IR ILLUMINATION-ASSISTED SMART HEADLIGHT GLARE REDUCTION

Thesis
Submitted to
The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree of
Master of Science in Electrical Engineering

By
Larry Sanders
Dayton, Ohio
December, 2017
IR ILLUMINATION-ASSISTED SMART HEADLIGHT GLARE REDUCTION

Name: Sanders, Larry Dean

APPROVED BY:

Keigo Hirakawa, Ph.D.
Advisory Committee Chairman
Associate Professor, Electrical and Computer Engineering

Vijayan Asari, Ph.D.
Committee Member
Professor, Electrical and Computer Engineering

Eric Balster, Ph.D.
Committee Member
Associate Professor, Electrical and Computer Engineering

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and Innovation Professor
School of Engineering

Eddy Rojas, Ph.D., M.A., P.E.
Dean, School of Engineering
ABSTRACT

IR ILLUMINATION-ASSISTED SMART HEADLIGHT GLARE REDUCTION

Name: Sanders, Larry
University of Dayton

Advisor: Dr. Keigo Hirakawa

Smart headlights, or headlights with some form of programmable technology, are rapidly being incorporated in today’s automotive world. Recently proposed smart headlights utilize projector-like displays to fully control the light beam. One application of this technology is the reduction of reflected glare from a driver’s headlights illuminating precipitation. Specifically, the headlight de-illuminates the rain drops and snowflakes detected by integrated cameras, allowing the driver to better focus on the road. Up until now, these efforts have been developed in the visible spectrum only. In this work, we propose a new hardware configuration aimed at improving the rain glare reduction system by leveraging infrared (IR) illumination. We demonstrate a better light efficiency, the increased ability to detect and track rain, increased stability, and reduced computational load.
To my Lord and Saviour,

With whom all things are possible
# TABLE OF CONTENTS

ABSTRACT ....................................................... iii

DEDICATION .................................................... iv

LIST OF FIGURES .............................................. vii

LIST OF TABLES ................................................ ix

I. INTRODUCTION ............................................. 1

II. BACKGROUND AND RELATED WORK ...................... 3
   2.1 Notational Conventions ................................. 3
   2.2 Smart Headlights ....................................... 3
   2.3 Infrared (IR) Illumination ............................. 4
   2.4 Homography ............................................. 5
   2.5 Lens Distortion Removal ............................... 7
   2.6 Related Work: CMU Smart Headlight ................. 9

III. HARDWARE DESIGN ....................................... 12
   3.1 Proposed Smart Headlight Design ................... 12
   3.2 Hardware Prototyping ................................ 13
   3.3 Calibration ............................................. 19
IV. SOFTWARE DESIGN ............................................... 21
  4.1 Software Development and Operation .............................. 21
  4.2 Image Processing .............................................. 23
  4.3 Blob Identification ........................................... 26
  4.4 Fish-Eye Correction ........................................... 28
  4.5 Blob Tracking ................................................ 29
  4.6 Anti-Blob Location Determination and Display ................. 30

V. EXPERIMENTS ..................................................... 32
  5.1 Large Objects ................................................ 32
  5.2 Single Drop Rain .............................................. 33
  5.3 Multiple Drop Rain ............................................ 34
  5.4 Limitations .................................................... 36

VI. CONCLUSIONS ................................................... 37

BIBLIOGRAPHY ...................................................... 38
LIST OF FIGURES

2.1 Quantum efficiency for Point Grey camera [1] . . . . . . . . . . . . . . 4
2.2 Example layout (top view) of co-located camera and projector . . . . . 5
2.3 Graph of fish-eye correction factor curves vs. the distance from the center in pixels of the undistorted image . . . . . . . . . . . . . . . . 9
2.4 Calibration images with strength values corresponding to Figure 2.3 . 9
2.5 CMU basic design structure [2] . . . . . . . . . . . . . . . . . . . . . 10
2.6 Progression of CMU design of SLM de-illumination of rain drop . . 11

3.1 Proposed setup to decouple SLM and detection camera systems. Note IR light in red and visible light in blue . . . . . . . . . . . . . . . . . 13
3.2 Final hardware prototype . . . . . . . . . . . . . . . . . . . . . . . . 14
3.3 Transmittance curve for visible cut filter [3] . . . . . . . . . . . . . . 15
3.4 Transmittance and reflectance curve for cold mirror [4] . . . . . . . 16
3.5 Diagram of system latency test . . . . . . . . . . . . . . . . . . . . . 16
3.6 Hardware configurations to show captured imagery with a coupled and decoupled system. Gamma correction has been applied . . . . . . 18
3.7 Progression of calibration image calculation . . . . . . . . . . . . . . 19
3.8 Distortion corrected calibration image and image with re-calculated display corners . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

4.1 Rain drop tracking algorithm software diagram . . . . . . . . . . . . 22
4.2 Example progression of image processing using captured data of a rain drop from the detection system. The Figure progression from (a) to (e) matches the system progression.

