CSC3185:Introduction to Multimedia Systems

Project Report
Group 2

徐博石 Xu Boshi 122040075

罗开诚 Luo Kaicheng 120090770

王心志 Wang Xinzhi 120090191

1 Introduction

1.1 Background

According to statistics, in recent years, China has experienced nearly 200,000 traffic accidents annually, with 60% of these incidents occurring during the nighttime, constituting 50% of the total traffic accident fatalities. Nighttime driving poses significant safety risks, often attributed to factors such as poor illumination and reduced visibility, leading to driver misjudgments and errors in assessing road conditions. Infrared thermal imaging technology is impervious to the limitations of visible light, enabling accurate identification of vital signs regardless of day or night conditions. With the added advantage of glare resistance, infrared thermal imaging technology holds great potential for enhancing nighttime driving safety.

1.2 Motivation

Currently, the cameras, radar, and lidar used in autonomous driving rely on visible light to acquire information. They primarily depend on visible light sensors to receive light reflected by objects or emitted by the objects themselves. The received light signals are then converted into electrical signals, which are further transformed into images through image processors. It is well known that visible light is dependent on illumination. In conditions of low visibility or when encountering obstacles, such as in weather conditions like fog, haze, rain, snow, or dust, visible light imaging systems will be ineffective in detecting target objects from the background, especially in low-light or no-light conditions like night or tunnel scenarios. This necessitates the complementation of infrared technology.

Multi-sensor fusion is essential for autonomous driving. Onboard infrared sensors can operate independently of light sources and can produce clear images even in low light, rain, snow, smoke, haze, and other harsh environments. This effectively enhances the all-weather perception capabilities, compensates for the shortcomings of other sensors, and improves driving safety. Infrared technology finds extensive applications in automobiles, enabling vehicles to achieve real-time 360-degree monitoring of the surrounding environment. It facilitates multi-sensor fusion in smart driving assistance, integrates with the Automatic Emergency Braking function to enhance braking effectiveness, and supports functions such as driver fatigue detection and identity recognition in digital cockpits.

2 Design

2.1 Design Overview

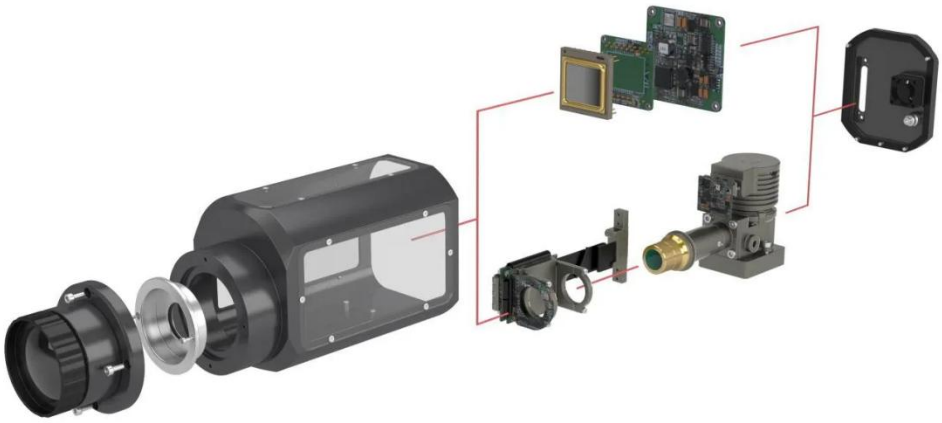
Our automatic braking system is designed to enhance driving safety through the integration of infrared (IR), visible light (RGB) images, and simulated infrared ranging devices. The system takes as input an IR photo, a visible light photo, and a set of distance data acquired by an analog infrared distance measuring device, such as the LDM301. The output consists of text-based commands, including "brake" "safe distance" and distance measurement including break distance and real distance, advising the driver to maintain driving distance.

The implementation method involves capturing RGB and IR images in the direction of the car's travel. These images undergo analysis and processing, like feature extraction, to identify objects in the driving direction. The system utilizes an ultrasonic ranging device in conjunction with a specialized algorithm to calculate the distance (D) to the nearest point of an object. Decision-making is then based on this distance, with appropriate commands generated to instruct the vehicle's behavior.

