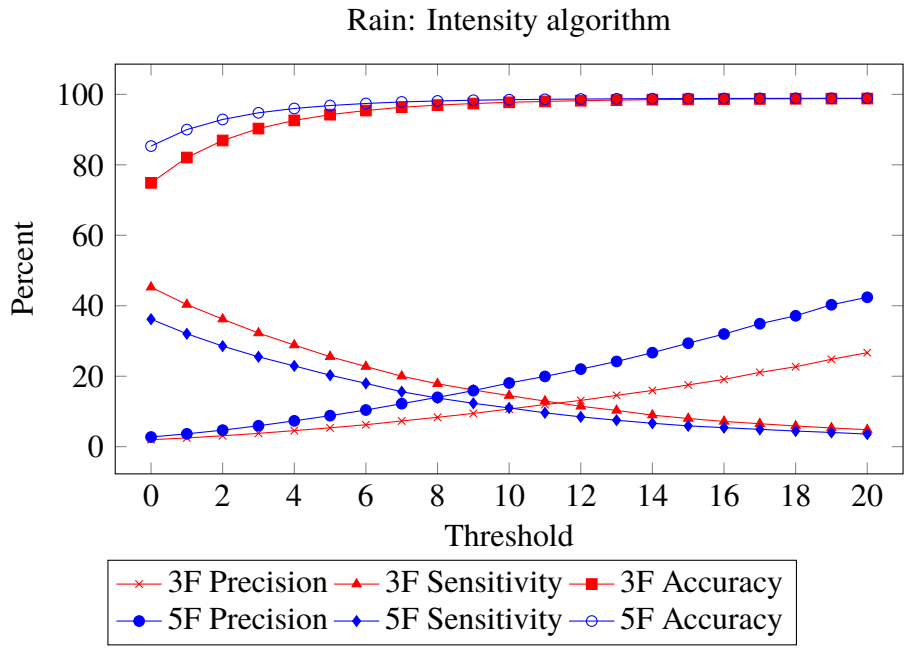

Figure 4.6: This plot visualizes three quality measures for the intensity algorithm for both three frames and five frames. Using five frames has its advantage in the precision and it becomes superior to three frames as the threshold is increased.

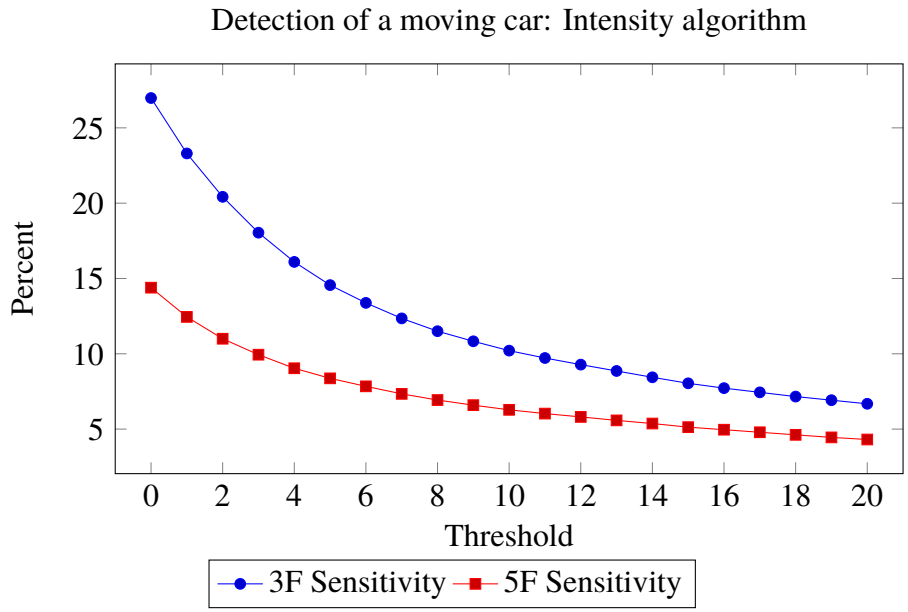

Figure 4.7: A plot of the intensity algorithm's ability to distinguish moving objects. The sensitivity in this case is a measure of how much of the moving objects in the scene are falsely detected

the second plot where it detects less than half of the car in relation to what it did before. However the the decrease in sensitivity is similar both for rain and moving objects. This second threshold (set to 3) may not be better than random removal of pixels from the

detection matrix and was discarded as an approach, but if one desires higher precision over sensitivity this is a possibility.

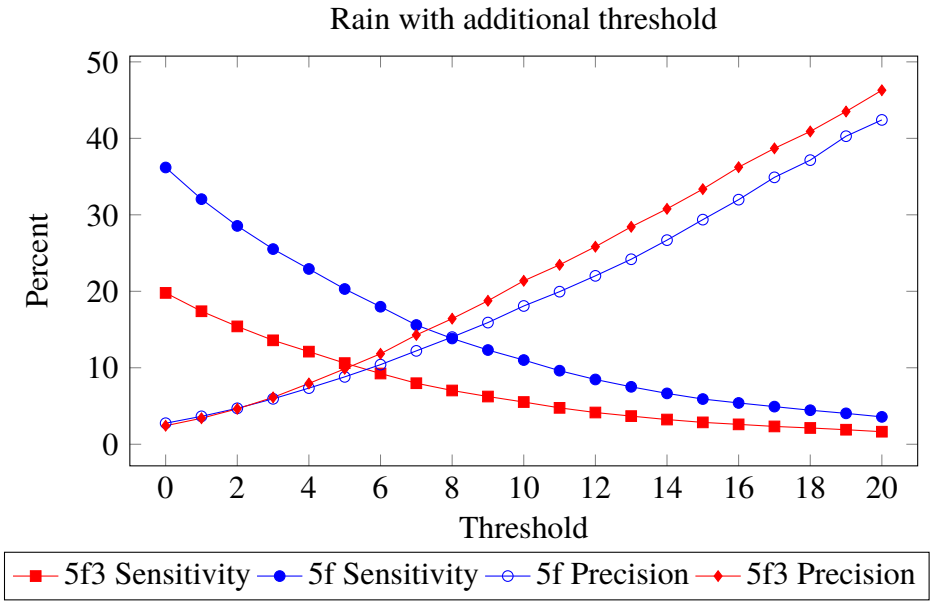

Figure 4.8: This graph shows what is gained and what is lost when applying a second threshold, it can be seen how the precision increases slightly but the sensitivity decreases. Also compare to 4.9 to see how it affects false detections in moving objects. 5f3 means five frames has been used and a small threshold of three in addition to the threshold along the x-axis.

