

Image Dehazing for Advanced Driver Assistance Systems

Benjamin Tilbury

1908820

A thesis submitted for the degree of Master of Science in Artificial Intelligence

Supervisor: Dr Vahid Abolghasemi

School of Computer Science and Electronic Engineering

University of Essex

August 2021

Abstract

Drivers are continually faced with adverse weather conditions that limit their visibility of the road and its surroundings. Prime examples of this are haze and fog, both caused by particles suspended in the atmosphere, scattering light as it travels to an observer. Existing techniques for dehazing images and video could be use in the form of an advanced driver assistance system (ADAS) to provide a driver with a clear view of the road at all times, through a mounted display. This study examines eight dehazing techniques and their suitability for use in such an ADAS. Evaluation is performed on a novel haze video dataset – designed for the scenarios an ADAS will encounter – using a framework of criteria based on the qualities required of an ADAS. This evaluation provides insight into the strengths and weaknesses of current dehazing methods applied to the context of an ADAS. Additionally, this study proposes a new dehazing method based on the outcomes of the prior evaluation. The proposed dehazing system employs novel smooth-extremum filters that removes the need for a refinement step as part of the dark channel prior (DCP) dehazing process. This method is significantly faster than existing approaches and achieves excellent results on the ADAS evaluation framework; including commonly used metrics such as multi-scale structural similarity (\uparrow MS-SSIM) and peak signal-to-noise-ratio (\uparrow PSNR)

Contents

Abstract	I
List of Algorithms	III
List of Equations	III
List of Figures	IV
List of Tables	V
Abbreviations	VI
1 Introduction	1
1.1 Haze & Dehazing	2
1.2 Research Focus	4
1.3 Contributions	5
1.4 Dissertation Structure	6
2 Literature Review	7
2.1 Single-Image Dehazing	7
2.2 Video Dehazing	13
2.3 ADAS Dehazing	14
2.4 Dehazing Datasets	15
3 Novel Video Haze Dataset	20
3.1 Methodology	20
3.2 Discussion	25
4 Evaluation of Existing Methods	28
4.1 Dehazing Method Selection	28
4.2 Implementation	29
4.3 Evaluation Criteria & Results	31
4.4 Discussion	41
4.5 Conclusions	45
5 Proposed Smooth-Extremum Dehazing System	46
5.1 Smooth-Extremum Filters	46
5.2 Smooth-Extremum-Based Dehazing	50
5.3 Parameter Optimisation	53
5.4 Results & Discussion	57
6 Conclusion	61
6.1 Future Work	61
References	62
Appendix: Code Overview	68
Appendix: Parameter Tables	69
Appendix: Normalisation Process	71

List of Algorithms

Algorithm 1: Smooth-Extremum Filter (SEF).....	48
Algorithm 2: Approximate Smooth-Extremum Local Filter (ASELF)	49

List of Equations

Equation 1: Hazy Image Formation	3
Equation 2: Transmission Map.....	3
Equation 3: Clear Image Recovery.....	4
Equation 4: The Dark Channel Prior	8
Equation 5: Dark Channel Prior Coarse Transmission.....	8
Equation 6: Dark Channel Prior Recovery	9
Equation 7: AOD-Net Learn	12
Equation 8: AOD-Net Recovery	12
Equation 9: Atmospheric Light Update Strategy	13
Equation 10: No Black Pixel Constraint Recovery	14
Equation 11: No Black Pixel Constraint Atmospheric Veil	14
Equation 12: No Black Pixel Constraint Luminance	15
Equation 13: Sky Mask Adjusted Transmission	24
Equation 14: Haze Generation.....	25
Equation 15: Best Possible Clear Image	32
Equation 16: Estimated Best Possible Clear Image	32
Equation 17: Smooth-Maximum (Smooth-Extremum).....	46
Equation 18: SoftArgMax	47
Equation 19: Smooth-Extremum from SoftArgMax.....	47
Equation 20: Whole Image Smooth-Extremum	47
Equation 21: Patch-Wise Smooth-Extremum (Smooth-Extremum Filter)	47
Equation 22: Smooth-Extremum Local Filter	48
Equation 23: Approximate Smooth-Extremum Local Filter	49

List of Figures

Figure 1:	Dehazing ADAS Process	2
Figure 2:	Hazy Image Formation	3
Figure 3:	Transmission Ambiguity Example	4
Figure 4:	Atmospheric Light Ambiguity Example.....	4
Figure 5:	Dark Channel Prior Full Process.....	8
Figure 6:	Dark Channel Prior Coarse/Refined Transmission Example.....	9
Figure 7:	Dark Channel Prior Transmission Estimation Process	9
Figure 8:	Depth Map Comparison	22
Figure 9:	ManyDepth Contextless Depth Prediction.....	23
Figure 10:	Novel Haze Dataset Examples.....	27
Figure 11:	Blob Distortion Example	38
Figure 12:	Normalised Criteria Scores for Existing Methods	41
Figure 13:	Qualitative Example Results for Existing Methods.....	44
Figure 14:	Smooth-Extremum Filter Example	48
Figure 15:	Proposed Dehazing System Process	50
Figure 16:	Local Method for Smooth-Extremum Dark Channel	51
Figure 17:	Global Method for Smooth-Extremum Dark Channel.....	52
Figure 18:	Opening Method for Smooth-Extremum Dark Channel.....	53
Figure 19:	Normalised Criteria Scores for All Methods	57
Figure 20:	Plot of \uparrow MPPS Against \uparrow MS-SSIM for All Methods.....	58
Figure 21:	Qualitative Example Results for All Methods.....	60

List of Tables

Table 1:	Overview of Existing Outdoor Haze Datasets	16
Table 2:	Selected KITTI Sequences for Novel Haze Dataset	20
Table 3:	Haze Categories	25
Table 4:	Novel Haze Dataset Overview	26
Table 5:	Real-time Capability Quantitative Results.....	33
Table 6:	Visibility Improvement Quantitative Results	34
Table 7:	Naturalness Quality Quantitative Results	35
Table 8:	Video Consistency Quantitative Results.....	36
Table 9:	Model Conformity Quantitative Results.....	37
Table 10:	Quality Metrics Failure Case	39
Table 11:	Proposed Dehazing System Parameters.....	54
Table 12:	Optimisation Dataset Overview	55
Table 13:	↑MS-SSIM of Proposed Dehazing System.....	56
Table 14:	Proposed Dehazing System Dense Haze Results.....	59
Table 15:	Proposed Dehazing System Moderate Haze Results	59
Table 16:	Proposed Dehazing System Light Haze Results.....	59

Abbreviations

ADAS	–	Advanced Driver Assistance System(s)
CNN	–	Convolutional Neural Network
CPU	–	Central Processing Unit
DCP	–	Dark Channel Prior
GPU	–	Graphics Processing Unit
HSV	–	Hue, Saturation, Value
MRF	–	Markov Random Field
MSE	–	Mean Square Error
PSNR	–	Peak Signal-to-Noise-Ratio
RGB	–	Red, Green, Blue
SSIM	–	Structural Similarity Index Measure
MS-SSIM	–	Multi-Scale Structural Similarity Index Measure

1 Introduction

Reduced visibility on the road poses a hazard for drivers and pedestrians. Both human drivers and self-driving systems require a clear view of the road, obstacles, and signage at all time to make split-second decisions that could have dangerous ramifications.

Visibility can be hindered in a variety of different ways, but the most pervasive is the phenomena of light being scattered by tiny particles suspended in the medium it is traveling through – causing haze and fog. Dehazing (also referred to as defogging) is the process of removing from an image or video, the effects caused by these suspended particles. Thus improving visibility within the image. While dehazing has applied in many contexts ranging from aesthetic photography to unmanned aerial vehicles [1] and underwater exploration [2], it is still a difficult task that poses a major challenge to computer vision [3, 4].

Advanced driver assistance systems (ADAS) are vehicle systems that they aim to augment the capabilities of human driver rather than replace them; as in the case of autonomous systems. Common examples of ADAS are rear-view cameras to aid in reversing [5], adaptive cruise control to maintain a cruising speed [6], and pre-crash systems to warn of sudden hazards [7]. It's easy to imagine a system similar to a rear-view camera, that could provide a dash-cam perspective displaying an ideal clear view of the road at all times (Figure 1). Though such systems have been discussed academically [8-13], there currently exists no commercial dehazing ADAS. While it's impossible to know the precise reasons for this, a likely contributing factor is the lack of cohesive analysis on current dehazing approaches and algorithms for use in ADAS.

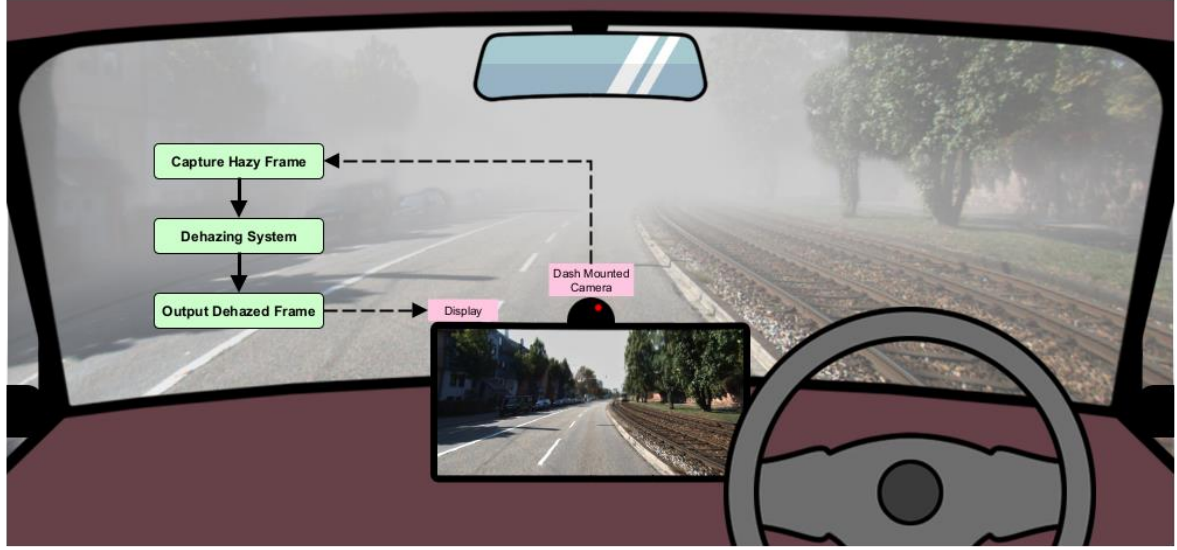


Figure 1: The process of a dehazing ADAS

This study will provide a comprehensive evaluation of existing dehazing methods from the perspective of an ADAS. The results of this evaluation will then be used to guide the development of a new dehazing system specialised for use in an ADAS.