4.3 Example of a blob to be detected

4.4 Example of blob detection, part one.

4.5 Example of blob detection, part two.

4.6 Example of blob detection, part three.

4.7 Image of rain drops from detection system. Yellow markers indicate a blob that meets identification requirements.

4.8 System diagram of blob tracking.

5.1 Tin foil balls used in initial system testing.

5.2 Progression of tin foil ball de-illumination.

5.3 Eye-dropper used for single rain drop testing.

5.4 Progression of single rain drop de-illumination.

5.5 Items used to create multiple rain drops.

5.6 Example of rain de-illumination system in operation.
LIST OF TABLES

4.1 Execution times for the tracking algorithm’s sections. . . . . . . . 22
CHAPTER I
INTRODUCTION

The goal of this research project is to establish a proof-of-concept of IR illumination-assisted glare reduction for smart headlights. When driving at night, headlights are used to increase the visibility both for the driver and other drivers on the road. This presents a problem when precipitation coincides with driving, however, since the same headlights used to increase the driver’s visibility also produce a backscatter illumination from the rain drops and snowflakes. During heavy precipitation, this is not only a distraction but it inhibits the driver’s ability to see the road or potential hazards.

A seminal solution to this problem has already been explored by Carnegie Mellon University using programmable automotive headlights. In their solution, CMU uses a digital micromirror device (DMD) as a spatial light modulator (SLM) to spatially modulate the headlight in a desired pattern, similar to a projector display. This enables control over the regions of the scene that are illuminated by the headlight via the pixel output. When combined with a machine vision camera capable of detecting rain drop locations and software capable of tracking the rain, the local regions of the headlights that would illuminate rain drops may be turned off [2]. As a result, the visibility of the driver is enhanced.

In this thesis, we improve upon the CMU solution by incorporating IR illumination and an IR sensitive camera. The inclusion of IR illumination allows for the detection
camera and SLM to be decoupled. The decoupling allows the detection camera to operate in the IR wavelength range while the SLM operates in the visible wavelength range. With a decoupled system, several benefits can be leveraged. First, the tracking system experiences less interference from the SLM. With an IR illumination source, filters can be used to remove any noise introduced by the SLM. Second, decoupling allows for a simpler tracking algorithm. When decoupled, the tracking system is robust to perturbation introduced by the SLM. The tracking algorithm does not need to be aware of the SLM’s state to successfully track and de-illuminate drops. Lastly, decoupling allows for greater light efficiency. To co-locate the decoupled detection camera and SLM, a cold mirror is used instead of a traditional beamsplitter solution. The cold mirror transmits nearly 100% of IR and reflects nearly 100% of visible light [4]. This provides a higher fidelity image for both the tracking camera and the spatial modulator.

This project demonstrates that using IR to decouple the detection and spatial modulation systems provides a simpler and more effective solution to reduce glare from precipitation.
CHAPTER II
BACKGROUND AND RELATED WORK

2.1 NOTATIONAL CONVENTIONS

In this thesis, the scalar variables will be denoted by a lowercase letter (e.g. $a$). A vector will be bolded lowercase letter (e.g. $\mathbf{a}$). The bolded uppercase letter is used for matrices (e.g. $\mathbf{A}$).

2.2 SMART HEADLIGHTS

In consumer markets, there are already precedents for smart headlights, or headlights with some programmable features. However, there are no currently implemented systems with the complexity of that proposed in this project.

A simple but notable example is the adaptive headlights on BMW motorcycles [5]. The motorcycle uses sensors to detect if the motorcycle is turning and can steer the headlight beam into the turn direction. This allows for earlier detection of road hazards during turns.

Another example is the matrix LED headlight developed by Audi [6]. The headlight developed by Audi is comprised of several smaller headlights arranged in a row. Sensors in the vehicle can detect oncoming vehicles or potential high-reflectance objects and reduce or remove illumination towards those sources. This allows for brighter lights, such as hi-beams, to be used more frequently thus increasing driver visibility.
2.3 INFRARED (IR) ILLUMINATION

Infrared (IR) light refers to electromagnetic radiation whose wavelength exceeds visible ranges. In this work, we are interested in the 700-1000 nanometers range, commonly referred to as “near infrared” or NIR. The infrared light wavelength is longer than what the human eye can detect and creates an invisible light source. This can be exploited in many ways. A simple but effective example is night vision. Night vision systems use cameras which can detect the NIR or IR wavelengths from IR illuminations sources. The output is then converted into a visible spectrum representation to be used by a human observer.

Most consumer cameras have CMOS or CCD image sensors that are sensitive to NIR. Figure 2.1 shows the quantum efficiency of a typical CMOS camera, spanning 380nm to 1000nm. Hence, in an ordinary imaging setup where the goal is to capture the scene that matches human observation, cameras incorporate IR cut filters inside to prevent the IR or NIR illumination sources from interfering with imaging. For our system, however, we make use of this sensitivity of CMOS cameras to IR.

IR light also allows for different beamsplitter options for co-location. Figure 2.2 is
a simplified example of a co-located camera and SLM using a beamsplitter. When the two devices are operating in the same wavelength range, the beamsplitter is used to divide the total incoming illumination to multiple directions. This comes at the sacrifice of light efficiency. For example, suppose a 50/50 beamsplitting mirror was used that operates at 45 degrees. For either a camera or projector only half the illumination (either into the camera or out of the projector) would reach its intended target. To remove this problem, a hot or cold mirror may be used. Hot mirrors reflect IR and transmit visible light while cold mirrors reflect transmit IR and reflect visible light. Now consider co-locating two cameras, one operating in IR and another in visible range. Both cameras can operate with nearly 100% efficiency in their respective wavelength ranges.