The plot in figure 4.10 shows how the time consumption changes when the threshold changes for the intensity algorithm. The change is small and is due to that there are a few more instructions which is executed if pixels are actually potentially rain.

4.2.4 Chromatic

The chromatic algorithm will be evaluated in its ability to detect moving objects which are not classified as precipitation. One way to use the chromatic property is to remove falsely detected objects from a detection where precipitation and falsely detected objects are present.

To determine the effectiveness of the approach the amount of pixels detected inside the moving objects of the mask in figure 4.2 are compared to the pixels detected in the static background and the rain. It does not really matter if the algorithm detects static parts of the scene as long as it does not detect the rain streaks, this distinction is not made with the evaluation with the mask.

The low value of the sensitivity is explained by the lacking detection in the middle of the car where a red color replaces a similar red color from frame to frame. It can also be seen in figure 4.11 how most of the detection is near the contours and details of the car which is to be expected.

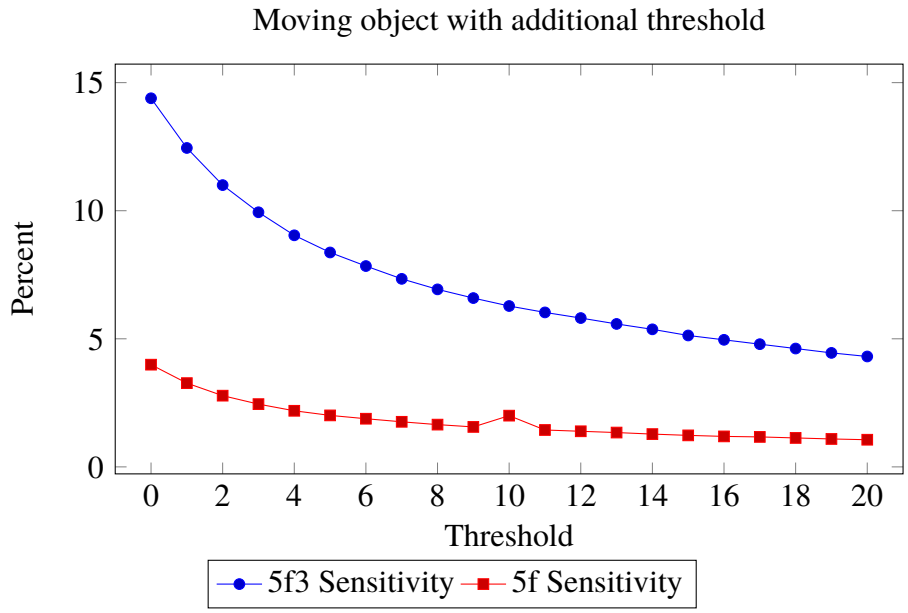

Figure 4.9: This graph shows the benefit of using an additional threshold as it detects less of moving objects. This graph relates strongly to 4.8 as it important that the additional threshold does not impact the detection of rain too much.

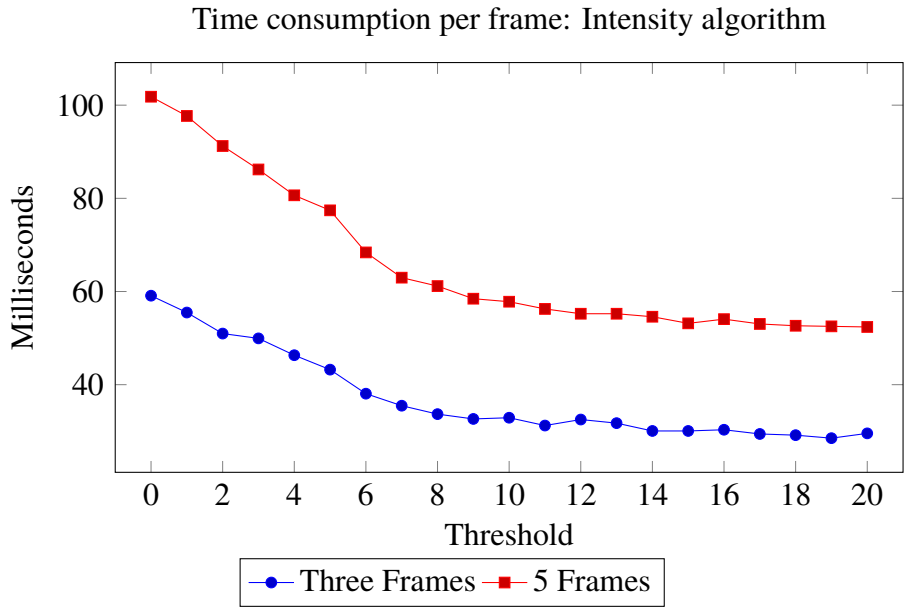

Figure 4.10: A plot detailing how the time consumption of the intensity algorithm decreases somewhat when it detects less pixels.

It can be seen in table 4.5 how the precision suffers with a low threshold. When combining this algorithm with other algorithms the low precision value will make the contribution of this algorithm reduce many correct detections made by the primary rain detection algorithm.

The behaviour of the algorithm needs to be analyzed in conjunction with whichever

Figure 4.11: The result from the chromatic algorithm which has been compared to the mask in 4.2. A threshold of 16 is used.

Moving car: Chromatic

Threshold	Precision (%)	Sensitivity (%)	Accuracy (%)
7	48.86	30.87	94.20
13	88.23	18.52	95.20
19	94.88	13.16	95.00
0.1%	11.38	56.65	72.33

Table 4.5: A table of how well the chromatic property can distinguish moving objects. This is primarily determined by the precision and sensitivity values.

other algorithm being used with it. The worse the original algorithm is the more benefit this algorithm will provide. Some good results have been observed, however for the particular evaluation frame with the car, the overlap with the intensity algorithm is rather poor and the result is not vastly improved.

The values in table 4.5 must be compared to the table 4.6, where it can be seen how much of the rain it detects. It can be seen that at lower threshold the result has too low precision and covers many rain streaks where it should not.

