博士論文

Detection and Removal of Raindrop Images in A Video Sequence and Their Applications to Computer Vision Algorithms

（ビデオシークエンス中の付着雨滴像の検出、除去
並びにそれのコンピュータビジョンアルゴリズムへの応用）

YOU，SHAODI

尤　少迪

A DOCTORAL DISSERTATION

東京大学
THE UNIVERSITY OF TOKYO

June 2015

Thesis Supervisor: Oishi TAKESHI　大石　岳史

ABSTRACT

Raindrops appeared on windscreens or window glass can degrade the visibility of the outside scenes. If we can detect and later remove the raindrops, many applications such as intelligent vehicle system will benefit from it.

In this thesis, we intend to focus on developing methods of automatic raindrop detection and removal. And we utilizes raindrop to perform a few image processing tasks. To achieve these goals, we have theoretically analyzed the imaging system with the presence of water drops. Based on our analysis, we have developed three automatic raindrop detection and removal systems. Further more, with the insights on properties of water drops, we developed an single image stereo system using water drops.

The first system utilizes the assumption of the smooth motion of camera/scene. The idea is to use long range trajectories to discover the motion and appearance features of raindrops locally along the trajectories. These motion and appearance features are obtained through our analysis of the trajectory behavior when encountering raindrops. These features are then transformed into a labeling problem, which the cost function can be optimized efficiently. Having detected raindrops, the removal is achieved by utilizing patches indicated, enabling the motion consistency to be preserved. Our trajectory based video completion method not only removes the raindrops but also complete the motion field, which benefits motion estimation algorithms to possibly work in rainy scenes. Experimental results on real videos show the effectiveness of the proposed method.

The second system is also based on the smooth motion which is a fast and robust method. In comparing with the first method, this method aims to speed up the video restoration by replacing the dense motion estimation with sparse motion and interpolation. In the situations that the motion is smooth and stable, the restoration quality is comparable with the first method. It is principally based on sparse matching and interpolation. First, SIFT, which is robust to arbitrary motion, is used to efficiently obtain sparse correspondences in neighboring frames. To ensure these correspondences are uniformly distributed across the image, a fast dense point sampling method is applied. Then, a dense motion field is generated by interpolating the correspondences. An efficient weighted explicit polynomial fitting method is proposed to achieve spatially and temporally coherent interpolation. In the experiment, quantitative measurements were conducted to show the robustness and effectiveness of the proposed method.

The third system is based on the contraction properties of water drops. The core idea is to exploit the local spatio-temporal derivatives of raindrops. First, we explicitly

model adherent raindrops using law of physics, and then, detect them based on these models in combination with motion and intensity temporal derivatives of the input video. Second, relying on an analysis that some areas of a raindrop completely occludes the scene, yet the remaining areas occlude only partially, we remove the two types of areas separately. For partially occluding areas, we restore them by retrieving as much as possible information of the scene, namely, by solving a blending function on the detected partially occluding areas using the temporal intensity derivative. For completely occluding areas, we recover them by using a video completion technique. Experimental results using various real videos show the effectiveness of the proposed method.

Based on the experience on detecting and removing rain drops, we find that water drops are not always noise in image/video but can be used to perform a variety of vision tasks. Therefore, we propose a novel single image stereo system, which utilizes a common camera with a few water drops. The key idea is that a single water drop adhered to window glass is totally transparent and convex, and thus can be considered as a fisheye lens. If we have more than one water drop in a single image, then through each of them we can see the environment with different view points, similar to stereo. To accomplish this idea, foremost, we need to rectify every water drop imagery to make distorted planar surfaces look flat. For this, we consider two physical properties of water drops: (1) a static water drop has constant volume, and its geometric convex shape is determined by the balance between the tension force and gravity. In other words, the geometric shape can be obtained by minimizing the overall potential energy, which is the sum of the tension energy and the gravitational potential energy. (2) The imagery inside a water-drop is determined by water-drop geometric shape and total reflection at the boundary. This total reflection generates a dark band commonly observable in any adherent water drops. Once the geometry of water drops recovered, we rectify the drop images through ray-tracing. Based on a set of the rectified images of water drops, we can compute depth using the concept of stereo. In addition, we can also refocus the whole input image. Experiments on real images and a quantitative evaluation show the effectiveness of our proposed method. To our best knowledge, never before have adherent water drops been used to estimate depth.

Acknowledgements

First and foremost, I would like to express my gratitude to my advisor, Prof. Katsushi Ikeuchi, for his supervision. He shares his vast knowledge and experience and insightful thoughts through the advising. While keeping me in track, he also gives freedom to follow my own interests. Other than that, we have also learned from him of academic cooperations and career planning as a researcher.

I would thank Prof. Takeshi Oishi for his kind support on all my admissions. Although his is not directly advising my research, I did learn a lot from him through academic events and collaborations.

I would like to express my gratitude to: Dr. Robby T. Tan at Yale-NUS, Singapore. Dr. Tan has been advising me since my master's. Dr. Tan uses to be very strict but patient, I have learned from him in every detailed aspects of doing research. He has intensively contributed to almost every part of this thesis. In addition I have also learned a lot from him in finding the personal values in becoming a researcher.

I would also express my gratitude to Dr. Rei Kawakami at The University of Tokyo. Dr. Kawakami also shares her vast knowledge and experience in research. She gives me suggestions in research life. As a native, she also helps me a lot in Japanese culture.

Many thanks goes to Prof. Yasuhiro Mukaigawa at NAIST. As a contributor to this thesis, Prof. Mukaigawa gives me a lot of insights on modeling the transparent objects.

During my internship in Microsoft, I have also learned a lot during my cooperating with Dr. Yasuyuki Matushita.

Many thanks go to the people that I have been working with, the current and former members of the Computer Vision Laboratory at the University of Tokyo. Special thanks go to Dr. Shintaro Ono, who gives me suggestions and support in Tohoku project; Boxin Shi, who gives me a lot of suggestions as a senior. I am also very proud of working with

all the Photometry group members. Although, due to limited space, I cannot name everyone who has helped me, I am very grateful to all the people I have met in this lab.

Finally, I would like to thank my family and my friends for constant support in all my life.

June 2015
Shaodi You

Contents

Abstract **i**

Acknowledgment **iii**

List of Figures **vii**

List of Tables **xiii**

1 Introduction **1**

1.1 Background 1
1.2 Motivation and Goals 3
1.3 Related Works 5
1.3.1 Bad Weather Visibility Enhancement 5
1.3.2 Sensor/lens Dust Removal 5
1.3.3 Adherent Raindrop Detection and Removal 5
1.3.4 Video Completion 6
1.3.5 3D Reconstruction from a single image 6
1.3.6 Modeling of water 7
1.3.7 Transparency object modeling 7
1.4 Approaches to Achieve The Goal 8
1.5 Contributions 9
1.6 Thesis Overview 10

2 Modeling **13**

2.1 Modeling of Raindrops 15
2.1.1 Size 15
2.1.2 Shape 15
2.1.3 Minimum Energy Surface 18
2.1.4 Water-Drop Volume from Dark Ring 21
2.1.5 Dynamics 25

2.1.6 Distribution. 25
2.2 Modeling of Camera 27
2.2.1 Clear Raindrop Imagery 27
2.2.2 Blurred Raindrop Imagery 36
2.3 Modeling of Environment 42
2.4 Summary and Discussion 44

3 Long Range Trajectories Based Methods 47
3.1 Raindrop Detection and Removal from Long Range Trajectory. 48
3.1.1 Related Work 50
3.1.2 Trajectory Analysis 52
3.1.3 Raindrop Detection 62
3.1.4 Raindrop Removal 64
3.1.5 Experiments 66
3.1.6 Conclusion and Future Work 84
3.2 Robust and Fast Motion Estimation for Video Completion 85
3.2.1 Robust Sparse Matching 88
3.2.2 Fast Space-time Motion Interpolation 90
3.2.3 Experiments 92
3.2.4 Conclusion 103

4 Blend-in Model Based Method 105
4.1 Raindrop Detection 105
4.1.1 Feature Extraction 105
4.1.2 Refined Detection 106
4.1.3 Real Time Detection 108
4.2 Raindrop Removal and Image Restoration 110
4.2.1 Restoration 110
4.2.2 Video Completion 112
4.3 Experiments and Applications 113
4.3.1 Quantitative analysis on detection 113
4.3.2 Quantitative Comparison on Detection 123
4.3.3 Raindrop Removal 124
4.3.4 Applications 124
4.4 Summary 142

5 Single Image Stereo Using Water Drops 143
5.1 Image Formation 144
5.2 Methodology 147
5.2.1 Water Drop Detection 147
5.2.2 Water-Drop 3D Shape Reconstruction 149
5.2.3 Rectification of Water-Drop Image 153
5.2.4 Depth from Stereo 153
5.3 Experiments and Analysis 156
5.3.1 3D Shape Reconstruction and Image Rectification 156
5.3.2 Depth Estimation 156
5.4 Discussion 167

6 Discussion and Conclusion 169
6.1 Summary 169
6.2 Contributions 172
6.2.1 Applications 172
6.2.2 Relation with Neural Network 173

List of Figures

1.1 Examples of video taken in a rainy day during the digital archiving of the 2011 Japan Earthquake. 2
1.2 (a-e) The various appearances of raindrops. (e-f) The detection and removal result by our method. 4
1.3 An overview of the thesis. 11

2.1 The imagery model of the rainy scenes. 14
2.2 Balance at a raindrop surface 16
2.3 Smoothness and roundness of some shapes. 17
2.4 Parameters of a water drop. 18
2.5 Minimum energy surfaces with given the area and volume. 20
2.6 Refraction in a water drop. 22
2.7 Indicate dark ring from water drop geometry. 24
2.8 Raindrop dynamic of scenes in Chapter 4. 25
2.9 Adherent raindrops in different distributions. 26
2.10 Raindrop imagery formation. 28
2.11 Refraction model of a pair of corresponding points on an image plane. . 29
2.12 Local linear space 32
2.13 Simplified refraction model of the second refraction using principle curvature. 33
2.14 Observing the expansion ratio in x and y direction on real data. 35
2.15 The light path model on an image plane collecting light 38
2.16 Raindrop-plane-cut of the light path model 39
2.17 The appearance various from light path. 40
2.18 Raindrop appearance varying with aperture and direction. 41
2.19 Spatio-temporal space and dense trajectories. 43

3.1 An example of the results of our proposed detection and removal method. 49
3.2 The pipeline of our method. 50
3.3 Model of Raindrop Imaging System 53

3.4 Video in rainy scenes and events on the trajectories. 54
3.5 Ambiguity of correspondences 56
3.6 Appearance of trajectories in Fig. 3.4. 58
3.7 Raindrop features. 60
3.8 Raindrop detection via labeling. 62
3.9 The raindrop detection results using our method and the existing methods on synthetic data. Data 1: thick raindrops, car mounted camera. . . 67
3.10 The raindrop detection results using our method and the existing methods on synthetic data. Data 2: thin raindrops, surveillance camera. . . . 68
3.11 The raindrop detection results using our method and the existing methods on synthetic data. Data 3: thick and thin raindrops, car mounted camera. 69
3.12 The raindrop detection results using our method and the existing methods on synthetic data. Data 4: thick and thin raindrops, hand held camera. 70
3.13 The raindrop detection results using our method and the existing methods on real data. Data 5: thick and thin raindrops, hand held camera. . 71
3.14 The raindrop detection results using our method and the existing methods on real data. Data 6: thin raindrops, car mounted camera. 72
3.15 The raindrop detection results using our method and the existing methods on real data. Data 7: thick raindrops with glare, hand held camera. 73
3.16 The raindrop detection results using our method and the existing methods on real data. Data 8: thin raindrops with glare, car mounted camera. 74
3.17 Precision-recall curve on detection for the methods shown in Fig. 3.9. Evaluation at a pixel level. 75
3.18 Precision-recall curve on detection for the methods shown in Fig. 3.9. Evaluation at number of raindrops level. 76
3.19 The raindrop removal results. Data 0: thick raindrops. 79
3.20 The raindrop removal results. Data 1: thick raindrops. 80
3.21 The raindrop removal results. Data 2: thin raindrops. 81
3.22 The raindrop removal results. Data 3: thick and thin raindrops. 82
3.23 Comparison on motion field estimation before and after raindrop removal. 83
3.24 Video completion using the proposed method. 86
3.25 The proposed motion estimation method. 87

3.26 Video completion in fast moving area using our proposed methods and existing methods. 93
3.27 Video completion in slowly moving area using our proposed methods and existing methods. 94
3.28 Video completion in static area using our proposed methods and existing methods. 95
3.29 Raindrop removal on Tohoku data using our proposed methods and existing methods (I). 96
3.30 Raindrop removal on Tohoku data using our proposed methods and existing methods (II). 97
3.31 Two experiments on robust motion estimation. 100
3.32 Applications of video completion on logo removal. 101
3.33 Applications of video completion on raindrop removal. 102

4.1 The accumulated optic flow as a feature. 106
4.2 The accumulated intensity changes as a feature. 106
4.3 The detection pipeline. 107
4.4 Synthetic raindrops with various size and blur levels. 114
4.5 The precision and recall on detecting raindrops with various size and blur, evaluated at pixel level 115
4.6 The precision and recall on detecting raindrops with various size and blur, , evaluated at number of raindrops level 116
4.7 Appearance of synthetic moving raindrops. 118
4.8 The influence of number of frames on feature accumulation. 119
4.9 The precision on detecting raindrops with various raindrop speed and detection latency of Fig. 4.7 120
4.10 The recall on detecting raindrops with various raindrop speed and detection latency of Fig. 4.7. 121
4.11 Gaussian blur on a scene 122
4.12 The accumulated feature using intensity change and optic flow on textured and textureless scenes. 122
4.13 The precision and recall of raindrop detection on textured and textureless scenes. 123
4.14 The detection results of a night scene using our methods and the existing methods. 125

4.15 The detection results of raindrops with arbitrary shapes using our methods and the existing methods. 126
4.16 The detection results of raindrops with arbitrary size using our methods and the existing methods. 127
4.17 The detection results of raindrops with highlights using our methods and the existing methods. 128
4.18 The detection results of video taken by a car-mounted camera using our methods and the existing methods. 129
4.19 The detection results of video taken by a surveillance camera using our methods and the existing methods. 130
4.20 The precision(R)-recall(R) curves of our methods and the two existing methods. 131
4.21 Average $(R; G; B; dx; dy; dt)$ error 131
4.22 The raindrop removal results using our methods and the method of Wexler *et al*. [67] on a clear driving scene. 132
4.23 The raindrop removal results using our methods and the method of Wexler *et al*. [67] on a crowed driving scene. 133
4.24 The raindrop removal results using our methods and the method of Wexler *et al*. [67] on a textured scene. 134
4.25 The raindrop removal using our method on a video taken by hand-held camera. 135
4.26 The raindrop removal using our method on a video taken by a car-mounted camera. 136
4.27 Motion estimation using a clear video, raindrop video and repaired video on a synthetic data. 138
4.28 Motion estimation using a clear video, raindrop video and repaired video on a real video taken by a hand-held camera. 139
4.29 Motion estimation using a clear video, raindrop video and repaired video on a real video taken by a car-mounted camera. 140
4.30 Structure from motion using a clear video, raindrop video and repaired video. 141

5.1 The pipeline of the proposed method. 145
5.2 Model of the image system. 146
5.3 Selecting water drops from a single image. 148
5.4 Iteration of water drop 3D shape with a fixed volume. 150

5.5 Registration between the observed and estimated dark ring. 152
5.6 Rectified water drop images. 154
5.7 Illuminance compensation of water drop images. 155
5.8 Quantitative evaluation of water surface reconstruction and rectification assuming a round water drop. 157
5.9 Quantitative evaluation of water surface reconstruction and rectification assuming a eclipse water drop. 158
5.10 Quantitative evaluation of water surface reconstruction and rectification assuming a hanged water drop. 159
5.11 Quantitative evaluation of water surface reconstruction and rectification assume a irregular water drop. 160
5.12 Rectification of real water images using our data. 161
5.13 Rectification of real water images using data downloaded from the Internet. 162
5.14 Stereo using two dewarped water drop images. 163
5.15 Stereo using two rectified water drop images. 166

List of Tables

2.1 The methods and their replying properties 45

3.1 False alarms on Data 1-4 77
3.2 Comparison on average repairing error. 98
3.3 Average completion error 99
3.4 Average completion time per frame 99

5.1 Computation time for water drop 3D reconstruction with varying volume. 164
5.2 Computation time for water drop 3D reconstruction with varying mesh resolution. 164

6.1 The proposed methods and their applicabilities. 171

Chapter 1

Introduction

1.1 Background

Outdoor vision system is used for various tasks such as navigation and surveillance. It can be adversely affected by bad weather conditions. In a rainy day, it is inevitable that raindrops will appear on the windscreen, camera lens, or the protecting shield. These adherent raindrops will cause large area of data to be missing. Because of this, the performances of many algorithms of outdoor vision systems (such as feature detection, tracking, stereo correspondence, etc.) will be significantly degraded.

Especially, in order to digitally archive the 2011 Japan Earthquake, our lab uses car-mounted video camera to record street views in the earthquake area. Some of the video is taken in rainy day. Raindrops adhered to camera lens cause large area of data missing. Performance of pro-processing computer vision tasks such as object detection, image registration, video stabilization and frame interpolation are significantly degraded.

Figure 1.1: Examples of video taken in a rainy day during the digital archiving of the 2011 Japan Earthquake.

1.2 Motivation and Goals

To make the video with adherent raindrops visually satisfying and to enhance the performance of computer vision tasks, it is essential to remove the raindrops.

There are two main steps of adherent raindrops removal.:

1. Raindrops detection.
2. Video Repairing.

For raindrops detection, algorithms to automatically detect raindrops with any shape and size are demanded. Above that, efficient method which could detect raindrops in real-time or nearly real-time is demanded.

For video repairing, algorithm which could handle complex outdoor environment is demanded. The repairing algorithm should robustly repair video with both spatially large and temporally large data missing. The algorithm should work well on fast changing video, slowly changing video and static image. The video should work well on both textured and non-textured video. Above that, computational efficiency is also required for large scale data.

We find that water drops is not always noise in image/video by can be used to performance a variety of vision tasks, *e.g.*stereo. To achieve this goal, we need two steps:

1. Water drop geometry estimation
2. Drop image rectification.
3. Stereo

The 3D geometry estimation is not a trivial task, because a single image can only provide a 2D project. To achieve this goal we model the water drop as a minimum energy surface. To solve the minimum energy surface, we also need an efficient algorithm. Based on it, the drop image rectification and stereo can be performed.

Figure 1.2: (a-e) The various appearances of raindrops. (e-f) The detection and removal result by our method.

1.3 Related Works

1.3.1 Bad Weather Visibility Enhancement

Removing the influence of haze, mist, fog (e.g., [59, 11, 23, 39]), rain and snow (e.g., [3, 16]) have been well exploited. Dealing with rain, Garg and Nayar model rain streaks [15], and devise algorithms to detect and remove them [17, 16]. Later, Barnum *et al*. [3] propose a method to detect and remove both rain and snow. Single-image based methods are proposed by Kang *et al*. [30] and Chen *et al*. [7]. Unfortunately, applying these methods to handle adherent raindrops is not possible, since the physics and appearance of falling raindrops are significantly different from those of adherent raindrops.

1.3.2 Sensor/lens Dust Removal

Sensor dust removal is to some extent a related topic to raindrop detections. Willson *et al*. [68] give a detailed analysis on the imagery model with dust adhered to the lens. Dust blocks light reflected from objects and scatter/reflect light coming from the environment. The former is called a dark dust artifact, and the latter a bright dust artifact. Zhou and Lin [75] propose method to detect and remove small dark dust artifacts. Gu *et al*. [19] extend the solution to sufficiently blurred thin occluders. Although adherent raindrops can be considered as a kind of sensor dust, existing sensor dust removal methods cannot handle adherent raindrops, since raindrops can be large and are not as blurred as dust. Moreover, raindrop appearance significantly more varies than dust appearance.

1.3.3 Adherent Raindrop Detection and Removal

A few methods for detecting adherent raindrops have been proposed. Roser *et al*. attempt to model the shape of adherent raindrops by a sphere crown [46], and later, Bezier curves [47]. These models, however, are insufficient, since a sphere crown and Bezier curves can cover only a small portion of raindrop shapes. Kurihata *et al*. [32] and later Eigen *et al*. [10] approach the problem through machine learning. However, as shown in Fig. 1.2.a-d, collecting the training images for all various shapes, environment, illumination and blur are considerably challenging. Both of the methods are limited to detect small, clear and quasi-round rain spots. Yamashita *et al*. propose a detection

and removal method for videos taken by stereo [70] and pan-tilt [69] cameras. The methods utilize specific constraints from those cameras and are thus inapplicable for a single camera. Hara *et al*. [20] propose a method to remove glare caused by adherent raindrops by using a specifically designed optical shutter. As for raindrop removal, Roser and Geiger [46] address it using image registration, and Yamashita *et al*. [70, 69] utilize position and motion constraints from specific cameras.