2.4 HOMOGRAPHY

Pixel coordinates are the projection of a three-dimensional scene onto a two-dimensional plane. When two or more imaging devices are used, their pixel coordinates do not cor-
respond to the same point in the scene, because of the differences in the projections. Homography is used to calculate the pixel coordinate in the reference imaging device corresponding to the equivalent pixel location in the target imaging device. Homography is extensively used in panoramic imaging, for example. The coordinates from target images are mapped to the reference image to blend multiple images together.

Figure 2.2 shows a general layout of a camera and projector system requiring co-location. Here, the projector or spatial modulator creates an image (for example, when projected against a wall) that is observable by the camera. It is required to know what the pixel coordinate in the SLM domain is when given the pixel coordinate in the camera domain. A simple but effective way to co-locate the camera and projector is to create and use a homography matrix.

Suppose \((c, r)\) and \((j, k)\) refer to column-row pixel coordinates of two imaging devices observing (if camera) or illuminating (if projector) the same scene. The homography matrix \(H\) defines the relationship between \((c, r)\) and \((j, k)\) coordinates by the following relation from [7]:

\[
\begin{bmatrix}
    j \cdot \text{scale} \\
    k \cdot \text{scale} \\
    \text{scale}
\end{bmatrix} =
\begin{bmatrix}
    h_{11} & h_{12} & h_{13} \\
    h_{21} & h_{22} & h_{23} \\
    h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{bmatrix}
    c \\
    r \\
    1
\end{bmatrix}.
\]

The homography matrix \(H\) can be found from pairs of known correspondences between \((c, r)\) and \((j, k)\) coordinates. These correspondences are typically established by some calibration experiments (e.g. displaying a checkerboard pattern to find the corners). Suppose it was found that pixel coordinates \((c_1, r_1)\)...

in one imaging device...
correspond to the pixel coordinates \((j_1, k_1)\)... in another imaging device. Define the following:

\[
A = \begin{bmatrix}
    c_1 & r_1 & 0 & 0 & 0 & -c_1 * j_1 & -r_1 * j_1 & -j_1 \\
    0 & 0 & 0 & c_1 & r_1 & 1 & -c_1 * k_1 & -r_1 * k_1 & -k_1 \\
    c_2 & r_2 & 0 & 0 & 0 & -c_2 * j_2 & -r_2 * j_2 & -j_2 \\
    0 & 0 & 0 & c_2 & r_2 & 1 & -c_2 * k_2 & -r_2 * k_2 & -k_2 \\
    c_3 & r_3 & 0 & 0 & 0 & -c_3 * j_3 & -r_3 * j_3 & -j_3 \\
    0 & 0 & 0 & c_3 & r_3 & 1 & -c_3 * k_3 & -r_3 * k_3 & -k_3 \\
    c_4 & r_4 & 0 & 0 & 0 & -c_4 * j_4 & -r_4 * j_4 & -j_4 \\
    0 & 0 & 0 & c_4 & r_4 & 1 & -c_4 * k_4 & -r_4 * k_4 & -k_4
\end{bmatrix}, \quad h = \begin{bmatrix}
    h_{11} \\
    h_{12} \\
    h_{13} \\
    h_{21} \\
    h_{22} \\
    h_{23} \\
    h_{31} \\
    h_{32} \\
    h_{33}
\end{bmatrix}
\]

where \(h\) is a vectorized version of the original homography matrix \(H\). We solve for the vector \(h\) that minimizes the algebraic residuals of \(\text{norm}(Ah)\) subject to \(\text{norm}(h)\) [7]. The solution to optimal \(h\) is the Eigen vector corresponding to the smallest eigenvalue of \(A^T A\) [7]. Note \((j, k)\) are then found by dividing the results \((j \cdot \text{scale}, k \cdot \text{scale})\) by the scale factor.

2.5 LENS DISTORTION REMOVAL

Fish-eye distortion refers to a geometric warping introduced by a lens with a convex shape. Often fish-eye lenses are used to increase field of view. It is difficult to correct for a fish-eye distortion without the proper parameterization for the distortion. Additionally, geometric distortions are corrected using interpolation, which is often too computationally complex to be used in real time. Below, we review the fish-eye
The correction method proposed in [8].

To correct for the fish eye distortion, let there be an image size given by \( \text{cols} \) (number of columns) and \( \text{rows} \) (number of rows). Let the following be defined:

\[
\begin{align*}
  r_c &= \frac{\text{rows}}{2} \quad (2.3) \\
  c_c &= \frac{\text{cols}}{2}. \quad (2.4)
\end{align*}
\]

Let the following distances be defined with the parameter \( \text{strength} \) and a given pixel location in the \textit{corrected} image \( r_a \) and \( c_a \):

\[
\begin{align*}
  d &= \sqrt{(r_a - r_c)^2 + (c_a - c_c)^2} \quad (2.5) \\
  s &= \sqrt{\text{cols}^2 + \text{rows}^2} \quad \frac{\text{strength}}{2} \quad (2.6)
\end{align*}
\]

From the distance values \( d \) and \( s \) let the correction factor \( \theta \) be defined as follows:

\[
\theta = \frac{\text{atan}(d/s)}{(d/s)}. \quad (2.7)
\]

Lastly, the equivalent locations \( r_b \) and \( c_b \) in the \textit{distorted} image are found as:

\[
\begin{align*}
  r_b &= r_c + \theta \cdot (r_a - r_c) \quad (2.8) \\
  c_b &= c_c + \theta \cdot (c_a - c_c). \quad (2.9)
\end{align*}
\]

The inverse tangent operator is used in the correction factor since it ensures a smaller radian value than the input ratio value. The rate at which the correction factor decays depends on the \( \text{strength} \) value. A smaller strength value results in a weaker correction while a larger \( \text{strength} \) results in a stronger correction. Figure 2.3 shows a plot of
Figure 2.3: Graph of fish-eye correction factor curves vs. the distance from the center in pixels of the undistorted image.

