Dash-Cam Single Image Processing

Amin Ghandi, Bahar Oveis Gharan[a]

[a]Sharif University of Technology

1. Introduction

In recent years, car dashcams have become increasingly popular and are now commonly used by drivers worldwide. These cameras are compact devices that are installed on the dashboard of a car and constantly record the view of the road ahead. This has led to an abundance of video footage capturing various incidents, accidents, and events on the road.

However, manually analyzing hours of dashcam footage can be a time-consuming and arduous task. Therefore, researchers and engineers have turned to image processing techniques to automate and streamline the analysis of car dashcam videos. Image processing involves using algorithms and computer vision techniques to extract useful information from images or video frames.

The analysis of car dashcam videos using image processing offers numerous benefits. It enables the identification and classification of objects, such as vehicles, pedestrians, cyclists, and road signs, in the footage. This information can be used for various purposes, including accident investigation, traffic analysis, and improving autonomous driving systems.

By applying image processing algorithms, it becomes possible to detect and track objects in real-time, measure distances and speeds, and even predict potential hazards or collisions. Additionally, image processing techniques can enhance the visibility and clarity of the video, allowing for better analysis and understanding of the recorded content.

Overall, the analysis of car dashcam videos using image processing brings immense potential for improving road safety, traffic management, and driver assistance systems. By automating and enhancing the analysis of dashcam footage, this technology can revolutionize the way we understand and interpret events on the road, leading to safer and more efficient transportation systems.

2. Defoging

2.1. Introduction

This part focuses on the persistent issue of poor visibility in images captured in foggy weather conditions. Fog causes distant objects to lose contrast and blend into their surroundings, resulting in blurry and indistinct photographs. This phenomenon occurs due to the attenuation of reflected light and the scattering of atmospheric light by aerosols. Early approaches to tackle this problem relied on depth information or multiple observations of the same scene. Although recent advancements have been made in single image dehazing, it remains a complex challenge with limited knowledge about the underlying scene structure.The primary goal of this part is to generate high-quality dehazed images with accurate colors and precise edge details. This approach incorporates a boundary constraint on the scene transmission, contextual regularization using a filter bank, and an efficient optimization scheme.

2.2. Image Model

The linear interpolation model to explain the formation of a haze image is :

$$I(x) = t(x)J(x) + (1 \quad t(x))A \tag{1}$$

$$t(x) = e^{-\beta d(x)} \tag{2}$$

where I(x) is the observed image, J(x) is the scene radiance, A is the global atmospheric light, and t(x) is the scene transmission, correlated with the scene depth and assumed that the haze is homogenous.

The goal of image dehazing is to recover the scene radiance J(x) from I(x) based on so required to estimate the transmission function t(x) and the global atmospheric light A.

2.3. Boundary Constraint from Radiance Cube

Geometrically, according to Eq.(1), a pixel I(x) contaminated by fog will be “pushed” towards the global atmospheric light A. As a result, we can reverse this process and recover the clean pixel J(x) by a linear extrapolation from A to I(x). The appropriate amount of extrapolation is given by:

$$\frac{1}{t(x)} = \frac{\|J(x) - A\|}{\|I(x) - A\|} \tag{3}$$

$$C_0 \leq J(x) \leq C_1, \; x \in \Omega \tag{4}$$

Consider that the scene radiance of a given image is always bounded, we have C0 and C1, two constant vectors that are relevant to the given image. The above requirement on J(x), in turn, imposes a boundary constraint on t(x). Suppose that the global atmospheric light A is given. Thus, for each x, we can compute the corresponding boundary constraint point $J_b(x)$. Then, a lower bound of t(x) can be determined by using Eq.(3)

where$t_b(x)$is the lower bound of t(x), given by:

$$t_b(x) \quad = \quad min\left\{ \underset{c\in\{r,g,b\}}{max} \left(\frac{A^c \quad I^c(x)}{A^c \quad C_0^c}, \frac{A^c \quad I^c(x)}{A^c \quad C_1^c} \right),\ 1 \right\} \tag{5}$$

where $I_c, A_c, C_c, 0andC_c$ 1 are the color channels of I, A, C_0andC_1, respectively.
The boundary constraint of t(x) gives us a new way to look at the dark channel prior. Let's say the global atmospheric light is brighter than any pixel in the hazy image. We can calculate t_b J(x) using Equation (1) by assuming the dark channel of each pixel is zero. Similarly, if we assume the transmission in a local image patch is constant, we can quickly find the patch-wise transmission t̃(x) in He et al.'s method by applying a maximum filtering on t(x). The global atmospheric light is usually slightly darker than the brightest pixels in the image. However, these brighter pixels can come from light sources like the sky or car headlights. In cases like these, the dark channel prior fails to capture these pixels, but the proposed boundary constraint remains valid. Additionally, the commonly used constant assumption on transmission within a local image patch is demanding, leading to underestimated patch-wise transmissions. To address this, we introduce a more accurate patch-wise transmission that allows for slight variations within a local patch. The new patch-wise transmission is provided below.

$$\hat{t}(x) = \underset{y\in\Omega_x}{min} \quad \underset{z\in\Omega_y}{max}\ t_b(z). \tag{6}$$

2.4. Weighted L1-norm based Contextual Regularization

The assumption that pixels in a local image patch have similar depth values is often true, but can fail when there are abrupt depth jumps. This leads to halo artifacts in dehazing results. To address this issue, a weighting function is introduced that cancels the contextual constraint between neighboring pixels with a large depth difference. However, since depth information is not available in single image dehazing, the weighting function is instead constructed based on the color difference of local pixels. Two examples of this construction are provided, with one based on the squared difference between color vectors of neighboring pixels.

$$W(x,\ y) = (|l(x) \ - \ l(y)|^{\alpha} \ + \ \epsilon)^{-1}$$

where l is the log-luminance channel of the image I(x), the exponent > 0controls the sensitivity to the luminance difference of two pixels and $\epsilon$is a small constant (typically 0.0001) for preventing division by zero Integrating the weighted contextual constraints in the whole image domain leads to the following contextual regularization on t(x):

$$\sum_{j\in\omega} \|Wj \ \circ \ (Dj \ \otimes \ t\|_1 \tag{7}$$

where is an index set, $\circ$ represents the element-wise multiplication operator, $\otimes$ stands for the convolution operator, Dj is a first-order differential operator, $W_j(j \in \omega)$ is a weighting matrix.

2.5. SCENE TRANSMISSION ESTIMATION

This paragraph discusses the process of dehazing an image using a specific method.

To dehaze an image, it is necessary to estimate the transmission function (t(x)) and the atmospheric light (A). The paragraph mentions a method proposed by He et al., which uses the dark channel of an image to estimate the atmospheric light. They select the top 0.1 percent brightest pixels from the dark channel and choose the one with the highest intensity as the estimate for A. However, the paragraph states that the method proposed in this study is a modified version of He et al.'s method and is more efficient.

The modified method begins by filtering each color channel of the input image using a minimum filter with a moving window. The maximum value of each color channel is then taken as the estimate for the component of A.

To find the optimal transmission function, we have an objective function that needs to be minimized.

$$\frac{\lambda}{2}\|t - \hat{t}\|_2^2 + \sum_{j\in\omega} \|Wj \ \circ \ (Dj \ \otimes \ t\|_1 \tag{8}$$

The objective function consists of a data term and a contextual constraint term, with a regularization parameter (λ) used to balance the two terms.

To optimize the objective function, an efficient method based on variable splitting is employed. This method involves introducing auxiliary variables (u_j) and constructing a sequence of sub-problems. The solutions of these sub-problems converge to the optimal solution of the original problem. By fixing the weight (β) in the cost function and performing alternating optimization with respect to u_j and t, the sub-problems can be solved quite efficiently.