1.1 Haze & Dehazing

Driver visibility can be impaired by a variety of weather conditions, paramount of which is small particles suspended in the atmosphere. These particles attenuate light as it travels to an observer through scattering and absorption. This phenomenon can be known as **haze**, **fog**, **mist**, **smoke**, and **smog** – among others. Nomenclature varies drastically for the distinction of these terms based on size, concentration, composition, and distribution of particles or the degree to which visibility is reduced – but their underlying cause remains the same. For the sake of clarity and consistency, in this paper ‘**haze**’ refers to any such particles in the atmosphere as well as the effect of reduced visibility they cause.

As light travels from a scene point to an observer, the suspended particles cause it to scatter. This has the dual effect of attenuating true scene reflectance while compounding the overall atmospheric light of the scene. These effects occur in tandem as a convex combination that follows a monotonic relationship with scene depth.

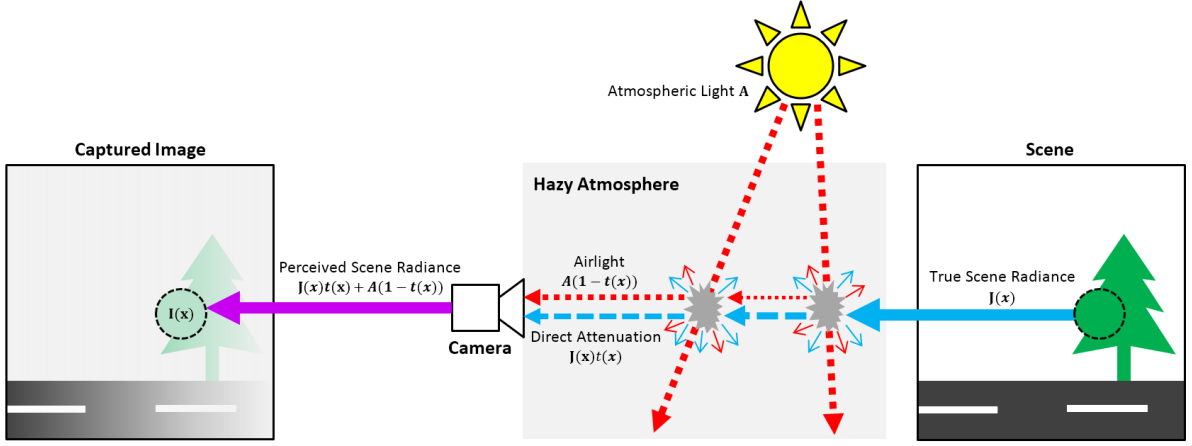


Figure 2: Diagram showing how haze impacts an observer. Here ‘camera’ refers to any optical sensor that projects the 3D scene into a 2D image.

How this impacts the image projected to an observer is described by the haze equation, derived from the haze models of Koschmieder [14] and McCartney [15]. The haze equation states that:

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

Where \mathbf{x} is a vector $[x, y]$ representing a *projected scene point*. \mathbf{A} is an RGB vector representing the atmospheric light. \mathbf{I} is the observed *hazy image*, with $\mathbf{I}(\mathbf{x})$ being an RGB vector representing its colour at \mathbf{x} . \mathbf{J} is the true radiance of the scene (the *clear image*), with $\mathbf{J}(\mathbf{x})$ being an RGB vector representing its colour at \mathbf{x} . Finally, $t(\mathbf{x})$ is the *transmission* at point \mathbf{x} .

$$0 \leq t(\mathbf{x}), I_c(\mathbf{x}), J_c(\mathbf{x}), A_c \leq 1 \text{ for all colour channels } c \in \{r, g, b\}$$

The multiplicative term $t(\mathbf{x})\mathbf{J}(\mathbf{x})$ is often referred to as *direct attenuation* [16, 17], while the additive term $(1 - t(\mathbf{x}))\mathbf{A}$ is often referred to as the *airlight* [14, 16-19]

The transmission $t(\mathbf{x})$ is defined by;

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

\mathbf{d} is the *depth map* of the scene. For a given projected point \mathbf{x} , $d(\mathbf{x})$ is the distance to the scene point from which it has been projected.

β is the scattering coefficient that is determined by the density of the haze. This form in which β is a constant accurately describes homogeneous haze, while only approximating heterogeneous haze. Though heterogeneous haze is more complex to model, many dehazing methods designed around homogenous also work well for heterogeneous haze.

Dehazing¹ is the process of recovering the clear image \mathbf{J} from the haze equation (1). However, the problem is ill-posed. For an h by w image with 3 colour channels there are $h \times w \times 3$ equations, but $h \times w \times 4 + 3$ unknowns. The clear image \mathbf{J} can be recovered if t and \mathbf{A} are known using the haze equation:

$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - \mathbf{A}}{t(\mathbf{x})} + \mathbf{A} \quad (3)$$

However, as illustrated in Figure 3 and Figure 4, the under-constrained problem introduces ambiguity between the atmospheric light, transmission, and true image.

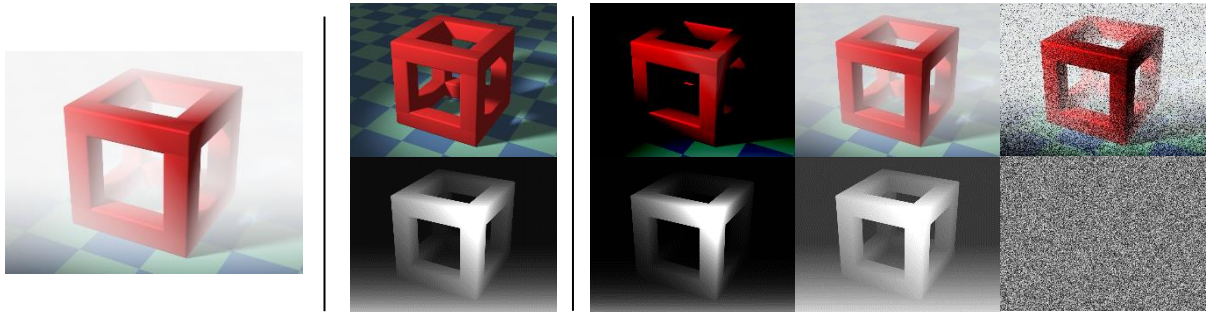


Figure 3: Transmission ambiguity. Using $\mathbf{A} = [1, 1, 1]$. The top row shows recovered 'clear' images, while the bottom shows transmission maps. Each column pair is a valid solution to the haze equation (1), with all equalling the hazy image on the left. However only the first column is the true unknown clear image and transmission.

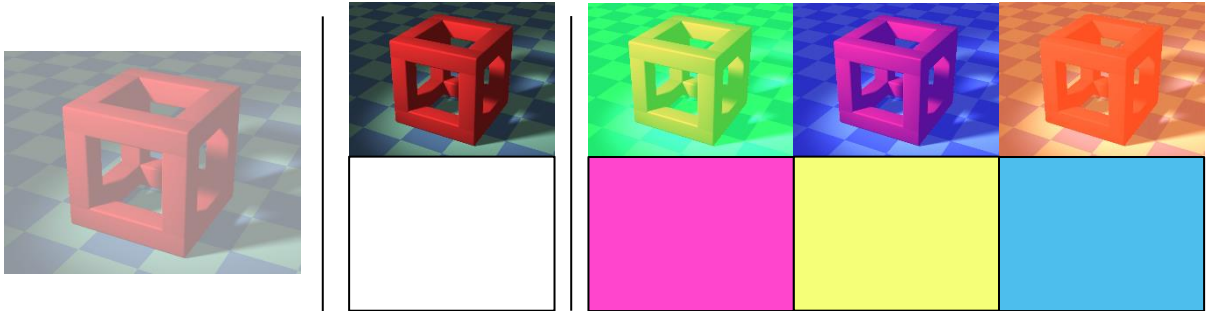


Figure 4: Atmospheric light ambiguity. Using $t(\mathbf{x}) = 0.5$. The top row shows recovered 'clear' images, while the bottom shows atmospheric light colours. Each column pair is a valid solution the haze equation (1), equalling the hazy image on the left. However, only the first column is the true unknown clear image and atmospheric light.

1.2 Research Focus

The primary goal of this dissertation is an investigation into current dehazing method's potential to work effectively under the conditions required of an advanced driver assistance

¹ As with the terms haze and fog, dehazing can also be referred in literature as defogging. Both terms refer to the same process of removing from an image the effects caused by suspended particles in the atmosphere.

system, and to develop on their shortcomings with a new dehazing system. This research looks at dehazing methods from high-level algorithmic perspective and is not concerned with the intricacies of optimised software and hardware implementations. Though these are indeed ultimately needed for the development of a dehazing ADAS, this work provides an important general look at what a dehazing ADAS needs and how the current dehazing techniques are suited for use in such a system. With the goal that these future hardware implementations are built on the most relevant principles with the most effective processes.

Though autonomy is at the forefront of vehicle related research, they are not the focus of this research for two key reasons. Firstly, autonomous driving systems are chiefly operated by convolutional neural networks for the tasks of segmentation, classification and tracking. Research has suggested [20, 21] these networks perform worse or no better on dehazed images than on hazy images if hazy images are included in their training data. Therefore, dehazing is less of an importance to the development of autonomous vehicles, and more of an enduring challenge that faces human drivers. Secondly, there are increasing moral and legal concerns over autonomous vehicles [22-24], hindering their widespread acceptance and usage. In contrast ADAS have become so vital to a safer driving experience that several countries, including the US [25] and Canada [26], require by law all new cars featuring a rear-view camera system.

1.3 Contributions

The project makes the following research contributions:

- A novel haze video dataset containing 18 driving sequences featuring a variety of different haze conditions, totalling 15,719 frames.
- An evaluation of existing dehazing methods using the novel dataset, and a framework of criteria assessing their suitability for use in a dehazing ADAS.
- Three novel image filters; the Smooth-Extremum Filter (SEF), the Smooth-Extremum Local Filter (SELF), and the Approximate Smooth-Extremum Local Filter (ASELF); based on smooth approximations of the maximum and minimum functions.
- A new dehazing system incorporating the SEF and ASELF that improves the dehazing performance and efficiency of the dark channel prior by combining dark channel estimation and refinement into a single step.