(a) strength = 0.1  (b) strength = 0.9  (c) strength = 1.5

Figure 2.4: Calibration images with strength values corresponding to Figure 2.3

the correction factor for the strength values of 0.1, 0.9, and 1.5. The calculations were based on a 640 column by 512 row image. Figure 2.4 shows several example calibration images, each with a different strength value. The strength value of 0.9 was found as the best choice for this project.

2.6 RELATED WORK: CMU SMART HEADLIGHT

As mentioned earlier, Carnegie Mellon University introduced the notion of a smart headlight that incorporates a spatial light modulator. A basic system design is shown in Figure 2.5. The CMU design has three main components: A camera, a Spatial
Figure 2.5: CMU basic design structure [2].

Light Modulator (SLM) and a beamsplitter. Like the example shown in Figure 2.2, the beamsplitter allows for the co-location of the SLM and the camera. The co-location allows the camera to operate effectively in-line with the SLM. The system can run at or above 1,000 Hz with a 1 to 1.5ms latency [2]. The system works by first processing images of the scene observed by the camera then a software algorithm separates rain drops from the background. The processor then predicts the rain location at the next time instance, and the SLM turns off the pixels that produce a glare from the identified rain drops.

The limitation of the CMU design is that the SLM and detection camera are coupled. Coupling has two adverse effects. The first is an interdependence between the captured image from the detection camera and the SLM. The camera relies on illumination provided by the SLM, thus it depends on the SLM’s state. For example, upon a successful de-illumination of a rain drop, the camera can no longer detect the rain drop’s location. The system must then wait for the drop to re-appear from out of the de-illuminated area before updating the SLM. Figure 2.6 is a simplified example of how the de-illumination works. Figure 2.6 (a) shows the initial rain drop that will be detected. Figure 2.6 (b) shows the SLM de-illuminating the rain drop in Figure 2.6 (a). Figure 2.6 (c) shows the rain drop begin to exit the de-illuminated area. Figure
2.6 (d) shows the update to the SLM after re-identifying the drop in Figure 2.6 (c). The progression in Figure 2.6 shows a penalty induced by re-acquisition of the rain drop. During this time, the rain causes a glare again until the system re-acquires the rain drop.

The second adverse effect of the system coupling is light efficiency. When a 50/50 beamsplitter is used, only half of the total illumination reaches the intended target. Additionally, upon return to the 50/50 beamsplitter, the arriving illumination is halved again. This results in a quarter of the total illumination reaching the detection camera.
3.1 PROPOSED SMART HEADLIGHT DESIGN

The goal of the hardware in this project is to decouple the spatial modulator from the detection camera. With a successful decoupling, any action from the spatial modulator will not interfere with the tracking image and thus will not interfere with the rain detection. The use of IR illumination also improves the light efficiency of the visible wavelength headlight as well as the camera. The proposed hardware configuration is shown in Figure 3.1. It is comprised of the following components:

1. IR illumination source
2. Visible spectrum spatial light modulator
3. CCD or CMOS camera that is sensitive to NIR
4. Visible light cut filter for the detection camera
5. Cold mirror that reflects visible wavelength and transmits IR

In Figure 3.1, visible light (blue) is reflected off the Cold mirror. This allows for the spatial light modulator to operate in the forward direction. Additionally, the IR light (red) can pass through the cold mirror to the detection camera. A visible cut filter is added to the detection camera to prevent any visible light from entering the camera. This is useful because it not only cuts down ambient visible light, but the projector
Figure 3.1: Proposed setup to decouple SLM and detection camera systems. Note IR light in red and visible light in blue.

Illuminator may have some IR leakages that can be seen by the detector. Moreover, the cold mirror is also not perfect—a blob of light (presumably from diffusion) can be observed at the cold mirror surface, which can interfere with the camera. With this setup, a co-location is still possible while the SLM and detection camera are decoupled.

3.2 HARDWARE PROTOTYPING

To guide the design of a hardware prototype, the following assumptions were made:

1. The typical distance from a driver to their headlight is approximately 3 - 4 feet.

2. The rain drops which produce the most glare occur with 1 - 3 feet from the headlight.

Figure 3.2 shows the hardware setup using the final components. Note that the figure numbers match those of Figure 3.1.

For the IR illumination devices (item 1 in Figure 3.2), two Univivi U06R illuminators
were used. Both illuminators operated at 850 nanometers which is in the NIR range. One of the illuminators was a 60-degree output angle and another was a 90 degree. Eventually, the cover was removed from the 60-degree lamp to remove an IR cut effect in the glass.