Figure 1: Dehazing a frame

3. Road Analysis

This section employ Bird eye view to analyze the lane and water mark in segmentation, enhancing the understanding.

Bird's eye view in image processing refers to a technique wherein images are analyzed and processed from a top-down perspective, It involves transforming images captured from a different viewpoint into a top-view representation.

The main goal of using a bird's eye view in image processing is to provide a comprehensive and holistic understanding of a scene or an environment. By analyzing images from this particular perspective, it becomes easier to identify and extract relevant information such as object positions, dimensions, trajectories, and patterns. One of the key applications of bird's eye view in image processing is in computer vision for autonomous vehicles. In the context of self-driving cars, overhead cameras or lidar sensors are used to capture images from an elevated position. These images are transformed into a top-view representation to help the car understand its surroundings more accurately.

By employing advanced image processing techniques such as perspective correction, geometric transformations, and stitching, a bird's eye view can be generated. This top-view representation allows the autonomous vehicle to perceive the environment in a manner similar to a human driver, enabling better decision-making for navigation, object detection, lane recognition, and obstacle avoidance.

3.1. Mark detection

For segmenting the mark on the road, there are different methods like: Canny edge detection, color model detection, and histogram analysis.

3.1.1. Canny edge Detection

Detecting marks on roads using the Canny method for segmentation has become a popular approach in computer vision applications for real-time analysis and interpretation of traffic scenes. The Canny edge detection algorithm is widely used due to its ability to accurately identify edges in an image while minimizing noise interference.

The Canny method for segmentation involves several steps to detect road marks effectively. First, the input image is converted to grayscale to simplify the process. This step eliminates the need for color analysis, which can sometimes introduce additional complexities.

After converting the image to grayscale, the Canny edge detection algorithm is applied. This algorithm involves several key steps that work together to determine the presence of edges in the image. The steps include:

- Gaussian smoothing: The image is convolved with a Gaussian filter to reduce noise and make it more resistant to false edge detection caused by pixel variations.

- Gradient calculation: The algorithm then calculates the gradient magnitude and direction in the image, which provides information about how rapidly intensity changes occur.

- Non-maximum suppression: This step helps to thin out the detected edges by ensuring that only local maxima in the gradient directions are kept, while the rest are suppressed. This enhances the accuracy of edge detection.

- Hysteresis thresholding: Finally, two threshold values are used to differentiate strong edges from weak edges. Pixels with gradient magnitudes above the higher threshold are considered strong edges, while those between the two thresholds are considered weak edges. Weak edges are only considered strong if they are connected to strong edges, creating a continuous line of edges.

Once the Canny edge detection is applied, the detected edges are further processed to isolate road marks specifically. This is achieved by filtering out edges that do not meet certain criteria based on their shape, size, or orientation. For example, lines that are not sufficiently straight or parallel to each other can be discarded.

The resulting segmented image will showcase the road marks prominently, making it easier to analyze and interpret the information. Further post-processing techniques, such as morphological operations or contour detection, can be employed to refine the segmentation output and extract the road marks more precisely.

3.1.2. Color Model Detection

Color model segmentation, such as HSV or LAB, is a technique commonly used in image processing and computer vision tasks, including watermark detection in road images. Watermark detection refers to the process of identifying and extracting embedded watermarks or logos from an image.

The HSV (Hue, Saturation, Value) and LAB (Lightness, A, B) color models are popular choices for segmentation due to their ability to represent colors effectively and capture the perceptual differences between them.

In the case of road images, watermarks or logos are often added to enhance branding or ownership of the image. These watermarks can vary in size, color, and intensity, making them challenging to detect using traditional methods. However, employing color model segmentation techniques can significantly improve the success rate of watermark detection in road images.