1.4 Dissertation Structure

The remainder of this dissertation is divided into five chapters. Chapter 2 provides a review of dehazing techniques. Chapter 3 details the development of a novel haze dataset. Chapter 4 provides an in-depth comparison of dehazing methods using the novel dataset and specially designed criteria. Chapter 5 proposes and evaluates a dehazing system using novel smooth-extremum filters, based upon the outcomes of chapter 4. Chapter 6 concludes this dissertation with a reflection of the research focus and highlights potential future developments.

2 Literature Review

This section provides a look into the history of tackling the dehazing problem, the techniques developed over the years and the current state-of-the-art. This section also includes a discussion on the currently available dehazing-related datasets.

2.1 Single-Image Dehazing

The ill-posed nature of dehazing can be reduced by leveraging additional images of the same scene; through the use of stereo images [27] captured by calibrated cameras, images of a stationary scene under different weather conditions [19, 28, 29], differently polarised images [30, 31], or a known depth [32]. The difficulty in acquiring and utilising these images has led dehazing research to focus more on single-image dehazing, in which the only image available is the hazy image \mathbf{I} . Due to the lack of additional information, single-image dehazing approaches rely on imposing priors and assumptions derived from observations and reasoning of the scenes to be dehazed.

Tan [16] introduced the concept of single-image dehazing. Their method is based on maximising visibility by estimating a transmission value that maximises the magnitude of gradients for each local patch. While this method removes haze almost entirely, it results in overly saturated unnatural images and is easily prone to magnifying noise within the haze.

Fattal [18] represents the true scene radiance as a colour direction and luminance magnitude. Their method uses independent component analysis (ICA) to separate the hazy image into luminance and transmission based on the assumption that they are statistically independent. The method is only valid for pixels around which the luminance and transmission vary significantly and requires the use of a Markov random field (MRF) to fill invalid pixels. While the results are more natural than those of Tan’s method, the method proves to be computationally expensive and inaccurate in dense haze.

2.1.1 Dark Channel Prior

He et al. [17] introduced the concept of the dark channel prior (DCP) for single-image dehazing. The dark channel prior is the observation that for most natural outdoor haze-free images, the value of the darkest channel is zero within a local patch, excluding sky regions. This is due to the prevalence of dark objects, nearby shadows or highly saturated colours. Formally the dark channel prior states that for a natural outdoor haze-free colour image \mathbf{J} ,

$$\min_{\mathbf{x}' \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} (J_c(\mathbf{x}')) \right) \approx 0 \quad (4)$$

For all non-sky regions. Where $\Omega(\mathbf{x})$ is a local patch centred on \mathbf{x} , and J_c is a single colour channel of \mathbf{J} with $c \in r, g, b$. This operation of taking the channel and patch minimum of a colour image is known as the dark channel operation and it can be used to dehaze an image. The dark channel prior method follows the process in Figure 5.

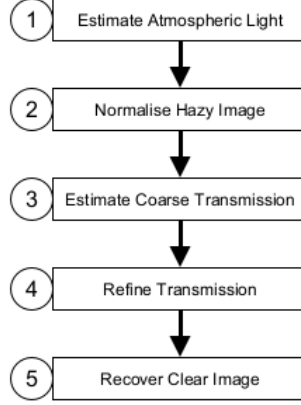


Figure 5: Diagram showing an overview of the DCP dehazing process

Firstly, the atmospheric light \mathbf{A} is estimated by taking the dark channel of the original image \mathbf{I} , finding the brightest 1% of pixels in the dark channel and averaging the values they correspond to in \mathbf{I} . A coarse transmission map is then estimated by inverting the dark channel of the atmospheric light normalised image:

$$\tilde{t}(\mathbf{x}) = 1 - \omega \min_{\mathbf{x}' \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} \left(\frac{I_c(\mathbf{x}')}{A_c} \right) \right) \quad (5)$$

Where A_c is the value of the atmospheric light \mathbf{A} for each colour channel $c \in r, g, b$, and ω is a parameter used to retain a small amount haze in the dehazed image to preserve visual cues, usually set to 0.95.

Alongside the DCP assumption, the coarse transmission map also assumes that transmission is constant within each patch, resulting in a blocky appearance and severe artifacts when used to recover a clear image (Figure 6). As a result, a transmission refinement step is required, this is most commonly accomplished through the use of a guided filter [33, 34]. The full process of transmission map estimation (steps 3 & 4 of Figure 5) is shown in greater detail in Figure 7



Figure 6: The need for refinement in dark channel prior dehazing. The leftmost image is the hazy input. The first row shows the coarse transmission map followed by the result of recovering the image (6) using the coarse transmission map. The bottom row shows the transmission map after guided filter refinement followed by the result of recovering using the refined transmission map. Because the erosion operation to generate the coarse transmission map is inherently ‘blocky’, if it is directly used to recover an image it results in blocky halo borders, particularly around detailed regions with large depth discontinuities.

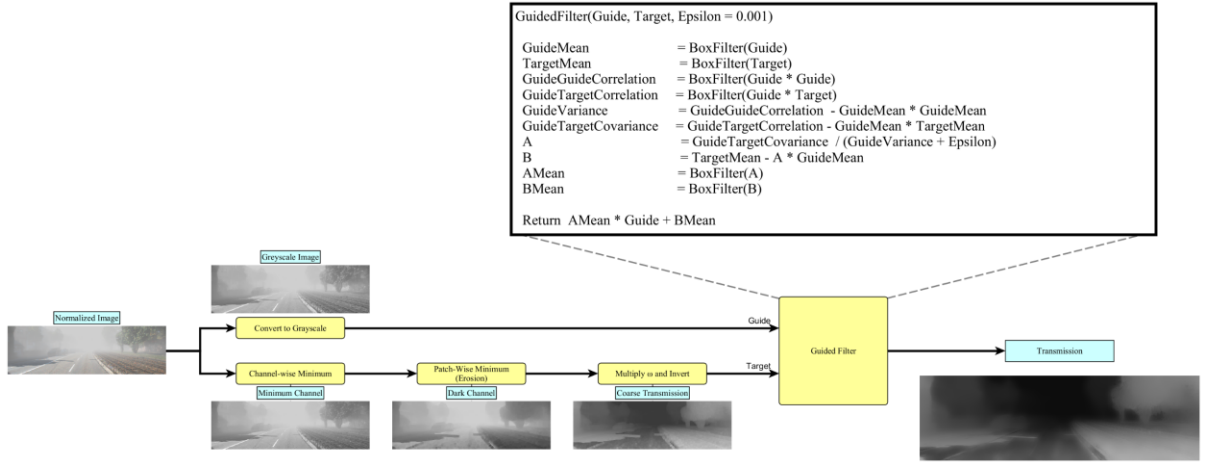


Figure 7: Diagram showing steps 3 & 4 of the DCP prior dehazing process (Figure 5) in greater detail. In this instance, the guided filter is used for refinement, with pseudocode included.

Once the atmospheric light and refined transmission have been estimated, the dehazed image is recovered as follows:

$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - \mathbf{A}}{\max(t(\mathbf{x}), t_0)} + \mathbf{A} \quad (6)$$

t_0 is a parameter used to avoid magnifying noise in dense regions of haze, usually set to 0.1.

The dark channel prior dehazing method is very effective at removing haze and for the most part, produces natural looking images. However, a major downside of the dark channel prior is that it cannot handle large bright unsaturated objects, as these tend to break the DCP assumption. These bright objects become overly saturated and noisy when the image is recovered. Increasing the patch size decreases this effect – as a larger patch size decreases the likelihood of DCP assumption being broken – but doing so also increases the likelihood that the constant-patch transmission assumption is broken and results in a low-information transmission map. Furthermore, the dark channel prior method suffers from halo artefacts around depth discontinuities that result from the blocky coarse transmission map, while refinement intends to reduce this as much as possible, these artefacts still remain. Finally, while most of the process is computationally fast, the refinement step is usually a costly procedure.

Since its inception, the DCP has become ubiquitous with dehazing and many dehazing methods incorporate the assumptions made by the DCP in some way, or follow the DCP process [35]. DCP-based methods improve upon the steps shown in Figure 5, in various ways. For example, the refinement step for the original DCP method used a soft-matting MRF, however later DCP-based methods have used the bilateral filter [36] and guided filter [33] amongst others. Due to fast implementations [34], the guided filter is now the dominant transmission refinement filter for DCP-based methods [35].

2.1.2 Haze-Lines

The majority of dehazing methods are local [3, 35], they apply operations on local patches of an image which can result in artefacts around depth discontinuities. Berman et al. [37] introduced a non-local dehazing method based upon the concept of haze-lines. Their method uses the prior that the number of unique colours within a clear image is vastly less than that of a hazy image, because any pixels of a same unique colour in a clear image can be mapped to multiple different colours in the presence of haze depending on their depth. Unique colours form clusters in RGB space in the clear image, that become haze-lines in the hazy image, as each colour is linearly transformed to some point on the RGB space line between its original colour and the atmospheric light colour.

To estimate the transmission of a hazy image (2), all pixels are translated so that the atmospheric light is at the origin. Each pixel is then converted into spherical coordinates such that each latitude and longitude pair define a haze-line, while the radius component represents a distance along the haze-line. Pixels are grouped based on their latitude and longitude under

the assumption they belong to the same haze-line perturbed by noise. The colour with the furthest radial distance for each haze-line is assumed to be the true haze-free colour of all pixels in that group. Transmission is calculated for each pixel colour by dividing its radial distance by the maximum radial distance in its haze-line. Finally, transmission is spatially regularised using weighted least squares before being used to recover the dehazed image.

Though unique for being non-local and avoiding certain artefacts common to local methods, this benefit is not very pronounced and in fact most of the local methods mitigate these artefacts themselves when properly parameterised. Furthermore, despite its complexity being linear in the number of pixels, the grouping of haze-lines and regularisation of the transmission map is computationally expensive compared to many other dehazing methods.

Berman et al. later introduced a method of estimating the atmospheric light [38] by finding the intersection of haze-lines using Hough voting on the RG, RB and GB colour planes.

2.1.3 Colour Attenuation Prior

Zhu et al. [39] introduced the colour attenuation prior. This prior assumes that the depth of a hazy image can be modelled as a linear function of the saturation and value components of the image in HSV space. Using gradient descent to optimise the maximum likelihood function of their linear model, the authors found optimal coefficients that can be used for any subsequent dehazing efforts. After pixel depth is calculated from the linear model a patch minimum operation is applied – similar to DCP methods – followed by a colour guided filter refinement. As with other dehazing methods that require a refinement step, this process is the most computationally expensive. This is especially true for variant of the guided filter used – the colour guided filter – as it requires solving the inverse of a 3 by 3 matrix for each pixel. Despite this, their method outperformed the original DCP method (using soft-matting) and other contemporary methods at the time in both speed and SSIM scores for the dataset used. Though it’s worth noting their evaluative dataset consisted of less than 10 images. The method also uses a similar atmospheric light estimation method to the DCP method. The brightest 0.1% of pixels are selected from the estimated depth map, then of the corresponding pixels in the hazy image, the colour with the largest Euclidean norm is chosen as the atmospheric light.