1.3.4 Video Completion

Video completion has been intensively exploited by computer vision researchers. Only those methods work with large spatio-temporal missing areas can be used to remove detected adherent raindrops. Wexler *et al*. [67] propose an exemplar based inpainting method by assuming the missing data reappears elsewhere in the video. Jia *et al*. [27] exploit video completion by separating static background from moving foreground, and later [26] exploit video completion under cyclic motion. Sapiro and Bertalmio [50] complete the video under constrained camera motion. Shiratori *et al*. [52] and Liu *et al*. [35] first calculate the motion of the missing areas, and then complete the video according to the motion. Unfortunately, outdoor environments are too complex to satisfy static background, cyclic motion, constrained camera motion, etc. Therefore, we use cues from our adherent raindrop modeling to help the removal.

1.3.5 3D Reconstruction from a single image

Exisiting researches have explored the 3D reconstruction of opaque object from a single image. Because the problem is ill-posed, addition assumptions has been used to solve the problem. For example, shape from shading by [25], shape from texture by [38], shape from defocus by [12] and piece-wise planarity by [24]. Specifically, a few approaches has been proposed from shape from silhouette which aims to reconstruction a bounded smooth surface surface [60, 22, 44, 28, 42, 64]. Among which, [44] reconstruct the surface with minimum area and [42] proposed a speed-up version. However, none of the above method directly aim to model water or other transparent liquid from a single image.

1.3.6 Modeling of water

[16, 46, 71] propose methods on modeling air-borne or adherent rain drops. However, they considers raindrop as noise, 3D reconstruction are not discovered. [47] exploited fitting water drop surface using B-splines, however, they are only fitting the sihoulletes using 1D splines. A few researches has exploit the under water imaging [40, 61, 41, 29]. However, they assume the water surface are dynamic which are dominant by transition of waves and does not suit our problem.

1.3.7 Transparency object modeling

Stereo and light field using perspective camera with extra mirrors and lens has been explored in the society. For example, [2, 58] propose methods using sphere mirror; [33] propose method uses arrays of planar mirrors; [57, 45] propose the axial cameras. However, all the above methods assume the ridial or planar symetry of the media(mirror/lens) which are not satisfied in the case of water drops.

1.4 Approaches to Achieve The Goal

To achieve the above mentioned goals, we proposed a few approaches in this thesis.

Modeling First of all, we theoretically model the properties on the adherent water drops of its physical and imagery properties. Specifically, we model the 1. The environment; 2. The raindrop intrinsic parameters; and 3. The camera parameters.

Detection Based on the modeling, we propose three detection methods: 1. The long-range trajectory based method. 2. The contraction based method. 3. The edge detection based method. These methods will be introduced in Sec. 3, Sec. 4 and Sec. 5 correspondingly.

Raindrop Removal and Video Completion After the water drops are identified, we remove then and repair the video. Specifically, we propose three different methods on video completion. 1. The long-range trajectory based method, with will be introduced in Sec. 3.1. 2. The smooth camera motion based method which will be introduced in Sec. 3.2 and The blend-in model based method which will be introduced in Sec. 4.

Utilization of water drops To accomplish the idea on utilization of water drops for stereo, foremost, we need to rectify every water drop imagery to make distorted planar surfaces look flat. For this, we consider two physical properties of water drops: (1) a static water drop has constant volume, and its geometric convex shape is determined by the balance between the tension force and gravity. In other words, the geometric shape can be obtained by minimizing the overall potential energy, which is the sum of the tension energy and the gravitational potential energy. (2) The imagery inside a water-drop is determined by water-drop geometric shape and total reflection at the boundary. This total reflection generates a dark band commonly observable in any adherent water drops. Once the geometry of water drops recovered, we rectify the drop images through ray-tracing. Based on a set of the rectified images of water drops, we can compute depth using the concept of stereo. In addition, we can also refocus the whole input image.

1.5 Contributions

In this thesis, three complete algorithms to remove adherent raindrops in video are proposed. And an algorithm to perform single image stereo using water drop images.

For raindrops detection:

I. Algorithms which could detect raindrops with any size and shape are proposed.

II. Accuracy of our algorithm outperforms all existing algorithms.

III. Our proposed real-time computational efficiency which is essential for many outdoor vision tasks.

For video repairing:

I. Algorithm which could repair video with both spatially and temporally large missing area is proposed.

II. Case by case solution which could handle complex situations (complex motion and complex structure) in outdoor vision system is proposed.

III. Computational efficiency is achieved by using the proposed sparse matching based motion estimation.

For single image stereo:

I. A novel single image stereo method which utilized only water drops and a common 2D camera is proposed.

II. A novel single image liquid geometry estimation which utilized the miminum energy surface and total, reflection is proposed.

III The system enables more image processing tasks such as stereo and refocus.

1.6 Thesis Overview

Chapter 2 is the modeling of the imaging system and raindrops. The image system is based on three parts: 1. The environment; 2. The raindrop intrinsic parameters; and 3. The camera parameters. In the first section, we model the intrinsic properties of raindrops: those properties which are not depending on the environment or the camera setting. In the second section, we model the properties of raindrop imaging which are relying on the camera setting. In the third section, we model the properties of raindrop imaging which are depending on the environment.

Chapter 3 describes the trajectory based methods, which is replying on our modeling of the smooth motion from the camera and environment. There are two methods relying on the modeling. The first one is: Raindrop Detection and Removal from Long Range Trajectory. [73], which will be introduced in the first sub-chapter and the second method is: Robust and Fast Motion Estimation for Video Completion [72], which will be introduced in the second sub-chapter.

Chapter 4 introduces the method using the assumption which is relative more depending on the raindrop intrinsic modeling, says, the blend-in modeling. As introduced in Chapter 2.2, using the blend-in model, the proposed method does not need to assume the camera undergoes a smooth motion. The method is appeared in [71].

Chapter 5 introduces the methodology single image stereo system which utilizes a common camera with a few water drops. And applications on stereo, refocus are also introduced.

Chapter 6 is the discussion and conclusion.

1.3 is an illustration of the overview of the thesis.

Figure 1.3: An overview of the thesis.

Chapter 2

Modeling

Before proposing methods to detect and remove adherent raindrops in video. In this chapter, we explicitly model the imaging system in rainy scene.

Figure 2 illustrates the imagery model of the rainy scenes. As can be seem, the raindrop appearance is based on three parts: 1. The environment; 2. The raindrop intrinsic parameters; and 3. The camera parameters.

In the first section, we model the intrinsic properties of raindrops: those properties which are not depending on the environment or the camera setting.

In the second section, we model the properties of raindrop imaging which are relying on the camera setting.

In the third section, we model the properties of raindrop imaging which are depending on the environment.

The last section is the summary.

Figure 2.1: The imagery model of the rainy scenes.
The raindrop appearance is based on three parts: 1. The environment; 2. The raindrop intrinsic parameters; and 3. The camera parameters..

2.1 Modeling of Raindrops

In this section, we model the intrinsic properties of raindrops: those properties which are not depending on the environment or the camera setting. Specifically, we model the raindrop size, shape and dynamics.

2.1.1 Size

Unlike estimating the size of airborne raindrops, which is mentioned in the work of Garg *et al*. [16], estimating the size of adherent raindrops is not trivial. Since it depends on the gravity, water-water surface tensor, water-adhering-surface tensor and many other parameters.

Fortunately, it is possible to set an upper bound of the size by using few parameters. As illustrated in Fig. 2.2.a, to prevent raindrops from sliding down, both the two-phase point (water-air) and three-phase points (water-air-material), the surface tensor should balance the pressure. This also prevents the water drop from breaking down. Although estimating the balance and upper boundary of the three phase points is intractable due to the unknown parameters of the material, estimating the balance and upper bound of two-phase point has been studied by physicists, and can be used to derive an upper bound of raindrop size, i.e., 5mm [65].

2.1.2 Shape

Although most existing methods assume the shape of raindrops to be circle or ellipse, the real raindrop shape varies in a large range. Despite this, however, we can still find some regular patterns due to the surface tensor. Raindrop boundaries are smooth and raindrops are convex in most cases. Hence, we can quantitatively characterize raindrop shape using two features: shape smoothness and roundness. As illustrated in Fig. 2.2.b, given a raindrop area on the image plane, denoted as R, we can integrate the change of the tangent angle along the boundary. The integration is denoted as $\mathcal{S}(R)$:

$$\mathcal{S}(R) = \oint_{\mathbf{x}\in\partial R} |\mathrm{d}\theta(\mathbf{x})|, \tag{2.1}$$

where ∂R is the boundary of the raindrop, and $\mathbf{x} = (x, y)$ is the the $2D$ coordinates on the image plane. For convex shape, $\mathcal{S}(R) \equiv 2\pi$. For non-convex or zig-zag shape, the smoothness will be greater than 2π. Fig. 2.3 shows some examples.

Figure 2.2: Balance at a raindrop surface

(a) Balance at a raindrop surface. A denotes a two-phase point. B denotes a three-phase point. **T** denotes a surface tensor, and **P** for pressure. At two-phase point A, surface tensor **T** and pressure **P** are balanced. Three-phase point is an intersection of water, air and glass, while two-phase point is an intersection between air-water. (b) Change of the angle of tangent along a raindrop boundary.

Shape						
Smoothness	$2\pi(6.28)$	$2\pi(6.28)$	$3\pi(9.42)$	11.10	9.41	54.15
Roundness	$1/4\pi(0.080)$	0.075	0.050	0.029	0.058	0.016

Figure 2.3: Smoothness and roundness of some shapes.

Roundness, denoted as $O(R)$, is the area of the shape divided by the square of its perimeter:

$$O(R) = \frac{\iint_{\mathbf{x}\in R} \mathrm{d}x\mathrm{d}y}{\left(\oint_{\mathbf{x}\in\partial R} |\mathrm{d}\mathbf{x}|\right)^2}. \tag{2.2}$$

A rounder shape has a larger roundness value and a perfect circle has the maximum roundness value: $\frac{\pi r^2}{(2\pi r)^2} = \frac{1}{4\pi} = 0.080$. Fig. 2.3 shows some examples. Both the smoothness and roundness are invariant to scaling and rotation. Unlike our previous method [71], which used the roundness, our current method employs smoothness. This is because the computational complexity of roundness is $O(n^2)$ while smoothness is $O(n)$.

2.1.3 Minimum Energy Surface

Figure 2.4: Parameters of a water drop.
(a) In the global coordinates, the geomertic shape is determined by the gravity and the surface tension. (b) The parameters in the camera coordinates. Point A is a two phase point (water-air), where the tensor force balances the pressure. Point B is a three phase point (water-air-material).

We explore the minimum energy surface to estimate the 3D shape of a water drop. We first introduce a local coordinate system determined by the camera, which is illustrated in Fig. 5.2.a and d. In this coordinates, the water drop 3D shape can be parameterized as:

$$\mathcal{S} = \{z(x, y), (x, y) \in \Omega_R\} \tag{2.3}$$

where Ω_R indicate the raindrop area attached to glass. (x, y) is any point in the attachment area and z is the height.

A static water drop has a constant volume, and its 3D shape $\tilde{\mathcal{S}}$ minimizes the overall potential energy $\mathbf{E}$, which can be written as:

$$\begin{aligned} \mathbf{E}(\tilde{\mathcal{S}}) &= \min \mathbf{E}(\mathcal{S}) = \min(\mathbf{E}_T(\mathcal{S}) + \mathbf{E}_G(\mathcal{S})), \\ V(\mathcal{S}) &= constant \end{aligned} \tag{2.4}$$

where $\mathbf{E}_T$ is the tension energy, and $\mathbf{E}_G$ is the gravitational potension energy, V is the volume. Therefore, to solve the geometry of a raindrop, we need to find the surface $\tilde{\mathcal{S}}$ which minimizes the overall potential energy with the constraints of constant volume. Fig. 2.4 illustrate the 3D shape of a water drop. Point A is a two-phase (water-air) balanced point, where surface tension $\mathbf{T}$ balances pressure $\mathbf{P}$. Point B is a three

phase point (water-air-material), where the tension is from both the water, $\mathbf{T}_w$, and the adhesion surface, $\mathbf{T}_m$. These two types of tension balance the gravity, $\mathbf{G}$.

With the parameterized surface, we can write the surface tension energy as:

$$E_T(\mathcal{S}) = \int_{\Omega_R} \sigma \mathrm{dA} = \int_{\Omega_R} \sigma \sqrt{1 + |\nabla z|^2} \mathrm{dxdy}, \tag{2.5}$$

where σ is the surface tension index for water, dA denotes a unit surface area and ∇ is the gradient [13]. As we can see, the tension energy is proportional to the area of the surface.

The gravitational potentional energy can be expressed as:

$$E_G(\mathcal{S}) = \int_{\Omega_R} \mathrm{dxdy} \int_0^z (x \cos\theta_x + y \cos\theta_y + z \cos\theta_z) g\rho \mathrm{dw}, \tag{2.6}$$

where θ_x, θ_y and θ_z denotes the angle between the x, y, z coordinates and the gravity correspondingly. g is the gravity and ρ is the density of water, which are generally known. Moreover, we can add a constraint that:

$$\int_{\Omega_R} z \mathrm{dxdy} \equiv V. \tag{2.7}$$

Therefore, the parameterized surface S is estimated by minimizing the over potential energy determined by Eq. (2.4), (2.5) and (2.6) with the constraints of constant volume in Eq. (2.7). We will discuss the detailed algorithm in Section 4. Figure 2.5 shows examples of the surface found by using the technique.

To uniquely determine the geometry of a water drop, we need to know both the 2D area where the water drop attached to glass, Ω_R, and the volume V. While the former can be directly inferred from the image, the latter is not straightforward to obtain. The subsequent section will discuss how we can possibly determine the volume.

(a) Area of water attached to the adhesion material

(b) Minimum energy surface with a small constant volume

(c) Minimum energy surface with a greater constant volume

Figure 2.5: Minimum energy surfaces with given the area and volume.

2.1.4 Water-Drop Volume from Dark Ring

The basic idea of our volume estimation is based on the dark ring at the boundary of a water drop. We found that the wider the dark ring the larger the volume of the water. This section will explain this idea.

Refraction model Fig. 2.6.a illustrates a ray coming from the environment is refracted twice before reaching the camera. Since, we are only interested in the rays that can reach the camera, we can use a backward raytracing to know the paths of the rays. Moreover, we assume that the glass is so thin that we can ignore the refraction due to the glass. To further simplify the model, we remove the refraction between the glass and the air by moving the camera from position C to C', as shown in Fig. 2.6.b. (This simplification is strict if the glass is a plane.) By doing so, the perpendicular distance from the camera to the refraction plane, denoted as C_z, is changed as: $C'_z = \frac{n_w}{n_a}C$, where n_w and n_a are the refractive indexes of water and air, respectively.

Dark Ring and Total Reflection The dark ring at the boundary of a water drop is because light coming from the environment is reflected back inside the water, instead of being transmitted to the camera. This phenomenon is known as the total reflection, and applies applies to all light rays whose relative angles to the water's surface normal, are larger than the critical angle, denoted as θ_W.

To analyze the correlation between the critical angle with the water-drop 3D shape $S : z(x, y)$, we start with stating the Snell's law:

$$\theta_W = \sin^{-1} \frac{n_a}{n_w}. \tag{2.8}$$

where n_w and n_a are the refractive indexes of water and air, respectively. As indicated in Fig. 2.6.c, we denote the surface normal as $\mathbf{N}$, which can be derived from z as: $\mathbf{N} = (N_x, N_y, N_z) = \frac{\mathbf{N}'}{\|\mathbf{N}'\|}$ where, $\mathbf{N}' = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, 1)$, and $\| \quad \|$ denotes the ℓ_2 norm.

The angle between the surface normal and the z-axis denoted as θ_N is the sum of the incidence angle of water, θ_w, and the angle between the incidence ray and z-axis $\theta_{C'}$:

$$\theta_N = \theta_w + \theta_{C'}. \tag{2.9}$$

where $\theta_{C'}$ is determined by the position of camera and the position of refraction. Considering the z component of the normal N_z is also defined as: $N_z = \cos\theta_N$, we know that when $N_z \leq \cos(\theta_W + \theta_{C'})$, the corresponding water drop area is totally dark.

Figure 2.6: Refraction in a water drop.
(a) A ray coming from the environment is refracted twice before reaching the camera C. (b) A backward ray tracing where a ray emitted from the camera passes through the same path as in (a); although, for simplification we remove the second refraction by moving the camera position to C'. (c) When θ_w is greater than the critical angle, light will not be transmitted but reflected inside. (d) Two polarized components, R_s and R_p, of the incidence ray.

For instance, when $\theta_{C'}$ is 0, and $\frac{n_a}{n_w}$ is approximatedly $\frac{3}{4}$, we have $N_z \leq 0.661$. Figure 2.7 shows some examples of synthetically generated dark rings.

As we can observe, a greater volume of the water drop indicates a wider dark ring. Therefore, it is possible to infer the water drop volume from the dark ring.

Dark Ring and Fresnel Equation Although the dark ring can be theoretically inferred from the water-drop geometry. Detecting them from an image is not trivial. Due to the sensor noise and the leak of light, dark rings are not totally dark. Moreover, there are textures in the environment that can be darker than dark rings. To resolve the problem, we employ the Fresnel equation and formulate the brightness values near the critical angle.

The refraction coefficients, denoted as $\mathcal{T}_s$ and $\mathcal{T}_p$, for two orthogonal polarized components for the light rays traveling from air to water are written as:

$$\mathcal{T}_s = 1 - \left(\frac{\sin(\theta_w - \theta_a)}{\sin(\theta_w + \theta_a)}\right)^2 \tag{2.10}$$

$$\mathcal{T}_p = 1 - \left(\frac{\tan(\theta_w - \theta_a)}{\tan(\theta_w + \theta_a)}\right)^2 \tag{2.11}$$

where θ_w and θ_a are depicted in 2.6.d. In our case, we assume the light from the environment is not polarized, and thus the overall refraction coefficient is $\mathcal{T} = \frac{1}{2}(\mathcal{T}_s + \mathcal{T}_p)$.

Concerning the dark rings, we are interested in two critical conditions. First, when the incidence angle θ_a is close to 0. In such a condition, $\sin\theta_a \approx \theta_a$, $\cos\theta_a \approx 1$, and consequently:

$$\mathcal{T} = \frac{4n_a n_w}{(n_w + n_a)^2}. \tag{2.12}$$

Substituting the value for water gives us $\frac{n_a}{n_w} = \frac{3}{4}$, and thus we have $\mathcal{T} \approx 0.980$.

Second, when incidence angle θ_a is close to $\frac{\pi}{2}$, (the locations near the dark ring). In such a condition: $\sin\theta_a \approx 1$, $\cos\theta_a \approx \frac{\pi}{2} - \theta_a$, as a result:

$$\mathcal{T} = 2\sqrt{1 - \left(\frac{n_a}{n_w}\right)^2}\left(\frac{n_a}{n_w} + \frac{n_w}{n_a}\right)\left(\frac{\pi}{2} - \theta_a\right) \tag{2.13}$$

Similar to the first condition, substituting the value $\frac{n_a}{n_w} = \frac{3}{4}$, we will obtain $\mathcal{T} \approx 2.76(\frac{\pi}{2} - \theta_a)$.

From these constrained values of $\mathcal{T}$, we will estimate the width of the dark ring. The detailed algorithm is discussed in the next section.

Figure 2.7: Indicate dark ring from water drop geometry.
(a) Water drop geometry. (b) The surface normal. (c) The indicated dark band. (a′)(b′)(c′) The case with a greater water drop volume.

Data	Camera speed	Camera shaking	Max raindrop speed observed
Experiment 1 - 4	5km/h	yes	0.48 pixel/s
Car-mounted	30km/h	yes	0.01 pixel/s
Surveillance	0	no	0.40 pixel/s

Figure 2.8: Raindrop dynamic of scenes in Chapter 4.

2.1.5 Dynamics

In rainy scenes, some raindrops might slide sporadically. The sliding probability and speed depend on a few attributes, such as, surface tensor coefficients, surface tilt, wind, raining intensity, raindrop size, etc. An exact modeling of raindrop dynamics is intractable. Fortunately, in light rainy scenes, we find it reasonable to assume most raindrops are quasi-static. We observed the motion of real adherent raindrops in scenes in Chapter 4. Focusing on a raindrop, we compared the current location with the location one minute later and convert it to speed (pixel per second). Table 2.8 lists the maximum speed observed in each scene. In our work, we only need to assume the raindrops to be static within seconds, and will quantitatively evaluate the tolerance of raindrop dynamics in Section 7.

2.1.6 Distribution.

The density of raindrops adhere to a given surface depends on the precipitation rate and the time interval that raindrops are collected. With other parameters fixed, the heavier the rain is or the longer raindrops are collected, the denser the adherent raindrops are.