For the Spatial Light Modulator (item 2 in Figure 3.2), a Texas instruments DLP4500 evaluation module was used. The DLP4500 is a digital micromirror device capable of a 120Hz 8-bit grayscale refresh rate with 150 Lumens intensity [9]. Additionally, the DLP4500 can convert an incoming 24-bit RGB data stream into 24 bit-planes to achieve a refresh rate of 2880Hz. However, the bit planes are only updated every 120 Hz. This means that to achieve the 2880Hz refresh rate, we must effectively predict rain drop location 24 future frames at a time (albeit the frames are short and therefore not as challenging). This would allow for a hybrid refresh rate that could effectively reach beyond 120Hz. For this project, the projector was set to pattern sequence mode, operating at 120Hz. However, the bit plane interpolation was not utilized. The output resolution was set to 912R x 912C.

For the detection camera (item 3 in Figure 3.2), a point grey Chameleon3 1.3 MP
Figure 3.3: Transmittance curve for visible cut filter [3].

mono camera was used. The camera features a resolution of 1024R X 1280C at 150Hz or a resolution of 512R x 640C at 400Hz [1] and was connected to a host computer via USB 3.0. Due to a need for high refresh rates, the lower resolution mode was used. Additionally, this camera did not have an IR cut filter installed and could detect the 850 nanometer light emitted by the IR illuminators. A graph of the camera’s quantum efficiency is shown in Figure 2.1.

For the visible cut filter (item 4 in Figure 3.2), an FGL780S was used from Thorlabs. The filter transmits light with a wavelength longer than 800 nanometers with approximately 80% efficiency. A plot of the transmittance is shown in Figure 3.3. For the cold mirror (item 4 in Figure 3.2), a cold mirror from Edmunds Optics was used. The mirror transmits 90% of IR and reflects 90% of visible light below 700nm. A plot of the transmittance is shown in Figure 3.4.

To measure the system latency of the system, the SLM was cycled fully on and off while the camera sampled the illuminated scene. A diagram of the latency measurement is shown in Figure 3.5. The latency was measured as the time between the cycle command and the camera observation of the cycle. The measured latency was approximately 55 ms. The latency variation is due to the Windows operating system.
Once all the hardware choices were made, various configurations were tested to observe the improvement. Figure 3.6 shows the tested configurations (note that gamma correction has been applied). Figure 3.6 (a) uses a 50/50 mirror without any IR illumination or visible cut filter, and is comparable to the CMU system in [2]. Notice the diffusion from the mirror and interference from the side of the mirror. Figure 3.6 (b) illustrates the best decoupling possible without using a cold mirror. Note that for Figure 3.6 (c) through (f) IR illumination is present. Figure 3.6 (d) illustrates the diffusion present in the cold mirror from the visible SLM. Figure 3.6 (e) shows the final configuration with the cold mirror and visible cut filter installed. Figure 3.6 (e) vs. (d) shows very little difference when the SLM is in operation, thus illustrat-
ing a successful hardware decoupling. Lastly, Figure 3.6 (f) shows an example of IR illuminated rain using the system in Figure 3.6 (e).
Figure 3.6: Hardware configurations to show captured imagery with a coupled and decoupled system. Gamma correction has been applied.
3.3 CALIBRATION

To co-locate the camera and SLM, the cold mirror was replaced by a 50/50 beam-splitting mirror and the visible cut filter was removed. This was done so the detection camera can observe a calibration pattern displayed by the SLM. Otherwise, since the two systems are decoupled, the detection camera, which operates in the IR domain, could not observe the calibration pattern displayed by the visible domain SLM. Once the hardware was setup, the SLM set all pixels to their max value (255 for an 8-bit value) to capture a fully illuminated background image (Figure 3.7 (a)). The calibration image was displayed on a surface approximately two and a half feet from the projector. After the background image was displayed, a checkboard pattern was displayed so that corners could be identified (Figure 3.7 (b)). This is beneficial since all the display image corner coordinates are known and can be used to calculate the homography matrix shown in Section 2.4. Once the background and checkerboard images were captured, the final calibration image was calculated by dividing the checkerboard image element-wise by the background image (Figure 3.7 (c)). This has the effect of removing the background from the checkerboard pattern, preventing the scene from obscuring the calibration pattern. Once the final calibration image was calculated, the fish-eye distortion from the lens was removed using the algorithm.

![Figure 3.7: Progression of calibration image calculation](a) Background (b) Checkerboard (c) Final)
shown in Section 2.5 (Figure 3.8 (a)). To calculate the homography matrix, four corners were chosen in the calibration image and their coordinates recorded. The matching checkerboard pattern corner coordinates were then identified and both sets of coordinates were put into the form shown in Section 2.5 and the resulting homography matrix was calculated. Lastly, to check for a correct calibration, the pre-defined projector coordinates were converted to the calibration image domain. These new points are then plotted over the calibration image as seen in Figure 3.8 (b) as red X marks.
CHAPTER IV
SOFTWARE DESIGN

4.1 SOFTWARE DEVELOPMENT AND OPERATION

Once the hardware was successfully decoupled, a software algorithm to locate and de-illuminate rain was developed. The algorithm was required to have some noise tolerance and to operate approximately two to three feet away from the cold mirror surface. A diagram of the software operation is shown in Figure 4.1. Although the figure shows a single frame in operation, the algorithm will run repeatedly for a specified number of frames. The software ran on a Windows 10 machine using an Intel Xeon CPU and Nvidia GeForce GTX 780 GPU.