2.1.4 Convolutional Neural Networks

In recent years, convolutional neural networks (CNN) have dominated the literature of computer visions problems, and dehazing is no exception. A seminal dehazing CNN is DehazeNet [40]. This network uses a multi-structured architecture flow of feature extraction,

multi-scale mapping, local extremum and non-linear regression to estimate transmission maps. The model is trained on image pairs consisting of small (16x16) patches of natural images and equivalent patches with synthetic haze added from a constant transmission map. Despite the assumptions made in the training data – namely constant transmission in a 16x16 region and that the transmission values are independent of image content – the network generalises relatively well and produces decent results on a variety of hazy images.

Li et al. introduced the All-in-One Dehazing Net [41] (AOD), a very lightweight dehazing CNN composed of only 5 convolutional layers in a multiscale skip-concatenate pattern and a final reconstruction layer. Through experimentation, the authors demonstrated that reformulating the haze equation and estimating on a combined form of t and \mathbf{A} provides better learning than estimating either individually. The network learns to predict the vector valued \mathbf{K} defined as:

$$\mathbf{K}(\mathbf{x}) = \frac{\frac{1}{t(\mathbf{x})}(\mathbf{I}(\mathbf{x}) - \mathbf{A}) + \mathbf{A} - b}{\mathbf{I}(\mathbf{x}) - 1} \quad (7)$$

\mathbf{K} is predicted by the convolutional layers, which is then used in the reconstruction layer to output the dehazed image as follows:

$$\mathbf{J}(\mathbf{x}) = \mathbf{K}(\mathbf{x})\mathbf{I}(\mathbf{x}) - \mathbf{K}(\mathbf{x}) + b \quad (8)$$

Where b is an additional scalar bias term learned by the reconstruction layer. This allows the network to directly predict clear images and train only using pairs of hazy and clear images without the need for ground truth atmospheric light and transmission maps. The network outperforms early dehazing CNNs and prior-based methods in \uparrow SSIM and \uparrow PSNR on the dataset tested. Additionally, the network is light enough to run in less than a second on a CPU.

Patch-Map Hybrid-Learning DehazeNet (PMHLD) [42] is recent dehazing CNN that combines many state-of-the-art neural network and dehazing techniques, incorporating both DehazeNet and the DCP. A sub-network predicts a patch-map to use in conjunction with the DCP method of transmission estimation, where each value in a patch-map dictates the size of the patch minimum to apply centred on that pixel. It combines this sub-network with atmospheric light estimation, and transmission refinement sub-networks to achieve the final dehazing network. They augment the network’s learning by training it as part of a generative adversarial network (GAN) comprised of the dehazing network as a generator and a

discriminator network that learns to predict if a given output of the dehazing network is hazy or clear. As the discriminator learns to detect hazy and clear images better, the dehazing network is forced to produce more realistic images. Though it performs exceptionally well at dehazing a variety of scenes, outperforming AOD in \downarrow MSE, \uparrow SSIM and its use of combining machine learning with the DCP is novel, compared to AOD the network is very large – requiring a GPU to achieve decent speeds.

2.2 Video Dehazing

Any single-image dehazing method can be extended to video by simply applying the dehazing to each individual frame of the video. While this works effectively, having no information about the previous frames causes flickering and other temporal artefacts [4]. Various video dehazing methods have been designed specifically for leveraging video features to speed up computation, or to impose additional priors through regularisation – reducing temporal distortions.

Dong et al. [43] use the advantage that for a static camera, most pixels will remain constant over time. They use this to only dehaze regions – using the DCP method – in regions that have changed significantly over time.

Cai et al. [44] introduced a dehazing approach based on the DCP using a spatio-temporal Markov random field (ST-MRF) to smoothly regularise the transmission map spatially and temporally. The algorithm calculates a coarse transmission map similar the DCP however using non-overlapping blocks rather than pixel centred patches. The refinement process maximises a log-likelihood function relating the intensity of nearby pixels and its intensity changes over time, to the concentration of haze. The atmospheric light is estimated by averaging the pixels in the original image corresponding to the largest average value over the block-minimum of each colour channel. This estimated light is updated smoothly over time such that:

$$\mathbf{A}_n = \rho \mathbf{A} + (1 - \rho) \mathbf{A}_{n-1} \quad (9)$$

Where \mathbf{A} is the atmospheric light estimated each frame, and \mathbf{A}_t is the atmospheric light used to dehaze frame n . In order to dehaze a given frame it requires the use of both adjacent frames for the MRF, meaning that for a live video it will incur delay proportional to the framerate of the live feed. Due to the simplistic likelihood function used in the MRF, this method achieves fast frame rates and had the best MSE values of the methods tested on the

small number of images evaluated. This paper also introduced the concept of using the correlation between mean intensity values of a clear and dehazed video over time as a measure of temporal quality, with their method scoring highest in this new metric when tested on constant transmission hazy images.

2.3 ADAS Dehazing

ADAS dehazing is a form of video dehazing, thus all image dehazing methods can be adapted into an ADAS by applying their process frame-by-frame. Similarly, video dehazing methods could be used in an ADAS, however it's important to note two inherent aspects concerning the video used by a dehazing ADAS. Firstly, as the camera is intended to replicate the view of the driver, it is relative to the vehicle as it moves about the scene. In other words, the camera's pose (position and orientation) throughout the video is constantly changing. This precludes the use of any techniques that rely on identifying static background pixels or otherwise leveraging the constancy of certain regions over time, such as [43]. Secondly, the video is live. This means that any methods requiring frames ahead of the current frame incur a delay, as the system must wait for these frames to become available. For example, Cai et al.'s method [44] (described in the previous section) can be used as an ADAS, however it will always be one frame behind its live feed. How impactful this will be, depends upon the framerate of the camera used for the ADAS.

Because of these limitations, many attempts at dehazing for an ADAS stick to dehazing frame-by-frame. Tarel, Hautière and their team have done extensive work on dehazing from the perspective of an ADAS [8-12]. In [11] they introduce the no-black pixel constraint (NBPC) and combine it with the planar assumption (PA) [9] to create a dehazing algorithm. The no black pixel constraint states that for all local patches in \mathbf{J} , the standard deviation of the patch multiplied by a constant f , is lower than the mean. Assuming the atmospheric light is 1, the NBPC is applied to find \mathbf{J} in the following way:

$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - V(\mathbf{x})}{1 - V(\mathbf{x})} \quad (10)$$

Where:

$$V(\mathbf{x}) = \rho \min(L(\mathbf{x}), \bar{L} - f \text{std}(L)) \quad (11)$$

and:

$$L(\mathbf{x}) = \min_{c \in \{r, g, b\}} (I_c(\mathbf{x})) \quad (12)$$

Finally, ρ is a parameter controlling the amount of haze remaining in the dehazed image, set to 0.95 in the paper. The planar assumption leverages the knowledge that for a car mounted dehazing system, there will always exist a road in the image. With the uncommon exceptions of a visible change in road gradient, the surface of the road can be perfectly modelled as a plane projected onto the image. Using the camera’s calibration parameters and a known vertical position of the horizon line, the depth to the surface of the plane can be calculated. As nothing will ever be visible beneath the road, this projected depth map provides an upper bound for the depth map of the image as a whole. If β is known this can be converted into a lower bound for the transmission map. In [11] they use a fixed β representing worst case visibility. Used on its own, the planar assumption method does not dehaze well, but when used as a lower bound combined with the NPCA it prevents colours on the road that are close to the atmospheric light colour from becoming overly saturated when dehazed due to underestimated transmission. Finally, part of the study in [11] explored the potential of dehazing systems to reduce the reaction time of drivers by examining Piéron’s law [45] relating stimulus intensity to reaction time.

Tsai et al. [13] use a DCP-based approach adapted for video. Atmospheric light is estimated to be the average of all pixels with an intensity brighter than a given threshold. Similar to Cai [44] this method calculates the atmospheric light every frame but updates it as a linear combination of the new and previous value. The method incorporates an addition to the guided filter’s process in order to generate a sky mask from the differences between local means and standard deviations. Using this sky mask they clamp the transmission in regions detected as sky to avoid overblown colours.

2.4 Dehazing Datasets

To aid in the training of machine-learning of dehazing models, and for the evaluation of dehazing systems in general, many hazy image datasets have been developed. This section presents a review of existing dehazing datasets. While there are a wide variety of dehazing datasets, only the subset relevant to this project – those featuring outdoor daytime scenes – will be discussed. An overview of available datasets is presented in Table 1.

In an idealised dehazing dataset each image or frame would contain real ground truth values for all variables in the haze model; their relationship would satisfy the haze model; and their values – including the image itself – would be sampled from the true underlying distribution of which a dehazing system would be applied. In reality though this is a very difficult task and there is often a trade-off between the number of known variables, the accuracy of variables with regard to the scene itself, and the divergence of the distribution from which \mathbf{I} and \mathbf{J} are sampled to the underlying distribution. Dehazing datasets can be generally grouped into three categories that exemplify this trade-off: fully-synthetic, haze-synthetic, and real haze.

Table 1: An overview of existing outdoor haze datasets

Dataset	Type	Total Clear Images	Total Images	Video	Road
FRIDA [12]	Fully-Synthetic	18	90	No	Yes
FRIDA II [11]	Fully-Synthetic	66	330	No	Yes
RESIDE (OTS) [46]	Haze-Synthetic	313,950	313,950	No	No
HazeRD [47]	Haze-Synthetic	14	70	No	No
Foggy Cityscapes [48]	Haze-Synthetic	5000	15000	No	Yes
O-Haze [49]	Real Haze	45	45	No	No
Dense-Haze [50] (outdoor)	Real Haze	33	33	No	No

2.4.1 Fully-Synthetic Datasets

Fully-synthetic datasets are comprised of clear and hazy images taken from computer generated environments. As the environment is synthetic, all values in the haze model can be known with full accuracy. A set of images are taken from the clear scene, for each of these, a wholly accurate depth map is known. \mathbf{A} and β are selected based on estimations of their underlying distribution relating to the real-world counterpart of the artificial environment. These values are input into the haze equation to generate a set of hazy images. The three-dimensional environment along with a value for \mathbf{A} and β constitutes a scene. Scenes with a β of zero are therefore considered clear scenes and those with a value greater than zero are considered hazy scenes.