One commonly used technique is to convert the road image from the standard RGB color model to either HSV or LAB model. In both models, each pixel is represented by a combination of different channels that represent specific characteristics of the color.

For HSV, the hue channel represents the color itself, the saturation channel represents the intensity or purity of the color, and the value channel represents the brightness or lightness of the color. By isolating the hue channel, one can extract color information while ignoring variations in brightness and saturation. This allows for the detection of specific colored watermarks in road images.

On the other hand, the LAB color model separates the image into three channels: lightness (L), which represents the perceived lightness of the color, and two color-opponent channels A and B, which capture the color information. The A channel represents colors ranging from green to red, while the B channel represents colors ranging from blue to yellow. By analyzing the A and B channels, one can identify specific color regions in the image, which can help in locating watermarks.

Once the image is converted to the desired color model, various image processing techniques, such as thresholding, clustering, or morphological operations, can be applied to segment the region of interest (ROI) containing the watermark. The exact method used may vary depending on the complexity and size of the watermark, as well as the characteristics of the road image itself.

Color model segmentation techniques, such as HSV or LAB, have proven to be effective in detecting watermarks in road images due to their ability to isolate specific color information. However, it is important to note that these techniques may produce false positives or false negatives depending on the quality of the image and the complexity of the watermark. Hence, it is often necessary to combine color model segmentation with other techniques, such as edge detection or texture analysis, to improve the accuracy of watermark detection in road images.

3.1.3. Histogram Analysis

Segmenting road marks using histogram analysis involves the following steps:

- Preprocessing: Convert the input image to grayscale to simplify the analysis. Remove any noise or unwanted artifacts by applying image enhancement techniques such as blurring or sharpening.

- Histogram calculation: Calculate the histogram of the preprocessed image. A histogram represents the distribution of pixel intensities in an image. Each pixel value is counted and grouped into intensity bins.

- Threshold determination: Analyze the histogram to determine the appropriate threshold value for segmenting the road marks. The threshold value will separate the road marks from the background. Common methods for threshold determination include Otsu's method, adaptive thresholding, or analyzing peaks and valleys in the histogram.

- Binary segmentation: Using the determined threshold value, convert the grayscale image into a binary image. In this binary image, the road marks will be represented by white pixels (foreground) and the background will be represented by black pixels.

- Morphological operations: Perform morphological operations (e.g., erosion, dilation, closing) to refine the segmented road marks. These operations can help remove any noise or fill any gaps present in the binary image.

- Postprocessing: Further refine the segmented road marks by removing any small objects that are not likely to be road marks. This can be done by applying size-based filters or analyzing shape properties of the segmented regions.

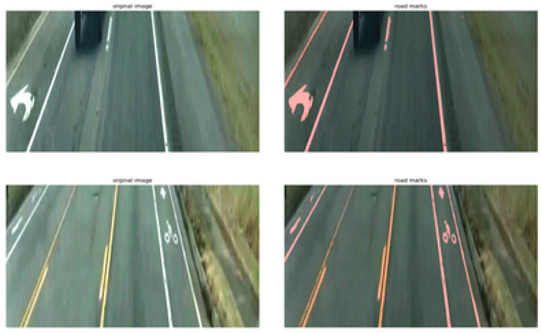

Figure 2: Mark Detection

Conclusion: With use of these three different methods that mentioned above, color model for segmentation choosed. Specifically, mark segmented using the light component in the LAB color model and the value component in the HSV color model, resulting in the best outcome.

3.2. Lane Detection

Lane detection is a critical task in the field of digital image processing and computer vision. It involves accurately identifying and tracking the lanes on a road or highway from images or video feeds captured by cameras. Lane detection plays a crucial role in advanced driver assistance systems (ADAS) and autonomous vehicles, providing essential information for lane departure warning systems, lane-keeping assistance, and autonomous lane following.

