



**HAL**  
open science

# Vision Algorithms for Rain and Traffic Lights in Driver Assistance Systems

Raoul de Charette

► **To cite this version:**

Raoul de Charette. Vision Algorithms for Rain and Traffic Lights in Driver Assistance Systems. Other. Ecole Nationale Supérieure des Mines de Paris, 2012. English. NNT : 2012ENMP0053 . pastel-00802707

**HAL Id: pastel-00802707**

**<https://pastel.hal.science/pastel-00802707v1>**

Submitted on 20 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale n°432 : Science des Métiers de l'Ingénieur

**Doctorat ParisTech**

**T H E S E**

pour obtenir le grade de docteur par

**Mines ParisTech**

**Spécialité « Informatique, temps réel, robotique et automatique »**

*présentée et soutenue publiquement par*

**Raoul de CHARETTE**

le 17 septembre 2012

**Algorithmes de vision pour la pluie et les feux tricolores  
pour les systèmes d'aide à la conduite**

**Vision Algorithms for Rain and Traffic Lights  
in Driver Assistance Systems**

Directeur de thèse : **Fawzi NASHASHIBI**

**Jury :**

**M. Alain DUFAUX**, Docteur, Ecole Polytechnique Fédérale (Suisse)

**M. Didier AUBERT**, Directeur de recherche, IFFSTAR

**M. Srinivas NARASIMHAN**, Associate professor, Carnegie Mellon University (USA)

**M. Davide MANETTI**, Docteur, Institut Fédéral de la Propriété Intellectuelle (Suisse)

**M. Abdelaziz BENSRAHAI**, Professeur, INSA Rouen

**M. Fawzi NASHASHIBI**, Directeur de recherche, INRIA Rocquencourt

**M. Benazouz BRADAI**, Docteur, VALEO

Rapporteur

Rapporteur

Examinateur

Examinateur

Examinateur

Directeur de Thèse

Invité

**MINES ParisTech**  
**Centre de Robotique**

60 bd Saint Michel, 75006 PARIS - France

**T  
H  
È  
S  
E**



# Contents

<b>Acknowledgment</b>	<b>vii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Rain</b>	<b>7</b>
1.1 Appearance of a raindrop . . . . .	7
1.1.1 ...in the air . . . . .	8
1.1.1a Shape . . . . .	8
1.1.1b Optics . . . . .	9
1.1.1c Photo-geometric mapping . . . . .	9
1.1.2 ...on a screen . . . . .	13
1.1.2a Shape . . . . .	13
1.1.2b Optics . . . . .	14
1.1.2c Photo-geometric mapping . . . . .	14
1.2 Appearance of rain . . . . .	15
1.2.1 Physics . . . . .	15
1.2.1a Drops Size Distribution (DSD) . . . . .	16
1.2.1b Dynamics . . . . .	17
1.2.1c About physics models inaccuracies and limitations . . . . .	19
1.2.2 Vision in rain . . . . .	20
1.2.2a Imaging rain . . . . .	20
1.3 Rain and vision-based ADAS . . . . .	23
1.3.1 Introduction . . . . .	23
1.3.2 Detecting focused raindrops on a windscreen . . . . .	23
1.3.2a Dedicated sensors . . . . .	24
1.3.2b Model based . . . . .	25
1.3.2c Gradient based . . . . .	25
1.3.2d Statistical approaches . . . . .	26
1.4 Conclusion . . . . .	28
<b>2 Detection of Unfocused Drops on a Windscreen with Blur Assessment</b>	<b>31</b>
2.1 Introduction and motivation . . . . .	31
2.2 Detection using photometry and lack of gradients . . . . .	33
2.3 Detection using blur assessment . . . . .	35
2.3.1 No-reference defocus blur detection - State of the Art . . . . .	36
2.3.1a Gradients analysis . . . . .	37
2.3.1b Frequency analysis . . . . .	38

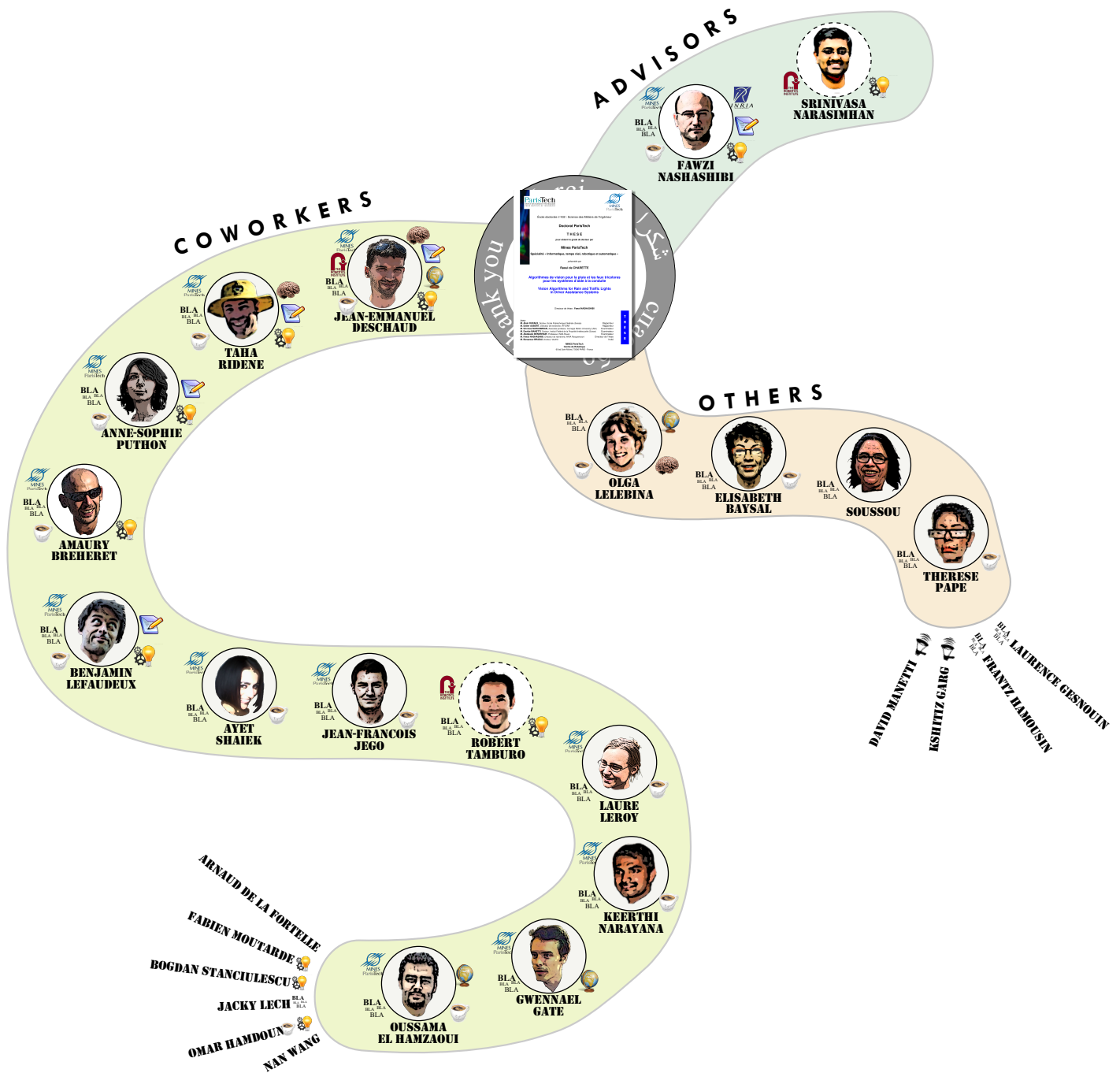
2.3.1c	Chromatic . . . . .	39
2.3.1d	Statistics of natural scenes . . . . .	40
2.3.2	Blur detection for unfocused drops detection . . . . .	41
2.3.2a	Haar Wavelet Transform and Edges characterization . . . . .	43
2.3.2b	Slope of the amplitude spectrum . . . . .	46
2.3.2c	Weighted FFT . . . . .	48
2.3.2d	Future experiments . . . . .	50
2.4	Conclusion . . . . .	51
<b>3</b>	<b>Illumination through Rain at Night</b>	<b>55</b>
3.1	Introduction . . . . .	55
3.2	Related work . . . . .	57
3.3	Precipitation at night . . . . .	58
3.3.1	Light scattering . . . . .	58
3.3.1a	Operating range required . . . . .	58
3.3.2	Psychophysical studies . . . . .	59
3.3.3	Observations . . . . .	60
3.4	Overall approach . . . . .	60
3.4.1	Mitigation strategies . . . . .	60
3.4.2	Our collocated imaging-lighting approach . . . . .	61
3.5	Feasibility study using simulations . . . . .	62
3.5.1	Overview . . . . .	62
3.5.2	Particles system simulator . . . . .	63
3.5.3	Operating range . . . . .	65
3.5.4	Performance analysis for a stationary system . . . . .	65
3.5.5	Performance analysis for a moving system . . . . .	67
3.5.6	Recipes for complete systems . . . . .	69
3.6	Proof-of-concept – Prototype . . . . .	71
3.6.1	Hardware . . . . .	71
3.6.2	Test bed . . . . .	72
3.6.3	Detection and Prediction Algorithms . . . . .	72
3.6.4	Performance . . . . .	73
3.6.4a	Test bed experiments . . . . .	73
3.6.4b	Simulating a moving prototype . . . . .	76
3.6.5	Using the prototype to assess the simulations . . . . .	76
3.7	Discussion toward a real system for rain . . . . .	77
3.7.1	Drops detection and tracking with vision algorithms . . . . .	77
3.7.2	Physics models worldwide . . . . .	79
3.7.3	Influence of air flow and wind . . . . .	80
3.7.4	Conservativeness . . . . .	80
3.8	Conclusion . . . . .	81
<b>4</b>	<b>Traffic Light Recognition</b>	<b>85</b>
4.1	Introduction . . . . .	85
4.1.1	Appearance and detectability . . . . .	86
4.1.2	Wireless communication . . . . .	86
4.2	Vision-based approaches . . . . .	88
4.2.1	Existing traffic lights recognition systems . . . . .	88

4.2.2	Our approach . . . . .	90
4.3	Spot light detection . . . . .	91
4.3.1	Related work . . . . .	91
4.3.1a	Chromatic segmentation . . . . .	92
4.3.1b	Gradients and transforms . . . . .	93
4.3.2	Our approach: Grayscale segmentation . . . . .	94
4.3.2a	High trends . . . . .	94
4.3.2b	Discrimination . . . . .	96
4.3.2c	Performance . . . . .	101
4.3.2d	Influence of the parameters . . . . .	102
4.4	Traffic Light Classification . . . . .	102
4.4.1	Related work . . . . .	102
4.4.1a	Statistical learning . . . . .	103
4.4.1b	Structural rules . . . . .	104
4.4.1c	Shape and template matching . . . . .	104
4.4.2	Our approach: Template matching . . . . .	105
4.4.2a	Template matching . . . . .	105
4.4.2b	Light color identification with low dynamic sensor . . . . .	108
4.4.2c	Performance . . . . .	113
4.5	Performance of our Traffic Light Recognition . . . . .	115
4.5.1	Temporal filtering . . . . .	116
4.5.1a	Geometrical tracking . . . . .	116
4.5.1b	Light sequence . . . . .	118
4.5.2	Performance . . . . .	119
4.5.2a	France . . . . .	121
4.5.2b	Switzerland, China . . . . .	123
4.5.3	Timing . . . . .	124
4.5.4	Quantitative comparison . . . . .	124
4.6	Conclusion . . . . .	127
	<b>Conclusion</b>	<b>129</b>
	<b>Publications</b>	<b>133</b>
	<b>Bibliography</b>	<b>135</b>






# Acknowledgment



Special thanks to Fawzi Nashashibi (thesis advisor), Arnaud de La Fortelle (head of the lab) and to everyone who contributed to the friendly atmosphere in the lab.






## Laboratories



-  Robotics institute, CMU
-  Imara team, INRIA
-  Robotics centre, Mines ParisTech

## Thesis

-  Scientific help
-  Manuscript review

## Miscellaneous

-  Endless discussions
-  Countless coffees
-  Travelled together

-  Vital moral support
-  Distant interaction





# Introduction

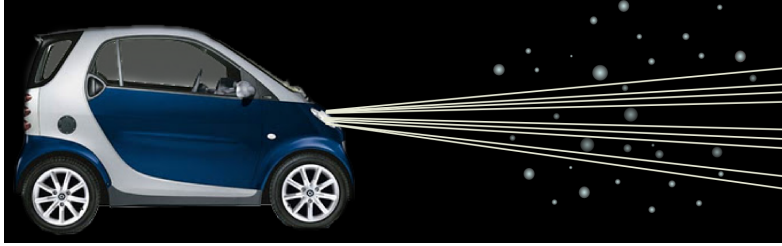
Computer vision algorithms are now used in most intelligent systems like gsm, surveillance camera, video games, and many others. In the automotive context, the cheap cost of the imaging sensors has pushed forward the development of new vision-based Advance Driver Assistance Systems (ADAS) in an effort to use a single vision sensor for all sensing tasks. Compared to their early beginning in the 70's, the vision-based ADAS are at use in series car so they are supposed to be helping the driver in all kind of scenarios, such as: dense urban environment, traffic jam, degraded conditions, etc. Not only do they have to work in these situations, but they have to be dependable since the drivers tend to overrely the assistance systems [Inagaki, 2008].

This thesis deals with vision algorithms for ADAS in the new encountered scenarios that are the degraded conditions and the dense urban environment (figure 1). For degraded conditions, our research focuses on rain and for dense urban environment we have been working on traffic light recognition. While these two research subjects may appear unrelated, both of them shall participate to the expansion of the working range of existing assistance systems.

Rain is a year-round phenomenon that has been investigated in major computer-vision studies [Garg and Nayar, 2007, Barnum et al., 2007]. Researchers have shown that rain changes greatly the appearance of a scene and that vision algorithms are prone to failures in such conditions. Surprisingly, the effect of rain on the driver and its ability to *see* the scene has been very little studied. To our knowledge even less work was conducted on the effect of rain on ADAS. Given that the latter are intended to support the driver, they should be able to handle degraded conditions such as rain. To that matter, understanding the effect of rain on the imaging sensor is a prerequisite to the development of new assistance systems for rainy weathers.

In **chapter 1**, we review rain facts that one requires for the development of new computer vision algorithms and vision-based ADAS. We provide the reader with physics, optics and meteorological knowledge on rain that we use in other rain-related chapters. To some extent we also compare the models and highlight some misuses in the existing computer vision algorithms. The little work on rain in ADAS is also reviewed in the first chapter including the methods dealing with raindrops detection on the windscreen [Kurihata et al., 2005, Halimeh and Roser, 2009, Cord and Aubert, 2011]. However, the whole literature intends to detect the drops while the windscreen is in *focus*, which we prove to be an ill-conditioned problem as the other vision algorithms require the focus to be set on the road.

We propose novel methods to detect *unfocused* raindrops on a windscreen in **chapter 2**. Our first attempts use either photometry and lack of gradients, or blind blur assessment to locate the unfocused drops. Though our experiments are not yet fully conclusive, they exhibit intermediate results. This is an unprecedented problem that would allow using of a single multi-purpose camera for all vision-based applications. In such case, the detection of raindrops can be used to lower the confidence of other vision algorithms in rainy situations, so as to avoid failures. For the driver,



(a) Illumination through rain at night



(b) Unfocused raindrop detection



(c) Traffic light recognition

Figure 1: Three vision-based ADAS applications of the research we present in this manuscript, for rain (a-b) and traffic lights (c).

the automatic activation of wipers is also a way to improve his visibility since [Bhise et al., 1981] have shown that the water on the windscreen is mostly responsible of the loss of visibility during *daytime*.

In **chapter 3** we show that for *nighttime*, the wipers are not sufficient to improve the driver's visibility. The effect of the headlights reflection in the falling drops is responsible of high spatio-temporal frequencies, that we prove to cause fatigue of the driver while lowering his ability to perceive the scene. Ergo, we propose an illumination solution to the problem of falling raindrops that consists in the selective illumination of a scene, similar in spirit to the anti-blinding systems. Our idea is applicable to other particulate weathers as well (rain, snow, hail), and uses a fast reactive control to illuminate the scene without shining the falling particles. To narrow down the challenge we have developed a comprehensive simulator (including physics, hardware, imaging, etc.). Vision algorithms are also investigated and our experiments show that accurate drops detection can be achieved at night with a high speed camera. As a result of our findings, an actual prototype was built in laboratory conditions that effectively removes drops appearance *in the scene* which improves the overall visibility.

Apart from the need to take into account degraded conditions such as rain, vision-based ADAS are now supposed to be working in dense urban environments. Dealing with these new scenarios brings a number of new challenges, among which traffic light detection stands as an important one. Not only traffic lights are used for traffic regulation, but they also acknowledge the driver of dangerous areas. Still, in 2005 in the USA 18.6% of the accidents pertained to intersections *with* traffic signals [Choi, 2010]. The reason is that the driver are often distracted by other light sources and advertisements that together reduce the visibility of the traffic lights.

In **chapter 4**, we propose a vision-based chain of algorithms to detect traffic lights and identify their state in real time from an in-vehicle camera. Traffic light detection has been attempted several times in the past [Lindner et al., 2004, Hwang et al., 2006, Kim et al., 2007] but is usually limited to simple environments and handle only the traffic lights imaged against the sky that are easier to detect. Indeed, their undetailed appearance together with their small size in the picture make the traffic lights hard to detect accurately from an on-board camera. The algorithm we propose is meant to work in dense urban environment where neon-signs, vehicle lights, and advertisements are often source of confusion [Kimura et al., 2007]. It relies on an intensity-based spot lights detection, a modular template matching classifier and a temporal filtering to ensure the reliability of the output. Hence, in contradiction to the existing works, our algorithm detects traffic lights in real-time even in highly dynamic environments. Furthermore, it has been designed to be modular so it can be used for different types of traffic lights. We evaluated its performance on live sequences from France, China and Switzerland.

As a summary, the main contributions of this thesis are:

- A review of the rain facts and the comparison of physics models (DSD, velocity, etc.) that has highlighted the misuse of several models in computer vision (chapter 1).
- A method to detect the appearance of unfocused raindrops using photometry and lack of gradients. And first experiments on the detection of unfocused raindrops on a windscreen using blind blur assessment (chapter 2).
- The investigation of a novel fast reactive illumination through rain at night, including an exhaustive simulator and the validation of the findings with a prototype (chapter 3).
- A modular real-time chain of algorithms to detect traffic lights using grayscale morphological spot light detection and template matching classification, that we evaluated on sequences from France, China, Switzerland (chapter 4).



# Chapitre

# Pluie

*Below is a French summary of the following chapter “Rain”.*

Bien qu'elle puisse paraître chaotique, la pluie est en fait conditionnée par de multiples facteurs atmosphériques rendant chaque pluie *unique*. Ainsi, un volume de pluie est composé de nombreuses gouttes (de 0.1mm à 5mm environ) qui chutent de plusieurs mètres par seconde. A l'image, celles-ci forment des trainées (ou cordes) de pluie, dont l'apparence dépend des propriétés physique, météorologique, et optique des gouttes de pluie. Des études majeures existent - telle que [Garg and Nayar, 2003] - mais aucune revue de la littérature n'a été publiée dans ce domaine, et nous entendons couvrir ce manque dans ce chapitre. Il ne s'agit pas de faire une liste exhaustive de tous les modèles mais plutôt d'évaluer les domaines de validité de chacun et d'en extraire les clés nécessaires aux algorithmes de détection ou synthèse de la pluie par vision. Pour cela nous passons en revue l'apparence des gouttes en l'air et sur une vitre (cf. section 1.1), puis l'apparence de la pluie (cf. section 1.2). Notre expertise montre qu'une partie des algorithmes de vision utilisent des modèles approximatifs - voire inappropriés - et que le simple usage de modèles adéquats permettrait l'amélioration des performances.

Dans la dernière partie (cf. section 1.3) nous présentons les méthodes de détection de la pluie dans les ADAS. Malgré qu'il existe plusieurs méthodes pour caméras embarquées, nous montrerons que l'influence de la profondeur de champs empêche l'usage de ces dernières dans les ADAS.



# Chapter 1

## Rain

Rain may appear as chaotic but is in fact driven by many atmospheric conditions making each rain *unique*. A volume of rain is composed of numerous particles called raindrops, which diameter mostly lies between  $0.1mm$  and  $5mm$ , falling at high relative speed<sup>1</sup>. In addition, the transparent nature of a raindrop make its interaction with the light complex and remarkable. Its optical properties have been used for instance to capture the world in a drop [Garg and Nayar, 2007] while properties like the relation of raindrop size to rain intensity is used to predict weather [Laws and Parsons, 1943]. Rain has been extensively studied in fields like meteorology, physics, optics or even chemistry leading to a wide variety of models. Because of the vast literature, it is not trivial to find accurate information or to pick the right model. In this chapter we aim at filling this gap through a review of the knowledge on rain. It is not our objective to improve or to list all the existing models but to sieve out the key properties of rain that might be needed to design computer vision applications to characterize, to simulate or to detect rain.

Most of the models we introduce in the following chapter sustain the ideas we describe in chapters 2 and 3, and help understanding the challenge of rain in vision.

### 1.1 Appearance of a raindrop

We now describe the appearance of a single drop (see figure 1.1) and provide the reader with a review of physics, optics and computer-vision related models. These models have been used in the literature to simulate raindrops [Halimeh and Roser, 2009], to capture the scene in a drop [Garg and Nayar, 2003], or to detect drops [Roser and Geiger, 2009]. While most of the models focus on raindrops solely they could as well be applied to water drops of any kind (i.e. generated from sprinklers). In order to cover the literature for the rest of this thesis, we deal with two cases separately depending on whether the drop is in the air or on a transparent screen (i.e. protecting glass, wind-shield, etc.).

---

<sup>1</sup>As an example, a raindrop of  $3mm$  diameter reaches a velocity of  $8m/sec$  which is about 2,600 times its own size.



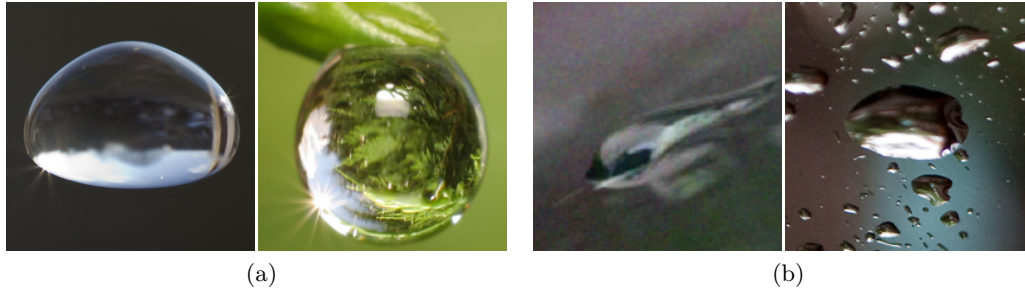


Figure 1.1: Appearance of a drop in the air (a) or on a transparent screen (b).

### 1.1.1 ...in the air

#### 1.1.1a Shape

Generally speaking the size of a raindrop lies in the range  $0.1\text{mm} - 5\text{mm}$  [Desaulniers-Soucy et al., 2001] though the mechanical limit of a drop without splitting is  $10.5\text{mm}$  [Pruppacher and Pitter, 1971, p.7]<sup>2</sup>.

As it falls the complex shape of a raindrop is affected by surface tension, hydrostatic pressure, aerodynamic pressure, internal circulation and electrical stress [Lim, 2006]. Small drops below  $1\text{mm}$  remain spherical but larger drops tend to flatten on the bottom and deviate from a perfect sphere because the forces become relatively bigger in comparison to the surface tension. When the external forces become too strong a drop splits in two or more smaller drops.

Many numerical models have been published to describe drops shape and are listed in the reviews of [Szakáll et al., 2010] and [Beard et al., 2010]. To express the radius  $r$  as a function of the angle  $\theta$  widely used is the cosine series distortion of a sphere also referred as *Chebyshev series*:

$$r(\theta) = a_0 \times \left( 1 + \sum_{n=0}^N c_n \times \cos(n \times \theta) \right) \quad (1.1)$$

$a_0$  being the radius of the undistorted sphere,  $c_n$  the distortion coefficients that are estimated using one of the existing model,  $\theta$  the polar angle (with  $\theta = 0$  the fall direction), and  $N$  the number of order. [Pruppacher and Pitter, 1971] claims that order over 9 can safely be neglected even if some use  $N = 11$  [Beard, 1976, Chuang and Beard, 1990] or  $N = 12$  [Lim, 2006].

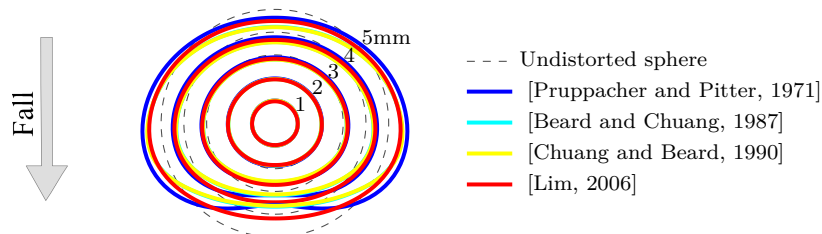


Figure 1.2: Comparing shape models for raindrops.

<sup>2</sup>Note that raindrops of this size have never been observed in real rain. The largest raindrop recorded was found at Sumatra, Indonesia and measured  $8.59\text{mm}$  [Fujiyoshi et al., 2008].

Figure 1.2 compares models for the sessile shape of a raindrop (aka the equilibrium shape). For computer vision related tasks [Garg and Nayar, 2003, p.10] first proposed to use the model from [Beard and Chuang, 1987] known as the “most accurate model” for drops below  $5mm$  diameter. Above this limit, dimple *may* appear on the bottom of the drop and precise simulation will benefit of more complex models. Yet as drops are usually few pixels wide when imaged the use of either models does not make much change and one shall even use the simple numerical model of [Green, 1975] described by Beard as “almost as accurate as the more elaborate theoretical models”.

Strictly speaking the equilibrium shape approximation remains valid only a fraction of time as the forces that apply on a falling drop produce oscillations that are modelled in [Beard, 1984] and in [Foote, 1973]. As we study here the stationary shape of the drop we neglect these oscillations in the present chapter but readers may refer to the above cited documents. Note however that shape alterations modify the optical properties and play an important role when a falling drop is imaged with relatively long exposure time. Hence, these oscillations - while not clearly visible - are noticeable to the naked eye and produce the slightly wavy streaks we glimpse in a rain scene<sup>3</sup>.

### 1.1.1b Optics

Due to its shape and transparent nature a drop acts as a tiny lens with a wide field of view (FOV) reflecting the world upside-down. The FOV of a spherical drop in the air was found to be  $\approx 165^\circ$  [Garg and Nayar, 2003] (close to a fish-eye) thus much larger than the part of the background occluded. This has great implications. First, the average drop radiance is very little dependent of its background. Second, each point on the drop reflects a point of the scene. Finding the coordinates and radiance of the latter can be achieved through *photometric-geometric mapping* which in turn requires to understand how light is scattered by a raindrop.

Light scattering by particles includes four main components that are refraction, diffraction, phase shift, and absorption thus only highly complex models can accurately describe their effect. However, for particles much larger than the light wavelength<sup>4</sup> and with a refractive index sufficiently different from its surrounding, the wave character of light may be neglected and a good approximation can be obtained by *geometric optics* [van de Hulst, 1957, p.201] (i.e. ignoring phase shift and absorption). The study of light scattering through a drop agrees with both conditions<sup>5</sup>. Hence, the geometric tracing of a stencil light through a drop results of Snell-Descartes laws while its photometric alteration is defined by Fresnel equations. Derivation of these equations provides accurate enough photometric-geometric mapping models that we describe below.

### 1.1.1c Photo-geometric mapping

Three major studies were published in computer vision [Macke and Groß klaus, 1998, Ross, 2000, Garg and Nayar, 2003]. Surprisingly, the two earlier models used a more accurate - yet more complex - model that is the equilibrium shape of a drop while Garg and Nayar assumed a simple spherical approximation. For simplicity we start with the latter.

<sup>3</sup>These wavy streaks are more visible when illuminated by a spot light source.

<sup>4</sup>Recall that the visible wavelength is  $380nm < \lambda < 740nm$  while drops are in the order of millimetres.

<sup>5</sup>For these conditions to be fulfilled we consider only the visible domain ( $380nm < \lambda < 740nm$ ) and the drops that are in the air (refractive index  $\mu \approx 1$ ) or on a screen which index is significantly different from those of the surrounding medium and the drop. This holds for water drops ( $\mu = 1.33$ ) on glass ( $\mu = 1.5$ ), acrylic ( $\mu = 1.49$ ), etc.

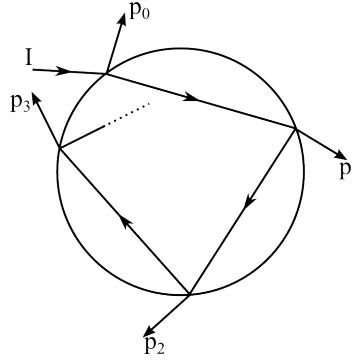


Figure 1.3: Light scattering through a spherical drop: forward ray tracing. Tracing of a single ray through a drop where  $p_0, \dots, p_3$  denotes the rays leaving the drop (the subscript stands for the number of refractions). Adapted from [Ross, 2000].

**Spherical approximation** Here we assume the shape of the drop to be spherical with a diameter  $d$  ( $mm$ ). Such assumption remains strictly valid for drops below  $1mm$  and is somewhat acceptable for drops smaller than  $2mm$  (cf. figure 1.2) that constitute most of the drops in a rain event (cf. section 1.2.1a).

Figure 1.3 shows the path of a single light ray that hits a spherical drop where  $p_0, p_1, \dots, p_N$  denotes the leaving rays that together form the radiance of the drop. Fresnel equations state that the intensity of  $p$  decreases rapidly with  $N$ , the number of refractions, due to absorption and light transfer, so an accurate approximation may ignore light rays after few interfaces. Ross wrote a sophisticated ray tracing process to measure the amount of light carried by each ray through all possible angles of incidence [Ross, 2000, p.8]. He reported that  $p_1$  alone carries 88.37% of the incident light and that the intensity of rays after two interfaces ( $p_3, \dots, p_N$ ) is below 1% and may be safely neglected<sup>6</sup>.

Garg and Nayar drew the same conclusion three years later. However, they used a reverse ray tracing model which is more convenient for computer vision applications and further simplifies geometrical mapping [Garg and Nayar, 2003]. Figure 1.4a illustrates this model. Consider the ray  $\hat{x}$  leaving the drop at point A with the normal  $\hat{n}$  at the surface of the drop. The radiance  $L(\hat{n})$  of point A can be computed as the sum of the radiance  $L_r$  of refracted ray  $\hat{r}$ ,  $L_s$  of reflected ray  $\hat{s}$  and  $L_t$  of internal reflection ray  $\hat{t}$ . Thus:  $L(\hat{n}) = L_r(\hat{n}) + L_s(\hat{n}) + L_t(\hat{n})$ .

Derivation of  $L(\hat{n})$  was performed in [Garg and Nayar, 2003] using geometrical mapping of the incident rays  $\hat{r}$ ,  $\hat{s}$  and  $\hat{t}$  to the normal  $\hat{n}$ , the environmental radiance  $L_e$  and the radiance transfer functions  $R$ ,  $S$  and  $T$  - refraction, specular reflection and internal reflection, respectively - and results in this equation:

$$L(\hat{n}) = R \times L_e(\hat{r}) + S \times L_e(\hat{s}) + T \times L_e(\hat{t}) \quad (1.2)$$

Using Fresnel equations Garg and Nayar derivate the transfer radiance functions  $R$ ,  $S$  and  $T$ , and

<sup>6</sup>Note that [Ross, 2000] computed energy transfer for two different polarisations (parallel or perpendicular to the scattering plane) but we refer here to the sum of both polarisations and hence assume an ambient lighting. Exact numbers are  $p_0 = 6.64\%$ ,  $p_1 = 88.37\%$ ,  $p_2 = 4.08\%$  and  $\sum_{i=3}^{\infty} p_i = 0.93\%$  (error margin of  $\pm 0.01\%$ ).

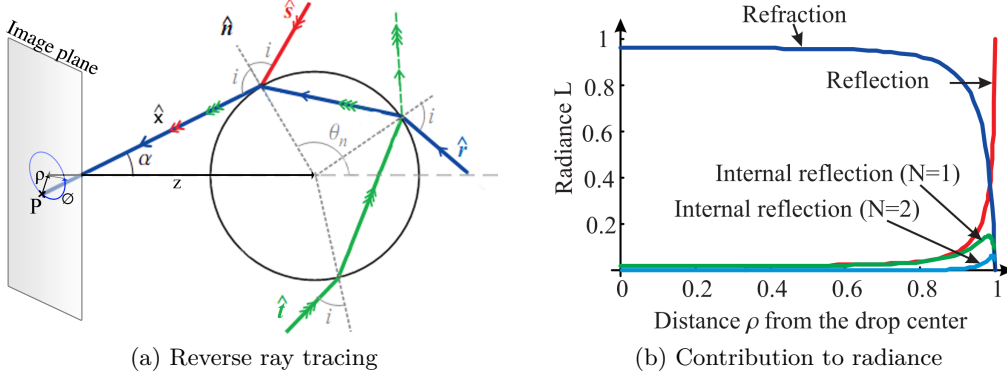


Figure 1.4: Light scattering through a spherical drop: reverse ray tracing. (a) Contribution to the intensity of a pixel  $P$ . The ray  $\hat{x}$  leaves the drop and is composed by the contribution of refraction  $\hat{r}$ , specular  $\hat{s}$  and inner reflection  $\hat{t}$ . (b) Contribution to the drop radiance  $L$  of refraction, reflection and after  $N$  internal reflections  $N \in \{1, 2\}$ . Note that internal reflections play very little role in the total radiance. (a) and (b) adapted from [Garg and Nayar, 2003].

further rewrite equation 1.2, into:

$$L(\hat{n}) = \left( (1 - k(i, \mu_{water})) \times L_e(\hat{r}) + k(i, \mu_{water}) \times L_e(\hat{s}) \right) + \sum_{N=1}^2 k(i, \mu_{water})^N (1 - k(i, \mu_{water}))^2 \times L_e(\hat{t}) \quad (1.3)$$

$i = \pi - \theta_n + \alpha$  being the incident angle,  $\mu_{water}$  the refractive index of water and  $k$  being the Fresnel's reflectivity coefficient which depends on light polarization. See [Garg, 2007] for complete derivation. Note that for the transfer radiance function  $T$  (internal reflection) rays with more than two inner reflections were neglected (i.e.  $N = \{1, 2\}$ ). Using equation 1.3 Garg and Nayar estimated the contribution to the drop radiance, cf. figure 1.4b. It appears that - and according to what was shown by Ross earlier - the drop radiance results mainly of refraction. If we admit  $L(\hat{n}) \approx R \times L_e(\hat{r})$  then geometrical mapping relies only on the ray-tracing of the refracted ray. Now, lets see how to retrieve the point of the scene refracted in the drop. This is used for image registration for instance.

According to [Garg and Nayar, 2003] and to figure 1.4a the coordinates of a pixel  $P$  can be defined as  $(\rho, \varnothing)$ , where  $\rho$  is the distance of the point to the center of the raindrop image and  $\varnothing$  is the azimuth angle:

$$\rho = \frac{f}{z} 0.5d \times \sin \theta_n, \quad \varnothing = \varnothing_n \quad (1.4)$$

$f$  being the focal length of the camera and  $z$  the distance between the center of the drop and the camera. Note that such relation is valid only if the drop and the camera axis are aligned horizontally.

Using Snells-Descarte laws,  $\theta_n$  and  $\varnothing_n$  are retrieved from the incident ray refracted  $\hat{r}$ <sup>7</sup>:

$$\theta_r = 2(\pi - \theta_n) + \alpha - 2 \sin^{-1} \left( \frac{\sin(\theta_n - \alpha)}{\mu_{water}} \right) \quad (1.5)$$

<sup>7</sup>We found found some off guards mistakes in [Garg and Nayar, 2003] (eq. 8 in their article should be eq. 1.4 and eq. 12, 15, 18 should be  $i = \pi - \theta_n + \alpha$ ). Readers should rather refer to [Garg, 2007, Garg and Nayar, 2007].

$$\varnothing_r = \pi + \varnothing_n \quad (1.6)$$

Assuming that the relative position of the raindrop to the camera is known, the mapping of the environment through the drop may be performed using equations 1.5 and 1.6 while iterating through  $\theta_n \in [0; \pi]$ .

The model of Garg and Nayar is well shaped for computer vision as it allows tracing back a refracted ray through a drop. The authors verified their model using a drop hanging from a pipette or an acrylic ball with a checker board and found that the RMSE (Root Mean Square Error) is 1% for the geometric mapping and 1.17% for the photometric mapping [Garg, 2007]. However real drops are not perfect sphere and such model shall not be accurate enough to simulate large close-up drops because they significantly deviate from a sphere.

**Series of cosine distortion of a sphere** Unlike the spherical approximation, the series cosine distortion of a sphere representation is widely used in the atmospheric sciences and accurate for drops of any size (cf. section 1.1.1a).

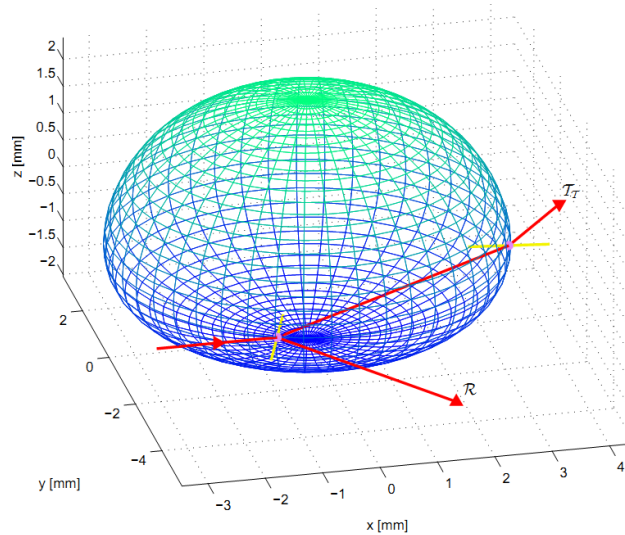


Figure 1.5: Light scattering through a non-spherical drop approximated through a 3D model obtained from the revolution of the series of cosine. Adapted from [Ross, 2000]

Here we present a ray tracing approach which has been used in [Macke and Groß klaus, 1998, Ross, 2000] and more recently for simulation in [Rousseau et al., 2006]. Ray tracing is often used for better approximation of the light path but is often presented as costly and slow. Conversely to ray tracing with the spherical approximation it requires an iterative numerical process as it can not be solved analytically. Instead, Ross proposed a detailed 3 stages ray-tracing process<sup>8</sup> which corresponds to a ray: 1) entering the drop, 2) inside the drop, and 3) leaving the drop. Usually, this process starts from the incident ray down to the refracted ray outside the drop but to remain consistent with the model described previously we present it starting from a point on the image plane.

We now describe how to find the corresponding point of a scene that is imaged by a pin-hole camera after being refracted through a drop of radius  $r$  (in  $mm$ ). Like Ross we assume the

<sup>8</sup>Note that Ross distributes the code for the ray-tracing process in [Ross, 2000, Appendix A].

coordinate system to be right-handed Cartesian and with the origin at the center of mass of the drop, as in figure 1.5. The spherical coordinates of a point on the drop are  $(r, \theta, \phi)$ ,  $r$  and  $\theta$  are defined as in equation 1.1 and  $\phi$  is the azimuth angle.

We denote  $P$  an image point of the drop and  $Q$  a virtual point which initial position in our coordinates system is  $Q(L + f, P_y, P_z)$ ,  $f$  being the focal.  $Q$  belongs to the ray  $\widehat{r}_{tt}$ . Finding the point where the ray  $\widehat{r}_{tt}$  hits the drop can be solved through the iterative process of moving  $Q$  along the ray until the following condition is violated:  $r(\theta) < |Q|$ . Finding the incident ray  $\widehat{r}_t$  that refracted into  $\widehat{r}_{tt}$  requires the normal  $\hat{n}$  at drop surface which is approximated through partial derivatives:  $\hat{n} = \left( \frac{\partial r}{\partial \theta} \times \frac{\partial r}{\partial \phi} \right)$ . Now the angle of incidence  $\alpha_{i1}$  is computed using Snell-Descartes:

$$\alpha_{i1} = \sin^{-1} \left[ \frac{\mu_{air} \times \sin(\alpha_{t2})}{\mu_{water}} \right] \quad \text{with} \quad \alpha_{t2} = \cos^{-1}(\hat{n} \cdot r_{tt}) \quad (1.7)$$

Using  $\alpha_{i1}$  we get  $\widehat{r}_t$ , the ray in the drop. The incident ray  $\widehat{r}$  is found with the same process by first moving  $Q$  along  $\widehat{r}_t$  until the condition:  $r(\theta) > |Q|$  is violated. We denote  $\hat{m}$  the normal at the drop surface. Finally  $\alpha_r$  the angle of incidence of ray  $\widehat{r}$  is:

$$\alpha_r = \sin^{-1} \left[ \frac{\mu_{water} \times \sin(\alpha_{t1})}{\mu_{air}} \right] \quad \text{with} \quad \alpha_{t1} = \cos^{-1}(\hat{m} \cdot r_t) \quad (1.8)$$

As for the spherical approximation, Ross exhibits that even distorted drops scatter forward more than 90% of the incident light. Thus the radiance of the drop is mostly due to refraction and the coordinates of the point of the scene imaged - after refraction through the drop - may be retrieved using the above described approach. However, it is interesting to note that the forward scattering is non-monotonic with the size of the drops if we refer to the experiments of [Macke and Groß klaus, 1998].

### 1.1.2 ...on a screen

For applications such as surveillance or driver assistance systems, the camera is mounted behind a transparent screen that is a protecting glass or a windscreen and the raindrops are therefore visible either in the air or on the screen. In such scenario their shape, optics and appearance is dramatically different from those of a drop in the air, see figure 1.1. Studies for the detection of raindrops on a screen were conducted even prior the famous work of Garg and Nayar, but to the best of our knowledge only two studies have described the appearance of the raindrops when on a screen.

#### 1.1.2a Shape

Exact shape of a drop on a solid is modelled using a large body of physics laws but computer-vision applications use instead simplified models that are the spherical section approximation or the Bézier representation as in figure 1.6. Both of these models were used to generate virtual raindrops as in [Halimeh and Roser, 2009, Roser and Geiger, 2009] or [Roser et al., 2010].

**Sphere section approximation** In the first representation, figure 1.6a, a section of a sphere or *sphere cap* represents a drop lying on a solid surface. Its shape is defined by the radius of the virtual sphere  $C_{sphere}$  and the maximum height of the cap  $h$ . The relation between a drop in the air and the virtual sphere radius may be computed given that the volume  $v$  shall remain constant

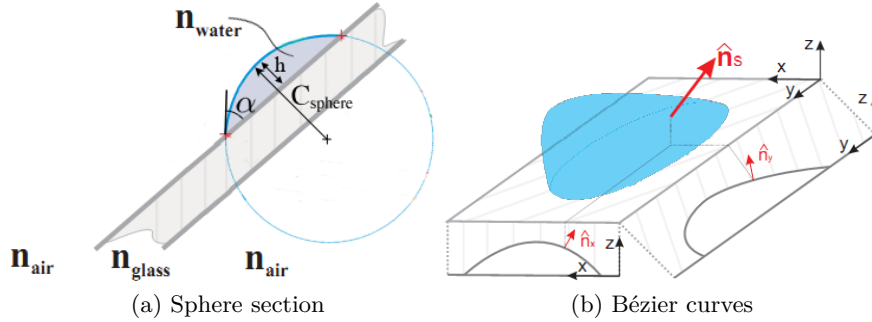


Figure 1.6: Models for the representation of a sphere on a solid screen. (a) Sphere section approximation. Adapted from [Roser and Geiger, 2009]. (b) Bézier curves. Adapted from [Roser et al., 2010].

in the air or on the screen. Thus:  $C_{sphere} = \sqrt[3]{\frac{3}{\pi} \times \frac{v}{(2-3\sin\alpha + \sin^3\alpha)}}$ . Where  $\alpha$  is the contact angle given by Young-Laplace equation.

**Bézier approximation** Alternately, Roser *et al.* proposed using two orthogonal 2D Bézier curves to model the effect of gravity on the shape of a drop on a tilted screen<sup>9</sup>, cf. figure 1.6b. In this representation the Bézier curves have four points each to control the shape of the drop but for better physical interpretation the authors rather define the shape using angles  $\alpha_1$  and  $\alpha_2$  that correspond to the same contact-angles given by the Young-Laplace equation, and two coefficients  $w_1$  and  $w_2$  that model the effect of gravity. Two orthogonal profiles are used to model the 3D shape. In [Roser et al., 2010] the parameters were learned from 20 real images of drops on a tilted screen at different angles through a Bézier curve fitting process. A distinction between surface plane and raindrops points is performed by a RANSAC<sup>10</sup> line fitting in the Canny image.

### 1.1.2b Optics

Surprisingly the optics of a drop on a screen has not been investigated that much in the related papers and most of them refer to the study of Garg and Nayar though it is obviously not suitable for a drop on a screen.

### 1.1.2c Photo-geometric mapping

Due to the number of refractions, the tracing is more complex than those for a drop in the air and to the best of our knowledge a single ray-tracing approach has been proposed by Halimeh and Roser assuming the sphere section approximation.

**Sphere section approximation** Though it is the sole model to describe the appearance of a drop on a screen [Halimeh and Roser, 2009] uses a rather simple ray-tracing approach similar to those of [Ross, 2000] and [Macke and Groß klaus, 1998], cf. figure 1.7a. Similarly, the tracing

<sup>9</sup>In the specific case where the screen surface is perfectly horizontal the Young-Laplace equations say that the two contact angles of the profile of a drop are identical and the sphere section approximation is perfectly accurate.

<sup>10</sup>RANSAC stands for RANdom SAMple Consensus and is a widely used strategy to fit data that contains both inliers and outliers [Fischler and Bolles, 1981].

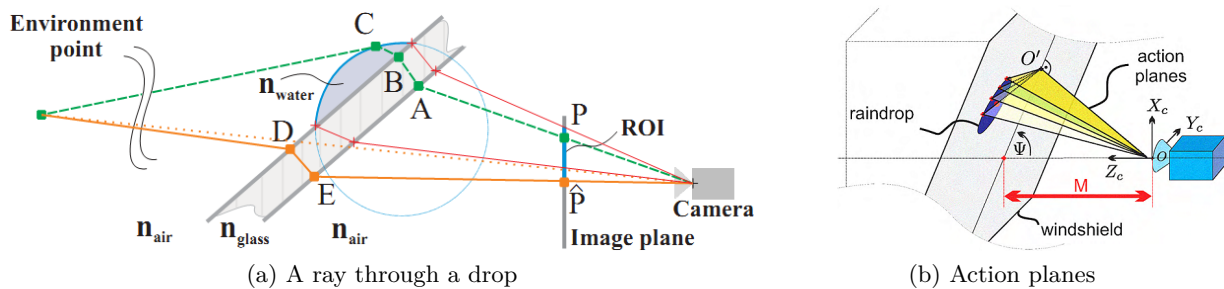


Figure 1.7: Model of Halimeh and Roser for the photo-geometric mapping of a ray through a drop (a) using the action planes (b) according to the Snell's law of refraction. Source: [Halimeh and Roser, 2009]

through a drop is achieved by an iterative process and the use of the law of geometric optics. The noticeable difference is that Halimeh and Roser take benefits of the so-called *action planes* depicted in figure 1.7b. The latter are defined according to Snell's law of refraction which says that the ray passing through a drop to the optical center never leaves this plane ever since coming from the drop surface. The use of action planes allows tracing multiple rays through a drop to the optical center of the camera using simple 2D geometry.

Details on Halimeh and Roser's ray tracing are not given here as it is somewhat identical to the process described in section 1.1.1c but additional details may be found in [Halimeh and Roser, 2009]. This model is used to generate virtual drops [Halimeh and Roser, 2009] and to detect the drops on a windscreen through image registration [Roser and Geiger, 2009].

## 1.2 Appearance of rain

Hitherto, we have described the appearance of a stationary drop but when observed in real rain the multitude of drops and their dynamics make the appearance of the scene varying dramatically. At long distance, the drops appear as aggregated particles that reduce contrast and visibility distance making the scenes look greyish. At short distance, and because they act as tiny lenses the drops are responsible of high frequency patterns that one may want to model for further detection and removal. We study in this section the visual effect of rain, and present the various models that were developed for dealing with rain. We first begin with an introduction on physics.

### 1.2.1 Physics

Particulate weathers have been broadly classified in two groups: *steady* weathers (i.e. fog, mist and haze) and *dynamic* weathers (i.e. rain, hail, graupels) [Narasimhan, 2004, Garg and Nayar, 2004]. Rain differs from the steady weathers in that it is formed by numerous large falling particles producing *precipitation*. To characterize the intensity of a precipitation we measure the amount of water that fell on the ground within an hour. As a rough idea, light rain is about  $5mm.h^{-1}$ , shower about  $25mm.h^{-1}$ , and storm over  $100mm.h^{-1}$ .

Aloft, rain is the result of moisture particles merging into droplets (from  $1\mu m$  to  $10\mu m$ ) which in turn coalesce with other droplets and form larger particles that eventually fall [Mason, 1975].



By convention, droplets are called *raindrops* when they reached at least  $0.1mm$  diameter<sup>11</sup>. Measuring the size of raindrops has first been performed in the 19th century by exposing to the rain a thin sheet of absorbent paper or a pan filled with flour<sup>12</sup> [Bentley, 1904] but is now performed with disdrometers. The latter measure drops size using either: the mechanical momentum of an impacting drop (i.e. JWD Joss-Waldovegel Disdrometer [Joss and Waldvogel, 1967]), the velocity to size relation (i.e. Doppler Radar/Lidar [Atlas et al., 1973] or VIDIAZ Spectropluviometer [Donnadieu, 1980]), or video capture (i.e. Video Disdrometer [Schönhuber et al., 1995]).

Numerous experimental studies using disdrometers measured rain physics due to its implication in the meteorological field. But question can be asked whether all the existing literature is relevant to deal with rain in computer vision ? In the following we review interesting physics properties, compare the existing models, and exhibit that some of the models currently used are not consistent for studying rain in computer vision.

### 1.2.1a Drops Size Distribution (DSD)

[Laws and Parsons, 1943] first noticed that the number of occurrences of the drops decreases as their size increases while being scaled by the weather intensity. In other terms, there is a higher probability to find large drops in heavy rain. The MP law [Marshall and Palmer, 1948] models such relation and defines the concentration of drops of diameter  $d$  ( $mm$ ) as an exponential function of the fallrate  $R$  ( $mm.h^{-1}$ ):

$$N_d = N_0 \times e^{-\Lambda \times d} \quad (1.9)$$

where  $N_d$  is the number of drops per unit volume having diameter  $d$ ,  $N_0 = 8000m^{-3}.mm^{-1}$  is the initial concentration, and  $\Lambda = 4.1 \times R^{-0.21}mm^{-1}$ .  $N_0$  and  $\Lambda$  are set empirically.

The MP law is widely used but deviates from the observations of small drops ( $d \leq 1mm$ ) and fails to model heavy rain ( $R > 25mm.h^{-1}$ ). There are two reasons for such failure: (a) earlier disdrometers incorrectly estimated the number of small drops [Rinehart, 1983, Nešpor et al., 2000], (b) there is a depletion of small droplets at ground level and thus MP would hold better above the clouds than below [Best, 1950, Ulbrich, 1983, Bradley et al., 2000]. In the light of these observations [Best, 1950] proposed the use of a Gamma distribution:

$$N_d = N_0 \times d^\mu \times e^{-\Lambda \times d} \quad (1.10)$$

where  $\mu$  refers to the type of rain. At ground level,  $\mu > 0$  fits the experiments for widespread or thunderstorm rains [Ulbrich, 1983] and models correctly the lack of small drops that results of coalescence and evaporation. Gamma distribution is convenient because it models a wide variety of rain events and because it reduces to the exponential form if  $\mu = 0$ . Exhaustive reviews of DSDs may be found in [Best, 1950, Rosenfeld and Ulbrich, 2003].

---

<sup>11</sup>Some argue that smallest drops produce drizzle. The exact cutoff between drizzle and rain has been arbitrarily set to  $0.2mm$ ,  $0.5mm$  or even  $1mm$ . Indeed smallest drops exhibit quite different physical and optical properties. In vision too, small drops are often ignored because they appear as *fog-like* veil.

<sup>12</sup>It is worth reading the beautiful story of Wilson A. Bentley - aka "the snowman" - who was the first American to measure the size of raindrops by exposing a thin pan filled with flour to rain and measuring the size of the roundish granule of dough [Heidorn, 2000, Blanchard, 2004]. Notice that earlier measurements - much less accurate - were conducted using a thin sheet of paper. Because Bentley's snowflakes pictures were very famous many people consider Bentley one of the first to discover the shape of flakes while ignoring his contribution to the study of rain.

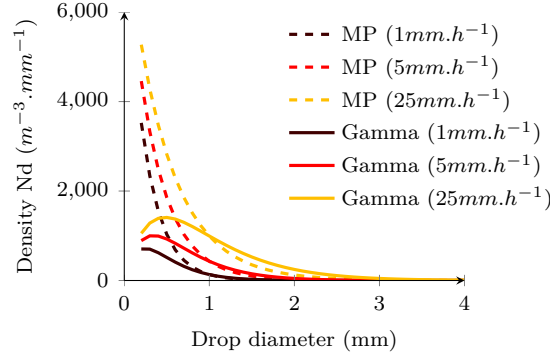


Figure 1.8: Comparing Marshall-Palmer and Gamma Distribution for  $1mm.h^{-1}$ ,  $5mm.h^{-1}$  and  $25mm.h^{-1}$  rain. The Gamma distribution parameters ( $\mu = 1.0$ ) are average of the thunderstorm rains reported in [Ulbrich, 1983]. Notice that MP exhibits many more small drops.