Multiple hazy images of different hazy scenes can be generated for each clear image of a clear scene, which allows for large amounts of samples can be generated for a variety of haze conditions. Furthermore, as all values are known without significant error, these datasets allow for accurate analysis and subsequent inferences of the relationships between \mathbf{I} , \mathbf{J} , the hazy scene, and the clear scene.

The downsides of fully-synthetic haze datasets are that **I**, **J**, the hazy scene, and the clear scene are less representative of real-world images and scenes that would be encountered by an actual dehazing system. How closely **I** and **J** match the distribution of natural hazy and clear images, respectively, depends on the level of photorealism the environment can be rendered into an image. Furthermore, although heterogeneous haze can be easily generated for each image **J** of the clear scene, by using a scattering map function $B(x)$ in place of a scalar β , this scattering map is only easily calculated if it is a function of variables already projected by the observers perspective (i.e. depth map values, pixel co-ordinates, etc.) leading to inconsistent haze for a non-static observer. To be scene-consistent the scattering map would need to be a function of scene itself before projection. This is not a problem for the dataset if the hazy images are treated in isolation as each can be considered belonging to its own unique hazy scene, but it becomes a problem when considering video or scene-based tasks.

FRIDA [12] and FRIDA II [11] are full-synthetic haze datasets both feature images of artificial road scenes under different haze conditions. While their focus on road scenes is ideal for evaluating dehazing ADAS, their usefulness is limited by their small size, featuring 18 and 64 base clear images respectively, and lack of video considerations.

2.4.2 Haze-Synthetic Datasets

Given a real image of a clear scene, a hazy image can be generated if the depth map for the scene – given the observer’s perspective – is known. **A** and β can be chosen from their well-known natural distribution resulting in enough information to calculate the hazy image **I**. Haze-synthetic datasets follow this practice by using images with known depth information to generate equivalent hazy images. This type of dataset is the most commonly used as there are many existing datasets with ground truth depth information, and a large field of work in depth prediction. Like fully-synthetic datasets, many different haze conditions can be generated from a single clear scene, however the advantage over fully-synthetic datasets is that the images **I** and **J** more closely match the natural underlying distribution.

How well generated hazy samples match the distribution of true hazy images, and the accuracy of the relationship between **I** and its hypothetical hazy scene, depends largely on the accuracy of the depth map. Errors in the depth map will propagate to the generated hazy image. Another downside is that in the haze model, the original image **J** is assumed to be clear, if it not the generated hazy image will be inaccurate. There are also issues with lighting conditions. For example, in real-world haze – unless it is very localized – the sky above will be overcast,

an image captured in these scenes will lack strongly defined shadows bright specular highlights. These will remain when haze is added resulting in a less accurate **I**. Finally, the same problems that fully-synthetic datasets face with heterogeneous haze apply to haze-synthetic datasets as well.

RESIDE (OTS) [46] is haze-synthetic dataset featuring a significant number of outdoor hazy images. However, its usefulness is limited by the poor quality of the depth maps used to generate the hazy images is poor.

HazeRD [47] features 14 clear images of outdoor scenes with depth maps and provides code to generate hazy images.

Foggy Cityscapes [48] is built upon the Cityscapes dataset [51], featuring hazy scenes generated from real world road scenes. The large number of images and its vehicle perspective make this the most ideal existing dataset for evaluating dehazing ADAS, however it is not designed for video use and only contains disparate, non-sequential images – precluding its use in training and evaluating dehazing systems with video considerations.

2.4.3 Real Haze

Since haze is real phenomena, it is a simple task to gather images of real scenes with real haze. While a dataset that contains only labelled instances of real clear and real hazy images is still useful, it is far more useful to contain corresponding pairs of clear and hazy images. Unfortunately, it is very difficult to capture the exact same scene under both hazy and clear conditions. The only changing factors between the two scenes should be the scattering coefficient / function, however haze takes a long period of time to form and dissipate naturally. When an image of one weather condition is captured, by the time this has changed to a different weather condition, other changes to the scene invariably occur such as lighting differences and object movement. To overcome this, some researchers have used synthetic haze machines that physically haze within the scene. Because these machines can generate haze much faster than it forms naturally there is minimal different in the scene before and after the haze is introduced.

As the haze is physically present in the scene, the resulting captured images are far more natural in appearance. Allows for more diverse and realistic haze ‘shapes’ as the (unknown) scattering function in the haze model for the hazy images is a realistic function of the particles in the air of the scene, that may have differing concentrations according to many different physical laws and properties of the scene. This means that the hazy images obtained

are sampled from a distribution almost exactly matching that of the natural distribution from which hazy images requiring dehazing will come.

The significant downside of real haze datasets is the limited amount of ground truth available. Moreover, it is time consuming to acquire many different hazy scenes for a given clear scene.

O-Haze [49] and Dense-Haze [50] are real-haze datasets featuring images of clear scenes and corresponding images of the same scene after haze is introduced using a haze machine.

3 Novel Video Haze Dataset

Currently there exists no video haze datasets for road scenes; a necessity when evaluating how a dehazing system would perform as an ADAS. One of the contributions of this study is the introduction of a novel hazy video dataset² featuring 18 hazy sequences for a total of 15,719 hazy frames and corresponding transmission map.

3.1 Methodology

The haze-synthetic approach (described in detail in 2.4.2) was chosen for the new dataset due to the benefits outlined in that section, as well as the availability of existing dashcam video datasets to use as a base. The process of generating the new dataset follows four steps; selection of the base data to use for clear ground-truth images, depth map generation, sky mask generation, and haze generation.

3.1.1 Base Clear Sequences

KITTI [52] is a widely used driving dataset containing a large amount of dashcam video sequences with a featuring a variety of additional ground truth information. The varied range of road scenes and metadata available in the KITTI dataset made it the ideal choice for the clear image sequences. The following rectified KITTI sequences were hand selected to contain a varied mix of different driving environments³.

Table 2 Selected KITTI sequences for the novel dataset

Sequence	Frames	Runtime (m:ss)	KITTI Category
2011_09_26_drive_0001	108	0:10	City
2011_09_26_drive_0009	447	0:44	City
2011_09_26_drive_0015	297	0:29	Road
2011_09_26_drive_0022	800	1:20	Residential
2011_09_26_drive_0028	430	0:43	Road
2011_09_26_drive_0036	803	1:20	Residential
2011_09_26_drive_0051	438	0:43	City
2011_09_26_drive_0061	703	1:10	Residential
2011_09_26_drive_0064	570	0:57	Residential
2011_09_26_drive_0093	433	0:43	City
2011_09_26_drive_0096	475	0:47	City
2011_09_26_drive_0101	936	1:33	Road
2011_09_26_drive_0117	660	1:06	City

² The completed dataset can be downloaded from <https://essexuniversity.box.com/s/7hzugcfiv14b6kevpb02ppp1z4d8jfrs>.

³ These sequences can be downloaded from the KITTI website: http://www.cvlibs.net/datasets/kitti/raw_data.php

2011_09_28_drive_0002	376	0:37	City
2011_09_29_drive_0071	1059	1:45	City
2011_09_30_drive_0028	5177	8:37	Residential
2011_10_03_drive_0042	1170	1:57	Road
2011_10_03_drive_0047	837	1:23	Road

3.1.2 Depth Map Generation

Given a clear frame from a sequence, all that is required to generate an arbitrarily hazy frame is a depth value for each pixel. As dense ground truth depth maps are not available, for each frame of the KITTI sequences, generating full depth maps is required.

Each KITTI sequence contains stereo images and sparse LiDAR maps. LiDAR maps are sparsely populated ground truth depth maps. Given this, the natural assumption to produce a dense depth map would be to use stereo-matching or sparse depth completion techniques as these make the most use of ground truth data available and are likely to produce the most accurate depth maps. However, due to various factors this is not true. In the case of depth completion using KITTI LiDAR maps, what’s important to note is that the LiDAR maps only cover the bottom half of the image [Need picture examples]. The majority of depth completion methods – including those that leverage the original image as a guide – simply ignore the regions above which there is no LiDAR data, as no ground truth is available to learn how this region should be predicted or evaluate how accurately it has been predicted. To ignore this region is satisfactory in the vast majority of cases where the completed depth map is required for further algorithms, as all information needed is contained within or near to the region mapped by the LiDAR. However, the focus of this research is dehazing for the benefit of a human end-user in the form of an ADAS. If a human driver is only presented with a clear view corresponding to roughly the bottom half of their windscreen, they will find this unfamiliar and disorienting – regardless of whether this view contains all the information they need to drive safely. In order to be fully useful as a tool for developing and evaluating ADAS dehazing systems, the dataset needs to contain the upper region, thus a dense depth map for this region is required, ruling out the possibility of using depth completion.

Three different methods (described below) were used to generate depth maps for the sequences: MonoDepth2, ManyDepth, and LEAStereo. From this comparison, depth maps

from ManyDepth were deemed the most suitable and are used in the generation of the final hazy dataset. Each method is compared below.