The last entry in the table 4.5 shows the result from when using a percentage based approach rather than an absolute value. The result is poor and there is no additional theory

Rain: Chromatic

Threshold	Precision (%)	Sensitivity (%)	Accuracy (%)
7	7.57	14.7	97.1
13	12.15	1.68	98.79
19	7.72	0.13	98.89

Table 4.6: The chromatic property has been used to detect objects which are not rain. Low values of sensitivity is good. The top value of 14.7% sensitivity means that the algorithm covers 14.7% of the rain streaks, which is very bad, but it gets more acceptable with lower thresholds.

which supports the notion that color changes of rain would depend upon the intensity of the values. The 0.1% threshold equals that the individual $\Delta R, \Delta G, \Delta B$ is within 0.1% of the $mean(\Delta R, \Delta G, \Delta B)$

The chromatic algorithm takes around 90ms to produce the detection matrix. The time required does not vary much if the threshold is changed.

4.2.5 Morphological

Aspect ratio will be treated as an addition to the intensity algorithm. Its goal is to remove false detections based on regions aspect ratio's, it will increase precision while not decreasing sensitivity.

The Morphological algorithm contains a component which removes clusters of pixels depending on the size of the cluster. This operation is helpful and could be combined with the other algorithms to improve the results, as it removes noise pixels.

When the testing was performed without known values of the exposure time of the camera, the range of aspect ratios allowed was chosen to catch most rain streaks. Removal of too small regions was also performed, this removes a lot of noise and reduces the computation time significantly.

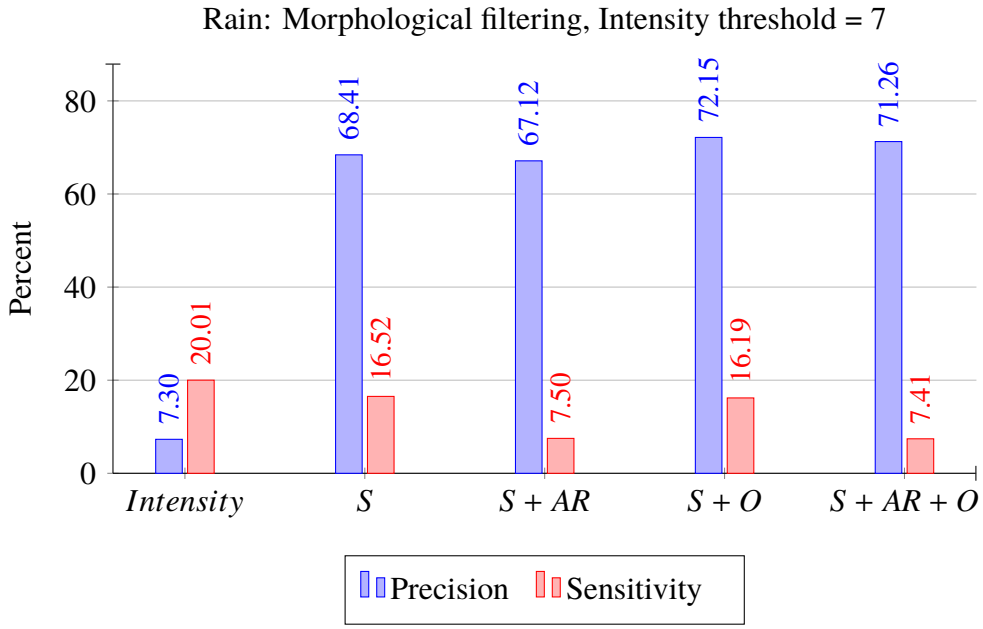

Figure 4.12: Morphological properties applied to an intensity detection matrix one by one. S stands for size, AR is aspect ratio and O is orientation. Precision is increased significantly in all cases due to the removal based on size. Aspect ratio removal does not improve results in this case.

The diagram in figure 4.12 shows that the morphological properties of rain can grant an significant increase in precision while decreasing the sensitivity. In the evaluation several bounds for aspect ratio and orientation were used. The diagrams shown here reject regions which have an orientation larger than 60 °, 90 ° being horizontal. With proper estimation of orientation this property could provide larger benefits.

The removal of regions based on size is done with a fixed value for all results, chosen at nine after experimentation. The removal based on size happens first in the algorithm and affect the time consumption significantly. Calculations are performed per region and the size removal saves a lot of further calculations when aspect ratio and orientation is computed.

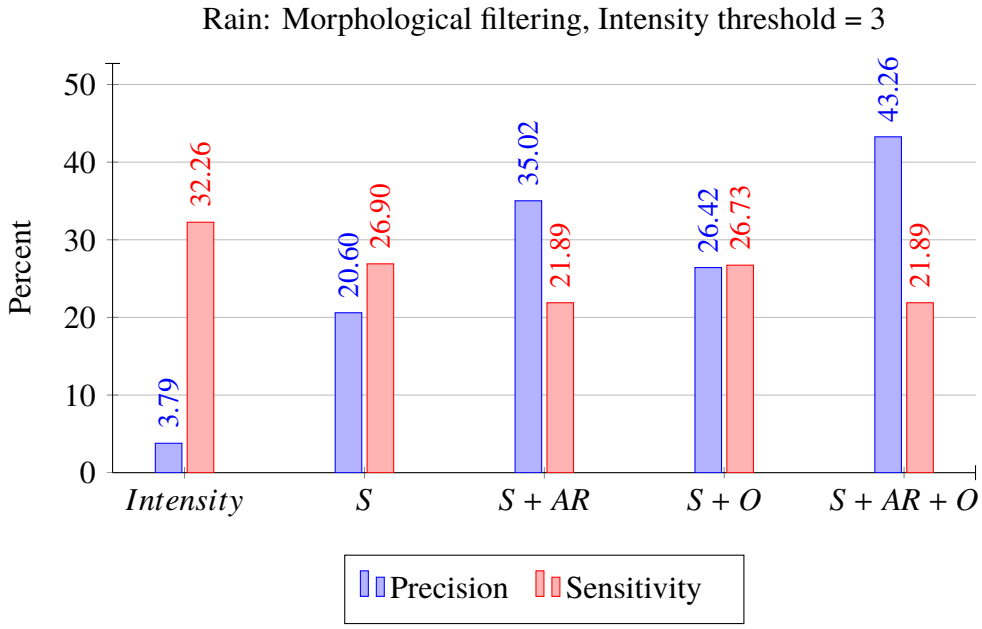

Figure 4.13: Morphological properties applied to an intensity detection matrix one by one. S stands for size, AR is aspect ratio and O is orientation. In this case the intensity threshold is set to allow more potential pixels through and in turn rely more on the morphological algorithm.