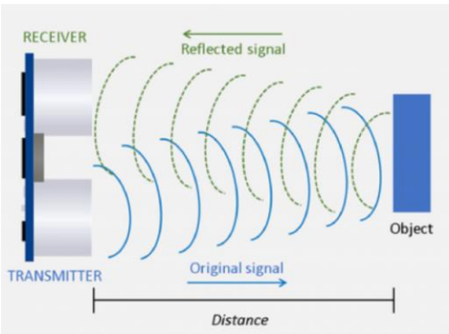
2.2 Collection

There are two sensors included in the system: a IR + RGB camera and a ultrasonic sensor.

First, the IR camera will collect the data follow the driving direction, and take a IR photo and a RGB photo. It will be send and process in the Image process program immediately. We will get a group of information marking the position of key points pointing out the objects on the driving direction.



Second, according to the position of those key points, the ultrasonic sensor will point out to them and get the distance information as required.



Overall, this process will provide a group of key points with their position in the image and their actural distance between the sensor and the objects.

2.3 Processing

2.3.1 Image Processing

For the image processing part, we design a process which helps find object centers according to the input IR picture and visual light picture.

The outline:

Image Input, Image Denoise, VL-IR Image Fusion, Useful Zone Define, Edges Finding, Contours Finding, Centers Finding, Layer Add

Image Input:

In this part, we use open-cv library in python to do image operations. The reading image inputs as Grayscale image. Because Grayscale image can eliminate the difference between IR Image Color space and visual light Image color space. In our design, original_image is IR image, original_image2 is VL image.

```
original_image = cv2.imread(image_path1, cv2.IMREAD_GRAYSCALE)
original_image2 = cv2.imread(image_path2, cv2.IMREAD_GRAYSCALE)
```

Image Denoise:

In this part, we use Gaussian Algorithm to denoise the input image, based on the open-cv function GaussianBlur. The second parameter (5,5) stands for the gaussian convolution kernel. Large size can denoise better, but it also leads to the distortion of the image. After manual test, we decide use (5,5) size kernel in our project. The third parameter is the standard variance of the convolution. Zero represents the standard variance is calculated automatically according to the gaussian kernel size.

```
# use GaussianBlurry to denoise
blurred_image1 = cv2.GaussianBlur(original_image, (5, 5), 0)
blurred_image2 = cv2.GaussianBlur(original_image2, (5, 5), 0)
```

VL-IR Fusion:

In this part, we fuse IR image and visual light image based on their denoised version. The variable vl-part represent the visual light image propotion of the fused image. Because in our design, the input visual light images are hard to recognize due to overexposures, we set the VL propotion squared.

```
# IR VL image fusion
vl_part = 0 #visual light image part
blurred_image = cv2.addWeighted(blurred_image1, 1-vl_part**2,
blurred_image2,vl_part**2, 0)
```

Useful Zone Define:

In this part, we define the useful area to catch useful information in the final picture, we only concentrate the points in the area and pay less attention on things outward the area.

```
# Define the trapezoid coordinates
height, width = original_image.shape
trapezoid_top = int(height * 0.3)
trapezoid_bottom = height
trapezoid_left = int(width * 0.3)
trapezoid_right = int(width * 0.7)
```

Edges Finding:

In this part, we use Canny Algorithm to detect the edge, and allow two parameter setting.

The lower_threshold means that the edge detection value lower than lower_threshold will be abandoned. The upper_threshold means that the edge detection value higher than upper_threshold will be viewed edges. And other value between them will be considered whether connection to the edges.

```
# use Canny Algorithm to detect the edge
lower_threshold = 50
upper_threshold = 150
edges = cv2.Canny(blurred_image, lower_threshold, upper_threshold)
```

Contours Finding:

In this part, we use open-cv function findContours. This function will return the contour we use to find the centers. Parameter RETR_EXTERNAL means the function only detects the outward edges. Parameter CHAIN_APPROX_None is that storing all points in contours.

```
# find contours
contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
```

Center Finding:

In this part, we set an area_threshold to eliminate the small contours caused by noise and judge those centers whether it lies in the trapzoid. Then, we calculate the center of contours, and store them in array “centers”

```
area_threshold = 10
centers = []
for contour in contours:
    area = cv2.contourArea(contour)
    if area > area_threshold:
        M = cv2.moments(contour)
        if M["m00"] != 0:
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])

            # Check if the center is within the trapezoid
            if trapezoid_top <= cY <= trapezoid_bottom and
trapezoid_left <= cX <= trapezoid_right:
                centers.append((cX, cY))
```

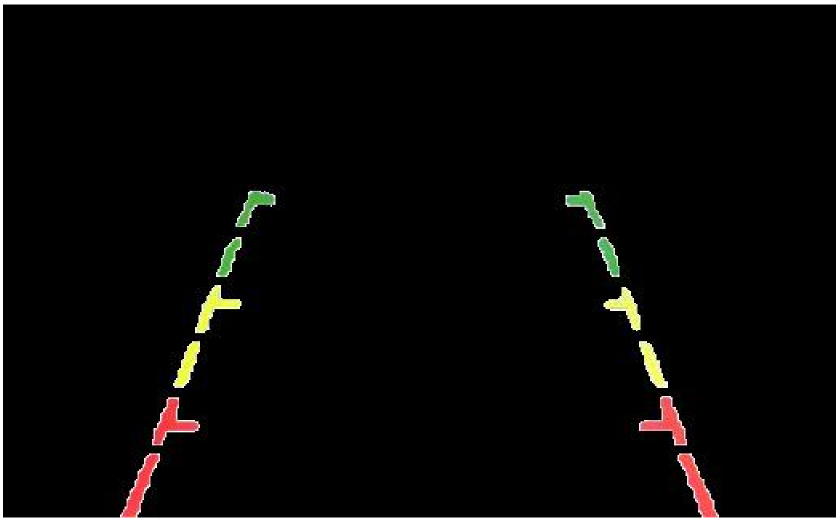
Layer Add:

In this part, we add those centers into the original image and combine them with a detection layer, to help user better notice the object centers in vision area.

```
# mark center in IR-VL fusion picture and Visible light picture
marked_image1 = blurred_image.copy()
for center in centers:
    cv2.circle(marked_image1, center, 5, (255, 0, 0), -1)

marked_image2 = original_image2.copy()
for center in centers:
    cv2.circle(marked_image2, center, 5, (255, 0, 0), -1)

marked_image1 = cv2.addWeighted(marked_image1, 1, new_image, 1, 0)
marked_image2 = cv2.addWeighted(marked_image2, 1, new_image, 1, 0)
```



2.3.2 Data Processing

For the data processing, depends on the distance and the speed of cars, the system then will judge whether if the speed of the car is safe or not. To be simplified, we used a function with fixed coefficient to measure the breaking distance:

```
S=v^2/(2a)
```

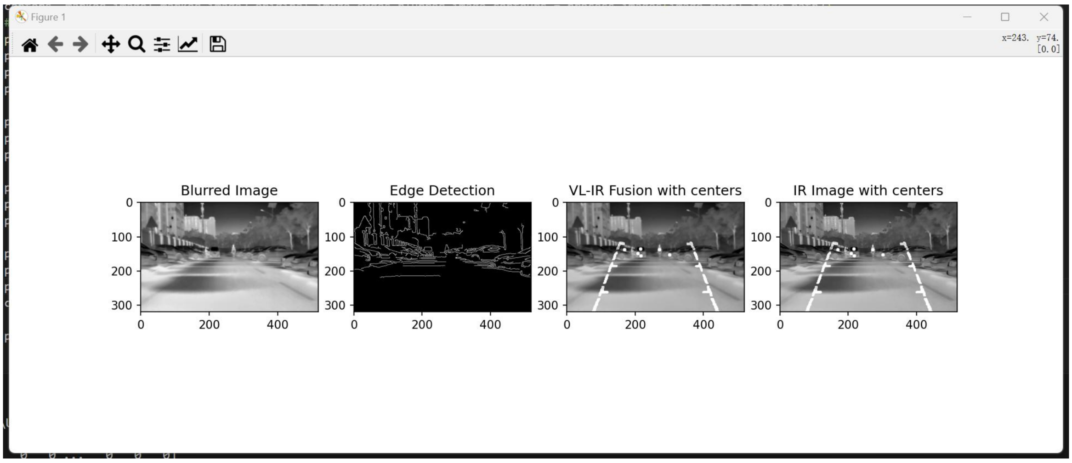
The program will output the required data and transfer to do further processing.