Since the detection system and display systems were decoupled, they were not required to have a locked refresh rate in relation to each other. The goal refresh rate for the detection software was 240Hz (twice the 120Hz projector refresh rate). The execution times for the software sections are shown in Table 4.1. As the Table shows the detection system was operating at approximately 250 Hz which satisfies the refresh rate goal of 240 Hz.
Figure 4.1: Rain drop tracking algorithm software diagram.

<table>
<thead>
<tr>
<th>Step</th>
<th>Mean, ms</th>
<th>Variance, ms²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Aq.</td>
<td>1.17</td>
<td>0.0394</td>
</tr>
<tr>
<td>Processing</td>
<td>1.30</td>
<td>0.0006</td>
</tr>
<tr>
<td>Blob Ident.</td>
<td>0.30</td>
<td>0.0001</td>
</tr>
<tr>
<td>Blob Track and Disp.</td>
<td>1.15</td>
<td>0.0520</td>
</tr>
<tr>
<td>Sys. Reset</td>
<td>0.10</td>
<td>0.0001</td>
</tr>
<tr>
<td>Total</td>
<td>4.02</td>
<td>0.0956</td>
</tr>
</tbody>
</table>
4.2 IMAGE PROCESSING

The image processing consisted of the following steps, in order of execution:

1. Minimum pixel value threshold
2. Subtraction of previous input image with current input image
3. Calculation of integral image based on subtracted image
4. Convolution of subtracted image with a specified window size

Prior to any image processing, every pixel in the current image is checked for a minimum value. This allows for a smaller minimum difference between the current and previous image during the image subtraction. A smaller minimum difference allows the tracking algorithm to be more sensitive. The convolution of the subtracted image with a window is to reduce false positives. Often, single, non-connected pixels will remain after image subtraction. If these pixels are not removed, they are treated as positives (rain drops) and thus potentially causing a false track.

Figure 4.2 shows the progression of rain drop throughout the processing algorithm. Figure 4.2 (a) shows the previous image and Figure 4.2 (b) shows the current rain drop image. Note that all the pixel coordinates in Figure 4.2 absolutely match from Figure 4.2 (a) to (e). Also, the images in Figure 4.2 (a) and (b) have been scaled by dividing by the maximum pixel value to increase visibility. The original maximum pixel value of the current image Figure 4.2 (b) was 21 out of 255.

The first step of the processing algorithm occurs from (b) to (c) in Figure 4.2 where the current image Figure 4.2 (b) is threshold by a minimum pixel value to produce the image in Figure 4.2 (c). Let $Q$ represent the current image in (b) and let $P$
Figure 4.2: Example progression of image processing using captured data of a rain drop from the detection system. The Figure progression from (a) to (e) matches the system progression.

Represent the previous image in Figure 4.2 (a). The image $U$ is formed as follows:

$$U = \begin{cases} 
1, & Q \geq \text{min pixel value} \\
0, & Q < \text{min pixel value},
\end{cases} \quad (4.1)$$

where $U$ is a mask of all pixels with the required minimum threshold (Figure 4.2 (c)). Let $V$ represent the subtraced image (Figure 4.2 (d)) and be formed as follows:

$$V = \begin{cases} 
1, & (U \cdot (Q - P)) \geq \text{min difference} \\
0, & (U \cdot (Q - P)) < \text{min difference}.
\end{cases} \quad (4.2)$$

The process to calculate the convolved image (Figure 4.2 (e)) is a two-step process: first, calculate an integral image over the image in Figure 4.2 (d) and second, check for a minimum amount of “on” pixels in a specified window size. Let the integral image $I$ be the same size as $V$ and be initialized to all 0’s. $I$ is found using the
method in [10], assuming a 0-based index:

\[
I = \sum_{r=1}^{n} \sum_{c=1}^{m} V(r, c) + I(r, c - 1) + I(r - 1, c) - I(r - 1, c - 1).
\] (4.3)

In the previous equation, the term \(I(r - 1, c - 1)\) is subtracted since that value is contained both the \(I(r, c - 1)\) and \(I(r - 1, c)\) terms and is added twice.

Lastly, let the window convolution image \(W\), be the same size as \(V\) and be initialized to all 0’s. Also, let an odd, square window size be specified no smaller than 3 as width and the window distance from the center pixel be defined as \(t\). Let the row and column bounds be defined as follows:

\[
t = (\text{width} - 1)/2
\] (4.4)

\[
r_s = t + 1
\] (4.5)

\[
c_s = t + 1
\] (4.6)

\[
r_e = \text{rows} - t
\] (4.7)

\[
c_e = \text{columns} - t.
\] (4.8)

Let the convolution image \(W\) be found as follows:

\[
W = \sum_{r=r_s}^{r_e} \sum_{c=c_s}^{c_e} I(r + d, c + d) + I(r - d - 1, c - d - 1)
- I(r + d, c - d - 1) - I(r - d - 1, c + d).
\] (4.9)