Comparing figure 4.12 to 4.13 it can be seen that the initial sensitivity value is much higher than the latter and ends up with a more balanced result after using the morphological algorithm. Starting with a higher sensitivity value gives the algorithm more to work with and it can be seen more clearly in this case how the aspect ratio is helping precision.

The use of the algorithm also improves the result on the moving car frame. As can be seen in 4.14 the amount of falsely detected pixels in the moving areas are reduced drastically. Having few false detections in specifically moving objects is number one in the prioritization in 3.5 this is a large benefit.

The amount of calculation is heavily dependent on how many regions the initial detection matrix consists of. In the case above where an intensity threshold of seven was used the computation time for one frame was roughly one second, compared to when the threshold of three was used it increased to 21 seconds. For a video to be real-time the computation time may not exceed 30 ms.

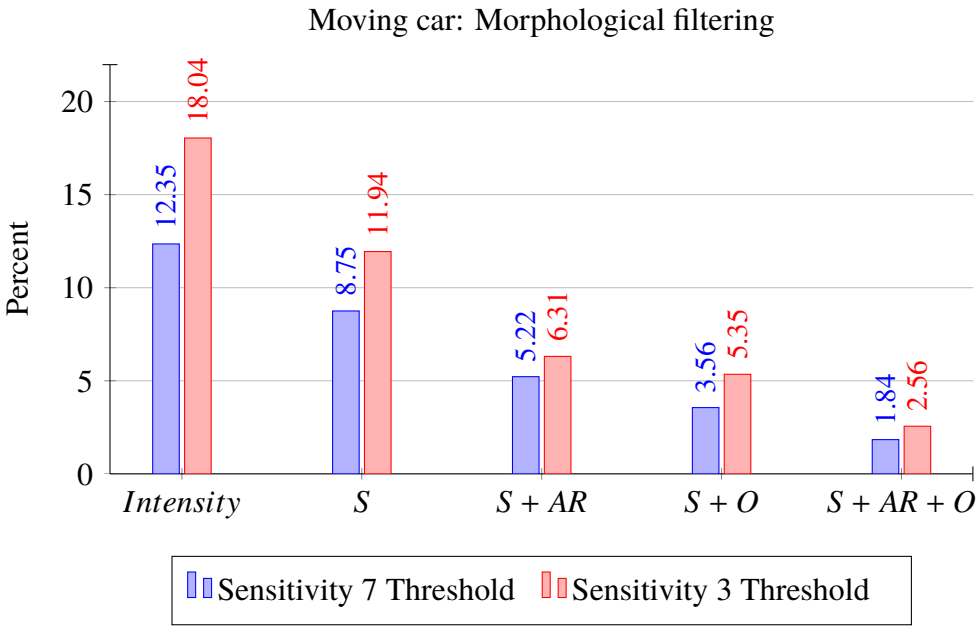

Figure 4.14: How much false detection, in the important moving objects, the morphological method reduces. The boundaries for aspect ratios relate to the camera exposure time.

4.3 Evaluation of Restoration Algorithms

In this section the restoration algorithms will be evaluated. The original image and the detection frame is from figure 4.2.

Figure 4.15: The result from the restoration algorithms using the temporal mean.

Figure 4.16: The result from the restoration algorithms using the temporal median.

Figure 4.17: The result from the restoration algorithms using the subtraction approach.

In the results in figure 4.15, 4.16 and 4.17 the manually created rain detection frame from figure 4.2 were used. Since that rain detection frame is manually created it is considered as a perfect rain detection. Under such circumstances the restoration algortihms using the mean, median and subtraction approach are considered equally good. There are slight differences but they are very small.

4.3.1 False detection

If the rain detection also yields false detections then the mean, median and subtraction replacement algorithms will yield different results. In the examples below the bird is falsely detected and treated.

The original untreated image zoomed in on the bird is shown in figure 4.18

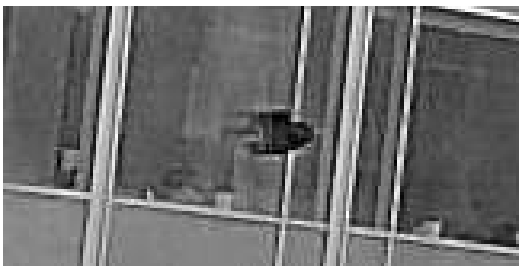

Figure 4.18: The original bird.

The difference is that mean and median will produce a subtle false detection of the bird. This can be seen both in front of the bird and behind the bird. There should only be one bird in the image, but there are three birds. One that is distinctly visible (in the middle) and two others that are faintly visible (to the left and right of the distinct bird). The number of faintly false bird being introduced depends on the number of frames one takes the mean or median over. In this case the mean and median was taken over two frames, therefore two false birds.

Subtraction on the other hand will only produce one false detection of the bird. This false detection can be seen behind the bird. Since subtraction only use one additional frame for restoration only one false bird will be introduced.

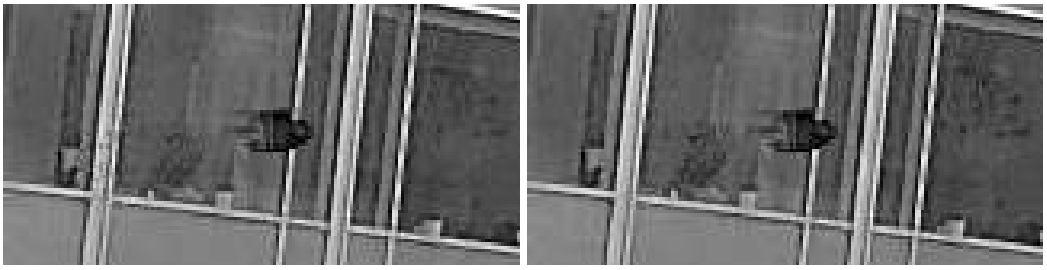

Figure 4.19: The results of the mean restoration approach to the left and median to the right. Notice the faintly visible birds next to the distinct bird. They appear both left to the bird and right to the bird. The left one can be seen slightly behind and below the bird. The right one can be seen a bit in front of and above of the bird.