As illustrated in Fig. 2.9, the density is a continuous parameter. In this thesis, we give the threshold that the distribution of adherent raindrops are considered to be sparse that:

1: There are clear intervals between raindrop area and non-raindrop area so that raindrop area is detectable.

2. Roughly less than 20% of the image are covered by raindrops so that there are sufficient information from non-raindrop area to repair the video.

Figure 2.9: Adherent raindrops in different distributions.

2.2 Modeling of Camera

Raindrop appearance is highly depending on the camera intrinsic parameters. In the first subsection, we first assume a pin-hole camera and non-blurred raindrops, and explore raindrop imagery properties in this condition. Based on our analysis in the first subsection, we model blurred raindrops in the next subsection. Unlike the previous methods [47, 32, 70, 69, 20], which try to model each raindrop as a unit object, we model raindrops locally from the derivative properties that involve only few parameters.

2.2.1 Clear Raindrop Imagery

Contract Imaging

As shown in Fig. 2.10(a), the appearance of each raindrop is a contracted image of the background, as if it is taken from a catadioptric camera. Mathematically, for a given raindrop, we describe the smooth expand mapping start from raindrop area Ω_r into the environment scene Ω_e as φ:

$$\varphi : \Omega_r \rightarrow \Omega_e \tag{2.14}$$

The appearance of the raindrop and the environment share the same image plane and coordinates. In order to distinguish, in this thesis, we denote the points and coordinates in raindrop Ω_r as: $P_r = (u, v)$ and the corresponding points and coordinates in environment Ω_e as $P_e = (x, y)$. Then φ can be expressed as:

$$P_e = (x, y) = \varphi(P_r) = \varphi(u, v) = (\varphi^1(u, v), \varphi^2(u, v)) \tag{2.15}$$

As illustrated in Fig. 2.11, if A. all the camera inner parameters; B. all the geometric information of the raindrops; and C. all the depth information of the environment are determined, then φ is uniquely determined according to the refraction model.

For detection, except A, all other parameters should not be assumed known a priori. Roser et al. assumed B as a part of an ideal sphere [46] or Bezier curves [47], which only covers a small group of possible shapes.

Our task is detection, therefore, other than exhaustively solve φ, we extract differential properties of φ which are sufficient for detection. In Section 3, we theoretically estimate the linear expansion ratio of φ. Based on it, we propose dense motion based detection method. In Section 4, we theoretically estimate the area expansion ratio and the intensity change based detection method is thus proposed.

(a)

(b)

Figure 2.10: Raindrop imagery formation.
(a) The appearance of each raindrop is a contracted image of the background, as if it is taken from a catadioptric camera. (b) For a given raindrop area Ω_r, there is a smooth expand mapping φ start from the Ω_r into the environment scene Ω_e.

Figure 2.11: Refraction model of a pair of corresponding points on an image plane. There are two refractions on the light path through a raindrop. (The camera lens cover or protecting shield is assumed to be a thin plane and thus neglected.)

While all the previous methods try to model each raindrop as a unit object, we model raindrops locally from its derivative properties. Modeling a whole raindrop needs too many parameters which are impractical when those parameters are unknown; on the contrary, modeling the derivative properties needs only few parameters.

In this thesis, we model the derivative properties between a raindrop pixel and non-raindrop pixel that are originated from the same point in the environment. We observed that the imagery of an adherent raindrop is in fact the contraction of the environment. Based on this, we theoretically found that the contraction ratio is at least $\frac{1}{8}$ in any place on the raindrop. Based on this property, we propose the raindrop detection method by using dense motion estimation (e.g. optical flow). We also found that each pixel in a raindrop represents at least 64 pixels in the environment. Because of this ratio, raindrop area changes less compared to the environment. Relying on this analysis, we propose a detection method based on the change of intensity. Both methods detect raindrops on a pixel basis, making them generally applicable for raindrops with any shape and size. Fig. 1.2(d) shows one result of our proposed detection method.

Spatial Derivative of Contract Imaging

In this section, we first mathematically form the derivative of φ and the expansion ratio. Then, we theoretically estimate a lower boundary of the expansion ratio according to the refraction model. Based on it, lastly, we propose a detection method based on dense motion.

Local differentials and expansion ratio Referring to Eq.(2.15), the local differentials at $P_r = (u, v)$ is defined as:

$$J_\varphi(P_r) = J_\varphi(u, v) = \begin{pmatrix} \varphi_u^1(u, v) & \varphi_v^1(u, v) \\ \varphi_u^2(u, v) & \varphi_v^2(u, v) \end{pmatrix} \tag{2.16}$$

with: $\varphi_u^1(u, v) = \frac{\partial \varphi^1(u,v)}{\partial u}$.

The local motion at (u, v), denoted as $(\delta u, \delta v)^T$, and the local motion at (x, y), denoted as $(\delta x, \delta y)^T$, is linearly associated by $J_\varphi(u, v)$:

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = J_\varphi(u, v) \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} \tag{2.17}$$

Instead of modeling φ or $J_\varphi(u, v)$, we are interested in the ratio between $\| (\delta x, \delta y) \|$ and $\| (\delta u, \delta v) \|$. According to Eq.(2.17):

$$\| (\delta x, \delta y) \|^2 = (\delta x, \delta y)^T (\delta x, \delta y) = (\delta u, \delta v)^T (J_\varphi(u, v))^T J_\varphi(u, v)(\delta u, \delta v) \tag{2.18}$$

with $(J_\varphi(u,v))^T J_\varphi(u,v)$ is symmetric and positive-semidefinite, and can be diagonalized as:

$$(J_\varphi(u,v))^T J_\varphi(u,v) = E^T \begin{pmatrix} \lambda_1^2(u,v) & \\ & \lambda_2^2(u,v) \end{pmatrix} E \tag{2.19}$$

where E is an orthogonal matrix, and $0 \leq \lambda_1(u,v) < \lambda_2(u,v)$. Therefore, according to Eqs.(2.18) and (2.19), for any directional motion $(\delta u, \delta v)$ at (u,v):

$$\frac{\| (\delta x, \delta y) \|}{\| (\delta u, \delta v) \|} \geq \lambda_1(u,v) \tag{2.20}$$

We can give a lower boundary, denoted as λ_{lower}, for all $\lambda_1(u,v)$ inside the raindrop area Ω_r:

$$\lambda_{lower} \leq \min\{\lambda_1(u,v) | (u,v) \in \Omega_r\} \tag{2.21}$$

We call it the lower boundary of the contraction ratio.

Estimating the Lower Boundary of Contraction Ratio

A light ray passing through a raindrop undergoes two refractions: first, the refraction from the air to the water, and second, the refraction from the water to the air. Thus, the mapping function, φ, can be separated as two continuous mappings:

$$\varphi = \overset{a-w}{\varphi} \circ \overset{w-a}{\varphi} \tag{2.22}$$

index a stands for air, and w stands for water.

Assuming the contact surface between the camera lens cover and raindrop is flat, $\overset{a-w}{\varphi}$ should be analytically solvable.

According to Snell's law, where $n_a \sin\theta_i = n_w \sin\theta_o$, we can have:

$$\frac{P_e}{P_r} = \frac{d\tan\theta_e}{d\tan\theta_i} = \frac{n_a}{n_w} \frac{1 + \frac{n_w}{n_a}\frac{d}{D}}{1 + \frac{d}{D}} = constant \tag{2.23}$$

where the notation is defined in Fig. 2.11, and $\frac{n_w}{n_a}$ is approximately $\frac{4}{3} > 1$. Thus the ratio:

$$\frac{\mathrm{d}P_e}{\mathrm{d}P_r} = \frac{P_e}{P_r} = \frac{n_a}{n_w} \frac{1 + \frac{n_w}{n_a}\frac{d}{D}}{1 + \frac{d}{D}} > \frac{n_a}{n_w} = 0.75 \tag{2.24}$$

Hence, a lower boundary of the expansion (contraction here) ratio is:

$$\overset{a-w}{\lambda}_{lower} = 0.75 \tag{2.25}$$

Now, we estimate the expansion ratio of the second refraction $\overset{w-a}{\varphi}$. Note that, referring to Fig. 2.11, although in the first refraction the direction and position of

Figure 2.12: Local linear space
Given a point on raindrop surface S, its two principle curvatures vectors k_1, k_2 and the normal n are orthogonal to each other.

the emergence light trace could be analytically solved, the position and angle of the incident light of the second refraction is still unsolvable. This is because we have no knowledge of the position and shape of the raindrop.

To estimate the expansion ratio of the second refraction, we start from the differential geometry on the outer surface of the raindrop. For a given position (u, v) on the surface of the raindrop, its up to second order differential geometry values are illustrated as in Fig. 2.12 [76]. The upper principle curvature vector, k_1, points to the direction where the raindrop surface bends most. And the lower principle curvature vector, k_2, points to the direction where the surface bends least. The curvature vector of any other direction, k, is the linear combination of k_1 and k_2. The values of any curvature vector k is bounded by k_1 and k_2:

$$k_2 \leq k \leq k_1 \tag{2.26}$$

The reciprocal of curvature is called curvature radius: $R = \frac{1}{k}$. In any direction, it is bounded by two principle curvature radius: $R_1 \leq R \leq R_2$.

As illustrated in Fig. 2.11, we now consider the second refraction locally at given point (u, v). Mention that there is no knowledge about how this local coordinates is aligned to the global coordination. First, we try to estimate the angular ratio $\frac{\mathrm{d}\theta_o}{\mathrm{d}\theta_i}$.

Figure 2.13: Simplified refraction model of the second refraction using principle curvature.

Refraction model of the second refraction when assuming the normal of refraction is very close to the image system axis. The notation are same with Fig. 2.11, R is the curvature radius at the place and direction where the refraction happens.

According to Snell's law, we have:

$$\frac{\mathrm{d}\theta_o}{\mathrm{d}\theta_i} = \frac{\frac{n_w}{n_a}\cos\theta_i}{(1-(\frac{n_w}{n_a})^2\sin^2\theta_i)^{\frac{1}{2}}} \tag{2.27}$$

where we know that $\frac{n_w}{n_a} > 1$, thus Eq.(2.27) gets its minimum when θ_i=0:

$$\min\left(\frac{\mathrm{d}\theta_o}{\mathrm{d}\theta_i}\right) = \left.\frac{\mathrm{d}\theta_o}{\mathrm{d}\theta_i}\right|_{\theta_i=0} \tag{2.28}$$

This is in accordance with the observation of real raindrop image shown in Fig. 2.10(a).

As illustrated in Fig. 2.13(b), according to Eq.(2.28), we may put the normal of the raindrop surface considerably close to the image system axis. Assuming every angle is significantly small:

$$\theta_i \ll 1, \theta_o \ll 1, \theta_e \ll 1, \theta_r \ll 1 \tag{2.29}$$

According to Eq.(2.29), we can use the following approximation: $\sin\theta = \tan\theta = \theta$, $\frac{\mathrm{d}P_e}{\mathrm{d}P_r} = \frac{P_e}{P_r}$. The expansion ratio is estimated as:

$$\frac{\mathrm{d}P_e}{\mathrm{d}P_r} = \frac{P_e}{P_r} = \frac{\theta_e}{\theta_r} = 1 - \frac{n_w}{n_a}\frac{d}{R}\frac{D}{d+D} \tag{2.30}$$

To estimate Eq.(2.30), we only need to give an estimation of the upper boundary of the curvature radius R for any raindrop at any given position. Since, the camera lens cover is vertical to the ground, big raindrops will slide down, and according to our observation of real data, almost all raindrops has a diameter smaller than $5mm$: $2R < 5mm$. Then $R < 5mm$ is a very safe assumption. Assuming $d > 100mm$ and $D > 1m$, we have:

$$\frac{P_e}{P_r} < -11 \tag{2.31}$$

The negative sign means the image on the raindrop is inverted, this is in accordance with our observation on real data.

The expansion ratio of the second refraction is then estimated as:

$$\overset{w-a}{\lambda}_{lower} > 11 \tag{2.32}$$

Substituting Eqs.(2.25) and (2.32) into Eq.(2.22), we have the overall expansion ratio estimation:

$$\lambda_{lower} > 10 \tag{2.33}$$

This means the motion in the environment $\| (\delta x, \delta y) \|$ is at least 10 times as great as the corresponding motion $\| (\delta u, \delta v) \|$ on the raindrop. Fig. 2.14 is an observation of real data, which demonstrates our estimation.

Figure 2.14: Observing the expansion ratio in x and y direction on real data.

2.2.2 Blurred Raindrop Imagery

In contrast to raindrop imagery with a pin-hole camera, for a normal lens camera, when the camera focuses on the environment scene, raindrops will be blurred. To handle this, we model blurred raindrops, and theoretically derive the temporal property of raindrop pixels.

Blurred Raindrop

As illustrated in Fig. 2.15, Fig. 2.16, and Fig. 2.17,the appearance of a pixel on an image plane depends the collection of light rays. These rays can come from light emitted directly by an environment point (Fig. 2.15.A), light refracted from a raindrop (Fig. 2.15.B), and a mixture of environment light and raindrop light (Fig. 2.15.C). We denote the light intensity collected by pixel (x, y) as $I(x, y)$. We also denote the light intensity formed by an environment point that intersects with the line of sight as $I_e(x, y)$; and, the light intensity reached (x, y) passing through a raindrop as $I_r(x, y)$. Hence, pixel (x, y) collecting light from both the raindrop and the environment can be described as:

$$I(x, y) = (1 - \alpha)I_e(x, y) + \alpha I_r(x, y), \tag{2.34}$$

where α is the proportion of the light path covered by a raindrop, as depicted in Figs. 2.16.

Blending coefficient α is determined by the area of the light path and the raindrop. Using the model in Fig. 2.15, the diameter of the light path on the raindrop plane can be estimated using:

$$\frac{D}{D+d}A = \frac{D}{D+d}\frac{f}{N}, \tag{2.35}$$

where $\frac{f}{N}$, called the f-stop, is the convention expression for the camera aperture setting.

A more convenient way to express α on the image plane is to use a blur kernel. First, as illustrated in Fig. 2.15, α is either 0 or 1. We denote the blending coefficient of clear raindrops as α_c. Hence, α of blurred raindrops can be calculated by convoluting α_c with a disk kernel, where the diameter of the kernel is given by:

$$\ell = \frac{(D-d)}{(D-f)}\frac{f}{d}A, \tag{2.36}$$

which is proportional of the aperture size A. The derivation of Eq. (2.36) can be found in the literature of depth from defocus [55]. Consequently, if a raindrop is significantly

blurred, the blending coefficient is smaller than 1. In such a case, the raindrop cannot totally occlude the environment. Fig. 2.17.b shows an example. Fig. 2.16 shows real blurred raindrops, and Fig. 2.17 show raindrop appearance, which is highly directional.

Temporal Derivative of Blurred Raindrop

We avoid estimating the exact appearance of blurred raindrops due to its intractability. Instead, we explore the temporal derivative features. In consecutive frames, we observe that the intensity of blurred pixels (case B and C) does not change as distinctive as that of environment pixels (case A). To analyze this property, let us look into the intensity temporal derivatives of blurred pixels. Referring to Figs. 2.15, case B and C, light collected from a raindrop is actually originated from a large area in the environment. We denote the area as $\Omega_r(x, y)$. At time t, we expand $I_r(x, y)$ in Eq. (2.34) as:

$$I_r(x, y, t) = \sum_{(z,w)\in\Omega_r(x,y)} W(z, w) I_e(z, w, t), \tag{2.37}$$

where $W(z, w)$ is the weight coefficient determined by the raindrop geometry. $W(z, w)$ and $\Omega_r(x, y)$ can be considered constant in a short period of time.

If we take the difference of intensity between time t_1 and t_2 in Eq. (2.35), and consider the triangle inequality, we have:

$$\begin{aligned} &|I_r(x, y, t_1) - I_r(x, y, t_2)| \\ &\leq \sum_{(z,w)\in\Omega_r(x,y)} W(z, w)|I_e(z, w, t_1) - I_e(z, w, t_2)|. \end{aligned} \tag{2.38}$$

Here, by taking into account Eq. (2.33), we know the area ratio is more than one hundred when the raindrops clearly appear, namely,

$$\mathcal{E}_\varphi^2 > 100 \gg 1 \tag{2.39}$$

Notice that φ is not conformal, and the proof is provided in the supplementary material. For blurred raindrops, the area ratio further expands. Referring to the model in Fig. 2.15, in addition to the expanded area caused by a raindrop, the out-of-focus blurring also causes the area to expand. Thus, we can consider $\Omega_r(x, y)$ to be a sufficiently large area. According to the law of large number, we can have:

$$E|I_r(x, y, t_1) - I_r(x, y, t_2)| \ll E|I_e(x, y, t_1) - I_e(x, y, t_2)|, \tag{2.40}$$

where E denotes the expectation.

Figure 2.15: The light path model on an image plane collecting light from A: environment, B: raindrop, C: both. Green light: the light coming from environment point; Blue light: the light refracted by a raindrop.

Figure 2.16: Raindrop-plane-cut of the light path model
from A: environment, B: raindrop, C: both when the raindrop can fully cover the light path. And A′, B′, C′: when the raindrop cannot fully cover the light path.

Figure 2.17: The appearance various from light path.
from A: environment, B: raindrop, C: both when the raindrop can fully cover the light path. And A′, B′, C′: when the raindrop cannot fully cover the light path.

Since the temporal derivatives work as a high pass filter, we may also consider Eq. (2.40) in a frequency domain, where the temporal high frequency component of a raindrop is significantly smaller than those of the environment, described as:

$$\mathcal{I}_r(x, y, \omega) \ll \mathcal{I}_e(x, y, \omega), \omega = \omega_{th}, \omega_{th} + 1, \cdots, N \tag{2.41}$$

where $\mathcal{I}$ is the Fourier transform of sequence $I(x, y, t), t = t_1, t_2, \cdots, N$, and ω_{th} is currently an undetermined threshold for the high frequency.

Figure 2.18: Raindrop appearance varying with aperture and direction.
a. α-channel of a disk with various blur kernels. b. Raindrops with varying f-stop. c. Raindrops with varying angles. Raindrop appearance is highly directional.

2.3 Modeling of Environment

Dense long range trajectories Given a video sequence, we can form a $3D$ spatio-temporal space as illustrated in Fig. 2.19.a, where the spatial position of each pixel is indicated by (x, y) and the time of each frame i by t_i. The notation T in Fig. 2.19.b represents a trajectory consisting of a number of concatenated nodes $N(i)$, shown in Fig. 2.19.c, and can be expressed as:

$$\begin{aligned} T &= \{N(i)\}, i_{start} \leq i \leq i_{end} \\ N(i) &= (x(t_i), y(t_i)) = (x_i, y_i), t_{start} \leq t_i \leq t_{end}, \end{aligned} \tag{2.42}$$

where i is the index of the video frame, and (x_i, y_i) is the position of the node. The start and end of a trajectory are denoted by t_{start} and t_{end} respectively. Note that the nodes are arranged in a temporal ascending order, where a trajectory has only one node at each frame.

Figure 2.19: Spatio-temporal space and dense trajectories.
(a) 3D Spatio-temporal space; (b) A 2D slice visualizes the dense trajectories. (c) A trajectory consists of a number of concatenated nodes.

2.4 Summary and Discussion

In this chapter, we explicitly model the imaging system in rainy scene. In the first section, we model the intrinsic properties of raindrops: those properties which are not depending on the environment or the camera setting. In the second section, we model the properties of raindrop imaging which are relying on the camera setting. In the third section, we model the properties of raindrop imaging which are depending on the environment.

Based on different modeling, in the following three chapters, we propose different methods on detecting and removal the adherent raindrops and restoring the video.

In Chapter 3, we describe the trajectory based method, which is replying on our modeling of the smooth motion from the camera and environment. In Chapter 4, we describe the blend in model based method, which is replying on our modeling of clear and blurred camera imagery. In Chapter 5, we describe the ray-tracing model based method, which is replying on our modeling of physical properties of raindrops.

Table. 2.1 is a summary of the methods and their replying properties from the modeling.

Table 2.1: The methods and their replying properties

Model	Chapter 3 Trajectories	Chapter 4 Blend-in	Chapter 5 Ray-tracing
Raindrop			
Size		✓	✓
Boundary		✓	✓
Surface			✓
Environment	✓		
Camera			
Clear imagery		✓	✓
Blurred imagery	✓	✓	

Extrinsic properties → Intrinsic properties

Chapter 3

Long Range Trajectories Based Methods

In this chapter, we describe the trajectory based methods, which is replying on our modeling of the smooth motion from the camera and environment.

There are two methods relying on the modeling. The first one is: Raindrop Detection and Removal from Long Range Trajectory. [73], which will be introduced in the first sub-chapter and the second method is: Robust and Fast Motion Estimation for Video Completion [72], which will be introduced in the second sub-chapter.

3.1 Raindrop Detection and Removal from Long Range Trajectory.

The performance of outdoor vision systems can be degraded due to bad weather conditions such as rain, haze, fog and snow. On rainy days, it is inevitable that raindrops will adhere to camera lenses, protecting shields or windscreens, causing failure to many computer vision algorithms that assume clear visibility. One of these algorithms is motion estimation using long range optical flow. In this case the correct correspondence of pixels affected by adherent raindrops will be erroneous, as shown in Fig. 3.1.b.