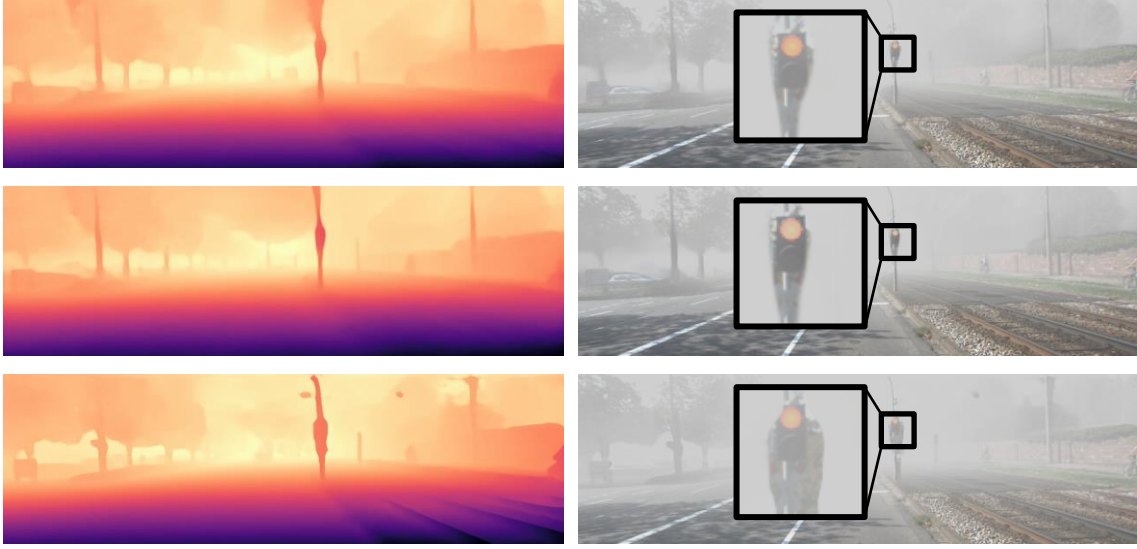


Figure 8: A comparison of depth maps and corresponding generated hazy images (with sky masks applied) for the three depth map generation methods. From the top; MonoDepth2, ManyDepth, LEAStereo. Haze was generated using a $\mathbf{A} = [0.814, 0.814, 0.814]$ and $\beta = 0.0342$. Note also that the centre top region, where the sky is visible in the clear image they derive from, all three depth maps contains varying non-maximal depths, however these should be a flat region of maximum value (represented as white)

MonoDepth2 [53]

MonoDepth2 is a monocular depth estimation neural network. MonoDepth2 estimates a depth map using only the input colour image. MonoDepth2 was chosen for its simplicity to provide an initial baseline for how the hazy generated sequences would look. Despite only being a monocular depth method, the haze generated using MonoDepth2 depth maps proved to be far more visually correct than expected, though still the worst out of the three methods tested.

ManyDepth [54]

ManyDepth is a video depth estimation neural network with a similar architecture to MonoDepth2, with the added benefit of leveraging the information between adjacent frames. ManyDepth estimates a depth map for a given frame using its image and the images of its adjacent frames. As this method was designed specifically for video, the generated hazy sequences using these depth maps featured fewer flickering artefacts and in general suffered from less temporal artefacts. While in general the values are not as accurate as those of LEAStereo depth maps, this proved to be less of an issue for the generated hazy images than the edge issues caused by LEAStereo.

LEAStereo [55]

LEAStereo is a stereo-matching neural network, using an architecture developed through a neural architecture search. At the time of developing the dataset, LEAStereo was the highest ranked method on the KITTI stereo ranking. LEAStereo estimates a depth map from a pair of stereo images. Compared to the other two methods, LEAStereo produced sharper depth discontinuities. While the gradient of these depth discontinuities is more accurate to a true depth map, it made errors in the location of the depth discontinuities more apparent when the depth discontinuities were estimated in the wrong place. This resulted in unrealistic haze-free borders around objects more pronounced than the other two methods. LEAStereo depth maps were also more susceptible to temporal noise.

3.1.3 Sky Mask Generation

Depth maps from all three methods suffer from issues with sky regions. Depth prediction neural networks – both stereo and mono – learn from LiDAR data that is generally not available in the upper half of the image. As the networks have no training data or any textural cues for sky regions, they tend to treat the upper edge of an images as they do the lower edge (Figure 9), which invariably features the ground. In essence what the neural networks learn is that; without any additional information to say otherwise, as you reach the edge of the image, depth decreases.



Figure 9: The result of inputting a uniformly white image into ManyDepth. Note how the depth values change toward both the top and bottom edges.

As mentioned in the previous section, this is more of an issue with depth completion methods in which the sparse LiDAR map is the primary input – in these cases the ‘tunnelling’ effect spreads beyond the sky region into objects and cannot be easily fixed, this was major factor in the choice of using depth prediction rather than depth completion. However, for depth prediction neural networks, that use images as the primary input, this effect is usually only

limited to the sky regions meaning that if the values in sky region can be replaced by infinity (or a large enough value), the depth map will be accurate.

In order to replace the sky regions, a mask needs to be generated indicating which pixels are sky. A modern sky detection neural network – SkyAR [56] – was employed to generate a sky mask for each frame of the original KITTI sequences. Rather than applying to the generated depth maps \mathbf{d} – which would require the sky regions to be masked with the value of infinity – the transmission maps derived from the depth maps are multiplied by the inverted sky mask. The full sky adjusted transmission is calculated as:

$$t'(\mathbf{x}) = (1 - m(\mathbf{x}))e^{-\beta d(\mathbf{x})} \quad (13)$$

This forces the transmission values predicted as sky toward zero, where a transmission of zero represents complete attenuation and only atmospheric light reaches the camera.

No dehazing system will be able to predict the information in a true sky region (i.e. infinite depth and zero transmission) as in the hazy image this becomes equal to the atmospheric light colour. Due to this, the ground truth clear image for a given hazy image in this dataset is considered to be the original clear image matted with the atmospheric light colour according to the sky mask. This is explained in greater detail in 4.3.

3.1.4 Haze Properties

While many authors use a fixed value of $\mathbf{A} = [1,1,1]$ for atmospheric light in the generation of haze, this is rarely reflective of real-world. FoggyCitiscaes [48] uses an atmospheric light estimated from the clear image in a similar manner to how atmospheric light is estimated for hazy images. However, this will not work for truly clear weather scenes featuring sunny conditions as the atmospheric light is invariably estimated to be the blue of the sky – resulting in blue synthesized haze, which is far from a common occurrence in the real world. Because of the large number of images in the dataset, it is impractical to multiple sequences from each clear sequence with different \mathbf{A} and β values. Instead each clear sequence was assigned a fixed, randomly chosen \mathbf{A} and β . \mathbf{A} was selected to be monochromatic with an intensity chosen uniformly at random between 0.7 and 1 as in [46]. β was selected to represent a visibility chosen uniformly at random between 100m and 1000m. This is a similar range to [47] however visibility less than 100m is omitted, as this is the visibility below which fog lights are recommended, resulting in considerably different scenes from a drivers perspective that cannot be accurately recreated in the dataset.

Each visibility level is grouped into a haze category (Table 3) to reflect common notions of haze through linguistic labels and to allow for concise comparisons of results on haze with similar properties.

Table 3: Haze categories delineated by visibility

Visibility Range (m)	β Range	Haze Category
100–400	39–9.8	Dense
400–700	9.8–5.5	Moderate
700–1000	5.5–3.9	Light

3.1.5 Haze Generation

With a generated depth map \mathbf{d} , generated sky mask \mathbf{m} , and original clear image \mathbf{J} for each frame, alongside a chosen atmospheric light \mathbf{A} and scattering coefficient β for each sequence, A hazy frame \mathbf{I} is generated according to the following:

$$\mathbf{I}(\mathbf{x}) = t'(\mathbf{x})\mathbf{J}(\mathbf{x}) + (1 - t'(\mathbf{x}))\mathbf{A} \quad (14)$$

Where t' is the sky mask adjusted transmission:

$$t'(\mathbf{x}) = (1 - m(\mathbf{x}))e^{-\beta d(\mathbf{x})} \quad (13)$$

3.2 Discussion

Using this method, 18 hazy sequences totalling 15,719 frames were generated. The dataset itself contains the generated transmission t' , and the generated hazy image \mathbf{I} for each frame, as well as a property file for each sequence detailing the atmospheric light and visibility alongside additional metadata. Table 4 contains a summary of the dataset. Examples of the dataset can be seen in Figure 10.

Table 4: Overview of the generated haze dataset.

Sequence	Frames	Runtime (m:ss)	KITTI Category	A Intensity	Visibility (m)	Haze Category
2011_09_26_drive_0001	108	0:10	City	0.8137	114	Dense
2011_09_26_drive_0009	447	0:44	City	0.7623	252	Dense
2011_09_26_drive_0015	297	0:29	Road	0.7898	719	Light
2011_09_26_drive_0022	800	1:20	Residential	0.7260	382	Dense
2011_09_26_drive_0028	430	0:43	Road	0.9342	618	Moderate
2011_09_26_drive_0036	803	1:20	Residential	0.7288	516	Moderate
2011_09_26_drive_0051	438	0:43	City	0.8488	745	Light
2011_09_26_drive_0061	703	1:10	Residential	0.8664	977	Light
2011_09_26_drive_0064	570	0:57	Residential	0.9053	941	Light
2011_09_26_drive_0093	433	0:43	City	0.8276	147	Dense
2011_09_26_drive_0096	475	0:47	City	0.8956	698	Moderate
2011_09_26_drive_0101	936	1:33	Road	0.7546	385	Dense
2011_09_26_drive_0117	660	1:06	City	0.8878	568	Moderate
2011_09_28_drive_0002	376	0:37	City	0.8986	351	Dense
2011_09_29_drive_0071	1059	1:45	City	0.8137	380	Dense
2011_09_30_drive_0028	5177	8:37	Residential	0.7594	759	Light
2011_10_03_drive_0042	1170	1:57	Road	0.9689	630	Moderate
2011_10_03_drive_0047	837	1:23	Road	0.9650	433	Moderate

While the dataset as presented as is, with a single hazy sequence for a given KITTI clear sequence, code is available for generating sequences with arbitrary atmospheric light colours and visibility using any provided depth maps and sky masks. Code is also available for generating new depth maps using the three methods and for generating sky masks with SkyAR.



Figure 10: Example images of the novel haze video dataset showing light (top), moderate (middle) and dense (bottom) haze categories. First column shows the original KITTl image, second column shows the generated hazy image. Top row shows sequence 2011_09_26_drive_0061 frame 0. Middle row shows sequence 2011_09_26_drive_0101 frame 217. Bottom row shows sequence 2011_09_26_drive_0001 frame 0. \mathbf{A} and β for these sequences can be found in Table 4.

The novelty of this dataset brings with it imperfections that are ripe for further research and improvement, though must be kept in mind during the following evaluations. The problems with the dataset decrease as the density of synthetic haze increases, meaning that the upcoming results and inferences are more valid for the dense haze category than the moderate or light categories. The sky masks and depth maps used are not entirely accurate. In the case of the sky masks, this inaccuracy is mitigated when using full-reference metrics, by incorporating the atmospheric light into the ground truth using the sky mask – effectively cancelling out any erroneous sky regions. The most notable flaw of the dataset, however, arises from the fact that the original KITTl data is gathered mostly from sunny days. As a result, the sequences feature strong directional shadows, glares, and reflections that would not be present in a homogeneous haze environment as the sky would be overcast. The process of synthesising haze is unable to accommodate for these features unless there is a sufficient density of haze to cover them up.