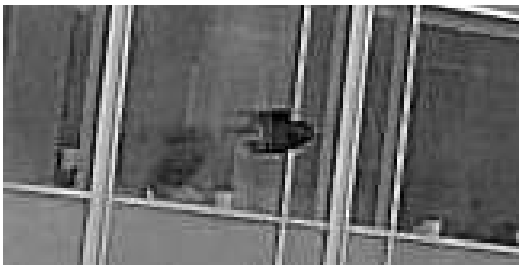

Figure 4.20: The result of the subtraction restoration approach. Compared to figure 4.19 there is only one faintly visible bird. It is located slightly left to the real bird

In figure 4.19 and figure 4.20 the intensity detection approach is used. In order to demonstrate false detections the threshold is deliberately set to zero.

4.3.2 Comparison metrics

The true color behind a raindrop can not be known for sure since the raindrop can be concealing something which disappears as soon as the raindrop is gone. But an estimate of the color which the raindrop is covering can be made. This estimate is done by taking the mean of 100 frames in the regions where precipitation is present. For a further discussion of the limitation of this method, refer to section 5.3.1.

In order to evaluate the restoration algorithms the intensity deviation is calculated.

It is defined as

$$intensity\ deviation = \frac{\sum(mean(image_1 ... image_{100}) - treated\ image)}{number\ of\ pixels}. \quad (4.4)$$

The intensity deviation indicates how much the pixel intensity for the rain area in the treated frame differs from pixel intensity in the rain area from the correct frame. The number of pixels are the number of non-zero pixels in the upper image in figure 4.2.

A value of 0 indicates that there is absolutely no difference in color and pixel intensity between the treated frame and the estimate. An absolutely correct match, in other words.

Algorithm	Intensity deviation (value)	Intensity deviation (%)
Mean	4.76	1.8
Median	5.37	2.1
Subtraction	10.8	4.2

Table 4.7: A table showing the intensity deviation between the mean of 100 frames and the treated frame. The range between the maximum and the minimum pixel intensity is 256.

The larger the value, the larger the difference. This difference is an approximation of how much each pixel deviate from the true value.

As can be seen from table 4.7 the difference is very small. Such small differences are not visible to the naked eye.

The results in table 4.7 are expected since the restoration algorithms used are in this case somewhat similar. The difference between the mean or median of two neighbouring frames in time and taking the value of the previous frame is not large. This is because the regions around the precipitation are rather static.

4.4 Algorithm results

This section will present algorithm outputs for the different algorithms covering the detection and replacement steps. The results are shown in brief before and after pictures. Additional descriptions and clarifications is presented in section 4.2 and section 4.3.

4.4.1 Detection Algorithms

The detection algorithm results will be presented as a gray scale image of the detected pixels. Since rain is a dynamic phenomenon three different precipitation scenarios will be presented. The classifications are as following: light, moderate and heavy precipitation. For each algorithm the same detection frame of each video will be shown.

An exception was made for the chromatic algorithm as its purpose changed during the thesis from detecting rain to detecting all moving objects except rain.

Original images

Figure 4.21: These are the original images in which the rain is detected from. The two on the top are from [1]. The one in the bottom is from [9].

Frequency

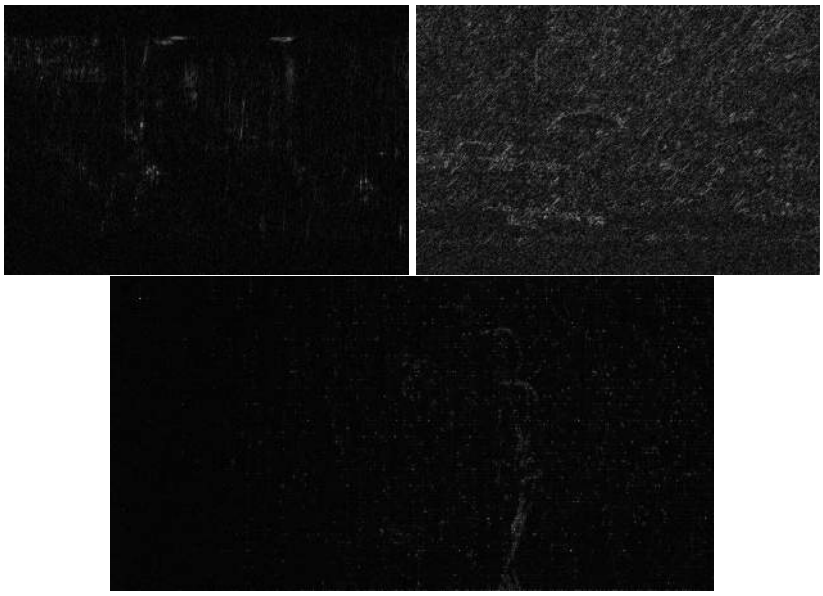

Figure 4.22: Rain detected by frequency algorithm with different parameters manually fitted since camera parameters were unknown.

Skewness

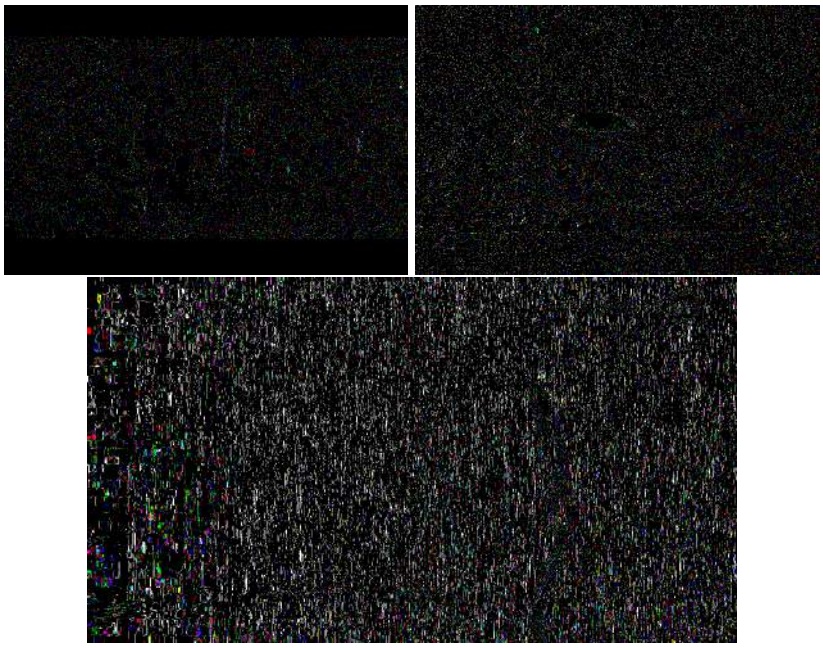

Figure 4.23: Rain detected by the skewness algorithm. The thresholds are set such that all the pixels with intensity higher than 90% or lower than 60% of the maximum intensity will be zero.