The process of lane detection typically starts with pre-processing, where the input image is first transformed to enhance the lane markings and remove noise. Various techniques can be applied during this stage, such as color space conversion, contrast enhancement, and noise reduction filters.

After pre-processing, the image is usually segmented to identify potential lane markings. Common segmentation techniques include thresholding, edge detection, and

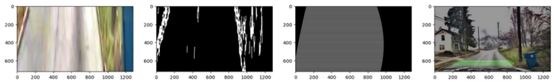

Figure 3: lane Detection

region-based methods. Thresholding techniques use a specific threshold value to distinguish between lane and non-lane pixels based on their intensity or color values. Edge detection algorithms, such as Canny edge detection, identify sharp changes in pixel intensity, typically indicating lane markings. Region-based methods involve grouping pixels together based on their spatial proximity and color properties.

Once the lanes are segmented, post-processing techniques are applied to refine the detection results. These techniques often include line fitting and curve fitting algorithms. Line fitting algorithms, like the Hough transform, aim to extract straight lines from the segmented lane markings. Curve fitting algorithms, such as polynomial fitting or spline interpolation, are employed when dealing with curved lanes.

In addition to lane detection, lane tracking is another essential aspect of the process. This involves updating and estimating the position of the lanes in consecutive images or frames of a video. By tracking the lanes, the system can handle challenges like occlusions, abrupt lane changes, or irregular lighting conditions. Various tracking algorithms, including Kalman filters, particle filters, and Bayesian methods, can be employed for this purpose.

Furthermore, lane detection can be enhanced with the use of deep learning and machine learning techniques. Convolutional neural networks (CNNs), for instance, can be trained to detect and classify different road elements, including lanes. By utilizing large amounts of annotated data, these models are capable of achieving high accuracy in identifying lane markings under varying conditions.

Despite significant advancements in lane detection techniques, challenges still exist. These include cases of fading or missing lane markings, poor road conditions, varying lighting conditions, and complex road geometries. Researchers and engineers are continually working to improve lane detection algorithms to address these challenges and enable safer and more reliable lane-keeping functions in automated vehicles.

4. Resources

- Meng, G., Wang, Y., Duan, J., Xiang, S., and Pan, C. (2013). Efficient image dehazing with boundary constraint and contextual regularization. Proceedings of the IEEE international conference on computer vision, 617-624.

- Gautam, S., Gandhi, T. K., and Panigrahi, B. K. (2020). An Improved Air-Light Estimation Scheme for Single Haze Images Using Color Constancy Prior. IEEE Signal Processing Letters, 27, 1695-1699. doi: 10.1109/LSP.2020.3025462.

- Raikwar, S. C., and Tapaswi, S. (2020). Lower Bound on Transmission Using Non-Linear Bounding Function in Single Image Dehazing. IEEE Transactions on Image Processing, 29, 4832-4847. doi: 10.1109/TIP.2020.2975909.Towards Data Science. (n.d.). Image segmentation with classical computer vision-based approaches. Retrieved from https://towardsdatascience.com/image-segmentation-with-classical-computer-vision-based-approaches-80c75d6d995f

- Korting, T. (n.d.). Region-growing-segmentation. Retrieved from https://github.com/tkorting/youtube/blob/master/region-growing-segmentation/main.m

- Towards Data Science. (n.d.). Image segmentation with clustering. Retrieved from https://towardsdatascience.com/image-segmentation-with-clustering-b4bbc98f2ee6

- Ayca, Y. (n.d.). Image-Segmentation. Retrieved from https://github.com/YCAyca/Image-Segmentation/tree/main/Clustering

- Cao, J., Song, C., Song, S., Xiao, F.,and Peng, S. (2019). Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments. Sensors, 19(14), 3166. doi: 10.3390/s19143166

- YouTube. (n.d.). Lane Detection for Autonomous Driving: Conventional and CNN approaches. Retrieved from https://www.youtube.com/watch?v=xIRT3rgWrFQ