The choice of the gamma distribution against MP (cf. figure 1.8) should be based on two criteria: the need to take into account small drops, and the need to model non-convective or non-continental rain. MP can't reflect accurately the DSD in tropical regions and is valid only for convective continental rain below  $25mm.h^{-1}$  and for  $d > 1mm$ . Though MP law is mostly used against gamma distribution in the literature it is because most of the existing literature is in the field of meteorology and study the rain at cloud level. At ground level, Marshall-Palmer themselves stated that the exponential form is overestimating the measurements of small drops [Marshall and Palmer, 1948].

From our Particles System Simulator (see section 3.5.2) we have compared the impact of either distribution in the image space. Assuming a light rain ( $5mm.h^{-1}$ ) being imaged by a 1 mega pixels camera<sup>13</sup>, using the MP law instead of a gamma distribution ( $\mu = 1.0$ ) results in an exceedance of +41% of the number of drops imaged (+58% for  $25mm.h^{-1}$ ).

### 1.2.1b Dynamics

When they hit the ground the raindrops have reached their terminal velocity (up to  $9m/sec$ ). Small drops velocity obey to the Stoke law but bigger ones require more complex model. Three models stand out: [Gunn and Kinzer, 1949] for its fame, [Foote and Du Toit, 1969] for its accuracy, and [Atlas et al., 1973] for its ability to take into account the altitude. A fourth model [Van Boxel, 1997] will be presented for non-constant velocity.

**Terminal velocity** The semi-empirical model of Gunn and Kinzer expresses the terminal velocity ( $v_t$ ) of the raindrops in the air as a function of the gravity ( $g$ ), drop diameter ( $d$ ), pressure of the surrounding medium ( $P_m$ ) and the drop medium ( $P_s$ ), and the drag coefficient ( $c$ ):

$$v_t(d) = \sqrt{\frac{4}{3} \times g \times \frac{P_s - P_m}{P_m \times c}} \times \sqrt{d} \quad (1.11)$$

Using physics constants this model reduces to  $v_t(d) = 1414.21 \times \sqrt{d}$ . It has been used in computer vision by Garg in [Garg and Nayar, 2003, Garg and Nayar, 2004, Garg, 2007] as part of a segmentation process but it deviates dramatically from real experiments. To quantify its performance we

<sup>13</sup>We ignore all drops which size is less than one pixel in the image space.

have compared the velocity estimated with this model, with the velocity of laboratory experiments. The ground truth we refer to is also from [Gunn and Kinzer, 1949] as – even today – it is commonly used as a benchmark. The error of each model is evaluated as the Root Mean Square error (RMS) as the distance to the ground truth. For the above described model the RMS is **76.9cm.sec<sup>-1</sup>**.

Conversely, the model of Foote and Du Toit is known for its accuracy and its simplicity, cf. equation 1.12. It is an empirical model in the form of a  $N^{th}$  degree polynomial ( $N = \{3, 5, 9, 13\}$ ) which coefficients were obtained through least square curve fitting technique.

$$v(d) = \sum_{n=0}^N a_n \times d^n \quad (1.12)$$

It has been used for vision-related tasks by Barnum in [Barnum et al., 2007, Barnum et al., 2010a, Barnum, 2011] and in [Brewer and Liu, 2008]<sup>14</sup>. It is convenient because it allows adjusting the precision through  $N$  but is explicitly designed to fit terminal velocities *aloft*. RMS from the measurements of Gunn and Kinzer is **4.0cm.sec<sup>-1</sup>** with  $N = 3$ , and **2.6cm.sec<sup>-1</sup>** with  $N = 5$ .

Alternatively, Atlas et al. proposed a slightly less accurate model which allows computing the terminal velocity at various altitudes taking into account the pressure at observation level ( $P_m$ ) and ground level ( $P_{m_0}$ ):

$$v(d) = [965 - 1030 \times \exp(-6d)] \times \left( \frac{P_m}{P_{m_0}} \right)^{-0.4} \quad (1.13)$$

RMS from Gunn and Kinzer experiments is **10.4cm** which is less accurate than the model of [Foote and Du Toit, 1969] but unlike previous models it does not deviate even for large raindrops over  $5mm$ .

Figure 1.9a compares the different models for terminal velocity. The most important conclusion is the inadequate use of the model of Gunn and Kinzer which deviates dramatically. We believe that the use of the model of Atlas et al. is the most convenient because it takes into account altitude and does not deviate even for large drops.

**Acceleration** [Van Boxel, 1997] presents a numerical model for the velocity of raindrops that did not reach terminal velocity. This is needed for instance to simulate the fall of drops with no initial velocity. The model from Van Boxel computes the acceleration  $a(t, d)$  of a drop with diameter  $d$ , as the sum of gravity ( $F_g$ ) and drag force ( $F_d$ ):

$$a(t, d) = F_g(d) + F_d(t, d) \quad (1.14)$$

with

$$F_g(d) = \frac{g\rho_w\pi d^3}{6}, \quad F_d(t, d) = 3\pi d\mu_a v(t)C_t C_d \quad (1.15)$$

$$C_t = 1 + 0.16Re^{2/3}, \quad C_d = 1 + 0.013(We + 2.28)^{2.12}$$

<sup>14</sup>Note that the coefficients  $a_n$  reported in [Barnum et al., 2007, Brewer and Liu, 2008, Barnum et al., 2010a, Barnum, 2011] are rounded values of those written in the original [Foote and Du Toit, 1969]. Even in their working range ( $d \leq 3mm$ ) these rounded values lead to a RMS of  $51.31cm.sec^{-1}$  and a maximum error of  $134cm.sec^{-1}$ .

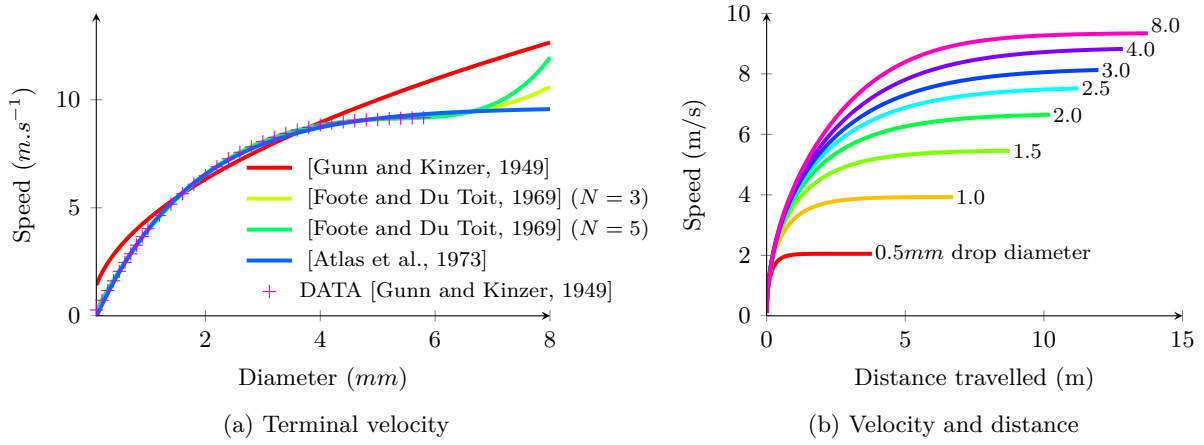


Figure 1.9: Velocity of a raindrop. (a) compares 4 models widely used with the measurements from [Gunn and Kinzer, 1949]. Note that the model from Gunn and Kinzer and the model from Foote and Du Toit deviate significantly for large drops. (b) shows velocity of drops of varying diameter versus vertical distance travelled when released with no initial velocity.

$g$  being the gravitational force ( $m \cdot sec^{-2}$ ),  $\rho_w$  the density of air ( $kg/m^3$ ),  $\mu_a$  the viscosity of air ( $Pas$ ),  $Re$  and  $We$  respectively the Reynold and Weber number of the drop (*dimensionless*).

Figure 1.9b plots the speed of raindrops when released with no initial velocity. Important fact is the distance of free fall required to reach the terminal velocity. To be accurate an artificial rain-generator (e.g. using sprinklers) should be positioned over  $10m$  for the drops to reach their terminal velocities. This parameter is often neglected though, obviously we have taken it into account for sprinkler-rain generation in our experiments for the illumination through rain (cf. chapter 3).

### 1.2.1c About physics models inaccuracies and limitations

The physics models we have described are known to be inaccurate in some scenarios. As an example, velocity greater than the terminal velocity is often encounter but limited to small drops and refered as *super-terminal velocity* [Montero-Martínez et al., 2009]. More important is to say that the DSD may differ according to the location. Tropical rain for example exhibits larger drops than other type of rain. In section 3.7.2 we will exhibit how different locations on earth affect a system dealing with rain.

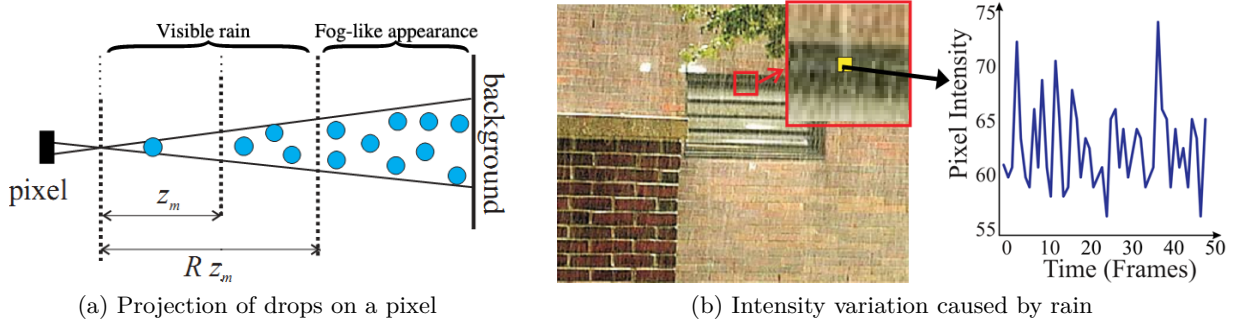


Figure 1.10: Imaging rain and variation of intensity. (a) Projection of raindrops on a pixel. The effect of rain on the intensity of a pixel can be categorized given the distance of a drop to the camera. Rain appear as fog-like when the variation due to a single pixel is below the noise level of the imaging sensor. (b) Effect of rain on the intensity of a single pixel (camera and scene stationary). Right is display the intensity of the chosen pixel over time, where peaks correspond to close or large drops. Adapted from: [Garg and Nayar, 2005].

## 1.2.2 Vision in rain

In the previous section we have seen that rain is composed of numerous particles falling at high speed. The photometric and geometric properties of raindrops were also discussed in section 1.1 but the visual effect caused by falling particles is quite different. In contradiction to fog, the effect of rain on the optical depth is rather limited<sup>15</sup> but the close-by falling drops produce high frequency patterns.

The most comprehensive work on vision in rain is the thesis of Kshitiz Garg [Garg, 2007] and the articles [Garg and Nayar, 2005, Garg and Nayar, 2007]. As we do not intend to be exhaustive the reader should note that the following is – for the most part – a summary of Garg and Nayar’s work.

### 1.2.2a Imaging rain

When imaging a rainy scene of enough depth, all pixels are affected by rain but the effect caused on the image is uneven according to the distance of the drops to the optical center. Drops close to the imaging system produce bright streaks while those far away produce a greyish veil as for fog. [Garg, 2007] proposed a cut-off between *visible rain* and *fog-like rain* based on the projection size of the drops, see figure 1.10a. According to Garg and Nayar, *fog-like rain* occurs when the distance  $z$  of a drop to the optical center is such that the effect of the latter on a pixel radiance is below the noise level of the camera (that is,  $z \geq R z_m$ ). At such distance, the variation of intensity of a pixel results of the aggregation of numerous raindrops. In heavy rain, this explain why rain scenes appear as darker and low-contrasted. Often, only *visible rain* is taken into account in computer vision as the resulting artefacts are more likely to cause algorithm failures.

<sup>15</sup>From the experiments of [Ulbrich and Atlas, 1985], for the visibility distance (i.e. contrast lower to 5%) to fall below 400m the fall rate has to be approximately  $\approx 200mm.hr^{-1}$  or higher. Such conditions are seldom observed. Even in heavy rain of 50mm/hr the visibility is  $\approx 1,000m$ .

**Variation due to a falling raindrop** Consider a drop (with diameter  $d$  and velocity  $v$ ) falling in the *rain visible region* (i.e.  $z < Rz_m$ ) the change of intensity on a pixel due to this drop is different whether the drop projects on more than a pixel (i.e.  $0 < z < Z_m$ ) or not ( $Z_m \leq z < Rz_m$ ).

According to Garg and Nayar, when  $z < Z_m$  the intensity variation is not dependent of the distance  $z$ . Actually, this is true only if we admit that the drop intensity is identical on each point of the drop which seems an acceptable hypothesis in the image space since the drops are generally only few pixels wide. Hence, the change of intensity  $\Delta I$  due to a single drop is expressed as a function of the brightnesses of the raindrop  $L_r$  and the background  $L_b$ , that is<sup>16</sup>:

$$\Delta I = I_r - I_b = \frac{\tau}{T} \times (L_r - L_b) \quad (1.16)$$

$\tau$  being the time the drop remains in the FOV of a pixel, and  $T$  the exposure time. General boundaries for  $\tau$  are given in [Garg and Nayar, 2005]:  $0 < \tau < d/v$ . There are important facts from this model: (a) the variation depends of the exposure time, (b) as we know from section 1.1.1b that the drop's radiance is always brighter than the radiance of its occluded background, thus  $L_r > L_b$ . Hence, the variation due to a drop shall *always* remain positive<sup>17</sup>.

If  $z \geq Z_m$ , the variation of intensity must take into account the drop projection size. In this case, equation 1.16 is turned to:

$$\Delta I = I_r - I_b = \frac{f \times d}{z} \times \frac{\tau}{T} \times (L_r - L_b) \quad (1.17)$$

$f$  the camera focal. This follows the intuition that the variation of intensity is smaller for drops farther from the camera. Meanwhile, an other important conclusion from equations 1.16 and 1.17 is that the length of the streaks in the image space increases with the exposure time and is inversely related to the intensity variation. From these simple observations we show that varying the parameters of the camera enhance or reduce the appearance of rain, which has been investigated in [Garg and Nayar, 2005]. As an example, imaging the scene with a high speed camera makes the raindrops more visible, which we use in chapter 3 to precisely capture the position of raindrops for the illumination through rain.

In [Garg and Nayar, 2004] the authors posited that  $\Delta I$  is both constant for all the pixels that belong to a single streak, and linearly related to the background intensity. Such assumption is in fact not valid because they omit that a pixel may image more than one drop at the same time. In this case the variation results of a volume of rain that we *briefly* described next.

**Variation due to a volume of rain** The derivation of the variation due to multiple layers of rain is complex and written in [Garg and Nayar, 2005]. A key property is that the effect caused by each of the rain layers is additive assuming that the background and camera are stationary so the variation of intensity due to rain will still remain positive. Based on this simple property, temporal frame subtraction is often used in computer vision to detect rain in stationary scenes.

Figure 1.10b depicts the pixel intensity over time with falling rain. As the intensity is varying over time, the sharp variations correspond to close/large drops momentarily occluding the background. Likewise, the low variations are related to smaller/farther drops. Remarkably, one can note that the true background intensity can be guessed from the absolute minima in pixel intensity – assuming the noise is negligible –.

<sup>16</sup>For simplicity, we dropped the camera coefficient  $k_0$  due to linear radiometric response [Garg and Nayar, 2005].

<sup>17</sup>Since light diffraction is negligible the change in R, G and B channels due to a drop are roughly similar conversely to the opposite statement in [Zhang et al., 2006, Liu et al., 2009].

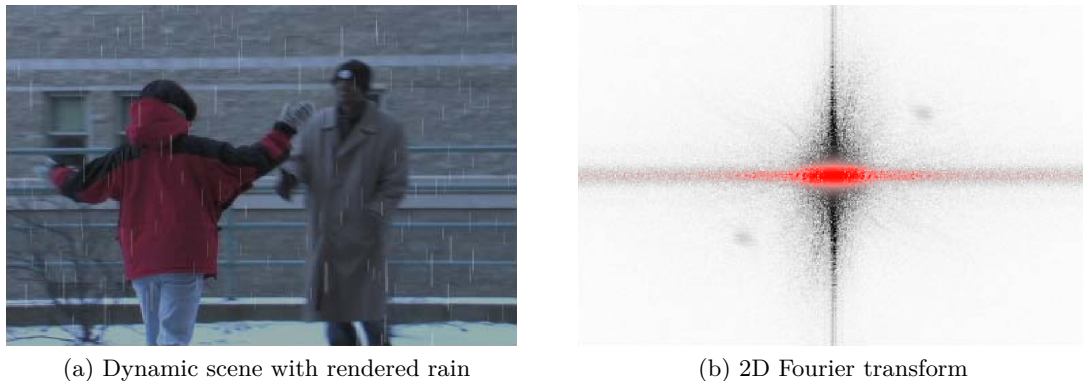


Figure 1.11: Rain component in the frequency domain. (a) Shows a sample dynamic scene with a static camera and artificially rendered rain. The multitude rain streaks produce high frequencies clearly visible in the 2D Fourier transform (b). For readability the frequency rain component is highlighted in red. Though this example is made simple by the use of rendered rain, the same type of frequency component is visible in scenes with real rain. Source: [Barnum et al., 2010a].

**Variation in the frequency domain** Instead of quantifying the effect on each pixel intensity Barnum and Narasimhan use the repeatable pattern of falling raindrops to model the streaks in the frequency domain [Barnum et al., 2010a]. Figure 1.11 illustrates their finding on a sample dynamic scene with rendered rain. As locally the streaks are roughly falling in the same direction the effect of the *rain layer* is clearly visible in the two dimensional Fourier transform, figure 1.11b. As may be seen the vertical streaks produce an horizontal well separable component in the frequency domain that we highlighted in red. Though, this sample uses rendered rain their model was proved valid with real rain.

To be accurate the rain model in the frequency space should be computed for each possible drops diameter, depth and orientation but Barnum and Narasimhan have shown that using the mean of all streaks is sufficient to approximate the rain model. Likewise, the model of a single streak is approximated with an oriented Gaussian which is also convenient in the frequency-space.

While the intensity models from Garg and Nayar assume the scene to be static, the frequency component of rain from Barnum and Narasimhan is only little dependent of the scene dynamic. However, the camera motion is problematic for both of them (though [Barnum et al., 2010a] simplify camera motion through a plane homography). In the presence of fast motion handling the rain is definitely challenging. This is the case for all automotive applications that we describe next.

## 1.3 Rain and vision-based ADAS

### 1.3.1 Introduction

The effects of rain on driving are multiple. Concerning the infrastructure: the roadway becomes slippery and the risk of skids is high which requires to increase the distance to the other vehicles. Concerning the visibility: the streaks, the splashes, the wet reflecting road and the water on the windscreen all together degrade the visibility. Some of these effects on visibility have been studied separately like wet weather visibility of pavement markings [Schnell et al., 2004] or the effect of the water film on the windscreen and wipers [Ivey et al., 1975, Bhise et al., 1981]<sup>18</sup>. Both of these studies have shown that for the driver, the loss of visibility in daytime rain is rather due to the irregular film of water on the windscreen than to the falling particles in the air<sup>19</sup>. At night however, our own experiments have shown that streaks become the brightest elements in the scene as they reflect all lights toward the driver eye, hence our proposal of illumination through rain – see chapter 3. Surprisingly, very little work was conducted on Advance Driver Assistance Systems (ADAS) in rain though the help of such system would be significant.

**Failures in rain.** The implicit posit of all ADAS is that the sensors accurately reflect the environment whereas in rain their output is not reliable. Hence, detecting rain is not only a way to assess the presence of degraded conditions but also a way to evaluate the confidence in sensors and thus prevent failures of vision algorithms. As for the driver, the drops on the windshield are the most likely to alter the performance of vision-based ADAS since the camera is usually mounted behind the windshield. Detecting rain can prevent failures of vision algorithms. This is especially important as the drivers tend to over-rely the ADAS so that any incorrect output of the latter is dangerous [Inagaki, 2011]. The only paper we found on the quantification of failures due to rain is [Nishigaki et al., 2000]. This Japanese team exhibits that the drops on the windshield produce artificial clusters that alter the computation of the visibility distance with a camera.

**How to assess rain ?** To detect rain some projects reviewed in [Ditze et al., 2010] classify the road surface based on its reflectivity coefficient or the time-of-flight of sensing beam from LiDAR sensors. However, the road may be wet while not raining. Except at night with the headlights *on* the precipitation is not visible, so the drops can only be detected when on the windshield. If the drops are detected, the visibility can be restored by activating the wipers which is also a way to notify to the driver that the driving conditions are degraded<sup>20</sup>. In the ideal case where the drops are perfectly detected one can also remove raindrops appearance through image processing but again this have drawn very few interest.

### 1.3.2 Detecting focused raindrops on a windscreen

Solutions to detect raindrops on a windscreen imply either the use of dedicated sensors (usually infra-red sensors or multiple cameras) or an on-board camera. To that day, all the previous

---

<sup>18</sup> [Bhise et al., 1981] proposed a model to compute the distance of visibility, that is:  $D = c_0(r \times t)^{-c_1} e^{c_2 \times L_b}$ .  $c_0, c_1, c_2$  three positives constants,  $(r \times t)$  the characterization of the amount of water accumulated on the screen (with  $r$  the rain intensity,  $t$  the time between wiper movement), and  $L_b$  the background luminance (i.e. sky luminance).

<sup>19</sup>In daytime, the impact of the falling particles (streaks) on the driver visibility is further weakened by motion blur due to the vehicle speed.

<sup>20</sup>A field study on the behavior of drivers of all ages have shown that given two identical weather conditions, the headway time margin was increased when wipers were on [Wetzel et al., 2004]. Hence a conclusion is that in addition to enhance the visibility of the driver, they are also a way to assess of the degraded condition



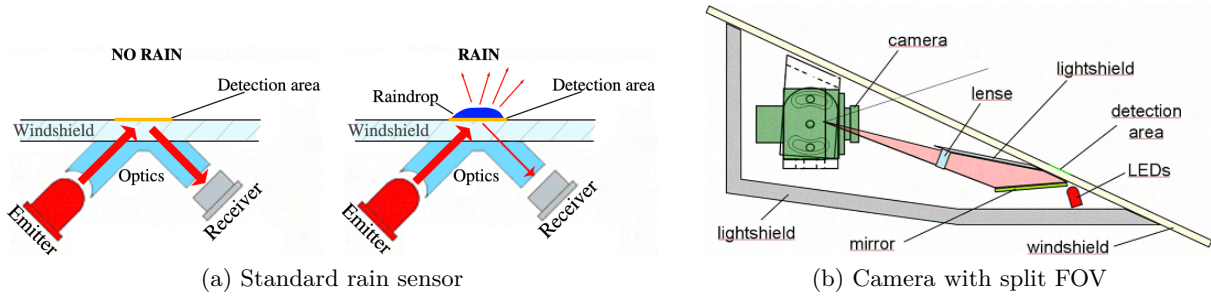


Figure 1.12: Two dedicated rain sensors. (a) shows the principle of the rain sensor used in all series cars. A photodiode measures the light reflected by the windscreen. Adapted from: [Görmer et al., 2009]. (b) a multi functions camera where the FOV is split to allow detecting the drops on the windscreen and other standard vision tasks. Source: [Görmer et al., 2009].

attempts with a single imaging sensor assume the windscreen to be in the Depth Of Field (DOF) (i.e. the drops lying on the screen are imaged *focused*). We will discuss in chapter 2 that this is a somewhat ill-conditioned problem as the vision-based ADAS usually need the focus to be set on the road.

### 1.3.2a Dedicated sensors

The principle of the most common rain sensors used in series cars is depicted in figure 1.12a. All use a light receiver with an infrared emitter that projects light on the windscreen<sup>21</sup> (e.g. [Bos et al., 2001]). If a raindrop hits the screen on the detection area, the water surface will refract part of the light and less amount of light is reflected towards the photodiode. The same principle using a camera instead of a photodiode was applied in [Park et al., 2006]. The light emitter makes the drops clearly visible and thus easy to segment through simple edge detection. In [Görmer et al., 2009] an original hardware solution is proposed that splits in two the camera Field Of View (FOV), see figure 1.12b. While the upper part of the picture images the scene, the lower part focuses on the windscreen thanks to mirrors and lenses cleverly fixed. Again LED emitter is used so that the image processing is straightforward.

Other alternatives that require multiple point of views were experimented by a Japanese team. The raindrops are detected from three cameras [Yamashita et al., 2003], two cameras [Yamashita et al., 2005, Tanaka et al., 2005], or a single rotating camera [Yamashita et al., 2008]. Having applied an image registration, the strategy they used to detect the sticking drops is either to detect the regions with the minimum variance in all views [Yamashita et al., 2003], or to compute the disparity and select regions which disparity correspond to the predictable windscreen disparity [Yamashita et al., 2005, Tanaka et al., 2005]. The performance they obtain is good both qualitatively and quantitatively.

Now, we detail the raindrop detection methods using a single vision sensor. In the last six years, a dozen of researches have attempted to tackle this challenge and we have grouped the latter in three categories: model based, gradient based, and statistical approaches.

<sup>21</sup>The second generation of rain sensor uses an additional mirror to increase the detection area.

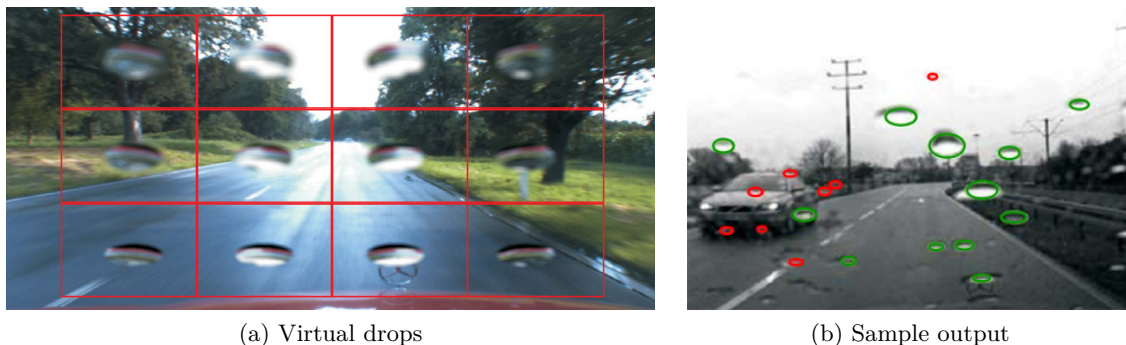


Figure 1.13: Model based methods. (a) Twelve virtual raindrops generated from the RIGSEC mapping for different positions on the windscreen. (b) Sample output of the raindrops detector using the intensity cross correlation with the virtual drops (green: true positive, red: false positive). Source: [Roser and Geiger, 2009]

### 1.3.2b Model based

Halimeh and Roser have developed a simplified photogeometrical mapping of a drop on a screen which is coined *Raindrop Intelligent Geometric Scanner and Environment Constructor* (RIGSEC) [Halimeh and Roser, 2009]. The RIGSEC allows to generate virtual raindrops as they would appear on the windscreen using geometric optics and Fresnel refraction model (see section 1.1.2). Real raindrops are then located in the image space by minimizing the distance to the model through iterative intensity cross-correlation. The baseline is improved in [Roser and Geiger, 2009] where five planes homographies approximate the scene geometry. The limitation of such a method is that only a portion of the windscreen can be covered by the drops, since the whole picture serves to generate the appearance of the virtual drop. Additionally, the position and size of the actual drops must be fixed *a priori* in order to generate accurate raindrops. In section 1.1.2c we have shown that the drops image the scene with a wide angle, which means that their appearance only change for radically different positions on the windscreen. Likewise, [Roser and Geiger, 2009] claim that generating twelve virtual drops (positioned on a  $4 \times 3$  grid) is sufficient to match the raindrops at all possible positions on the windscreen. We show sample of virtual drops and output of the matching process for two *unrelated* pictures in figure 1.13. As may be seen, the approach have shown some great success but the homographies restrict the working range to the scenes where the flat-road assumption remains valid (no cars or pedestrian before for instance).

### 1.3.2c Gradient based

Methods based on the gradients do not explicitly use the models of the raindrops but they attempt to detect strong gradients that result in fact of the upside down image of the scene inside the drop. Chronologically, the approach of [Leleve et al., 2007, Challita, 2008] first proposed to analyse the gradients with an operator called *declivity* described in [Miché and Debie, 1995]. Drops are regions that respect horizontal and vertical declivity criteria. They report good results in [Challita, 2008] but the database used is limited to 50 images and the output is binary (*rain* or *no-rain*).

[Cord and Aubert, 2011] also looks for strong local gradients temporally invariant to assess the position of the drops. The authors detect the strongest gradients in  $21 \times 21$  blocks and rule out most of them with geometrical or temporal constraints. This French team is capable of detecting

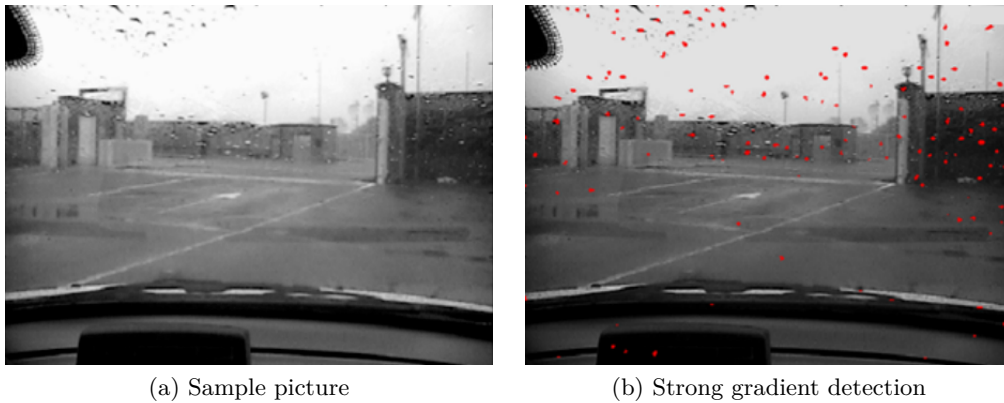


Figure 1.14: Gradients based method. (a) Shows a sample image and (b) the detection of drops from highest local gradients detection. Note: sample outputs do not differentiate true positives and false positives. Source: [Cord and Aubert, 2011].

most of the drops, as showed in figure 1.14. The proposed method also output several false positives but “without inferring an incorrect rain status”.

### 1.3.2d Statistical approaches

Statistical generalization from a subset of examples was also attempted to detect drops in focus. In [Kurihata et al., 2005] *eigendrops* are defined as a subset of the eigenspace built from a Principal Component Analysis (PCA) and serve as features for the classifier. A classifier is trained separately for drops over the sky or not. This approach was proved inefficient when evaluated on the whole picture but acceptable if restricted to the sky area (especially with the improvement in [Kurihata et al., 2006]). Other attempts include SURF [Roser and Geiger, 2009] or a strange combination of SVM and Cellular Neural Network [Schwarzlmüller et al., 2010]. In the latter, an interesting feature they used is the distance-vectors between center and contours of the drops, shown in figure 1.15.

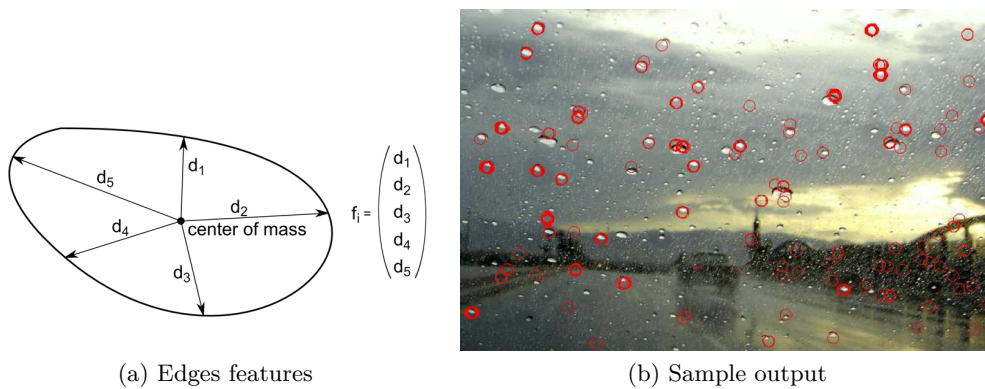


Figure 1.15: Statistical methods. (a) Exhibits the edge feature proposed to train an SVM classifier. (b) is a sample output. Note: sample outputs do not differentiate true positives and false positives. Source: [Schwarzlmüller et al., 2010].

Noteworthy, the set-up used in all these experiments is such that a large part of the windscreen is visible. Hence, when it rains a large number of raindrops are visible (up to a hundred in [Cord and Aubert, 2011]) and there is no need to detect all of them. Conversely, this allows a large number of false detection without inferring a change in the rain status output.

## 1.4 Conclusion

Hopefully, the present chapter provides a comprehensive knowledge one requires to deal with rain in computer vision. Though it already covers several fields of research, additional reviews on rain detection, rain simulation, and rain removal would have been required so as to be exhaustive. These topics were not treated because they are clearly borderline with the scope of this thesis.

Important conclusions may be drawn from the current study starting with the inadequate use of physics and meteorological models in some computer vision applications. The reason of the misuse is maybe that finding and understanding the existing physics/meteorological models required a long review process. While reviewing the physics-related knowledge of rain (section 1.2.1) we have shown for example: that Marshall-Palmer DSD overestimates small drops and is inadequately used for heavy rain (section 1.2.1a), that the Gunn-Kinzer model for drop velocity is not adequate (section 1.2.1b), or that Foote and Du Toit is often not used with the *ad-hoc* parameters. Still, an open question that would require deeper analysis, is how much the accuracy of the existing models is affected by the misuse of these models.

In the last section we have also introduced the problem of ADAS in rain and emphasized the lack of existing systems to help the driver in such scenarios. A brief analysis of the accidents in rain also exhibit that drivers would benefit of accurate rain detection to prevent failures in vision algorithms or at best, to remove the effect of rain. Part of the remaining chapters will focus on these questions.

## Chapitre

# Détection de Gouttes Defocalisées sur le Parebrise par Estimation du Flou

*Below is a French summary of the following chapter “Detection of Unfocused Drops on a Windscreen with Blur Assessment”.*

De jour, la détection de la pluie par camera embarquée dans un véhicule ne peut s’effectuer que par la détection des gouttes sur le pare-brise. En traitement d’image, peu de méthodes existent et toutes exigent que le pare-brise soit dans le champs de profondeur de la camera afin de détecter les forts contours *dans* la goutte qui résultent de la vue inversée du monde dans celle-ci (cf. optique de la goutte, section 1.1.2b). Si le plan focal de la caméra est placé à l’infini les gouttes sont alors floues et toutes les méthodes actuelles impuissantes. En effet, contrairement aux gouttes focalisées la détection de gouttes floues ne peut s’appuyer sur la localisation de forts gradients a l’instar de [Cord and Aubert, 2011] ou [Kurihata et al., 2005]. Cependant détecter les gouttes floues permettrait de n’utiliser qu’une seule caméra multi-taches puisque les autres applications ADAS (détection de piétons, de marquages, etc.) requièrent un focus à l’infini.

Notre première méthode (cf. section 2.2) détecte *l’apparition* des gouttes par ses propriétés photométriques, le manque de gradients et la stabilité temporelle [Nashashibi et al., 2010]. Bien que fonctionnelle, cette méthode montre cependant des limites importantes dans le cas de scènes hautement dynamiques.

Une seconde méthode innovante est basée sur l’estimation de la carte de floue de l’image source (cf. section 2.3) pour y détecter *la présence* de gouttes defocalisées. Le challenge consiste alors à estimer les faibles variations de flou causées par les gouttes sur le parebrise. Dans la carte de flou, les positions de ces dernières étant alors obtenues par l’extraction d’extrema locaux temporellement stables. Des résultats préliminaires montrent que la WFFT (Transformée de Fourier pondérée) ainsi que la pente du spectre d’amplitude permettent d’estimer le flou local suffisamment précisément pour extraire la position des gouttes dans l’image.



## Chapter 2

# Detection of Unfocused Drops on a Windscreen with Blur Assessment



(a) Focused raindrops



(b) Unfocused raindrops

Figure 2.1: Raindrops on a windscreen have dramatically different appearance whether they are imaged focused (a) or unfocused (b). Currently, the entire literature detects only drops in focus which prevent from using a single multipurpose camera. Source of (a) [Schwarzlmüller et al., 2010], (b) is from a database recorded by LIVIC laboratory for the ICADAC project.

### 2.1 Introduction and motivation

The problem of raindrops detection on a windscreen has been tackled several times in the past and led to a number of researches we have described in section 1.3.2. However, none of the existing approach is capable of detecting unfocused raindrops. To image the raindrops in focus the camera is generally moved afar from the windscreen so that the raindrops lies in the Depth Of Field (DoF) whereas in practice the camera is set-up closer for other object-recognition tasks. To allow the use of a single vision sensor we investigate in this chapter the detection of the raindrops when not in focus (hereafter referred as *unfocused raindrops*). Such task is very challenging and to that day we did not find any fully working solution. Because the appearance of raindrops in-focus and unfocused is very different (see figure 2.1) the existing methods are unsuitable to that use. After a brief investigation of classical image processing approaches we investigate the idea of blur detection to detect local variation and identify the drops.



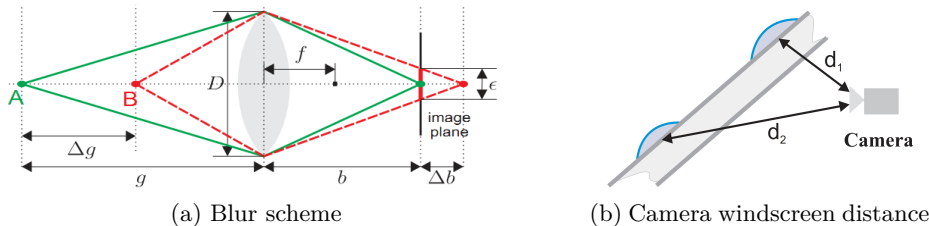


Figure 2.2: Optics of defocus raindrops. (a) Illustrates the general optics phenomenon of defocus. Here point B is not in the Depth Of Field and is projected on the image plane on a circle of confusion with diameter  $\mathcal{E}$ . The bigger  $\mathcal{E}$ , the blurrier the image of B. (b) Distance from the camera to the drops when they are lying on the windscreen. Source: [Roser and Geiger, 2009].

Let us emphasize that this chapter is an introduction to the problem of detecting unfocused raindrops as the work was conducted at the end of the thesis and the results are only preliminary.

The research is part of an European project coined ICADAC for *Improved CAmera based Detection under Adverse Conditions*. Currently, a dedicated rain sensor is used in series car to detect rain and automatically activate the wipers. Concerning on-board camera, all regular vision tasks (object detection, lane detection, etc.) require the focus to be set to infinity so that the road is focused. Because in series cars the camera is mounted between the interior rear mirror and the windscreen, the distance between the latter and the optical center of the camera is only few centimetres (below 5cm). Except if using macro imaging sensor the windscreen (and thus the raindrops) will then be out of the Depth of Field, *in extenso* blurred. The exact blur extent being function of the camera-windscreen distance and the imaging sensor parameters.

Apart from saving the cost of additional sensors, using the on-board camera to detect raindrops is a way to assess the image quality. Indeed, the blurred drops may infer failures in other vision-based algorithms. This could be avoided if we detect the position of the drops, that is if we assess which pixels belongs to a drop and are *not reliable*. Such application is not feasible with a dedicated rain sensor as it is not looking at the same portion of the windscreen. Additionally, we will discuss later on that restoration of the image is possible in some cases, which gives us a glimpse of what is achievable in a near future.

**Unfocused raindrops appearance** We have described in section 1.1.2 the geometrical mapping of light rays in a drop. If the camera is sufficiently far from the windscreen and the drop is in the Depth Of Field (DOF) they are imaged sharp and the up-side down reflection of the scene is clearly visible in the drop. Conversely, if the drop is out of the DOF each point of the drop project on the image plane with a large *circle of confusion* resulting in out-of-focus blur. From figure 2.2, the closer the camera is to the windscreen, the larger the circle of confusion and the blurrier it is imaged. Note that some cars may have extremely tilted screen so that the distance to the camera varies and thus the raindrops are imaged sharper on the bottom. However, experiments with our set-up and different cars (Citroen C3, Renault Clio, Audi) have shown that all raindrops exhibit no visible gradients and their appearance is such that it is not possible to distinguish the scene reflection in the drop, as seen in figure 2.1b. Since the existing methods rely on gradients analysis [Park et al., 2006, Cord and Aubert, 2011] or the drop refraction model [Halimeh and Roser, 2009] they are not capable of handling such unfocused drops. In [Roser and Geiger, 2009], the authors have extended the model of Halimeh and Roser to allow detecting slightly defocused

drops but their method being based on the *a priori* that the scene reflection is visible in the drop, it *will* fail at detecting any severely defocused raindrops.

To tackle the problem of unfocused raindrops we have investigated two different approaches. The first approach attempts to detect the apparition of raindrops using photometry and lack of gradients. It exhibits preliminary performance though we found it to be a dead-end. As a second attempt, we have investigated the ability to detect drops using local blur assessment. Though blur may appear for other reasons (e.g. motion blur) this serves as a strong cue to reduce the search space and identify the position of the drops. To this aim we will review in section 2.3.1 the blind blur assessment techniques.

## 2.2 Detection using photometry and lack of gradients

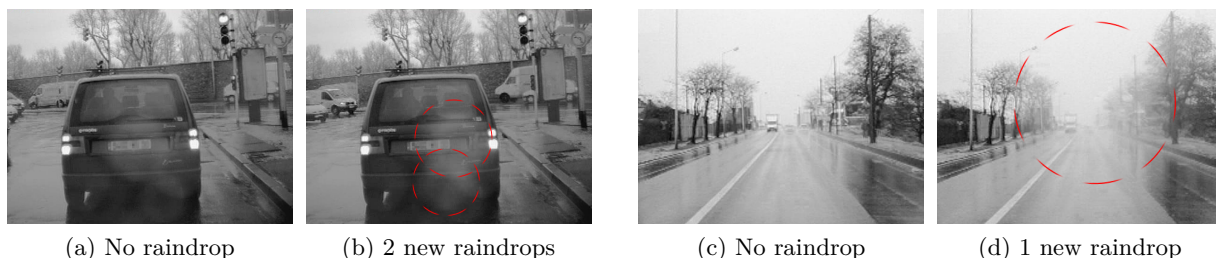


Figure 2.3: Sample images of severely defocused raindrops. Notice the bright appearance of the drops (related to their photometry) and the lack of gradients that we use to detect the latter.

As a first attempt, we describe a method using the photometry properties of a raindrop on a windscreen, to detect *not the presence* of a raindrop but its *apparition*. This method was tested on sequences that experienced extremely defocused drops seen in close-up, as in figure 2.3.

**Acknowledgement.** This method was first initiated by Alexandre LIA (former intern).

**Idea** The general idea is to first detect the apparition of potential raindrops areas by assuming that the car is static and the scene stationary, and then to rule out the regions falsely validated due to our strong assumption. There are three major observations that we use to detect a raindrop apparition:

- (A) From optics of a raindrop (see section 1.1.1b and 1.1.2b), a raindrop has a large enough Field Of View so that it exhibits a brighter intensity than its background.
- (B) The shape of a drop on a screen (see section 1.1.2a) is known to be if not perfectly circular, roughly circular.
- (C) Because the drop is imaged defocused it shall not exhibit strong gradients on its borders.

**Algorithm** The implementation of the algorithm is rather straightforward and follows the observations listed above. Still, note that it requires a final stage due to our strong assumption (static camera and stationary scene).

Following observation (A), we build a binary map  $M$  where pixels are classified into *rain* and *non-rain*. Suppose we are studying frame  $n$ , the binary map segments the pixels which intensity

became higher from  $n - 1$  to frame  $n$ , while having similar intensity to frame  $n + 1$ . That is:

$$M(x, y) = \begin{cases} 1 & \text{if } (I_n - I_{n-1}) \geq p_1 \text{ and } (I_n - I_{n+1}) \leq p_2 \\ 0 & \text{else} \end{cases} \quad (2.1)$$

Obviously, non rain pixels are also segmented with this simple map so we rule out most of them using raindrop geometric property, observation (B). From the rain pixels map  $M$ , we label all blobs with an 8-connected component extractor and remove those that do not respect a circularity constraint. *In extenso*, a blob  $B$  is validated if:  $|\frac{B^2_{Area}}{4\pi \times B_{Perimeter}}| \leq p_3$ .

Having reached this point non-raindrop blobs remain (see figure 2.4b). To rule out most of them, we use the defocus aspect of the drop. Observation (C) states that unfocused drops do not exhibit strong gradients. To do so, we compute  $C$  the sum of the Canny at frame  $n - 1$  and  $n$ , figure 2.4c. The next step is to verify whether each connected component have borders that are edges in  $C$ . For each previously validated blob, we denote  $B_{edge} \in [0; 1]$  the ratio of blob contours that are also edges in  $C$ . Only blobs that validate  $B_{edge} \leq p_4$  are kept as possible drop, as seen in figure 2.4d.

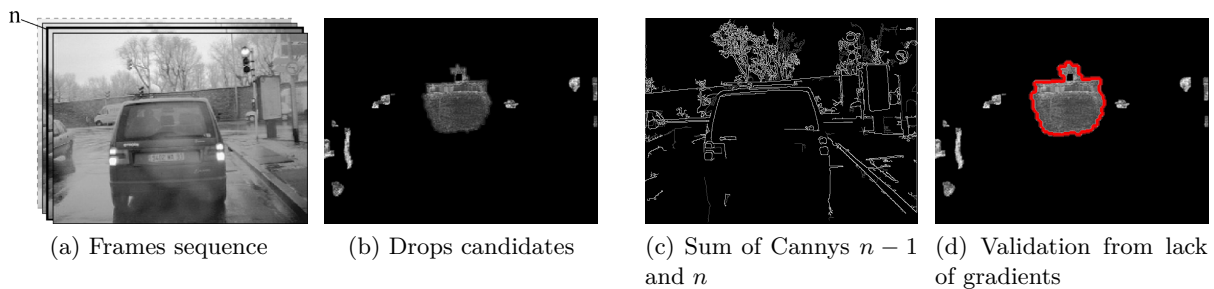


Figure 2.4: Using photometry and lack of gradients to detect unfocused drops. (a) is a sample sequence. (b) Is the map built from equation 2.1 with only pixels verifying raindrop photometry. (d) Is the result of *lack of gradients* where only the larger blob (highlighted) is validated since it shares few contours with the edge map (c) built from the sum of Canny's at  $n - 1$  and  $n$ .

Finally, we must verify that our strong assumption did not infer some incorrect segmentation. Recall, that so far we have assumed the scene to be static and the camera to be stationary whereas this is in fact not the case. Though we have compared the intensity between frames, the positive variation of intensity might have been caused by motion rather than by a drop apparition.

To verify this, we discretize the frame in  $5 \times 5$  blocks. For each block that belong to an alleged drop, we extract the strongest eigenvector and verify that it matches the block at same position in frame  $n - 1$  and  $n + 1$ . If it does not, this means that the block is not imaging the same portion of the scene. *In extenso*, either the ego-vehicle moved too much and the point of view is different, or an object (possibly a car) crossed the field of view. In both case, it proves that our previous comparisons were erroneous, hence the alleged drop is dismissed.

**Experiments** Using this algorithm we were capable of detecting unfocused raindrops in a *7min* video sequence of  $640 \times 480$  recorded from an in-vehicle camera. The method and detailed performance are published in [Nashashibi et al., 2010]. The parameters we used in our experiments are:  $p_1 = 10$ ,  $p_2 = 40$ ,  $p_3 = 2.7$ ,  $p_4 = 0.1$ . They were chosen empirically after several trials. The output from this *7min* sequence is fair and allows inferring a correct rain status. Other experiments with no rain shows it successfully *not* detect drops. However the drops are mostly detected during

simple portion of the rain sequences, e.g. when the car is moving slowly and the scene is rather static.

Indeed, as for many other raindrops detection algorithms, the output of our system might be valid despite several false detections if the number of the latter remains below a predefined threshold. Hence, from a *system* perspective this method provides fair performance. From a vision perspective, the algorithm however suffers from major drawbacks. First, it is very sensitive to the choice of the parameters. Second, the fact that it detects only the apparition of drops is not optimal. Hence, if a drop is not detected on its first visible frame there is no way to detect it later.

For these reasons, we believed we had to find other ways to detect unfocused drops.

## 2.3 Detection using blur assessment

We now investigate a drastically different approach relying on input image quality assessment. The main effect of defocus is blur and we use the latter to detect the drops by first computing the blur map of each frame. This map will serve to venture hypothesis about the positions of the unfocused drops in the image space.

**Blur definition** Defocus blur occurs when imaging objects that are out of the Depth Of Field of the camera, that is when a single point is projected on the image plane on a circle of confusion ( $\mathcal{E}$  in figure 2.2a) larger than the physical pixel size. In the image space a blurred image  $I_b$  may be described as a smoothing filter denoted  $H$  applied to an otherwise sharp image  $I_s$ , that is:

$$I_b(u, v) = I_s(u, v) \otimes H(u, v) + N(u, v) \quad (2.2)$$

$N$  being the additive sensor noise (often neglected), and  $\otimes$  represents the convolution. There are three major varieties of blur – defocus blur, motion blur and Gaussian blur –. Depending on the type of blur the kernel of convolution might be symmetrical or not. In the case where the kernel of convolution  $H$  (or Point Spread Function – PSF) is accurately estimated this allows restoring the true original sharp image. For standard circular aperture, the PSF of defocus blur is well approximated as a uniform disk with radius  $R$ :

$$H(u, v) = \begin{cases} 1 & \text{if } \sqrt{u^2 + v^2} \leq R \\ 0 & \text{else} \end{cases} \quad (2.3)$$

Despite that the PSF is defined by a single parameter  $R$  its estimation is not trivial because it is with two dimensions [Wu et al., 2009]. However, for most applications the goal is not much to estimate  $H$  but to rate the pictures that are being altered by blur. Hence, knowing only the order of the blur extent is sufficient to estimate picture quality and to rank pictures based on their blur amount. This makes also sense in terms of human perception since blur is a subjective notion that may vary from one individual to an other though individuals generally agree on the relative judgement (i.e. deciding whether a picture is blurrier or sharper seems easier than estimating its amount of blur).

**Reference methods versus No-Reference methods** It exists two different categories of blur assessment methods often named: *reference* based, or *no-reference* based. On one hand, the *reference* blur assessment uses two or more observations of the same scene (with different blur amount) and estimates which of the pictures is the sharpest. It is used for autofocus applications in most of the mobile cameras. On the other hand, the *no-reference* blur assessment (a.k.a. blind

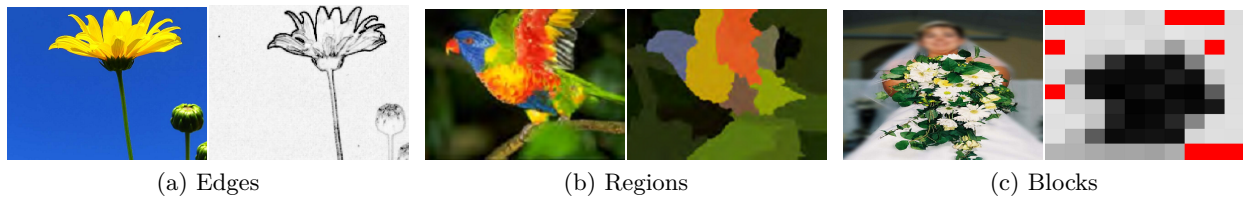


Figure 2.5: Local definition used for blur assessment. (a) Use of a pre edge map and estimation of the blur extent along the edges. Source: [Ke et al., 2006]. (b) Segmentation of the original picture and estimation of the blur for each regions. Here a top-down pyramidal segmentation. Source: [Da Rugna et al., 2003]. (c) Discretization of the original image in blocks of same size and estimation of the blur in each of them. Source: [Hsu and Chen, 2008].

blur assessment) consists of estimating the blur amount from a single observation of a scene. There is a surge of interest for such an approach given the recent development of applications using content based image retrieval (e.g. search engines, photo albums, etc.). For the detection of unfocused raindrops, we aim at estimating where blur occurs in each picture but except if the car is stationary there is no way for us to capture two pictures of the scene, with the same point of view and different blur amount.

We review in the next section the literature of no-reference defocus blur detection since it seems to be the sole conceivable approach for our application.

### 2.3.1 No-reference defocus blur detection - State of the Art

**Acknowledgment** The literature is mostly dealing with focus detection rather than defocus blur detection. Since we list either methods, the reader should be careful reading as the focus metrics decrease with blur, while defocus metrics increase with blur.

No-reference or blind blur assessment may be achieved either globally - on the entire picture - or locally - for each sub-regions -. The vast majority of the existing researches are meant to assess quality globally and thus estimate the overall blur extent. The problem of applying these methods on a local basis could be summarized as a problem of “lack of information”. Assuming that the regions are sufficiently detailed (texture, resolution, etc.) we might use any of the global blur assessment methods irrespective of whether the region is the sub-region of a larger one or not. In the literature three definitions (pictured in figure 2.5) are commonly used for local metrics:

- **Edges.** The blur extent is evaluated along the edges.
- **Regions.** A segmentation is first applied and each region is evaluated independently.
- **Blocks.** The image is divided into blocks of same size evaluated independently.

The measure of the blur along the edges is well suited for scenes that exhibit sharp edges. In the case where the out-of-focus subjects are separated from the rest of the scene (e.g. macro picture), a pre-graph segmentation is often used with a region-analysis. However, if the segmentation mix focused and unfocused regions as the tail of the parrot in figure 2.5b the correct blur estimation is impossible. In such case, a block-wise analysis is preferable as it makes no assumption on the image structure.