Intensity

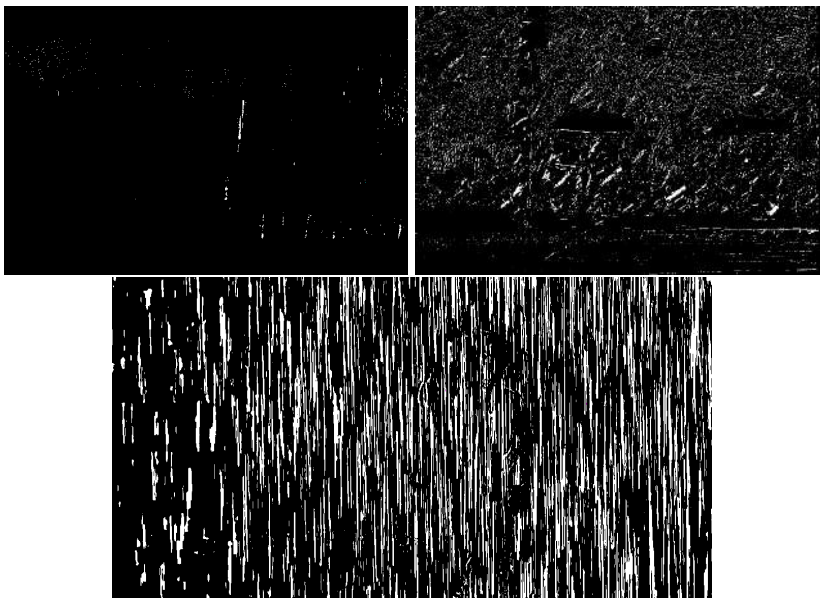

Figure 4.24: The outputs from the detection algorithm using the intensity approach.

Chromatic

Figure 4.25: Showing the detection of the chromatic algorithm. The red moving car and the moving reflection of the headlights are detected while the stationary blue truck remains undetected.

Morphological

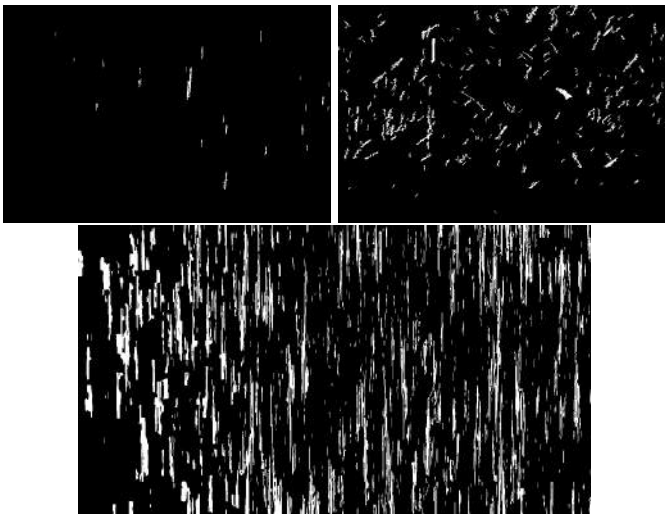

Figure 4.26: The morphological properties are used to reject clusters of pixels not fitting in size, aspect ratio and rotiation. Using morphological features to do this can not introduce false detections to the detection matrix.

4.4.2 Restoration algorithms

The different restoration algorithms implemented all use temporal data. This means that the rain will be replaced with a blend of the color from the two previous frames where rain

is not present. This approach is used by the mean and median replacement algorithm. The subtraction algorithm simply removes the additional pixel intensity increment introduced by the rain.

Their distinctions is mostly visible in heavy rain conditions.

Mean

Figure 4.27: The figure to the left shows the original scene. The scene to the right shows the scene with reduced precipitation. The precipitation is removed by taking the mean of three images. This scene contains light amount of precipitation.

Figure 4.28: The same approach to remove precipitation as in figure 4.27 is used. This scene contains medium amount of precipitation.

Figure 4.29: The same approach to remove precipitation as in figure 4.27 is used. This scene contains heavy amount of precipitation.

Median

The brightness of the scene does not change drastically between the frames. Therefore the results of the median as a replacement method will be similar to the result of the mean replacement method. Se section 5.3 for the deeper discussion.

Subtraction

Figure 4.30: The figure to the left shows the original scene. The figure to the right shows the scene with reduced precipitation. The restoration algorithm for removing the precipitation is subtraction. This scene contains light amount of precipitation

Figure 4.31: The same approach to remove precipitation as in figure 4.30 is used. This scene contains medium amount of precipitation.

Figure 4.32: The same approach to remove precipitation as in figure 4.30 is used. This scene contains heavy amount of precipitation.

Chapter 5
Discussion & conclusions

This chapter will discuss some important topics more thoroughly. Conclusion about the thesis will also be drawn in this chapter.

5.1 Detection quality

It was found that across the algorithms, often there was a set of variables which was in need to be adjusted with regards to not only the particular camera's properties but also the particular rain fall's properties. Algorithms where parameters need to be changed to achieve best results get a lot of added complexity when the reason for the change is in the rain itself.

When experimenting with the algorithms there has in every case been a trade off between rain removed and amount of false detections. The ultimate goal of 100% precision and 100% sensitivity has always been far fetched.

One aspect to acknowledge is that the statistics presented in the evaluation does only give an indication of the quality of the algorithm. What matters in the end is what the human eyes can see. Movements being disturbed by false detections have been observed. This must be avoided. If a human observer does not see the erroneous distortions made by our algorithms they are not severe. As the context is surveillance cameras, the event of frames being analyzed one by one has to be considered as well.