In this chapter, our goal is to detect and remove adherent raindrops (or just raindrops for simplicity) by employing long range trajectories. To accomplish this goal, our idea is to first generate initial dense trajectories in the presence of raindrops. Surely, these initial trajectories are significantly affected by raindrops, causing them to be terminated and drifted. We analyze the motion and appearance behavior of the affected trajectories, and extract features from them. We formulate these features in a Markov-random-field energy function that can be optimized efficiently. Having detected raindrops, we use trajectory linking to repair the terminated or drifted trajectories. Finally, we remove the raindrops using the trajectory based video completion (Fig. 3.1.c and d). The overall pipeline is described in Fig. 2.

Unlike some existing methods, in this work, first we introduce a novel detection method applicable for both thick and thin raindrops as well as raindrops of any size, shape, glare, and level of blurring. We call a raindrop thick when we cannot see the objects behind it, and thin, when it is sufficiently blurred, but still allows us to partially see the objects behind it. Second, we perform a systematic analysis of the behavior of thick and thin raindrops along motion trajectories based on appearance consistency, sharpness, and raindrop mixture level. This analysis is novel, particularly when applied to raindrop detection. Third, we devise a method to detect and remove raindrops that allows us to recover the motion field. In addition, to our knowledge, our method is the first to address the problem of adherent raindrops in the framework of long range motion trajectories.

(a) Video with raindrop (b) Dense trajectories

(c) Trajectory matching (d) Raindrop removal

Figure 3.1: An example of the results of our proposed detection and removal method. (a) Scene with raindrop. (b) Dense long trajectories. (c) Matching of trajectories occluded by raindrop. (d) Trajectory based video completion.

Figure 3.2: The pipeline of our method.

3.1.1 Related Work

Bad weather has been explored in the past decades including: haze, mist, fog (e.g.,[59, 11, 23, 39]), falling rain and snow (e.g., [3, 16, 7]). For falling rain, Garg and Nayar study the physical model first [15], and later detect and remove it by adjusting camera parameters [17, 16]. Barnum*et al*. [3] detect and remove both rain and snow. Recently, single image based methods are proposed by Kang *et al*. [30] and Chen *et al*. [7]. Unfortunately, applying these methods to handle adherent raindrops is infeasible, because of the significant physics and appearance differences between falling raindrops and adherent raindrops.

A number of methods have been proposed to detect thick adherent raindrops caused by sparse rain. Eigen *et al*. [10] and Kurihata *et al*. [32] proposed learning based methods, which are designed to handle raindrops, but not specifically to differentiate raindrops from opaque objects such as dirt. Both of the methods work only with small and clear (non-blurred) raindrops. Yamashita *et al*. utilize specific constraints from stereo and pan-tilt cameras [70, 69], and thus is not directly applicable for a single camera. Roser *et al*. propose a ray-tracing based method for raindrops that are close to certain shapes [46, 47], and thus can cover only a small portion of possible raindrops. You *et al*. [71] propose a video based detection method by using intensity change and optical flow. The method is generally useful to detect raindrops with arbitrary shapes, however the detection of thin raindrops are not addressed, and it requires about 100 frames to have good results. In comparison, our method only needs 24 frames, assuming the video frame rate is 24 fps.

As for raindrop removal, Roser and Geiger [46] utilize image registration, while Yamashita *et al*. [70, 69] align images using the position and motion constraints from specific cameras. You *et al*. use temporal low-pass filtering and patch based video completion [67]. Generally, there are some artifacts in the repaired video because

none of these methods consider motion consistency which is sensitive to human visual perception. Eigen *et al*. [10] replace raindrop image patches with clear patches through a neural-network learning technique, causing the method to be restricted on the raindrop appearance in the training data set. This method can only replace small and clear raindrops.

Sensor dust removal might be related to raindrop detections, [68, 75, 19], by considering raindrops as dust. Unlike dust however, raindrops could be large, not as blurred as dust, and affected by the water refraction as well environment reflection, making the sensor dust removal methods unsuitable for detecting raindrops.

For video based motion estimation, dense and temporally smooth motion estimation is desired. Sand *et al*. [49] propose particle video which generates motion denser than sparse tracking and longer than optical flow. Later, this idea is improved by Sundaram *et al*. [56] by utilizing GPU acceleration and large displacement optical flow [6]. Volz *et al*. [66] archive a pixel-level density by a new optical flow objective function, however their latency is limited to several frames. Rubinstein *et al*. [48] extend the temporal latency of methods [49, 56] by linking the trajectories occluded by solid objects. This paper uses [56] for initial trajectory estimation but with different termination criteria, and utilizes trajectory linking as in [48] but with the features derived from our trajectory analysis over raindrops. As a result, the motion field estimation of degraded videos by raindrops can be much improved, compared to those that do not consider such degradation.

3.1.2 Trajectory Analysis

To find features that differentiate raindrops from other occlusions, as well as to identify thick and thin raindrops, we need to analyze the appearance of patches along individual trajectories and the consistency of forward/backward motion. For this, we first need to know the image formation model of raindrops, and the computation of long range trajectories.

Raindrop model Unlike opaque objects, raindrops can look different in different environments due to the focus of the camera on the environment. Fig. 3.3 illustrates a raindrop physical model. Given a pixel located at (x, y), the appearance of the clear environment is denoted as $I_c(x, y)$ and the raindrop appearance as $I_r(x, y)$. For raindrops, the following mixture function models the intensity [71]:

$$I(x, y) = (1 - \alpha(x, y))I_c(x, y) + \alpha(x, y)I_r(x, y), \tag{3.1}$$

where $\alpha(x, y)$ denotes the mixture level, which is dependent on the size and position of the raindrop as well as the camera aperture.

Dense long range trajectories Given a video sequence, we can form a $3D$ spatio-temporal space as illustrated in Fig. 2.19.a, where the spatial position of each pixel is indicated by (x, y) and the time of each frame i by t_i. The notation T in Fig. 2.19.b represents a trajectory consisting of a number of concatenated nodes $N(i)$, shown in Fig. 2.19.c, and can be expressed as:

$$\begin{aligned} T &= \{N(i)\}, i_{start} \leq i \leq i_{end} \\ N(i) &= (x(t_i), y(t_i)) = (x_i, y_i), t_{start} \leq t_i \leq t_{end}, \end{aligned} \tag{3.2}$$

where i is the index of the video frame, and (x_i, y_i) is the position of the node. The start and end of a trajectory are denoted by t_{start} and t_{end} respectively. Note that the nodes are arranged in a temporal ascending order, where a trajectory has only one node at each frame.

We employ GPU-LDOF [56] to generate the initial dense trajectories. However, we ignore its trajectory termination criteria; since, [56] considers only solid occlusions, while in rainy scenes, there are thin raindrops, where the occluded scenes can still be seen. Another reason is that [56] considers occlusion boundaries to be sharp, while in our case, raindrop boundaries are usually soft due to the out-of-focus blur. We generate trajectories in a forward motion, from the first to the last frame. In this case occlusions by raindrops or other objects might cause some trajectories to stop, and consequently

(a) Imagery model

(b) Clear raindrop

(c) Blurred raindrop

Figure 3.3: Model of Raindrop Imaging System
(a) Raindrop model. (b) Appearance of a clear raindrop. (c) Appearance of blurred raindrop observed on the image plane.

Figure 3.4: Video in rainy scenes and events on the trajectories.
(a) A clear day scene. (b) A scene with a thick raindrop. (c) A scene with a thin raindrop. The clear scene data is from [49]. Four trajectory events are labeled as, A: Occluded by a solid non-raindrop object and drifted. B: Occluded by a thick raindrop and drifted. C: Occluded by a thin raindrop and drifted. D: Occluded by a thin raindrop but not drifted. The trajectory appearance of each event is shown in Fig. 6.

some areas in some frames will not have trajectories. To cover these areas, we also generate trajectories in a backward motion.

Fig. 3.4 shows an example of the dense trajectories in a clear day scene and in a scene with a thick and in a scene with a thin raindrop. In our findings, with regard to occlusions, a trajectory can encounter the following events: (A) it is occluded by a solid non-raindrop object and drifted; (B) it is occluded by a thick raindrop and drifted; (C) it is occluded by a thin raindrop and drifted; and (D) it is occluded by a thin raindrop but not drifted.

These events encountered by trajectories allow us to identify the presence of raindrops. We consider that occlusions by thick raindrops or opaque objects will cause abrupt changes in both the appearance and the motion along trajectories, while occlusions by thin raindrops will mainly cause changes in the appearance, particularly the sharpness. The details of the analysis are as follows.

Motion Consistency Analysis

For a trajectory T generated by forward tracking, we consider a node $N(i)$ on frame t_i. Its succeeding $N(i+1)$ is found by referring to the forward optical flow $\mathbf{f}_i^+ = (u^+, v^+)_i$ from frame i to frame $i+1$:

$$N(i+1) = (x_i, y_i) + (u^+(x_i, y_i), v^+(x_i, y_i))_i = N(i) + \mathbf{f}_i^+(N(i)). \tag{3.3}$$

Similarly, given a trajectory T' generated from backward tracking, nodes are related by the backward motion:

$$N'(i) = N'(i+1) + \mathbf{f}_{i+1}^-(N'(i+1)), \tag{3.4}$$

where $\mathbf{f}_{i+1}^- = (u^-, v^-)_{i+1}$ is the backward optical flow from frame t_{i+1} to frame t_i.

If nodes along a trajectory are not occluded and the optical flow is correctly estimated, the following equation stands with negligible (sub-pixel) error:

$$\begin{aligned} m^+(N(i)) &= \|\mathbf{f}_i^+(N(i)) + \mathbf{f}_{i+1}^-(N(i) + \mathbf{f}_i^+(N(i)))\|_2 = 0 \\ m^-(N'(i)) &= \|\mathbf{f}_i^-(N'(i)) + \mathbf{f}_{i-1}^+(N'(i) + \mathbf{f}_i^-(N'(i)))\|_2 = 0 \end{aligned} \tag{3.5}$$

where $m^+(N(i))$ and $m^-(N(i))$ are the forward motion consistency and the backward motion consistency of node $N(i)$, respectively. $\| \ \|_2$ is the L2 norm.

Motion inconsistency caused by occlusions Given a trajectory from the forward tracking (or the backward tracking), the motion consistency $m^+(N(i))$ might not be zero if

Figure 3.5: Ambiguity of correspondences
When a point is covered by a thin raindrop, it has two correspondences in other frames: the raindrop and the covered object. The causes incorrect tracking for optical flow that assumes only one correspondence.

$N(i+1)$ is occluded. In events A and B, $N(i+1)$ is completely occluded by an opaque object or a thick raindrop. In this case, $N(i)$ does not have a corresponding node in the next frame. However, the inter-frame optical flow $\mathbf{f}_i^+$ still gives correspondence for $N(i)$. This is because the optical flow regulation forces every pixel to have correspondence. Thus, corresponding node $N(i) + \mathbf{f}_i^+(N(i))$ is wrong, resulting in a non-zero motion consistency.

In event C, $N(i+1)$ is occluded by a thin raindrop, which according to Eq. (3.1), can generate a partial occlusion. As illustrated in Fig. 3.5, in this event, the consistency is likely to be non-zero, since the pixel at $N(i+1)$ is the mixture of both the tracked node and the raindrop, where each of them has correspondence in the previous frame; causing both the forward and backward optical flow to likely generate wrong correspondence. Here, the mixture level α plays an important role for the wrong correspondence. Overall, the thicker the raindrop, the more likely the consistency is to be non-zero.

In event D, $N(i+1)$ is occluded by a considerably thin raindrop, where $N(i+1)$ is sufficiently visible such that both the forward and backward optical flow correctly match $N(i)$ with $N(i+1)$. In this event, the mixture level is close to zero, usually less than 0.2.

Motion consistency feature Since events A, B and C might result in a non-zero motion consistency value, we can use the consistency, $m^+(N(i))$ and $m^-(N(i))$, as features to indicate the presence of occlusion, which in some cases, can be raindrops.

We calculate the motion consistency feature for each frame at t_i by collecting m^+ and m^- of all the nodes in the frame, denoted as M_i. Assuming the video frame rate

is 24 fps [1] and raindrops are static in a short time period (one second), we sum up the features over 24 frames:

$$\mathcal{M}_i = \sum_{i-24<j\leq i} M_j. \tag{3.6}$$

Some pixels might not have consistency values due to the failure of optical flow to track. In this case, we obtain the values from linear interpolation. Fig. 3.7.a shows an example of $\mathcal{M}_i$.

As for event D, since possible occlusion can not be detected by the motion consistency, we detect it based on the appearance analysis, discussed in the subsequent section.

Appearance Analysis

Given a trajectory T, we crop a small image patch, denoted as $P(i)$, centered at each node $N(i)$ with length r, where r is set to 21 pixels by default (based on the resolution of our videos). Fig. 3.6 shows an example of patches sequenced along trajectories for events A, B, C, and D.

Appearance consistency As can be seen in Fig. 3.6, all four events might generate appearance changes, particularly for events A, B and C. We calculate the appearance consistency for node $N(i)$ using:

$$a(N(i)) = \|SIFT(P(i+1)) - SIFT(P(i))\|_2, \tag{3.7}$$

where $SIFT(\ )$ is the SIFT descriptor [37], converts patch P to one feature array. For color images, RGB channels are converted separately and, later combined.

The reason of choosing SIFT is to achieve robustness against some degrees of affine deformation. Since even without occlusions, the appearance of an image patch might change. Note that within a few frames (i.e., fewer than 24 frames), these changes should be within the degrees where SIFT can still work, since they represent less than 1 second in real time.

[1] The 24fps framerate only for reference on how we can deal with raindrop dynamics since our method assumes static raindrops during the detection process, while in fact in the real world raindrops can move. Hence, assuming the widely adopted framerate, it means we assume raindrops at least do not move in 1-second period of time. Obviously, a higher framerate does not pose any problem (except for the computation time), however a much lower framerate will create a large displacement problem, which can affect the optical flow accuracy.

Figure 3.6: Appearance of trajectories in Fig. 3.4.
The patch size (21×21 pixels by default) is set to 41×41 pixels for better visualization.

Similar to the motion consistency, we compute the appearance consistency for frame t_i, denoted as A_i, by collecting the appearance consistency of all of the nodes in the frame. The integration of A_i over 24 frames is denoted as $\mathcal{A}_i$. Fig. 3.7.b shows an example of $\mathcal{A}_i$.

The appearance consistency is able to detect all of the occlusion events (A, B, C, and D), however, it lacks the ability to distinguish a non-raindrop occlusion from a solid raindrop occlusion.

Sharpness analysis We define the sharpness of patch $P(i)$ as:

$$s(P(i)) = \sum_{(x,y)\in P(i)} \left\| \frac{\partial}{\partial x} I(x,y), \frac{\partial}{\partial y} I(x,y) \right\|_2 \tag{3.8}$$

where $I(x, y)$ is the intensity value of pixel (x, y). For color images, RGB channels are calculated separately and added up afterward.

Unlike blurred raindrops that have low sharpness in the area including the boundary, non-raindrop objects will have large sharpness at their boundary, or inside their area when they are textured. Therefore, by evaluating the sharpness, we can differentiate non-raindrop objects (Event A) from raindrops (Events B, C and D). The sharpness for frame t_i, denoted as S_i is the collection of the sharpness of all nodes in the frame. The integration of S_i over 24 frames is denoted as $\mathcal{S}_i$, Fig. 3.7.c shows an example of $\mathcal{S}_i$.

Raindrop mixture level Analyzing the sharpness along trajectories does not only enable us to distinguish raindrops from non-raindrop objects, but it also allows us to estimate the raindrop mixture level, α. For a given patch $P(i)$, Eq. (3.1) can be rewritten as:

$$\begin{aligned} P(i) &= (1-\alpha(i))P_c + \alpha(i)P_r(i) \\ \alpha(i) &= \alpha(N(i)) = \alpha(x(t_i), y(t_i)), \end{aligned} \tag{3.9}$$

where P_c is the clear patch component and $P_r(i)$ is the raindrop component. $\alpha(i)$ is the mixture level of the patch. In the equation, we have made two approximations: First, the mixture level α inside a patch is constant. Second, the change of clear patch component P_c along the trajectory is negligible in a short time period (i.e., within 24 frames for a video with 24 fps).

From Eqs. (3.8) and (3.9), we can write the following:

$$\begin{aligned} s(P(i)) &= s[(1-\alpha(i))P_c + \alpha(i)P_r(i)] \\ &\leq s[(1-\alpha(i))P_c] + s[(\alpha(i)P_r(i)] \\ &= (1-\alpha(i))s(P_c) + \alpha(i)s(P_r(i)). \end{aligned} \tag{3.10}$$

(a) Motion consistency $\mathcal{M}$ (b) Appearance consistency $\mathcal{A}$

(c) Sharpness $\mathcal{S}$ (d) Mixture level $\mathcal{B}$

Figure 3.7: Raindrop features.
(a) Accumulated motion consistency $\mathcal{M}$. (b) Accumulated appearance consistency $\mathcal{A}$. (c) Accumulated sharpness $\mathcal{S}$, colormap is inversed for visualization. (d) Mixture level estimation $\mathcal{B}$.

$$s((1-\alpha(i))P_c) = s[P(i) - \alpha(i)P_r(i)] \leq s(P_c) + \alpha(i)s(P_r(i)). \tag{3.11}$$

Assuming the raindrop is sufficiently blurred, we have: $s(P_r(i)) = 0$. Substituting this in Eqs. (3.10) and (3.11) and comparing them, we have: $s(P(i)) = (1-\alpha(i))s(P_c)$. Thus, we can estimate the mixture level of patch $P(i)$ by comparing the sharpness with a clear patch in the same trajectory as:

$$\alpha(i) = 1 - s(P(i))/s(P_c). \tag{3.12}$$

For a given patch $N(i)$, sharpness of a clear patch $sh(P_c)$ is obtained by evaluating the patch sharpness for m neighbor patches along the trajectory:

$$s(P_c) = \max s(P(i \pm j)), j \leq m, \tag{3.13}$$

where $m = 10$ as default. When the clear patch has less texture, $s(P_c)$ is small and will result in a large error in Eq. (3.13). Hence, we only use textured patches to estimate the mixture level. Note that, if m is too small, the trajectory interval is too short, making us unable to have clear patches. On the contrary, if m is too large, the tracking drift will accumulate, causing the trajectories to be incorrect. In our observation for our test videos, $m = 10$ could avoid the problem.

Similarly, we can collect the mixture level for frame t_i, denoted as B_i. The integration of B_i is denoted as $\mathcal{B}_i$. Fig. 3.7.d is an example of $\mathcal{B}_i$.

(a) Binary labeling (b) Multi labeling

Figure 3.8: Raindrop detection via labeling.
(a) Binary labeling of the raindrop area. (b) Multiple labeling of the mixture level.

3.1.3 Raindrop Detection

The detection of raindrops can be described as a binary labeling problem, where for given a frame, the labels are raindrop and non-raindrop. Similarly, the mixture level can be described as a multiple labeling problem. The labeling can be done in the framework of Markov random fields (MRFs).

Raindrop labeling In the previous section, three features are shown for raindrop detection: motion consistency $\mathcal{M}$, appearance consistency $\mathcal{A}$ and sharpness $\mathcal{S}$. Thus, to detect raindrops, we combine these three features, after normalizing them, to form the following data term:

$$\begin{aligned} E_{data}(\mathbf{x}) &= \|\mathcal{F}(\mathbf{x}) - (w_m + w_a)L(\mathbf{x})\|_1 \\ \mathcal{F}(\mathbf{x}) &= (w_m\mathcal{M}(\mathbf{x}) + w_a\mathcal{A}(\mathbf{x}))\max(0, 1 - w_s\mathcal{S}(\mathbf{x})) \end{aligned} \tag{3.14}$$

where w_m, w_a and w_s are the weight coefficients for the three features. And $\mathcal{F}(\mathbf{x})$ is the combined feature. The weights were chosen empirically by considering the precision-recall curve, where a larger weight enabled more sensitive detection. We set $w_m = 16$, $w_a = 16$ and $w_s = 1$ by default. $L(\mathbf{x}) \in \{0, 1\}$ is the binary label, with 0 being non-raindrop. The normalization of the three features is done by setting the mean value to 0.5 and the variance to 0.5.

Since the boundaries of raindrops are significantly blurred, we can use a smoothness prior term for labeling neighboring pixels:

$$E_{prior}(\mathbf{x}) = \sum_{\mathbf{x}_j \in V(\mathbf{x})} |L(\mathbf{x}_j) - L(\mathbf{x})|, \tag{3.15}$$

where $V(\mathbf{x})$ is the neighbor of $\mathbf{x}$. We use graphcuts [14, 4, 5, 31] to solve the optimization. Fig. 3.8.a is an example of the labeling result.