Reviews were published on quality metrics (among which [Lin and Jay Kuo, 2011] stands as a complete one) but they detail only briefly the blur assessment methods. Other short state of the art were published as introductions in [Liu et al., 2008, Wu et al., 2009] and provide a decent overview of the existing methods at the time. Here we provide a comprehensive listing of metrics designed for blind blur assessment . We have grouped the metrics in three categories depending on whether they use gradients, frequency, chromatic, or statistic of natural scenes.

### 2.3.1a Gradients analysis

Gradients are prominent features affected by defocus blur. Edges are smoother when defocused, so their width become larger. We detail the metrics that use such features.



Figure 2.6: Difference of gradients to detect defocus blur using [Crete et al., 2007]. Left is a sharp version of a picture. Right is an digitally blurred version. The method correctly ranks the right picture as blurrier using the difference of derivatives.

**Difference of gradients** Despite that blur operation is not idempotent, the sharper is a picture the higher the differences to a blurred version of this picture. The easiest blur metric [Crete et al., 2007] uses this property to compute the blur extent from the difference of the derivatives of the original image with those of an artificially re-blurred picture. The re-blurred image is the result of the convolution with an average filter. The maximum negative variations of the horizontal and vertical derivatives serve as a blur index which is weighted to range from 0 to 1. An example of this method is shown in figure 2.6. The method Crete *et al.* propose is popular for its simplicity to put in practice<sup>1</sup>. A small improvement uses derivatives in all directions [Ojansivu et al., 2011] but does not seem to change the results considerably.

**Edges width** Having computed the edge map from Canny, Kurtosis, Sobel, or wavelet decomposition a classical approach is to compute the blur metric from the edges width. A simple method is [?] where they estimate the edge's spread from the nearest intensity local minima or maxima on the x-abcissa (i.e. it works only for vertical edges). The idea was extended later for all edges using the intensity profile along the edge's normal (i.e. gradient direction). [Chung et al., 2004] proposed to fit a normal distribution (the variance being used as the blur extent). The problem of using only intensity profile is the reliability of both edge location and gradient width. This issue is discussed in the early work of Elder and Zucker who proposed a different mathematical framework. In their paper [Elder and Zucker, 1998], the edge location is computed from the zero-crossing of the second derivative of the intensity function along the gradient direction. The gradient's width is obtained from the distance of the lower and upper extrema of the second derivative that is estimated with

<sup>1</sup>Matlab code is available at: <http://www.mathworks.com/matlabcentral/fileexchange/24676-image-blur-metric>

a steerable second derivative of the Gaussian operator. The strength of this model is that it is robust to low Signal-Noise Ratio (SNR) assuming that the sensor noise is a zero-mean white noise with an *a priori* known variance. The three above-cited methods are depicted in figure 2.7.

For each edge the blur extent is computed from its width [Elder and Zucker, 1998, ?] or the slope of the gradient [Wang et al., 2008] (i.e. basically the same). Often the edge width is compared to those of the sharpest edge (that is the edge with the minimum width) or to the number of edges in the picture [?]. In [Chung et al., 2004] they compute the standard deviation of the edge intensity profile to the normal distribution and use the latter together with the magnitude to estimate the sharpness of the edge. The underlying of these approaches is that the picture contains sharp edges and that the smooth edges all result of sharp edges smoothed by defocus blur.

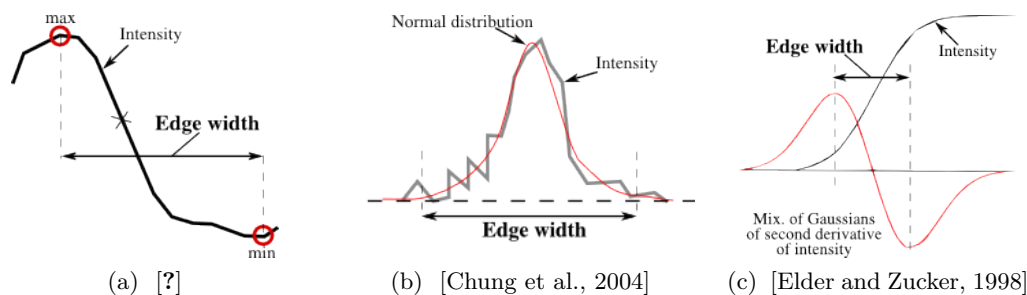


Figure 2.7: Strategies to measure the edge spread to detect defocus blur. Here, three schemes illustrate the methods commonly used when assessing blur.

**Edges characterization** An other approach consists in the characterization of the edges into: Dirac, Roof, A-Step or G-Step, based on the important fact that defocus blur induces the disappearance of both Dirac structures and A-Step structures. [Rooms et al., 2002] propose to detect smooth edges (i.e. G-Step or Roof) in the wavelet domain and to characterize the later given their Lipschitz exponent<sup>2</sup>. When applied to the whole picture Rooms *et al.* states that a peak should appear in the Lipschitz histogram (i.e. corresponding to the blur extent) whereas this is in fact only true if all edges of the virtual focused image are of the same sharpness.

A popular paper is [Tong et al., 2004] where the edges are directly classified into one the four aforementioned types. From the multiscale decomposition with Haar Wavelet Transform (HWT), Tong *et al.* found that it exists a unique relation of the wavelets responses for each of the four edges types. Hence, they classify each edge given its wavelet response and the overall blur extent is computed from the ratio of smooth to sharp edges. The approach of Tong *et al.* is rather popular because its implementation is both simple and efficient. However, an important drawback is its sensitivity to noise since the straightforward classification assumes a clean signal. On the other hand, the classification is made regardless of the gradient orientations which also allows detecting anisotropic blur (e.g. motion-blur).

### 2.3.1b Frequency analysis

In the frequency domain, the defocus blur is characterized by the reduction of high frequencies due to its circular Point Spread Function (PSF). Some have proposed to detect the lack of high frequencies or to estimate the PSF from the spectral distribution.

<sup>2</sup>The Lipschitz exponent is a measure of how many times a signal is differentiable. For image processing, this is used as a smoothness metric.



Figure 2.8: Spectral analysis to detect defocus blur using weighted frequencies in patches of  $8 \times 8$  to highlight the element in focus. Note that even sharp regions have focus values close to 0. Source: [Dai and Wu, 2007].

**Spectrum** Trivial counting of the high frequencies that exceed an empirical threshold is used in [Ke et al., 2006] to assess sharpen areas of a photography. The idea is extended in [Dai and Wu, 2007], where the power spectrum of the Fourier is convolved by a 2D kernel that promotes high frequencies. In [Dai and Wu, 2007] the sum of the results of the convolution is normalized using the sum of pixel values. The drawback of such normalization is that even sharp values tend to be close to 0. This rather simple approach provides good performance when evaluated on input images with negligible noise and shallow depth of field, as seen in figure 2.8.

**Cepstrum** In the frequency domain the PSF of the defocus blur exhibits radial zero-crossing that are nearly periodic. As stated by Cannon, “although the zero-crossings are well defined theoretically, they are obscured in practice by the extreme randomness” of the input image and the noise [Cannon, 1976]. An early proposal is thus to analyze the cepstrum domain where the zero-crossings of a cylinder should appear as periodical negatives rings. The cepstrum being the Fourier inverse of the power spectrum logarithm<sup>3</sup>. Logically it is easier to detect a global negative than near-zero values. Besides, theoretically the maximum negative peak should occur at  $2R$  ( $R$  the radius of the defocus PSF, see eq. 2.3) which allows retrieving the PSF [Rom, 1975]. The true improvements to increase its robustness to noise were proposed by [Cannon, 1976] and [Fabian and Malah, 1991]. In the former, spatial filters are applied and in the latter Fabian and Malah posited that additive Gaussian-noise may be suppressed through spectral subtraction and thus apply an additional cepstral filter.

The nice property of the cepstrum analysis is that there is no need of any assumptions about the true image structure. [van Zyl Marais and Steyn, 2009] is a good application of the cepstrum analysis to identify the presence of defocus blur. The south african’s team have shown the superiority of the cepstrum analysis against other spectral methods when applied to images of unknown structures, assuming that the resolution is sufficiently large.

### 2.3.1c Chromatic

Generally speaking, chromatic is not used to assess blur since each color channel is affected evenly. Furthermore, experiments conducted in [Crete et al., 2007] exhibited that using color does not

<sup>3</sup>The mathematical definition of cepstrum  $\mathcal{C}$  is  $\mathcal{C}(u, v) = \mathcal{F}(\log |\mathcal{P}(u, v)|)$ , with  $\mathcal{F}$  the Fourier transform and  $\mathcal{P}$  the power spectrum. But one should note that there are other definitions using the inverse Fourier transform instead. Instructive overview of the cepstrum properties can be found in [Rom, 1975].



improve the performance of blur detection significantly.

**Saturation** Still, [Liu et al., 2008] use the saturation of a local area compared to the maximum picture-saturation. The authors posited that local blurred areas exhibit lower saturation when compared to the maximum picture saturation. The metric from [Liu et al., 2008] can only assess local blur, and remains valid only if the true image structure is *similar* in the blurred and sharp areas.

### 2.3.1d Statistics of natural scenes

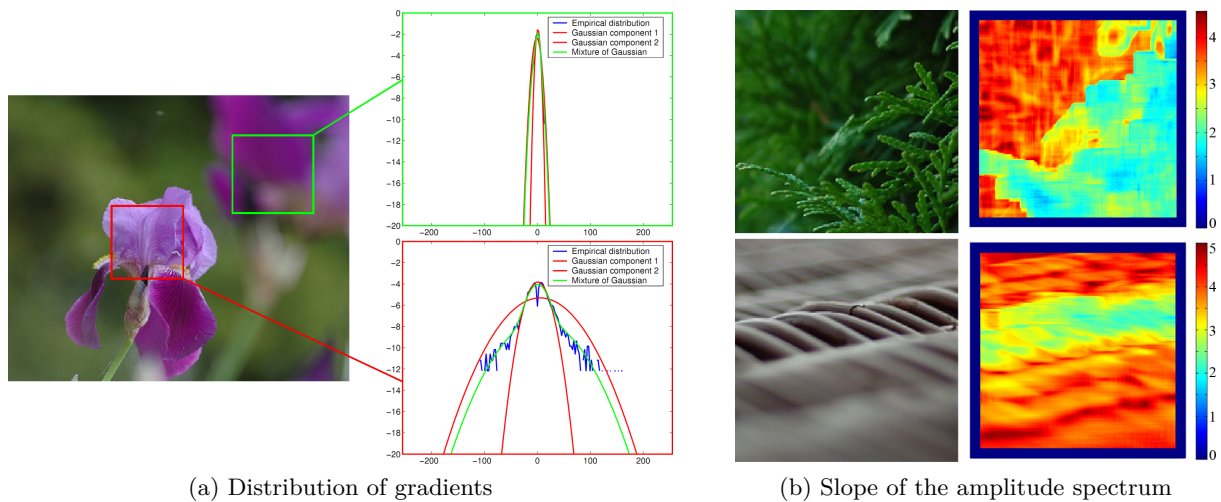


Figure 2.9: Using statistics of natural scenes to infer blur maps. (a) Local slope of the amplitude spectrum when calculated over patches of  $17 \times 17$ . (b) Two sample distribution of gradients for a defocused patch (green) and a focused one (red). From (a) and (b), the two metrics exhibit high correlation with the defocus blur amount. Still, reader should note that the photographs have very shallow depth of field which is dramatically easier. Source: [Liu et al., 2008].

Finally, based on the discovery that natural scenes share some statistical properties two other features are used to assess blur: the slope of the amplitude spectrum and the distribution of gradients. The study of these shared statistical properties is a wide promising field and for more information the reader should refer to [Field, 1987, Tollhurst et al., 1992, Cecchi et al., 2010].

**Distribution of gradients** Natural scenes tend to exhibit heavy-tailed gradients distribution. Together with other features [Hsu and Chen, 2008, Liu et al., 2008] use the shape of the gradients distribution as a weak blur feature. In [Liu et al., 2008] a mixture of two Gaussian is used to fit the distribution. The Gaussian with the maximum variance being responsible of the tail. This is depicted in figure 2.9a. According to Liu *et al.* this property “alone is not suitable to be used directly as blur detection”. Indeed, this measures in fact the likelihood to be a natural scene structure with acceptable sharpness. The above mentioned authors used a contrast feature to balance this feature.

**Slope of the amplitude spectrum** The study of the human visual system shows that cortical cells discriminate blur using a second derivative statistic that is the *slope of the amplitude spec-*

*trum*<sup>4</sup>. In natural scenes the local variation of this slope is responsible of the feeling of blurriness. In image processing, the slope is obtained from the exponential fitting of the histogram of the frequencies-amplitudes (the computational details are provided in section 2.3.2b). The steeper the frequency-amplitude histogram, the bigger is the slope and the blurrier is the picture [Tadmor and Tolhurst, 1994]. [Liu et al., 2008] used this property to promote blurred regions based on their local slope when compared to the global slope of the scene. This feature exhibits good performance for pictures captured with shallow depth of field, as seen in figure 2.9b.

### 2.3.2 Blur detection for unfocused drops detection



Figure 2.10: Sample images with defocused drops from our database. Note in (b) that the drops are not easily distinguishable when imaged against low textured background (sky, plain wall, etc.).

Figure 2.11 shows some sample images acquired with a high dynamic camera (Omnivision,  $1,280 \times 800pxl$  @25FPS) with settings that exhibit unfocused raindrops. They correspond to the kind of drops appearance we wish to detect. Several observations can be drawn from the analysis of a complete rain sequence:

- (A) Unfocused raindrops do not produce a strong visible defocus blur and occupy only a small portion of the whole picture. In contradiction, the samples used in the literature of blur assessment all exhibit strong and large defocused areas which is consequently easier to detect.
- (B) The raindrops are not easily distinguishable from their background. When displayed over low textured background (e.g. sky, plain wall, etc.) – as in figure 2.10b – even observers were unable to see the drops.
- (C) When they hit the front screen the shape of the drops is roughly circular but the streaming due to vehicle speed or wipers movement make their exact shape unpredictable, as in the bottom left corner of figure 2.10c.
- (D) Temporally the raindrops generally remain static but they are only visible for a fraction of second when the wipers are activated to their highest speed. In [Wetzel et al., 2004] it is stated that the highest speed of sedans wipers is 3.2 wiping per second<sup>5</sup> which correspond to a maximum of 7 frames (@25FPS) until the drops are being wiped out.

<sup>4</sup>The study of the amplitude spectrum is a wide field in vision. We refer to [Hansen and Hess, 2006] as a well-written comprehensive study on the discrimination of the amplitude spectrum slope in the human visual system.

<sup>5</sup>Given that the two front wipers cover the whole camera FOV, and that each wiper cycle include two wipings (on the way down and the way back).

In light of these observations, the pictures we are dealing with are challenging essentially due to the small blur extent and the small size of the blur patches (observation A). To detect the latter we aim at computing a blur map which is accurate enough to identify these small local variations of blur. The blur map is computed from a block-wise strategy as it makes no *a priori* on the true image structure or on the shape of the defocused drops (observation C). The input image is thus discretized in square blocks of  $N \times N$  and each block is evaluated independently. The blocks step distance for the blocks is  $D \in \mathbb{Z}$ , with  $1 \leq D \leq N$  (i.e. if  $D < N$  the blocks overlap). To ensure that *at least* one block lies entirely in the smallest drops  $N$  should be fixed such that it is smaller than the minimum radius of the raindrop in the image space.

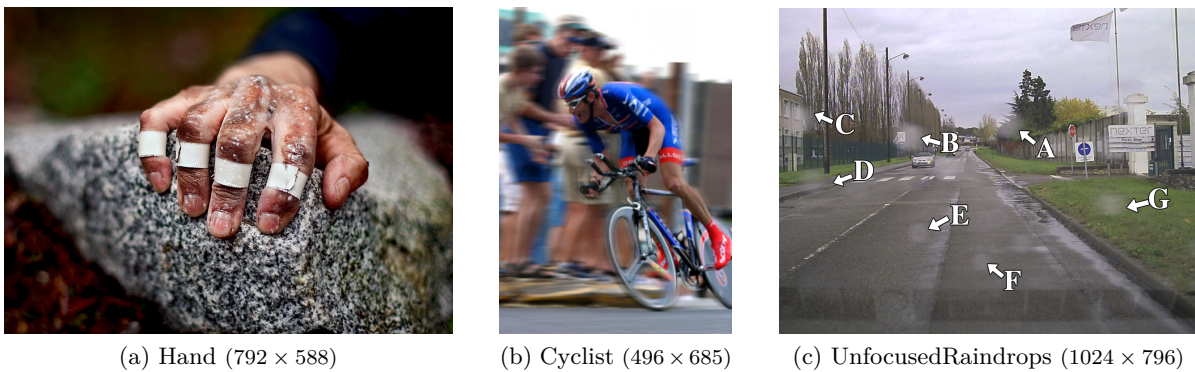


Figure 2.11: Sample pictures used for qualitative evaluation of the proposed methods. (a) and (b) are typical blur assessment pictures. (b) contains both defocus blur and motion blur. (c) is a sample frame from our unfocused raindrops sequence. The scene is rather complex and the blur to detect is light which makes this picture the most difficult.

To investigate this idea we have tested out a number of blur detection methods that assess blur either globally or locally. The conclusion is that none of the blur assessment methods we propose is sufficient to *accurately* detect the drops but some of them exhibit promising preliminary results. We describe three of them that we think are the most promising and would benefit of deeper investigation. In the first proposed approach, section 2.3.2a, we characterize each edge through wavelet transform. Two other approaches are detailed next using frequency analysis through weighted Fourier (section 2.3.2c) or the slope of the amplitude spectrum (section 2.3.2b). *In fine*, we will discuss how deeper investigation of this problem could lead to an accurate detection of unfocused raindrops. Other non-described method like the Cepstrum analysis has shown great success but in specific scenarios (e.g. very heavy rain).

For each experiment we show the blur map for three sample pictures, displayed in figure 2.11. The first two (*Hand* and *Cyclist*) are often used for blur assessment. *Hand* is a rather simple picture as it exhibits both shallow depth of field and strong black and white pattern on the rock. The difficulty is to build a smooth blur map that shows the graduate defocus blur. *Cyclist*, is very well known in blur assessment and exhibits both defocus and motion blur at the same time. The challenging part is the teeshirt that exhibits no strong gradients though in focus, the thin front wheel, and the background with gradients along the motion direction. Finally, *Unfocused Raindrops* is clearly the most complicate picture since the scene has different levels of details and since the drops exhibit a light defocus. From now on, we refer to these pictures as *Hand*, *Cyclist*, and *Unfocused Raindrops*.

### 2.3.2a Haar Wavelet Transform and Edges characterization

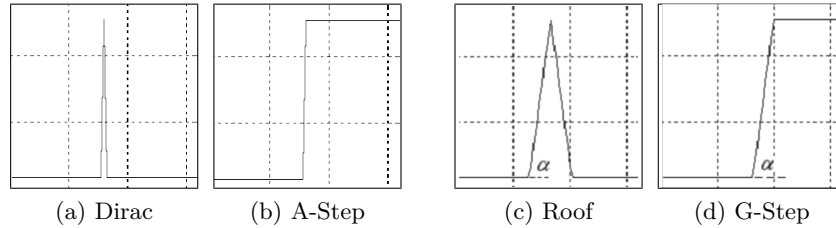


Figure 2.12: Type of edges used for blur detection. Dirac (a) and AStep (b) are sharp edges while Roof (c) and GStep (d) are smooth edges which smoothness depends on the value of  $\alpha \in ]0; \frac{\pi}{2}[$ .

Edges can be separated into four categories that are: Dirac, Roof and AStep and GStep given their sharpness, cf. figure 2.12. Dirac and AStep are sharp edges while Roof and GStep are smooth edges which smoothness depends on the value of  $\alpha$  (with  $\alpha \in ]0; \frac{\pi}{2}[$ ). The smaller the  $\alpha$  value, the smoother is the edge.

**Idea** The basic idea behind edge classification for blur detection is that when out-of-focus blur occurs we observe that sharp edges disappear and smooth edges become even smoother. For now let's considerate the following assumption to be true: all smooth edges are defocus blurred version of sharp edges. This assumption is used in most blur-assessment methods but is obviously *biased* as smooth edges may also result of scene-structure, shadows, etc. But again, let's considerate the assumption is valid.

From the above assumption, a local blurred area should exhibit a lack of sharp edges. Ergo, given that Dirac and AStep are sharp structures, they indicate a region is in focus whereas GStep and Roof structure are smooth and assess of the presence of blur. To classify the edges and estimate their smoothness (inverse of  $\alpha$  in figure 2.12), we use a wavelet decomposition approach presented in [Tong et al., 2004]. The authors proposed using a multi-scale Haar Wavelet Transform (HWT) to classify each edge based on its response to wavelets. Wavelet decomposition is a standard image processing tool that approximates a complex signal with a variety of wavelets<sup>6</sup>. A pyramidal representation is shown in figure 2.13 for three levels of decomposition.

Having applied a 3-levels HWT decomposition, Tong *et al.* have shown that there is a singular relation between the value of the edge map at each scale and the type of edges. The relation of edges type and response to wavelets is illustrated in table 2.13b. As an example, the response of wavelet decomposition to Dirac edge at level 1 is higher than the response at level 2 which again is higher than response at level 3. In keeping with the relation highlighted in figure 2.13b, it is possible to classify the edges into Dirac, AStep, GStep or Roof. We wish to use the type of edges in each block as a way to assess of the defocus blur extent. Next, we describe how to implement this idea and the results of our experiments.

**Algorithm** As a summary, this approach aims at classifying the edges and estimate their smoothness, from the response of wavelet decomposition.

We first apply a 3-level HWT decomposition on each block of the picture (recall that the input picture was discretized into  $N \times N$  blocks). At each level  $i$ , the result of the wavelet decomposition is four sub-band images that correspond to horizontal detail ( $HL_i$ ), vertical detail ( $LH_i$ ), diagonal

<sup>6</sup>The reader may refer to [Vidakovic and Mueller, 1994] as an easy-to-read introduction to wavelets processing.

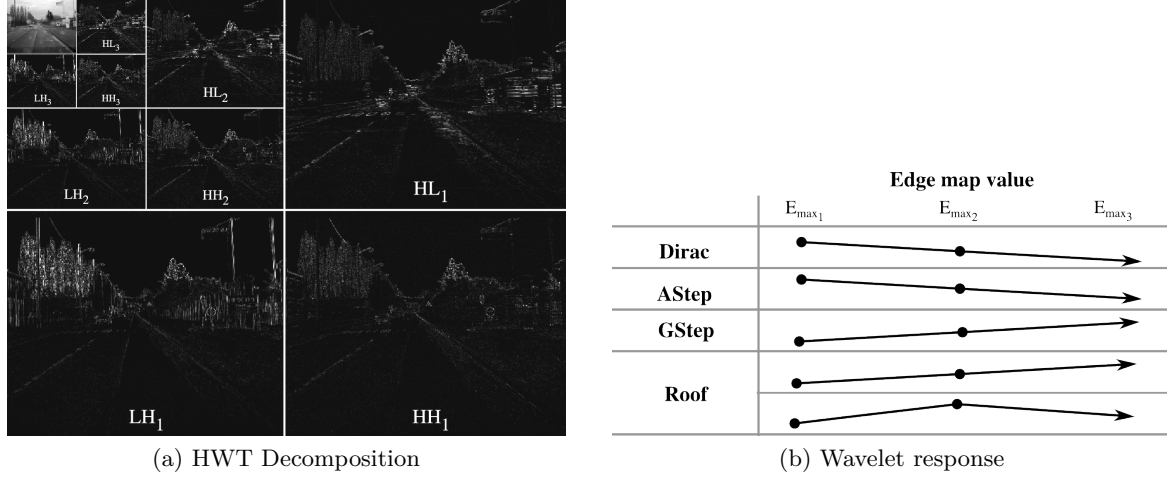


Figure 2.13: Wavelet decomposition and response to different type of edges. (a) Pyramidal representation of 3-level decomposition with HWT. For readability, brightness is +50% and contrast +30%. (b) The relation between wavelet response and the edge type.

detail ( $HH_i$ ) and a fourth one corresponding to the low pass filter ( $LL_i$ ). Again, refer to figure 2.13a for a visual output. The prerequisite to classify the edges is to build an edge map  $E_{map}$ , which is achieved at each level such that:

$$E_{map_i} = \sqrt{LH_i^2 + HL_i^2 + HH_i^2} \quad \text{with, } i \in \{1, 2, 3\} \quad (2.4)$$

The edge maps are then partitioned with window of decreasing size at each scale ( $8 \times 8$ ,  $4 \times 4$  and  $2 \times 2$ ) to reflect the wavelet decomposition downscaling. The coarsest window ( $8 \times 8$ ) being for the first scale ( $i = 1$ ) and the finest ( $2 \times 2$ ) for the highest scale ( $i = 3$ ). The result of the partition process is a map denoted  $E_{max_i}$  which value is computed from the maximum value of  $E_{map_i}$  over the partition window. Hence, the partition maps  $E_{max_i}$  at all scales (1, 2, 3) have the same size. To decide whether in each map a point  $(u, v)$  is an edge we use a threshold  $t$ . For each edge detected in the first max edge map ( $E_{max_1}(u, v) \geq t$ ), the value at level 2 and 3 is computed and the edge is classified in keeping with the observations shown in table 2.13b. Note that, following the observations of Tong *et al.* it is not always possible to differentiate GStep from Roof type, but since both of them correspond to smooth edges there is no need to do so.

Now, to identify whether a  $N \times N$  block is defocused we count the number of edges in each block. For GStep and Roof structure we estimate their smoothness from the output value of the partitioned map at level 1 and 2. To compute the blur extent  $\Lambda(u, v)$  of the edge  $(u, v)$  (recall  $E_{max_1}(u, v) \geq t$ ) we propose to compute the ratio between the response at level 1 and the highest response at level 2 or 3. Hence:

$$\Lambda(u, v) = \begin{cases} 0 & \text{if } (u, v) \text{ is an edge } \mathbf{Dirac} \text{ or } \mathbf{AStep} \\ 1 - \frac{E_{max_1}(u, v)}{\max(E_{max_2}(u, v), E_{max_3}(u, v))} & \text{if } (u, v) \text{ is an edge } \mathbf{GStep} \text{ or } \mathbf{Roof} \end{cases} \quad (2.5)$$

with  $\Lambda \in [0; 1]$ . The block blur index  $B_\Lambda$  is computed as the mean of the blur indices of all edges in block  $B$ , so that  $B_\Lambda \in [0; 1]$ . If no edge was detected in block  $B$ , the block corresponds to a flat region and the blur extent is not rated.

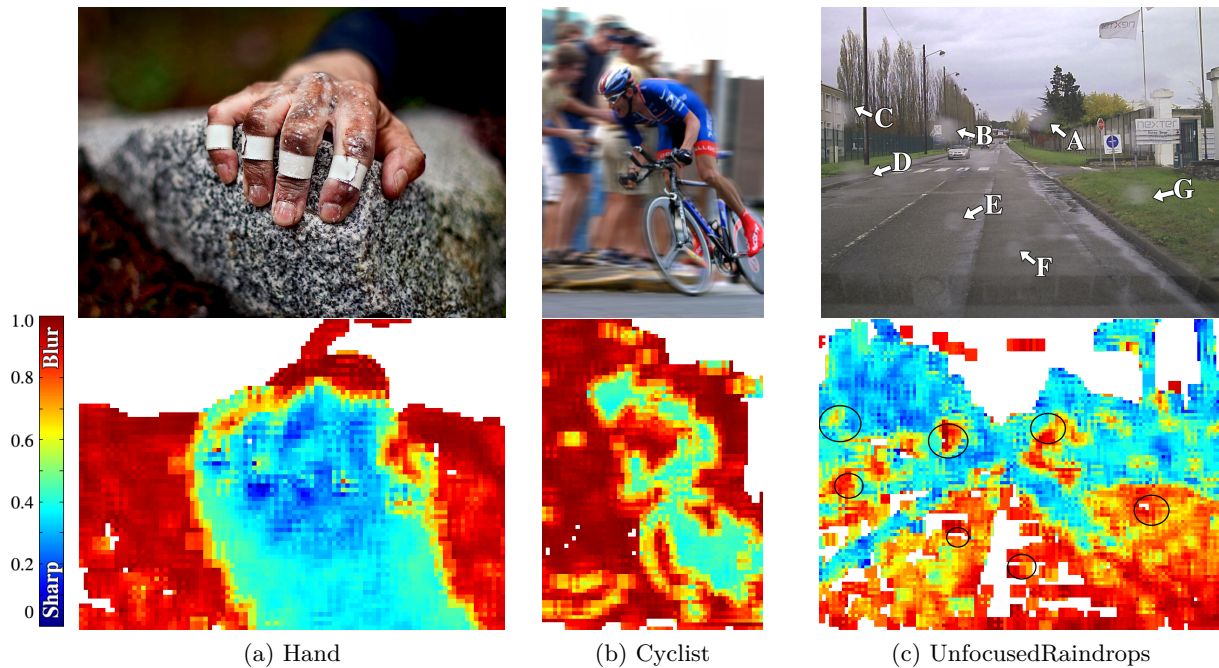


Figure 2.14: Estimation of the blur from multiscale Haar Wavelet Transform. The color scale is shown on the left. Transparent/white areas are not ranked since no edge was found ( $E_{max_1}(u, v) < t$ ). In (c) the drops are circled in the blur map for readability.

**Experiments** From the proposed implementation we compute the blur maps for the three reference pictures, see figure 2.14. The flat regions (*unrated*) are shown transparent in the blur maps. The parameters are:  $t = 0.1$  (edge threshold),  $N = 32$  (block size), and  $D = 8$  (block step). Note that  $N$  should be fixed such that,  $N = k \times 8$  (with  $k \in \mathbb{Z}$ ) since the coarsest window is  $8 \times 8$ .

From the blur maps in figure 2.14, we see that the blur detection is working efficiently especially for *Hand* and *Cyclist*. Hence, the blur areas are correctly identified without any high level segmentation. For *Hand*, the blur map gradient show that with this we are capable of detecting and rate the amount of blur. The blur map of *Cyclist* is however less accurate. Some blocks are seen sharp due to the motion blur effect that exhibits strong edges in the motion direction (e.g. along the wall). On the other hand, the cyclist (but his tee-shirt) is correctly seen in focus as for the front wheel, even though it exhibits few gradients.

Concerning the picture *UnfocusedRaindrops*, the blur map is much more complex to analyse. At first sight, the map is noisy and the raindrops are not well separated from their surroundings. However, despite the chaotic aspect of the map all drops – except drop C – belong to blocks ranked blurred. When compared to blur maps from *Hand* and *Cyclist*, detecting drops as blurrier than their surroundings is difficult (or even impossible). In picture *Hand* and *Cyclist*, the border of the blur is the border of the element in defocus whereas drops may be imaged over two objects (e.g. drops A-D) and thus exhibit strong edges.

Though we used the general idea from [Tong et al., 2004], the latter were only evaluating the blur globally. The scheme we proposed to turn their global metric to a local blur metric seems to work well for photographs. Regarding that we use no smoothing process or semantic, the output is acceptable. However, for the sake of unfocused raindrops detection the output seems too

noisy. Additionally, the discrete output of the edge characterization induces that the blur extent is somewhat discrete too, which prevent from a fine evaluation of the blur extent.

### 2.3.2b Slope of the amplitude spectrum

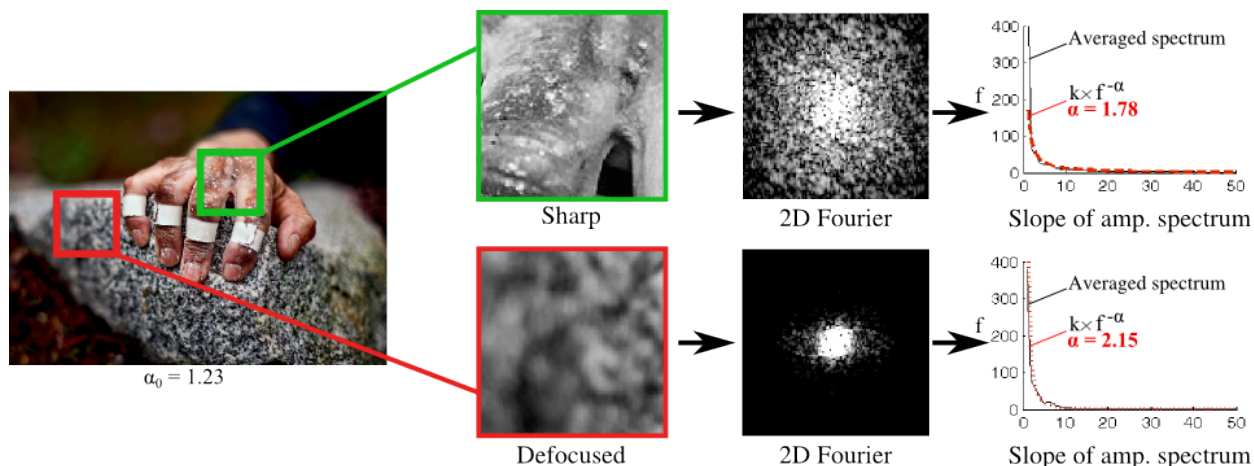


Figure 2.15: Slope of the amplitude spectrum of a sharp block (green) and a defocused block (red) in *Hand*. The process is *illustrated* while omitting multiple intermediate steps. From left to right: a block, its 2D Fourier log magnitude, and the plot with the averaged spectrum and its exponential regression ( $k \times f^{-\alpha}$ ). As expected, the blurred block exhibit higher (steeper) alpha value.

**Idea** As we discussed in section 2.3.1d, natural outdoor scenes share a number of statistical properties among which the slope amplitude spectrum stands as an important one. [Field, 1987] has first shown that the distribution across spatial frequencies  $f$  regardless of their orientation falls by a factor of  $f^{-\alpha}$  —  $\alpha$  the *slope of the amplitude spectra*. While this has lead to a vast number of studies advocating that all natural scenes have a similar slope of  $\alpha \approx 1.2$  [Tolhurst et al., 1992], more interesting is to say that human visual systems is optimized to exploit the falloff to discriminate the alpha. This is notably used to discriminate non blurred areas from blurred areas that exhibit steeper alpha values [Tadmor and Tolhurst, 1994, Hansen and Hess, 2006]. In other words, a portion of the image that is blurred has a higher alpha value when compared to the overall alpha value of the picture.

From figure 2.15, the *defocused block* from picture *Hand* exhibits a larger alpha value ( $\alpha = 2.15$ ) compared to the *sharp block* ( $\alpha = 1.78$ ).

**Algorithm** Among other ways to measure the power spectrum of the Fourier transform is often used to compute *alpha*, which is convenient for us. Hereafter we denote a block  $B$ . Prior to any process in frequency domain, we apply a Gaussian windowing filter on the block to avoid *artificial high frequencies* that would result of a border effect. Hence:

$$B_w = B \times G(\sigma) \tag{2.6}$$

where  $G(\sigma)$  is a 2D centred Gaussian with  $\sigma = (N - 1)/4$ . As this reduces the “useful area”  $N$  has to be large enough.

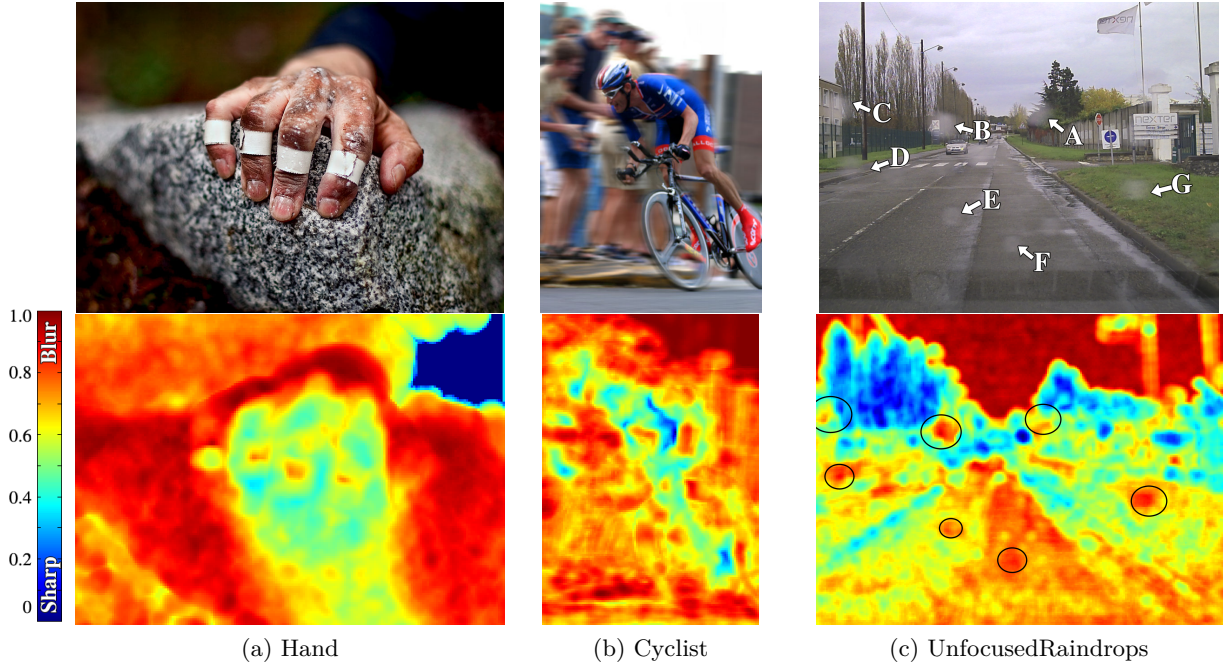


Figure 2.16: Estimation of the blur from the slope of the amplitude spectrum. The color scale is shown on the left. In (c) the drops are circled in the blur map for readability.

After computing the power spectrum  $\mathcal{P}$  of the block  $B_w$  (as in equation 2.10), we transform the power spectrum 2D space to a polar coordinates  $(f, \theta)$ , where  $f$  is the frequency and  $\theta$  the orientation. Frequencies are then averaged over all orientations  $\theta$ , such that:

$$\mathcal{Q}(f) = \sum_{\theta \in [0; 2\pi]} \mathcal{P}(f, \theta) \approx k \times f^{-\alpha} \quad (2.7)$$

$k$  an unused constant. The  $\alpha$  value is approximated through regression. To tackle the problem we use a linear regression on  $\log(\mathcal{Q})$  which is easier and less costly than an exponential one. As stated by [Hansen and Hess, 2006] the alpha value is steeper for blurred area but only on a relative basis — that is when comparing to the global alpha value. Hence, prior to this step we compute  $\alpha_0$  the slope of the picture and use the relative blur extent of the block  $B_\Lambda$ , such that:

$$B_\Lambda = \frac{\alpha - \alpha_0}{\alpha_0} \quad (2.8)$$

**Experiments** The blur maps for the three reference pictures are displayed in figure 2.16. The parameters are:  $N = 49$  (block size), and  $D = 4$  (block step). Note that  $N$  is larger than for HWT since the Gaussian windowing reduces the “usefull area”. Also to avoid asymmetrical artifacts from the Fourier transform  $N$  is odd.

From figure 2.16, one can see that when compared to the blur maps with HWT the dynamic of the map is much higher with this method which provides a finer blur analysis. For *Hand*, the map is of very good quality, since it highlights correctly the sharp area and exhibits a high dynamic. Conversely, the map for the *Cyclist* is not at all accurate but reader should note that the slope of the amplitude spectra *can only* estimate the defocus blur. As *Cyclist* exhibits strong gradients along the motion direction the inaccurate blur map is predictable.



Interesting conclusion can be drawn from the *UnfocusedRaindrops* blur map. Conversely to HWT, the map is not only finer but can better distinguish the drop from their background. In this case, drops B,D,E,G could be quite easily extracted using a rank order filter for instance. Also one should note that it is capable of accurately estimating the sharp appearance of the left side of the road, which we found to be a recurrent issue.

We now investigate another spectral analysis using simpler process.

### 2.3.2c Weighted FFT

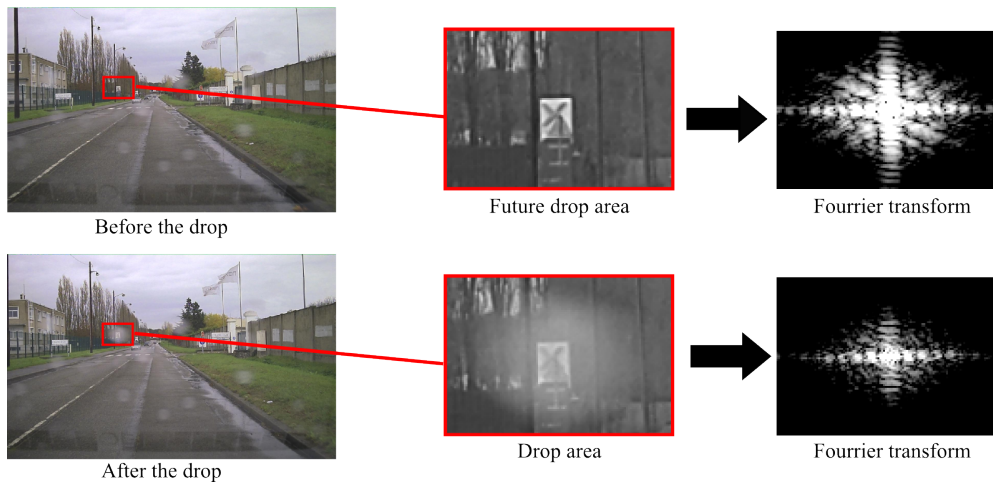


Figure 2.17: Fourier transform of a region before and after a drop hit the screen. From the logarithmic Fourier magnitude (right) of the sub-regions it is easily noticeable that the magnitudes of the high frequencies lowered. Here the sharp background emphasizes the reduction of high frequencies.

**Idea** In the frequency domain, blur induces the reduction of high frequencies. Figure 2.17 shows the Fourier transform of a sub-portion of a scene before and after a drop hit the windscreen, producing a blurry effect. The magnitude of the Fourier is lower (i.e. darker) with the blurred drop. In this specific sample the effect is strong – thus easy to see – thanks to the sharp *drop background*. While some regions might not exhibit high frequencies, a simple way to rule out regions that do so is to assess of there magnitude. [Dai and Wu, 2007] have proposed a simple way to estimate the focus of a region using a weighted kernel to count the high frequencies.

**Algorithm** As for the amplitude spectrum, we first apply a Gaussian windowing to avoid border effect. Then, following the proposal of Dai and Wu, we use a weighted kernel  $W$  to sum all high frequencies (i.e. the more high frequencies the more in focus) and compute a *focus* score:

$$\sum_{(u,v) \in B_w} \mathcal{P}(u,v)W(u,v) \quad (2.9)$$

where  $\mathcal{P}$  is the power spectrum of the Fourier transform  $\mathcal{F}$  and is computed as:

$$\mathcal{P}(u,v) = \sqrt{|\mathcal{F}(u,v)|^2} \quad (2.10)$$

$u$  and  $v$  the x and y coordinates with  $(u = 0, v = 0)$  the center of the block.  $W$  is the 2D weighted kernel that has the same size as the block. In contradiction to Dai and Wu, we use an inverse

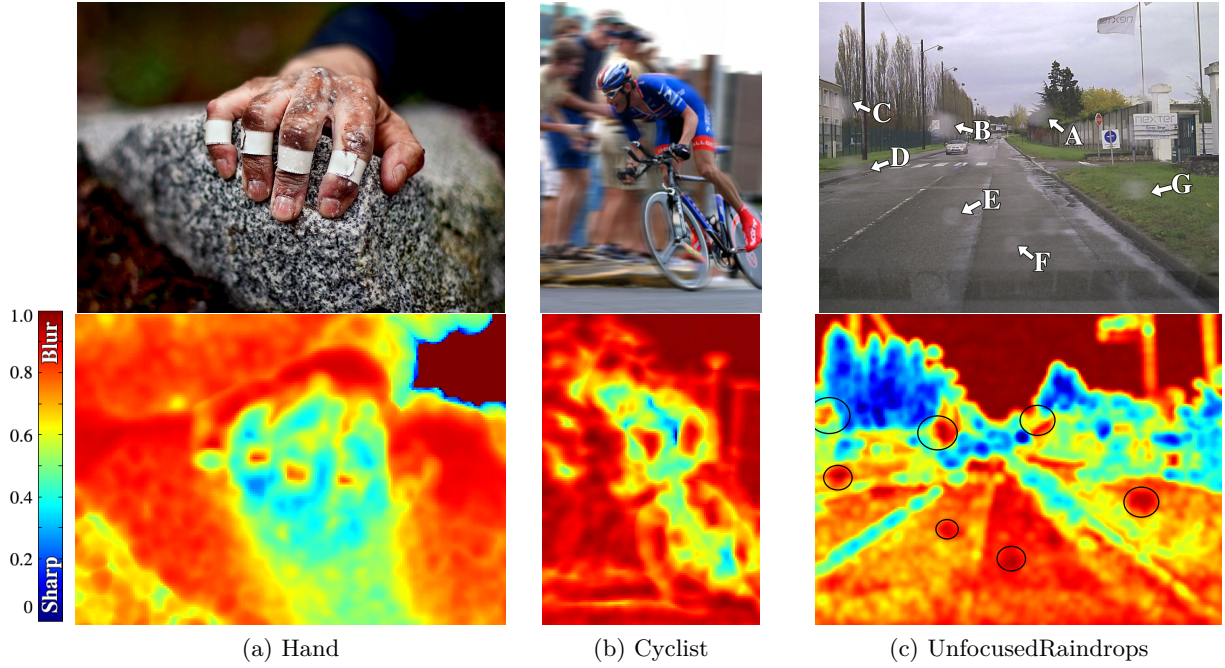


Figure 2.18: Estimation of the blur using weighted frequencies. Color scale is shown left. In (c) the drops are circled in the blur map for readability. *This figure is better seen on a computer as the “valleys” in the circled drops may not be visible when printed.*

exponential clamped kernel which computes the weight of a frequency from the distance to the zero frequency. Hence:

$$W(u, v) = \begin{cases} \sqrt{e^{-((N-u)^2+(N-v)^2)/\sigma_w}} & \text{if } \sqrt{u^2 + v^2} \leq (N - 1)/2 \\ 0 & \text{else} \end{cases} \quad (2.11)$$

with  $\sigma_w = 0.25N^2$ . The kernel aims at promoting blocks in focus and thus values more the high frequencies. The kernel is clamped so that frequencies which distance to the center is greater than the block radius  $-(N - 1)/2$  are simply ignored. The reason is that these *corner frequencies* are only computed on few angles and thus not fully reliable. By removing these values we increase the robustness to noise.

To normalize the focus score from equation 2.9 it is divided by the sum of the power spectrum. Hence, the real focus score  $f$  is such that:

$$f = \frac{\sum_{(u,v) \in B_w} \mathcal{P}(u, v) W(u, v)}{\sum_{(u,v) \in B_w} \mathcal{P}(u, v)} \quad (2.12)$$

The main problem of this metric that counts frequencies is that  $f$  is always close to zero due to the normalization process. This makes complicate the identification of focused/blurred regions since any blur threshold will be very sensitive. This is also noticeable in the experiments of [Dai and Wu, 2007]. Hence, instead of computing the absolute focus of each block we compute the relative focus to the rest of the picture. So, to compute the focus of each block we first compute

the focus score  $f_0$  of the whole picture (as in equation 2.12). Then the block blur extent  $B_\Lambda$  with weighted FFT is defined as:

$$b = \frac{(1 - f) - (1 - f_0)}{(1 - f_0)} = \frac{f_0 - f}{1 - f_0} \quad (2.13)$$

**Experiments** Figure 2.18 shows samples blur map computed with the Weighted FFT (WFFT). The parameters are:  $N = 49$  and  $D = 4$ . Again, to avoid artefacts from Fourier and due to the windowing process,  $N$  is odd and larger than for HWT.

Once again the WFFT provides a much finer and accurate blur map than the HWT, but comparable to the slope of the amplitude spectrum (section 2.3.2b). On the up-side, looking at the rock in the *Hand* picture we can see that the WFFT blur map is highly detailed and matches accurately the blur extent. On the down-side, this method suffers from its normalization process. Hence, by normalizing with the block magnitude, the focus metric is much sensitive in dark areas (i.e. low magnitude). This is visible also in the top right corner of the *Hand* picture where the blur amount is underestimated.

The performance on the *UnfocusedRaindrops* picture is rather good and especially much better than with HWT. The most important fact is that all drops correspond to fall off in the blur map (*this is better seen on a computer*). This is better noticed on drops B,D,E,F,G. There is a number of ways to extract these local minima using morphological operators for instance.

### 2.3.2d Future experiments

From the above experiments, one can conclude that none of the blur assessment approaches allows a fine enough estimation to clearly distinguish the drops from the rest of the scene. However, the two last experiments – namely, the slope of the amplitude spectra and the Weighted FFT – are promising. As an example of our latests experiments, we found that using a temporal averaging and weighted rank filter to locate the blur maxima (i.e. the drops) provides intermediate results when coupled with weighted FFT. Future experiments shall focus on post-processing filter to refine the blur map adjustment.

## 2.4 Conclusion

Detecting unfocused raindrops is a challenging task that can not be solved with the existing methods designed to detect focused raindrops. Clearly, this chapter describes only preliminary experiments and the analyses could be greatly improved through deeper investigation. However, our work in the matter has substantially narrowed down the problem of detecting unfocused raindrops by investigating a number of doomed leads.

In the first part, the method based on the photometry and lack of gradients was published as a first attempt to detect unfocused raindrops [Nashashibi et al., 2010]. Though this method was working on our databases at the time, our expertise shows that it shall fail when the scenes is highly dynamic.

In the second part, we have reviewed and experimented various methods for blind blur detection. Though, we have attempted to apply blur detection to the unfocused raindrops detection our experiments should be seen as general blur detection methods for computer vision. Hence, we have provided results on pictures recognized for blur detection. The blur assessment methods are based on the literature but we have made some improvements to enhance their performance (as for HWT and Weighted FFT). On one hand, the results we presented on photographs are convincing and show that accurate blur map can be computed without high-level semantic. On the other hand, when applied to unfocused raindrops the results are not yet convincing. Nevertheless, the preliminary results of the Weighted FFT and the slope of the amplitude spectrum seem encouraging. Deeper investigation will be required to quantify their performance.

Other experiments that we did not detail here – due to their poor performance – include chromacity, derivatives, texture, etc. Additionally, a large work was conducted on the estimation of the Point Spread Function (PSF) from the Cepstrum analysis. As a brief summary, we found that Cepstrum analysis can lead to good result but only if a large portion of the picture is covered with rain. Indeed, to estimate the PSF we need to apply several spatial and temporal averaging which requires a large amount of *blurred* data. We have omitted to present Cepstrum experiment since it requires a different database (with heavy rain) and thus can not be compared to the three others.

Finally, let us inform the reader that a team from the LIVIC research laboratory<sup>7</sup> is also working on the detection of unfocused raindrops within the scope of the ICADAC project. The method they are working on – not yet published – is based on a morphological watershed with initial markers that are chosen from the gradients of the smoothed image (reinforced by geometrical rules and a temporal smoothing filter). Their evaluation on 70 images of the same database we use, exhibits an acceptable recall of 67.64% given the challenging task. However, they have in average 7.42 false positives *per image* which is too high for now to be used. We shall compare our results once we will have run quantitative evaluation.

---

<sup>7</sup><http://www.inrets.fr/en/linstitut/unites-de-recherche-unites-de-service/livic>



## Chapitre

# Illumination Nocturne au Travers des Gouttes de Pluie

*Below is a French summary of the following chapter “Illumination Through Rain at Night”.*

**Remerciements.** Le projet présenté dans ce chapitre a été effectué lors d’un séjour au Robotics Institute, de Carnegie Mellon University, PA, USA sous la direction de Srinivas Narasimhan.

De nuit, la lumière des phares est réfléchiée par les gouttes de pluie produisant de hautes fréquences lumineuses qui induisent à la fois une perte de visibilité de la scène et une fatigue pour le conducteur. Pour palier à cet effet indésirable, nous proposons un nouveau système d’éclairage capable d’illuminer une scène de nuit, sans en éclairer les gouttes de pluie. Notre proposition atténuerait l’apparence de la pluie *dans la scène* ce qui améliorerait la visibilité pour le conducteur mais aussi pour tous les algorithmes de vision embarqués. L’approche que nous proposons (décrite dans la section 3.4) utilise une paire caméra-projecteur colocalisée pour détecter, localiser et prédire la position des particules afin de ne désactiver que les rayons lumineux qui intersectent ces dernières. Des expérimentations en conditions réelles et contrôlées nous ont permis de valider notre approche.

Dans un premier temps nous avons conçu un simulateur haute fidélité qui reproduit le comportement d’un tel éclairage dynamique dans des conditions climatiques variées (cf. section 3.5.2). Avec celui-ci, nous avons pu estimer les paramètres logiciels et matériels nécessaires à la conception de notre système.

Fort de cette analyse théorique, un premier prototype *bas coût* a été construit et démontre la faisabilité de notre système (cf. section 3.6). En environnement contrôlé ce prototype montre non seulement une diminution de l’apparence de la pluie, mais aussi une nette amélioration de la visibilité générale de la scène grâce notre éclairage dynamique.



## Chapter 3

# Illumination through Rain at Night



Figure 3.1: At night, illuminated rain appears as a bright flickering (distracting) pattern that reduces driver visibility.

**Acknowledgement.** The following research project was conducted during a PhD visiting scholar in the ILIM team of the Robotics Institute at Carnegie Mellon University, PA, USA under the supervision of associate professor Srinivas Narasimhan.

### 3.1 Introduction

The most complex property of a raindrop is certainly its ability to exhibit varying appearances. Because of their lens-like optics there are times when raindrops are hardly visible and one might look for dark background to assess rain and there are times when they appear as the brightest elements in the scene. When driving at night in rain for instance, the light of the front headlamps is refracted by the raindrops back to the driver eye making them look like bright flickering rain streaks as in figure 3.1. These high frequency patterns are in addition to the splashes and to the blurry windshield responsible of the degradation of driver's visibility [Hautière et al., 2010].