The statistics pulled from the evaluation is also biasing the result by the rain streak areas being large in the manual detection matrix. This would benefit a less accurate detection of the raindrops as long as they remained in the marked areas. A problem of the detection is that they often mark too little of each rain streak which still gives the impression of a weak rain streak in the resulting frame.

Refering back to the prioritization in section 3.5, the results of many algorithms are modest with regards to detection. Both frequency and skewness algorithms have not proved themselves useful and their time consumption is longer than the other algorithms.

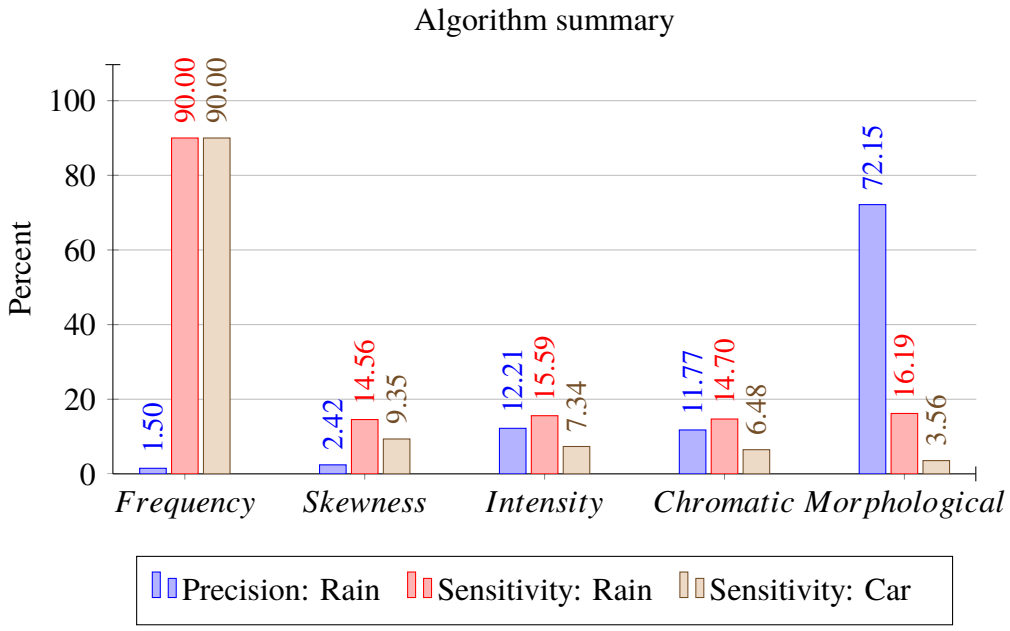

Figure 5.1: A comparison of the different algorithms. Chromatic and morphological are applied to an intensity detection. The third value is the sensitivity in the case of moving objects, here it is good to have a low value whereas the normal precision and sensitivity should be as high as possible.

Therefore they are less suitable for an environment which requires 30 frames per second. Intensity algorithm coupled with morphological or chromatic shows more promise.

5.2 Real-time constraint

If a surveillance camera supplies 30 frames per second, the additional time consumption introduced the precipitation removal must be less than 30ms. Otherwise lagging might be introduced.

No time have been spent optimizing the algorithms. They were implemented in a straightforward way. This results in storing large images in memory. This can be avoided if the work is done over a part of the image at a time, instead of the entire image at once. The values presented in table 5.1 can be used to compare the algorithms processing time between each other and provide an idea of their time consumption in relation to each other.

The larger the image, the more processing time will it require due to the amount pf pixels in the image. This is because larger images contains more pixels to be processed. In our case the resolution of the tested images were 720x480 (Section 4.4) and 1650x800 (Section 4).

Another caveat of some of the algorithms are that they introduce one or two frames of lag because they need the information from these other frames. This is never desirable when it comes to surveillance cameras no matter the effect achieved. This can be solved by taking the information from previous frames if they are stored in the memory.

In the case of the video used in the medium precipitation case only detection and re-

Detection algorithms - Time consumption

Algorithm	Time (ms)
Frequency	8000
Skewness	375
Intensity	40-60
Chromatic	90
Aspect ratio	1000 - 21000

Table 5.1: Table showing approximately how much time each detection algorithm need in order to produce the detection frame. The resolution is 1650x800 pixels. Time consumption for intensity and aspect ratio varies, depending on how much precipitation being detect. This was done on a computer with a Intel(R) Core(TM) i7-3770 CPU 3.40GHz processor

placement or removal was not sufficient. Color neutralization and blurring were also required in order to remove the artifacts introduced by the precipitation removal algorithm. This of course also affects the processing time. The processing time for one image in this case was 56.2 ms. By applying color neutralization and blurring the processing time increased to 86.1 ms. This means that the additional quality enhancing color neutralization and blurring costs 86.1 - 56.2 = 29.9 ms per image. This applies to images with the resolution 720x480 in Section 4.4.

In the case of the video used in the heavy precipitation case only detection and replacement or removal was sufficient.

In table 5.1 the results, which gives an overview over how much time is needed for the different detection algorithms, are presented.

5.2.1 Detection frame - color or gray scale

When using the intensity approach one can get the detection frame as a color image or a gray scale image. The difference between them is that the color image has its RGB channel preserved while the gray scale image has replaced its RGB channel with a mean of its RGB channels.

The difference between a color detection frame and a gray scale detection frame is that in a color detection frame how much the pixel intensities change in every color channel van be seen. This can be used in order to identify the false detections and reduce them.

In a gray scale detection frame on the other hand this level of detail can not be achieved. That which can be achieved though is that one can detect whether there is precipitation or not.

The level of detail which a color detection frame provides can be used in order to remove false detections. If the color of the false detection does not match with the color of precipitation, the false detection can be filtered by a threshold. By filtering, the false detection can be reduced or even removed. As a result, any false detection which can be identified by a threshold can be removed by color neutralization. For an example, see section 5.2.2.

5.2.2 Reducing false detections

If the detection algorithm falsely detects any objects whose colors are not white and the detection frame is in color, then color neutralization can be used in order to remove the false detections.

Take the rightmost figure in figure 4.19. It can be seen that to the left of the real bird there is a faintly visible black bird. This black bird does not exist in reality and is falsely introduced due to false detection.