Mixture level labeling Having obtained the binary labeling of the raindrop areas, we further label the raindrop mixture level $\alpha(\mathbf{x})$ through multi-level labeling. We use the estimated mixture level $\mathcal{B}$ (Eq. (3.12)) as a clue. The data term is expressed as:

$$E'_{data}(\mathbf{x}) = w_b \|\mathcal{B}(\mathbf{x}) - \alpha(\mathbf{x})\|_1 + w_L \|\tilde{L}(\mathbf{x}) - \alpha(\mathbf{x})\|_1, \tag{3.16}$$

where $\tilde{L}(\mathbf{x})$ is the binary labeling result, w_b and w_L are the weight coefficients which are set to $w_b = 8$, $w_L = 2$ by default. $\alpha(\mathbf{x})$ has 21 uniform levels from 0 to 1. The prior term is set in a similar way to that of the binary labeling. Fig. 3.8.b shows our estimated mixture level for all pixels.

3.1.4 Raindrop Removal

Having detected the raindrops, the next step is to remove them. The idea is that given a detected area of a raindrop, we collect the patches along the corresponding trajectories, and use these patches as a source of information to fill in the detected raindrop area.

Based on the binary labeling result, we first remove nodes in the trajectories that are labeled as raindrops, since these trajectories are likely to be incorrect or drifted. By this operation, some of the trajectories will be shortened, and the others will be broken into two trajectories.

To replace the removed nodes of trajectories, we match the corresponding existing trajectories based on [48], where the data term is based on SIFT, temporal order, and inter-frame motion. Fig. 3.1.c is an example of matched trajectories. After matching, we interpolate the missing nodes. Given a matched trajectory pair T_i and T_j, the last node of T_i, denoted as $N^i(end) = (x(t^i_{end}), y(t^i_{end}))$, is matched to the first node of T_j, denoted as $N^j(1) = (x(t^j_{start}), y(t^j_{start}))$. Here, $t^i_{end} < t^j_{start}$ means for all matched pairs. We linearly interpolate the missing nodes between frames t^i_{end} and t^j_{start} based on:

$$N(k) = \frac{t^j_{start} - t_k}{t^j_{start} - t^i_{end}} N^i(end) + \frac{t_k - t^i_{end}}{t^j_{start} - t^i_{end}} N^j(1), \quad t^i_{end} < t_k < t^j_{start}. \tag{3.17}$$

Trajectory-based Video Completion

Having obtained the trajectories for the raindrop areas, the raindrop completion is done by propagating the clear background pixels along a trajectory towards the raindrop area. Using the guidance of trajectories, we propose a removal strategy which preserves both spatial and temporal consistency.

The completion is done frame by frame. First, we start from the first frame and move forward until we find a frame t which contains interpolated nodes. For the frame t, inside a raindrop area, we denote the interpolated nodes as $\{N_i(t)\}$, where i is the trajectory index. According to the trajectory, we find the corresponding nodes in the previous frame: $\{N_i(t-1)\}$. A transformation can be determined between the two sets of nodes. Depending on the number of nodes in the set, we use affine transformation for three and more matches, translation and rotation for two matches, and translation for one match. Then, the image patch from $t-1$ is transformed and placed at the raindrop area in t. By utilizing information from groups of nodes, we preserve both spatial consistency and temporal consistency. This process continues until it reaches

the last frame. For the repaired patch, we denote its confidence as: $C(t) = C(t-1) - 1$. The confidence degrades by 1 every time it is propagated. And the non-interpolated patches have a confidence of 0.

Similarly, we do the backward process starting from the last frame. As a result, for each repaired area, there are two solutions: one from the forward process, and one from the backward process. We chose the one with the higher confidence. As for static or quasi-static areas where no linked trajectory exists, we use the video inpainting method by Wexler *et al*. [67] for repair. An example of the repaired video is shown in Fig. 3.1.d.

Thin raindrops For thin raindrops (event D, generally $\alpha < 0.2$), the trajectories inside the raindrop areas are already correct, therefore we do not need to propagate the appearance from other frames, since we can directly enhance the appearance. As discussed in Sec. 3.2, thin raindrops can be relatively blurred, hence to enhance them, for a node N with appearance P, we convert P to $\mathcal{P}$ using 2D-DCT and set the constant component $\mathcal{P}(0,0) = 0$. Then, we enhance the sharpness according to the mixture level: $\mathcal{P}' = \frac{1}{1-\alpha}\mathcal{P}$. We replace the constant component which is the one with a non-raindrop node along the trajectory. Finally, the enhanced patch P' is obtained using inverse-DCT.

3.1.5 Experiments

We conducted both quantitative and qualitative evaluation to measure the accuracy of our detection and removal method. Our video results are included in the supplementary material.

Raindrop Detection

Dataset In our experiments, the video data were taken from different sources to avoid data bias and to demonstrate the general applicability of our method. Data 1 was from Sundarum *et al*. [56], data 3 was from KITTI Benchmark [18], data 5 and 7 were from You *et al*. [71] and the rest were downloaded from the Internet. In these data, the camera setups vary from a car mounted camera, a hand held camera to a surveillance camera.
Comparison with state-of-the-art We used both synthetic and real raindrops, and compared our method with three state-of-the-art methods, Eigen *et al*.'s [10], You *et al*.'s [71] and Roser *et al*.'s [46]. The results are shown in Fig. 3.9. As can be seen, Eigen *et al*. ' s method failed to detect large and blurred raindrops, and mislabeled textured areas (such as trees) as raindrops. As for You *et al*.'s method, although it correctly detected thick raindrops, thin raindrops were simply neglected. Roser *et al*.'s method detected round raindrops and thin raindrops only when the background was textuerless.
Quantitative evaluation For the synthetic raindrops, data 1-4 in Fig. 3.9, we quantitatively evaluated using the precision-recall curve. In addition of number of raindrop level evalutation, we also performed pixel-level evaluation. The precision is defined as the number of the correctly labeled pixels divided by the number of all pixels labeled as raindrops. The recall is defined as the number of the correctly labeled pixels divided by the number of the actual raindrop pixels. The result is shown in Fig. 3.17. As can be seen, our proposed method outperformed some existing methods for both accuracy and recall. Our method have a low false alarm rate for both thick and thin raindrops. As for the real raindrops, data 5-8, our method successfully labeled thin raindrops as well as thick raindrops and achieved better precision.
False alarm rate evaluation To test the robustness of our method, we ran our algorithm on the first four data shown in Fig. 3.9 with all the synthetic raindrop removed. Table 3.1 shows the number of raindrop spots detected, although there is no raindrop in the input videos. Our method shows a significantly low false alarm rate compared to the other methods.

Figure 3.9: The raindrop detection results using our method and the existing methods on synthetic data. Data 1: thick raindrops, car mounted camera.

Figure 3.10: The raindrop detection results using our method and the existing methods on synthetic data. Data 2: thin raindrops, surveillance camera.

Figure 3.11: The raindrop detection results using our method and the existing methods on synthetic data. Data 3: thick and thin raindrops, car mounted camera.

Figure 3.12: The raindrop detection results using our method and the existing methods on synthetic data. Data 4: thick and thin raindrops, hand held camera.

Figure 3.13: The raindrop detection results using our method and the existing methods on real data. Data 5: thick and thin raindrops, hand held camera.

Figure 3.14: The raindrop detection results using our method and the existing methods on real data. Data 6: thin raindrops, car mounted camera.

Figure 3.15: The raindrop detection results using our method and the existing methods on real data. Data 7: thick raindrops with glare, hand held camera.

Figure 3.16: The raindrop detection results using our method and the existing methods on real data. Data 8: thin raindrops with glare, car mounted camera.

Figure 3.17: Precision-recall curve on detection for the methods shown in Fig. 3.9. Evaluation at a pixel level.
Dashlines indicates the range where no data is available.

Figure 3.18: Precision-recall curve on detection for the methods shown in Fig. 3.9. Evaluation at number of raindrops level.
Dashlines indicates the range where no data is available.

Table 3.1: False alarms on Data 1-4 (Fig. 3.9) with all synthetic raindrops removed. Evaluated by number of spots erroneously detected as raindrops.

	Proposed	**Eigen et al.**	**You et al.**	**Roser et al.**
Data 1	0	67	8	17
Data 2	1	48	16	12
Data 3	1	140	6	5
Data 4	0	12	4	2

Speed On a 1.4GHz notebook with Matlab and no parallelization, the interframe optical flow was about one minute per frame. The tracking and feature collecting together was about 0.2 second per frame. Graphcut was about 5 second for one detection phase. While our algorithm is not real time, we consider it to be still useful for offline applications, such as road accident analysis, Google-like street data collection, etc.

Raindrop removal

Fig. 3.19 shows the results of raindrop removal of a few methods, along with the groundtruth. The results include those of Eigen *et al*.'s [10] and You *et al*.'s [71]. Roser *et al*.'s method does not provide the implementation details for raindrop removal, and thus it was not included. As can observed in the figure, our method removed both thin and thick raindrops. Eigen *et al*.'s method failed to remove large raindrops and it erroneously smoothed textured area. You *et al*.'s method failed to remove thin raindrops, and the quality is affected by the detection accuracy.

Repaired motion field Fig. 3.23 shows the results of the motion field estimation, before and after the raindrop removal. As shown in the figure, our method can improve the dense motion estimation, by removing the raindrops, and then repairing the motion fields.

Ground truth

Input

Proposed

Eigen *et al*. (2013)

You *et al.* (2013)

Figure 3.19: The raindrop removal results. Data 0: thick raindrops.

Figure 3.20: The raindrop removal results. Data 1: thick raindrops.

Ground truth

Input

Proposed

Eigen *et al*. (2013)

You *et al.* (2013)

Figure 3.21: The raindrop removal results. Data 2: thin raindrops.

Ground truth

Input

Proposed

Eigen *et al*. (2013)

You *et al.* (2013)

Figure 3.22: The raindrop removal results. Data 3: thick and thin raindrops.

Figure 3.23: Comparison on motion field estimation before and after raindrop removal.

3.1.6 Conclusion and Future Work

We have introduced a method that automatically detects and removes both thick and thin raindrops using a local operation based on the long trajectory analysis. Our idea is using the motion and appearance features that are extracted from analyzing the trajectories-raindrops encountering events. These features are transformed into a labeling problem which is efficiently optimized in the framework of MRFs. The raindrop removal is performed by utilizing patches indicated by trajectories, enabling the motion consistency to be preserved. We believe our algorithm can be extended to handle other similar occluders, such as dirt or dust. For future work, we consider exploring dense-trajectory analysis of dynamic raindrops and improving the computation time.

3.2 Robust and Fast Motion Estimation for Video Completion

Video completion repairs damaged or undesired regions by filling them with the most suitable data, and thus makes the whole video visually as realistic as possible. The damaged regions can be caused by watermarks, logos, mud, undesired objects, raindrops adhered to the lens, etc, which possibly occupy large space and appear in a few consecutive frames. Completing these regions is challenging, since properly interpolating large damaged regions spatially and temporally is rather problematic.

Methods based on the motion field is usually used to solve the completion problem. They assume the target objects or regions to be removed are either static or moving smoothly in consecutive frames. If the motion trajectory can be correctly modeled, they can fill in the damaged regions by copying the pixels along its trajectory. However, in real videos, the whole environment motion can be arbitrary and complex. It forces them to focus on modeling the specific motion of the target regions in specific perspective and to have strong constraints to simplify the environment motion. Zhang *et al*. [74] and Jia *et al*. [26] limit the background motion to be translation only. Jia *et al*. [27] and Patwardhan *et al*. [43] assume the background to be static. Shiratori *et al*. [52] and Liu *et al*. [35] uses an existing optical flow method [1] to calculate the motion. Moreover, the accuracy of optical flow calculated from damaged videos poses another problem, since the existing methods of optical flow assume the input video does not contain any damaged regions.

Instead of modeling specific object motion, in this paper, we focus on modeling more general environment motion. We propose a method that utilizes sparse matching and interpolation to estimate the environment motion. First, we employ SIFT [36], which is robust to arbitrary motion, to find sparse correspondences in neighboring frames. We remove the pixel correspondences in the damaged regions, and thus avoid their influences. We adopt a fast dense point sampling method to ensure the correspondence is uniformly distributed. Then, we generate a dense motion field by interpolating the sparse correspondences. To achieve spatially and temporally coherent interpolation, we propose a weighted explicit 2D polynomial fitting method. Unlike 3D polynomial fitting, the proposed 2D fitting has significantly efficient computational time. Finally, we finish the video completion by copying the correspondences indicated by the motion trajectory.

The proposed method is generally applicable to spatially and temporally smooth

(a) Damaged video (b) Motion estimation (c) Completed video

Figure 3.24: Video completion using the proposed method.
(a) Input video with large and consecutive damage. (b) Motion estimation using the proposed method. (c) Video completion using the motion.

motion, and is robust to handle a severely damaged video. In our experiment, it also achieved high computational efficiency which was 7 times faster than the optical flow based methods. Fig. 3.24 shows the result of the proposed method.

The rest of the paper is organized as follows. Section 2 describes the sparse matching method. Section 3 explains the interpolation and completion method. Section 4 shows quantitative experiments and applications in motion estimation and video completion. Section 5 concludes the paper.

(a) Damaged input and local matching areas
(b) Distributed matching
(c) Damaged area
(d) Sparse motion
(e) Dense motion

Figure 3.25: The proposed motion estimation method.
(a) For 2 consecutive frames, sparse matching is performed in each corresponding squared windows. (b) The sparse matching results, which are well distributed across the image. (c) The damaged area. (d) The motion (green needles), which is calculated by the matching. Motion correspondences in the damaged regions are removed. (e) The interpolated dense motion using the proposed weighted polynomial fitting. Left image: represented by needles. Right image: represented by color.

3.2.1 Robust Sparse Matching

Sparse Matching In video, the appearance of an object can continuously change in terms of scale, position, direction and perspective. [74, 27, 26, 43] make constraints to simplify the motion estimation. Although [1, 34] are generally applicable to arbitrary motion, like the methods by [52, 35], they suffer from the presence of damaged regions (or the regions of undesired objects).

In the proposed method, first, the SIFT-based sparse matching is used to overcome the changes of appearance. To some extent, SIFT keypoints are invariant to scale, position, rotation, and perspective transformation [36]. Second, in sparse pixel correspondences, one pixel correspondence can be assumed to be independent from the other correspondences. Therefore, deleting the correspondences that represent the damaged regions does not influence the correspondences of non-damaged regions.

Well Distributed Correspondences The proposed method uses sparse correspondence as anchor points for motion interpolation. It requires that the sparse correspondences are distributed uniformly across the images, in such a way that in any area, there exist sufficient anchor points for interpolation. However, the original SIFT algorithm tends to find correspondences in highly textured regions and to ignore others. To address this problem, we modify the matching strategy of SIFT. As illustrated in Fig. 3.25 (a), we do not apply the SIFT matching for the whole images, but for small windows. As default, the size of each window is 80×80 pixels and at least 3 matching pixels are found in each pair of windows. These small windows across the whole image ensure the correspondences are well distributed. For the two neighboring windows, there are 30 pixels overlapping so that the matching pixels in the window's boundary is not neglected. This matching strategy does not influence the computational time significantly, since only half of a window is matched twice. This strategy is inspired by Tuytelaars [63].

Fig. 3.25 (b) shows an example of the sparse matching. Mathematically, we denote all the N matching pixels found between frame t_1 and frame t_2 as:

$$\{(x_k, y_k, x'_k, y'_k)\}_{t_1,t_2}, k = 1, 2, \cdots, N, \tag{3.18}$$

where (x_k, y_k) is a pixel in frame t_1 and (x'_k, y'_k) is its correspondence in frame t_2. We apply the matching between consecutive frames. Specifically, for a given frame, the matching is found in both the previous and the subsequent 5 frames.

Correspondence to Motion Estimating the motion of corresponding pairs is straightforward. Referring to the notation in Eq. (3.18), for a corresponding pair (x, y) and (x', y'), the motion at (x, y) is denoted as $(\delta x, \delta y)$, which is equal to $(x' - x, y' - y)$. Specifically, we can denote all the corresponding pairs of the sparse motion between frame t_1 and t_2 as:

$$\{(x_k, y_k, \delta x_k, \delta y_k)\}_{t_1,t_2}, k = 1, 2, \cdots, N. \tag{3.19}$$

Figs. 3.25 (c)and(d) shows an example, where the sparse motion is represented by short arrows. Erroneous matching are directly removed.

3.2.2 Fast Space-time Motion Interpolation

Explicit Polynomial Fitting Having found the sparse motion, we estimate the dense motion by doing interpolation based on 2D explicit polynomial fitting. First, we introduce the un-weighted 2D explicit polynomial fitting, where an m degree 2D explicit polynomial P_m can be expressed as:

$$P_m(x, y) = \sum_{i+j=0,1,\cdots,m} a_{ij} x^i y^j, \tag{3.20}$$

with $\{a_{ij}\}$ the polynomial coefficients. We interpolate the sparse motion in the x direction and the y direction separately. The interpolation, in the x direction for example, implies finding the polynomial coefficients $\{a_{ij}\}$ that minimizes the squared sum fitting error:

$$\sum_{k=1,2,\cdots,N} |\delta x_k - P_m(x_k, y_k)|^2, \tag{3.21}$$

where $\{(x_k, y_k, \delta x_k)\}_{t_1,t_2}$ are found by the sparse matching (Eq. (2)). We use the eigen-based method, which is significantly fast, to solve Eq. (3.21). More details about the method can be found in [62].

Temporal Coherent Weighted Fitting The fitting introduced in Eqs. (3.20) and (3.21) is temporally incoherent, since each frame is fitted independently. To make it temporally coherent, we propose an efficient weighted 2D polynomial fitting method to fit multiple frames simultaneously. Referring to the notation in Eqs. (3.19) and (3.21), the weighted fitting means to find the 2D polynomial P_m that minimizes the following error function:

$$\sum_{j=-J}^{J} \Big(W(t_j - T) \sum_{k=1}^{N} |\delta x_{k,t_j} - P_m(x_{k,t_j}, y_{k,t_j})|^2 \Big), \tag{3.22}$$

where T is the center frame and $\{t_j\}$ are its previous and subsequent J frames. $W(\cdot)$ is a weight function which only depends on the temporal distance. $W(\cdot)$ is expressed as:

$$W(\Delta t) = \frac{1}{(\Delta t)^2} \frac{|J - \Delta t + 1|}{J}, \tag{3.23}$$

where $\Delta t = t_j - T$ is the temporal distance between the corresponding pairs, $\frac{1}{(\Delta t)^2}$ is called the speed term and $\frac{|J-\Delta t+1|}{J}$ is called the coherent term. The speed term converts the distance of the corresponding pairs to average speed. The coherent term is a pyramid function, such that high weight is given to temporally close matching pairs. Having

P found from the fitting, the motion at any place (x, y) to any temporal distance Δt is calculated as:

$$(\delta x, \delta y)_{\Delta t} = \mathbf{P}(x, y)\Delta t = (P^x(x, y)\Delta t, P^y(x, y)\Delta t). \tag{3.24}$$

Considering the balance between accuracy and efficiency, as default, we set $J = 5$ and $m = 10$. Fig. 3.25(e) shows an example of the interpolated motion field.

Considerable efficiency can be achieved by the proposed 2D fitting. Referring to Eq. (3.20), the number of coefficients $\{a_{ij}\}$ to be solved is in $O(m^2)$ complexity. If we use 3D polynomial, $P_m(x, y, t) = \sum_{i+j+k=0,1,\cdots,m} a_{ijk}x^i y^j t^k$, the number of coefficients $\{a_{ijk}\}$ is $O(m^3)$, which is considerably time consuming.

Video Completion We fill in the damaged regions in the input video by utilizing the estimated motion function. For a given frame with its motion function **P**, the damaged regions are completed in a pixel-by-pixel basis. For a given damaged pixel (x, y), its correspondence (x', y') in other frames is found by:

$$(x', y') = (x + P^x(x, y)\Delta t, y + P^y(x, y)\Delta t). \tag{3.25}$$

According to Eq. (3.25) we can find one correspondence in each of the neighboring frame. The spatially and temporally closest undamaged correspondence is considered to be the most coherent and thus chosen to be the best. Then, (x, y) is completed by copying the best correspondence. In the final stage, for those pixels whose correspondence is in the damaged regions, we adopt an image inpainting method [9] to complete them. Fig. 3.24(c) shows the result of a completed video.

3.2.3 Experiments

In this section, experiments to quantitatively analyze the effectiveness of the proposed video completion methods are provided.

Experiments and settings

To evaluate the effectiveness of the video completion method, we use data taken by vehicle mounted camera in cloudy days. The cloudy day environment mimics the rainy day environment but do not have adherent raindrops. Raindrop areas to be removed are manually label. All the situations: fast moving area, slowly moving area and static area are tested.

Comparison with existing methods.