Suppose you are driving an intelligent vehicle during a rainy night: rain will be detected and your wipers will automatically be switched on, the breaks will adapt to wet conditions and speed regulation will adjust to agree with the speed limitations in rainy conditions. Now if you have an in-vehicle vision sensor the pixels affected by rain may be detected, digitally enhanced and then displayed on a unit display. However, it would require splitting your attention between the road and the display. Actually, none of the existing ADAS will improve your *scene visibility*. More than that your own headlights are responsible of the visual artefacts produced by the raindrops.



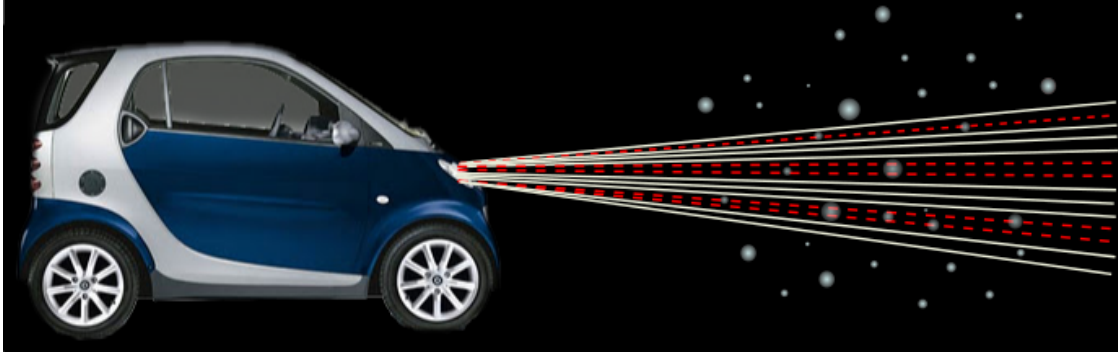


Figure 3.2: Our idea of illumination through rain. We propose a reactive illumination system that deactivates only those light rays that intersect particles (here, red dashed rays) such as raindrops, snowflakes, and hailstones diminishing the visual effect of falling precipitation.

Following this observation, we propose an approach that directly removes the appearance of rain from the scene through fast reactive control. Though making a tangible object invisible is impossible, recall that raindrops are visible because they refract the light. Not shining the raindrops would make them invisible or at least *less* visible. This could be achieved by selectively illuminating between and around the raindrops while avoiding direct lighting of any particles. The generally idea is depicted in figure 3.2, where the blue car illuminates the scene though no drops are shined. Noteworthy, the same principle might be applied for the illumination through any type of precipitation (i.e. rain, snow, hail). As for the fog lights, such adaptive lighting should enhance visibility for both ego driver and the other drivers.

A particulate weather avoidance system would need to first identify where the particles (i.e. raindrops, snowflakes, hailstones) are located and then reactively deactivate only the rays of light that would intersect the particles, as in figure 3.2. There are significant hurdles to realize a system capable of not shining element in the order of millimetres falling of several thousands times their own size each second (refer to section 1.2.1b for physics). First, the system must response fast enough to avoid precipitation particles and second it must maximize the *light throughput* of the headlights to avoid losing too much light. This requires both a low-latency high-speed system as well as accurate tracking and prediction algorithms.

In this chapter we describe the spirit of our idea and verify that such type of adaptive headlights could effectively improve the visibility for the drivers. After a very brief description of the related work, we investigate the effect of head lights in drops for the ego driver and the drivers of the oncoming cars (section 3.3). Also, to sustain the idea that such adaptive illumination is feasible we discuss the result of several night-time vision experiments of both artificial and real rain and investigate the software and hardware using a comprehensive simulator we have designed. *In fine*, we describe a first off-the-shelf prototype built in the laboratory using a high speed camera and projector that has been tested in controlled settings using rain generated from sprinklers.

## 3.2 Related work

As far as we know the proposed approach is unprecedented and only few related works can be gathered. Still, this project was motivated by the recent progress in automotive applications and augmented reality.

**Automotive** Headlights for inclement conditions are limited to foggy conditions but adaptive headlights which involved anti-blinding systems were developed in the mid-twenty century [Babinow, 1953]. Since then, major improvements have been made to use finer lighting resolution [Beam, 2000] and to solve the problem of mutual adaptive lighting [Hull and Gallegos, 2007]. Technically speaking the process in charge of the adaptive lighting is a simple threshold or histogram process to detect glare or highly contrasted scene and does not require fast reactive response or highly accurate illumination.

**Augmented reality** The closest existing works in spirit to our idea of illumination through precipitation probably belong to the field of augmented reality. In this field light is often used as a way to convey visual information in real time. Shader lamps for example require as well real-time interpretation of a scene to project textures on real objects that are either stationary [Raskar et al., 2001] or movable [Bandyopadhyay et al., 2001]. This is achieved using a process unit that analyse the pictures from the camera and then defines the light pattern to project. This often involves multiple cameras and projectors but overlaying real-time enhanced interpretation of a scene has also been achieved using a single projector [Bimber and Iwai, 2008, Grossberg et al., 2004]. Even sharing the same optical path as in [Wang et al., 2010] but again with long latency (50-70ms) – far too long for our application.

**Water displays** Eitoku et al. have also proposed to synchronize drops emitter and the projector so as to project a picture on the drops from below [Eitoku et al., 2006]. The picture is projected regardless of the positions of the drops making part of the background shined at the same time. Finally, work conducted in the same team as the present project has proposed a projector capable of selectively illuminating drops from multiple layers of artificial rain in a controlled environment [Barnum et al., 2009, Barnum et al., 2010b]. The projection on the hydrometeors is made possible by detection, tracking and prediction of the latter which is conceptually the exact inverse of what we are trying to do. To overcome the difficult challenge of localizing the drops Barnum *et al.* used a pair of camera and rigorous *a priori* about raindrops position and speed. The raindrops are generated using controllable water drop emitter manifold generating about 60 drops per seconds and detected as they penetrate the camera field of view using either a green laser plane [Barnum et al., 2009] or background subtraction [Barnum et al., 2010b]. They achieved tracking (and prediction) through a fitting-model approach where the trajectory model results of few seconds of observation. Except from the artificial drops generator we used no algorithms or hardware from that prior research.

It is interesting to highlight that conversely to all the existing applications we aim at selectively *removing* visual information by selectively choosing the part of the scene we *don't* want to illuminate.

### 3.3 Precipitation at night

#### 3.3.1 Light scattering

The effects of one's headlights involve back scattering for the ego driver and forward scattering for the drivers of the oncoming cars. Optics and physics tell us that the scattering effect is anisotropic [Macke and Groß klaus, 1998] and increasing with precipitation density/intensity.

Back scattering is the effect of the light being refracted in the opposite direction (i.e. scattering angle  $> 90^\circ$ ) and is responsible of the bright appearance of precipitation at night for the ego driver. For raindrops, this backscattering property is known to be monotonically *decreasing* with drops radius below  $3.5mm$  radius [Ross, 2000, p.45] and increasing beyond  $3.5mm$  – meaning that larger percent of the incoming light is backscattered for a  $2mm$  drop than for a drop of  $3mm$ . However, the total *amount* of light being scattered in the opposite direction depends on the cross-sectional area of raindrops, much larger for large drops making large drops backscatter more light than small one [Nousiainen, 2000]. This is in adequacy with our observations that the above described bright flickering effect is more important for large drops (i.e. heavy downpours) and function of the driver angle with the headlights beams. The angle-dependency property has been investigated in [Macke and Groß klaus, 1998, Nousiainen, 2000] and found critical especially for angles  $110 - 130^\circ$  and above  $170^\circ$ .

Forward scattering is the effect of the light being refracted toward the other drivers and further affects their visibility (i.e. oncoming cars) by producing blinding glare effect surrounding light sources (i.e. headlamps). Glare is especially visible in rain or hail and directly related to the density of particles, as shown in figure 3.1. It has been reported to affect especially older drivers [Davidse, 2006]. Because of the high reflectance property of snowflakes forward scattering is negligible and no glare effect is visible.

#### 3.3.1a Operating range required

Now that we have shown the backscattering effect of headlights, we discuss how far from the light sources are the particles visible at night. Hitherto, we have formulated the idea that illumination through precipitation requires turning off the light rays that intersect particles. Since the area covered by a light ray increases as the distance from the projector increases, every single ray will eventually intersect a particle and thus the operating range must be restricted.

To investigate the required depth of interest we captured images of  $4mm$  diameter water drops (released from pipettes) illuminated by a halogen lamp (Lowel Pro-light which outputs  $\approx 4,000lux$ ) at different distance and imaged the latter with a high speed camera (Photron 1024 PCI, resolution  $1024 \times 1024$ , exposure time  $4ms$ ). Results of these experiments are shown in figure 3.3 with pictures as insets. Experiments exhibit that raindrops are no more visible farther than  $3m$  which was found to be related to the light fall off rather than to the camera resolution.

This reinforces the idea that only close-by raindrops produce flickering streaks. The result was confirmed by informal observations to the naked eye for both artificial and real rain but an exhaustive user study would be *required* to assess their validity. Note that figure 3.3b contains also drop-off of a 2006 Honda Civic headlight and a ViewSonic PJD6251 DLP projector we used for the prototype (color wheel removed and illuminated through a 50% beamsplitter). Finally, we have conducted similar experiments in real rain and outdoor environment but could not assess the visibility distance as it would have required a complex setup (long overhang for instance).

We now discuss the impact of the bright flickering streaks caused by falling precipitation.

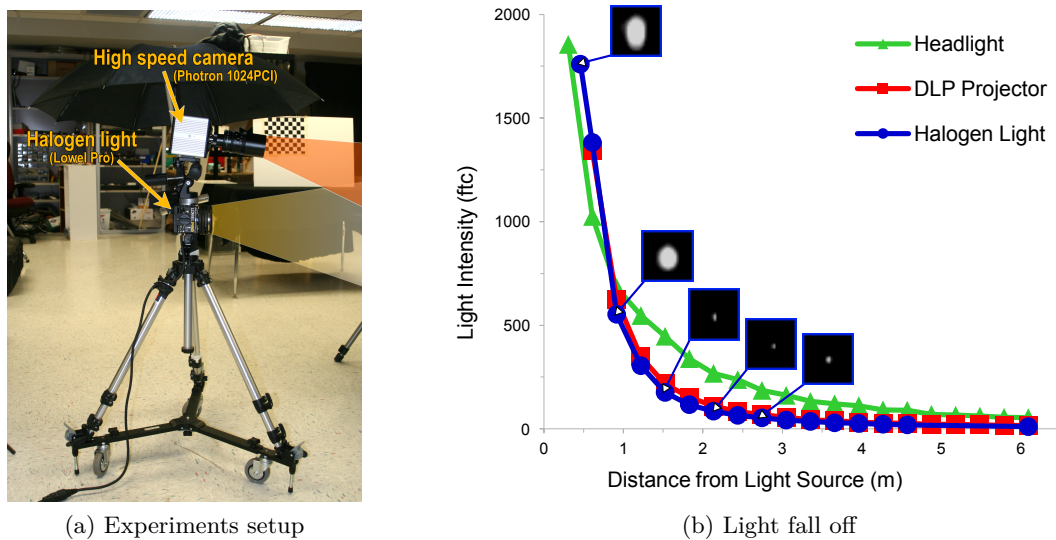


Figure 3.3: Laboratory experiments set-up (a) to study raindrops visibility.  $4mm$  water drops were illuminated by a halogen light and imaged for comparison at different distances. (b) Experiments show that water drops are not visible farther than  $3m$  with an exposure time of  $4ms$ . Also shown is the light fall-off for a Honda Civic (2006) factory headlight (late night with no street lamps) and Viewsonic PJD6251 DLP projector (color wheel removed and illuminated through a 50% beamsplitter).

### 3.3.2 Psychophysical studies

Lately a large number of researches focused on identifying the causes of driver fatigue or boredom (a good review being [Lal and Craig, 2001]) and all have highlighted night-time driving as the source of important fatigue, this of course increases the risk of having an accident. Though we found no literature that specifically study the perception of rain at night while driving, several psychophysical researches have been published on the impact of repetitive patterns for the human visual systems.

Sekuler and Pantle have for example shown that individuals being exposed at least 2 minutes to highly contrasted visual artefacts having the same orientation were less sensitive to visual stimuli having similar orientation [Sekuler and Ganz, 1963, Pantle and Sekuler, 1969]. This phenomenon is termed *direction-specific adaptation* and results of the threshold accommodation of the orientation-specific cells in the human visual system. Noteworthy, the same authors have also shown that this effect is function of the level of contrast which implies that reducing the contrast of the visual stimuli lowers the effect of accommodation. Other effects of repetitive patterns have been investigated later. An other study conducted in [Schor and Tyler, 1981] advocates that vertical stimuli involve a significant reduction of the clear vision area (Panum's area).

From the above cited studies, it is logical to assume that the drivers field of view is reduced in rain and its vision altered by the presence of contrasted bright flickering streaks. Having this said, a pessimistic interpretation - that would require a deeper understanding of driver perception - would say that the visibility of the white reflective lane marks is further diminished because of the similar orientation with the bright streaks<sup>1</sup>.

<sup>1</sup>At high speed falling particles have similar visual orientation due to perspective. This was assessed using the

### 3.3.3 Observations

There are three main observations that we can draw from the above studies:

- First, both night-time experiments in real rain and laboratory experiments that we have conducted have shown that due to light fall-off *it is not necessary to consider drops beyond a relatively short distance ( $\approx 3m$ )*.
- Second, *smallest drops can safely be ignored* as they do not reflect a significant amount of light [Ross, 2000, Nousiainen, 2000].
- Third, the psychophysical studies have shown that *decreasing the contrast of the flickering particles reduces the visual discomfort* for the driver [Sekuler and Ganz, 1963, Pantle and Sekuler, 1969, Schor and Tyler, 1981]. That is to say that a driver would benefit of such system even if it exhibits performance well below perfect accuracy.

In the light of these results, if we avoid shining even a small amount of the large close-by particles while loosing 5 to 10 percent of the light this should still improve the driver’s visibility.

## 3.4 Overall approach

### 3.4.1 Mitigation strategies

We have discussed the need to detect, track and predict the position of the close-by particles as a prerequisite to be able to selectively illuminate a scene but the particles. The major issue of such needs is the interdependence of the tasks. For example, predicting the next particle location requires to track the latter long enough to guess the trajectory model but tracking also requires several observations (detection) of the same particle. Tracking and prediction may be narrowed down if we admit that the particles are following a straight line trajectory during the time they remain in the field of view (less than  $200ms$  for raindrops with our setup, see figure 3.9). This is not the case for detection that requires a deeper analysis.

**Ranging or Imaging** Traditionally, detection tasks in computer vision use either ranging or imaging sensors but for the very specific case of particulate weathers the choice is rather limited.

Ranging sensor (such as LIDAR) response to rain has been studied by Macke and Großklaus who conclude that raindrops can’t be accurately sensed with such device. This result in fact of the random orientation of raindrops and non-monotonic behaviour of the scattering effect [Macke and Groß klaus, 1998] making impossible their precise localization.

Alternately the use of imaging sensors raises the problem of imaging particles though our primary goal is to *not* shine them. There are different strategies to tackle this issue. First, the particles could be imaged with an infra-red sensor while still being invisible to the naked human eye but this would require an additional infra-red spot light. Second, very fast lighting may be used to shine the particles short enough so the light would not be visible to the human eye. This has already been used for traffic light recognition for instance [Premachandra et al., 2009] and is better known as “visible light communication”. Both of these solutions are valid and present different pros and cons. A third option - which we have chosen-, is to shine the particles only for a short period of time and to stop shining the latter once their tracking is robust enough to predict their position for the time they remain in FOV of the projector.

---

simulator detailed in section 3.5.2.

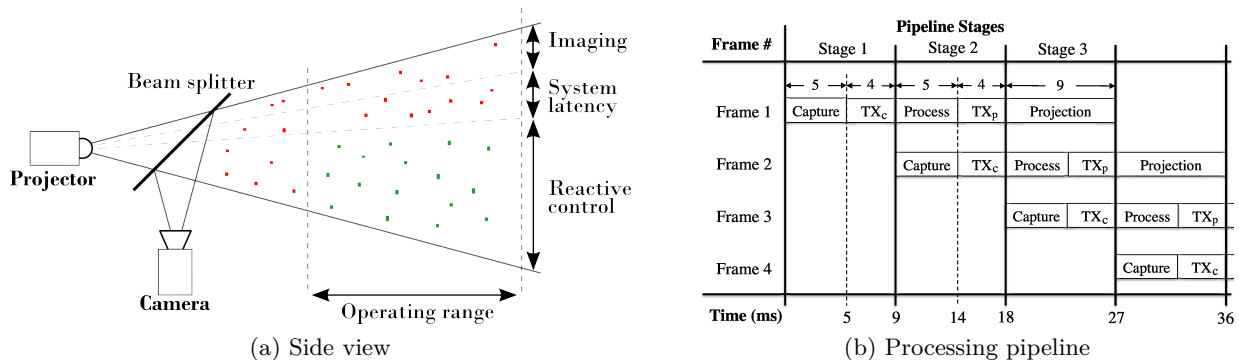


Figure 3.4: Overall approach for illumination through precipitation. (a) Sketch of the setup used. A co-located camera-projector system images and illuminates a volume of precipitation. All particles are first detected by illuminating them in a very short period of time, their future locations are predicted, and then the rays intersecting them are reactively turned off. The time duration between image capture and reactive illumination is the latency of the system. The operating range is approximately  $3m \times 3m \times 4m$ . (b) Pipeline stages of our system with execution times (in  $ms$ ).  $TX_c$  and  $TX_p$  denote data transfer between camera and computer and between computer and projector, respectively. Process refers to drop detection and prediction, and generation of the projection image.

**3D localization** 3D localization is often said as a major issue and a huge number of writings have proposed various way to tackle this challenge using additional sensors or stereo imaging (spatial or temporal) to name just a few in the long list. For water drop display, precise knowledge about the drop position is of high importance as the particles we are trying to project on are in the order of millimetres leaving us with very small error tolerance for the projection of the light pattern. To overcome the problem of drops localization Barnum et al. have used either a stereo imaging system [Barnum, 2011] or a single sensor but in addition to a calibrated laser beam hitting the drops as they enter camera FOV [Barnum et al., 2009].

None of these approaches meet our requirements and to avoid the geometrical mapping inaccuracies we have used a collocated approach. Indeed there is no need for us to know the particle position but to deactivate the right light rays. Hence if the imaging and the lighting system share the same optical path, the 3D position of the drops is useless as their 2D position is identical in both coordinate spaces.

### 3.4.2 Our collocated imaging-lighting approach

The main components of our co-located imaging and illumination system are shown in figure 3.4. At the top of the camera’s Field Of View (FOV), particles (raindrops, snowflakes, and hailstones) are illuminated by a projector and observed within a short period of time (typically, a few milliseconds). This region of the image is used to predict the trajectory of particles across following frames. In the tracking region, rays of light are prevented from illuminating particles by projecting around predicted particle locations. Additionally, any wrongly illuminated drops can be detected by the camera and used to update the predicted motion paths. The use of a co-located camera and projector simplifies tracking since the positions of particles only need to be determined in a 2D image instead of a full 3D coordinate space.

We define *system latency* to be the time required to transfer an image, process the data, and reactively control an illumination source. A system that suffers from long latencies will perform poorly for two reasons: (1) The particle may have already left the imaging/illumination FOV before the next control cycle, and (2) Motion effects like wind and camera vibration could increase detection and tracking error. In order to increase system responsiveness, many of the tasks can be executed in parallel, which improves processing throughput.

The timing diagram of our three-stage processing pipeline is shown in figure 3.4b with times measured from our prototype system (described in section 3.6). Capture refers to camera integration and  $TX_c$  refers to image data transfer to the computer. Process refers to drop detection and prediction, and generation of the projection image.  $TX_p$  denotes the computer-projector transfer time and projection is the refresh time of the projector. The timing values show that the typical execution time of each component and the pipeline is staged based on the overall critical path (in this case stages 1 and 2 are almost identical).

We evaluate the performance of our system using two competing metrics:

- **Light throughput** is a measure of the amount of light radiating from the light source. This is computed as the percentage of projector pixels that remain on per each frame.
- **Accuracy** is the percentage of particles in the system FOV that are *not* being illuminated.

Thus, with a headlight that is constantly *on*, light throughput is 100% and accuracy is 0% and when the headlight is *off* the light throughput is 0% while the accuracy is 100%. Given these two competing metrics, our goal is to maintain high light throughput while maximizing accuracy.

Before implementing our system, we designed a simulation environment to explore the following design-space questions: What intensities of precipitation (drizzle, widespread, shower, etc.) can the system handle? How much light does the system lose? How far from the light source or viewer should the particles be considered? How will the speed of the vehicle affect performance? How fast should the reactive control be? What is the trade-off between system latency and complexity of the prediction algorithm?

The simulations are described next.

## 3.5 Feasibility study using simulations

### 3.5.1 Overview

None of the particulate weathers we are dealing with (i.e. rain, snow, hail) are predictable in such a way that we could ensure having the exact same conditions for our tests, so we used simulation to investigate the feasibility of our project. An other advantage of using simulation is the ability of testing over a wide variety of weather conditions. An exhaustive research has been required to lay down the groundwork of the simulator as well as to avoid some of the common mistakes<sup>2</sup>.

The discrete-time simulator we have developed can reproduce the statistical behaviour of any particulate weather from the vast body of physics models in literatures emulating the sizes, densities and dynamics of their particles. We generate Particle Size Distribution (PSD) using [Marshall and Palmer, 1948, Gunn and Marshall, 1958, Cheng and English, 1983], and velocities/dynamics from [Van Boxel, 1997, Langleben, 1954, Dennis et al., 1971]. Note that the choice of either PSDs

<sup>2</sup>Physic and meteorological models are often misused as discussed in section 1.2.1.

may alter the performance so we chose only distributions obtained from measurements in Canada (Alberta, Ottawa and Montreal) which is fairly close to the birthplace of this project (Pittsburgh PA, USA).

Additionally the simulator reproduces all pieces of hardware and software involved for adaptive lighting. A virtual camera and a projector are used to model the impact of additional system parameters including camera exposure time, camera optics, camera resolution, processing accuracy and delays, projection resolution, projection speed and optics. Figure 3.5 shows the main components used in the simulation. The virtual components are made asynchronous so as to accurately capture the effect of timing dependencies (i.e. software & hardware delays).

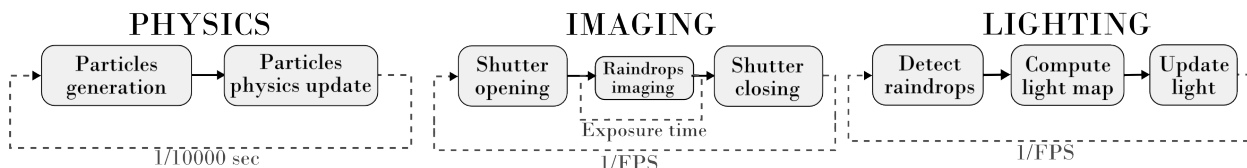


Figure 3.5: The three main components used in our simulator.

### 3.5.2 Particles system simulator

Density and number of particles are related to the precipitation rate  $R$  which is the amount of water that falls on the ground over an hour. If  $S_w \times S_h$  is the projection of the particles emitter on the ground, the amount of water collected in one hour is  $S_w \times S_h \times R$  liters (with  $R$  in mm/hr and  $S_w$  and  $S_h$  in meters). If the particles system is updated with a frequency of  $f_\sigma$  the amount of water  $W$  to fall is computed after each small time duration:

$$W(t + \delta t) = W(t) + \frac{S_w \times S_h \times R}{3600 f_\sigma} \quad (3.1)$$

For real precipitation, the particle size distributions (PSD) are modeled using gamma functions (cf. section 3.7.2) scaled by the precipitation rate  $R$ . In other words, particles of large size are more likely to occur in heavy precipitation. In our simulation, given the weather parameters we compute the probability function modeling the corresponding PSD and use it to determinate the size of each upcoming particle.

Each time a particle is generated, its *water equivalent* is removed from the remaining water and the same process is repeated as long as there is enough water. For raindrops, the *water equivalent* is the volume of the meteor but for snowflakes or hailstones it is the result of the melting process assuming a water density of 30% and 80%, respectively. For example,  $R = 14\text{mm/hr}$  water equivalent of snow in our simulator corresponds to  $14 * 24 * 30\% = 1\text{m}$  of accumulated snow in one day.

The sizes, PSDs, and velocity models used in the simulator are listed in table 3.1. We will discuss the physics we have chosen in section 3.7.2. In most natural scenarios, it is sufficient to model the dynamics of particles using a constant terminal velocity, denoted  $v_t(d)$ . If particles are artificially generated (say, using sprinklers as for our prototype), acceleration must also be taken into account and for that we will use the model from [Van Boxel, 1997]. Unless specified otherwise for now on we consider the particles to have reached their terminal velocity (described in section 1.2.1b).

We operate our simulator at a frequency of  $f_\sigma = 10,000\text{Hz}$  providing a  $0.1\text{ms}$  time granularity.



	Diameter	Particle Size Distribution $N(d) = N_0 \times d^\mu \times \exp(-\Lambda d)$	Velocity
<b>Rain</b>	0.1 - 10mm	$N_0 = 8000$ , $\mu = 0$ , $\Lambda = 4.1R^{-0.21}$ (from [Marshall and Palmer, 1948])	$a(t, d) = Fg(d) + Fd(t, d)$ with $Fg(d) = \frac{g\rho_w\pi d^3}{6}$ , $Fd(t, d) = 3\pi d\mu_a v(t)C_t C_d$ $C_t = 1 + 0.16Re^{2/3}$ , $C_d = 1 + 0.013(We + 2.28)^{2.12}$ (from [Van Boxel, 1997])  $v_t(d) = 9.65 - 10.3 \times \exp(-600d)$ (from [Atlas et al., 1973])
<b>Snow</b>	1 - 10mm	$N_0 = 3800 \times R^{-0.87}$ , $\mu = 0$ , $\Lambda = 25.5 \times R^{-0.48}$ (from [Gunn and Marshall, 1958])	$v_t(d) = 2. \times (10d)^{0.31}$ (from [Langleben, 1954])
<b>Hail</b>	1 - 50mm	$N_0 = 810$ , $\mu = 0$ , $\Lambda = 0.33$ (from [Cheng and English, 1983])	$v_t(d) = \sqrt{\frac{4 \times 10^3 \times g \times \rho_i \times d}{3 \times \rho_a \times 0.6}}$ (from [Dennis et al., 1971])

Table 3.1: Listing of the particles size, Particles Size Distribution (PSD) and velocity models used to simulate rain, snow, or hail.  $R$  being the intensity or fall rate measured in  $mm/hr$ .  $N(d)$  being the number of particles of diameter  $d$  ( $mm$ ) in a  $1m^3$  volume. Terminal velocity  $v_t$  ( $m/sec$ ) is sufficient to simulate real rain, snow or hail but to study the effect of initial velocity for raindrops we used the additional numerical model [Van Boxel, 1997] and compute the acceleration  $a(t, d)$  ( $m/sec$ ) as a function of time  $t$  and drop diameter  $d$  through sum of the gravitational force  $Fg$  and the drag force  $Fd$ .  $v(t)$  being the instantaneous velocity of the particle,  $g$  the gravitation force ( $m/sec$ ),  $\rho_a$ ,  $\rho_w$  and  $\rho_i$  the density of air, water and ice ( $kg/m^3$ ),  $\mu_a$  the viscosity of air ( $Pa \cdot s$ ),  $Re$  and  $We$  respectively the Reynold and Weber number of the particle (*dimensionless*).

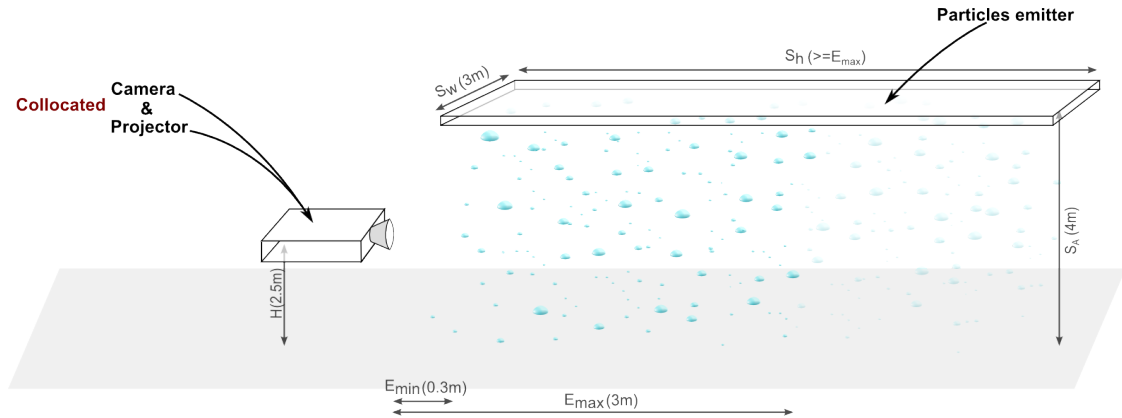


Figure 3.6: The set-up used in our simulator. Note that the camera and the projector are coaxial which avoid any geometrical mapping. The virtual camera visibility response is mapped with results of real experiments ( $E_{min} = 0.3m$ ,  $E_{max} = 3m$ ). Logically, the size of the particles emitter is such that  $S_h \geq E_{max}$  while  $S_w$  and  $S_A$  are set to  $3m$  and  $4m$  respectively for the particles to cover the entire FOV of the camera.

### 3.5.3 Operating range

Studying drops visibility and light fall-off have shown us that particles are no more visible beyond a relatively short distance we denote  $E_{max}$ , as proved in section 3.3.1a. To ensure high fidelity simulation, we mapped the response of the virtual camera to the one measured. Hence,  $E_{max} = 3m$ . A minimum distance  $E_{min}$  is also set to avoid particles too close to the camera from occluding the scene. This may also be seen as the distance between the camera and the protecting glass that will eventually be mounted for actual headlights. We arbitrarily set  $E_{min} = 0.3m$  as a good trade-off. If stationary the depth of the particles emitter  $S_h$  is set to  $E_{max}$  but in case of a moving camera we increase  $S_h$  to simulate a continuum of rain. Figure 3.6 depicts the virtual set-up used in the simulation.

As discussed the light pattern is computed according to the particles imaged and detected with the camera. Thus each pixel covered by a particle in the camera picture corresponds to a projector pixel being turned off (again, recall that the camera and projector are collocated). Garg and Nayar have discussed the relationship between the integration time of the camera and the velocity of the particles as being directly responsible for the area covered [Garg and Nayar, 2005]. Figure 3.7 shows rain and snow captured with a  $30ms$  and  $1ms$  exposure time. For each type of precipitation, longer exposure time creates image streaks (similar to what a human perceives) and reduces the light throughput. We'll investigate the acceptable exposure time in the following section.

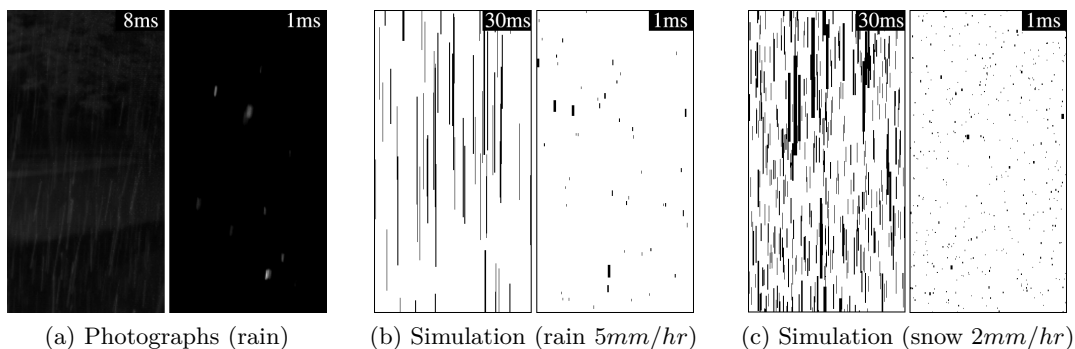


Figure 3.7: Pixel area covered by particles in (a) outdoor rain experiments (*unknown intensity*) and with simulations of (b) moderate rain ( $5mm/hr$ ) or (c) moderate snow ( $2mm/hr$ ) imaged with  $30ms$  and  $1ms$  exposure time. A longer exposure lowers light throughput. Thus, the camera exposure time should be no more than a few milliseconds to ensure meaningful performance.

### 3.5.4 Performance analysis for a stationary system

We first consider a theoretical system where image capture, processing, and projection are instantaneous and tracking is perfect. The accuracy is 100% and the light throughput can be used as an upper bound for what is possible in reality.

Figure 3.8 shows the light throughput achieved for a variety of weather conditions including mild, moderate, and heavy rain, snow and hail. Notice that in the case of zero exposure time, the light throughput is above **99%** for a shower ( $17mm/hr$ ) and hailstorm ( $10mm/hr$  equivalent quantity of water), and above **98%** for heavy snow ( $10mm/hr$  equivalent quantity of water). These

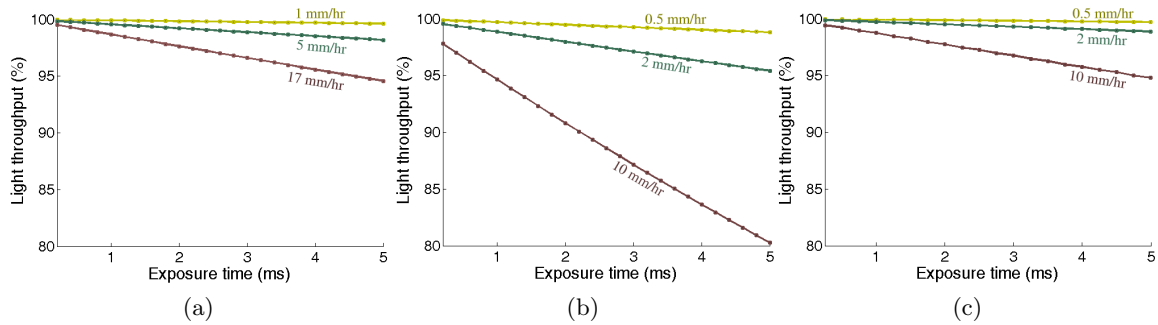


Figure 3.8: Ideal light throughput of our system for rain (a), snow (b) and hail (c) versus exposure time of the camera. Three precipitation rates are reported (light, moderate, and heavy). Particles falling slower stay longer in the field of view and thus require shorter exposure time. To achieve light throughput over 90% the exposure must be set to 2ms or less to accommodate the various precipitation types and intensities. For just rain and hail, 4ms is short enough.

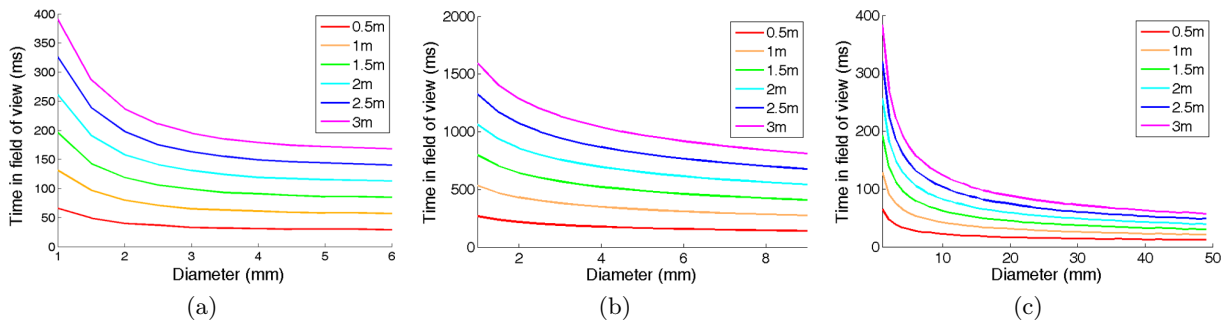


Figure 3.9: Duration particles of varying diameter stay in the field of view with our camera settings for rain (a), snow (b), and hail (c) given various distances from the camera. Avoiding illumination of raindrops or hailstones requires significantly faster response time since they fall faster than snowflakes.

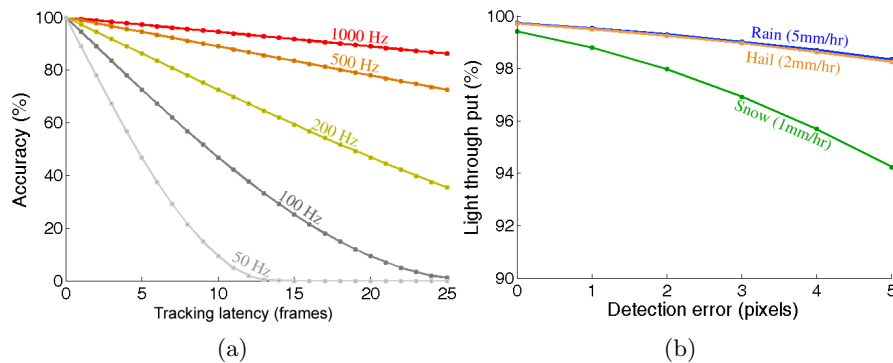


Figure 3.10: (a) System accuracy versus tracking latency (measured in frames) for various update frequencies. Tracking may require several frames to initialize, lowering the accuracy of the system especially with a low update frequency. (b) Light throughput versus detection error for rain, snow, and hail. As expected, light throughput decreases when errors occur.

results show that reactive illumination in bad weather could theoretically provide very high light-throughput. We now investigate system performance with parameters based on current technology.

The impact of camera exposure time on light throughput is shown in figure 3.8. We see that to guarantee light throughput of 90% under most weather conditions, camera exposure time should be no longer than  $2ms$ . As a comparison, common computer-vision applications are running at 30FPS ( $\approx 30ms$ ) which would imply a throughput below 30% in heavy snow which is not acceptable.

The time period for which particles stay in the projector’s FOV also varies with different types of weather conditions. From figure 3.9, snowflakes tend to stay longer in the FOV than either rain or hail. This has two effects — the light throughput will be lower in snow, but at the same time better accuracy is achieved (less particles illuminated) since a higher system latency can be accommodated. We also investigate the impact of latency on tracking accuracy in our system. Assuming a raindrop diameter of  $3mm$  and a response time of  $10ms$ , the drop will have moved by  $80mm$  (26 times its own size) by the time the projector illuminates the scene.

Figure 3.10 shows that less than 6% of the raindrops will be illuminated if tracking initialization requires 10 frames and the system operates at  $1000Hz$ . In contrast, at  $100Hz$  more than 50% of the drops will be illuminated.

Additionally, we define the error as a margin surrounding the actual positions of the drops. This serves to model detection/tracking inaccuracies. Figure 3.10 shows that light throughput decreases with increasing error in drop detection. An error of 5 pixels will result in **95%** light throughput.

### 3.5.5 Performance analysis for a moving system

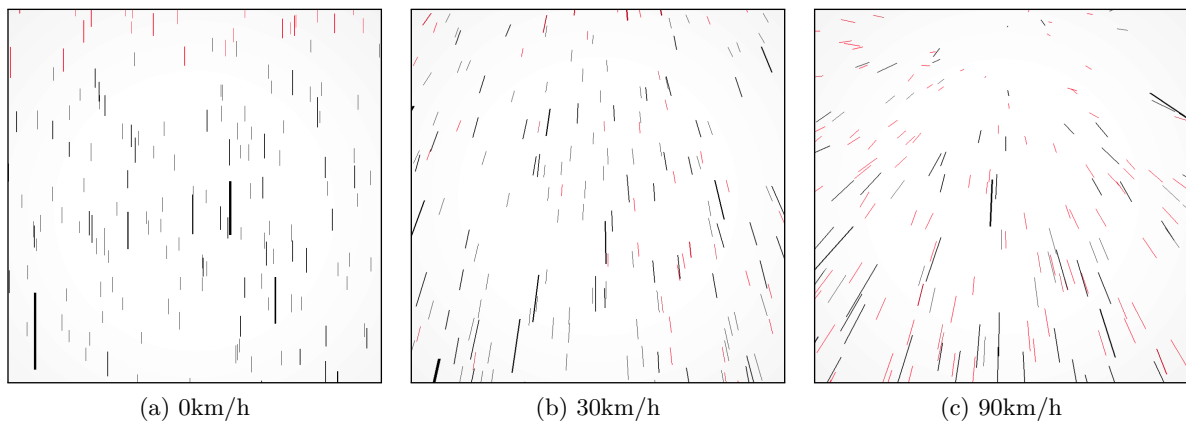


Figure 3.11: Simulation of medium rain ( $5mm/h$ ) with 3 frames latency given various vehicle velocities. Shown is the light output by the projector (black=pixel light *off*, all others=pixel light *on*) where red correspond to light rays that are incorrectly turn on. Note, than when driving at high speed the particles penetrate the projector FOV at various positions.

Hitherto, we assumed the system to be stationary inferring thus that the particles may penetrate the FOV only from the top. Obviously such assumption is violated if the car is moving as the falling particles may penetrate the frustum from the front, the sides or the top – making detection and tracking much more complex. Also, as the particles stay less time in the imaging/projector FOV the tracking latency affects dramatically the performance. Actually, tracking the particles on the edge of the system frustum may be simply impossible if the vehicle is moving too fast. In other words our system FOV virtually reduces with speed — just as for an individual who’s

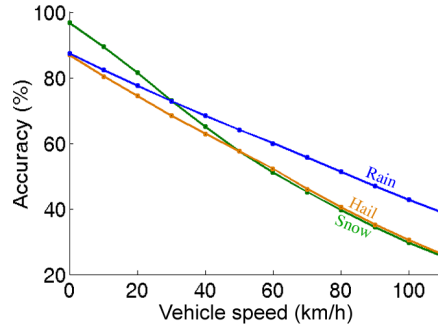


Figure 3.12: Accuracy versus vehicle speed in rain, snow, and hail. As the speed increases the accuracy severely drops but even at high speed our system successfully *not* illuminate some of the particles. Note that the accuracy is the percent of particles not illuminated which is not related to the density of particles (i.e. weather intensity). The accuracies displayed in this plot for rain, snow or hail stand for any weather intensities.

field of clear view (Panum’s area) vanishes. In figure 3.11 are shown outputs from our simulator of the identical rain event at  $0\text{km/h}$ ,  $30\text{km/h}$  and  $90\text{km/h}$ . In red are shown the particles being shined due to our simulated inaccuracy and tracking latency. As expected at the speed increases the particles shined are sparser with our system.

Now, to assess throughput and accuracy for different vehicle speeds under various precipitation rates simulations were performed with a moving vehicle. Data were collected using system parameters described in Section 3.5.3 at  $120\text{Hz}$  with 3 frames latency and 2 pixel detection/tracking error.

A continuum of particles is obtained by resizing the depth of the particles emitter ( $S_h$ ) according to the vehicle speed of the vehicle ( $v$ , in  $\text{m/sec}$ ) so the particle volume fill the whole camera FOV as the vehicle move forward. Thus:

$$S_h = E_{max} + \frac{10 \times v \times P_g(d_{min})}{36} \quad (3.2)$$

where  $P_g(d_{min})$  is the time (in sec) required for the particle of diameter  $d_{min}$  to hit the ground, and recall  $E_{max}$  is the maximum visibility distance (in m).

Figure 3.12 exhibits the influence of vehicle speed on accuracy in rain, snow and hail. Simulations with a moving system give the following accuracy (light throughput) in  $5\text{mm/hr}$  rain: Vehicle stationary **87.31%** (**94.82%**), vehicle moving at  $30\text{km/h}$  **72.83%** (**95.16%**), vehicle moving at  $100\text{km/h}$  **42.73%** (**96.46%**). Thus, the accuracy degrades with increasing speed but the light throughput maintains similar level.

Remarkably, the accuracy measures the percent of particles not being illuminated which means that it is consequently *not related* to the weather intensity (i.e. particles density) which we did not indicate in the figure. Hence, at  $60\text{km/hr}$  accuracy for any intensity of rain, snow, or hail is respectively, **59.98%**, **51.09%** and **52.20%**. The accuracy is however closely related to the duration the particles stay in the FOV which dramatically reduces if the vehicle is in motion.

Conversely, it is interesting to note that the throughput is of similar level whatever is the speed, which require a two-fold explanation — First, as the speed increases the accuracy drops which mean that less number of pixels need to be turned off. Second, as the speed increases the particles are imaged on more pixels due to perspective. These two factors balancing each other the light throughput varies slowly with speed. This is an interesting property of our approach as it

means that the adaptive lighting may remain switched on while driving in a drizzle or in a heavy storm without losing much of the light output.

Now, let us discuss snow in figure 3.12. In contradiction to the good performance with a stationary system, snow accuracy drops dramatically which is due to the dynamics of the flakes. As the latter fall slower than any other particles, they remain longer in the air and each time the vehicle moves forward a large number of flakes enter the FOV and will be illuminated before being tracked 3 frames later. However, notice that even if the accuracy drops with speed our system reduces precipitation visibility (i.e. accuracy  $> 0\%$ ) even at high speed. Assuming the extreme scenario of a vehicle moving at  $110km/hr$ , **38.74%** of the raindrops in headlight FOV will *not* be illuminated, **26.02%** of the hailstones, and **25.48%** of the snowflakes.

### 3.5.6 Recipes for complete systems

		Prototype	Realistic	Future	Ideal
Imaging	CCD resolution (pxl size)	$120 \times 244$ ( $7.4\mu m$ )	$1024 \times 1024$ ( $17\mu m$ )	$1024 \times 1024$ ( $17\mu m$ )	$1024 \times 1024$ ( $17\mu m$ )
	Focal	$4mm$	$33.5mm$	$33.5mm$	$33.5mm$
Lighting	Refresh rate	$111Hz$	$125Hz$	$400Hz$	$\infty$
	Resolution	$120 \times 244$	$1024 \times 1024$	$1024 \times 1024$	$1024 \times 1024$
Process	Tracking latency	1frame	3frames	1frame	0frame
	Detection error	2pixels	2pixels	2pixels	0pixel
Timings	Exposure time	$5ms$	$5ms$	$1ms$	$0.1ms$
	Transfer Cam>PC ( $TX_c$ )	$4ms$	$4ms$	$1.5ms$	$0ms$
	Process	$4ms$	$5ms$	$1ms$	$0ms$
	Transfer PC>Proj ( $TX_p$ )	$4ms$	$4ms$	$1.5ms$	$0ms$
System	<b>Response time</b> ( $TX_c + process + TX_p$ )	$12ms$	$13ms$	$4ms$	$0ms$
	<b>Frequency</b>	$111Hz$	$125Hz$	$400Hz$	$10,000Hz$

Table 3.2: Recipes for four complete systems to achieve illumination through particulate weathers. Parameters identical to all recipes are omitted. Delays for the prototype were measured once built and others result of estimation given bandwidth rate and computer capabilities. Note that the imaging resolution for the prototype is much lower to ensure an acceptable overall frequency.

The main parameters being investigated separately we aim at defining a possible configuration to build a working prototype given current technical limitation of both hardware and software.

We further defined four recipes which details are reported in table 3.2. The first three recipes reflect respectively, currently low cost available, close future available, and ideal system. We will refer to them as *Realistic*, *Future*, and *Ideal*. A fourth recipe *Prototype* is proposed as to match the hardware available at the time in our laboratory and will be detailed later.

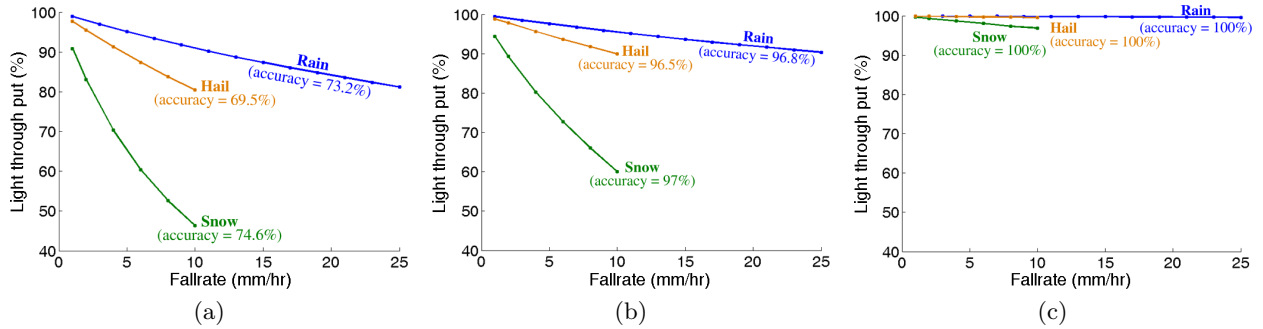


Figure 3.13: Performance of simulation for three systems at  $30\text{km/h}$  using today's low cost (a), future (b) and ideal (c) hardware and software technology for various precipitation rate. All of these systems achieve acceptable accuracy ( $> 69\%$ ). However, today's system (a) has  $13\text{ms}$  latency and thus can't handle extreme conditions or snow without losing too much light. System (b) with  $1.5\text{ms}$  latency gives us hope concerning the performance it could claim in a near future. Not only the accuracy is over  $96\%$  but the light throughput is very high. Finally, ideal system (c) is the upper-bound for what is possible in reality admitting an instantaneous and perfect system. Notice that snow exhibits higher accuracy since particles fall slower although the throughput is significantly lower. Refer to table 3.2 for complete settings.

In the first *realistic* system, camera and projector resolution is  $1024 \times 1024$  pixels, camera exposure is  $1\text{ms}$ , tracking latency of 3 frames, detection error of 2 pixels and with a camera motion of  $30\text{km/h}$ . Based on USB2 limitation we assume our system to have a response time of  $13\text{ms}$  (pipeline identical to figure 3.4b). The plot in figure 3.13(a) shows the results for different precipitation rates of rain, snow and hail when driving at  $30\text{km/h}$ . Notice that the accuracy of our system is quite high ( $> 85\%$ ) for rain/hail/snow as particles are illuminated for only  $25\text{ms}$ . However, light throughput suffers from the long latency of the system making such system meaningless in extreme conditions as it requires to switch off most of the rays. In mild or moderate rain/hail, the driver's visibility can still be improved significantly while losing maximum  $20\%$  of light.

Figure 3.13(b) shows the performance at  $30\text{km/h}$  of a more advanced system (*future*) with the same parameters as the previous system, but with a  $1\text{ms}$  exposure time and a  $1.5\text{ms}$  total system latency. This can be achieved by using a gigabit interface and camera with a better quality sensor, eliminating the data transfer latencies by constructing an embedded and integrated imaging and projection system with a faster DLP projector. Notice the significant improvement in light throughput and accuracy in all weather conditions. This gives hope that such a system may be realized in the near-future.

Finally, a last recipe simulates to the performance of a near-ideal system at  $30\text{km/h}$  that runs at  $10\text{kHz}$  (Fig. 3.13(c)). This presents the upper-bound of what would be feasible in a far future. From table 3.2, such system admit only a  $0.1\text{ms}$  latency. There are several reason of thinking that this may be achieved in some years. For example, experiments on imaging rain at high speed have shown us that raindrops detection could be processed cluster-wise. If so, the whole pipeline may be streamed to speed up the process using specialized hardware.

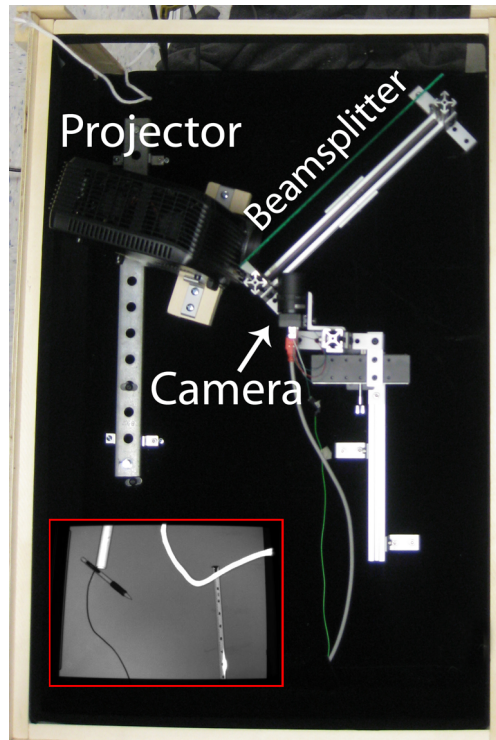


Figure 3.14: Co-located prototype. Camera and projector are made coaxial using a 50/50 beam splitter while we avoid the reflections on the glass by the use of a light trapper made of thick absorbing material. The red square inset is a view from the camera. Note that the lack of shadows assesses from the accurate co-location.

## 3.6 Proof-of-concept – Prototype

Simulations have shown us that good performance may be achieved while still maintaining high light throughput even including latencies and errors and even at high speed. Based on the previous simulations we have built a proof-of-concept prototype to assess the validity of our approach and the findings of simulations. The prototype systems we have built used off-the-shelf components and consists of a co-located camera and projector operating at  $120Hz$ . We use a similar set-up to the one used in [Wang et al., 2010] for context aware lighting, though our application requires both lower latencies (in contradiction to the  $50 - 70ms$  latency of Wang *et al.*) and higher refresh rate to avoid any flickering effect.

**Acknowledgements** Please note that both machining, set up and processing of this prototype were conducted by Robert Tamburo – project scientist at the robotics institute, CMU. Obviously there are number of engineering challenges that we do not detail here.

### 3.6.1 Hardware

The system we have built is shown in figure 3.14. It consists of a box with co-axial camera and projector and a standard desktop computer for the processing. We use a monochrome camera (Point Grey Flea3) with ethernet interface that is capable of capturing  $120FPS$  over a  $120 \times 244$



region of interest. The projector we used is a Viewsonic PJD6251 DLP, and has a native resolution of  $1024 \times 768$  (maximum vertical frequency of  $120Hz$ ). It outputs a brightness of 3700 ANSI lumens, which is the equivalent of the popular D2S HID headlight (approx. 3200 lumens) [Dong, 2008].

The optical axes of the camera and projector are co-located to avoid the computational overhead associated with stereo matching while providing finer granularity. This is accomplished by placing a 50/50 beam splitter in front of the camera and projector lenses as illustrated in figure 3.14. The camera is mounted to a stage that permits fine-grained control over translation and rotation for calibration. To ensure that the co-location is accurate we projected a plain white pattern and assess that the shadows cast by the projector are not visible by the camera as shown in the inset of figure 3.14.