4 Evaluation of Existing Methods

4.1 Dehazing Method Selection

This section describes the dehazing methods implemented as dehazing systems and the reasoning for their choice in the comparison. The evaluated dehazing systems are based upon state-of-the-art dehazing algorithms and techniques, with methods chosen to provide a broad mix of ubiquity and uniqueness. In total 8 existing dehazing methods were implemented as dehazing systems. These systems were then evaluated on the novel haze dataset. A full list of the parameters used for each dehazing system, and the origin of their values, can be found in the appendices.

Berman et al. [37, 38]

The **BermanDehazer** uses the haze-line method of dehazing [37] and atmospheric light estimation [38]. This method was selected as it is fairly unique amongst dehazing methods for being non-local. The details of this method are described in 2.1.2

Cai et al. [44]

The **CaiDehazer** is based on the DCP with spatio-temporal Markov random field regularisation [44]. This method was selected as it provides consideration for video rather than acting individually on each frame. The details of this method are described in 2.2. It's worth noting that as the regularization requires both adjacent frames, it incurs a delay of a single frame. The time delay this would cause is inversely proportional to the frame rate of the input stream.

Chen et al. [42]

The **ChenDehazer** uses a pre-trained the PMHLD neural network. This method was selected as it is a state-of-the-art neural network that also leverages the commonly used DCP. The details of this method are described in 2.1.4 Note that although the PMHLD network predicts atmospheric light and transmission internally, **ChenDehazer** uses the recovered output of the network and does not provide an estimate for atmospheric light or transmission.

He et al. [17, 33, 34]

The **HeDehazer** implements the DCP method [17] and uses a fast guided filter for transmission refinement [33, 34]. This method was selected as the DCP is now ubiquitous with dehazing, with many other methods being based upon it. The details of this method are described in 2.1.1

Li et al. [41]

The **LiDehazer** uses a pre-trained AOD neural network [41]. This method was selected as it is a state-of-the-art dehazing neural network that is also very lightweight and has been shown to run efficiently even on CPU. The details of this method are described in 2.1.4

Tarel et al. [9-12]

The **TarelDehazer** is based on the NBPC+PA [9-12]. This method was selected as it was designed to specifically handle road scenes and provided the best results of all methods compared for such a purpose. The details of this method are described in 2.3

Tsai et al. [13]

The **TsaiDehazer** is an improvement of the DCP method with adaptive atmospheric light and sky region detection [13]. This method was selected as it is a modern extension of the DCP method, and its authors focus on its potential use as an ADAS. The details of this method are described in 2.3

Zhu et al. [39]

The **ZhuDehazer** is based on the colour attenuation prior [39]. This method was selected for its unique prior and its demonstrated fast speed. The details of this method are described in 2.1.3

4.2 Implementation

This section details the implementation of the dehazing systems.

Most methods had MATLAB code available from the authors of the original papers, while this was used as a foundation – to incorporate aspects of the algorithm as intended by the original authors but not covered fully in the paper – much of the code has been rewritten entirely. In order to provide a fair assessment of all dehazing methods, all dehazing systems were developed to minimize any differences in implementation by sharing common operations. This is to ensure that the only differentiating factor between the dehazing systems is the explicit algorithm as described in each paper.

4.2.1 Environment

All code was executed on an AMD Ryzen 5 4500U CPU running Windows. No GPU or parallelization was explicitly employed for any methods. The dehazing systems run in a MATLAB R2021a, with the exception of the neural network methods based on Chen et al. (PMHLD) and Li et al. (AOD), as MATLAB does not support certain layers required by these

networks. Furthermore, due to the needs of each network, PMHLD uses a TensorFlow environment while AOD uses a PyTorch environment. To facilitate ease of communication between the MATLAB evaluator and these neural networks, a dehazing MATLAB-Python interface was developed.

4.2.2 Common Interface

All dehazing systems, including the neural network methods, are represented by an abstract MATLAB `BaseDehazer` class object. Each specific dehazing system is a non-abstract subclass of `BaseDehazer`. A dehazing system inheriting from this class must provide a value for the following properties:

- **FrameDelay**: indicates whether this dehazing system has a frame delay and will output a prediction for a previous frame when given a new image.
- **PredictsA** – whether this dehazing system outputs a prediction of the atmospheric light colour.
- **PredictsT** – whether this dehazing system outputs a prediction of the transmission map.

The class features a `dehazeFrame` function that takes a single RGB colour image as input and outputs the following:

- **predImage**: The predicted, dehazed clear image.
- **predT**: The predicted transmission – empty if system does not predict transmission.
- **predA**: The predicted atmospheric light colour – empty if the system does not predict atmospheric light colour.
- **timeImage**: The time taken to predict the clear image, in seconds.
- **timeA**: The time taken to predict the atmospheric light colour, in seconds – empty if the system does not predict atmospheric light colour.

`BaseDehazer` contains a `SequenceState` property, a structure representing state information that persists between calls to `dehazeFrame`. This is used by dehazing systems are explicitly designed for dehazing video. `SequenceState` can be reset by calling `newSequence` and optionally providing known information about the camera and scene such as calibration parameters.

The MATLAB wrapper for neural-network dehazing systems is a subclass of `BaseDehazer` called `ExternalPythonDehazer` that instantiates and maintains a Python process running a `pyinterfacer` script so the neural-network models can be kept in memory

for the lifetime of the `ExternalPythonDehazer` object. The `pyinterfacer.py` script can be easily adapted to run other python-based dehazing systems. `ExternalPythonDehazer` communicates with a `pyinterfacer` instance using the .NET Framework. Due to this requirement, the neural network dehazing systems can only run on Windows.

The dehazing system interface is independent of the evaluation code, as such `BaseDehazer` can be used to develop dehazing systems for use with other evaluators or for other purposes where a standardised format of dehazing systems is required.

4.2.3 Common Operations

Where possible dehazing systems will use the same common operations. All methods that make use of a box filter, unless they require specific border functionality, use an $O(N)$ implementation available in `BaseDehazer`. Similarly, this $O(N)$ box filter is used for the guided filter implementations that are available to all dehazing systems in the `BaseDehazer`. Optimisation is limited, so as to keep an even playing field, however minor optimisations have been made for example when a method called for sorting a ‘top-k’ function is used; as no method ever requires the fully sorted list, only the top few items. The most prominent form of optimisation is one that has been applied to all dehazing systems uniformly, great effort has been spent to ensure that all dehazing systems contain no ‘for-loops’, by vectorising every operation that would require them. This has been done due to the fact that for-loops are notoriously slow in interpreted languages like MATLAB and to provide a consistent set of operations.

4.2.4 Pretrained Neural Network Models

Both the `LiDehazer` (AOD) and `ChenDehazer` (PMHLD) use pretrained models provided by their respective authors. They have not been retrained on the new dataset. This is for two reasons; firstly due to limited resources available for the scope of this project, and secondly to allow all dehazing systems to be evaluated on the full dataset, It is worth pointing out both papers discuss the generalization their trained model’s achieve is sufficient for all dehazing tasks. Future work is encouraged to explore the possibilities of training neural network methods using the new dataset.

4.3 Evaluation Criteria & Results

One of the contributions of this work is the quantification of qualities required to make a performant dehazing ADAS. These are expressed in the form of five criteria, with each being

judged by various quality metrics, as well as a qualitative assessment. Each dehazing system is evaluated on these metrics using the new dataset, henceforth referred to as the evaluation dataset or simply dataset. The five criteria are as follows: Real-time Capability, Visibility Improvement, Naturalness Quality, Video Consistency, Model Conformity.

For full-reference metrics that compare the dehazed image with the true clear image \mathbf{J} , an additional operation is applied on the clear image. In the original images, the sky contains arbitrary colours and textures. Generally, in homogeneous haze the sky will be the colour of the atmospheric light⁴ – this has been replicated in the dataset by modifying the transmission with a sky mask as described in 3.1.3. It is impossible for a dehazing system to recover the unknown sky colour or texture of a true clear image from a given hazy image. The closest representation of the clear image \mathbf{J} a dehazing system is capable of producing is denoted by \mathbf{J}^* . This is an image in which all pixels not in the sky region are equal to \mathbf{J} and all pixels within the sky region are equal to the true atmospheric light

$$\mathbf{J}^*(\mathbf{x}) = \begin{cases} \mathbf{J}(\mathbf{x}), & \mathbf{x} \notin S \\ \mathbf{A}, & \mathbf{x} \in S \end{cases} \quad (15)$$

Where S is the set of all sky region pixels.

Therefore, when evaluating the hazy images against the clear images, rather than use \mathbf{J} as the ground truth image, an estimation of \mathbf{J}^* is used instead, defined as follows:

$$\mathbf{J}^*(\mathbf{x}) = \mathbf{A}m(\mathbf{x}) + \mathbf{J}(\mathbf{x})(1 - m(\mathbf{x})) \quad (16)$$

Where $m(\mathbf{x})$ is an estimated sky mask from 3.1.3 in which values close to 1 indicate high probability of being in the sky region. Note that this operation is equivalent to generating haze as in 3.1.5 with a constant transmission of 1. A constant transmission of 1 according to (1) introduces no haze. This approach also alleviates some evaluative errors caused by artifacts introduced by sky mask inaccuracies.

Below details the evaluation criteria, the metrics that compose them, and the results of the eight existing method dehazing systems. For convenience all quality metrics within this dissertation are written with an arrow before their name, indicating whether it is an error that should be minimised (\downarrow) or a score that should be maximised (\uparrow).

⁴ The only instance where this is not the case for purely homogenous haze is when there are additional sources of light around the sky region. This does not apply to any images featured in the dataset.

4.3.1 Criterion 1: Real-time Capability

An important factor for any ADAS is the speed at which it can perform its intended task. In the case of a dehazing ADAS, this means the time it takes to dehaze a given input frame. An ideal dehazing ADAS will induce minimal delay and closely match the framerate at which footage is being input to avoid the need for skipping frames which may contain crucial information.

Unfortunately, this criterion is often the most difficult to judge. As mentioned in 4.2, in order to provide an equal footing, all dehazing systems have been implemented using the same common operations and level of optimisation. However this may not be entirely reflective of its performance if a dehazing system were implemented in an ADAS. For example, a slower dehazing algorithm may lend itself more easily to optimisation on hardware. While it is beyond the scope of this project to develop and compare efficient hardware implementations, this quality measure can still provide a useful comparison of the speeds at which each algorithm operates. Real-time Capability (RC) is measured with a single metric.