To demonstrate the effectiveness, our method are compared with existing methods. As introduced in related works, the following 3 methods are chosen:

1. Image inpainting. (Criminisi et al. [8, 9]).
2. Space-time completion. (Wexler et al. [67])
3. Motion based completion. (Shiratori et al. [52])

Quantitative Evaluation

For quantitative analysis, we calculate the average intensity difference between the repaired image and the ground truth.

For each pixel in the repairing area, it is considered to be a RGB pixel:

$$I = (R, G, B) \tag{3.26}$$

And each pixel is 8-bit from 0 to 255.

For each repaired pixel I_r, its error from the groundtruth I_o is calculated as:

$$E = ||I_r - I_o||_0 = |R_r - R_0| + |G_r - G_0| + |B_r - B_0| \tag{3.27}$$

For each experiment and each repairing method, the repairing error is calculated as the average error of all repaired pixels. The average error of 4 methods in 3 experiments are listed in Table 3.

Figure 3.26: Video completion in fast moving area using our proposed methods and existing methods.

Figure 3.27: Video completion in slowly moving area using our proposed methods and existing methods.

(a) Ground truth

(b) Input video

(c) Image inpaiting

(d) Space-time video completion

(e) Dense motion based video completion

(f) Our proposed method

Figure 3.28: Video completion in static area using our proposed methods and existing methods.

(a) Input video

(b) Image inpaiting

(c) Space-time video completion

(d) Dense motion based video completion

(e) Our proposed method

Figure 3.29: Raindrop removal on Tohoku data using our proposed methods and existing methods (I).

(a) Input video

(b) Image inpaiting

(c) Space-time video completion

(d) Dense motion based video completion

(e) Our proposed method

Figure 3.30: Raindrop removal on Tohoku data using our proposed methods and existing methods (II).

Table 3.2: Comparison on average repairing error.

	Fast motion area	Slow motion area	Static area
Image inpainting	13.5	24.4	90.0
Space-time video completion	10.0	19.6	1.7
Dense motion based video completion	80.9	77.1	190.7
Our method	13.2	20.7	3.5

Tohoku Data

Lastly, we show the results of video completion on real data with adherent raindrop taken in Japanese northeastern area. Our experiments are also compared with three existing methods. The results are shown in Figs. 3.26 and 3.2.3.

Robustness Real videos captured by a car-mounted camera were used to test the robustness of the proposed motion estimation method. As shown in Fig. 3.31, we randomly deleted one third of the frames which makes the video seriously damaged. Since the car was moving along the road, the motion of the foreground should point to the end of the road, and the nearer object should have larger motion. For comparison, two typical optical flow methods were also tested: L-K-flow [1] which is used by Shiratori *et al*. [52] and SIFT-flow [34] which is the state-of-art. As one can see, only the proposed method estimated the motion more robustly.

Efficiency

Under the same hardware and environment, the average time used to repair one frame (640×480) using the proposed method and the two other methods is listed in Table 3.4. As shown in the table, the proposed method is significantly faster. The proposed

Table 3.3: Average completion error

	Fast moving area	**Slowly moving area**	**Static area**
Before completion	120.1	79.5	171.7
Criminisi 2003	23.5	34.4	**3.5**
Shiratori 2006	80.9	77.1	190.7
The proposed method	**13.2**	**20.7**	**3.5**

Table 3.4: Average completion time per frame

Criminisi 2003	**Shiratori 2006**	**The proposed method**
80s	145s	19s

method is also generally applicable to any large and consecutive video damage, as shown in Fig. 3.32.

(a) L-K flow (b) SIFT flow (c) Proposed method

Figure 3.31: Two experiments on robust motion estimation.
Row 1 and 3: input video and motion needles. Row 2 and 4: motion visualized by color.

Figure 3.32: Applications of video completion on logo removal.

Figure 3.33: Applications of video completion on raindrop removal.

3.2.4 Conclusion

We have proposed a sparse matching and interpolation based motion estimation method for completing video with large and consecutive damage. The SIFT based matching is used to estimate the initial sparse correspondences, followed by a dense motion interpolation using a weighted 2D polynomial is applied. Limitations of this method include the inaccuracy in capturing sharp and small motion, which we consider to be our future work.

Chapter 4

Blend-in Model Based Method

In previous chapter, we have introduced the methods based on the modeling of smooth camera motion. Other than utilizing the extrinsic properties. In this chapter, we use the assumption which is relative more depending on the raindrop intrinsic modeling, says, the blend-in modeling. As introduced in Chapter 2.2, using the blend-in model, the proposed method does not need to assume the camera undergoes a smooth motion.

4.1 Raindrop Detection

4.1.1 Feature Extraction

We generate two features for the detection: a motion feature (OF) which is based on the analysis of clear images in Sec. 3; and the intensity change feature (IC) which is based on analysis blurred images in Sec. 4. We calculate the motion feature using a robust optic flow algorithm, *e.g.*, SIFT-flow [34], which is shown in Fig. 2.14.b, and calculate the intensity change feature using $|I(x, y, t_1) - I(x, y, t_2)|$, which is shown in Fig. 4.2.b.

In the examples, the two features are calculated using only two consecutive frames. In fact, the features will be more informative if they are calculated using data accumulated over more frames. Statistically the more frames used, the more descriptive the features are. Unfortunately, raindrop positions can shift over a certain period of time, making the detection using long frames erroneous. In our observation, with moderate

(a) Image sequence (b) Inter frame SIFT flow (c) Summation of (b) over 100 frames

Figure 4.1: The accumulated optic flow as a feature.

(a) Image sequence (b) Inter frame intensity change (c) Summation of (b) over 100 frames

Figure 4.2: The accumulated intensity changes as a feature.

wind, raindrops can be considered static over a few seconds. As default, we calculate over 100 frames which is about 4 seconds for the frame rate of 24 fps. Figs. 2.14.c and 4.2.c show examples of the two accumulated features.

We employ both features to have optimal accuracy. If time is a concern, however, we can use only intensity change.

4.1.2 Refined Detection

Having calculated the features, we use level sets [54] to identify raindrops. First, a convolution with Gaussian (σ = 2 pixels by default) is employed to reduce noise. Then, level sets are calculated, as illustrated in Fig. 4.3. Specifically, for the normalized 2D feature, we calculate the level-sets range from -2 to 2 with the step 0.05.

The following criteria are applied further for determining raindrop areas:

1. Feature threshold. As analyzed previously, raindrop areas should have smaller feature values. Hence, we normalized the accumulated feature with the mean

Figure 4.3: The detection pipeline.
Our method can work in real time if using only the intensity change.

value 0 and variance 1. In our experiment, those pixels with feature values less than -0.7 are considered to be raindrop pixels.

2. Smoothness. As analyzed in Sec. 3.1, (Eq. 2.1), raindrop contours usually have a smoothness value at 2π. Thus, we set the threshold for smoothness as 2.5π.

Note that, unlike [71], we do not utilize the closure explicitly, since it is already represented by the smoothness, which cannot be defined to non-closed lines. We also do not use size, as it varies significantly. Fig. 4.3 shows the detecton pipeline. For each detection, we accumulate the feature for the past 4 seconds and compute the level sets to detect raindrops. The overall detection algorithm is described in Algorithm. 1.

4.1.3 Real Time Detection

The detection method can work in real time if we use only the intensity change as the feature. We ran our program on a 3.1GHz CPU and Matlab with no parallelization. The video was 1280×720, 24fps. Accumulating the feature took 0.0086s per frame, which was 0.10s for 12 frames. Gaussian filter took 0.04s. The level sets took 0.22s. Selecting contours took 0.06s. The overall computing time for each detection phase was 0.42s.

Algorithm 1 Raindrop detection

```
Default parameter settings
  Video: 1080 × 720, 24fps
  Feature accumulating period: 4s(96frames)
  Number of detection phases:  2 per second
  Feature threshold:
    −0.7 for intensity change
    −0.4 for optic flow
  Smoothness threshold: 2.5π
while (not video end)
  compute the feature for new frames
  Accumulate the feature in specified period
  if (Detection phase)
    reduce noise of feature, σ = 2 Gaussian filter
    normalize feature to average = 0, variance = 1
    calculate level sets of the feature image.
    for (all contours)
      if (feature < threshold
          & smoothness < threshold )
          This contour circles a raindrop
      end
    end
    Displace result for current detection phase
  end
end
```

4.2 Raindrop Removal and Image Restoration

Existing methods try to restore the entire areas of the detected raindrops by considering them as solid occluders [46, 69]. In contrast, we try to restore the raindrop areas from the available information about the environment whenever possible. Based on Eq. (2.34), we know that some areas of a raindrop completely occludes the scene behind, however the rest occludes only partially. For partially occluding areas, we restore them by retrieving as much as possible information of the scene, and for completely occluding areas, we recover them by using a video completion technique.

4.2.1 Restoration

A blurred image can be recovered by estimating $I_e(x, y)$ in Eq. (2.34), in the condition that the blending value is moderate, *i.e.*, $\alpha(x, y) < 1$. To do this, we first have to calculate α in Eq. (2.34). Note that, based on our previous detection phase, the positions and shapes of raindrops on the image plane are known. Using the out-of-focus blur model in Fig. 2.15, the diameter ℓ of the equivalent light path area on the image plane is given by:

$$\ell = \frac{(D-d)}{(D-f)}\frac{f^2}{Od}, \tag{4.1}$$

where f is the focal length. O is the relative aperture size (also called f-stop) which can be found in the camera setting. D can be assumed to be infinite, and d is estimated empirically (we assumed constant throughout our experiments). The derivation of Eq. (2.36) can be found in the literature of depth from defocus [55]. Thus, a circle centered at (x, y) with diameter ℓ on the image plane can be drawn, as shown in Figs. 2.16 and b′. The blending coefficient $\alpha(x, y)$ is the proportion of the circle that overlaps with the raindrop.

Having obtained α, we recover I_e from the frequency domain. According to Eq. (2.41), the high frequency component of raindrop I_r is negligible. Thus, for frequency higher than a threshold ω_{th}, we have:

$$\mathcal{I}_e(x, y, \omega) = \frac{1}{1-\alpha(x, y)}\mathcal{I}(x, y, \omega), \quad \omega > \omega_{th}, \tag{4.2}$$

where $\mathcal{I}(x, y, \omega)$ is the Discrete Cosine Fourier Transform (DCT) of $I(x, y, t)$ on N consecutive frames. ω_{th} is set as $0.05N$ as default. As for the low frequency component, we replace it with the mean of its spatial neighborhood (from only the non-raindrop

Algorithm 2 Raindrop removal

if (default)
  $N = 100$, $\omega_{th} = 0.05N$, $\Delta x = \Delta y = \pm 1$pixel
  $th1 = 250$, $th2 = 40$
end
Load N continuous frames
Calculate $\alpha(x, y)$ for each pixel $I(x, y, \cdot)$.
if ($max(I(x, y, \cdot)) > th1$ & $\alpha(x, y) > 0$) $\{(x, y)$ is glare$\}$
for (non-glare pixels and $0 < \alpha(x, y) < 0.9$)
 for ($(R; G; B)$ channel separately)
  while ($\exists$ pixel unprocessed)
   Find pixel with smallest α $(I(x, y, \cdot))$
   Find neighbors of (x, y) in $(x + \Delta x, y + \Delta y)$
   Remove neighbors (intensity difference $> th2$)
   Do DCT: $\mathcal{I}(x, y, \omega) = I(x, y, t)$
   $\mathcal{I}(x, y, \omega_{th} : N) = \frac{1}{1-\alpha(x,y)} \mathcal{I}(x, y, \omega_{th} : N)$
   $\mathcal{I}(x, y, 1 : \omega_{th}) = mean(\mathcal{I}(x + \Delta x, y + \Delta y, 1 : \omega_{th}))$
   Do inverse-DCT
  end
 end
end
Repair the remaining areas using an inpainting method.

pixels or the already restored pixels):

$$\mathcal{I}_e(x, y, \omega) = \text{mean}(\mathcal{I}(\text{x} + \Delta\text{x}, \text{y} + \Delta\text{y}, \omega)),\ \omega \leq \omega_{\text{th}}, \tag{4.3}$$

where $(x + \Delta x, y + \Delta y), \Delta x, \Delta y \leq 1$ pixel are spatial neighborhood of (x, y). When averaging, we exclude neighboring pixels that have intensity differences larger than 40 (in 8-bit RGB value). By combining Eqs. (4.2) and (4.3), and performing inverse-DCT, we recover $I_e(x, y, t)$.

4.2.2 Video Completion

Having restored the partially occluding raindrop pixels, there are two types of remaining areas to complete:

- When α is close or equal to 1.0, I_e will be too scarce to be restored, as shown in Eq. (4.2). Because of this, we do not restore pixels with $\alpha > 0.9$.
- When there is glare, the light component from raindrop will be too strong and therefore saturated.

For those areas, we adopt Wexler *et al.*'s [67] space-time video completion method. As discussed in the related work, the method [67] only assumes that missing data reappears elsewhere in the video, which is most likely to be satisfied in outdoor scenes. The overall algorithm of our proposed raindrop removal algorithm is shown in Algorithm 2.

4.3 Experiments and Applications

We conducted quantitative experiments to measure the accuracy and general applicability of our detection and removal method. To show the benefits of our method, we include two real applications of our method on motion estimation and structure from motion. Results in video are included in the supplementary material.

4.3.1 Quantitative analysis on detection

We evaluated how raindrop size, blur, motion, scene complexity affect the detection using synthetic data, and estimated the optimal parameters. We also conducted the detection on various real scenes and compared the performance with that of the state-of-art methods. We use the precision-recall curve for our evaluation, where precision is defined as the number of the correct detection divided by the number of all the detection, and recall as the number of correct detection divided by the number of the detectable raindrops.

Raindrop size and blur As discussed in Sec. 3.2, our detection method is based on the fact that raindrops behave like a fish-eye lens and contract the environment. Obviously, a larger raindrop contracts less than a smaller raindrop does. Hence, raindrop physical size, which is limited by the raindrop tensor, affects the contraction ratio. Moreover, since our input is an image, the distance between the raindrop and the camera lens also affect the contraction ratio.

When raindrops are close to the lens, we need to consider the effect of out-of-focus blurring. Since, the closer to the lens, the more blur the raindrop is, implying lesser visibility. In our experiment, we explored how raindrop size and blur affect the detection accuracy. As illustrated in Fig. 4.4, we generated synthetic raindrops with fixed positions, but with various size and blurring levels. We fixed the detection thresholds. The thresholds of the normalized intensity-change and optic flow feature were set to -0.4 and -0.3, respectively, and the smoothness was set to 2.5π.

The detection precision and recall were evaluated using two methods: pixel-based and number-of-raindrop based methods. For the pixel-based method, the ground truth is the pixels with the raindrop blending coefficient $\alpha > 0.1$. Fig. 4.5 shows the results. As we can see, for highly visible raindrops, the detection precision and recall rate was not obviously affected by raindrop size. The recall rate was mainly affected by raindrop visibility. When the raindrops were too small and hardly visible, the detection recall

Figure 4.4: Synthetic raindrops with various size and blur levels.
The image size is 720×480, raindrop size (long axis) varies from 20 to 60 pixels, and the radius of the disk-blur-kernel varies from 0 to 40 pixels.

Intensity change

Precision: pixel based

Raindrop Size / Blur	20	30	40	50	60
0	0.53	0.76	0.86	0.88	0.94
5	0.70	0.90	0.97	0.98	0.99
10	0.76	0.95	0.99	0.98	1
20	0.44	0.93	0.95	0.98	1
30		0.39	0.74	0.93	1
40			0.64	0.79	1
50				0.83	1
60					1

Recall: pixel based

Raindrop size / Blur	20	30	40	50	60
0	0.92	0.97	0.98	0.98	0.84
5	0.83	0.89	0.91	0.91	0.91
10	0.60	0.71	0.74	0.76	0.77
20	0.10	0.25	0.39	0.47	0.46
30		0.03	0.04	0.09	0.15
40			0.03	0.03	0.01
50				0.02	0.01
60					0.00

Optical Flow

Precision: pixel based

Raindrop Size / Blur	20	30	40	50	60
0	0.58	0.56	0.61	0.68	0.72
5	0.74	0.75	0.77	0.83	0.90
10	0.95	0.93	0.94	0.95	0.97
20	1	1	1	1	1
30		1	1	1	1
40			1	1	1
50				1	1
60					1

Recall: pixel based

Raindrop Size / Blur	20	30	40	50	60
0	0.40	0.65	0.82	0.94	0.67
5	0.39	0.65	0.85	0.91	0.84
10	0.11	0.42	0.63	0.69	0.74
20	0	0.01	0.07	0.18	0.27
30		0	0	0.01	0.03
40			0	0	0.00
50				0	0
60					0

Figure 4.5: The precision and recall on detecting raindrops with various size and blur, evaluated at pixel level

(Fig. 4.4). The detection threshold was fixed for all of the data. The threshold of the normalized feature was set to 0.4 for the intensity change, and 0.3 for the optic flow. And the smoothness threshold was set to 2.5π

Intensity change

Precision: number of raindrops based

Raindrop size	20	30	40	50	60
Blur 0	0.59	0.74	0.87	0.74	1
5	0.65	0.80	0.95	0.91	1
10	0.61	0.87	0.95	0.83	1
20	0.27	0.82	0.72	0.83	1
30		0.36	0.57	0.87	1
40			0.63	0.76	1
50				0.78	1
60					1

Recall: number of raindrops based

Raindrop size	20	30	40	50	60
Blur 0	0.95	1	1	1	0.85
5	1	1	1	1	1
10	0.95	1	1	1	1
20	0.2	0.7	0.9	1	1
30		0.2	0.4	0.65	0.75
40			0.25	0.65	0.15
50				0.35	0.15
60					0.05

Optical Flow

Precision: number of raindrops based

Raindrop size	20	30	40	50	60
Blur 0	1	1	1	1	1
5	1	1	1	1	1
10	1	1	1	1	1
20	1	1	1	1	1
30		1	1	1	1
40			1	1	1
50				1	1
60					1

Recall: number of raindrops based

Raindrop size	20	30	40	50	60
Blur 0	0.6	0.75	0.85	0.95	0.8
5	0.6	0.8	0.95	1	0.9
10	0.25	0.65	0.8	0.95	1
20	0	0.05	0.05	0.65	0.75
30		0	0	0.05	0.2
40			0	0	0.05
50				0	0
60					0

Figure 4.6: The precision and recall on detecting raindrops with various size and blur, , evaluated at number of raindrops level

(Fig. 4.4). The detection threshold was fixed for all of the data. The threshold of the normalized feature was set to 0.4 for the intensity change, and 0.3 for the optic flow. And the smoothness threshold was set to 2.5π

rate dropped, and when the raindrops were blurred, their visibility decreased and the recall rate went down accordingly.

When evaluated by the number of pixels, the precision rate was higher on detecting larger raindrops. When evaluated by the number of raindrops, however, the precision rate was about the same for raindrops with any size. As the raindrop visibility decreased, the precision did not drop drastically, which indicated a low false alarm rate of our method.

Raindrop motion and detection latency As discussed in Sec. 5, our features are more accurate if they are accumulated overtime. In our experiment, we accumulated the features over 100 frames, which took around 4 seconds for a video with 24 fps. Hence, we assumed the raindrops need to be static within 4 seconds.

We investigated the tolerance of our method on detecting raindrops which is not quasi-static. As illustrated in Fig. 4.7, we generated synthetic raindrops with controlled motion speed. The raindrop size was 40 pixels and the raindrops were blurred with a 5 pixel disk kernel. The speed of raindrops varied from 0 to 4 pixels/frame (0 to 100 pixels per second).

Accumulating features will increase the distinction between raindrop and non-raindrop areas. However, when raindrops are moving, this is inapplicable anymore. Hence, we need to know how many frames needed to reliably detect raindrops robustly. An example is illustrated in Fig. 4.8. Here, the threshold for the normalized intensity change and optic flow features were set to 0.4 and 0.3 respectively. The raindrop parameter was set to 60 pixels to 120 pixels. The smoothness was set to 2.5π. The precision and recall of all data is listed in Fig. 4.9.

As shown, when raindrops are quasi-static, the detection accuracy was stable. The detection accuracy dropped significantly when using less than 10 frames. When using 100 frames and the raindrop moving speed was less than 0.4 pixel per frame (10 pixel per second), the detection accuracy was considerably stable. However, when the speed was increased to more than 0.4 pixel per frame, accumulating less than 100 frames increased the accuracy. In this experiments, the optimal number of accumulated frames was 20. The limit raindrop speed of our method was 4 pixel per frame (100 pixel per second). When raindrops moves faster and 4 pixels per frames, our method failed to detect them. Fortunately, 4 pixels per frames is considerably fast, which is rare in light rainy scenes.

Textureless Scenes Our method assumes the environment is sufficiently textured. Hence, in this experiment, we investigated how significant the absence of textures influences the detection accuracy. In this experiment, the threshold for normalized

Figure 4.7: Appearance of synthetic moving raindrops.
The raindrop size were 40 pixels and were blurred with a 5 pixel disk kernel. The speed of raindrops varied from 0 to 4 pixels/frame (100 pixels per second).