### 3.6.2 Test bed

To evaluate the performance of our prototype we used the drops generator test bed built by Peter Barnum and which engineering is detailed in his thesis [Barnum, 2011, Appendix A]. The entire set-up is depicted in figure 3.16 and display the colocated prototype in its box, the drops generator and a sample solenoid valve. The use of artificially generated water drops simplifies dramatically the testing phase while ensuring repeatability of the tests. Drawbacks are essentially the use of a controlled environment and the non-realistic physics of the generated water drops (drop size distribution and dynamics differ from real rain). We will discuss these issues later on.

The drops generator we use is similar in spirit to [Eitoku et al., 2006] though contrary to their set-up ours use 16 solenoid valves instead of a single one. The 16 direct-acting miniatures solenoids receive slightly pressurized water from a tank elevated overhead. Each of them can release drops at up to  $60Hz$  with different spatio-temporal offsets.

Contrary to [Barnum et al., 2010b, Barnum, 2011] the geometry chosen is unknown from our prototype - the emitters being randomly positioned. Depending on the geometry, drops may even occlude each other from the camera point of view which could also be the case in real rain.

### 3.6.3 Detection and Prediction Algorithms

As our prototype is run in a controlled environment (dark backdrop, stationary camera) both detection and prediction are quite trivial but should be fast enough to ensure low system latency.

First a background subtraction is performed to segment foreground (streaks) followed by median filtering and thresholding to find the bright water spots. These spots are then segmented with connected components. Prediction results of a simple shift on the vertical axis given the streaks length. Observations have shown that though slightly accelerating the speed can be approximated with a constant ( $16pixels/frame$ ) as the drops fall throughout the FOV.

Results of detection and prediction are shown for a single drop from a single emitter in figure 3.15 and from all 16 emitters in figure 3.15 releasing 32 drops per second. Since the rain generator is completely controlled, experiments are repeatable and we can evaluate the system accurately by comparing the predictions locations to the actual positions made visible by illuminating them throughout the field of view. For the single drop case from figure 3.15 we achieved an average of **99.7%** throughput and **83.6%** accuracy. Even using trivial processing with 32 drops/sec shown in figure 3.15 the system maintains very high light throughput **98.1%** though the accuracy drops to **54.16%**. Obviously the performance could easily be improved given the simplicity of the task but we choose execution time rather than accuracy.

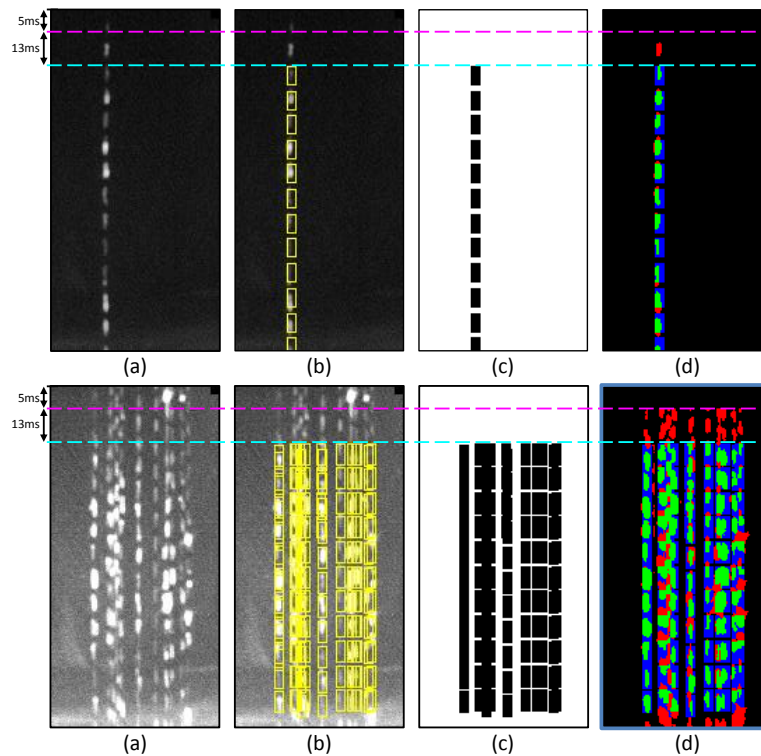


Figure 3.15: Predicting and avoiding illumination of a single drop (top) and of 16 drops falling at 32 drops per seconds. Drops are first imaged with  $5ms$  camera exposure time (shown above magenta dashed line). The system latency is shown above the cyan dashed line. Images shown are *composites* of the frames needed for the drops to traverse the FOV. a). Since the experiments are repeatable, we show the ground-truth image on the left with drops illuminated throughout the entire FOV. b). The drops are falling with near constant velocity at 16 pixels per frame, so prediction is straightforward (shown as yellow boxes). c). The projected frames show the pattern the system uses to avoid illuminating the drop. d). Red shows part of drop lit erroneously, green shows drop missed correctly, and blue shows misclassified pixels. In the single drop experiment (top), we achieved 99.7% light throughput and 83.6% accuracy. In the 16 drops experiments: light throughput is of 98.1%, and accuracy of 54.14%. Clearly, the prediction/tracking requires improvement and is neither fast nor accurate enough to handle complex situations.

### 3.6.4 Performance

#### 3.6.4a Test bed experiments

Using the test bed described above with 32 drops / sec we have taken photography of the full system running and they are shown in figure 3.17. The visual difference between naive illumination (i.e. plain light) and our selective illumination approach is manifest. As expected with our simulations, the drops appear bright and clear on top (i.e. system latency) but once tracked and being blacked out by the projector they are much less visible. From figure 3.17 – not only are the drops less visible, the visual discomfort caused by the falling raindrops is reduced and the overall visibility of the scene enhanced.

One may note from figure 3.17 that even the drops being blacked out remain slightly visible – our prototype deactivating *only* the rays that *directly* hit the drops. Indeed, other light stencils do

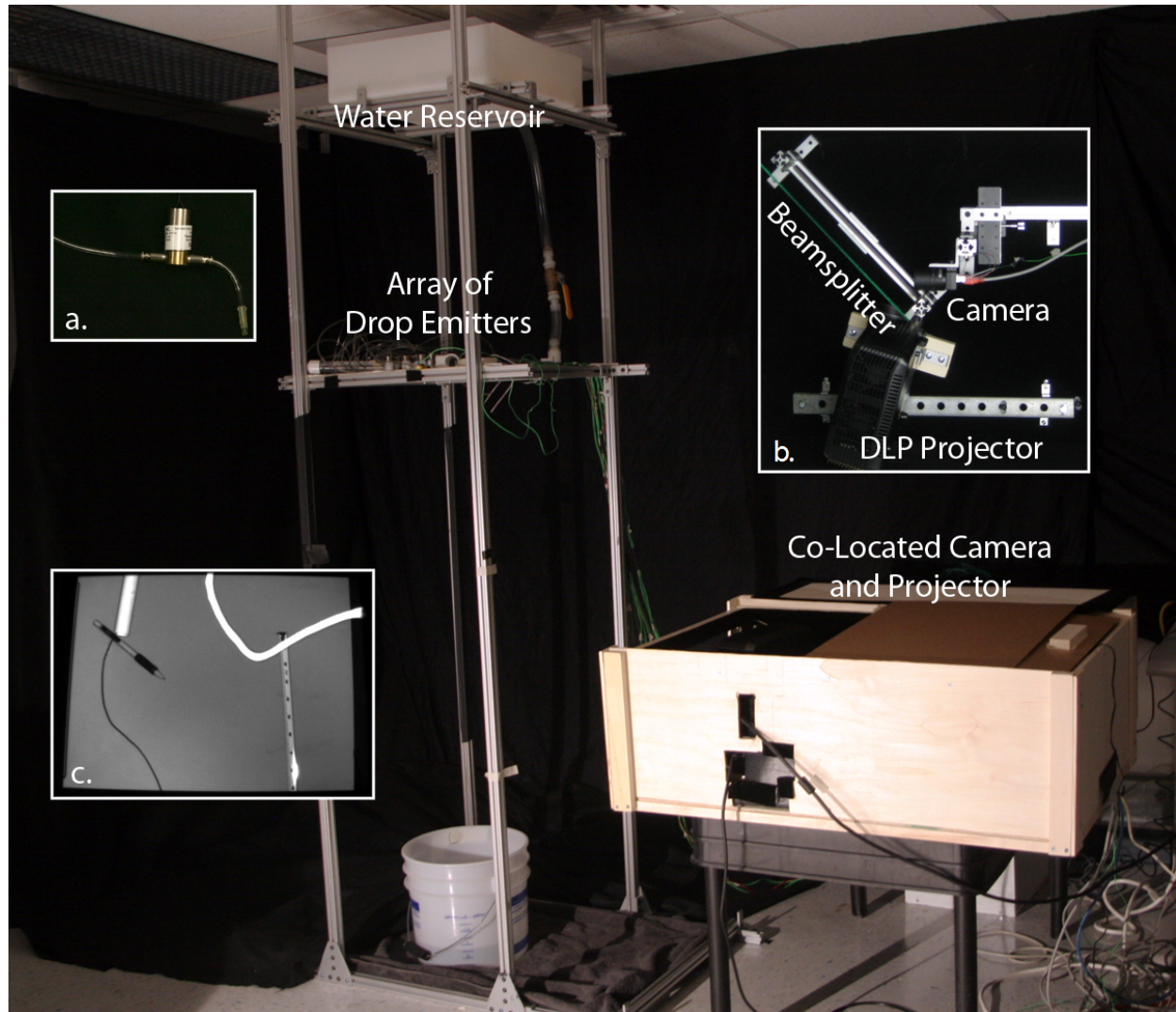


Figure 3.16: Overview of our prototype and artificial rain generator. a). One of 16 direct-acting miniature solenoids for generating a water drop. b). The co-located system – identical consists of a camera, projector, and beamsplitter. The camera is mounted on a stage that allows a fine adjustment. c). An example of two random objects being imaged by the co-located camera and projector pair. Notice the lack of visible shadows of the pen and the tape measure in this inset which assesses of the accurate co-location.

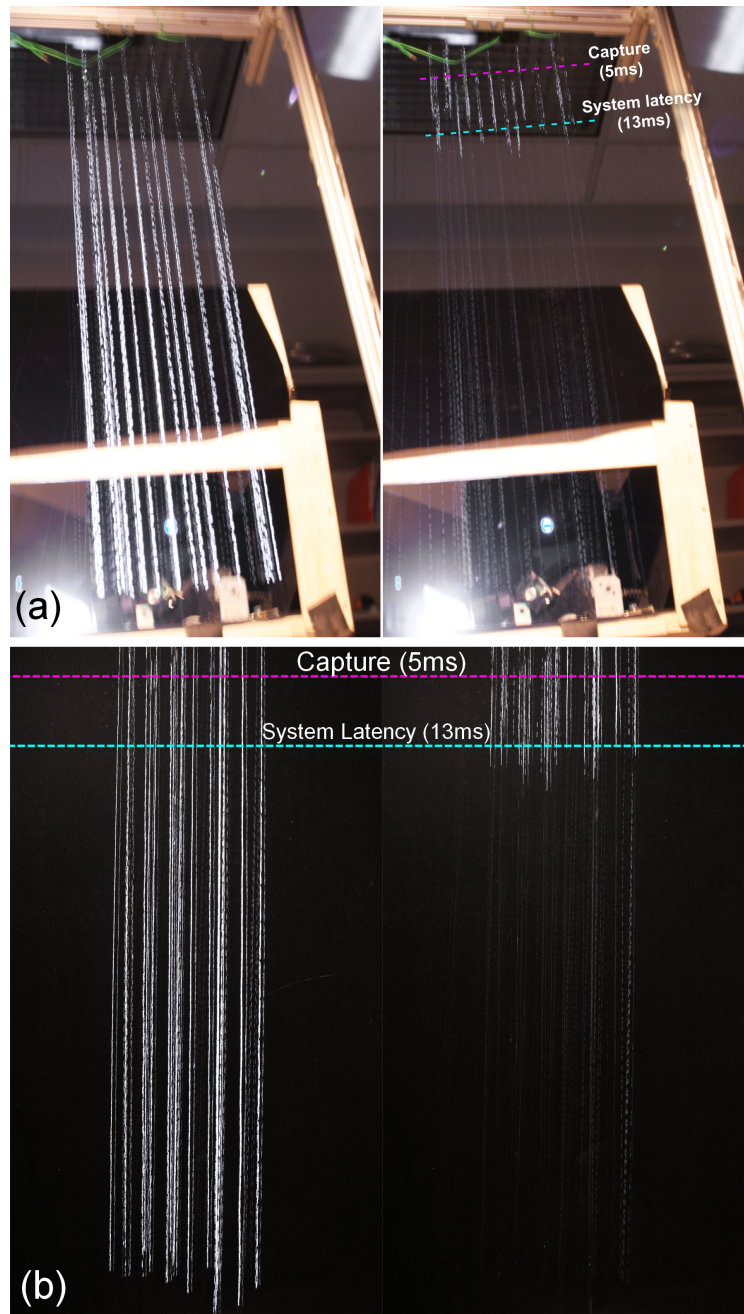


Figure 3.17: Naive illumination (a) versus fast reactive illumination (b) with our prototype running at  $120Hz$  (system latency of  $13ms$ ). Photos were captured with a  $2.5s$  exposure time. The prototype successfully improves overall visibility by selectively turning off rays intersecting generated particles. The same repeatable rain is used in both experiments with 16 drops emitters that generates drops at  $2Hz$ . Note the impact of system latency (above cyan line) where drops are lit for a small period of time. Below this line our reactive illumination clearly reduces drops visibility and high frequencies patterns that cause visual discomfort. Even when tracked drops remain *slightly* visible all along (notice the light streaks over dark background) due to prediction inaccuracies and global illumination.

intersect the drops after reflection and refraction (i.e. ambient lighting) making them appear as *dimmed* instead of invisible. Our system processing inaccuracy is also responsible of their dimmed appearance. The choice of speed instead of precision make detection/tracking the weakest link of our prototype as even few pixels errors is sufficient to shine a drop.

In the above cited example the loss of light intensity is negligible and the 60 seconds averaged throughput is of 97.4% (std of 0.83%) showing the validity of our approach. The current performance are limited by the processing inaccuracy but also by the projector refresh that requires a projection time of about  $9ms$  which further lowers the throughput of our system.

### 3.6.4b Simulating a moving prototype

We used the *prototype* recipe (cf. table 3.2) to evaluate the performance in a wide variety of weather conditions that our test bed and prototype can't allow us testing. Timings were measured from the real prototype and reported in the simulator<sup>3</sup>.

Despite the low resolution of  $120 \times 244$  and the  $12ms$  system latency, we can see from plot in figure 3.18 that our prototype shall perform reasonably in light or medium precipitations but can't handle extreme conditions. Noteworthy, while it performs poorly at high speed regarding system latency the throughput remains high at least for rain and hail. The faster the vehicle is moving the lower the accuracy making our system behaves just as standard headlights would do. For snow however, our system is clearly not accurate enough and the amount of light being loose is above the acceptable limit.

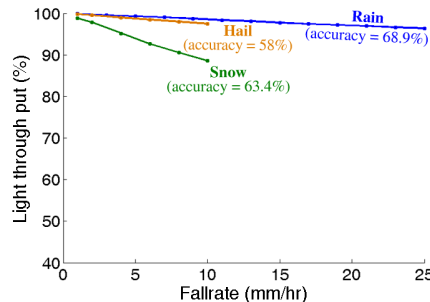


Figure 3.18: Performance the prototype system could reach at  $30km/h$  in rain, hail, and snow versus precipitation rate. Simulation shows that the light throughput remains high in all scenarios and that only 40% of the particles will be illuminated in the worst case, which should significantly improve driver's visibility.

### 3.6.5 Using the prototype to assess the simulations

Note that to validate our findings – and in the meantime the validity of our simulator –, we should have compared the performance obtained with identical experimental conditions. This could be done by reproducing the behaviour of the drops generator in our simulator. However, despite the software-controlled solenoid valves we found that it is not possible to predict *precisely* the size and birth time of each drop generated. Hence, it is questionable whether any comparison would be possible. Though the performance of our simulator seems in adequacy to our night-time experiments, further investigation is definitely required to assess of the validity of all simulations.

<sup>3</sup>We used a maximum refresh rate of  $111Hz$  that is a projection time of  $9ms$ .

### 3.7 Discussion toward a real system for rain

Up to that point we have presented a system with results from simulations and our off-the-shelf prototype. Because the challenge of illumination through precipitation is complex and unprecedented it was necessary to progress step by step but doing so we dismissed some difficulties while admitting some moot assumptions. We now investigate all these questionable choices to answer the following questions: Can our system accurately detect *real* raindrops at night ? What is the effect of using either physics models ? Can our system perform better somewhere else on earth ? How wind affects the light throughput ?

#### 3.7.1 Drops detection and tracking with vision algorithms

The heart of the project is the detection of the particles. To assess of this difficulty we have conducted outdoors experiments at night with sprinkler-rain and real rain. The set-up used here is the same as in figure 3.3 that is a camera Photron Fast-Cam 1024 PCI and a Powel Pro Light ( $\approx 4,000lumens$ ). When using short exposure time very few elements are visible. Then raindrops appear as bright elements that are segmented easily through gradients analysis and second moment to find the orientation of the streaks. Figure 3.19 shows pictures of this experiment. The raindrops were detected frame per frame, then tracked and prediction was computed with two or more observations. The results exhibit the feasibility of our system.

Analysing 3,437 of rainy frames with  $2ms$  exposure time we found that the average prediction error was of **1.99pxl** (std 2.27). More interesting is to note that this inaccuracy results of error in velocity estimation rather than of trajectory deviations. For every single drop we measured the distance of each observation to a linear trajectory (obtained through linear regression of all observations) and got an average distance of **0.30pxl** (std 0.44). These numbers show that even for real outdoor precipitation the approximation of linear trajectory is valid for drops while in the FOV. We also measured the velocity in the image space as a distance to a constant velocity and found that the average distance is **2.00pxl** (std 1.53). Hence, the velocity of the drops is not really constant but varies of  $\pm 13.35\%$ .

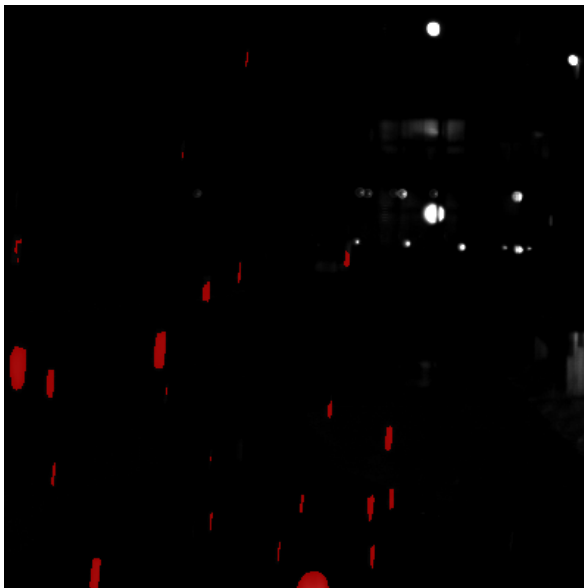
More interestingly, we found that tracking and prediction were even more accurate when using shorter exposure. Analysing 6,080 rainy frames with  $1ms$  exposure using the above described strategy we measured: an average prediction error of **0.88pxl** (std 1.16), a distance to linear trajectory of **0.25pxl** (std 0.32) and a distance to constant velocity of **1.41pxl** (std 1.49) which is  $\pm 30\%$ .

In light of these results, the linear trajectory approximation is valid and the velocity fluctuation is almost sole responsible of error in prediction. This in fact serves our cause, as it means that detection/tracking inaccuracy may be restricted only to the direction of the trajectory. The reason why velocity is not constant is not clear but there are two possible explanations – a) This might result of external forces (wind, drag) making the particles velocity slightly changing over time b) Inaccurate detection of the extrema of the drops may also be at stake as the start and end point (along the trajectory) are the most subject to motion blur. The second assumption is more likely to be factual.

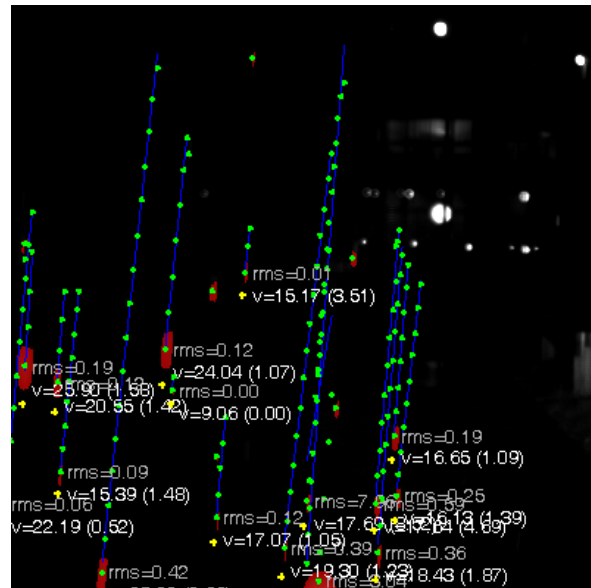
Measures of the prediction average error with these experiments (**1.99pxl** and **0.88pxl**) are also in fair adequacy with the detection/tracking error we arbitrarily used in our simulations (2pxl, cf. table 3.2).



(a) 2ms exposure (contrast +25%)



(b) Drops detection



(c) Tracking and Prediction

Figure 3.19: Experiments at night in rainy conditions with 2ms exposure time (set-up is identical to figure 3.3). Shown is a sample picture *cropped and enhanced* for readability (a), detection of drops (b) and tracking (c). In (c) green squares correspond to previous positions of the drops and yellow square next predicted position. Blue is the line regression of drops trajectory.

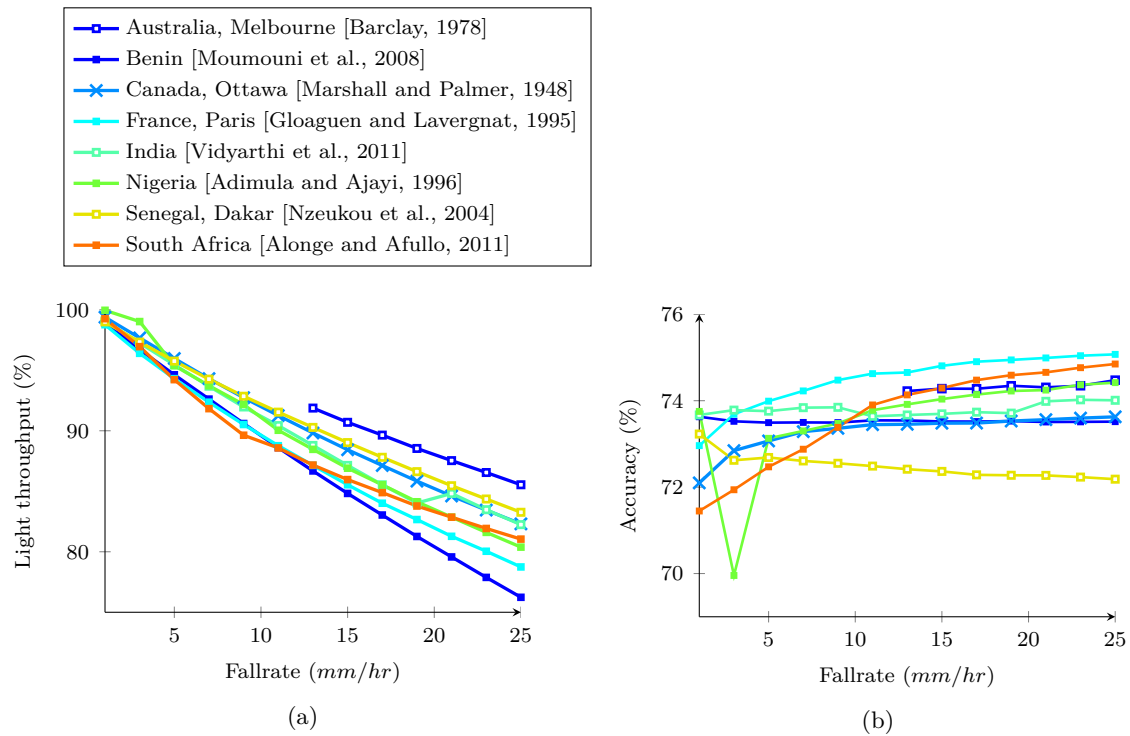


Figure 3.20: Performance of our system around the world using various physic models for particles size distribution (using recipe *realistic* from table 3.2). This shows that the throughput of our system (a) shall vary significantly depending on the location, whereas the accuracy (b) shall remain roughly similar ( $\pm 1\%$ ). Based on these simulations, the best place on earth would be either Australia or Africa (Senegal, South Africa or Nigeria) in terms of light through put, and Paris in terms of accuracy. The reason the through put is varying is the different type of rain around the world (different PSDs). On the other hand, the accuracy is only little varying since it is related to the mean raindrops velocity and not directly related to the PSD. To draw these plots we have implemented all the PSDs cited in the legend. Note: the valley of the Nigerian curve is due to discrete parameters in [Adimula and Ajayi, 1996].

### 3.7.2 Physics models worldwide

An outrageous number of experiments have conducted experiments in rain world wide and fitting these observations have shown the similar drops dynamics but the variability of Particles Size Distributions (PSD) around the world. Only Canadian experiments have been used to model precipitation in the simulation of section 3.5.2 but we may wonder what is the exact role of using either models on our system performance. We implemented the three most popular PSDs in our simulator (exponential [Marshall and Palmer, 1948], gamma [Best, 1950], log normal [Feingold and Levin, 1986]) as to simulate different types of rain events world wide. Then, having gathered a wide set of parameters from the vast body of meteorological literature we report in figure 3.20a the throughput of our studies for rain in 8 different countries. The settings used for simulation being the *realistic* recipe that is running at  $111H_z$  (see table 3.2 for complete settings).

The funny thing is that our system will indeed perform quite differently regarding the location. From figure 3.20a, our system shall reach a throughput of **85.54%** if run in **Australia** (Melbourne) while the exact same system would reach **76.24%** throughput in **Benin** (minus 9.3%) in  $25\text{mm/h}$  rain. Such difference is in fact due to the density of drops of each diameter. Hence, the same



volume of rain may be achieved with fewer but larger drops while in the meantime our system will perform better if fewer (thus larger) drops are present. Conversely, accuracy (figure 3.20b) is only slightly varying as it solely relies on particles size. This time, simulations using our set of PSDs show higher accuracy when our system is run in France (Paris) based on experiments conducted in Paris suburbs. In combination with the low throughput of France (Paris) from figure 3.20a it may infer that rain there exhibits larger number (i.e. lower throughput) of small particles (i.e. lower velocity, higher accuracy).

Notice that Canada (Ottawa) – the setting we used in all previous simulations – exhibits averaged performance for both throughput and accuracy. This in fact assesses that our simulation corresponds somewhat to a *world average*.

### 3.7.3 Influence of air flow and wind

In addition to vehicle motion, air flow and wind may affect the performance of the system. So far, we admitted the linear trajectory of the particles but in case of strong wind this may be violated. We believe that wind does not pose a significant problem because particles remain within the field of view for a short duration of time (few milliseconds). It may be assumed that the wind direction mostly remains stable within this time period. However, if the wind is too chaotic, higher tracking latency will be required. Using the same system parameters described above with 6 frames latency instead of 3 frames, accuracy (light throughput) in  $5mm/hr$  rain for a vehicle stationary is **74.68%** (**96.56%**), a vehicle at  $30km/h$  is **51.00%** (**96.92%**), and a vehicle at  $100km/h$  is **13.74%** (**98.60%**). Thus, in extreme situations, our headlight functions as a normal headlight with only a 14% reduction in precipitation visibility.

### 3.7.4 Conservativeness

Now, suppose that we are capable to perfectly detect the particles, choosing the light rays to switch off is function of the *conservativeness* of our system. Theoretically a particle imaged on a pixels array does not cover all pixels entirely so the question can be raised whether light rays corresponding to pixels partially covered should be switched off or not. Figure 3.21 depicts the general idea of conservativeness of the adaptive illumination. Let's denote the area of pixel  $P$  covered by the particles  $P_A \in [0; 1]$ . A conservative system shall deactivate all rays of pixels with  $P_A > 0$ , whereas a moderate one will deactivate only  $P_A \geq 0.5$  and a flexible one  $P_A = 1$ . Obviously the choice of either strategy alter the throughput of the headlight. As this could introduce a bias in the simulations we have chosen the moderate setting in all simulations shown in this chapter (i.e. a light ray is deactivated if  $P_A \geq 0.5$ ).

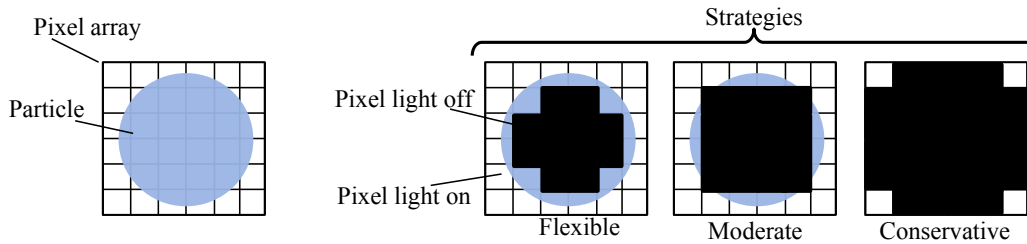


Figure 3.21: Strategies for conservativeness of the illumination. Left shows the projection of a single particle on a pixel array. The choice of which light pixels shall be turned off depends on the strategy. The more conservative our adaptive illumination, the lower the light throughput.

### 3.8 Conclusion

In this chapter we have presented the groundwork for a novel approach to improve the visibility of the drivers at night in precipitation. This was partially published in [de Charette et al., 2012].

Our proposal is to use a co-located system to image the particles and then reactively illuminate the scene without shining the particles. Using simulation we have shown that the system latency should remain below **2ms** if we are willing to maintain **90%** throughput for any type of precipitation though apart from these ideal and *theoretical* results reasonable performance can be achieved with higher system latency. Given current hardware limitation *13ms* latency may be accommodated (*realistic* recipe from table 3.2). In these conditions a vehicle moving at **30km/h** would reach accuracy of **73.2%** in rain, **69.5%** in hail, and **74.6%** in snow. The throughput being function of the particles density we've shown that with the same recipe the throughput may remain above **80%** for rain (*1mm/h - 25mm/h*) and hail (*1mm/h - 10mm/h*).

For our system to reach good accuracy in hail seems not possible for now (due to high velocity of hailstones) but snow is definitely easier to deal with as it allows accommodating higher latency (flakes fall slower). However, we've shown that the throughput suffers dramatically in snow as the particles are dense and thus require turning off a large number of light-pixel. Rain is a good compromise between particles velocity and particles density making it easier to deal with as a first step. So as to validate the findings of simulations a prototype for rain has been built and showed the intermediate performance we can reach with off-the-shelf components. The prototype exhibits the feasibility of our novel approach and visual results validate our expectations. The pictures from figure 3.17 not only show that the discomfort caused by raindrops is reduced but that the overall visibility is enhanced with our system, that is in adequacy with our review of the psychophysical studies (cf. section 3.3.2).

Additionally, to assess of the strong assumptions we have made (drop visibility, linear trajectory, constant velocity) we've conducted a number of real outdoor experiments in rain. Processing the data, we have exhibited that prediction of particles position may be achieved with an error as small as **0.88pxl** (std 1.16) when imaged with **1ms** exposure and **1.99pxl** (std 2.27) with **2ms**.

Obviously, the work presented here is only preliminary and all the experiments we have conducted have assessed of the road ahead to build a fully working system in precipitation. Still, our experiments (both real and through simulation) have shown that such type of novel illumination is already possible up to a certain extent giving us hope that a real system may be working in a near future.



## Chapitre

# Reconnaissance de Feux Tricolores

*Below is a French summary of the following chapter “Traffic Light Recognition”.*

Les feux tricolores sont des éléments clés d’une scène urbaine qui signalent des zones dangereuses où se croisent piétons et véhicules. Pour les ADAS en milieu urbain, reconnaître les feux tricolores est donc une condition *sinequanone* à leur fonctionnement. La variété des signalétiques, le peu d’amers, et leur petite taille à l’image rend cependant la tâche complexe.

Dans ce chapitre, nous proposons une nouvelle approche pour détecter les feux tricolores et identifier leur état par traitement d’image. Contrairement à la majorité de l’état de l’art, notre méthode est conçue pour être modulaire et permettre la reconnaissance de feux tricolores de différents pays. Une première étape consiste à identifier tous les spots lumineux par opérations morphologiques (cf. section 4.3). Pour chaque spot des candidats sont générés et évalués selon des contraintes géométriques et algorithmiques définies préalablement (cf. section 4.4).

Notre approche a été évaluée sur quatre bases de données de France, Suisse, et Chine (cf. section 4.5) avec et sans filtrage temporel. De plus pour permettre une future comparaison quantitative notre base de données française a été rendue publique et est déjà utilisée comme référence dans la littérature.



## Chapter 4

# Traffic Light Recognition

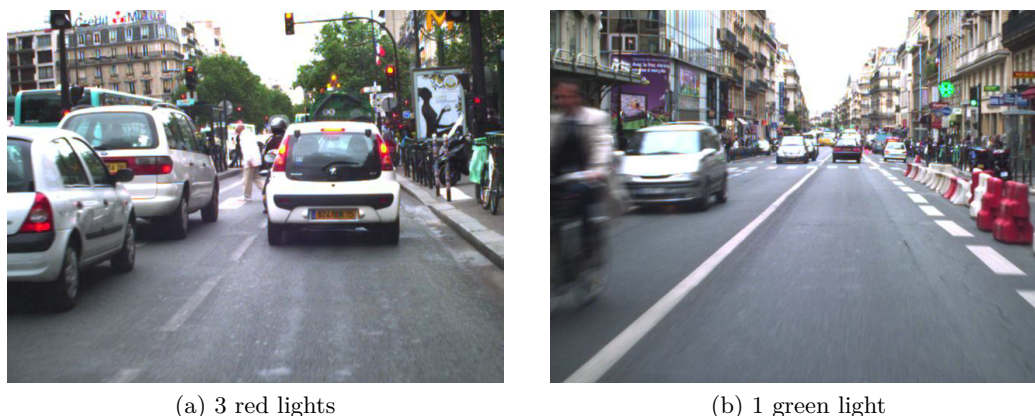


Figure 4.1: Example of dense urban scenes with traffic lights. The presence of other light sources and neon signs make difficult the identification of the signals in the scene. Below each sample is written the ideal output of a traffic light recognition system.

### 4.1 Introduction

Driver assistance systems (ADAS) have long time dismissed the problem of traffic lights recognition in favour of other detection tasks like traffic signs, pedestrians, or vehicles detection. Since ADAS are now supposed to be working in a large variety of scenes (highways, rural, urban) [Franke et al., 1998] they require more accurate understanding of the environment. The wide cluster of elements that intelligent systems must handle in these scenes includes traffic lights. The latter play a key role in the interpretation of an urban scene. For the driver they indicate the next action to take (go, stop or slow down) but they also suggest stationary vehicles on the road, incoming traffic, near-by intersection, and possibly pedestrians crossing.

From a NHTSA<sup>1</sup> study, in the USA 18.6% of the accidents in 2005-07 pertained to intersections with traffic signals [Choi, 2010]. Some of these accidents are due to red-light runners, roadside

<sup>1</sup>National Highway Traffic Safety Administration

advertisements [Crundall et al., 2006], inattention [Klauer et al., 2006] or timely detection [Davidse, 2006]. Concerning the latter Davidse advocates that 5 to 10% of the accidents could be avoided with adequate traffic signs/signals alerts. One of the reasons for a driver not to see a traffic light lies in his ability to distinguish colors [Nathan et al., 1964]. In the USA, 7 to 10% of the males are affected by red-green blindness<sup>2</sup> [RightDiagnosis, 2012, Montgomery, 2012] implying that they can not separate light color accurately (cf. figure 4.2b). A comprehensive understanding of such implications is described in the early work of [Kim et al., 2007]. In this paper, the American team described the positive effects of an ADAS helping the driver to “see the traffic lights”.

Hwang *et al.* have proposed to use also traffic lights recognition to refine the fusion of camera and map data based on the strong assumption that they suggest intersection right ahead [Hwang et al., 2006]. If so, the matching of the detected traffic lights and the intersection position in the map data allows projecting road direction on the windscreen, as in figure 4.2c. Lately, large firms like Google have also started mapping the positions of all traffic lights (cf. figure 4.2d) through a semi-automatic pipeline [Fairfield and Urmson, 2011] using advanced localization sensors. Given the surge of interest for on-board Geographic Information System (GIS) this can be used to enrich existing databases and improve path-planning algorithms.

Despite the existing literature a large amount of work remains to be done as none of the existing vision-based approach exhibits good enough performance for driver assistance. In this chapter, we present a two-stages traffic light recognition system intended for driver assistance. For such an application our system must be reliable enough but can supposedly miss occurrences of traffic lights. Researches have been conducted for the last few years on this topic though often limited to simple scenario or one type of traffic lights. Our approach relies on a single vision-sensor and a non-learning process that allows detecting various types of traffic lights. While first intended to be working in France we describe its performance also in China, and Switzerland.

#### 4.1.1 Appearance and detectability

In the image space traffic lights are usually only few pixels wide (much smaller than any other object, see figure 4.3b-e) and their appearance varies depending on the *active* light. Generally speaking traffic lights are made of three or more lights and oriented either horizontally or vertically. Other less frequent traffic lights with 1 or 2 lights may be seen in the USA, sometimes with 'X' shaped lights as in figure 4.3b.

Their position also changes from place to place. Traffic lights may be mounted on poles situated on the roadside, or hung from an horizontal pole or wire over the right-of-way. Hereafter referred as *road-side* and *over-the-road*. Figure 4.3 shows some traffic light samples from France and the USA. *Over-the-road* traffic lights are identified more easily as often represented against the sky whereas the complicate background texture of building and advertisements make the *road-side* traffic lights harder to identify [Kimura et al., 2007].

#### 4.1.2 Wireless communication

To avoid the problem of varying appearances some have proposed unusual solutions based on infrastructure-to-vehicle communication. The status of the traffic signal is then broadcasted using either radio communication (WIFI or RFID) [Gradinescu et al., 2007, Al-Khateeb and Johari,

---

<sup>2</sup>Reg-green blindness is the most common color blindness disease and refers to any difficulty to tell apart red from green. There is a strong male prevalence so only 0.4% to 2% of the US females are affected [Montgomery, 2012]. Noteworthy, the drivers may not even sense they are not seeing the color accurately.

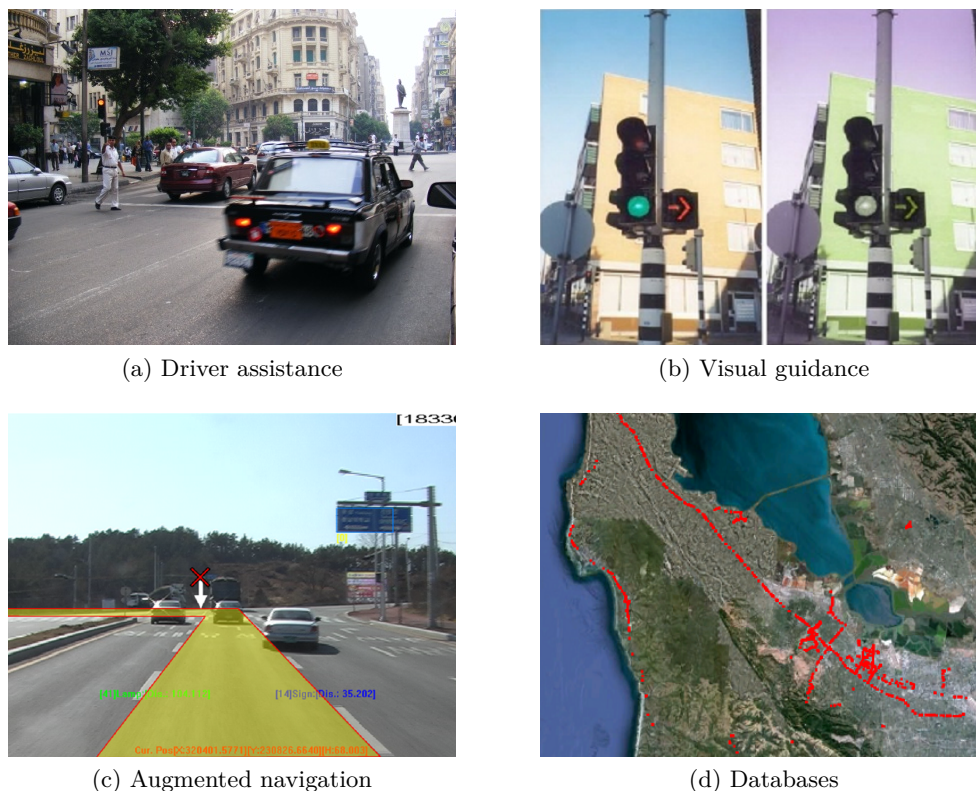


Figure 4.2: Examples of applications for traffic light recognition. (a) Driver assistance. Here a dense urban scene where the driver might not detect the red light on the left. (b) Visual guidance makes difficult the identification of the light status especially when the rule 'red always on top' is violated. Comparison of normal vision and a type of red-green blindness. Source: [Hwang et al., 2006] (c) Augmented navigation: Using traffic light structural information for better map-vision matching. Source: [Hwang et al., 2006]. (d) Databases: Enriching GIS databases like Google Maps in this case is achieved through large-scale traffic light recognition. Source: [Fairfield and Urmsom, 2011].

2008] or light communication with fast changing LEDs patterns [Sugawara et al., 2002, Wada et al., 2005, Premachandra et al., 2009]. Radio communications present the advantage of not requiring line-of-sight while having a longer range. This makes wireless broadcasting barely sensitive to weather conditions (snow, rain, glare, etc.). Hence, in a controlled environment they would reach almost perfect performance. However, and in contradiction to Pang and Liu who advocate that infrastructure-to-vehicle is the future of traffic light recognition [Pang and Liu, 2001] we believe that the effort required to replace the existing infrastructure is not worth it and clearly not suitable for the current needs of today ADAS. Since they require to be working in a wide variety of *unknown* environments. We ignore hereafter these approaches and focus only on passive vision-based solutions that do not require modification of the existing infrastructure.

An overview of the literature to detect traffic lights comes next.





Figure 4.3: Sample traffic lights from USA and France. (a)-(c) are over-the-road traffic lights and (d)-(e) are road-side traffic lights from our database. (f) is both.

## 4.2 Vision-based approaches

### 4.2.1 Existing traffic lights recognition systems

With Traffic Light Recognition (TLR) we are looking at finding the signals in the image space and identifying their current state. The varying appearance worldwide has led to numerous different approaches, listed in figure 4.4, using either a single vision sensor or additional localization sensor (GPS, IMU) to reduce the need of traffic light recognition to pre-mapped positions. We further describe the existing vision-based approaches while ignoring the algorithms details provided later on. The paragraphs below are named after the approaches listed in figure 4.4.

**DT** Until recently, many TLR were focusing solely on recognizing *over-the-road* traffic lights using a single light detection stage. They use the fact that lights colors (red, amber, green) are not likely to be found in the sky and considerate any colored spot in the upper part of the picture as a valid signal [Hwang et al., 2006, Kim et al., 2007, Shen et al., 2009, Yu et al., 2010]. Except for [Shen et al., 2009] the color thresholds are chosen empirically in the RGB or HSI colorspace. American TLRs use the multiple visible lights at the intersections in the USA as a 'more-than-one-light' rule for validation [Kim et al., 2007, Yu et al., 2010] while others take advantages of simple hysteresis as in [Shen et al., 2009] or [Kim et al., 2007]. The latter achieved impressive performance detecting 80% of the lights from as far as  $\approx 90m$ . Generally speaking note that all these existing systems perform well but with a working range limited to the simplest environments and assuming that traffic lights are located only in the upper part (i.e. against the sky).

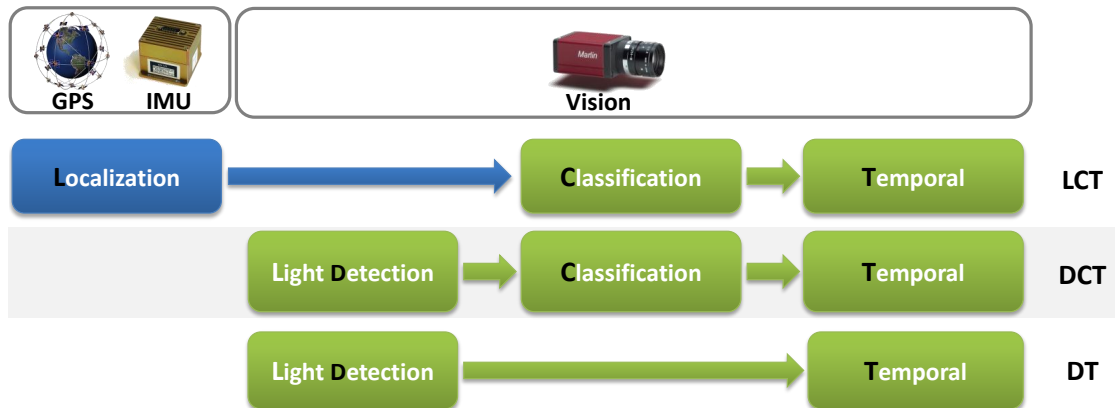


Figure 4.4: Various vision-based approaches to detect traffic light. A large number of the existing approaches are only looking for reddish or greenish lights in the sky. Such approaches can not be used when dealing with roadside traffic lights due to the complex background.

**DCT** To deal with more complex scenarios as well as to recognize traffic lights with cluttered background, an additional classification stage is used to tell apart traffic lights from other spurious lights. A number of structural methods were proposed using a binary shape detector [Lindner et al., 2004] or gradient features to either classify the light [Lu et al., 2008] or detect the backboard [Omachi and Omachi, 2010]. Statistical methods were also attempted to train classifiers for each light-state so as to find discriminative features. The Daimler research team has proposed using Neural Network [Franke et al., 1998] and Adaboost [Lindner et al., 2004] for German *over-the-road* and *road-side* traffic lights achieving results of  $\approx 90\%$  accuracy but with too many false positives. Posterior to our work other researches were conducted on both Chinese and German traffic lights using Adaboost [Gong et al., 2010] and SVM [Nienhüser et al., 2010], respectively. The latter using our spot light detection method (detailed in section 4.3) achieves performance similar to [Franke et al., 1998, Lindner et al., 2004] in a somewhat similar environment we are dealing with.

**LCT** For the specific application of autonomous vehicle a third approach has been proposed using a localization sensor (GPS). Taking advantage of a prior map matching the visual traffic light recognition is executed only when approaching an intersection as in [Lindner et al., 2004]. The same technique was used by [Angin et al., 2010] who developed a traffic light recognition cloud-service from a mobile phone for blind navigation. This is efficient but inherently limited to traffic signals at the intersection. Extending the idea of localization, Fairfield and Urmson proposed to pre-map (semi-automatically or manually) the position of traffic lights [Fairfield and Urmson, 2011]. The Junior Stanford autonomous car [Levinson et al., 2011] also uses pre-mapping with a GPS and an Inertial Measurement Unit (IMU). The latter provides information about vehicle movement that allows restricting the search space. Curiously, even with the additional IMU they achieved only  $\approx 92\%$  accuracy while Google’s team from Fairfield achieved  $\approx 99\%$ .

### 4.2.2 Our approach

**Requirements** Conversely to some of the aforementioned techniques our willing is to develop a solution for Traffic Light Recognition that would be working with any type of traffic lights (*over-the-road, road-side*) in dense urban environments. In such environments the task of having a practical and accurate TLR is significantly more complex due to the number of look-alike elements. As for the driver, we will show later on that advertisements or lights from other vehicles may be a source of distraction for any TLR. Additional requirements from our industrial partner (Valeo) is the use of a cheap solution based on a single vision sensor (i.e. no localization unit) and the need to keep a computation time 'acceptable' for a real-time on-board system (min. 15 FPS). Interestingly, the artefacts that comes with a cheap sensor (motion blur, low dynamic, noise, etc.) prohibit *de facto* the use of some techniques while making the traffic lights detection challenging.

In terms of functionality for an ADAS application, the output of our algorithm must remain reliable enough while detecting traffic lights sufficiently far from the ego vehicle so as to alert the driver in time. This means being able to detect traffic lights when they are only few pixels wide. From figure 4.3 one might observe that traffic lights all over the world are drastically different so that a single system working in all countries is clearly impossible. Hence, our system is initially designed to be working with French traffic lights though our willing is to have a system modular enough to be used in different countries with minimal adjustments. Such requirements imply not to use country-specific properties of the traffic lights. We compare and discuss the performance of our system in both France, Switzerland and China in section 4.5.

Measuring the performance of our TLR is achieved using two metrics that we are willing to maximize simultaneously:

**Recall** is the ability of our system not to miss any traffic light in sight.

$$\text{Recall} = \frac{\# \text{ of Traffic lights detected}}{\# \text{ of Traffic lights}} \quad (4.1)$$

**Precision** is the ability of providing accurate output.

$$\text{Precision} = \frac{\# \text{ of Traffic lights detected}}{\# \text{ of Traffic lights detected} + \# \text{ of False alarms}} \quad (4.2)$$

For ADAS the most important is to maintain a high level of precision so as to not alert the driver by mistake. That is reducing false alarms.

The framework we're using is depicted in figure 4.5 and is similar to the literature.



Figure 4.5: Overflow of our complete Traffic Light Recognition.

In the first stage, we seek to restrict the search space to the spot lights that could either be the light of a traffic signal or any other spot. Then the second stage extract candidates regions

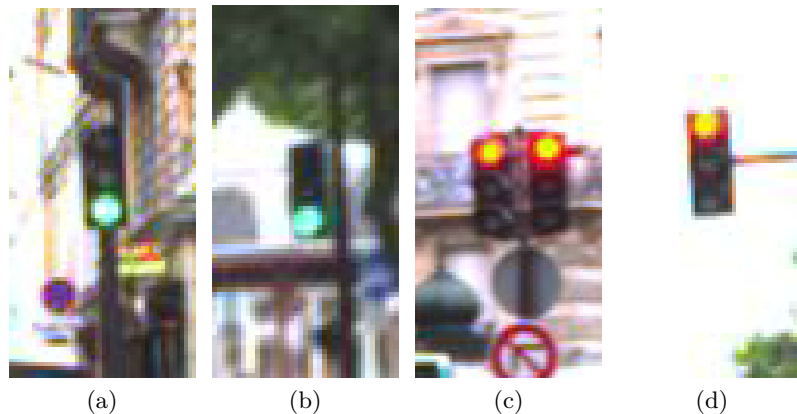


Figure 4.6: Enlarged samples of traffic lights from our Paris dataset. Note that the light emitted produces saturation of the color due to the poor dynamic of our vision sensor. (a)(b) Green lights oversaturate to cyan or white and (c)(d) red lights are imaged mostly amber.

from the spot lights and classifies them into  $\{Green, Amber, Red, BackGround\}$  so as to build a number of observations that are being refined over time with the temporal stage to identify the traffic lights and their current state  $S \in \{Green, Amber, Red\}$ .

For each stage, a review of the state of the art is provided. *In fine*, the performance of the whole system is described in section 4.5 and compared to some extent to the state of the art.

### 4.3 Spot light detection

The key property of all traffic lights among the world is their ability to emit light. Therefore detecting the spot lights as a first step ensure not to dismiss any of the traffic lights in sight. We use the following definition for a spot light: any spot *in the field of view* emitting light. Hence, we dismiss any source of global illumination (such as the sun). While this may seem trivial, when imaged with a low dynamic sensor (which is the case with the camera used) the light being emitted oversaturates which prevents from sensing the light color accurately. As an example, samples from our Paris dataset are shown in figure 4.6. In this figure the saturation caused by the light emission is obvious as the green and red lights are respectively imaged cyan and yellow. This is even better sensed in figure 4.9. Additionally, spot light detection is made harder by the artefacts that produces the sensor in the immediate vicinity as may be seen in figure 4.6b and 4.6d. The light emitted appear to be in contact with the background where the fact is that the French traffic lights are surrounded by a thin dark board. Such artefacts are mainly due to the on-board camera sensor but may also result of motion blur. So as to detect the spot lights accurately we are seeking to robustly deal with such artefacts.

Next we review the state of the art for spot light detection.

#### 4.3.1 Related work

To identify spot lights two types of approaches may be found in the literature the first one relying heavily on chromatic segmentation and the other on gradients and transforms, often using a chromatic segmentation first.

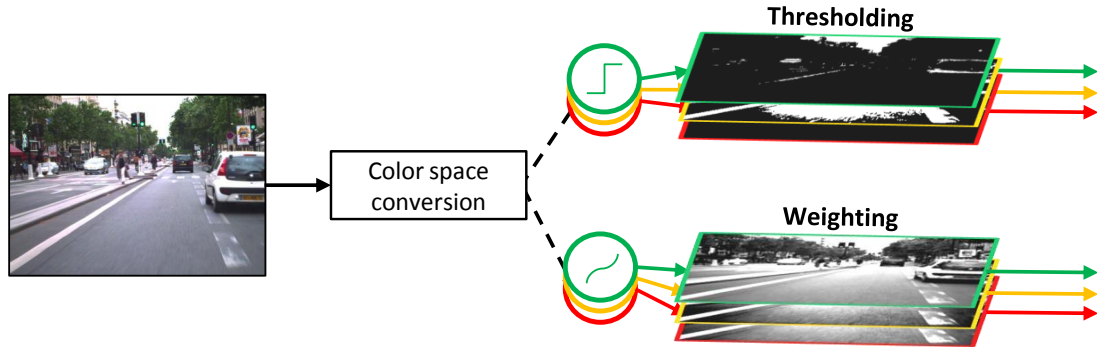


Figure 4.7: Different strategies applied for chromatic segmentation in order to isolate from the rest of the image. Most often the thresholding strategy is used but weighting provides finer output. Note that amber is shown but often dismissed.