2.4 Visualization and Output

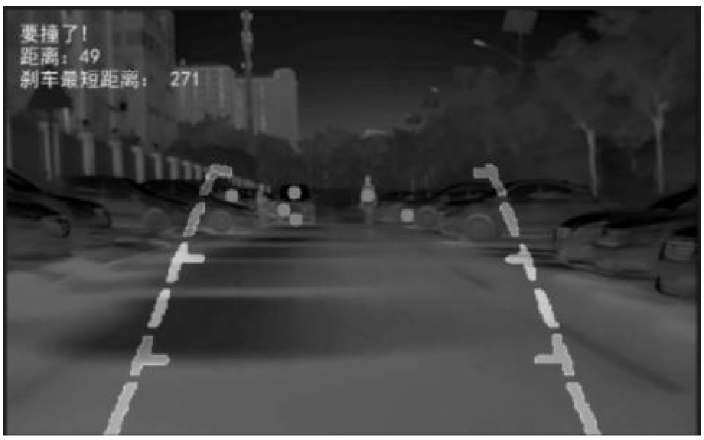
To execute the entire project, using the command"

```
python handler.py
```

It will first perform the data processing procedure by calling the previously mentioned simulation program to obtain the necessary data. Then, the image processing procedure will be executed. After the image processing is completed, a chart will be displayed, including the original two images and the processed image with annotated objects.



The processed image will also be saved. Finally, the evaluation will be performed, and the results will be displayed on the image along with the image itself.



3 Application and Benefit

These are the key advantages of integrating intelligent systems with infrared subsystems. Point 1: Enhanced Recognition of Living Beings: Infrared imaging excels in recognizing living organisms by capturing heat radiation, providing a clear picture of anything with a temperature above absolute zero. Point 2: Strong Night Imaging Capability: The system's robust night imaging is a game-changer, working effectively in both daylight and darkness, overcoming limitations of visible light imaging at night. Point 3: Robust Interference Resistance in Adverse Weather: With better penetration due to its longer wavelength, infrared technology performs reliably in adverse weather conditions such as rain or fog. Point 4: Glare-Free Operation: The subsystem's glare-free operation is achieved by selectively capturing mid-to-long wave information, ensuring clear images in scenarios with varying light conditions.

In summary, it is hoped that this project can further develop and even collaborate with other collision avoidance systems to ensure the personal safety of drivers and pedestrians.

4 Conclusion

In conclusion, our project addresses the critical safety issues associated with nighttime driving and advances autonomous driving technology. Recognizing the alarming rate of nighttime traffic accidents in China, we emphasize the potential of infrared technology to enhance safety, especially in adverse conditions. The design integrates infrared and visible light images with simulated infrared ranging devices, aiming to improve decision-making in autonomous systems.

The collection process involves an IR + RGB camera and an ultrasonic sensor, providing essential data for analysis. The image and data processing stages showcase effective methods to enhance the system's capabilities. Visualization and output provide a comprehensive assessment, and the project's application highlights the benefits of infrared technology in recognizing living beings, night imaging, and adverse weather conditions.

Overall, the project contributes to collision avoidance and nighttime driving safety, offering real-time commands for enhanced vehicle safety. The hope is for continued development and potential collaboration to ensure the safety of both drivers and pedestrians on the road.

5 Reference and Appendices

The following references are for the implementation of the coding and testing part:

[1]Xuanyuan Intelligent Driving Technology Co., Ltd.. "Vehicle Thermal Imaging Obstacle Avoidance System." xy-idrive.com, Dec 17, 2023, https://www.xy-idrive.com/portal/article/index.html?id=23&cid=13.

[2]敖宏伟, 陈学文,and 荣同康."安全距离-时间模型的汽车紧急制动分层控制策略研究." 重庆理工大学学报(自然科学) 36.01(2022):31-38.

Here you can check the source code and all related materials from the Github repository with the link:

https://github.com/xh2002/CSC3185_final

---------------------- END OF THE REPORT ---------------------