Figure 4.8: The influence of number of frames on feature accumulation. Row 1, the accumulated feature. Row 2, the detection result. Row 3, the detection result where the white area indicate raindrop. The raindrop size were 40 pixels (long axis) and blurred with a 5 pixel disk kernel, raindrops were moving with a speed 1.2 pixel per frame (30 pixel per second).

Figure 4.9: The precision on detecting raindrops with various raindrop speed and detection latency of Fig. 4.7

. The detection threshold was fixed for all the data. The normalized feature threshold was set to 0.4 for the intensity change, and 0.3 for the optic flow. The raindrop roundness threshold hold was set to 2.5π

Recall

Intensity change

Frame	100	90	80	70	60	50	40	30	20	10	8	6	5	4	3	2
Speed0	1	1	1	1	1	1	1	1	1	0.95	1	0.95	0.9	0.85	0.65	0.1
0.20	1	1	1	1	1	1	1	1	1	0.95	0.9	0.85	0.85	0.8	0.45	0.1
0.40	1	1	1	1	1	1	1	0.95	0.95	0.95	0.9	0.9	0.8	0.75	0.6	0.15
0.72	0.8	0.9	0.95	1	1	1	1	1	1	0.95	0.9	0.8	0.75	0.65	0.3	0
1.20	0.25	0.4	0.4	0.7	0.85	0.85	1	1	1	0.9	0.85	0.6	0.5	0.5	0.2	0
1.52	0.05	0.1	0.2	0.3	0.6	0.65	0.9	0.95	1	0.7	0.65	0.7	0.55	0.3	0.1	0
2.00	0	0	0.05	0	0.15	0.35	0.5	0.7	0.85	0.6	0.6	0.45	0.3	0.2	0.15	0
3.00	0	0	0	0	0	0	0.1	0.1	0.35	0.3	0.25	0.1	0.1	0.05	0.05	0
4.00	0	0	0	0	0	0	0	0	0.05	0.05	0.05	0.05	0.05	0	0	0

Optical Flow

Frame	100	90	80	70	60	50	40	30	20	10	8	6	5	4	3	2
Speed0	1	1	1	0.95	1	1	1	1	1	1	1	1	1	0.95	0.9	0.85
0.20	1	1	1	1	1	1	1	1	1	1	1	1	1	0.95	0.95	0.8
0.40	0.9	0.85	0.85	0.9	0.95	1	0.95	1	1	1	1	1	1	0.95	0.85	0.9
0.72	0.9	0.9	0.85	0.9	0.95	0.95	0.9	1	0.95	0.95	0.95	0.9	0.95	0.9	0.8	0.9
1.20	0.6	0.65	0.75	0.65	0.75	0.85	0.85	1	1	0.95	1	0.95	0.85	0.8	0.85	0.85
1.52	0.65	0.7	0.75	0.65	0.75	0.8	0.85	1	1	0.95	0.95	0.9	0.95	0.9	0.9	0.85
2.00	0.65	0.6	0.55	0.55	0.65	0.75	0.75	0.9	1	1	0.95	0.95	1	0.85	0.85	0.85
3.00	0.6	0.6	0.55	0.55	0.6	0.7	0.85	0.9	1	0.95	0.95	0.95	0.85	0.85	0.85	0.95
4.00	0.15	0.2	0.15	0.2	0.35	0.3	0.35	0.35	0.65	0.75	0.85	0.85	0.75	0.7	0.75	0.65

Figure 4.10: The recall on detecting raindrops with various raindrop speed and detection latency of Fig. 4.7.

The detection threshold was fixed for all the data. The normalized feature threshold was set to 0.4 for the intensity change, and 0.3 for the optic flow. The raindrop roundness threshold hold was set to 2.5π

Figure 4.11: Gaussian blur on a scene
with σ varying from 0 to 10. The patch size is 120×120 pixels.

Figure 4.12: The accumulated feature using intensity change and optic flow on textured and textureless scenes.
100 frames are used for accumulation.

features was set to 0.4 for the intensity change while 0.1 for the optic flow. The smoothness was set to 2.5 π, and features were accumulated over 100 frames. As illustrated in Fig. 4.11, we performed Gaussian blur on the scene, with σ varying from 0 to 10, and generated synthetic raindrops with a fixed size (40 pixels) and position.

As illustrated in Fig. 4.12, when the scene was textureless, the intensity change was affected. The non-raindrop areas changed less on a less textured scene. The optic flow, however, was not affected, because optic flow is based on the motion of texture. In addition to that, most of the state of the art optic flow algorithms adopt the coarse-to-fine strategy in estimating the flow. The coarse estimation provides a robust global estimation while the fine estimation provides the accurate and detailed estimation. Thus the texture-less input only affects OF feature. The precision recall is listed in Fig. 4.13, which shows that when $\sigma > 5$, the accuracy of the intensity change based method dropped because the feature on a textureless scene was less distinctive, and the false alarm rate increased.

Figure 4.13: The precision and recall of raindrop detection on textured and textureless scenes.
The threshold for normailzed features was set to 0.4 for the intensity change and 0.1 for the optic flow. The raindrop parameter was set to 60 pixels to 160 pixels. The roundness threshold was set to 2.5π. Features were accumulated over 100 frames.

4.3.2 Quantitative Comparison on Detection

Real Scenes with Groundtruth We created a real data by dropping water on a transparent panel as the ground truth and taking videos in the real world. We had a few scenarios for the experiments. Experiment 1 included the disturbance of the light sources. Experiment 2 emphasized on the varying shape and size of raindrops. Experiment 3 focused on significantly blurred raindrops, and experiment 4 included glare. The input and results are shown in the first four columns in Fig. 4.14.

We compared our method with Eigen *et al.*'s [10], Roser *et al.*'s [46] and Kurihata *et al.*'s [32] method. Yamashita *et al.*'s [70, 69] methods require stereo cameras or a pan-tile camera and were, thus, not included in the comparison. The results are shown in the last two columns of Fig. 4.14.

We used the precision-recall curve to quantitatively analyze the performances. The results for each experiment are shown in Fig. 4.20. According to the results, both of our proposed method outperformed the existing methods. By combining IC with OF, we obtained the best performance to detect all of the raindrops (because of IC) while keeping a low false alarm rate (because of OF). The detection using the intensity change performed best. Unlike the existing methods that only detect the center and size of raindrops, our proposed method can detect raindrops with a large variety of

shapes. Our method also achieved high robustness in detecting highly blurred and glared raindrops.

Real scenes without groundtruth Fig. 4.14 shows the results of our detection method in the following 3 situations: (1) A daily use hand held camera, as in experiments 1-4. (2) A vehicle-mounted camera, which is widely used for navigation and data collection. (3) A surveillance camera which was stuck into a fixed location. Our method outperformed the existing methods in the all three situations as shown in the figure.

4.3.3 Raindrop Removal

Quantitative tests on raindrop removal As illustrated in the first two columns of Fig. 4.22, the synthesized raindrops were generated on a video, and used as an input. Our method was compared with the method proposed by Wexler *et al*. [67]. In [46], there is insufficient description for the removal algorithm and thus it was not compared here. The results are shown in the last four columns of Fig. 4.21.

As shown in Fig. 4.21, for the quantitative evaluation, we ran each of them on 100 continuous frames and calculated the average error per pixel for each frame. The same as Wexler *et al*. [67], the error was calculated on both the 8 bit $(R; G; B)$ value and spatial-temporal gradients $(dx; dy; dt)$. The proposed method benefits from the restoration in all the 3 situation. Using the same computer, our method needed 5 seconds per frame to remove raindrops, and Wexler *et al*.'s needed 2 minutes.

Quantitative evaluation We show a few results of removing raindrops in videos taken by a handle held camera and a vehicle-mounted camera, as shown in the first and second row of Fig. 4.25 we can see the significant improvement. To demonstrate the performance of our raindrop removal method, the manually labeled raindrops were also included.

Overall evaluation The overall automatic raindrop detection and removal results in videos taken by a hand held camera and a car mounted camera are shown in the third row of Fig. 4.25, where we can see the significant visibility improvement.

4.3.4 Applications

To show the benefits of our method, we applied it to two common applications in computer vision: motion field estimation and structure from motion.

Figure 4.14: The detection results of a night scene using our methods and the existing methods.

Figure 4.15: The detection results of raindrops with arbitrary shapes using our methods and the existing methods.

Figure 4.16: The detection results of raindrops with arbitrary size using our methods and the existing methods.

Figure 4.17: The detection results of raindrops with highlights using our methods and the existing methods.

Figure 4.18: The detection results of video taken by a car-mounted camera using our methods and the existing methods.

Figure 4.19: The detection results of video taken by a surveillance camera using our methods and the existing methods.

Figure 4.20: The precision(R)-recall(R) curves of our methods and the two existing methods.
The thresholds of our normalized features are labeled.

Figure 4.21: Average $(R; G; B; dx; dy; dt)$ error
of recovering 100 continuous frames of the experiment shown in Fig. 4.22.

Figure 4.22: The raindrop removal results using our methods and the method of Wexler *et al*. [67] on a clear driving scene.

Figure 4.23: The raindrop removal results using our methods and the method of Wexler *et al*. [67] on a crowed driving scene.

Figure 4.24: The raindrop removal results using our methods and the method of Wexler *et al*. [67] on a textured scene.

Figure 4.25: The raindrop removal using our method on a video taken by hand-held camera.
First row: the input sequence. Second row: the removal result with the raindrops manually labeled. Third row: the removal result with the raindrops automatically detected.

Figure 4.26: The raindrop removal using our method on a video taken by a car-mounted camera.
First row: the input sequence. Second row: the removal result with the raindrops manually labeled. Third row: the removal result with the raindrops automatically detected.

Motion estimation Adherent raindrops occlude the background and their motion is significantly different from the background motion. By removing the raindrops, we show that the motion in the background can be correctly estimated. We demonstrate the improvement on various scenes shown in Fig. 4.27. SIFT-flow [34] was used for the motion estimation; although, any optic flow algorithm can also be used.

In the scene of Fig. 4.27, we applied our method to a synthetically generated raindrop. As can be seen, the motion field of the raindrop images (the second row) is significantly degraded compared to that of the clear images (the first row). Having removed the raindrop, the motion field becomes more similar to that of the clear images (the third row). In the scene of Fig. 4.28, the images have global motion because of the shaking camera. Although the estimation on the repaired images reflects the global motion, the estimation on raindrop images is also significantly affected. In the last scene (Fig. 4.29), the car-mounted camera was moving forward and the motion on the repaired images correctly reflects the camera motion.

Structure from motion (SfM) Adherent raindrops move along with the camera adversely affect the camera parameter estimation. As a result, they also negatively affect the accuracy of the dense depth estimation. Hence, we expected that with raindrops being removed, the robustness of the camera parameter and depth estimation associated with the structure from motion technique can be improved. As illustrated in Fig. 4.30, we performed the structure from motion method by Snavely *et al*. [53]. We used a clear video, a video with adherent raindrop and a repaired video as inputs. Samples of those videos are shown in the second row of Fig. 4.22. As can be seen, the repaired video provides better results than that of the raindrop video on both the camera parameter estimation and dense depth estimation.

Figure 4.27: Motion estimation using a clear video, raindrop video and repaired video on a synthetic data.

Figure 4.28: Motion estimation using a clear video, raindrop video and repaired video on a real video taken by a hand-held camera.

Figure 4.29: Motion estimation using a clear video, raindrop video and repaired video on a real video taken by a car-mounted camera.

Figure 4.30: Structure from motion using a clear video, raindrop video and repaired video.
The input view are shown in the second row of Fig. 4.22.

4.4 Summary

We have introduced a novel method to detect and remove adherent raindrops in video. The key idea of detecting raindrops is based on our theoretical findings that the motion of raindrop pixels is slower than that of non-raindrop pixels, and the temporal change of intensity of raindrop pixels is smaller than that of non-raindrop pixels. The important idea of our raindrop removal is to solve the blending function with the clues from detection and intensity change in a few consecutive frames, as well as to employ a video completion technique only for those that cannot be restored. To our knowledge, our automatic raindrop detection and removal method is novel and can benefit many applications that possibly suffer from adherent raindrops.

Chapter 5

Single Image Stereo Using Water Drops

Depth from real images is often crucial information for many applications in computer graphics. Many algorithms particularly in computer vision have attempted to extract depth with various cues [25, 21, 12]. Unlike all these algorithms, in this paper, we explore a new possibility of using water drops adhered to window glass or a lens to estimate depth.

Water drops adhered to glass are totally transparent and convex, and thus each of them acts like a fisheye lens. As shown in Fig. 5.1.a, water drops' locations are normally scattered in various regions in an image, and if we zoom in, each of the water drops displays the same environment from its own unique point of view. Due to the proximity to each other, some have similar imageries, but some can be relatively different, particularly when the water drops are apart in the image. Therefore, if we can rectify each of the water drop imageries, we will have a set of images of the environment from relatively different perspectives, opening up the possibility of extracting the depth from these water drops, which is the goal of this paper.

To be able to achieve the goal, we need to rectify each water-drop's imagery, so that planar surfaces look flat. Rectifying water drops, however, is problematic. In contrast to existing work in catadioptic imaging, which assumes the geometry of the sphere is known a priori, water drops shapes can vary in a considerable range. To resolve this problem, we need to consider two physical properties of water drops. First, a static water drop has constant volume, and its geometric shape is determined by the balance between the tension force and gravity. Because the water drop is in balance, it minimizes the overall potential energy, which is the sum of the tension energy and

the gravitational potential energy. Based on this property, we introduce an iterative method to form the water-drop geometric shape. However, from a single 2D image, the volume cannot be directly obtained, implying that, based only on the first property, we do not know the thickness of the water drop. Second, water drops' appearance is determined by their geometric shape and also the total reflection, which occurs near the boundaries and triggers a dark band. We found that a water drop with a greater volume will have a wider dark band. Thus, we introduce a volume-varying-iteration framework that estimates the volume that best fit to the appearance. Having known the complete 3D shape of water drops, we perform the rectification on each of them by backward raytracing. With each of the water-drop images is rectified, we estimate the depth using the stereo concept. In addition, we also apply image refocusing as well as image stitching. Figure 5.1 shows the pipeline of our proposed method.

Contribution In this chapter, we introduce a new way to recover depth using water drops from a single image. We also propose a novel method to reconstruct the 3D geometry of water drops by utilizing the minimum surface energy and total reflection. Aside from estimating depth, we also apply image refocusing and image stitching through the information provided by water drops.

The rest of the paper is organized as follows. Section 2 discusses related work in depth estimation, water modeling and shape from transparent objects. Section 3 explains the theory behind the water-drop physical properties. Section 4 introduces the methodology of the geometry estimation, as well as water-drop image rectification. Section 5 shows the three applications on stereo, image refocusing and image stitching. Section 6 shows the experimental results and evaluation. Section 7 concludes the paper.

5.1 Image Formation

This section focuses on the theoretical background and the modeling of water drops. We first discuss briefly the image formation that shows the correlations between the environment, water-drops and camera. We subsequently model the raindrop 3D geometry, particularly the concept of minimum energy surface. Then, based on the image formation and the raindrop geometry, we discuss the total reflection inside water drops that is necessary to determine the water-drop's volume. All these aim for water-drop image rectification.

(a) Single image of water drops (b) Water drop silhouettes

(c) 3D reconstruction

(d) Dewarping

(e) Stereo and refocus

Figure 5.1: The pipeline of the proposed method.

Figure 5.2: Model of the image system.
(a) The light path model, assuming the camera is a pinhole camera. (b) Appearance of the environment when the camera focuses on the environment. (c) Appearance of water drops. (d) Image obtained by the camera.

Fig. 5.2 illustrates the image formation of an environment whose refracted rays pass through two water drops before hitting camera's image plane. Unlike the conventional image formation, water drops influence the trajectories of the passing rays, where each of the water drops acts like a fisheye lens or a catadioptric camera that warps the images. Fig. 5.2.c shows the warped images by a few water drops. Assuming we have a few water drops that are apart to each other, the imageries of the water drops will be slightly different to each other although the environment is identical, as shown in Fig. 5.2.d.

From the diagram in Fig. 5.2, we can conclude that the image captured by the camera through water drops is determined by three interrelated factors: (1) the depth of the environment, which we aim to estimate, (2) the three dimensional shape of water drops, which determine how the light rays emitted from the environment are refracted and, (3) camera's intrinsic parameters, which are assumed to be known. Therefore, to recover the depth of the environment, we need to obtain the 3D shape of every water drop.

5.2 Methodology

Based on the modeling in Chapter 2.1, in this section, we introduce the detailed algorithm for rectifying images of water drops. As illustrated in Fig. 5.1, in general, it have three main steps: (1) water drop detection, (2) water drop 3D shape reconstruction by minimizing energy surface, and (3) image rectification.

5.2.1 Water Drop Detection

Water-drops appearance is highly dependent on the environment, causing some level of complexity to detect them generally. Fortunately, in our case, we can assume that water drops are in focus and thus the environment is rather blur. For this, we can apply edge detection for detecting water drops. As illustrated in Fig. 5.3, we used Matlab edge detection to detect water drops. Having filled the area of the detected water drops, we select those that are sufficiently large, with diameter is greater than 20% of the image size. This is to ensure that rectified images are not too small.

(a) Input (b) Canny

(c) Fill hole (d) Selected adhesion area

(e) Selected appearance

Figure 5.3: Selecting water drops from a single image.

5.2.2 Water-Drop 3D Shape Reconstruction

Mesh representation and Initialization To reconstruct the 3D shape of water drops, we first represent the water surface using a parameterized mesh. Referring to Eq. (2.3), we can describe a surface as: $\mathcal{S} = \{z(i,j), (i,j) \in \Omega_R\}$, where (i,j) are the location of a pixel in the water drop area. Accordingly, the area of Ω_R is defined as: $B = \sum_{(i,j)\in\Omega_R} 1$, where 1 is the unit for a pixel's area.

At this initialization, the volume of the water drop is unknown yet. Thus, we give an initial guess of the volume as:

$$V = \alpha B^{\frac{3}{2}}, \tag{5.1}$$

where α is called as the volume coefficient and is set as 0.30 as default. Based on the equation, when the area B increases in square rate, the volume will increase in cubic rate.

We initialize the mesh as a cylinder by defining:

$$z(i,j) = \alpha B^{\frac{1}{2}}, (i,j) \in \Omega_R. \tag{5.2}$$

Figure 5.4.a shows an example.

Iteration with fixed volume We solve the constrained minimum energy surface using the iterative gradient descend. For iteration t we update the mesh with three steps: tensor energy update, gravity update, and volume update. This strategy is an extension of the smooth surface reconstruction proposed by [42].

Step 1: Tension energy update attempts to construct the surface as smooth as possible:

$$z_{t+1} = z_t - \tau \cdot \frac{\mathrm{d}E_T(\mathcal{S})}{\mathrm{d}z_t}, \tag{5.3}$$

where τ controls the update speed. We set $\tau = 0.5$ as default, σ is the tension coefficient introduced in previous section. We define:

$$\frac{\mathrm{d}E(\mathcal{S})}{\mathrm{d}z_t} = -\sigma \mathrm{div}\left(\frac{1}{\sqrt{1+|\nabla z_t|^2}}\nabla z_t\right), \tag{5.4}$$

where div is the divergence.

Tension coefficient of water in room temperatue is $\sigma = 73000N/m$. The size of a image pixel can be inferred from the image size and focus length.

Step 2: Gravity update tries to increase the height for the mesh points that will lower potential energy:

$$z_{t+1} = z_t - \tau \cdot \frac{\mathrm{d}E_G(\mathcal{S})}{\mathrm{d}z_t}, \tag{5.5}$$

Figure 5.4: Iteration of water drop 3D shape with a fixed volume.

which can be expressed as:

$$z_{t+1}(i,j) = z_t(i,j) - \tau\rho g((y_g - i)\cos\theta_y + (x_g - j)\cos\theta_x), \tag{5.6}$$

where (x_g, y_g) is the geometry center of the water drop:

$$x_g = \frac{1}{B}\sum_{(i,j)} z(i,j)\cdot j, \quad y_g = \frac{1}{B}\sum_{(i,j)} z(i,j)\cdot i \tag{5.7}$$

The physical coefficients has the number: $\rho = 1Kg/L$ and $g = 9.8m/s^2$.

Step 3: Volume update. After the update of tension and gravity, we check the current volume and compare it with the targeted volume V, and then readjust the volume by adding the same value to all the mesh point:

$$z_{t+1} = z_t + \left(\frac{V - \sum_{(i,j)\in\Omega_R} z_t(i,j)}{B}\right) \tag{5.8}$$

As default, assume the water drop size is 1, we set the converge threshold to $1e-8$, and run up to 4000 iterations if it does not converge. Fig. 5.4 shows an example of the iteration progress. We will evaluate the computational time in the experiments.

Iteration with varying volume After the iteration with a fixed volume, the surface normal is obtained, and now we can evaluate the brightness values near the dark ring.