By using color neutralization with a threshold of 0.2 the following reduced false detection is obtained in figure 5.2.

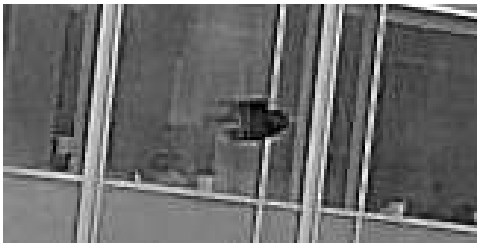

Figure 5.2: The rightmost figure in figure 4.19 after color neutralization. Note that to the left of the bird the falsely introduced bird is reduced. Now it is no longer clearly visible as in figure 4.20.

A disadvantage with color neutralization is that the threshold needs to be set based on the color of the false detection. If the threshold is badly chosen then color neutralization could end up reducing the amount of precipitation being removed.

5.3 Restoration quality

If perfect rain detection is assumed then the mean, median and subtraction approach performs equally well on streaks over static background. They also perform similarly when removing streaks in front of moving objects due to their small usage of past frames.

Another difference between the restoration algorithms is that while the mean and median approach requires at least three images in order to get a good replacement, the subtraction approach only requires two images.

In the example demonstrated in this thesis the mean and median approach for replacing the precipitation affected pixels yielded very similar results. The reason is because the brightness for the frames used did not drastically change.

But if the brightness would change very drastically then the difference of the median and mean replacements approach would be more evident.

5.3.1 Limitations with taking the mean as the correct value

In section 4.3.2 the mean of 100 images is taken. Then all the values are summed up and divided with the number of pixels to get a reference value for what the correct color would

be if the precipitation were not present.

A limitation of this approach is that if there is a moving object which occupies a majority of these 100 frames, then this moving object will be smeared out on the resulting frame. This will yield an incorrect value.

In our case the area around the raindrops are rather static. Therefore taking the mean of 100 frames can be used as an approach to find the correct value. The correct value refers to the correct pixel intensity if the raindrop were not covering the pixels.

5.4 Conclusions

The purpose of our work has been to relieve the viewer or image processing algorithms from the specific noise the precipitation introduces. This hypothesis involved a subjective judgement of a video which could be difficult to objectively assess. During the thesis we have watched a lot of videos with rain removed and under some circumstances the algorithms do suppress the rain effect satisfactory, but this is not enough as an algorithm must work well in many cases, preferably in all cases.

The false removal introduced by really aggressive rain removal algorithms could pose a problem if one attempts to analyze every detail in the image. If this is required then it is recommended that one should not remove the rain as aggressively by setting a higher threshold.

If it is a video for watching then the possible noise being introduced might not even be noticeable by the viewer due to the high frame rate.

In the algorithms implemented and tested in this thesis we found that the intensity algorithm coupled with morphological filter came closest to fulfilling the requirements.

With regards to the time aspect several algorithms struggle. Especially frequency and aspect ratio with certain settings takes to long time. Frequency needs even more computations (scalar and orientation approximation) to be applicable therefor we feel that it is not a candidate in real-time context with current computational power.

The morphological algorithms' time consumption grows quickly under the wrong circumstances and this needs to be contained. If it can be optimized enough it removes a lot of false detections and should be considered.

5.5 Improvements & Future work

In this section improvements for the algorithms are proposed. To find out if the algorithms are truly applicable in an actual camera further evaluation is required. In particular, the algorithm should be implemented and evaluated in a camera.

Automatically determine what the optimal parameters used in our algorithms should be set to in real-time if no setting is found to be good enough all around. The intensity threshold could e.g. maybe depend on the average intensity in the scene. The algorithms are different and some are more in need of this than others. Optimal parameters are parameters that will enable the rain removal algorithm to remove a lot more rain without falsely removing anything.

We did not have time to investigate if the rain suppression actually do help to avoid false triggers in motion detection algorithms, it depends on their sensitivity and the ferocity of the rainfall. A proposed work is to evaluate if precipitation would falsely trigger motion detection or hinder object detection.

Automatically detect if the scene contains rain or not. This will enable a clever use such that the rain removal algorithms are only activated and used when the scene contains rain and shut of when the scene is rain-free.

Our replacement algorithms are very similar. The method called Inpainting do not use the temporal aspect and replaces regions based on smart algorithms drawing information from nearby spatial information.

Bibliography

[1] Peter Barnum. The personal webpage of peter barnum, April 2014. http://www.cs.cmu.edu/~pbarnum/rain/rainAndSnow.html, acessed on 2014-05-12.

[2] Peter C. Barnum and Srinivasa Narasimhan. Analysis of rain and snow in frequency space, Dec 2008. http://www.cs.cmu.edu/~ILIM/publications/PDFs/BKN-IJCV09.pdf, acessed on 2014-05-12.

[3] Nathan Brewer and Nianjun Liu. Using the shape characteristics of rain to identify and remove rain from video, Dec 2008. http://www.nicta.com.au/pub?doc=1390, acessed on 2014-05-12.

[4] Xiaopeng Zhang. Et al. Rain removal in video by combining temporal and chromatic properties, 2006. http://www.comp.nus.edu.sg/~leowwk/papers/icme2006.pdf, acessed on 2014-05-12.

[5] Kshitiz Garg and Shree K. Nayar. When does a camera see rain?, 2005. http://pdf.aminer.org/000/293/101/when_does_a_camera_see_rain.pdf, acessed on 2014-05-12.

[6] Kshitiz Garg and Shree K. Nayar. Vision and rain, April 2006. http://www1.cs.columbia.edu/CAVE/publications/pdfs/Garg_IJCV07.pdf, acessed on 2014-05-12.

[7] Sajitha Krishnan and D.Venkataraman. Restoration of video by removing rain, April 2012. http://airccse.org/journal/ijcsea/papers/2212ijcsea02.pdf, acessed on 2014-05-12.

[8] Rong Deng Shuai Fang Ming Zhou, Zhichao Zhu. Rain detection and removal of sequential images, April 2011. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5968255&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5946174%2F5968129%2F05968255.pdf%3Farnumber%3D5968255, acessed on 2014-05-12.

[9] VernissageTv. Rain room by random international at moma new york, April 2014. http://www.youtube.com/watch?v=sktrJ8R0Wzo, acessed on 2014-05-12.