#### 4.3.1a Chromatic segmentation

Currently, almost the entire literature on traffic lights uses a first chromatic segmentation to detect traffic lights. Most often the segmentation is achieved by splitting the input color image into  $N$  planes each one corresponding to a colored light searched for – that is green, red and sometimes amber. The spot light detection algorithm is then applied on each plane. Segmentation of the light from the traffic signals has been attempted with different colorspace:

- **RGB:** [Kälberer, 2003, Lindner et al., 2004, Kim et al., 2007, Yu et al., 2010]
- **HSV:** [Nienhüser et al., 2010, Gong et al., 2010, Li et al., 2011]
- **HSI:** [Hwang et al., 2006, Lu et al., 2008, Shen et al., 2009]
- **Lab:** [Xu et al., 2010, Siogkas et al., 2012]

**Thresholding** To discriminate the most, the segmentation must be applied in a colorspace where the mapping of the emitted light occupies the narrowest area. Additionally, the bigger are the gaps between the mapping area of all light color, the more efficient is the segmentation. From our experiments shown in figure 4.9 the choice of the colorspace is determinative. RGB for instance is clearly inadequate as the color distributions of red, green and amber are too sparse and known to be very responsive to illumination changes. More appropriate is to use a colorspace that separates the color from its brightness such as HSV, HSI, Lab.

Often this is achieved using a pixel-wise empirical threshold as in [Hwang et al., 2006, Kim et al., 2007, Lu et al., 2008, Yu et al., 2010, Omachi and Omachi, 2010, Xu et al., 2010, Nienhüser et al., 2010], but Look Up Tables (LUT) built from a large set of hand-labelled samples are also used [Franke et al., 1998]. While empirical thresholding is definitely faster to put in practice, LUT are significantly more precise since they allow defining finer boundaries of the light color distribution.

A number of publications report the difficulty to distinguish amber from red due to their close color distribution [Hwang et al., 2006, Kim et al., 2007]. Such statement is reinforced when using a low cost sensor. From figure 4.9 we observe the difficulty to tell amber apart from red light as the color distributions of both lights overlap in all colorspace. More complex approaches than a simple threshold are required for that.

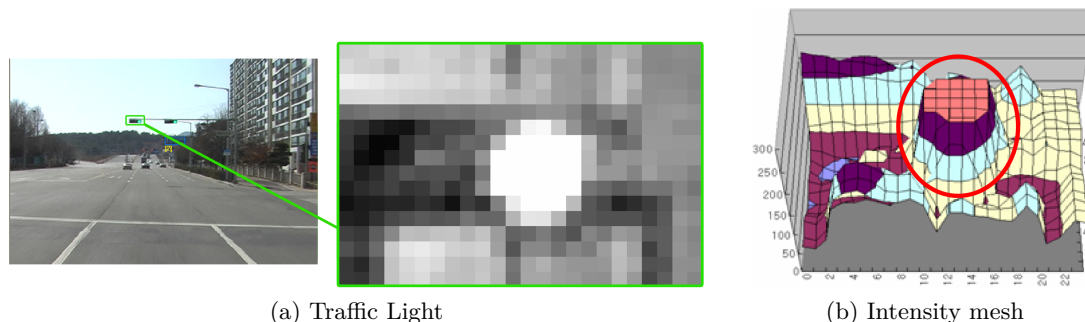


Figure 4.8: Illustration of the Gaussian convolution to find the center of the lights. (a) is the selection of the candidate which results of the chromacity segmentation applied first. (b) 3D mesh of the intensity of the candidate region. The light appear as a flatten Gaussian (circled in red) that may be detected using a multiscale Gaussian convolution. Adapted from [Hwang et al., 2006].

**Weighting** Instead of a binary threshold, some light detectors use weighted LUT [Levinson et al., 2011] or 2D Gaussian color-distribution [Lindner et al., 2004, Shen et al., 2009]. For example, Shen *et al.* use 2D features  $X(u, v) = [Hue(u, v), Saturation(u, v)]$  and define Gaussian color-distribution for each light based on the mean, and the covariance matrices computed on a test database. In addition to the ability of distinguishing red from amber (with a good quality sensor) it allows finer granularity in the detection.

Actually, when using a good quality sensor color segmentation makes perfect sense. In few countries, the colored light is even defined as a specific area in the CIE chromacity diagram and can thus be easily segmented using a centred Gaussian distribution. [Hwang et al., 2006] and [Shen et al., 2009] advocate that this is the case in South Korea and Ohio USA<sup>3</sup>. Needless to say it is obviously not true in France where regulation on traffic lights only state vague name "green", "amber" and "red".

Once each plane is segmented using one of the aforementioned techniques, spot lights are generally separated from spurious regions using geometric constraints but an other alternative is to use gradients or transforms.

#### 4.3.1b Gradients and transforms

Using the emitted light property or the shape of the spot light, transforms may be applied to segment spot light from spurious regions after the chromatic segmentation stage.

**Circular voting** Lindner *et al.* and recently Omachi and Omachi have used the circular property of a spot light to segment it with a Hough voting algorithm. The latter uses the normal to the gradients of the input image to accumulate votes. Hough has shown some great success when thick backboards surround traffic lights as in [Lindner et al., 2004]. But the voting doesn't permit to identify circular lights if they do not exhibit sharp edges. A small adjustment in the voting strategy was proposed by [Omachi and Omachi, 2010] to promote *filled* circles.

<sup>3</sup>Important is to say that Shen *et al.* do not provide any reference to reinforce their statement, and Hwang *et al.* provide an *unverifiable* reference.

**Gaussian convolution** A somewhat unpopular proposition is to apply a 2D Gaussian convolution on the input image [Hwang et al., 2006]. Since the light decay follows a Gaussian distribution (see figure 4.8) the 2D convolution output is higher when centred on the light. From Hwang *et al.* the performance are astonishing but they benefit of a strong color segmentation (using CIE chromacity regulation) and strict structural knowledge (i.e. world position of the traffic lights are known) that allows inferring the appropriate size of the light spot. Indeed, the most difficult challenge is to choose the right size for the kernel of convolution. Strategy using various sizes for the kernels is time consuming and does not guarantee to provide good results since due to the unknown light background the response of the convolution is non-monotonic regarding the size of the kernel.

### 4.3.2 Our approach: Grayscale segmentation

So far all the TLR first use a color segmentation to identify the spot lights. Regarding the quality of our vision sensor chromacity is not reliable and so as to deal robustly with this issue we use the intensity to detect spot lights regardless of their chromatic. There are several advantages of using only intensity. Most importantly, this drastically simplifies the problem caused by illumination variation while avoiding repeating the same process for each of the segmented color planes (as depicted in figure 4.7). On the other hand not using the color information prevents detecting the lights which intensity appear similar to their background. We will discuss later the cases where such an approach fails and propose different strategies to solve them.

Looking at a grayscale picture of a scene, two properties can be used to segment spot lights:

- The light appears to be separated from its background and significantly brighter.
- While the light shape is not a perfect circle (mostly due to saturation and artefacts) its shape remains somewhat circular.

We use these two properties to identify the spot lights.

#### 4.3.2a High trends

To isolate the bright spots we used the white top-hat transform from morphological mathematics, denoted  $T_w$  in equation 4.3. It is literally defined as the difference of the input image  $f$  and its opening by some structuring element  $b$ :

$$T_w(f) = f - f \circ b = f - (f \ominus b) \oplus b \quad (4.3)$$

where  $\circ$  is the morphological opening,  $\ominus$  is the erosion and  $\oplus$  the dilation. We refer to [Meyer, 1977, Serra, 1982] for its detailed definition. Result of white top hat operation is shown for a sample binary image in figure 4.10. When applied on grayscale image, the result of the white top hat operation is a floating matrix where slow trends are set to zero and others to strictly positive value.

**Structuring element** The size and shape of the Structuring Element (SE) impact greatly the performance and computational cost of the top hat. Most importantly, the size of the element  $S_{SE}$  is responsible of which elements are being segmented. The erosion is the critical step of the top hat as the elements need to be fully removed during this process so as to be highlighted by the top hat. This is the case only if its diagonal is *smaller* than  $S_{SE}$ . Remarkably, a spot is

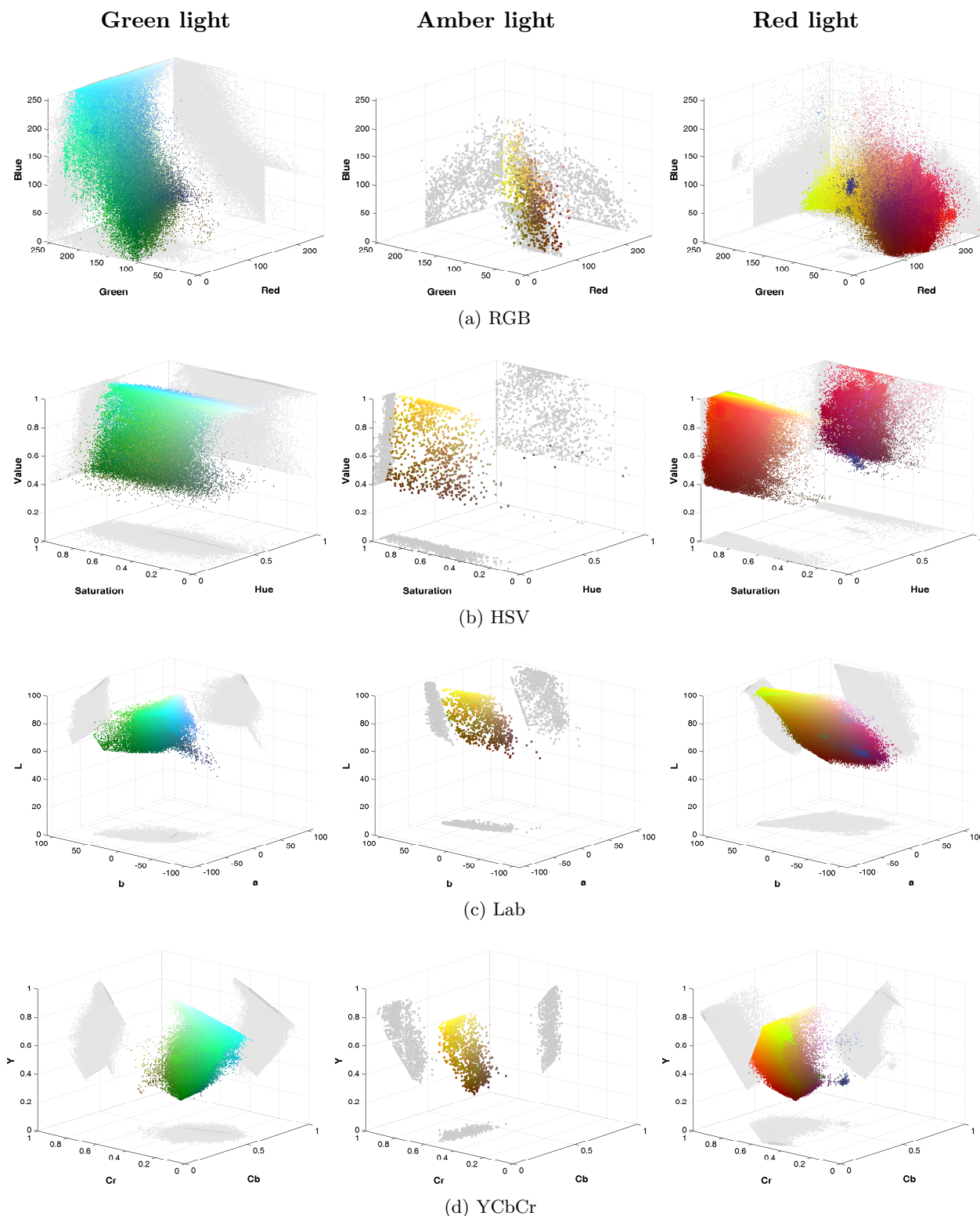


Figure 4.9: Area occupied by green, amber and red lights in various colorspace from more than 10,000 hand-labeled traffic lights. Pixels with brightness below 40% (using HSL definition) were dismissed as outliers. Projection of the colored pixel are displayed as gray dots on the three 2D planes. (a) RGB and converted to (b) HSV, (c) Lab, (d) YCbCr. As the area should remain as small as possible the use of RGB is clearly prohibited.



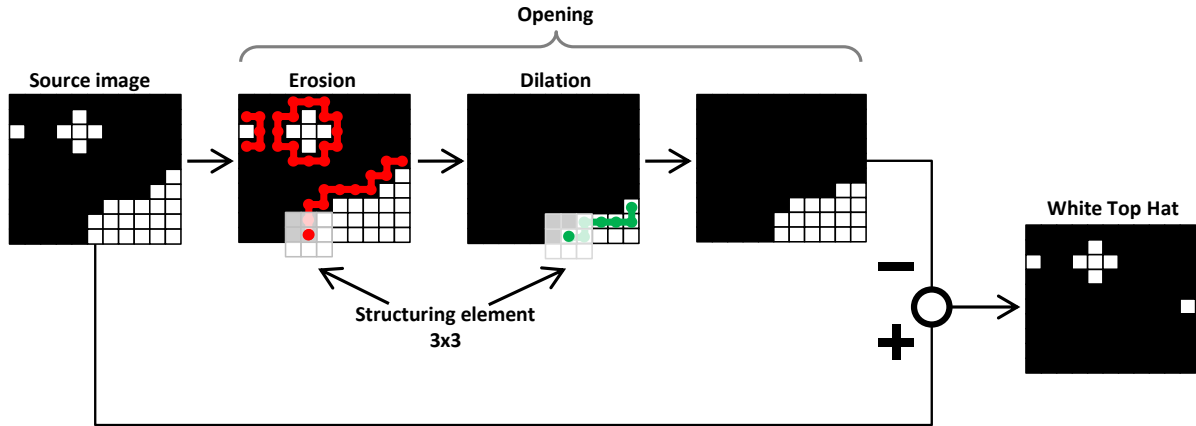


Figure 4.10: Illustration of the White Top Hat (with a 3x3 square element) on a binary image.

segmented identically by any centred structuring element which size ( $S_{SE}$ ) is larger than the spot diagonal. Thus, the size of the structuring element is only responsible of the largest element that is segmented. This has to be compared with Gaussian convolution (as in [Hwang et al., 2006]) that would require the use of multiscale kernel (i.e. longer processing time). Figure 4.11 illustrates the use of different sizes for the structuring element on a typical traffic scene.

Some have discussed the shape of the structuring element but it is not in fact as decisive as one may think since it is not related to the shape of the segmented elements. Figure 4.11 shows several examples of a urban scene after applying a top hat transformation with a circle or a square element of varying size. Not only the spots are segmented similarly with a circle or a square element but the circle element produces roundish artefacts (see the back of the car in figure 4.11f) and consequently slows down the morphological operation. Hence we used a square element.

Despite the fact that we use a square element (i.e. faster than circle) morphological operations are time consuming. The time required is directly related to the size of the structuring element (i.e.  $S_{SE}$  parameter)<sup>4</sup>. Given that the size of the structuring element also drives the segmentation it should be fixed such that  $S_{SE} = \text{MaxSpotSize} + 1$ , where  $\text{MaxSpotSize}$  is the largest spot diameter to segment in the image space.

### 4.3.2b Discrimination

Having highlighted the regions of interest using the top hat morphological process we aim now at isolating the spot light with a segmentation process. We use an efficient region growing approach and extract the connected components that are spot lights still using only intensity.

Often a simple binary threshold is used but the issue of such an approach lies in the appropriate choice of the threshold that dramatically impacts the shape of the components extracted. Here, we extract the local extrema  $E_{\max}\{E_1(x, y), E_2(x, y), \dots, E_n(x, y)\}$ . Theoretically, as the traffic signals emit light the very centre of each *on* lamp is a local maximum. Due to artefacts (i.e. noise, motion blur) this is rarely true so an eccentric expansion is not appropriate. Instead we use the local extrema as the seeds of a region growing algorithm that ensure keeping the regions

<sup>4</sup>A basic implementation of erosion/dilation has a complexity of  $O(n \times m \times S_{SE}^2)$  (the input image being  $n \times m$  pixels) so the complexity of the top hat operator is basically:  $O(n \times m \times (1 + 2S_{SE}^2))$ . Optimisations like those of [Van Droogenbroeck and Talbot, 1996] can lower the complexity to  $O(n \times m \times (1 + 2S_{SE}))$ .



Figure 4.11: Cropped picture of a typical urban scene with traffic lights at different distance from the camera. (a) is the sample picture and (b)-(g) are samples white top hat output with structuring elements of varying size. Note that when  $S_{SE} = 5$  the large spots (i.e. corresponding to close traffic light) are not segmented (b) and (e). Additionally one may see that the top hat is different with a square (b-d) or a circular element (e-g) though the spots are segmented identically.

homogeneous. Consider a sample region  $R_i$ . The expansion algorithm starts from the seed (one of the local extrema) and expands to the neighbour pixel (denoted  $Nbg$ ) which intensity distance to the mean region intensity is the smallest and agrees with a predefined threshold variance. That is:  $\min(\text{abs}(Nbg - \text{mean}(R_i))) \leq S_{\text{variance}}$ . We reiterate this process until none of the neighbours satisfies this condition, that is when the region is completed.

The choice of the adequate strategy for the region growing is crucial as it has an impact on the shape of the regions extracted. We could have been using an energy function that grows toward a perfect circular shape but experience have shown us that this it is not suitable for small elements. Additionally as our algorithm is based on the measure of variance it makes sense with the Gaussian behaviour of light emission. Hence, what we are doing here is to segment the flatten top of the Gaussian.

Since region growing is very time consuming, it is in fact performed on a clamped version of

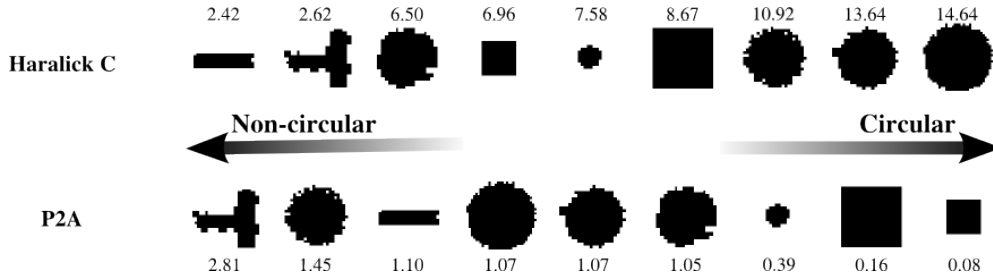


Figure 4.12: Circularity of digitalized shapes with P2A and Haralick C. The circularity scores are shown next. P2A is computed with:  $P2A = |1 - \frac{Perimeter^2}{4\pi \times Area}|$ . For perfect circles, P2A tends toward 0 while Haralick C tends toward  $+\infty$ . Notice that P2A indicates square with better circularity score than actual circles due to the discretized space.

the top hat image we denote  $T_c$ :

$$T_c(x, y) = \begin{cases} 0 & T_w(x, y) < S_{clamp} \\ T_w(x, y) & T_w(x, y) \geq S_{clamp} \end{cases} \quad (4.4)$$

Note that the region growing algorithm is applied on  $T_w$  (i.e. the top hat image) using extrema from  $T_c$  so as to be sure that the clamping does not play a role in the segmentation. Indeed it is only intended to speed up the process by selecting only local extrema bright enough.

Now that we have extracted homogeneous regions we need to discriminate those that could be actual spot lights. We use two major properties to tell apart spot from spurious regions.

**Circularity** To evaluate the circularity of the regions the  $P2A$  measure is generally used that refers to the distance between the ratio  $\frac{Perimeter^2}{Area}$  and those of a perfect circle that is  $4\pi$ . However, this measure suffers from drawbacks when applied to the discrete spaces [Rosenfeld, 1974]. Namely it implies two different isoperimetric regions to have the same  $P2A$  and a digitalized circle to have a dramatically different  $P2A$  than a circle in the non-discrete Euclidean space. Though we have first used P2A we found that it is not reliable enough for our application and instead used the Haralick C's measure, hereafter denoted  $C$  [Haralick, 1974]. The latter is computed using the position of all border points  $b$  as:

$$\begin{aligned} C &= \mu_b / \sigma_b \quad \text{with,} \\ \mu_b &= \text{mean}(\|b - \bar{c}\|) \\ \sigma_b &= \text{std}(\|b - \bar{c}\|) \end{aligned} \quad (4.5)$$

where  $\bar{c}$  is the centroid of the region that we estimate as the mean of  $b$ . Because it uses the relative position of the contours points the  $C$  measure is far less sensitive to the aliasing effect or fractal effect that occurs for digitalized shape. To verify this we have compared nine different digitalized shapes in figure 4.12, that we ranked from non-circular (left) to circular (right) with Haralick C or P2A. First obvious conclusion is that none of the metric is perfect. Still, Haralick C shows its superiority as P2A mis-ranked square as more circular than actual circle. On the other hand, the biggest inconvenient of using the  $C$  measure (instead of other circularity measure such as P2A or M) is that it is not an *objective* measure. Indeed,  $C \rightarrow \infty$  for a perfect circle.

As all other circularity metrics in the discrete space, note that the Haralick  $C$  measure is only rotation but not scale invariant. However, in their review Montero and Bribiesca have shown that the variations are more important for closed-circle shapes [Montero and Bribiesca, 2009]. Random shapes that are far from a circle keep a  $C$  measure that is somewhat similar regardless of their scale. The authors also advocate that using compactness is more robust. But none of the compactness measures distinguish a perfect square from a perfect circle that is crucial in an urban scene. Finally, an other commonly used approach is the *shape reference* that is often achieved through pixel-wise convolution. This is a robust technique to evaluate whether a region matches a predefined shape as a circle but not to evaluate its actual distance from the shape.

Hence, we define a region's circularity threshold  $S_{circularity}$  and validate only the regions which Haralick  $C$  measure  $C_i$  is:

$$C_i \geq S_{circularity} \quad (4.6)$$

**Continuity** Another simple and discriminative property of light from traffic signal is its continuity that is to say its ability to form a single solid region. Thus, we filter regions having holes with this simple rule:

$$Continuity_i = \begin{cases} 1 & \forall p_{x,y} \in R_i, p_{x,y} > 0 \\ 0 & \text{else} \end{cases} \quad (4.7)$$

**Validation** From the set of regions that were segmented from the top hat, only those that satisfy both circularity and continuity are validated as spot lights. The output of this stage is a set of spot lights defined as  $\{L_0(c_0; r_0), \dots, L_n(c_n; r_n)\}$  where each spot light is described by its centroid  $c(x, y)$  and its radius  $r$ . The estimation of the latter is obtained as a mean of the distance of all contours points to the centroid. Some sample outputs of the spot light detection stage are shown in figure 4.13 with the circularity indices of the spots.

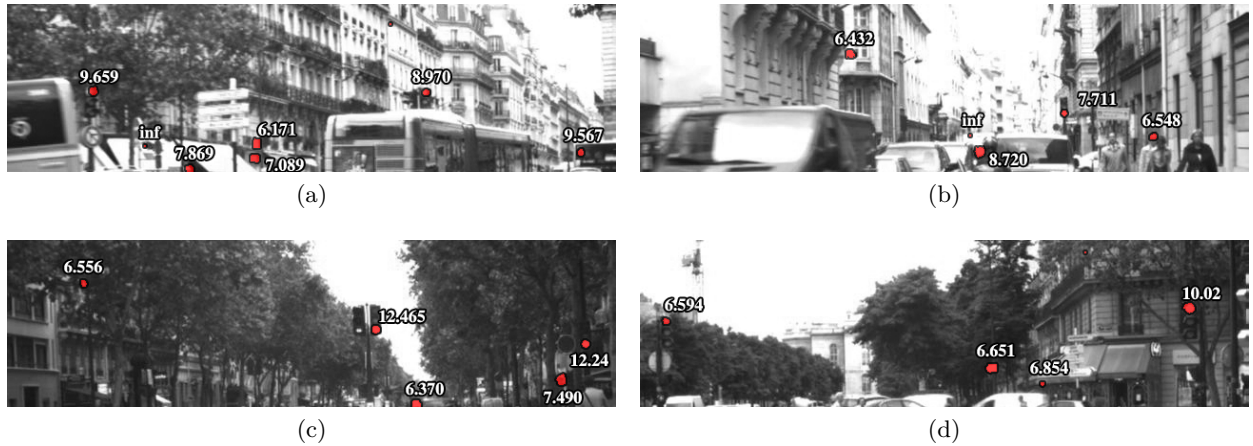


Figure 4.13: Sample outputs of the grayscale spot light detection. The circularity index is written for each spot. Note that spots that do not belong to traffic lights are segmented as well. *Pictures cropped to enhance visibility.*

The algorithm of the complete grayscale segmentation of spot lights is written as Pseudo-Code in algorithm 1. Next we detail its performance.

**Algorithm 1** Spot Light Detection

---

```

1: procedure EXTRACTSPOT( $f$ ) ▷  $f$  is the grayscale image
2:    $f \leftarrow \text{normalize}(f)$ 
3:    $T_w \leftarrow \text{whitetophat}(f, S_{SE})$ 
4:    $T_c \leftarrow \text{clamp}(T_w, S_{clamp})$ 
5:    $E_{max} \leftarrow \text{regionalextrema}(T_c)$ 
6:    $Spots \leftarrow \emptyset$ 
7:   for all  $Seed \in E_{max}$  do
8:      $Region \leftarrow Seed_{x,y}$  ▷ Use local extrema as seed
9:      $RNbg \leftarrow \text{ExtractNeighbours}(T_w, Region)$  ▷ Set of Region's neighbours
10:     $RVariance \leftarrow S_{variance} \times Seed_{val}$  ▷ Compute the variance allowed
11:    while  $\min(|RNbg - \text{mean}(Region)|) \leq RVariance$  do
12:      ▷ Append neighbour with closest intensity to the Region's mean
13:       $Region \leftarrow \text{argmin}_{p_{x,y}} (|RNbg - \text{mean}(Region)|)$ 
14:       $RNbg \leftarrow \emptyset$ 
15:       $RNbg \leftarrow \text{ExtractNeighbours}(T_w, Region)$ 
16:    end while
17:     $b \leftarrow \text{contours}(Region)$  ▷ Extract coordinates contours
18:     $\bar{c} \leftarrow \text{mean}(b)$  ▷ Estimated centroid
19:     $C \leftarrow \text{mean}(b - \bar{c}) / \text{std}(b - \bar{c})$  ▷ Compute the Haralick C
20:     $Continuity \leftarrow \text{IsContinuous}(Region)$ 
21:    if  $C \geq S_{circularity}$  &  $Continuity == \text{True}$  then
22:       $\bar{r} \leftarrow \text{mean}(b - \bar{c})$  ▷ Compute the estimated radius
23:       $Spots \leftarrow \{\bar{r}, \bar{c}\}$  ▷ Append the spot attributes to the set of spots
24:    end if
25:  end for
26:  return  $Spots$ 
27: end procedure

```

---

## 4.3.2c Performance

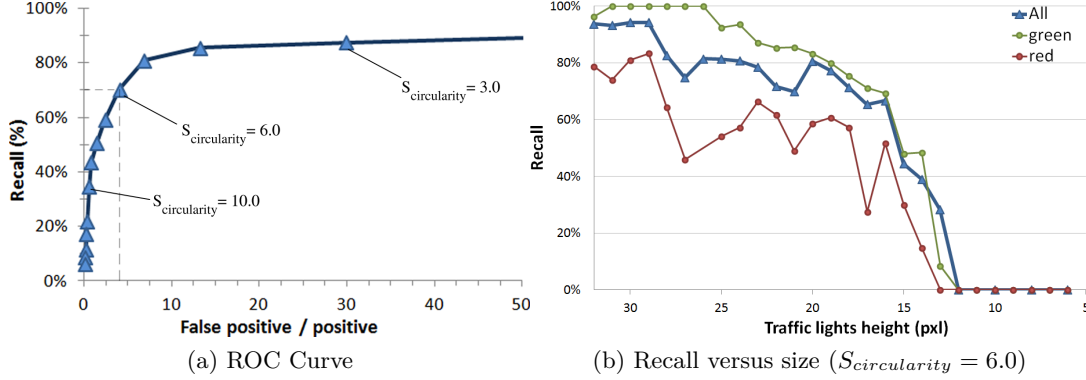


Figure 4.14: Performance of our spot light detection on two joint databases *dbFrance1\_8659* and *dbFrance2\_3574*. (a) is the ROC curve computed by varying the circularity threshold. The dashed line is the setting used in (b). With this setting, the recall is 70% and 4 false positives are output per true positive. (b) Recall versus size of the lights for all lights (red, green and amber) and for each red and green separately. The database has 12,233 occurrences of traffic lights (59 unique traffic lights). Notice that green lights are better detected by our system because they are imaged brighter. Amber lights are ignored since our database contains only lights close to the vehicle.

To evaluate the performance of the proposed spot light detection algorithm we have used two sequences ( $640 \times 480$ ) recorded in the city of Paris, France. There are 12,233 occurrences of traffic lights (5,690 green lights + 163 amber lights + 6,380 red lights). The complete details on this sequence are provided much later in table 4.1. Again, recall that the data and the camera are challenging. Given the artefacts caused by the sensor it is impossible to detect 100% of the lights.

As we aim here at measuring the performance of the spot light detection, we do not distinguish traffic lights from other “spot lights” (vehicle lights, bright spot, etc.). Our goal is to miss as few lights as possible regardless of the number of false detections. Hence, we use solely the recall defined as below:

$$\text{Recall} = \frac{\# \text{ of spots from traffic lights}}{\# \text{ of traffic lights}} \quad (4.8)$$

To decide whether a spot light belongs to a traffic light we verify that the hand labelled traffic lights entirely overlap spots output from our algorithm.

Figure 4.14 shows the performance of this stage. Figure 4.14a is the Receiver Operating Characteristic (ROC) curve which is computed by varying the circularity parameter ( $S_{circularity}$ ). Remarkably, our method can reach high recall despite the fact that we use only grayscale. The cost for a high recall is a large number of false positives. Hence, to get 70% of recall (that is with  $S_{circularity} = 6.0$ ) the output in average is five spots for one that do belong to a traffic light. Ideally, this stage is intended to be permissive rather than conservative. Indeed, if we mistakenly dismiss a light that actually belong to a traffic light it will be impossible to recognize the associated traffic light and the latter will be missed. However, when put in practice the right setting to use depends of the performance of the classifier.

In figure 4.14b, we exhibit the recall versus traffic lights size for a rather realistic scenario (that is with  $S_{circularity} = 6$ ). In spite of a global recall of 70% this shows that the recall is really high

for the bigger traffic lights. Red traffic lights have a lower recall due to the low intensity of the red color in both camera vision and human vision.

One may note that we have evaluated the output of this detection stage in one piece but a deeper analysis of its performance would be to evaluate the accuracy of the estimated centroids ( $\bar{c}$ ) and radii ( $r$ ) with some ground truth. However, hand labelling all the spot lights (i.e. not just the entire traffic lights) is a few weeks task that we did not do yet. Still, qualitative evaluation proves that the centroids are almost always accurate though the radius is quite frequently underestimated. Especially when traffic lights have a low light intensity.

#### 4.3.2d Influence of the parameters

Our spot light detection method uses 4 distinct parameters (i.e.  $S_{circularity}$ ,  $S_{variance}$ ,  $S_{SE}$  and  $S_{clamp}$ ).

We have discussed that  $S_{SE}$  is function of the largest spot to segment and should be fixed given the camera intrinsic/extrinsic parameters. The parameters  $S_{variance}$  and  $S_{circularity}$  have the same value in our sequences we have tested our algorithm on.  $S_{variance}$  drives the top hat segmentation and should be adjusted only if the source light to segment is more or less important. Despite the various designs the light decay is similar for all traffic lights so no adjustments should be required.

Paradoxically,  $S_{clamp}$  is the one parameter that was one not meant to be changed but that require adjustments. However, it prevents from detecting lights too dimmed. While testing we have noticed that in some countries the light intensity of the traffic signals is lower and thus requires to lower this threshold. As this parameter is intended only to speed up the process we have conducted experiments skipping the clamping step. Without clamping the recall is similar (about 1% higher) but the process is approximately 8 times slower.

## 4.4 Traffic Light Classification

Now that we have completed our spot light detection, we seek to verify that the segmented spots belong to traffic lights. Often in the literature traffic light classification is dismissed as Traffic Light Recognition (TLR) relies on the strong assumption that all colored lights belong to traffic lights. While such assumption may be true if the search is restricted to the sky only (i.e. detecting only over-the-road traffic lights) experiments show that when dealing with road-side traffic lights numerous elements pass through our spot light process while far from being actual lights. Some like neon-signs are tough to tell apart and will require temporal filtering but most of them (advertisements, vehicles lights, etc.) may be discriminated with classification.

### 4.4.1 Related work

As above mentioned a large number of TLR dismissed the classification assuming that any colored light above the road is an actual traffic light<sup>5</sup> [Hwang et al., 2006, Kim et al., 2007, Shen et al., 2009, Premachandra et al., 2009].

Obviously platforms that benefit of traffic lights pre-mapping and localization [Levinson et al., 2011, Fairfield and Urmson, 2011] also skip this stage as they intend to answer the question “What is the state of *this* traffic light ?” rather than “Is this region a traffic light ?”. Hence the autonomous Junior car [Levinson et al., 2011] uses a probabilistic histogram LUT that makes the

<sup>5</sup>Often a video-scheme or temporal filtering compensates the inevitable erroneous spontaneous detections.

traffic light region to fall in either category  $\{Green, Amber, Red\}$ . In [Fairfield and Urmson, 2011] it is assumed that the traffic light may be *off* or that inaccuracy produce incorrect ROI so they use another category *black* to discriminate this case. However, generally speaking the methods involved in [Levinson et al., 2011, Fairfield and Urmson, 2011] use logically the a priori that the region is already a traffic light making them unsuitable without pre-mapping or localization.

For the others, the techniques involved may be summarized in three categories we describe below: statical learning, template matching and structural-based.

#### 4.4.1a Statistical learning

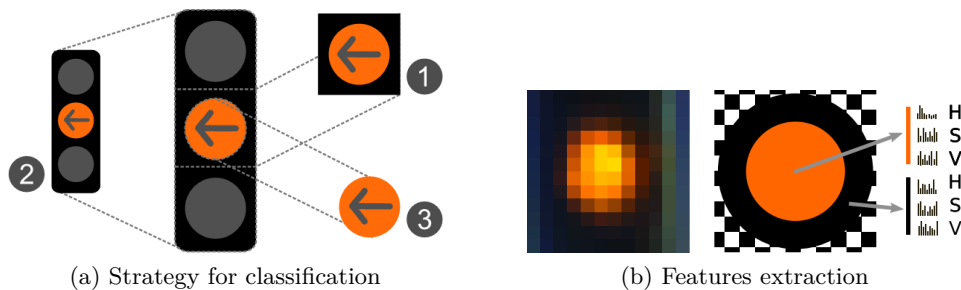


Figure 4.15: Strategy used for classification of traffic lights candidate in [Nienhüser et al., 2010]. (a) Depicts the 3 SVM classifiers trained where: 1) is the box that contain the active lamp, 2) is the classification of the whole traffic light region, and 3) classification of the active light shape. (b) Depicts the extraction of the HSV histogram of the active light and its direct vicinity. Source: [Nienhüser et al., 2010].

Chronologically, Neural Networks, Adaboost and Support Vector Machines (SVM) have been attempted to classify traffic lights. Hundreds or thousands of examples are then needed to train classifiers that identify traffic lights and their state at run time. Often classification is applied on the candidate regions extracted from a first color segmentation stage but [Lindner et al., 2004, Angin et al., 2010] have also used the classical sliding sub-window approach extensively described in the literature.

The papers that have attempted traffic light classification with statistical learning are listed below:

- **Neural Networks.** [Li and Tu, 2000, Gavrila et al., 2001, Lindner et al., 2004]
- **Adaboost.** [Lindner et al., 2004]
- **Support Vector Machine.** [Nienhüser et al., 2010]

The strategy commonly applied is a one-versus-all that is a classifier for each of the traffic light state. Usually a pre-color identification is used in order to select the right classifier as in [Franke et al., 1998, Lindner et al., 2004, Gong et al., 2010, Nienhüser et al., 2010] while in the third method from [Lindner et al., 2004] they iteratively test each candidate region with every possible classifier.

**Features** The challenge of classifying traffic lights lies in their undetailed appearance. The dark surrounding board of the bright light prevents from seeing its direct vicinity. In the absence of textures or significant gradients the use of corner features or histogram of gradients is clearly



prohibited. Instead contrast features are generally used (i.e. Haar) to classify the regions [Lindner et al., 2004, Angin et al., 2010, Gong et al., 2010]. Recently Nienhuser *et al.* proposed using the HSV histogram of both the active light and the board as two features of an SVM classifiers [Nienhüser et al., 2010], see figure 4.15. The novelty of their technique is the three stages classifier (see figure 4.15a) that iteratively tests the active light, the backboard and the whole traffic light. Though slightly different note that this is similar in spirit to our approach.

#### 4.4.1b Structural rules

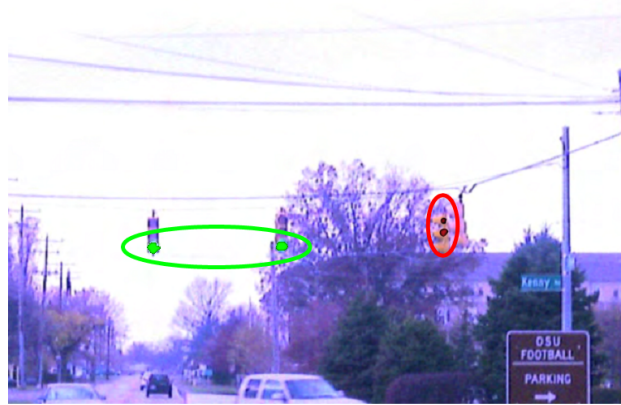


Figure 4.16: Structural rule to limit the validation of traffic lights. Shown is the output of Shen *et al.* where 4 traffic lights are detected but only the two greens horizontally aligned are validated (the two vertical red being mistakenly detected). Adapted from: [Shen et al., 2009].

So as to classify traffic lights [Shen et al., 2009, Yu et al., 2010] validate their detection if at least two traffic signals aligned horizontally have been detected (see figure 4.16). Needless to say that such strong assumption is not valid in many countries and generally works only for traffic lights *over-the-road*.

Another proposition – though unlikely to work – is to use the detection of the light to predict the position of the off lamp. [Omachi and Omachi, 2010] advocates that they can verify the hue of the lamp even when switched off by small variation in the center. Such assumption seems invalid to us as the color is often produced by a color bulb (impossible to detect when off) rather than by a color filter, not to mention that it seems doubtful that minimal hue variations may be detected accurately.

#### 4.4.1c Shape and template matching

In 2004, Lindner *et al.* have first proposed to use a coarse template matching that is to test the presence of a thick black box surrounding the circular spot [Lindner et al., 2004]. They benefit of the traffic lights design in Germany that requires to have a black backboard with a thick white border (this clearly enhance its visibility). Though their approach is somewhat trivial they claim surprisingly good performance (up to 85%) when used with stereo and GPS. This has been extended later by Xu to a global template matching approach [Xu et al., 2010]. [Xu et al., 2010] and [Li et al., 2011] have also extended the template matching approach but clearly based their work on our paper [de Charette and Nashashibi, 2009a]. Hence, we won't detail their experiments here.

### 4.4.2 Our approach: Template matching

From the previous stage (section 4.3) we have detected a set of spot lights  $\{(c_0, r_0), \dots, (c_n, r_n)\}$ . Now, to evaluate whether they do belong to traffic lights we study their direct vicinity. As we have disregarded color during spot light detection, classification plays here an important role. It intends not only to tell apart traffic lights from spurious regions but also to guess the current state of the traffic light. Hence, the outputs of our classification are  $\{BackGround, Green, Amber, Red\}$ . We denote *background* (or *bg*) any candidate that is not a traffic light (aka negative).

#### 4.4.2a Template matching

**Geometry** We define the geometric appearance of our templates using a top-down approach. In France traffic lights are made of 3 lights, one backboard and a pole that may be either horizontal or vertical. From French regulation and our own measurements we've defined the geometry of French traffic lights as in figure 4.17 :

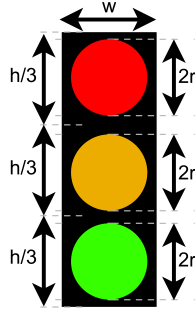


Figure 4.17: Geometrical template used for the classification of French traffic lights. The size of traffic lights being denoted  $w \times h$ . For French signals, the diameter of each lamp is approximately 80% of the sub-box that is:  $2r = 0.8 \times h/3$ .

Though traffic lights are physically made of one backboard we use three *virtual* sub-boxes of equal size. Each one contains a circular colored light that has a diameter that is 80% of the sub-box. We refer to the rectangle formed by the three sub-boxes as the *backboard*. Using the spots detected during the previous stage we generate Traffic Light candidates ( $TL_i$ ) with sizes as follows:

$$w(TL_i) = \frac{2r}{0.8} \quad h(TL_i) = 3 \times \frac{2r}{0.8} \quad (4.9)$$

Since the position of the *on* lamp changes regarding the traffic light states we can not create a single candidate that would be classified into one of the states. Instead, we have no choice than to create as many candidates as there are states (i.e. one-versus-all strategy). Hence, as it evaluates a state-candidate the template matcher behaves as a binary classifier with two outputs  $\{validated, rejected\}$ . This has consequences on the performance of our system as any falsely detected spot light during the first stage will have to be rejected by all the templates to be classified as *background*. In other terms it consequently increases the odds of misclassifying a spurious region in one of the traffic light state.

The full template-classifier process is depicted in figure 4.18 for French traffic lights. Note that the state of traffic lights follow a circular sequence: Green  $\Rightarrow$  Amber  $\Rightarrow$  Red  $\Rightarrow$  Green. We use this later on for post-processing.

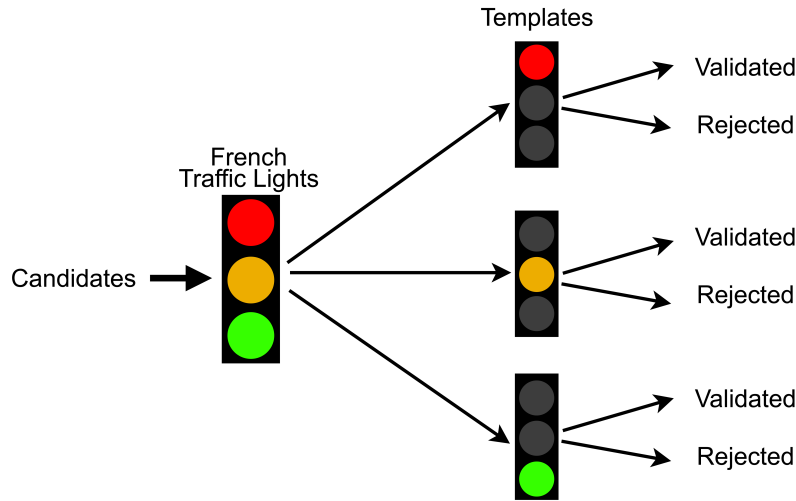


Figure 4.18: The three possible states of French traffic lights. As the *on* light position changes over time a unique template is required for each state.

**Classification** Now that we have generated the candidates we need to evaluate them. So far, the only property that has been verified is the circularity but to classify the candidate we need to verify the general appearance of the traffic light including the backboard. For each candidate the template is split in five parts that are the *on* light ( $L$ ), the backboard ( $BB$ ) and the three sub-boxes ( $SB_L$ ,  $SB_{B_1}$ ,  $SB_{B_2}$ ) as shown in figure 4.19. Each part is being evaluated separately and then a general verification is applied. Note that unlike some other traffic lights, French traffic lights have no thick border surrounding the backboard making them more complex to detect. Hence, when imaged against a dark background detecting the boundaries of the traffic lights is consequently impossible. The information to classify a template is then strictly restricted to the inner pixels of the candidate.

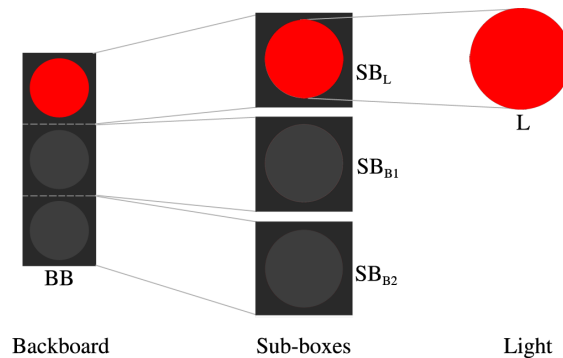


Figure 4.19: Geometrical decomposition of the French red traffic light template. To detect the traffic lights we use a bottom-up approach where each elements (backboard, sub-boxes, light) is evaluated independently.

Each part of the candidate (backboard, sub-boxes, light) is evaluated using fuzzy intensity-based constraints that form together the confidence index of the candidate. Our constraints

are called *operators* and use basic image processing techniques so as to quickly discriminate a candidate. The operators are defined as  $Op(f(X), min, max)$  where  $X$  is the region to evaluate ( $X \in \{BB, SB_L, SB_{B_1}, SB_{B_2}, L\}$ ). And  $f$  is a standard image processing function such as:  $Mean(X)$  (mean intensity of region  $X$ ),  $Entropy(X)$  (Shannon entropy of  $X$ ),  $Contrast(X, Y)$  (contrast between regions  $X$  and region  $Y$ ), etc. The output of each operator is a confidence index which is defined in a fuzzy way, with its maximum when  $f(X)$  lies between  $min$  and  $max$  ( $min$  and  $max$  are empirical settings). Hence:

$$Op(f(X), min, max) = \begin{cases} g(min - f(X)) & f(X) < min \\ 1 & min \leq f(X) \leq max \\ g(f(X) - max) & f(X) > max \end{cases} \quad (4.10)$$

where  $g(x)$  is a monotonically function which spread width is approximately half of  $|max - min|$ . The output function we use is a normal function, as in figure 4.20.

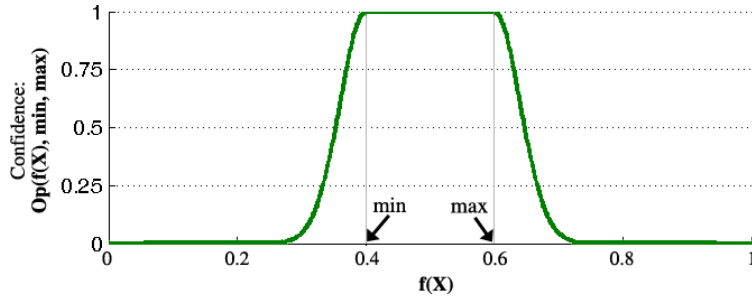


Figure 4.20: Fuzzy operator function  $Op(f(X), min, max)$  as a function of  $f(X)$ .

Generally speaking the traffic lights have a plain backboard which is said “dark” and should be highly contrasted with the light area. We verify that the candidate meet these requirements using a mean intensity check and an entropy verification. In the same way, we check that the sub-boxes of the off lamp are contrasted with the region of the *on light* (the detected spot light).

Therefore the confidence of a Traffic Light candidate is computed from the confidence indices of the backboard  $BB_{conf}$ , the sub-boxes  $SB_{conf}$  and the light  $L_{conf}$ , that is:

$$TL_{i_{conf}} = n_1 \times BB_{conf} + n_2 \times SB_{conf} + n_3 \times L_{conf} \quad (4.11)$$

$n_1$ ,  $n_2$  and  $n_3$  being normalization coefficients that respect the condition:  $n_1 + n_2 + n_3 = 1$ .

$$\text{with } \begin{cases} BB_{conf} & = Op(Mean(BB), 0, max) + Op(Entropy(BB), 0, max) \\ SB_{conf} & = Op(Contrast(SB_{B_1}, L), min, 1) + Op(Contrast(SB_{B_1}, SB_{B_2}), 0, max) \\ L_{conf} & = Op(Mean(L), min, 1) \end{cases} \quad (4.12)$$

The operators listed above are simple intensity image processes that behave as weak features of the whole classifier. With this initial classification about 87% of the spurious candidates are removed at very low processing cost ( $\approx 0.4ms$  per candidate).

To find the right traffic light status is more complex. There are two pieces of information that can be used to guess the status: the *on light* position and the *on light* color.

When traffic lights are imaged against the sky (i.e. traffic lights *over-the-road*) the *on light* position can easily be drawn from the boundaries of the backboard. However, when dealing

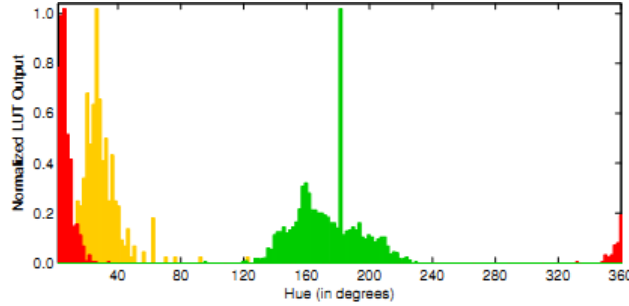


Figure 4.21: Hue histograms from Levinson *et al.*. Note that the red and amber hues overlap only partially allowing a probabilistic approach. Source: [Levinson et al., 2011].

with road-side traffic lights the latter might be imaged against dark building or trees that makes impossible to detect the backboard boundaries and thus the light position. In these cases we found that color information is needed to accurately guess the right signal status.

Hence, we use an additional light color operator ( $RedConf(X)$ ,  $GreenConf(X)$ ,  $AmberConf(X)$ ) in the template classifier. As an example, for the red template we turn the light confidence index computation to:

$$L_{conf} = Op(Mean(L), min, 1) + Op(RedConf(L), min, 1) \quad (4.13)$$

We now describe how to classify the light color when using a low dynamic sensor.

#### 4.4.2b Light color identification with low dynamic sensor

**Overview** The strategy generally used to classify the spot color is to verify the mean hue of the spot and apply a trivial thresholding. However, the literature reports that with low-dynamic cameras the hues of amber and red lights overlap. This has been mentioned as a common encountered issue with traffic light recognition [Hwang et al., 2006, Kim et al., 2007]. In these cases, amber lights are simply ignored.

An alternative proposal is to classify each pixel  $P_{x,y} \in L$  ( $L$  being the light region) in a probabilistic way into  $\{Green, Amber, Red\}$  regarding their hues. Then the light color can be guessed from the category having the highest number of pixels. This has been applied by two American teams [Shen et al., 2009, Levinson et al., 2011] – the latter using weighted histogram. It seems to work quite efficiently under the assumption that red-hues and amber-hues are only partially overlapped as for the Stanford team (see figure 4.21).

**Problem** Distinguishing green from red and amber is clearly not challenging and can be achieved from simple hue measures. However red and amber light identification depends on the color dynamic of the camera. Based on our sequences the aforementioned approaches (i.e. mean hue, probabilistic classification, etc.) are useless. Figure 4.22 shows the values of green, amber and red spot lights from a semi-automatic extraction process. The latter was achieved using our spot light detection algorithm (cf. section 4.3) with permissive settings but we kept only the spots that intersected the ground truth ( $\approx 10,000$  lights). The figure clearly shows that the hues covered by the red and the amber spot lights entirely overlap. When compared with figure 4.21 this exhibits that the aforementioned approaches will fail. There are two reasons for such an overlap: (a) The

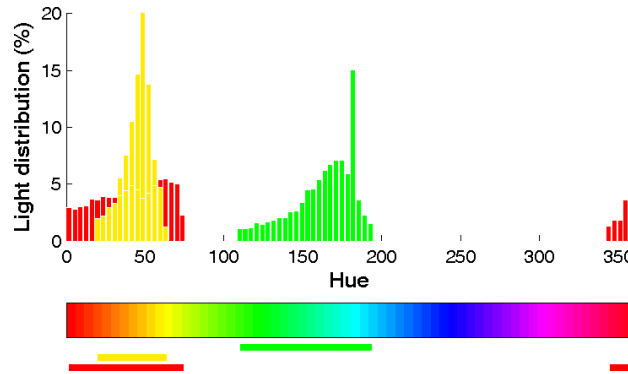


Figure 4.22: Our hues histograms of red, green and amber lights on our database *dbFrance1\_8659*. Note that our camera does not sense color accurately and that the area covered by each light is large. Even more important the red and amber hues fully overlap each other making impossible any probabilistic approach.

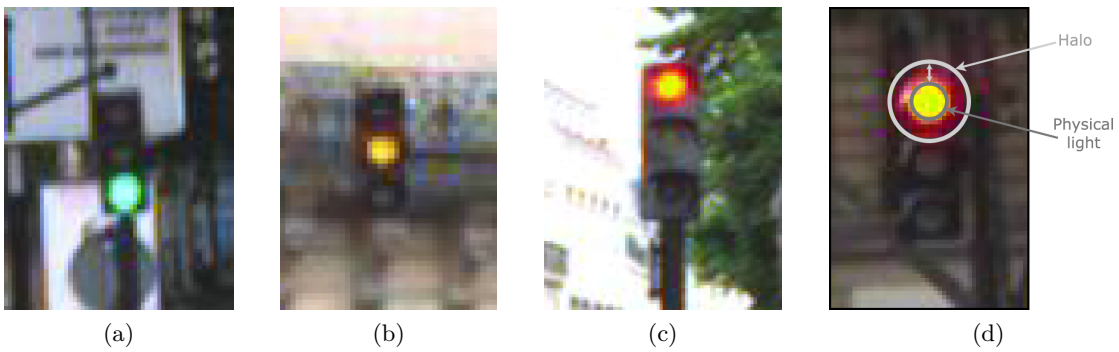


Figure 4.23: Illustration of the halo caused by the lights from the traffic signals. Here the camera artefacts are clearly visible making green traffic lights cyan or white (a) and red light over-saturating to yellow (c).

low dynamic (saturation) of our sensor (Marlin F-046C), (b) The lack of chromatic standard in France<sup>6</sup>. We assume that it is mostly due to saturation of our CCD sensor.

Studying closely the appearance of the three colored lights (figure 4.23) a glowing effect is visible. The latter being dramatically more important for lights with large wavelength (i.e. reddish). This glowing artefact exhibits in fact useful properties that we can use. First, it is monotonically decreasing with the distance from the spot. Second, it is isotropic given that the light is isotropic itself.