As discussed in Section 3.3, the surface geometry allows us to find the dark ring. Technically, we find the ring close to the circle of the critical angle: $\theta = \theta_N \pm 5^o$. As illustrated in Fig. 2.5.c. If the estimation is correct, the ourter half of the ring are in the dark area where the illuminance is close to 0. And the inner half of the ring are in the close to critical angle area where the illumination are determined by the refraction coefficient as Eq. (2.13). Integrating along the radial direction of the ring, we know the average refraction coefficient of the ring, $\mathcal{T}_r$, should be approximately equalt o 0.241. Consequently, the average brightness of the ring, I_r, is:

$$I_r = 0.241 I_b, \tag{5.9}$$

where I_b is the average brightness of the non water-drop areas.

In Fig. 5.5, we sample the brightness of the estimated ring. As can be seen, when the volume is underestimated, the dark ring is wider than the real one which results in less bright pixels. On the contrary, when the volume is overestimated, the dark ring

(a) Estimated geometry with varying volume

(b) Estimated ring of critical angle

Figure 5.5: Registration between the observed and estimated dark ring. Left: Underestimated volume. Middle: Correctly estimated volume. Right: Overestimated volume.

is smaller than the real one, resulting in a greater brightness value. With the above analysis, we update the volume every 400 iteration (as default):

$$V_{t+1} = V_t + \tau_r \cdot V_t \cdot (1 - \frac{I_t}{I_r}), \tag{5.10}$$

where I_t is the sampled brightness, I_r is the targeted brightness value, and τ_r is a weighting coefficient which is set to 0.5 as default.

5.2.3 Rectification of Water-Drop Image

With the location and 3D shape of water drops are estimated, using the camera model (Fig. 5.2) and the refraction model (Fig. 2.6.b), it is possible to apply backward raytracing from the camera to water drops and then to the environment. Currently, (in Fig. 5.2), the only unknown parameter is depth of the object in the environment D. We initialize it as a sufficient large value. Fig. 5.6 shows examples of the rectified water drop images.

According to Eq. (2.10) and Eq. (2.11), with the water drop geometry obtained, we can compensate the brightness values according to the refractive coefficient $\mathcal{T}$. Figure 5.7 shows an example of the brightness compensation.

5.2.4 Depth from Stereo

Once the water drop images are rectified, we can select a set of water drop images and apply stereo to estimate depth. Our water-drop based stereo does not require the camera parameter estimation, since for each water drop, we can consider its center (Eq. (5.7)) to be the camera location, and the camera aspect is along the z-axis.

Figure 5.6: Rectified water drop images.

Figure 5.7: Illuminance compensation of water drop images.
In (b), outer size of the drop is kept for better visualization, the value should be zero. In (d), area near the total reflection is not accurate because devided by a close to zero number. It is kept for visualization only.

5.3 Experiments and Analysis

We conducted some experiments using both synthetic data and real data to evaluate and analyze the performance of our method. In the experiment, we were mainly evaluating the estimated 3D shape of the water drop, and the depth estimation.

5.3.1 3D Shape Reconstruction and Image Rectification

To evaluate the accuracy of the 3D shape of water drops, we utilized synthetic data. We cannot use real data for quantitative evaluation, since unlike opaque objects, for water we cannot use automatic 3D acquisition systems, like the laser range finder. We will use real images, for the evaluation of the image rectification. Some of the real images were taken by ourselves and some were downloaded from the Internet.

Fig. 5.8 shows the generated synthetic water drops with a variety of boundaries. A quantitative evaluation was performed by comparing the ground truth 3D shape and estimated one. The error is normalized as the percentage of the scale of the water drop. As can be seen, the reconstruction error is less than 3% even for the most irregular water drop.

Figure 5.12 shows a collection of the rectified water drop images from real data. The input image is cropped for better visualization, yet the camera center is not at the cropped image center.

We implemented our method in Matlab and timed the performance without parallelization. For water 3D shape estimation, the time varied depending on the water drop volume and the mesh resolution. Table 5.1 shows the computation time of varying volume and fixed mesh resolution. And Table 5.2 shows the time of varying mesh resolution. At typical case, the resolution of mesh is set to 200×200 and the reconstruction time is about 10s.

We also mention that, because each of the water drop reconstruction are performed separately, we can simply parallelize each of the task. Thus, the overall computation time does not increase with the number of water drops.

5.3.2 Depth Estimation

Figure 5.8: Quantitative evaluation of water surface reconstruction and rectification assuming a round water drop.

Figure 5.9: Quantitative evaluation of water surface reconstruction and rectification assuming a eclipse water drop.

Figure 5.10: Quantitative evaluation of water surface reconstruction and rectification assuming a hanged water drop.

Figure 5.11: Quantitative evaluation of water surface reconstruction and rectification assume a irregular water drop.

Figure 5.12: Rectification of real water images using our data.
There is slant between the background and the water drop in some data, however the rectified image is not necessary to be rectangle.

Figure 5.13: Rectification of real water images using data downloaded from the Internet.

Figure 5.14: Stereo using two dewarped water drop images.

Table 5.1: Computation time for water drop 3D reconstruction with varying volume.

Volume (α)	0.05	0.1	0.2	0.3
Iterations	1300	1600	2600	3200
Time (s)	5.5	7.1	11.7	15.1

The mesh resolution is fixed to 200×200.

Table 5.2: Computation time for water drop 3D reconstruction with varying mesh resolution.

Mesh res.	50*50	100*100	200*200	400*400
Iterations	400	800	2600	5100
Time (s)	0.4	1.7	11.7	241

The volume is fixed to $\alpha = 0.2$.

Fig. 5.14 shows the generated synthetic data from the Middleburry data set [51]. As can be seen, the depth estimation result highly resembles the ground truth with only errors occur at object's boundaries.

The result on the real image is shown in Fig. 5.15. Although the water images are correctly rectified, the real image has certain level of out of focus blur from a real perspective camera, which eventually affected the overall quality of the layer segmentation. However, the result is promising for image refocusing.

Figure 5.15: Stereo using two rectified water drop images.

5.4 Discussion

In this paper, we had exploited the depth reconstruction from a single image with a few water drops. In our pipeline there are a few key steps: the water-drop 3D shape reconstruction, water-drop image rectification, and depth estimation using stereo. All are done using a single image.

We evaluated our method, and it shows that the method works effectively for both synthetic and real images. Nevertheless, there are still some limitations in it. One of the limitation is the common perspective camera which we have used in our experiment. Our method assumes the camera aperture is pin-hole so that the our-of-focus blurring of the perspective camera could be neglected. However, according to our experiments, we found it is necessary to perform image refocusing to improve the quality of the input image. Another possible limitation is that we did not use the advanced bundler for stereo. The water drop distortion is rather complex which cannot be assumed as radial/spherical distortion; thus, a special bundler is necessary.

Chapter 6

Discussion and Conclusion

6.1 Summary

In this thesis, focused on developing methods of automatic raindrop detection and removal. And we utilizes raindrop to perform a few image processing tasks. To achieve these goals, we have theoretically analyzed the imaging system with the presence of water drops. Based on our analysis, we have developed three automatic raindrop detection and removal systems. Further more, with the insights on properties of water drops, we developed an single image stereo system using water drops.

The first system utilizes the assumption of the smooth motion of camera/scene. The idea is to use long range trajectories to discover the motion and appearance features of raindrops locally along the trajectories. These motion and appearance features are obtained through our analysis of the trajectory behavior when encountering raindrops. These features are then transformed into a labeling problem, which the cost function can be optimized efficiently. Having detected raindrops, the removal is achieved by utilizing patches indicated, enabling the motion consistency to be preserved. Our trajectory based video completion method not only removes the raindrops but also complete the motion field, which benefits motion estimation algorithms to possibly work in rainy scenes. Experimental results on real videos show the effectiveness of the proposed method.

The second system is also based on the smooth motion which is a fast and robust method. It is principally based on sparse matching and interpolation. First, SIFT, which is robust to arbitrary motion, is used to efficiently obtain sparse correspondences in

neighboring frames. To ensure these correspondences are uniformly distributed across the image, a fast dense point sampling method is applied. Then, a dense motion field is generated by interpolating the correspondences. An efficient weighted explicit polynomial fitting method is proposed to achieve spatially and temporally coherent interpolation. In the experiment, quantitative measurements were conducted to show the robustness and effectiveness of the proposed method.

The third system is based on the contraction properties of water drops. The core idea is to exploit the local spatio-temporal derivatives of raindrops. First, we explicitly model adherent raindrops using law of physics, and then, detect them based on these models in combination with motion and intensity temporal derivatives of the input video. Second, relying on an analysis that some areas of a raindrop completely occludes the scene, yet the remaining areas occlude only partially, we remove the two types of areas separately. For partially occluding areas, we restore them by retrieving as much as possible information of the scene, namely, by solving a blending function on the detected partially occluding areas using the temporal intensity derivative. For completely occluding areas, we recover them by using a video completion technique. Experimental results using various real videos show the effectiveness of the proposed method.

Based on the experience on detecting and removing rain drops, we propose a novel single image stereo system which utilizes a common camera with a few water drops. The key idea is that water drops are totally transparent and convex, which can be considered as a fish eye lens. To rectify the water drop images, we utilize two physical properties. First: a static water drop, its volume is constant and its geometry is determined by the balance of the tension force and gravity; equivalently, its geometry minimize the overall potential energy which is the sum of the tension energy and the gravitational potential energy. Second: the water drops appearance is determined by its geometry and total reflection will happen near the boundary of the drop which will result in a dark band. With the geometry of water drops recovered, we rectify the drop images through ray-tracing. Based on a set of the rectified the image, we perform three image process tasks: stereo, refocus and scope extension. Quantitative experiments shows the effectiveness of the proposed system.

Because the proposed methods are based on different modeling and assumptions, they have different performances and applicabilities in varying situations. Table 6.1 is a summary of the applicabilities of the proposed methods.

Table 6.1: The proposed methods and their applicabilities.

Applicability	**Chapter 3 Trajectories**	**Chapter 4 Blend-in**	**Chapter 5 Ray-tracing**
Raindrop			
Shape	✓	✓	✓
Size	✓	✓	✓
Clear	✓	✓	✓
Blurred	✓	✓	
Input			
Video	✓	✓	✓
Image			✓
Camera			
Single camera	✓	✓	✓
Stable camera	✓	✓	✓
Shaking camera		✓	✓

6.2 Contributions

In this thesis, three complete algorithms to remove adherent raindrops in video are proposed. And an algorithm to perform single image stereo using water drop images.

For raindrops detection:

I. Algorithms which could detect raindrops with any size and shape are proposed.

II. Accuracy of our algorithm outperforms all existing algorithms.

III. Our proposed real-time computational efficiency which is essential for many outdoor vision tasks.

For video repairing:

I. Algorithm which could repair video with both spatially and temporally large missing area is proposed.

II. Case by case solution which could handle complex situations (complex motion and complex structure) in outdoor vision system is proposed.

III. Computational efficiency is achieved by using the proposed sparse matching based motion estimation.

For single image stereo:

I. A novel single image stereo method which utilized only water drops and a common 2D camera is proposed.

II. A novel single image liquid geometry estimation which utilized the minimum energy surface and total, reflection is proposed.

III The system enables more image processing tasks such as stereo and refocus.

6.2.1 Applications

Our algorithm coud benefit many other computer vision algorithms. As demonstrated by examples in this paper:

I. Motion estimation could be benefited from the restored vision.

II. Tracking, especially long range tracking could be benefited from the algorithm.

III. Multi-view stereo could be benefit by the proposed algorithms. As the underling technologies are tracking and matching.

For more general discussion, the proposed systems and algorithms could also benefit algorithms in many other areas. For example VR, automatic drive, and surveillance. All these areas many work in a rainy environment.

6.2.2 Relation with Neural Network

Theoretically, all the proposed methods are based on physical modeling of the water drops and other extrinsic feature. In a general machine learning view (or neural network view.) The detection could be considered as one specific case for a generalized pattern recognition work. Although, in general, this statement is true. However, practically, a general pattern recognition strategy based on appearance are rather difficulty. The difficulties are two folds.

First of all, a machine learning framework requires sufficient training examples. However, as mentioned in this thesis, for water drops, it is rater impractical to provide a dense enough sample base. First of all, the water drops are liquid which do not provide specific shapes. One might provide a few samples as possible shapes. However, it is theoretically not sound that such sample are compact. (Here compact means a discrete and limited sample set which could cover all the possible shapes within a given distance tolerance.) Similarly, and more disastrously, the water drops are transparent. To provide a compact appearance set for training means one need to sample every possible appearance of the outdoor environment. "Curse of dimension" is the classical word to describe the difficulty for this problem in neural network.

Secondly, even if the sample are provided for training, the training set are highly unseparable. This is because the water drops are totally transparent, the appearance of water drops are highly similar to the environment. Or equivalently, the raindrop and non-raindrop samples are highly mixed and cannot be separated by a few divisions in the feature space.

Bared on the reasons mentioned about, this paper provides a alternated strategy which avoids the problems in a general neural network framework. Based on the physical modeling, a generally linear separator (in Chapter 4) or a graph cut based label (in Chapter 3) could performance the detection efficiently. It is highly possible that other dedicated neural network strategies might working well based on the proposed features.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision(IJCV)*, 56(3):221–255, 2004.

[2] Simon Baker and Shree K Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196, 1999.

[3] P. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2-3):256–274, 2010.

[4] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on Pattern Analysis and Machine Intelligence (TPAMI)s*, 26(9):1124–1137, September 2004.

[5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Efficient approximate energy minimization via graph cuts. *IEEE transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(12):1222–1239, November 2001.

[6] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (TPAMI)*, 33(3):500–513, 2011.

[7] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. *ICCV*, 2013.

[8] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. *IEEE International Conference on Computer Vision*, 2003.

[9] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.

[10] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. *ICCV*, 2013.

[11] R. Fattal. Single image dehazing. *SIGGRAPH*, 2008.

[12] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):406–417, 2005.

[13] Richard P Feynman, Robert B Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Desktop Edition Volume I*, volume 1. Basic Books, 2013.

[14] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *IEEE International Conference on Computer Vision (ICCV)*, October 2009.

[15] K. Garg and S. Nayar. Photometric model of a raindrop. *CMU Technical Report*, 2003.

[16] K. Garg and S. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007.

[17] K. Garg and S. K. Nayar. Detection and removal of rain from videos. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 1:I–528, 2004.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] Jinwei Gu, Ravi Ramamoorthi, Peter Belhumeur, and Shree Nayar. Removing image artifacts due to dirty camera lenses and thin occluders. *ACM Transactions on Graphics (TOG)*, 28(5):144, 2009.

[20] T. Hara, H. Saito, and T. Kanade. Removal of glare caused by water droplets. *Conference for Visual Media Production*, 2009.

[21] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[22] Tal Hassner and Ronen Basri. Example based 3d reconstruction from single 2d images. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 15–15. IEEE, 2006.

[23] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *CVPR*, 2009.

[24] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232. ACM Press/Addison-Wesley Publishing Co., 1997.

[25] Katsushi Ikeuchi and Berthold KP Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1):141–184, 1981.

[26] J. Jia, Y. Tai, T. Wu, and C. Tang. Video repairing under variable illumination using cyclic motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):832–839, 2006.

[27] J. Jia, T. Wu, Y. Tai, and C. Tang. Video repairing: Inference of foreground and background under severe occlusion. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[28] Pushkar Joshi and Nathan A Carr. Repoussé: automatic inflation of 2d artwork. In *Proceedings of the Fifth Eurographics conference on Sketch-Based Interfaces and Modeling*, pages 49–55. Eurographics Association, 2008.

[29] Andrey V Kanaev, Weilin Hou, Sarah Woods, and Leslie N Smith. Restoration of turbulence degraded underwater images. *Optical Engineering*, 51(5):057007–1, 2012.

[30] L. Kang, C. Lin, and Y. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012.

[31] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2):147–159, February 2004.

[32] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, Hiroshi Murase, Yukimasa Tamatsu, and Takayuki Miyahara. Rainy weather recognition from in-vehicle camera images for driver assistance. *IEEE Intelligent Vehicles Symposium*, 2005.

[33] Marc Levoy, Billy Chen, Vaibhav Vaish, Mark Horowitz, Ian McDowall, and Mark Bolas. Synthetic aperture confocal imaging. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 825–834. ACM, 2004.

[34] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2006.

[35] M. Liu, S. Chen, J. Liu, and X. Tang. Video completion via motion guided spatial-temporal global optimization. *ACM international conference on Multimedia*, 2009.

[36] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[37] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[38] Jitendra Malik and Ruth Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International journal of computer vision*, 23(2):149–168, 1997.

[39] Gaofeng Meng, Ying Wang, Jiangyong Duan, Shiming Xiang, and Chunhong Pan. Efficient image dehazing with boundary constraint and contextual regularization. *ICCV*, 2013.

[40] Nigel Jed Wesley Morris. *Image-based water surface reconstruction with refractive stereo*. PhD thesis, University of Toronto, 2004.

[41] Omar Oreifej, Guang Shu, Teresa Pace, and Mubarak Shah. A two-stage reconstruction approach for seeing through water. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1153–1160. IEEE, 2011.

[42] Martin R Oswald, E Toppe, and Daniel Cremers. Fast and globally optimal single view reconstruction of curved objects. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 534–541. IEEE, 2012.

[43] K. Patwardhan, G. Sapiro, and M. Bertalmio. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Image Processing(TIP)*, 16(2):545–553, 2007.

[44] Mukta Prasad and Andrew Fitzgibbon. Single view reconstruction of curved surfaces. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1345–1354. IEEE, 2006.

[45] Srikumar Ramalingam, Peter Sturm, and Suresh K Lodha. Theory and calibration for axial cameras. In *Computer Vision–ACCV 2006*, pages 704–713. Springer, 2006.

[46] M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009.

[47] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. *Asian Conference on Computer Vision*, 2010.

[48] Michael Rubinstein, Ce Liu, and William T Freeman. Towards longer long-range motion trajectories. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2012.

[49] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision (IJCV)*, 80(1):72–91, 2008.

[50] G. Sapiro and M. Bertalmio. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007.

[51] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[52] T. Shiratori, Y. Matsushita, S. B. Kang, and X. Tang. Video completion by motion field transfer. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[53] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG)*, 25(3):835–846, 2006.

[54] O. Stanley and F. Ronald. Level set methods and dynamic implicit surfaces. *Springer-Verlag*, ISBN 0-387-95482-1, 2002.

[55] M. Subbarao. Depth recovery from blurred edges. *CVPR*, 1988.

[56] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. *European Conference on Computer Vision (ECCV)*, 2010.

[57] Rahul Swaminathan, Michael D Grossberg, and Shree K Nayar. Caustics of catadioptric cameras. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 2–9. IEEE, 2001.

[58] Yuichi Taguchi, Amit Agrawal, Ashok Veeraraghavan, Srikumar Ramalingam, and Ramesh Raskar. Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering. *ACM Transactions on Graphics-TOG*, 29(6):172, 2010.

[59] R. Tan. Visibility in bad weather from a single image. *CVPR*, 2008.

[60] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, 1988.

[61] Yuandong Tian and Srinivasa G Narasimhan. Seeing through water: Image restoration using model-based tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2303–2310. IEEE, 2009.

[62] N. Trefethen. *Spectral methods in MATLAB*, volume 10. Society for Industrial Mathematics, 2000.

[63] T. Tuytelaars. Dense interest points. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[64] Sara Vicente and Lourdes Agapito. Balloon shapes: reconstructing and deforming objects with volume from images. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 223–230. IEEE, 2013.

[65] Emmanuel Villermaux and Benjamin Bossa. Single-drop fragmentation determines size distribution of raindrops. *Nature Physics*, 5(9):697–702, 2009.

[66] Sebastian Volz, Andres Bruhn, Levi Valgaerts, and Henning Zimmer. Modeling temporal coherence for optical flow. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1116–1123. IEEE, 2011.

[67] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[68] Reg G Willson, M Maimone, A Johnson, and L Scherr. *An optical model for image artifacts produced by dust particles on lenses*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2005.

[69] A. Yamashita, I. Fukuchi, and T. Kaneko. Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information. *IROS*, 2009.

[70] A. Yamashita, Y. Tanaka, and T. Kaneko. Removal of adherent water-drops from images acquired with stereo camera. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[71] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi. Adherent raindrop detection and removal in video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[72] Shaodi You, Robby T Tan, Rei Kawakami, and Katsushi Ikeuchi. Robust and fast motion estimation for video completion. *IAPR Internation Conference on Machine Vision Applications*, 2013.

[73] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Makaigawa, and Katsushi Ikeuchi. Raindrop detection and removal from long range trajectories. *Asian Conference on Computer Vision (ACCV)*, 2014.

[74] Y. Zhang, J. Xiao, and M. Shah. Motion layer based object removal in videos. *IEEE Workshops on Application of Computer Vision*, 2005.

[75] Changyin Zhou and Stephen Lin. Removal of image artifacts due to sensor dust. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[76] V. Zorich and R. Cooke. Mathematical analysis. *Springer*, 2004.