Lastly, every element in \(W\) is checked for a minimum pixel value and if so a 1 is saved to that location, otherwise a 0 is saved. Once the convolution image is complete, we proceed to blob identification, explained in the next section.
4.3 BLOB IDENTIFICATION

The goal of the blob identification step is to search the convolution image for blobs and determine their centers. Note that a blob refers to a set of connected “on” pixels. For speed, it was desired that the blob detection only search the convolution image once and not rely on future values. From these requirements, a simple state-machine pixel tracking algorithm was developed to identify blobs and track their statistics. To track the statistics, a table is used that tracks the number of pixels in the blob, the sum of the row and column values (to compute the mean values) and if the blob has been connected to another blob. An example blob is given in 4.3 for reference. The algorithm processes one pixel at a time in a column-wise direction. If the pixel is not on at the current location, the algorithm has a state of 0. If an on pixel is found and the pixel above the current pixel is not on, the state becomes 1. If, when the current state is 1 an off pixel is located, all the previous row of on pixels are set to a unique index value. This is illustrated in Figure 4.4. Since no pixels beyond the current row
are seen the algorithm treats the blobs 1 and 2 as independent until the blobs have at least one neighboring pixel. If when a current pixel is on the above pixel is also on, the algorithm checks if the blob containing the above pixel has been connected to another blob. If the above pixel’s blob has not yet been connected, the statistics from the above pixel’s blob are added to the current blob. Also, a flag is set in the above pixel’s blob to indicate it has been connected to another blob and the connected blob’s index value is saved. This is illustrated in Figure 4.5. It is possible that once a blob has connected to another blob, more pixels are still connected to the now irrelevant blob. This is shown in Figures 4.5 and 4.6. Initially, the blob with Index three is connected to the blob with index 1. However, blob 3 has pixels that connect to blob 3 but not directly to blob 1. When this happens, the algorithm checks if the above blob (3) is connected to any other blob (1). If so, the algorithm copies the connected index (1) and assigns that index value to the current pixel and adds the statistics to the correct blob (1). Figure 4.6 shows that three unique blobs may exist but only one blob will have a “good” flag indication. Lastly, after the algorithm has searched the entire image, the table is searched for the remaining good blobs for processing.

An example image of rain drop blobs being detected is shown in Figure 4.7. The figure shows several rain drops that have been identified with yellow X’s marking
Figure 4.6: Example of blob detection, part three.

Figure 4.7: Image of rain drops from detection system. Yellow markers indicate a blob that meets identification requirements.

centers identified by the detection algorithm. Note that the left portion of the rain drops were purposely not tracked due to a high glare spot in the testing location.

4.4 FISH-EYE CORRECTION

To apply the fish-eye correction a polynomial curve fit was calculated for the correction factor for an input distance away from the image center. This was done because the method outlined in Section 2.5 calculates the location in the distorted image based on
the undistorted pixel location [8]. However, the blob center is given in the distorted image and needs to be transformed to the undistorted domain. To calculate the polynomial coefficients, distances from the image center in the undistorted image were calculated based on a 512R x 640C image size. The maximum distance from center is $\sqrt{(512/2)^2 + (640/2)^2} = 409.8$ pixels. Based on the maximum distance, a vector of distance values was constructed. The correction factors and corresponding distances in the distorted image were calculated and recorded. The distance in the distorted image was used as the input for an output of a correction factor. The final equation to convert a distance in the distorted image to a correction factor is given by:

$$distance = \sqrt{(blob \ column - 320)^2 + (blob \ row - 256)^2}$$ (4.10)

$$\theta = 1.66e^{-10}(distance^3) - 4.78e^{-7}(distance^2) + 2.95e^{-4}(distance) + 0.93.$$ (4.11)

Once the correction factor is calculated, the coordinates within the undistorted plane are calculated as follows:

$$column = (distorted \ column - 320 - \theta \cdot 320)/\theta$$ (4.12)

$$row = (distorted \ row - 256 - \theta \cdot 256)/\theta.$$ (4.13)

### 4.5 BLOB TRACKING

The goal of blob tracking was to establish a correspondence between the rain drops in the previous frame and the rain drops in the current frame. Due to the testing scenarios (which are explained in more detail in the Experiments chapter) the blob tracking was kept simple but was designed to be modular for future improvements.
For example, currently for a given previous blob, the current blob with the shortest distance from the previous blob is “tracked”. However, a different metric such as matching covariance matrices and eigen values could be used instead. A system diagram of the tracking solution is given in Figure 4.8. For each previous blob, the distance is calculated between the previous blob and every untracked current blob within an applicable window. The window is based on the previous blobs centroid column and row values. The window is any blobs with equal or less row value and within +/- a defined column range.