Hence, as the light intensity decreases, the over-saturation artefacts are dimmed and one may sense the true color of the traffic light. In other word, while we can not tell apart red from amber based on the inner spot color it is possible to do so measuring their surrounding halo. Such affirmation is supported by our measurements. Consider any spot light previously detected, we denote  $r$  its radius and  $\rho$  the distance to its centroid. In figure 4.22 we have shown the hue histograms for  $\rho \in [0; r]$ . Figure 4.24 shows the hue histogram for different distance intervals:

<sup>6</sup>The standard NF EN 12368 describes the appearance of traffic lights since 2006 though it is not actually applied.

$\rho \in [0; 0.5r]$ ,  $\rho \in [0.5r; r]$ ,  $\rho \in [r; 1.5r]$  and  $\rho \in [r; 2r]$ . As we expected the red Gaussian-like hue distribution slightly shifts from yellowish to reddish<sup>7</sup> when farther from the centroid.

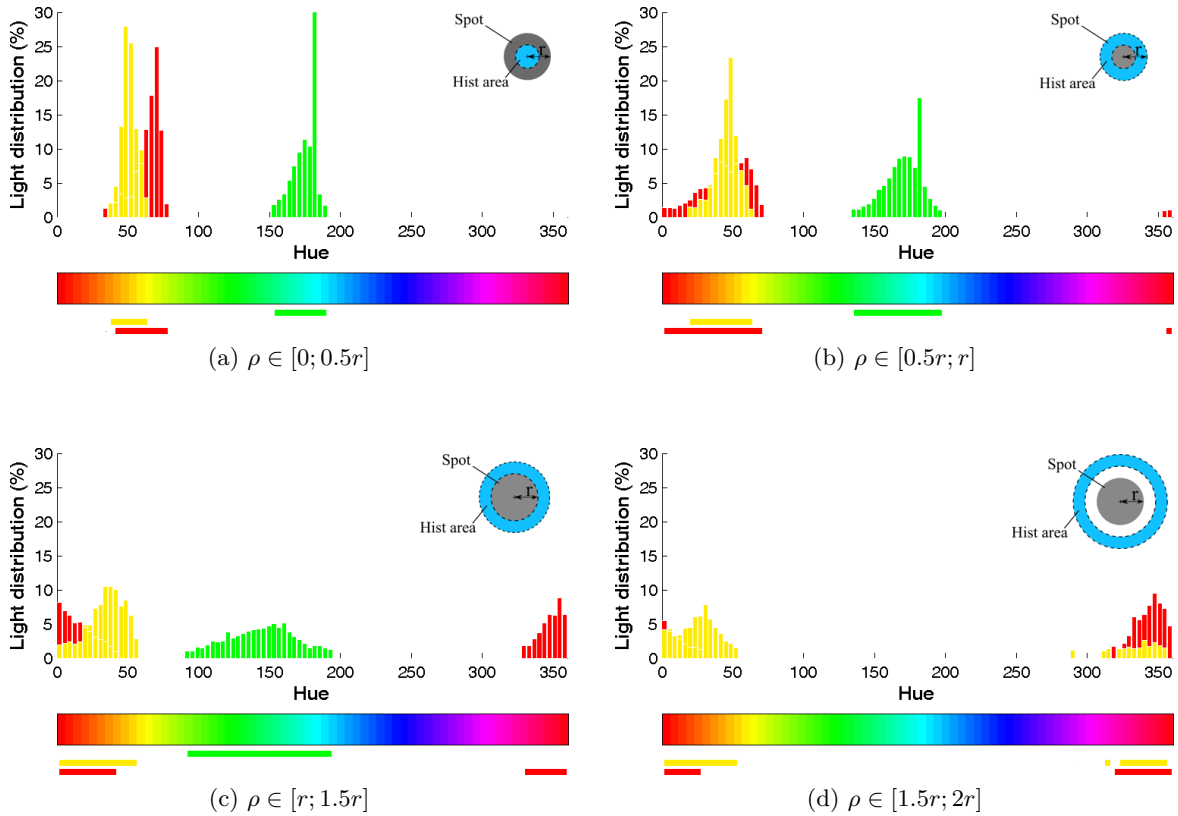


Figure 4.24: Histograms of pixel distributions for red/green/amber light according to the distance from the center of the spot light detected from semi-automatic extraction of more than 10,000 lights of traffic lights.  $r$  being its radius and  $\rho$  the distance to the centroid. Note the red Gaussian-like distribution that shifts to a pure red as  $\rho$  increases. Remarkably this shows that red may be distinguished from amber when  $\rho \geq r$ . Shown are the bins that are at least 1% of each light distribution. Green light is omitted in (d) because the halo of green light is not large enough.

**Halo estimation** The light intensity of a spot behaves as a centred flattened Gaussian  $G(\sigma, s)$  where  $\sigma$  controls the light spreading hereafter called halo and  $s$  is the radius of the flattened part. From the spot light detection stage we know the position of each spot and use their estimated centroids as the centres of the light sources. Assuming that we know the atmospheric conditions and the intensity of the light a physics-based model would certainly ideally estimate the halo of each light. However, the intensity may vary from traffic light to traffic light and the atmospheric conditions are unknown. Instead, we use a heuristic based on the isotropy and decay properties of the spot light emission.

<sup>7</sup>Recall that yellow carries more luminosity than red. Hence, a saturated red light appears yellow at the center.

For each spot light detected during the prior stage we do the following process:

- **Step 1.** Consider a polar space  $(\rho, \theta)$  with its origin  $(0; 0)$  the center of the current spot.
- **Step 2.** Extract the pixels with coordinates  $\rho \in ]0; r]$  ( $r$  the estimated radius).
  - Set  $r_{min} = r$ .
- **Step 3.** Extract pixels within the ring  $\rho \in ]r_{min}; r_{min} + r/3]$ .
  - Verify that the saturation is monotonically decreasing with  $\rho$  and above  $H_{minsat} \times \text{SpotSaturation}$ . If not, stop and set  $r_{halo} = r_{min}$ .
  - Verify that the intensity is monotonically decreasing with  $\rho$ . If not, stop and set  $r_{halo} = r_{min}$ .
  - Verify that the shifting hue direction is identical to the previous direction. If not, stop and set  $r_{halo} = r_{min}$ .
  - Set  $r_{min} = r_{min} + r/3$  and go to Step 3.

In keeping with the light property emission this heuristics successfully detects the halo surrounding a spot light with a relative precision of  $r/3$ . This discretization is intended to diminished the noise artefacts but it plays no role in the process.

No quantitative performance can be provided since we did not use any ground truth. However some examples of the halo estimation are shown in figure 4.25. From qualitative results the precision of our heuristic is sufficient for our application. The same parameters were used regardless of which sequences we applied our algorithm on showing that our algorithm is robust enough. We use  $H_{minsat} = 0.3$ , i.e. the eccentric expansion continues until the saturation of the new ring is less or equal to one third of the inner spot saturation.

While such process would definitely benefit of better physics embodiment it is sufficient to estimate the halo as we intended to. However note that for such heuristics to be valid the two following assumptions need to be true: (a) The center of the detected spot *is* the center of the light emission, (b) The direct vicinity of the light must have different hue than the light. From our experiments these two assumptions are fairly acceptable.

**Halo estimation for color disambiguation** The likelihood of each light to be either a *green*, *amber* or *red* light is measured based on the distance of the *spot hue* and the *halo hue* to those of experimental measures. We denote  $LightGreen(L)$ ,  $RedConf(L)$  and  $AmberConf(L)$  the confidence of a region  $L$  (extracted with the spot light detection algorithm) to be either *green*, *red* or *amber*. They are computed as follow:

$$GreenConf(L) = 1 - k \times \sqrt{2(\text{hue}(L_{spot}) - \text{hue}(G_{spot}))^2} \quad (4.14)$$

$$AmberConf(L) = 1 - k \times \sqrt{(\text{hue}(L_{spot}) - \text{hue}(A_{spot}))^2 + (\text{hue}(L_{halo}) - \text{hue}(A_{halo}))^2} \quad (4.15)$$

$$RedConf(L) = 1 - k \times \sqrt{(\text{hue}(L_{spot}) - \text{hue}(R_{spot}))^2 + (\text{hue}(L_{halo}) - \text{hue}(R_{halo}))^2} \quad (4.16)$$

where  $k$  is a constant normalization coefficient.  $G_{spot}$ ,  $A_{spot}$ ,  $R_{spot}$  the empirically measured hue of the inner spot and  $A_{halo}$  and  $R_{halo}$  for the halo.  $L_{spot}$  is the inner spot of the light and  $L_{halo}$  the halo ring. Note that green lights did not produce any halo in our experiments so we just ignored this term.



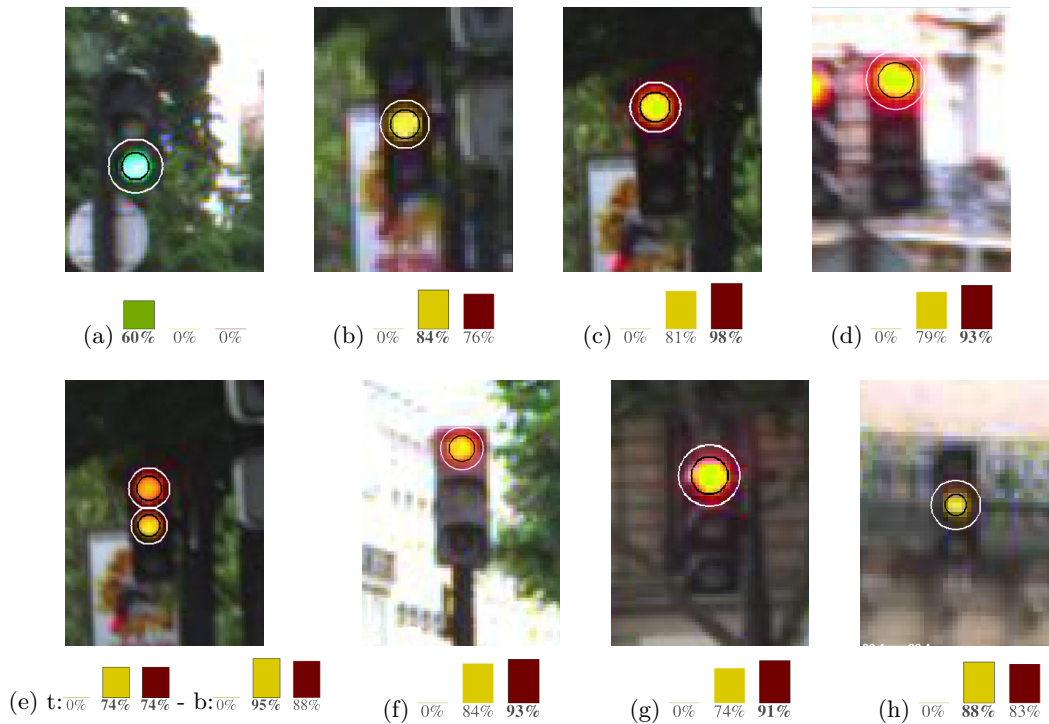



Figure 4.25: Estimation of the halos of the traffic lights. Both spot radius (black circle) and halo radius (white circle) are displayed. For each sample the color score is displayed below. Notice that the highest score correspond to the actual color in all samples. In (e) however, the score of the top light is similar for amber and red. The reason is that the amber and red light mix and that the red light just switched on so its intensity did not yet reach its highest value.

Confidence indices of each light to belong to either category are shown as bar histograms in legends of figure 4.25. From these charts, red and amber disambiguation perform well as expected. However 4.25e the top lamp obtained the same score for red and amber but note that the lamp did not yet reached its full intensity.

For the template matching classification, we make use of  $GreenConf(X)$ ,  $AmberConf(X)$ ,  $RedConf(X)$  to compute the light confidence of the *ad-hoc* templates (cf. equation 4.13).



		Classifier				Recall
		Bg	Green	Amber	Red	
Ground truth	Bg	141,515	57	493	321	99.39%
	Green	1,127	1,813	0	0	61.67%
	Amber	14	0	48	20	58.54%
	Red	518	0	54	2,988	83.93%
Precision		98.84%	96.95%	8.07%	89.76%	98.25%

(a) Sample frame

(b) Confusion matrix

Figure 4.26: Performance of our classifier when evaluated over the whole picture. (a) is a sample frame. (b) is the confusion matrix of our traffic light recognition. Note that the classification rate is 98.25% though a large number of false positives occur.

#### 4.4.2c Performance

To evaluate the performance of our template matching classification we have run our algorithm on two sequences that are *dbFrance1\_8659* and *dbFrance2\_3574* with respectively 8,659 and 3,574 occurrences of traffic lights in 21,269 frames. The two sequences were recorded with a low-dynamic camera Marlin F-046C ( $640 \times 480$  pixels, @25FPS) in the dense city heart of Paris, France (more details in table 4.1). Generation of the candidates follows the principle described previously that is: we generate three candidates for green, amber and red traffic lights for each region detected by our spot light detection. Prior to any template matching a normalization of the candidate region intensity is applied for illumination invariance. If two candidates are validated with the same spot light (as often the case for red and amber status) only the candidate with the best matching confidence is kept.

The confusion matrix of our classifier is shown in table 4.26b with the two joint sequences. The classification rate is 98.25%. Given that more than 148,968 candidates were evaluated about 871 spurious regions have been incorrectly classified as traffic lights. However, note that the two sequences used are doubtless more challenging than most of the sequences used in the literature of traffic light recognition. The neon-signs, the vehicle lights and the light sources are frequent in these sequences though hard to distinguish from actual traffic lights. Not to mention that the picture quality is not good.

One should note that 100% accuracy of the classification is *impossible* given our challenging sequences. Indeed, our ground truth includes only real *physical* traffic lights. Due to reflections of the traffic signals in several shopwindows and other vehicle windows, even a perfect visual classifier would detect these *look-alike elements* as “false positives” without the help of additional sensors (GPS, GIS, etc.). More importantly the use of our system to assist the driver requires to detect traffic lights when far. In table 4.26b the performance are indicated for candidates having a width of 8 pixels minimum (height  $\approx 18$  pixels), that is about 65m far with our camera settings.

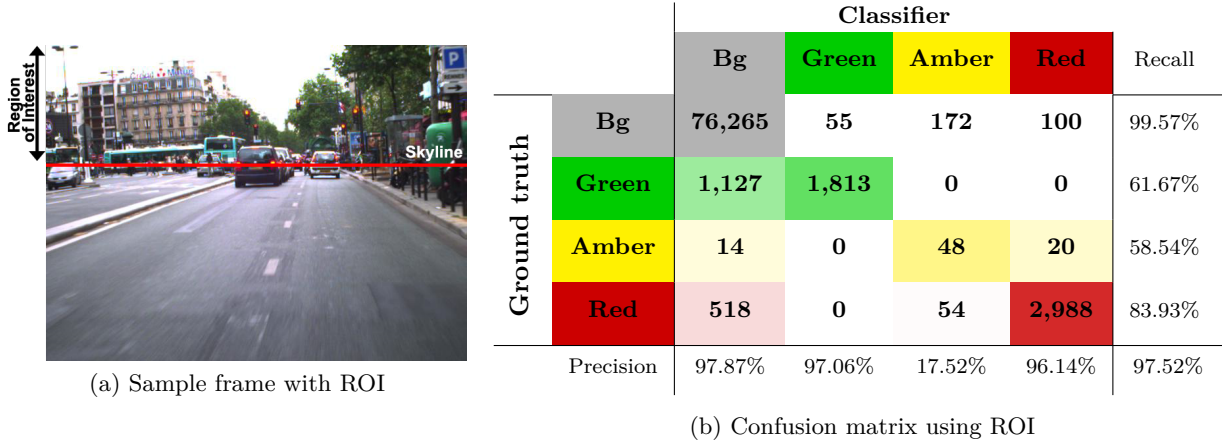


Figure 4.27: Confusion matrix of our traffic light recognition system when using ROI over the skyline. (a) a sample frame. (b) The matrix. Comparison with the evaluation on the whole frame (table 4.26b) shows that the number of false positives (first line) is lower while the true positives are unchanged. A perfect system would have zero values everywhere but in the diagonal.

From our experiments there are two main sources of false positives (i.e. spurious regions misclassified as traffic lights):

- Sun rays going through tree leaves randomly produce green spot lights that may be confused with green traffic lights for one or two frames. This is a commonly encountered issue when streets are lined by trees and it has been reported also in [Lindner et al., 2004] or [Fairfield and Urmson, 2011].
- Vehicles lights (brake lights or direction indicators) are sources of falsely detected red and amber traffic lights. Again this is known as a common issue in the literature.

An easy way to get rid of some of the false positives while *artificially* improving the performance of our classifier is to reduce the Region Of Interest (ROI) to credible positions of the traffic lights. This may be done by computing the coordinates of the traffic lights in the world but regarding our experiments this achieves similar performance than when using a ROI that is above the skyline. Performance with ROI are shown in table 4.27b (classification rate 97.52%). As anticipated a large number of false positives has been removed ( $\approx 62\%$ ) – essentially those related to the vehicles lights. One may note that the number of false positives is still high despite the limited ROI ( $\approx 330$  false traffic lights). To understand the reason we have plotted in figure 4.28 the false positives given the frame index for green, amber and red lights. This plot shows that most of the false positives result of two elements: (a) The amber blinking light of a traffic sign mounted on a garbage truck that is stopped right before our car ( $\rightarrow 137$  false positives as amber). (b) The net reflection of a nearby red traffic lights in a large bus window ( $\rightarrow 83$  false positives as red).

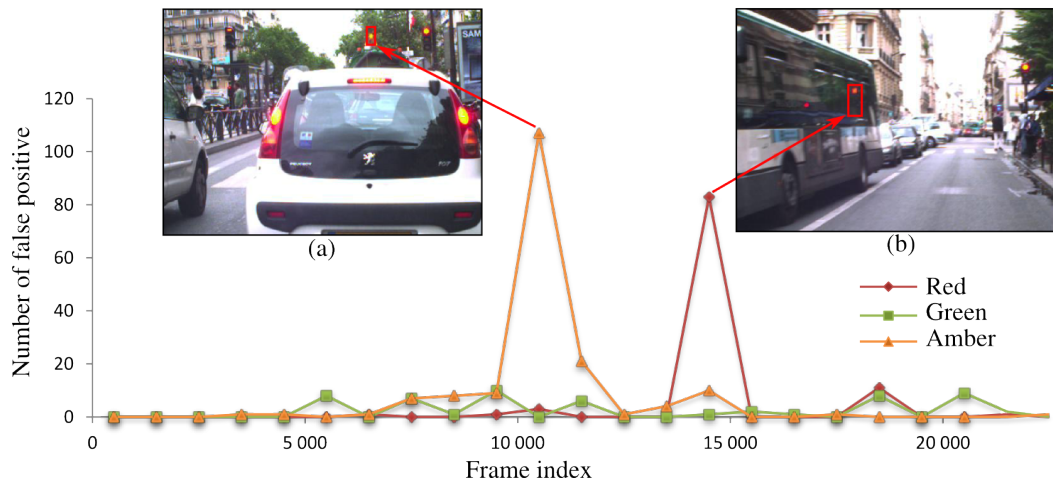


Figure 4.28: False positives versus frame index of the joint sequences *dbFrance1\_8659* and *dbFrance2\_3574*. As may be seen, most of the false positives are related to: (a) the light from a garbage truck ( $\rightarrow$  137 false positives), (b) a reflection of an actual traffic light ( $\rightarrow$  83 false positives). Note that each marker point cover the sum of false positives for 1,000 frames. This explains why the amber peak is lower than 137.

## 4.5 Performance of our Traffic Light Recognition

In the current section we describe the performance of an ADAS using our complete chain of traffic lights recognition with the two previously described stages that are the spot light detection and the template matching. Most importantly is the accuracy (precision) of the entire system but distance of detection is also crucial for an ADAS. Hence, it should be able to alert the driver when sufficiently far away for him to slow down. Among others this section aims at answering the following questions:

- What is the working range of our system ?
- Can our system detect traffic lights far enough to alert the driver in time ?
- How many false alarms may occur in real scenarios ?
- Does our system perform equally with green, amber and red lights ?
- To what extent can we compare the current system to other TLR ?
- Can our be used to recognize traffic lights from other countries ?

As a first step we describe in section 4.5.1 the extension of our traffic light recognition algorithm to take into account temporal information. To that point we have only considered detecting traffic lights in still images but the latter are imaged on multiple frames and identification of the same traffic light in successive frames is a prerequisite. A tracker and a temporal filter are used to that aim. Not only does it allow us to identify traffic lights over time but it also provides a way to remove spontaneous false detections and thus increases the global accuracy of our system.

We detail the performance of our system for different sequences and cameras in section 4.5.2. As our approach makes no use of any training stage it is easily adjustable to work for traffic lights

of other countries. We demonstrate such ability with sequences from France, Switzerland and China. Discussions about the adjustments made are provided.

### 4.5.1 Temporal filtering

From section 4.3 and 4.4 we have detected traffic lights in still images and classified their state. Applying this process on multiples frames we get a set of observations  $\{(t_0, p_0, s_0, a_0), \dots, (t_n, p_n, s_n, a_n)\}$  where  $p$  and  $s$  are centroid's position and size, and  $a$  is the status (the output of the classification).  $t$  is the timestamp provided by our vision sensor. From this set of observations we wish to group the observations that result of a single traffic light being imaged on several frames. It will allow us to predict its next position in order to refine the output of our recognition chain.

We are first looking at the tracking and prediction and focus later on the temporal clues we may use to rule out incorrect outputs. There are a number of ways to track single elements in multiple frames. A classical approach is to measure the visual likeness of two candidates having extracted keypoints (e.g. SIFT, SURF, Harris). However, recall that traffic lights are poorly detailed and most importantly that their state is changing over time meaning that two occurrences of the same traffic light may not look alike. Instead, we use a geometrical approach to track and then predict the positions of the traffic lights. Note that the traffic lights trajectory in the image space depends solely on the vehicle motion.

#### 4.5.1a Geometrical tracking

To simplify the problem of tracking and prediction we make two assumptions: (a) the trajectory of the vehicle is straight and with a constant velocity when observed over a small time period, (b) the camera projection's model can be approximated with the pinhole model (i.e. lens distortion is neglected). Though strong, we found that these two assumptions are valid in all tested scenarios while dramatically simplifying the tracking and prediction. Figure 4.29a depicts the pinhole tracker. Note that this simple tracker has been used previously in [Miura et al., 2000].

**Initialization** To first pair observations without any prior knowledge and without using visual appearance only position and size can be used. Let's denote  $R_1(t_1, p_1, s_1)$  and  $R_2(t_2, p_2, s_2)$  the corresponding regions of two validated candidates at times  $t_1$  and  $t_2$ , respectively (with  $t_1 < t_2$ ). To define whether these two regions needs to be paired and matched, we verify whether they overlap sufficiently:

$$\frac{R_1 \cap R_2}{R_1 \cup R_2} \geq E_{overlap} \quad (4.17)$$

If the condition is violated the two regions are two different objects so  $T_1\{R_1\}$  and  $T_2\{R_2\}$ . Otherwise, the two regions are paired as being two occurrences of the same object we denote  $T_1$ . Now we have  $T_1\{R_1, R_2\}$ .

**Prediction** Having identified at least two occurrences of a traffic light we use the estimation of its future location to improve the robustness of the tracker as well as to compensate drop outs from the classification stage. We estimate the future position  $\hat{p}_3$  and size  $\hat{s}_3$  at a given time  $t_3$  from the pinhole model and the two last observations  $R_1(t_1, p_1, s_1)$  and  $R_2(t_2, p_2, s_2)$ . Derivation of the model using Thalès theorem shows that the positions  $p_1$ ,  $p_2$  and  $p_3$  are independent of the

vehicle speed when speed is assumed to be constant during the time interval  $[t_1; t_3]$  [Miura et al., 2000]. Hence, we estimate the future position and size as follows<sup>8</sup> :

$$\hat{p}_3 \begin{cases} x(\hat{p}_3) &= \frac{x(p_1) \times x(p_2)}{(1+\alpha) \times x(p_2) - \alpha \times x(p_1)} \\ y(\hat{p}_3) &= \frac{y(p_1) \times y(p_2)}{(1+\alpha) \times y(p_2) - \alpha \times y(p_1)} \end{cases} \quad (4.18)$$

$$\hat{s}_3 \begin{cases} w(\hat{s}_3) &= \frac{w(s_1) \times w(s_2)}{(1+\alpha) \times w(s_2) - \alpha \times w(s_1)} \\ h(\hat{s}_3) &= \frac{h(s_1) \times h(s_2)}{(1+\alpha) \times h(s_2) - \alpha \times h(s_1)} \end{cases} \quad (4.19)$$

with  $\alpha = \frac{t_3 - t_2}{t_2 - t_1}$ . Similarly to the initialization, the overlap of the classifier outputs and the predicted region is computed for each frame (as in equation 4.17). Again, the output with the greatest overlap (greater or equal to  $E_{overlap}$ ) is associated to the tracked object.

An example of this tracker output is shown in figure 4.29b. It was found to be robust enough to deal with all the scenarios we had to face. Since the prediction is computed over 6 frames (i.e.  $\approx 250ms$  @25FPS) the assumptions remain acceptable.



Figure 4.29: Tracking using pinhole approximation. (a) scheme of the pinhole tracker. (b) example of traffic lights tracked. The dots indicate the previous positions of the traffic lights. Note, that if no pitch occurred the history points should be perfectly linear.

One remarkable case of failure occurs if the pitch changes *severely* during this time period (e.g. sudden braking or accelerating). This is the limitation of a geometrical tracker since it can not predict the shift. Hence prediction does not fit the observation. An interesting alternative proposed by Gong *et al.* uses a CAMSHIFT algorithm<sup>9</sup> to track the spot light itself. However while it solves this specific issue, a major drawback is that it can not track traffic lights when changing status (e.g. switching from green to amber).

<sup>8</sup>The full derivation of the pinhole model is described in [Bargeton, 2009] for tracking and prediction.

<sup>9</sup>The Continuously Adaptive Mean SHIFT (CAMSHIFT) algorithm tracks colored blobs efficiently through an iterative process converging to the best and closest solution [Bradski, 1998].

**Validation** With this tracker we have associated the observations that geometrically match, hence a traffic light is *temporally* defined as a set of observations:  $T\{(t_1, p_1, s_1, a_1, c_1), \dots, (t_n, p_n, s_n, a_n, c_n)\}$  where  $a$  is the validated status and  $c$  is the confidence ( $a$  and  $c$  are outputs of the classification stage). A temporal traffic light is thus validated if:

- (a) It has at least 5 observations ( $n \geq 5$ ).
- (b) Its confidence  $T_{conf}$  is sufficiently high ( $T_{conf} \geq E_{thresh}$ ) with  $T_{conf} = \sum_{i=n-5}^n \frac{c_i}{5}$ .
- (c) The latest observation was less than 250ms ago ( $CurrentTime - t_n < 250ms$ ).

This hysteresis is sufficient to rule out spontaneous false alarm due to the 5 frames delay ( $\approx 200ms @25FPS$ ).

### 4.5.1b Light sequence

We are now aware of unique traffic lights  $T$  being tracked over time and each traffic light is a collection of observations as defined in the above section. We write the *light sequence* of  $T$  as a succession of states. Recall that the state is  $a \in \{G, A, R\}$  (respectively, Green, Amber or Red). A perfect sequence would be:  $G_1G_2G_3A_4A_5A_6R_7R_8R_9$ . In reality, as the states are outputs of the previous stages any misclassification would result in unstable output. This occurs when dealing with red and amber lights (cf. confusion matrices table 4.27b).

Say, we are tracking a green traffic light that switched to amber. A possible light sequence in such a scenario is:

$$G_1G_2G_3\mathbf{R}_4A_5A_6\mathbf{R}_7A_8A_9 \tag{4.20}$$

The two states  $R_4$  and  $R_7$  are obvious misclassifications that are dangerous for an ADAS as it would provide an incorrect warning to the driver. Indeed, since the light sequence is known *a priori* and is an absolute certainty it is easy to guess that the correct outputs should have been:

$$G_1G_2G_3\mathbf{A}_4A_5A_6\mathbf{A}_7A_8A_9 \tag{4.21}$$

Feeding our system with the light sequence we compensate most of the inaccuracies of our classification. The scheme used invalidates any abnormal output. An example of a sequence is depicted in figure 4.30. Notice that the incorrect output of the classification is improved by our light sequence scheme acting as a model-fitting process. Hence, the output of our system is more stable.

Classification	G	G	G	G	R	A	-	A
	↓	↓	↓	↓	↓	↓	↓	↓
Tracker+Sequence	G <sub>1</sub>	G <sub>1</sub> G <sub>2</sub>	G <sub>1</sub> G <sub>2</sub> G <sub>3</sub>	G <sub>1</sub> G <sub>2</sub> G <sub>3</sub> G <sub>4</sub>	G <sub>1</sub> G <sub>2</sub> G <sub>3</sub> G <sub>4</sub> A <sub>5</sub>	G <sub>1</sub> G <sub>2</sub> G <sub>3</sub> G <sub>4</sub> A <sub>5</sub> A <sub>6</sub>	G <sub>1</sub> G <sub>2</sub> G <sub>3</sub> G <sub>4</sub> A <sub>5</sub> A <sub>6</sub> A <sub>7</sub>	G <sub>1</sub> G <sub>2</sub> G <sub>3</sub> G <sub>4</sub> A <sub>5</sub> A <sub>6</sub> A <sub>7</sub> A <sub>8</sub>
	↓	↓	↓	↓	↓	↓	↓	↓
System output	-	-	<b>G</b>	<b>G</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>

Figure 4.30: A sample of the light sequence filtering on the output of the classification. Here, note that the 5th and 7th classification are incorrect (i.e. green to red is impossible) but the tracker and the sequence scheme filter adequately so that the system output is correct.

**Discussion** It is interesting to note that apart from a German team no one has discussed the problem of the unstable light sequence output. Still, discussions with other researchers as well as live demo have highlighted that it is a common issue. The strategy proposed by the German authors in [Nienhüser et al., 2010] is doubtless better in the general sense to model the visual state of traffic lights. We show in figure 4.31, the Hidden Markov Model (HMM) they use to estimate the probability of switching from one state to another while considering misclassification as hidden states of the model. The elegant idea they have proposed would allow mapping any type of light sequence to a HMM (e.g. blinking lights).

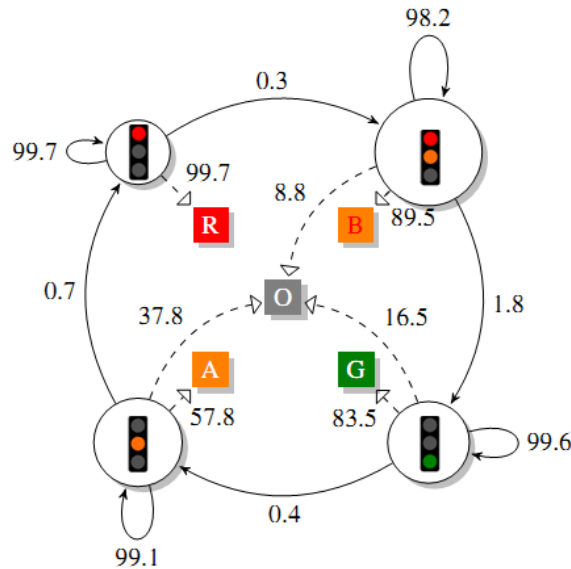


Figure 4.31: Hidden Markov Model for German traffic lights used in [Nienhüser et al., 2010]. Note that the traffic lights modelled here has 4 states. Spurious detections are modelled as hidden states. This idea is elegant and more efficient than ours. Adapted from: [Nienhüser et al., 2010].

## 4.5.2 Performance

We have tested our system on sequences in dense urban environments in (Paris) and also in Switzerland (Lausanne) or China (Shanghai). Sequences are  $640 \times 480$  (France and China),  $752 \times 480$  (Switzerland) and last from 8 to 12 minutes each. All were recorded in daytime urban environment but with different cameras and under different conditions. Sequences details are provided in table 4.1 and sample images in figure 4.32. Ground truth is obtained by hand-labelling traffic lights.

Performance is reported in table 4.2: frame per frame evaluation (table 4.2a) and temporal evaluation (table 4.2b), and some sample outputs are shown in figure 4.34. For an ADAS one may focus on the temporal performance which agrees with the output of an assistance system. For either evaluations we aim at maximising both the precision ( $P$ ) and the recall ( $R$ ) that are computed as follows:

$$P = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (4.22)$$

$$R = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (4.23)$$



(a) *dbFrance1\_8659* ( $640 \times 480$ , @25FPS)(b) *dbFrance2\_3574* ( $640 \times 480$ , @25FPS)(c) *dbChina1\_313* ( $640 \times 480$ , @11FPS)(d) *dbSwitzerland1\_5183* ( $752 \times 480$ , @25FPS)

Figure 4.32: Databases used for the evaluation of our traffic light recognition.

	Duration	Camera	Traffic Lights			
			# occurrences (# unique)			
			Green	Amber	Red	TOTAL
dbFrance1_8659	8min 48s 11, 179 frames	Marlin F-046C 640 × 480, @30FPS	3, 191(22)	58(2)	5, 410 (12)	8, 659 (36)
dbFrance2_3574	7min 07s 10, 090 frames	Marlin F-046C 640 × 480, @30FPS	2, 499(12)	105(2)	970 (9)	3, 574 (23)
dbSwitzerland1_5183	10min 40s 16, 009 frames	n/a 752 × 480, @25FPS	504 (6)	0 (0)	4, 679 (40)	5, 183 (46)
dbChina1_1657	12min 07s 9, 921 frames	Kodak Z7950 640 × 480, @11FPS	299 (10)	0 (0)	14 (1)	313 (11)

Table 4.1: Sequences used for the evaluation of the traffic light recognition system. The count includes any traffic light which minimum axis (width or height) is larger than 6 pixels. As an example, a traffic light imaged for 100 frames equals 100 occurrences, the notation is: 100 (1).

Generally speaking the performance our system exhibits is satisfying as in all sequences, the “temporal recall” is above 90% in France (and above 71% for all countries, cf. table 4.2b) which means that in France less than one traffic light over ten is missed. Often, missed traffic lights have low light intensity (i.e. spot light not detected) or do not remain visible long enough (temporal filter eliminate them) as when the car is turning. The precision is also satisfying but there is still room for improvement. Most importantly the data were recorded in dense urban environment where neon-signs, street lights and advertisements may possibly confuse our system. Some of the false alarms remain despite the temporal filtering. The latter are mostly due to other colored spot lights (neons, vehicles lights). Precision on all sequences is above 86%.

The temporal output of our system is perfectly stable (i.e. no more red/amber confusion) thanks to the halo estimation and the light sequence filtering.

#### 4.5.2a France

The two French sequences were recorded with our prototype vehicle *LaRA* and a camera Marlin F-046C mounted behind the interior rear mirror. Both sequences are publicly available with ground truth<sup>10</sup>. The challenge of these sequences is the dense environment where neon-signs are frequent and hard to distinguish from actual traffic lights. Additionally, the sensor is a low dynamic one that tends to saturate. Here our idea of the halo estimation is critical.

Concerning the light color identification, frame per frame evaluation shows our system confuses about 29.7% of the amber lights and 1.2% of the red lights (cf. table 4.2a). Since our database has much more red lights than amber lights (47 times more), the whole confusion rate is below 2%. This shows the efficiency of our color disambiguation scheme in the context of a low-dynamic sensor.

From table 4.2b the temporal evaluation exhibits that at least 93% of the traffic lights are detected in the French sequences. The few undetected traffic lights result usually of unstable detection due to partial occlusion, motion blur, or incorrect detection. Also, despite the tracking and filtering some errors occur. The latter are generally related to blinking lights or traffic lights reflection in shop windows / buses.

**Distance estimation** Now we investigate whether our system detects traffic lights far enough from the vehicle to alert the driver in time.

<sup>10</sup><http://www.lara.prd.fr/benchmarks/trafficlightsrecognition>

	Green	Amber	Red	All
<i>dbFrance1_8659</i>	98.6% (59.4%)	10.8% (55.3%)	99.0% (61.2%)	<b>95.9%</b> ( <b>60.8%</b> )
<i>dbFrance2_3574</i>	97.2% (35.2%)	77.6% (36.2%)	81.4% (50.9%)	<b>89.9%</b> ( <b>40.2%</b> )
<i>dbSwitzerland1_5183</i>	100% (23.2%)		93.2% (48.1%)	<b>93.5%</b> ( <b>45.7%</b> )
<i>dbChina1_313</i>	100% (44.8%)	fp=11 (fn=0)*	fp=7 (fn=14)*	<b>88.2%</b> ( <b>44.8%</b> )

(a) Frame per frame ( $C_{thresh} \geq 0.97$ )

	Green	Amber	Red	All
<i>dbFrance1_8659</i>	94.4% (94.1%)	33.3% (100%)	100% (100%)	<b>90.3%</b> ( <b>96.6%</b> )
<i>dbFrance2_3574</i>	88.9% (88.9%)	100% (100%)	80.0% (100%)	<b>88.2%</b> ( <b>93.8%</b> )
<i>dbSwitzerland1_5183</i>	100% (100%)		84.4% (67.5%)	<b>86.8%</b> ( <b>71.7%</b> )
<i>dbChina1_313</i>	100% (80.0%)	fp=1 (fn=0)*	fp=0 (fn=1)*	<b>88.9%</b> ( <b>72.7%</b> )

(b) Temporal ( $T_{conf} \geq 0.95$ )

Table 4.2: Performance of the traffic light recognition on sequences from France, Switzerland and China. The performance is shown as: precision (recall), computed from ground truth which minimum axis (width or height) is larger than 6 pixels. (a) Performance on still images ( $Detection \Rightarrow Classification$ ). (b) Performance using the temporal filtering described in section 4.5.1 ( $Detection \Rightarrow Classification \Rightarrow Temporal$ ). [The star \* indicates that zero true detection occurred so we can not compute the precision (recall) indices. Instead false positives (false negatives) are written and the reader should refer to the database details in table 4.1 to get the sense of their values.]

Given the real size of the traffic lights  $S(w; h)$  (in meters) it is trivial to estimate their position  $P(x; y; z)$  in the world space coordinate (the origin being the camera optical center). Assuming that the lens distortion is negligible there is a simple way to map the image space coordinates to the world coordinates using the intrinsic and extrinsic camera parameters. Here we need only the distance ( $P_z$ ) that may be computed with:

$$P_z = \frac{f \times S_h}{s_h \times k_u} \quad (4.24)$$

$f$  the focal (m),  $k_u$  the pixel size (m), and  $s(w; h)$  the size in the image space. Our settings are:  $f = 12 \times 10^{-3}m$ ,  $k_u = 8.3 \times 10^{-6}m$ . As for the real size of traffic lights, street measurements have shown that  $S_w = 0.36m$  and  $S_h = 1.08m$ .

Using the distance estimation we have drawn the recall versus distance from sub-sections of the sequences *dbFrance1\_8659* and *dbFrance2\_3574*, see figure 4.33. This is of course a frame per frame evaluation. From this plot we see that the recall is generally decreasing with distance which is logical. More than 50% of the traffic lights are recognized when closer than  $\approx 80m$  and about 80% when closer than 40 meters. However, the curve is somewhat chaotic. This is partially due to the sub-sequences used. As the car is often turning at street corners, some traffic lights are being visible only at specific distances before a turn or right after. To avoid this we should record new data in straight streets.

Now, using only traffic lights that we see from afar we have estimated the average minimum size of detection. Results of these experiments is that green traffic lights are output in average from our system when  $19.37pxl$  height (std 2.54) which means about  $\approx 80m$  from the vehicle. Red traffic lights in image space are in average  $22.57pxl$  height (std 4.08) when first output. That is  $\approx 69m$  from the vehicle. Given that the speed limit in French cities is  $50km/h$  the stopping

distance<sup>11</sup> is 30m. Hence, our system outputs traffic light status far before the critical distance and thus is able to alert the driver in time.

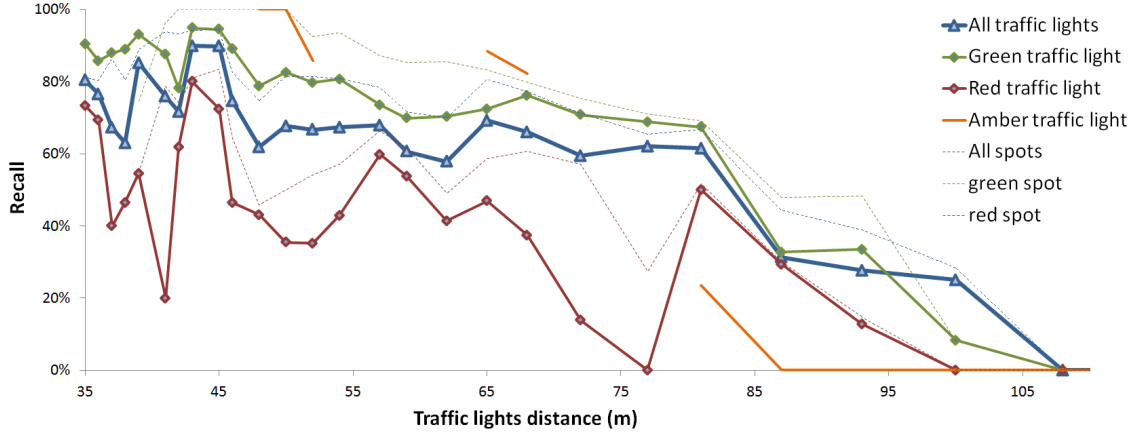


Figure 4.33: Recall versus distance without temporal filtering. Shown are the performance on two joint databases (*dbFrance1\_8659*, *dbFrance2\_3574*). The thin dotted lines indicate the performance of the spot light detection and act as the upper boundaries of our system. Note that, the very few amber lights did not allow plotting a continuous amber curve for every distances.

#### 4.5.2b Switzerland, China

We have verified the modularity of our system from the evaluation of its performance on sequences recorded in other countries. Samples from these sequences are shown in figure 4.32. Performance on sequences from Switzerland and China are also listed in table 4.2. The challenges faced in these sequences are numerous: low contrast, LED traffic lights, low frequency, different vision sensors, wide angle. Despite these challenges, the sequences are handled by our systems with high precision (over 88% on still images) but again with low recall ( $\approx 45\%$ ). A note on the parameters changed is provided at the end of this section.

The Swiss sequence<sup>12</sup> exhibits three types of traffic lights with different geometry including traffic lights that are *over-the-road* or *road-side*, and horizontal or vertical. It took about 8h of work for us to design the new templates and evaluate the sequence. An interesting point, is that this sequence counts many over-the-road traffic lights which we did not have in our French sequences.

On a frame per frame basis, recall on this sequence is again low (**45.7%**) but the precision is high (**93.5%**). Given that we have different three different geometrical templates the precision is good as the number of templates used increases the odds of having false positives.

We have also evaluated the performance on a non-professional sequence recorded with a personal camera in a cab in Shanghai, China. The FPS of this sequence is only 11 and the gray dynamic is really poor implying a low contrast and poor visibility of the scene. Remarkably, the

<sup>11</sup>The stopping distance  $D$  is the sum of the distance travelled during the driver's response time  $D_{response}$  and the braking distance  $D_{brake}$ . Here, we use a response time  $R = 1sec$ . The braking distance is approximated from the velocity  $v_0$  and the acceleration during braking  $a_0$ . We use an average value of  $a_0 = -6m.sec^{-2}$ . Hence driving at 50km/h,  $v_0 = 50km/h = 14m/sec$  and  $D = D_{response} + D_{brake} = v_0 \times R + \frac{-v_0}{2 \times a_0} = 14 + 16 = 30m$

<sup>12</sup>Please note that the Swiss sequences authorship goes to the École Polytechnique Fédérale de Lausanne (EPFL), and that the result of this evaluation might be removed in the public version of this manuscript.

geometry of the traffic lights in China were sufficiently similar for us to evaluate the template matching with the exact same templates as for France. The main challenge of this sequence is the LED refresh rate of the traffic lights. Hence, a traffic light currently *on* may be imaged *off* for few frames since the imaging sensor and the LED lights are asynchronous. Though this has a great effect on the performance, we did not adjust our system or groundtruth to get better result so that the traffic lights are annotated as if they were seen with the *light on* in all frames.

That being said, our performance are satisfactory. On still images recall is **44.9%** and precision is **88.2%**. In this sequence, the precision is the worst we obtained compared to other sequences.

**A note on the parameters** Our spot light detection runs with the exact same settings on 3 of 4 sequences (*dbFrance1\_8659*, *dbFrance2\_3574*, *dbSwitzerland1\_5183*). Given the low contrast and the non perfect circular pattern of the lights in the Chinese sequence we lowered the  $S_{clamp}$  value and the  $S_{circularity}$  to half of their original value.

Concerning the template matching, all databases – but the Swiss one – use the exact same settings (the geometry is similar in France and China). As aforementioned, we had to design other templates that match the geometry of Swiss traffic lights. Also, since the light colors appear faded in Switzerland we changed the reference values for the light color estimation. To pick the right color, we took the center of gravity of the hues histograms of red and green lights. All other operators used in the classification are identical for *dbSwitzerland1\_5183* than for the other databases, despite the use of different geometry.

The temporal settings remain unchanged in all sequences.

### 4.5.3 Timing

The complete traffic light recognition was written in C++ language and implemented as modules of the prototyping software RtMaps®. From 21,166 frames with resolution  $640 \times 480$  on a notebook Sony Vaio VPCZ1 Intel i5 core (single threaded) the average computation time is **42.30ms** (std  $15.27ms$ ) which is  $\approx 24$ FPS. The spot light detection stage is the most consuming with **37.41ms** in average (std  $13.94ms$ ). Template matching average is **4.87ms** (std  $3.38ms$ ), that is about  $0.45ms$  per candidate. Temporal filtering average is **0.02ms** (std  $0.03ms$ ).

The whole process is fast enough for “real-time” time constraints but could definitely be faster. More specifically the spot light detection stage should be optimized. Currently, the region growing is equal to 69% (std 14%) of the stage duration and the morphological top-hat 22% (std 10%).

### 4.5.4 Quantitative comparison

Any quantitative comparison should be conducted only using the same databases and the same methodology of evaluation. To that day no databases or source code have been made public to allow such comparison. Noteworthy, the German FZI institute will shortly publish their own database<sup>13</sup>.

Since, we have made our own database publicly available<sup>14</sup> it has drawn interest from others research teams. However, currently only [Siogkas et al., 2012] published the performance they achieved with our database. We report the performance of our system in table 4.3 when compared

<sup>13</sup><http://www.fzi.de/tks>

<sup>14</sup><http://www.lara.prd.fr/benchmarks/trafficlightsrecognition>

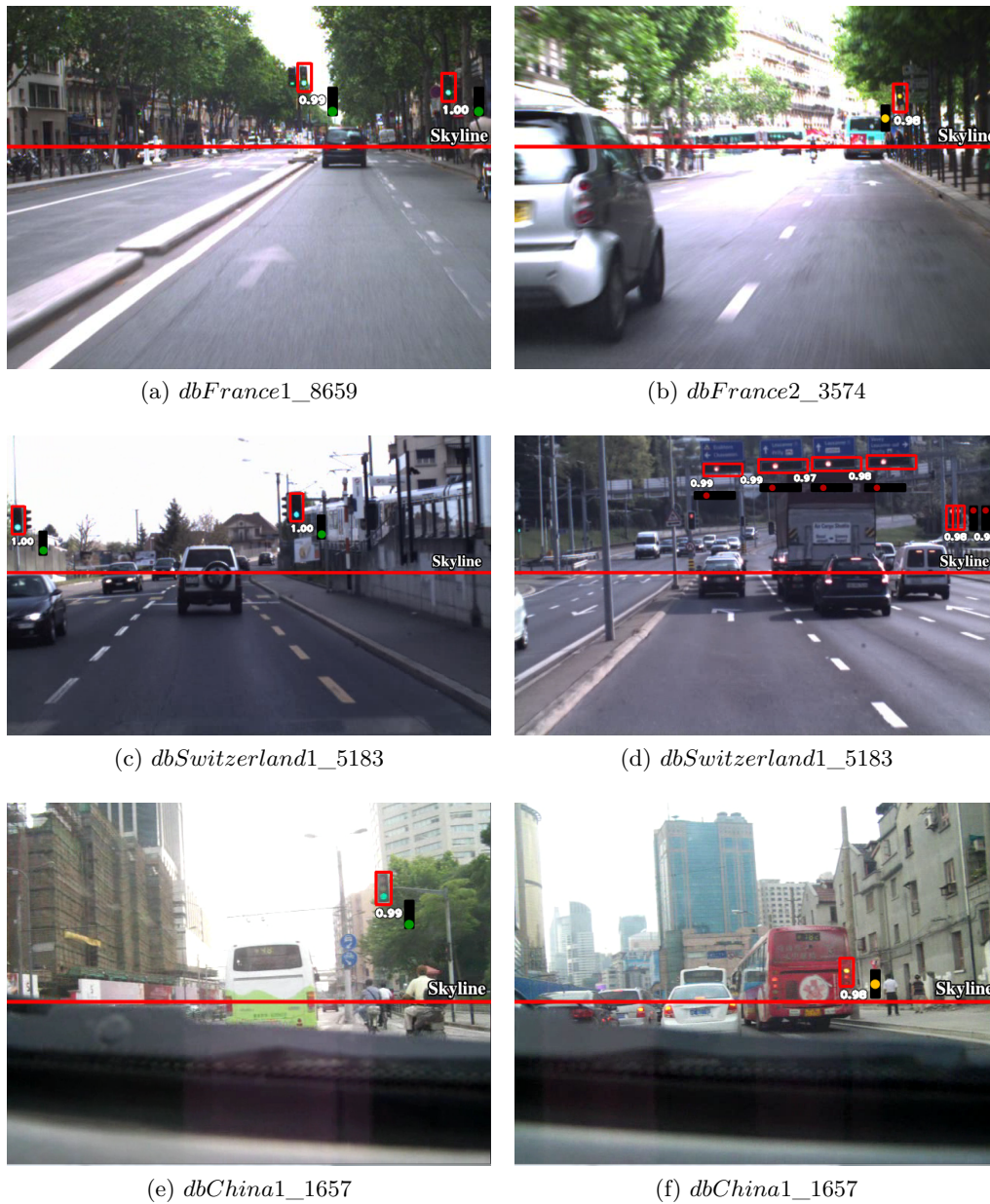


Figure 4.34: Sample outputs of the traffic light recognition in France (a-b), Switzerland (c-d) and China (e-f). For each traffic light output the confidence is written and the validated template is shown. (d) shows six successful recognition, but note that one traffic light on the left side is not detected. (f) shows a typical false alarm due to high vehicle lights.

to [Siogkas et al., 2012], and an Haar-Adaboost recognition (that we detail below). Let us emphasize that neither the Adaboost method nor Siogkas is capable of detecting amber lights so we ignored the amber lights output of our system.

	Frame Per Frame	Temporal
Our approach	<b>92.2%</b> (40.1%)	<b>88.9%</b> ( <b>93.8%</b> )
[Siogkas et al., 2012]	N/A	61.2% ( <b>93.8%</b> )
Adaboost	41.5% ( <b>41.9%</b> )	34.1% (70.0%)

Table 4.3: Comparison of performance with different approaches on a frame per frame basis and with temporal filtering on the sequence *dbFrance2\_3574*. The performance are indicated as: precision (recall), and bold values indicate the best values for each column. [Siogkas et al., 2012] do not provide frame per frame performance. Comparison exhibits our system is having a lower recall than Adaboost but our precision is *at least 2.2* times better than Adaboost and 1.5 times better than [Siogkas et al., 2012]. Note that precision is crucial for an ADAS.

On a frame per frame basis the recall is slightly better with Adaboost (+1.8%) but the precision of our approach is at least twice the precision of Adaboost (**92.2%** against 41.5%). Sadly, Siogkas *et al.* do not report the frame per frame performance so no comparison is possible. However, they state that their method output 1,481 false positives on a frame-per-frame basis, which is  $\approx 4.5$  times more than our system (327 false positives).

Temporally the recall of our system is higher than Adaboost and identical to Siogkas. However, the temporal precision (**88.9%**) is much better than what is obtained with Adaboost (34.1%) or Siogkas (61.2%). Besides, one should note that Adaboost can recognize traffic lights when they are minimum  $9 \times 21pxl$  ( $\approx 50m$ ) whereas our method as Siogkas is capable of detecting traffic lights from  $6 \times 15pxl$  size ( $\approx 80m$ ).

From these comparisons the Adaboost method seems to perform worst than the two others. On the bright side, our approach is doubtless better than Adaboost and [Siogkas et al., 2012] especially when comparing precision. Other non-quantitative tests (from other teams) on our database report high number of false positives which assess of the challenging database we are using.

We refer to the article of [Siogkas et al., 2012] for more information on their method. The following is a brief description of the Adaboost method we have implemented here.

**Statistical learning** Experiments were conducted using neural network (multilayer perceptron), genetic algorithm and Adaboost with features like: control points, connected control points, or Haar. The best results were obtained using Adaboost and Haar features. The strategy chosen for both training and testing, is rather fashion using 11,180 negative samples and 2,112 positive samples with a thin space surrounding the samples. Green and red traffic lights were extracted from sequence *dbFrance1\_8659* but amber could not be trained given the few samples for training. The final training counts 9 boosting classifiers and 321 features.

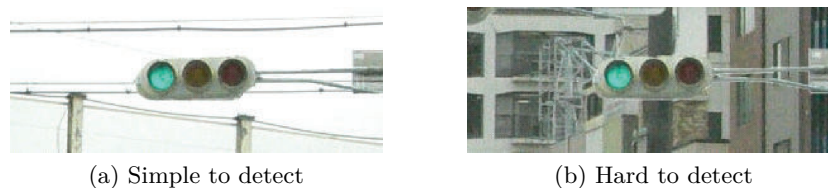


Figure 4.35: Traffic light visibility varies according to their spot intensity, the background, or the presence of nearby light sources. Source: [Kimura et al., 2007]

## 4.6 Conclusion

Unlike other object-recognition tasks traffic lights detection has only drawn little interest in the past and most of the prior works were intended to work in simple environments. In this chapter we have proposed a complete chain working even with low dynamics camera. With such cameras a common issue also discussed in [Hwang et al., 2006, Kim et al., 2007] is to distinguish amber from red as the two colors overlap. To overcome this problem we have proposed to use an halo estimation (section 4.4.2b) and identify the true color of the spot from both the inner hue and the halo hue. Other originality of our approach lies in the grayscale spot detection (section 4.3) and the use of template matching for classification (section 4.4).

An earlier version version of this traffic traffic light recognition was published in [de Charette and Nashashibi, 2009a] and compared with Adaboost in [de Charette and Nashashibi, 2009b].

To evaluate the performance of our system it has been tested on four different databases (from France, Switzerland and China) recorded in dense urban environment with neon-lights and other distracting elements. The precision our system exhibits is  $\approx 90\%$  or more without the support of any post processing and is capable of detecting traffic lights which minimum axis length is 6 pixels or more. Without tracking, the recall remains however low in most of the sequences generally due to motion blur, low light intensity or back-lighting.

With the use of post-processing (section 4.5.1) including tracking and light sequence filtering the recall is consequently higher, that is over **93%** on French sequences since we do not require to detect every single occurrences. Conversely, the precision is lower when evaluated on a temporal basis. The two reasons of false alarms are: other lights (e.g. pedestrian lights, neon-signs) and reflections of actual traffic lights in shop/buses windows. Assuming that we ignore false positives from reflections one can expect about 1 false alarm every 8 minutes in France. Further analysis on French sequences have shown that we detect red traffic lights when  $\approx 69\text{m}$  away from the vehicle ( $19.37\text{pxl}$ ) which is much farther than the critical stopping distance of  $30\text{m}$  when driving at the speed limit that is  $50\text{km/h}$ .

To quantitatively compare the performance of our TLR we have made public our own database and according to the numerous contacts initiated, several worldwide teams are using it. To that day only one team published results in [Siogkas et al., 2012]. It shows that our TLR performs better all the time, that is: on a frame per frame basis and a temporal basis (see section 4.5.4). A Haar-Adaboost learning strategy has also been implemented. As for our system, Adaboost suffers from a low recall on a frame per frame basis – though slightly better (+2.5%) – but our precision is at least twice better.



Finally, the complete chain we have presented is a good step towards a fully working traffic light detector but still requires some work. The most important improvement would be to handle special lights (i.e. arrow lights, blinking lights, etc.) that may occur quite often. Additionally, we would like to compare the performance of our system with the ability of real drivers to *see* traffic lights. Following the conclusion of a Japanese team, the visibility of traffic lights can be evaluated from the type of background (texture, sky, etc.) and the presence of nearby lights sources [Kimura et al., 2007], as in figure 4.35. It could be interesting to use their idea so as to evaluate whether our system is able to detect traffic lights with low visibility conditions.

# Conclusion

Though unrelated, urban scenes and adverse weathers are two scenarios the intelligent vehicles have to deal with. Hopefully, the research work we have presented provides a glimpse of what is possible to achieve now and in a close future, using a single imaging sensor and vision algorithms.

**Rain** For rain, we have first provided the reader with a comprehensive review on the physics, optics and meteorological models needed to develop rain-related applications not only in vision ADAS but in computer vision in general. To some extent we have compared the models qualitatively or quantitatively using real data from recognized experiments. As a conclusion, we have shown that computer visions applications are not always using the right models which may have an important impact on the output of such systems.

Among others inaccuracies, the exponential drop size distributions overestimates the number of drops and the dynamics models currently used in vision also deviate from measurements. In these cases more accurate models were found in the *ad-hoc* literature. The optics of rain has also been investigated and we have discussed the effect of the latter on imaging sensors. Among others we have highlighted photometry and frequency based models.

**Unfocused raindrops detection on a windscreen** In vision based ADAS, raindrop detection allow assessing rain, activating the wipers, or – at best – digitally removing rain on picture to enhance the performance of other vision algorithms. Since the existing systems were prove to be ill-conditioned, we have investigated ways to detect unfocused raindrops on the windshield. Vision algorithms usually rely on gradients thus detecting unfocused drops – transparent by nature – is a challenging unresolved problem.

While our first method using photometry and lack of gradients is prone to a dead-end, our experiments using blur techniques are more elegant and exhibit preliminary results. From blind blur assessment, we have attempted to compute a blur map to locate possible positions of the drops. Wavelet decomposition, slope of the amplitude spectrum and weighted FFT have been attempted. Among others, the two last experiments have shown promising results and though intermediate, we believe that with temporal filtering it should be a good basis to locate the drops.

Further work should be conducted to allow finer and more responsive blur map computation since drops do not exhibit strong defocus. More specifically an accurate estimation of the blur extent (ergo, Point Spread Function) would allow restoring the image quality using deconvolution techniques. For the driver, the activation of wipers are sufficient to restore the visibility at least in daytime.

**Illumination through rain at night** At night, we have proposed a novel idea to remove the distracting appearance of falling raindrops directly in the scene. Following our idea, the headlights could illuminate the scene without shining any falling particles such that they would

appear *invisible*. Using a camera and a projector, we have investigated this idea with a high fidelity simulator that emulates the physics, dynamics and densities of rain and other particulate weathers.

The investigation has shown that such illumination device would allow high light throughput if the system runs fast enough and in a predictable way (**4ms** for example allows **95%** light throughput in all tested scenarios). Through simulation we have also proved that the same principle works for all particulate weathers, like snow or hail. A prerequisite to illumination through rain is the vision algorithm for drops detection. A simple yet efficient algorithm was developed to detect rain in night-time experiments through gradients analysis. Our algorithm performance exhibit that the drops are detected with an average error of **1.99pxl** with **2ms** exposure time and **0.88pxl** with **1ms** exposure.

A stationary prototype was built by a member of our team to assess our findings, and the result is that selective illumination is already feasible in laboratory conditions. As intended the light throughput remains high and the removal of rain appearance clearly enhances the overall visibility. There is obviously a long road ahead until such illumination could be applied to real cars. The next step is to build a portable device that would allow outdoor experiments. Vision algorithms have also to be further studied to ensure that rain can also be detected with a moving camera at night. If such illumination device was effectively working this would help the driver in the complex scenario of night time driving in rain.

**Traffic Lights Recognition** For other complex scenarios like urban driving we have investigated the case of traffic light recognition. Traffic lights are key elements in the interpretation of a scene and their detection could be used to alert the drivers. Previous vision algorithms for traffic light recognition were developed during the last ten years but for the most part their working range is limited to the simplest scenarios. Most often, only traffic lights over-the-road are detected, and usually only in semi-urban environment.

Our proposal of a real-time chain of recognition for traffic lights detection is based on a first spot light detection stage where light sources are identified using top-hat morphological operators and geometric constraints. Following this stage, we use a modular template matching classifier to evaluate candidates and identify their state. In the context of low dynamic camera, we have proposed a simple heuristic to estimate the halo and then accurately classify the amber/red lights which is a well known issue for traffic lights recognition. Results from 14,000 traffic lights exhibit that with this improvement **98%** of the amber/red lights are accurately distinguished.

Thanks to our modular template algorithm, the evaluation of this chain of recognition was achieved on databases with various types of traffic lights from France, China or Switzerland. The performance shows the system is having a high precision over **88%** on still images but the recall is low (**40%** – **61%**). On a temporal basis the precision is over **87%** but the recall is at least **72%**. From French sequences we first detect the red lights when their height is greater than 22 pixels ( $\approx$  **60m**) which is sufficient for an ADAS. Using our public database we have compared performance with another article using our database and an Adaboost system. In both case our system performed better, especially for the precision.

Possible improvements include a large variety of avenues. From a computer vision perspective, we could experiment detecting the lights from physics models. Even more important would be to investigate the reliability of the spot light detection output. We believe that many failures of the existing traffic lights recognition systems are caused by the noisy output of this stage. Accuracy of the centroid position and the radius estimation should be investigated. From an ADAS perspective, better modelling of the changing states of traffic lights could consequently improve the temporal performance. Finally, deeper evaluation is required, such as comparing performance

with other systems. Evaluation in adverse conditions could also be a nice improvement, like rain for instance...



# Publications

## Journals

- [2012] de Charette, R. and Nashashibi, F., 2012. Rain in Computer Vision: a State of the Art. **Work In Progress**.

## Conferences

- [de Charette et al., 2012] de Charette, R., Tamburo, R., Barnum, P., Rowe, A., Kanade, T., Narasimhan, S., 2012. Fast Reactive Control for Illumination through Rain and Snow. IEEE International Conference on Computational Photography (ICCP), Seattle, USA.
- [Nashashibi et al., 2010] Nashashibi, F., de Charette, R., 2010. Detection of Unfocused Raindrops on a Windscreen using Low Level Image Processing. IEEE International Conference on Control Automation RobotiCs and Vision (ICARCV), Singapore, Singapore.
- [de Charette and Nashashibi, 2009a] de Charette, R., Nashashibi, F., 2009. Traffic light recognition using image processing compared to learning processes. IEEE Intelligent Robots and Systems (IROS), Saint Louis, USA.
- [de Charette and Nashashibi, 2009b] de Charette, R., Nashashibi, F., 2009. Real time visual traffic lights recognition based on Spot Light Detection and Adaptive Traffic Lights Templates. IEEE Intelligent Vehicles Symposium (IV), Xi'An, China.



# Bibliography

- [Adimula and Ajayi, 1996] Adimula, I. and Ajayi, G., 1996. Variations in raindrop size distribution and specific attenuation due to rain in Nigeria. *Annals of Telecommunications* 51(1), pp. 87–93. [↗](#)
- [Al-Khateeb and Johari, 2008] Al-Khateeb, K. and Johari, J. A. Y., 2008. Intelligent dynamic traffic light sequence using RFID. In: *IEEE International Conference on Computer and Communication Engineering 2008 (ICCCE)*, Kuala Lumpur, Malaysia, pp. 1367–1372. [↗](#)
- [Alonge and Afullo, 2011] Alonge, A. A. and Afullo, T. J., 2011. Rainfall rate modeling for various rainfall types in South Africa. In: *IEEE Africon 2011*, Livingston, Zambia, pp. 1–6. [↗](#)
- [Angin et al., 2010] Angin, P., Bhargava, B. and Helal, S., 2010. A Mobile-Cloud Collaborative Traffic Lights Detector for Blind Navigation. In: *International Conference on Mobile Data Management 2010 (MDM)*, Kansas City, USA, pp. 396–401. [↗](#)
- [Atlas et al., 1973] Atlas, D., Srivastava, R. C. and Sekhon, R. S., 1973. Doppler radar characteristics of precipitation at vertical incidence. *Reviews of Geophysics* 11(1), pp. 1–35. [↗](#)
- [Babinow, 1953] Babinow, J., 1953. Automatic headlight dimmer. [↗](#)
- [Bandyopadhyay et al., 2001] Bandyopadhyay, D., Raskar, R. and Fuchs, H., 2001. Dynamic Shader Lamps: Painting on Movable Objects. In: *IEEE ACM Augmented Reality 2001*, New York, NY, USA, pp. 207–216. [↗](#)
- [Barclay, 1978] Barclay, P., 1978. Properties of Rain for Microwave Propagation Studies (Report MEE-78). Technical report, School of Engineering, Monash University, Melbourne, Australia. [↗](#)
- [Bargeton, 2009] Bargeton, A., 2009. Fusion multi-sources pour l’interprétation d’un environnement routier. PhD thesis, Mines ParisTech, Paris, France. [↗](#)
- [Barnum, 2011] Barnum, P. C., 2011. Light and Water Drops. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.
- [Barnum et al., 2007] Barnum, P. C., Kanade, T. and Narasimhan, S. G., 2007. Spatio-temporal frequency analysis for removing rain and snow from videos. In: *Proceedings of the First International Workshop on Photometric Analysis for Computer Vision 2007 (PACV)*, Rio de Janeiro, Brazil. [↗](#)
- [Barnum et al., 2010a] Barnum, P. C., Narasimhan, S. and Kanade, T., 2010a. Analysis of rain and snow in frequency space. *IEEE International Journal of Computer Vision (IJCV)* 86(2), pp. 256–274. [↗](#)



- [Barnum et al., 2009] Barnum, P. C., Narasimhan, S. G. and Kanade, T., 2009. A projector-camera system for creating a display with water drops. In: *IEEE Computer Vision and Pattern Recognition Workshops, 2009 (CVPR Workshops)*, Miami, USA, pp. 17–23. [↗](#)
- [Barnum et al., 2010b] Barnum, P. C., Narasimhan, S. G. and Kanade, T., 2010b. A multi-layered display with water drops. *SIGGRAPH 2010* 29(4), pp. 761–767. [↗](#)
- [Beam, 2000] Beam, N., 2000. Adaptive/anti-blinding headlights. [↗](#)
- [Beard, 1976] Beard, K. V., 1976. Terminal velocity and shape of cloud and precipitation drops aloft. *Journal of Atmospheric Sciences* 33(5), pp. 851–864. [↗](#)
- [Beard, 1984] Beard, K. V., 1984. Oscillation models for predicting raindrop axis and backscatter ratios. *Radio Science* 19(1), pp. 67–74. [↗](#)
- [Beard and Chuang, 1987] Beard, K. V. and Chuang, C., 1987. A New Model for the Equilibrium Shape of Raindrops. *Journal of the Atmospheric Sciences* 44(11), pp. 1509–1524. [↗](#)
- [Beard et al., 2010] Beard, K. V., Bringi, V. N. and Thurai, M., 2010. A new understanding of raindrop shape. *Atmospheric Research* 97(4), pp. 396–415. [↗](#)
- [Bentley, 1904] Bentley, W. A., 1904. Studies of raindrops and raindrop phenomena. *Monthly Weather Review* 32(10), pp. 450–456.
- [Best, 1950] Best, A., 1950. The size distribution of raindrops. *Quarterly Journal of the Royal Meteorological Society* 76(327), pp. 16–36. [↗](#)
- [Bhise et al., 1981] Bhise, V. D., Meldrum, J. F., Forbes, L. M., Rockwell, T. H. and McDowell, E. D., 1981. Predicting Driver Seeing Distance in Natural Rainfall. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 23(6), pp. 16. [↗](#)
- [Bimber and Iwai, 2008] Bimber, O. and Iwai, D., 2008. Superimposing dynamic range. *ACM Transactions on Graphics* 27(5), pp. 1. [↗](#)
- [Blanchard, 2004] Blanchard, D. C., 2004. From raindrops to volcanoes: adventures with sea surface meteorology. Dover Pubns. [↗](#)
- [Bos et al., 2001] Bos, B. J., Schofield, K. and Larson, M. L., 2001. Rain sensor. [↗](#)
- [Bradley et al., 2000] Bradley, S., Stow, C. and Lynch-Blosse, C., 2000. Measurements of rainfall properties using long optical path imaging. *Journal of Atmospheric and Oceanic Technology* 17(6), pp. 761–772. [↗](#)
- [Bradski, 1998] Bradski, G. R., 1998. Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technical Journal* 2(2), pp. 12–21. [↗](#)
- [Brewer and Liu, 2008] Brewer, N. and Liu, N., 2008. Using the Shape Characteristics of Rain to Identify and Remove Rain from Video. In: *Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR & SPR), SSPR & SPR '08*, Berlin, Germany, pp. 451–458. [↗](#)
- [Cannon, 1976] Cannon, M., 1976. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24(1), pp. 58–63. [↗](#)
















- [Cecchi et al., 2010] Cecchi, G. A., Rao, A. R., Xiao, Y. and Kaplan, E., 2010. Statistics of natural scenes and cortical color processing. *Journal of vision* 10(11), pp. 21. [↗](#)
- [Challita, 2008] Challita, G., 2008. Amélioration du positionnement des véhicules communicants équipés de GPS et de système de vision. PhD thesis, INSA Rouen, Rouen, France. [↗](#)
- [Cheng and English, 1983] Cheng, L. and English, M., 1983. A Relationship Between Hailstone Concentration and Size. *Journal of the Atmospheric Sciences* 40(1), pp. 204–213. [↗](#)
- [Choi, 2010] Choi, E.-H., 2010. Crash factors in intersection-related crashes: An on-scene perspective. Technical report, National Highway Traffic Safety Administration, Washington D.C., USA. [↗](#)
- [Chuang and Beard, 1990] Chuang, C. and Beard, K. V., 1990. A Numerical Model for the Equilibrium Shape of Electrified Raindrops. *Journal of Atmospheric Sciences* 47, pp. 1374–1389. [↗](#)
- [Chung et al., 2004] Chung, Y.-C., Wang, J.-M., Bailey, R. R., Chen, S.-W. and Chang, S.-L., 2004. A non-parametric blur measure based on edge analysis for image processing applications. In: *IEEE Conference on Cybernetics and Intelligent Systems 2004*, Singapore, pp. 356–360. [↗](#)
- [Cord and Aubert, 2011] Cord, A. and Aubert, D., 2011. Towards Rain Detection through Use of In-Vehicle Multipurpose Cameras. In: *IEEE Intelligent Vehicles Symposium 2011 (IV)*, Baden-Baden, Germany, pp. 833–838.
- [Crete et al., 2007] Crete, F., Thierry, D., Ladret, P. and Nicolas, M., 2007. The blur effect: perception and estimation with a new no-reference perceptual blur metric. *SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging*. [↗](#)
- [Crundall et al., 2006] Crundall, D., Van Loon, E. and Underwood, G., 2006. Attraction and distraction of attention with roadside advertisements. *Accident; analysis and prevention* 38(4), pp. 671–7. [↗](#)
- [Da Rugna et al., 2003] Da Rugna, J., Konik, H. and Others, 2003. Automatic blur detection for metadata extraction in content-based retrieval context. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol. 5304, pp. 285–294. [↗](#)
- [Dai and Wu, 2007] Dai, Z. and Wu, Y., 2007. Where are focused places of a photo? In: *International conference on Advances in visual information systems 2007 (VISUAL)*, VISUAL'07, Berlin, Germany, pp. 73–83. [↗](#)
- [Davidse, 2006] Davidse, R. J., 2006. Older Drivers and ADAS - Which Systems Improve Road Safety? -. *IATSS Research* 30(1), pp. 6–20. [↗](#)
- [de Charette and Nashashibi, 2009a] de Charette, R. and Nashashibi, F., 2009a. Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates. In: *IEEE Intelligent Vehicles Symposium 2009 (IV)*, Xian, China, pp. 358–363. [↗](#)
- [de Charette and Nashashibi, 2009b] de Charette, R. and Nashashibi, F., 2009b. Traffic light recognition using image processing compared to learning processes. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems 2009 (IROS)*, Saint Louis, MO, USA, pp. 333–338. [↗](#)














- [de Charette et al., 2012] de Charette, R., Tamburo, R., Barnum, P. C., Rowe, A., Kanade, T. and Narasimhan, S. G., 2012. Fast Reactive Control for Illumination Through Rain and Snow. In: IEEE International Conference on Computational Photography 2012 (ICCP), Seattle, WA, USA, pp. 1–10.
- [Dennis et al., 1971] Dennis, A. S., Smith, P. L., Peterson, G. A. P. and McNeil, R. D., 1971. Hailstone Size Distributions and Equivalent Radar Reflectivity Factors Computed from Hailstone Momentum Records. *Journal of Applied Meteorology* 10(1), pp. 79–85. [↗](#)
- [Desaulniers-Soucy et al., 2001] Desaulniers-Soucy, N., Lovejoy, S. and Schertzer, D., 2001. The HYDROP experiment: an empirical method for the determination of the continuum limit in rain. *Atmospheric Research* 59-60(2), pp. 163–197. [↗](#)
- [Ditze et al., 2010] Ditze, M., Geramani, K., Golatowski, F., Gustafsson, S., Varhelyi, A. and Lamm, N., 2010. A Survey on Intelligent Vehicle Safety Systems for Adverse Weather Conditions. In: FISITA 2010 World Automotive Congress, Budapest, Hungary, p. 8. [↗](#)
- [Dong, 2008] Dong, P., 2008. Analysis, modelling and robust control of automotive HID headlight systems. PhD thesis, The Hong Kong Polytechnic University, Hong Kong. [↗](#)
- [Donnadieu, 1980] Donnadieu, G., 1980. Comparison of Results Obtained with the VIDIAZ Spectropluviometer and the Joss-Waldvogel Rainfall Disdrometer in a “Rain of a Thundery Type”. *Journal of Applied Meteorology* 19(5), pp. 593–597. [↗](#)
- [Eitoku et al., 2006] Eitoku, S., Tanikawa, T. and Suzuki, Y., 2006. Display Composed of Water Drops for Filling Space with Materialized Virtual Three-dimensional Objects. In: IEEE Virtual Reality Conference 2006 (VR), Virginia, VA, USA, pp. 159–166. [↗](#)
- [Elder and Zucker, 1998] Elder, J. and Zucker, S., 1998. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 20(7), pp. 699–716. [↗](#)
- [Fabian and Malah, 1991] Fabian, R. and Malah, D., 1991. Robust identification of motion and out-of-focus blur parameters from blurred and noisy images. *CVGIP: Graphical Models and Image Processing* 53(5), pp. 403–412. [↗](#)
- [Fairfield and Urmson, 2011] Fairfield, N. and Urmson, C., 2011. Traffic light mapping and detection. In: IEEE International Conference on Robotics and Automation 2011 (ICRA), pp. 5421–5426. [↗](#)
- [Feingold and Levin, 1986] Feingold, G. and Levin, Z., 1986. The lognormal fit to raindrop spectra from frontal convective clouds in Israel. *Journal of climate and applied meteorology* 25(10), pp. 1346–1363. [↗](#)
- [Field, 1987] Field, D. J., 1987. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A (JOSA A)* 4(12), pp. 2379–2394. [↗](#)
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), pp. 381–395. [↗](#)

- [Foote, 1973] Foote, G. B., 1973. A numerical method for studying liquid drop behavior: Simple oscillation. *Journal of Computational Physics* 11(4), pp. 507–530. [↗](#)
- [Foote and Du Toit, 1969] Foote, G. B. and Du Toit, P. S., 1969. Terminal Velocity of Raindrops Aloft. *Journal of Applied Meteorology* 8(2), pp. 249–253. [↗](#)
- [Franke et al., 1998] Franke, U., Gavrilă, D., Görzig, S., Lindner, F., Paetzold, F. and Wöhler, C., 1998. Autonomous driving goes downtown. *IEEE Intelligent Systems and their Applications* 13(6), pp. 40–48. [↗](#)
- [Fujiyoshi et al., 2008] Fujiyoshi, Y., Yamamura, I., Nagumo, N., Nakagawa, K., Muramoto, K. and Shimomai, T., 2008. The Maximum Size of Raindrops - Can it be a Proxy of Precipitation Climatology ? -. In: *International Conference on Clouds and Precipitation 2008 (ICCP)*, Cancun, Mexico, pp. 7–11. [↗](#)
- [Garg, 2007] Garg, K., 2007. Rain in vision and graphics. PhD thesis, Columbia University, New York, USA. [↗](#)
- [Garg and Nayar, 2003] Garg, K. and Nayar, S. K., 2003. Photometric model of a rain drop. Technical report, Computer Science Department, Columbia University, New York, USA. [↗](#)
- [Garg and Nayar, 2004] Garg, K. and Nayar, S. K., 2004. Detection and removal of rain from videos. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004 (CVPR)*, Vol. 1, Washington D.C., USA, pp. 528–535. [↗](#)
- [Garg and Nayar, 2005] Garg, K. and Nayar, S. K., 2005. When does a camera see rain? In: *IEEE International Conference on Computer Vision 2005 (ICCV)*, Vol. 2, Beijing, China, pp. 1067–1074. [↗](#)
- [Garg and Nayar, 2007] Garg, K. and Nayar, S. K., 2007. Vision and Rain. *International Journal of Computer Vision (IJCV)* 75(1), pp. 3–27. [↗](#)
- [Gavrila et al., 2001] Gavrilă, D. M., Franke, U., Wohler, C. and Gorzig, S., 2001. Real time vision for intelligent vehicles. *IEEE Instrumentation & Measurement Magazine* 4(2), pp. 22–27. [↗](#)
- [Gloaguen and Lavergnat, 1995] Gloaguen, C. and Lavergnat, J., 1995. Raindrop size distribution near Paris. *Electronics Letters* 31(5), pp. 405–406. [↗](#)
- [Gong et al., 2010] Gong, J., Jiang, Y., Xiong, G., Guan, C., Tao, G. and Chen, H., 2010. The recognition and tracking of traffic lights based on color segmentation and CAMSHIFT for intelligent vehicles. In: *IEEE Intelligent Vehicles Symposium 2010 (IV)*, San Diego, CA, USA, pp. 431–435. [↗](#)
- [Görmer et al., 2009] Görmer, S., Kummert, A., Park, S.-B. and Egbert, P., 2009. Vision-based rain sensing with an in-vehicle camera. In: *IEEE Intelligent Vehicles Symposium 2009 (IV)*, Xi'an, China, pp. 279–284. [↗](#)
- [Gradinescu et al., 2007] Gradinescu, V., Gorgorin, C., Diaconescu, R., Cristea, V. and Iftode, L., 2007. Adaptive Traffic Lights Using Car-to-Car Communication. In: *IEEE Vehicular Technology Conference 2007 (VTC)*, Baltimore, MD, USA, pp. 21–25. [↗](#)
- [Green, 1975] Green, A. W., 1975. An Approximation for the Shapes of Large Raindrops. *Journal of Applied Meteorology* 14(8), pp. 1578–1583. [↗](#)

- [Grossberg et al., 2004] Grossberg, M. D., Peri, H., Nayar, S. K. and Belhumeur, P. N., 2004. Making One Object Look Like Another: Controlling Appearance Using a Projector-Camera System. In: IEEE Conference on Computer Vision and Pattern Recognition 2004 (CVPR), Washington D.C., USA, pp. 452–459. [↗](#)
- [Gunn and Marshall, 1958] Gunn, K. and Marshall, J., 1958. The distribution with size of aggregate snowflakes. *Journal of Atmospheric Sciences* 15(5), pp. 452–461. [↗](#)
- [Gunn and Kinzer, 1949] Gunn, R. and Kinzer, G. D., 1949. The Terminal Velocity of Fall for Water Droplets in Stagnant Air. *Journal of the Atmospheric Sciences* 6(6), pp. 243–248. [↗](#)
- [Halimeh and Roser, 2009] Halimeh, J. and Roser, M., 2009. Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation. In: IEEE Intelligent Vehicles Symposium 2009 (IV), Xi'An, China, pp. 610–615. [↗](#)
- [Hansen and Hess, 2006] Hansen, B. C. and Hess, R. F., 2006. Discrimination of amplitude spectrum slope in the fovea and parafovea and the local amplitude distributions of natural scene imagery. *Journal of vision* 6(7), pp. 696–711. [↗](#)
- [Haralick, 1974] Haralick, R. M., 1974. A Measure for Circularity of Digital Figures. *IEEE Transactions on Systems, Man, and Cybernetics* 4(4), pp. 394–396. [↗](#)
- [Hautière et al., 2010] Hautière, N., Bossu, J., Bigorgne, E. and Aubert, D., 2010. Détection de conditions réduites de visibilité par caméra bord de voies. Technical report, INRETS, Versailles, France. [↗](#)
- [Heidorn, 2000] Heidorn, K. C., 2000. Wilson A. Bentley: The Raindrop Man, too. [↗](#)
- [Hsu and Chen, 2008] Hsu, P. and Chen, B.-Y., 2008. Blurred image detection and classification. In: S. Satoh, F. Nack and M. Etoh (eds), International conference on Advances in Multimedia Modeling 2008 (MMM), MMM'08, Kyoto, Japan, pp. 277–286. [↗](#)
- [Hull and Gallegos, 2007] Hull, J. A. and Gallegos, J. J., 2007. Adaptive and interactive scene illumination. [↗](#)
- [Hwang et al., 2006] Hwang, T.-H., Joo, I.-H. and Cho, S.-I., 2006. Detection of Traffic Lights for Vision-Based Car Navigation System. In: PSIVT 2006, Vol. 4319/2006, Hsinchu, Taiwan, pp. 682–691. [↗](#)
- [Inagaki, 2008] Inagaki, T., 2008. Smart collaboration between humans and machines based on mutual understanding. *Annual Reviews in Control* 32(2), pp. 253–261. [↗](#)
- [Inagaki, 2011] Inagaki, T., 2011. Technological and Legal Considerations for the Design of Interactions between Human Driver and Advanced Driver Assistance Systems. In: NeTWork Worskhop, Control and Accountability in Highly Automated Systems 2011, Toulouse, France, p. 12. [↗](#)
- [Ivey et al., 1975] Ivey, D. L., Lehtipuu, E. K. and Button, J. W., 1975. Rainfall and visibility—the view from behind the wheel. Technical report, Texas Transportaton Institute, College Station, TX, USA. [↗](#)
- [Joss and Waldvogel, 1967] Joss, V. J. and Waldvogel, A., 1967. A raindrop spectrograph with automatic analysis. *Pure and Applied Geophysics* 68, pp. 240–246.

- [Kälberer, 2003] Kälberer, S. A., 2003. Detektion von Ampeln für ein Verkehrszeichenerkennungssystem. Master thesis, Universität Ulm.
- [Ke et al., 2006] Ke, Y., Tang, X. and Jing, F., 2006. The Design of High-Level Features for Photo Quality Assessment. In: IEEE Conference on Computer Vision and Pattern Recognition 2006 (CVPR), Vol. 1, New York, NY, USA, pp. 419–426. [↗](#)
- [Kim et al., 2007] Kim, Y. K., Kim, K. W. and Yang, X., 2007. Real Time Traffic Light Recognition System for Color Vision Deficiencies. In: IEEE International Conference on Mechatronics and Automation 2007 (ICMA), Harbin, China, pp. 76–81. [↗](#)
- [Kimura et al., 2007] Kimura, F., Takahashi, T., Mekada, Y., Ide, I., Murase, H., Miyahara, T. and Tamatsu, Y., 2007. Measurement of Visibility Conditions toward Smart Driver Assistance for Traffic Signals. In: IEEE Intelligent Vehicles Symposium 2007 (IV), Istanbul, Turkey, pp. 636–641. [↗](#)
- [Klauer et al., 2006] Klauer, S., Dingus, T., Neale, V., Sudweeks, J. and Ramsey, D., 2006. The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data. Technical report. [↗](#)
- [Kurihata et al., 2005] Kurihata, H., Takahashi, T., Ide, I., Mekada, Y., Murase, H., Tamatsu, Y. and Miyahara, T., 2005. Rainy weather recognition from in-vehicle camera images for driver assistance. In: IEEE Intelligent Vehicles Symposium (IV), Las Vegas, USA, pp. 205–210.
- [Kurihata et al., 2006] Kurihata, H., Takahashi, T., Mekada, Y., Ide, I., Murase, H., Tamatsu, Y. and Miyahara, T., 2006. Raindrop Detection from In-Vehicle Video Camera Images for Rainfall Judgment. In: International Conference on Innovative Computing, Information and Control 2006 (ICICIC), Vol. 2, Beijing, China, pp. 544–547. [↗](#)
- [Lal and Craig, 2001] Lal, S. K. and Craig, A., 2001. A critical review of the psychophysiology of driver fatigue. *Biological Psychology* 55(3), pp. 173–194. [↗](#)
- [Langleben, 1954] Langleben, M., 1954. The terminal velocity of snowflakes. *Quarterly Journal of the Royal Meteorological Society* 80(344), pp. 174–181. [↗](#)
- [Laws and Parsons, 1943] Laws, J. and Parsons, D., 1943. The relation of raindrop-size to intensity. *Transactions American Geophysical Union* 24(Part II), pp. 452–460. [↗](#)
- [Leleve et al., 2007] Leleve, J., Benschrair, A., Rebut, J. and Challita, G., 2007. Method of detecting rain on a windscreen. [↗](#)
- [Levinson et al., 2011] Levinson, J., Askeland, J., Dolson, J. and Thrun, S., 2011. Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles. In: IEEE International Conference on Robotics and Automation 2011 (ICRA), Shanghai, China, pp. 5784–5791. [↗](#)
- [Li and Tu, 2000] Li, R. and Tu, Z., 2000. Automatic Recognition of Civil Infrastructure Objects in Mobile Mapping Imagery Using Markov Random Field. Technical report, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH, USA. [↗](#)
- [Li et al., 2011] Li, Y., Cai, Z.-x., Gu, M.-q. and Yan, Q.-y., 2011. Traffic lights recognition based on morphology filtering and statistical classification. In: IEEE Seventh International Conference on Natural Computation 2011 (ICNC), Shanghai, China, pp. 1700–1704. [↗](#)

- [Lim, 2006] Lim, B., 2006. Derivation of the Shape of Raindrops. Technical report, School of applied and Engineering Physics, Cornell University, Ithaca, USA. 
- [Lin and Jay Kuo, 2011] Lin, W. and Jay Kuo, C.-C., 2011. Perceptual visual quality metrics: A survey. *Journal of Visual Communication and Image Representation* 22(4), pp. 297–312. 
- [Lindner et al., 2004] Lindner, F., Kressel, U. and Kälberer, S. A., 2004. Robust recognition of traffic signals. In: *IEEE Intelligent Vehicles Symposium, 2004 (IV)*, Parma, Italy, pp. 49–53. 
- [Liu et al., 2009] Liu, P., Xu, J., Liu, J. and Tang, X., 2009. Pixel Based Temporal Analysis Using Chromatic Property for Removing Rain from Videos. *Computer and Information Science* 2(1), pp. 53–60. 
- [Liu et al., 2008] Liu, R., Li, Z. and Jia, J., 2008. Image partial blur detection and classification. In: *IEEE International Conference on Computer Vision and Pattern Recognition 2008 (CVPR)*, Anchorage, AK, USA, pp. 1–8. 
- [Lu et al., 2008] Lu, K.-H., Wang, C.-M. and Chen, S.-Y., 2008. Traffic Light Recognition. *Zhōngguó gōngchéng xuékān* 31(6), pp. 1069–1075. 
- [Macke and Groß klaus, 1998] Macke, A. and Groß klaus, M., 1998. Light scattering by nonspherical raindrops. *Journal of Quantitative Spectroscopy and Radiative Transfer* 60(3), pp. 355–363. 
- [Marshall and Palmer, 1948] Marshall, J. and Palmer, W. M., 1948. The Distribution of Raindrops with Size. *Journal of Meteorology* 5(4), pp. 165–166. 
- [Mason, 1975] Mason, B. J., 1975. *Clouds, Rain and Rainmaking*. Cambridge University Press. 
- [Meyer, 1977] Meyer, F., 1977. Contrast features extraction. *Journal of Histochemistry & Cytochemistry* 27(1), pp. 128–135.
- [Miché and Debie, 1995] Miché, P. and Debie, R., 1995. Fast and self-adaptive image segmentation using extended declivity. *Annales des télécommunications* 50(3-4), pp. 401–410. 
- [Miura et al., 2000] Miura, J., Kanda, T. and Shirai, Y., 2000. An active vision system for real-time traffic sign recognition. In: *IEEE Intelligent Transportation Systems 2000 (ITSC)*, pp. 52–57. 
- [Montero-Martínez et al., 2009] Montero-Martínez, G., Kostinski, A. B., Shaw, R. a. and García-García, F., 2009. Do all raindrops fall at terminal speed? *Geophysical Research Letters* 36(11), pp. 2–5. 
- [Montero and Bribiesca, 2009] Montero, R. S. and Bribiesca, E., 2009. State of the art of compactness and circularity measures. *International Mathematical Forum* 27(4), pp. 1305–1335. 
- [Montgomery, 2012] Montgomery, G., 2012. *Color Blindness: More Prevalent Among Males*. 
- [Moumouni et al., 2008] Moumouni, S., Gosset, M. and Houngninou, E., 2008. Main features of rain drop size distributions observed in Benin, West Africa, with optical disdrometers. *Geophysical Research Letters*. 

- [Narasimhan, 2004] Narasimhan, S. G., 2004. Models and algorithms for vision through the atmosphere. Phd, Columbia University, New York, USA. 
- [Nashashibi et al., 2010] Nashashibi, F., de Charette, R. and Lia, A., 2010. Detection of Unfocused Raindrops on a Windscreen using Low Level Image Processing. In: IEEE International Conference on Control, Automation, Robotics and Vision 2010 (ICARCV), Singapore, Singapore. 
- [Nathan et al., 1964] Nathan, J., Henry, G. H. and Cole, B. L., 1964. Recognition of Colored Road Traffic Light Signals by Normal and Color-Vision-Defective Observers. *Journal of the Optical Society of America* 54(8), pp. 1041. 
- [Nešpor et al., 2000] Nešpor, V., Krajewski, W. F. and Kruger, A., 2000. Wind-Induced Error of Raindrop Size Distribution Measurement Using a Two-Dimensional Video Disdrometer. *Journal of Atmospheric and Oceanic Technology* 17(11), pp. 1483–1492. 
- [Nienhüser et al., 2010] Nienhüser, D., Drescher, M. and Zöllner, J. M., 2010. Visual State Estimation of Traffic Lights using Hidden Markov Models. In: IEEE Intelligent Transportation Systems, 2010 (ITSC), Madeira, Portugal, pp. 1705–1710. 
- [Nishigaki et al., 2000] Nishigaki, M., Saka, M., Aoki, T., Yuhara, H. and Kawai, M., 2000. Fail output algorithm of vision sensing. In: IEEE Intelligent Vehicles Symposium, 2000 (IV), Dearborn, MI, USA, pp. 581–584. 
- [Nousiainen, 2000] Nousiainen, T., 2000. Scattering of Light by Raindrops with Single-Mode Oscillations. *Journal of the Atmospheric Sciences* 57(6), pp. 789–802. 
- [Nzeukou et al., 2004] Nzeukou, A., Sauvageot, H., Ochou, A. D. and Kebe, C. M. F., 2004. Raindrop Size Distribution and Radar Parameters at Cape Verde. *Journal of Applied Meteorology* 43(1), pp. 90–105. 
- [Ojansivu et al., 2011] Ojansivu, V., Lepistö, L., Ilmoniemi, M., Heikkilä, J., Heyden, A. and Kahl, F., 2011. Degradation Based Blind Image Quality Evaluation. In: Scandinavian Conference on Image Analysis 2011 (SCIA), Vol. 6688, Springer Berlin / Heidelberg, Ystad Saltsjöbad, Sweden, pp. 306–316. 
- [Omachi and Omachi, 2010] Omachi, M. and Omachi, S., 2010. Detection of traffic light using structural information. In: IEEE International Conference on Signal Processing 2010 (ICSP), Beijing, China, pp. 809–812. 
- [Pang and Liu, 2001] Pang, G. and Liu, H., 2001. LED location beacon system based on processing of digital images. *IEEE Transactions on Intelligent Transportation Systems (ITS)* 2(3), pp. 135–150. 
- [Pantle and Sekuler, 1969] Pantle, A. and Sekuler, R., 1969. Contrast response of human visual mechanisms sensitive to orientation and direction of motion. *Vision Research* 9(3), pp. 397–406. 
- [Park et al., 2006] Park, J.-H., Kim, M.-H., Im, H.-J., Lee, K.-C. and Lee, S., 2006. Development of Vision based Control Smart Windshield Wiper System for Intelligent Vehicle. In: International Joint Conference SICE-ICASE, 2006, Busan, Korea, pp. 4398–4403. 



- [Premachandra et al., 2009] Premachandra, H. C. N., Yendo, T., Yamasato, T., Fujii, T., Tanimoto, M. and Kimura, Y., 2009. Detection of LED traffic light by image processing for visible light communication system. In: IEEE Intelligent Vehicles Symposium, 2009 (IV), IEEE, Xi'An, China, pp. 179–184. [↗](#)
- [Pruppacher and Pitter, 1971] Pruppacher, H. R. and Pitter, R. L., 1971. A Semi-Empirical Determination of the Shape of Cloud and Rain Drops. *Journal of the Atmospheric Sciences* 28(1), pp. 86–94. [↗](#)
- [Raskar et al., 2001] Raskar, R., Welch, G., Low, K.-l. and Bandyopadhyay, D., 2001. Shader Lamps: Animating Real Objects with Image Based Illuminations. In: Eurographics Workshop on Rendering Techniques 2001. [↗](#)
- [RightDiagnosis, 2012] RightDiagnosis, 2012. Prevalence and Incidence of Color blindness - RightDiagnosis.com. [↗](#)
- [Rinehart, 1983] Rinehart, R. E., 1983. Out-of-Level Instruments: Errors in Hydrometeor Spectra and Precipitation Measurements. *Journal of Applied Meteorology* 22, pp. 1404–1410. [↗](#)
- [Rom, 1975] Rom, R., 1975. On the cepstrum of two-dimensional functions. *IEEE Transactions on Information Theory* 21(2), pp. 214–217. [↗](#)
- [Rooms et al., 2002] Rooms, F., Pizurica, A. and Philips, W., 2002. Estimating image blur in the wavelet domain. In: IEEE International Conference on Acoustics Speech and Signal Processing 2002 (ICASSP), Orlando, FL, USA. [↗](#)
- [Rosenfeld, 1974] Rosenfeld, A., 1974. Compact Figures in Digital Pictures. *IEEE Transactions on Systems, Man, and Cybernetics* 4(2), pp. 221–223. [↗](#)
- [Rosenfeld and Ulbrich, 2003] Rosenfeld, D. and Ulbrich, C. W., 2003. Cloud Microphysical Properties, Processes, and Rainfall Estimation Opportunities. *Meteorological Monographs* 30(52), pp. 237–309. [↗](#)
- [Roser and Geiger, 2009] Roser, M. and Geiger, A., 2009. Video-based raindrop detection for improved image registration. In: IEEE International Conference on Computer Vision Workshops 2009 (ICCV Workshops), Kyoto, Japan, pp. 570–577. [↗](#)
- [Roser et al., 2010] Roser, M., Kurz, J. and Geiger, A., 2010. Realistic modeling of water droplets for monocular adherent raindrop recognition using Bézier curves. In: Asian Conference on Computer Vision 2010 (ACCV Workshop), Queenstown, New Zealand, pp. 235–244. [↗](#)
- [Ross, 2000] Ross, O. N., 2000. Optical remote sensing of rainfall micro-structures. Phd, University of Auckland, USA / Freie Universitat Berlin, Germany. [↗](#)
- [Rousseau et al., 2006] Rousseau, P., Jolivet, V. and Ghazanfarpour, D., 2006. Realistic real-time rain rendering. *Computers & Graphics* 30(4), pp. 507–518. [↗](#)
- [Schnell et al., 2004] Schnell, T., Aktan, F. and Lee, Y.-C., 2004. Wet Weather Visibility of Pavement Markings. Technical report, Federal Highway Administration, Iowa City, IA, USA.
- [Schönhuber et al., 1995] Schönhuber, M., Urban, H. E., Baptista, J. P. V. P., Randeu, W. L. and Riedler, W., 1995. Weather radar versus 2D-video disdrometer data. In: International Symposium on Hydrological Applications of Weather Radars, Sao Paulo, Brazil, pp. 159–171. [↗](#)

- [Schor and Tyler, 1981] Schor, C. and Tyler, C., 1981. Spatio-temporal properties of Panum's fusional area. *Vision research* 21, pp. 683–692. [↗](#)
- [Schwarzlmüller et al., 2010] Schwarzlmüller, C., Al Machot, F., Fasih, A. and Kyamakya, K., 2010. A novel Support Vector Machine classification approach involving CNN for raindrop detection. *ISAST Transactions on Computers and Intelligent Systems* pp. 52–65.
- [Sekuler and Ganz, 1963] Sekuler, R. and Ganz, L., 1963. Aftereffect of Seen Motion with a Stabilized Retinal Image. *Science* 139(3553), pp. 419–419. [↗](#)
- [Serra, 1982] Serra, J., 1982. Image analysis and mathematical morphology. [↗](#)
- [Shen et al., 2009] Shen, Y., Ozguner, U., Redmill, K. and Jilin Liu, 2009. A robust video based traffic light detection algorithm for intelligent vehicles. pp. 521–526. [↗](#)
- [Siogkas et al., 2012] Siogkas, G., Skodras, E. and Dermatas, E., 2012. Traffic Lights Detection in Adverse Conditions Using Color, Symmetry and Spatiotemporal Information. In: *International Conference on Computer Vision Theory and Applications (VISAPP 2012)*, Roma, Italy, pp. 620–627. [↗](#)
- [Sugawara et al., 2002] Sugawara, Y., Akanegawa, M., Tanaka, Y. and Nakagawa, M., 2002. Improvement of tracking methods in information providing system using LED traffic lights. In: *International Conference on Communication Systems (ICCS)*, Vol. 2, pp. 1207–1211 vol.2. [↗](#)
- [Szakáll et al., 2010] Szakáll, M., Mitra, S. K., Diehl, K. and Borrmann, S., 2010. Shapes and oscillations of falling raindrops — A review. *Atmospheric Research* 97(4), pp. 416–425. [↗](#)
- [Tadmor and Tolhurst, 1994] Tadmor, Y. and Tolhurst, D., 1994. Discrimination of changes in the second-order statistics of natural and synthetic images. *Vision Research* 34(4), pp. 541–554. [↗](#)
- [Tanaka et al., 2005] Tanaka, Y., Yamashita, A., Kaneko, T. and Miura, K. T., 2005. Restoration of Images Stained with Waterdrops on a Protection Glass Surface by Using a Stereo Image Pair. In: *Proceedings of the IAPR Conference on Machine Vision Applications 2005 (IAPR MVA)*, pp. 152–155. [↗](#)
- [Tolhurst et al., 1992] Tolhurst, D. J., Tadmor, Y. and Chao, T., 1992. Amplitude spectra of natural images. *Ophthalmic & physiological optics : the journal of the British College of Ophthalmic Opticians (Optometrists)* 12(2), pp. 229–32. [↗](#)
- [Tong et al., 2004] Tong, H., Li, M., Zhang, H. and Zhang, C., 2004. Blur detection for digital images using wavelet transform. In: *IEEE International Conference on Multimedia and Expo, 2004 (ICME)*, Vol. 1, Taipei, Taiwan, pp. 17–20. [↗](#)
- [Ulbrich, 1983] Ulbrich, C. W., 1983. Natural Variations in the Analytical Form of the Raindrop Size Distribution. *Journal of Climate and Applied Meteorology* 22(10), pp. 1764–1775. [↗](#)
- [Ulbrich and Atlas, 1985] Ulbrich, C. W. and Atlas, D., 1985. Extinction of Visible and Infrared Radiation in Rain: Comparison of Theory and Experiment. *Journal of Atmospheric and Oceanic Technology* 2(3), pp. 331–339. [↗](#)

- [Van Boxel, 1997] Van Boxel, J. H., 1997. Numerical model for the fall speed of raindrops in a rainfall simulator. In: Proceedings of the International Workshop on Technical aspects and use of wind tunnels for wind-erosion control, Combined effect of wind and water on erosion processes, Ghent, Belgium, pp. 77–85. [↗](#)
- [van de Hulst, 1957] van de Hulst, H., 1957. Light scattering by small particles. [↗](#)
- [Van Droogenbroeck and Talbot, 1996] Van Droogenbroeck, M. and Talbot, H., 1996. Fast computation of morphological operations with arbitrary structuring elements. *Pattern Recognition Letters* 17(14), pp. 1451–1460. [↗](#)
- [van Zyl Marais and Steyn, 2009] van Zyl Marais, I. and Steyn, W. H., 2009. Robust defocus blur identification in the context of blind image quality assessment. *Signal processing. Image communication* 22(10), pp. 833–844. [↗](#)
- [Vidakovic and Mueller, 1994] Vidakovic, B. and Mueller, P., 1994. Wavelets for kids. Technical report. [↗](#)
- [Vidyarthi et al., 2011] Vidyarthi, A., Jassal, B. S. and Gowri, R., 2011. Modeling of rain drop-size distribution for Indian region. In: *IEEE International Conference on Microwave Technology & Computational Electromagnetics 2011 (ICMTCE)*, Vol. 215number 1288, Beijing, China, pp. 350–353. [↗](#)
- [Wada et al., 2005] Wada, M., Yendo, T., Fujii, T. and Tanimoto, M., 2005. Road-to-vehicle communication using LED traffic light. pp. 601–606. [↗](#)
- [Wang et al., 2010] Wang, O., Fuchs, M., Fuchs, C., Davis, J., Seidel, H.-P. and Lensch, H. P. A., 2010. A context-aware light source. In: *IEEE International Conference on Computational Photography 2010 (ICCP)*, Cambridge, MA, USA, pp. 1–8. [↗](#)
- [Wang et al., 2008] Wang, X., Tian, B., Liang, C. and Shi, D., 2008. Blind Image Quality Assessment for Measuring Image Blur. Vol. 1, pp. 467–470. [↗](#)
- [Wetzel et al., 2004] Wetzel, J. M., Sayer, J. R. and Funkhouser, D., 2004. An Examination of Naturalistic Windshield Wiper Usage. Technical report, The University of Michigan, Transportation Research Institute, Ann Harbor, MI, USA. [↗](#)
- [Wu et al., 2009] Wu, S., Lin, W., Xie, S., Lu, Z., Ong, E. P. and Yao, S., 2009. Blind blur assessment for vision-based applications. *Journal of Visual Communication and Image Representation* 20(4), pp. 231–241. [↗](#)
- [Xu et al., 2010] Xu, C., Nai-Qiang, T. and Yan, L., 2010. Traffic lights recognition algorithm based on Lab color space and template match. *Journal of Computer Applications* 30(5), pp. 1251–1254. [↗](#)
- [Yamashita et al., 2008] Yamashita, A., Fukuchi, I., Kaneko, T. and Miura, K. T., 2008. Removal of adherent noises from image sequences by spatio-temporal image processing. pp. 2386–2391. [↗](#)
- [Yamashita et al., 2003] Yamashita, A., Kuramoto, M., Kaneko, T. and Miura, K. T., 2003. A virtual wiper-restoration of deteriorated images by using multiple cameras. In: *IEEE International Conference on Intelligent Robots and Systems 2003 (IROS)*, Vol. 4, Las Vegas, NV, USA, pp. 3126–3131. [↗](#)

- [Yamashita et al., 2005] Yamashita, A., Tanaka, Y. and Kaneko, T., 2005. Removal of adherent waterdrops from images acquired with stereo camera. In: IEEE International Conference on Intelligent Robots and Systems (IROS), Edmonton, AB, Canada, pp. 400–405. [↗](#)
- [Yu et al., 2010] Yu, C., Huang, C. and Lang, Y., 2010. Traffic light detection during day and night conditions by a camera. In: IEEE International Conference on Signal Processing 2010 (ICSP), Beijing, China, pp. 821–824. [↗](#)
- [Zhang et al., 2006] Zhang, X., Li, H., Qi, Y., Leow, W. K. and Ng, T. K., 2006. Rain Removal in Video by Combining Temporal and Chromatic Properties. In: IEEE International Conference on Multimedia and Expo 2006 (ICME), Toronto, Canada, pp. 461–464. [↗](#)

## Algorithmes de vision pour la pluie et les feux tricolores pour les systèmes d'aide à la conduite

**Résumé :** L'utilisation d'algorithmes de vision permettrait d'élargir le domaine d'application des systèmes d'aide à la conduite à d'autres situations telles que : les scènes urbaines ou les conditions météorologiques dégradées. A cette fin, trois nouvelles applications sont étudiées dans cette thèse pour la pluie et les feux tricolores.

La pluie est la condition météorologique dégradée la plus fréquente. Nous comparons les modèles physiques et photométriques existants pour la pluie et les gouttes de pluie. Lors d'une conduite en temps de pluie de jour, les gouttes sur le pare-brise diminuent considérablement la visibilité du conducteur. Lorsqu'elles sont vue par une camera embarquée standard celles-ci apparaissent défocusées. Ainsi, nous proposons de détecter ces gouttes hors-focus en utilisant soit une approche par manque de gradients soit par l'évaluation locale du flou. Lors d'une conduite de nuit sous la pluie, ce sont les phares qui paradoxalement diminuent la visibilité car leur lumière est réfléchiée par les gouttes vers le conducteur. Nous appuyant sur la conception d'un simulateur physique, nous proposons un éclairage adaptatif qui illuminerait la scène sans éclairer les gouttes qui tombent. Les résultats de notre simulateur et le premier prototype construit montre que l'idée avancée pourrait efficacement améliorer la visibilité générale d'une scène. D'autre part, nous étudions la détection et le suivi de gouttes de pluie à grande vitesse.

Les feux tricolores ont un rôle crucial dans la compréhension des scènes urbaines. Bien qu'il existe déjà des systèmes de détection de feux tricolores, les algorithmes actuels ne fonctionnent que dans des conditions simples. Ainsi, nous avons développé un algorithme de détection de feux tricolores qui utilise une détection en niveau de gris des spots lumineux et une classification par reconnaissance de modèle. L'approche ainsi conçue est assez flexible pour détecter différents types de feux tricolores même avec une camera à faible dynamique. Notre proposition a été évaluée sur des séquences acquises en France, Chine et Suisse.

## Vision Algorithms for Rain and Traffic Lights in Driver Assistance Systems

**Abstract:** Vision algorithms can be used to expand the working range of the assistance systems so as to deal with urban scenes or degraded weathers. To this end, three novel applications are investigated in this thesis for both rain and traffic lights.

Rain is the most frequent degraded weather condition. We review the various physics and photometry models for rain and raindrops, and highlight some of the misuses. When driving in daytime the raindrops on the windscreen lower the driver visibility. For standard on-board camera these drops appear as unfocused. Hence, we investigate the detection of unfocused raindrops using blur maps or lack of gradients with photometry. For nighttime driving in rain, the headlights paradoxically reduce the visibility due to light reflected off of raindrops back toward the driver. Relying on a physic-based simulator, we propose to build an illumination device that would illuminate the scene without shining the falling particles. The performance of the simulator and a proof-of-concept prototype sustain that our idea can efficiently improve the overall scene visibility. Fast reactive drops detection and tracking is also investigated.

To deal with urban scenes, traffic lights play a key role. Though traffic light recognition was attempted in the past, the existing algorithms can't handle complex scenarios. Hence, we have developed a traffic light recognition algorithm that uses a grayscale spot light detection and a template matching classification. Our approach is modular and capable of detecting various kind of traffic lights even when using a low-dynamic camera. We have evaluated our algorithm on sequences from France, China and Switzerland.