4.6 ANTI-BLOB LOCATION DETERMINATION AND DISPLAY

An anti-blob is a rectangular local region of “off” pixels to de-illuminate a rain drop. Given a tracked rain drop in the current image, the location of the anti-blob within the detection camera plane is the centroid value of the tracked blob. To transform the anti-blob location from the camera domain into SLM domain the homography matrix is applied. Using the homography matrix defined in Section 2.4, the equations
to calculate the projector coordinates are given as follows:

\[
\text{scale} = h_{31} \cdot \text{blob column} + h_{32} \cdot \text{blob row} + h_{33}
\]  

\[
\text{SLM column} = (h_{11} \cdot \text{blob column} + h_{12} \cdot \text{blob row} + h_{13}) / \text{scale}
\]  

\[
\text{SLM row} = (h_{21} \cdot \text{blob column} + h_{22} \cdot \text{blob row} + h_{23}) / \text{scale}.
\]

Lastly, when the correct anti-blob location has been calculated using the homography matrix, the anti-blob is applied. The anti-blob size is determined by a fixed width and height rectangle of off pixels. The rectangle may or may not be centered on the anti-blob location. Offset variables control the location of the anti-blob rectangle with relation to the anti-blob location.
5.1 LARGE OBJECTS

For verification, debugging, and system optimization, tin foil balls were used to simplify the testing parameters. The balls were approximately one inch in diameter and were dropped between 6 and 8 feet away from the system. A picture of the tin foil balls is shown in Figure 5.1. Testing an object larger than a rain drop helped ensure the tracking, calibration, and display systems were functioning properly. An example progression of a tin foil ball being de-illuminated is given in Figure 5.2. Note that the anti-blob size was kept large to make the anti-blob identification easier in post processing. The images in shown in Figure 5.2 were taken by a separate computer using a camera running at 300 frames per second. The camera was oriented at an angle to capture the tin foil ball and the anti-blob. The frame differences from Figure

Figure 5.1: Tin foil balls used in initial system testing.
Figure 5.2: Progression of tin foil ball de-illumination.

5.2 (a) to Figure 5.2 (b) and Figure 5.2 (b) to Figure 5.2 (c) are both 3 frames. Note the “shadows” or dark regions in Figure 5.2. The dark regions are the anti-blobs de-illuminating the detected tin foil ball. The results illustrate that a decoupled system using IR can successfully track and de-illuminate objects.

5.2 SINGLE DROP RAIN

After testing the basic tracking and display systems of the hardware, the system was tested on a single rain drop. An eye dropper was used (See Figure 5.3) to simulate a single rain drop. The rain drop was tested approximately two and a half feet from the system. Testing a single rain drop ensured the system could identify and track a single drop of rain before testing on multiple drops was conducted. An example progression of a single rain drop being de-illuminated is given in Figure 5.4. Figure

Figure 5.3: Eye-dropper used for single rain drop testing.
5.4 (a) shows the rain drop prior to de-illumination. The rain is visible for a short time since the IR illuminator area is smaller than the visible area that the testing distance. Figure 5.4 (b) shows the rain while de-illuminated, 6 frames after Figure 5.4 (a). The images in Figure 5.4 were taken using the same camera and setup as that in Section 5.1.

5.3 MULTIPLE DROP RAIN

The final phase of testing involved processing multiple rain drops simultaneously. To create multiple rain drops, a watering can was used to simulate rain while testing approximately two and a half feet away from the system. The hardware used to simulate multiple rain drops is shown in Figure 5.5. This testing ensured that multiple rain drops could be tracked and de-illuminated simultaneously. An example image of the system running while de-illuminating multiple drops of rain is shown in Figure
Figure 5.5: Items used to create multiple rain drops.

(a) With system running (b) Without system running

Figure 5.6: Example of rain de-illumination system in operation.

5.6. Figure 5.6 (a) shows the multiple drops of rain while the system is in operation. Note that the area of rain being de-illuminated is smaller than the area that the projector illuminates. This is due to objects in the testing area causing false positives of detection thus those areas in the image were ignored. Additionally, Figure 5.6 (b) shows the multiple drops of rain without the de-illumination system for comparison.
5.4 LIMITATIONS

Due to the simplicity of the hardware and software design, the anti-glare system has several limitations. First, while small variations in vehicle movement or wind direction are tolerable, large variations will cause tracking failure. This is due to the tracking algorithm only looking within a specified window. Secondly, due to the OS constraints, a large latency is present. This forces a large prediction in the anti-blob location. As a future research direction, we plan to develop an FPGA to track the rain with a dedicated hardware. Thirdly, although image processing is performed to reduce false positives, “large” areas of “on” pixels may still produce false positives. This is due to the larger variance in higher intensity illumination sources. Lastly, the IR illumination hardware system is not completely optimized. It is possible that 850 nanometer illumination is not optimal. Also, an IR specific camera would increase rain detection fidelity.

Additionally, it is important to recognize that the TI SLM only emits light at 150 Lumens which is considerably lower than a typical headlight. However, the proof-of-concept established here still holds for a brighter headlight.
The goal of this project is to implement a decoupled rain glare reduction system using IR illumination. The inclusion of IR allows for two separate wavelength ranges to be used concurrently. There are several benefits to a decoupled system. The first is a reduction in interference between the SLM and detection camera. Secondly, the decoupling allows for development of a simpler algorithm. The algorithm is simplified because the detection system has complete awareness of rain drop locations, regardless of the SLM state. Thirdly, there is increased light efficiency for both the detection camera and SLM. With the cold mirror, nearly all the SLM light illuminates the target scene and nearly all the IR illumination is captured by the detection camera. These benefits were confirmed through multiple types of testing. Larger testing objects were used to test the detection and tracking functionality while single drop testing allowed for optimization for rain drops.

From these conclusions, it is clear that a camera-projector decoupling enabled by an IR illumination has a better starting performance than a coupled system.
BIBLIOGRAPHY