*Table 5: Real-time Capability Quantitative Results.
Showing the \uparrow MPPS results for three different haze densities.*

	\uparrow MPPS		
	Dense	Moderate	Light
Berman	0.1749	0.1748	0.1747
Cai*	4.494	4.502	4.472
Chen	0.1129	0.1134	0.1111
He	3.728	3.748	3.711
Li	3.390	3.380	3.369
Tarel	2.210	2.187	1.993
Tsai	3.708	3.674	3.693
Zhu	<u>4.324</u>	<u>4.316</u>	<u>4.316</u>

Megapixels per Second Throughput (\uparrow MPPS)

This metric measures the speed at which a dehazing system can perform. It is calculated from the megapixels of a sequence, divided by the time each system takes to dehaze a frame. This metric is generally more useful than frames per second (FPS) as it takes into consideration the size of the frames being processed. It should be noted however, that comparing by megapixels is only valid as all compared methods are linear in complexity. The results for this metric are presented in Table 5.

4.3.2 Criterion 2: Visibility Improvement

Paramount to a dehazing system – whether it be in an ADAS or otherwise – is its ability to remove haze from a given image. As much haze as possible needs to be removed from an image presented to the driver to provide the safest possible view. The Visibility Improvement (VI) evaluation criterion is quantified by three metrics.

*Table 6: Visibility Improvement Quantitative Results.
Showing three sub-tables of the \uparrow PSNR, \uparrow MS-SSIM, and \downarrow FADE results for different haze densities*

	\uparrow PSNR			\uparrow MS-SSIM			\downarrow FADE		
	Dense	Moderate	Light	Dense	Moderate	Light	Dense	Moderate	Light
Berman	15.56	16.16	15.00	0.9001	0.9074	0.9157	1.509	1.546	1.284
Cai	17.71	18.35	18.61	0.8955	0.9117	0.9450	2.056	1.937	1.466
Chen	19.19	<u>18.41</u>	<u>22.51</u>	<u>0.9121</u>	0.9022	<u>0.9534</u>	<u>1.416</u>	1.309	<u>1.091</u>
He	<u>18.34</u>	19.77	18.97	0.9222	0.9312	0.9458	1.686	1.721	1.463
Li	16.31	16.83	16.98	0.8153	0.8444	0.8572	1.555	1.385	0.9705
Tarel	11.81	13.65	13.60	0.8148	0.8815	0.8760	1.358	<u>1.343</u>	1.134
Tsai	16.37	17.70	17.57	0.9099	<u>0.9207</u>	0.9440	1.596	1.626	1.376
Zhu	16.33	17.54	23.69	0.8850	0.9123	0.9692	2.579	2.284	1.621

Peak Signal-to-Noise Ratio (\uparrow PSNR)

\uparrow PSNR is a full-reference metric commonly used in image processing and computer vision applications [57, 58]. It is also often utilised in evaluation of dehazing methods by comparing the \uparrow PSNR of a ground truth clear image and a predicted dehazed image. The results for this metric are presented in the leftmost sub-table of Table 6.

Multiscale Structural Similarity Index Measure (\uparrow MS-SSIM) [59]

Similar to \uparrow PSNR, the structural similarity index measure (\uparrow SSIM) is a widely used full-reference metric. Unlike \uparrow PSNR however, it takes into consideration human perception. \uparrow SSIM is commonly used to evaluate dehazing performance [3, 4, 49]. \uparrow MS-SSIM is an improvement on the \uparrow SSIM that evaluates the image at multiple scales. It is calculated on each colour channel independently, then averaged to produce a final \uparrow MS-SSIM value. Though all metrics used for this evaluation are important and equally weighed with respect to their criteria, the \uparrow SSIM (and by extension \uparrow MS-SSIM) is particularly informative and well renowned [60, 61] as such when a single metric is needed for concise comparisons, the \uparrow MS-SSIM will be used. The results for this are presented in the centre sub-table of Table 6.

Fog Aware Density Evaluator (\downarrow FADE) [62]

\downarrow FADE is no-reference haze-oriented image metric designed with dehazing in mind, that aims to quantify the level of perceptual haze within an image. This metric is based on a various image features derived using natural scene statistics (NSS) of hazy and clear images. While \downarrow FADE is useful to quantify haze, a good \downarrow FADE value alone is not indicative of a good visibility image as it can suffer from problems (see **Error! Reference source not found.**). The results for this metric presented in the rightmost sub-table of Table 6.

4.3.3 Criterion 3: Naturalness Quality

While the removal of haze is important to provide a clear visual to a human user, it is equally important the user is provided with a natural, recognizable, distortion-free image that is not too disparate from the visuals they are used to driving with. This requirement forms the basis for the Naturalness Quality (NQ) evaluation criterion and is comprised of two metrics.

Table 7: Naturalness Quality Quantitative Results.
Showing two sub-tables of the \downarrow CIEDE2000 and \downarrow NIQE results for different haze densities

	\downarrow CIEDE2000			\downarrow NIQE		
	Dense	Moderate	Light	Dense	Moderate	Light
Berman	11.56	9.721	10.95	2.584	2.707	2.594
Cai	8.929	7.423	7.684	2.727	2.906	2.660
Chen	7.658	8.020	<u>5.664</u>	3.865	4.185	3.966
He	<u>7.695</u>	6.044	6.802	<u>2.633</u>	<u>2.768</u>	<u>2.651</u>
Li	10.77	9.049	9.689	2.707	2.931	2.769
Tarel	16.55	12.61	12.90	2.792	2.928	2.655
Tsai	10.96	9.065	8.907	2.693	2.914	2.716
Zhu	10.16	<u>7.239</u>	3.875	2.665	2.923	2.612

Mean CIEDE2000 Colour Difference (\downarrow CIEDE2000) [63]

The \downarrow CIEDE2000 is a full-reference image quality metric that quantifies the perceptual colour differences between two images. \downarrow CIEDE2000 has been used to evaluate the colour preservation of dehazing algorithms on new datasets [49]. The results for this metric are presented in the left sub-table of Table 7.

Naturalness Image Quality Evaluator (\downarrow NIQE) [64]

\downarrow NIQE is a no-reference image quality metric that quantifies perceptual distortions in an image. Lower scores indicate a more natural image. \downarrow NIQE decomposes an image into natural scene features based on mean subtracted contrast normalised (MSCN) local patches and uses a multivariate gaussian model – trained on natural images – to predict the naturalness

of an image. \downarrow NIQE is stated to detect any type of perceptually unnatural distortion, however there are some issues with it (and other natural quality evaluators), that is described in section 4.3.6. The results for this metric are presented in the right sub-table of Table 7.

4.3.4 Criterion 4: Video Consistency

While each frame taken individually may score highly on the above metrics, in a dehazing ADAS a human user will be provided with a live video feed that is susceptible to temporal variations introduced through the dehazing process. A sequence of dehazed frames should experience the same changes over time as an equivalent sequence of the clear scene. This is quantified in the Video Consistency (VC) evaluation criterion by two metrics.

*Table 8: Video Consistency Quantitative Results.
Showing two sub-tables of the \uparrow MIC and \downarrow SpEED-QA results for different haze densities*

	\uparrow MIC			\downarrow SpEED-QA		
	Dense	Moderate	Light	Dense	Moderate	Light
Berman	0.4342	0.7097	0.6025	6.210	<u>5.146</u>	5.039
Cai	0.7978	0.8924	0.9050	8.761	6.203	2.662
Chen	0.7189	0.8747	0.9131	<u>4.938</u>	7.521	0.9920
He	0.8060	0.8647	0.8959	3.537	2.979	<u>1.183</u>
Li	0.8573	0.9240	<u>0.9505</u>	12.40	8.283	4.032
Tarel	0.7073	0.8415	0.7971	26.90	11.84	11.27
Tsai	<u>0.8148</u>	<u>0.9016</u>	0.9322	7.461	5.313	2.784
Zhu	0.8071	0.8899	0.9643	16.98	13.22	2.672

Mean Intensity Correlation (\uparrow MIC)

\uparrow MIC is a full-reference metric that measures the correlation between the mean intensity of each frame for the dehazed and clear sequence. Any flickering that occurs in the dehazed sequence will be reflected as abrupt changes in the mean intensity not seen in the mean intensity of the clear sequence. \uparrow MIC is an extension of the metric used in [44], to include chrominance information. The correlation coefficient of mean intensities is calculated separately on each YCbCr colour channel, with the final \uparrow MIC value being a weighted average of these values. Twice as much weight is provided to the luminance channel to match human perception. The correlation in mean intensity should be linear, for this reason Pearson's correlation coefficient is preferred over Spearman's rank correlation coefficient. The results for this metric are presented in the left sub-table of Table 8.

Spatial Efficient Entropic Differencing for Quality Assessment (\downarrow SpEED-QA) [65]

\downarrow SpEED-QA is a reduced-reference video quality metric that quantifies spatio-temporal distortions between a given video and a reference. \downarrow SpEED-QA compares the conditional entropies of local patches between the original and dehazed video using a Gaussian-scale mixture model. The patches are calculated spatially as well as temporally by using frame differences. The results for this metric are presented in the right sub-table of Table 8.

4.3.5 Criterion 5: Model Conformity

The haze model is one that is well understood and can be used to assess a dehazing system based on the accuracy with which it predicts variables of the model, namely the atmospheric light \mathbf{A} and the transmission t . Not all methods predict these values however, opting to directly predict a dehazed image, as such these metrics are only valid for comparisons between methods that do. Model Conformity (MC) is quantified by two error metrics.

Table 9: Model Conformity Quantitative Results.
Showing two sub-tables of the \downarrow TError and \downarrow AError results for different haze densities

	\downarrow TError			\downarrow AError		
	Dense	Moderate	Light	Dense	Moderate	Light
Berman	0.2002	0.2052	0.2761	<u>0.09127</u>	0.05859	<u>0.1472</u>
Cai	0.1935	0.1974	0.2174	0.09920	<u>0.04277</u>	0.1557
Chen	—	—	—	—	—	—
He	0.1483	0.1401	<u>0.1985</u>	0.07590	0.02635	0.1307
Li	—	—	—	—	—	—
Tarel	<u>0.1710</u>	<u>0.1552</u>	0.1905	—	—	—
Tsai	0.1802	0.1938	0.2126	0.1137	0.05719	0.1479
Zhu	0.4604	0.4469	0.2907	0.1325	0.06479	0.1830

Transmission Root Mean Square Error (\downarrow TError)

This metric is the root mean square error of the system’s predicted transmission. The results for this metric are presented in the left sub-table of Table 9.

Atmospheric Root Light Mean Square Error (\downarrow AError)

This metric is the root mean square error of the system’s predicted atmospheric light. The results for this metric are presented in the right sub-table of Table 9.

4.3.6 Qualitative Assessment

It's important to bear in mind that these quantitative metrics – especially the blind metrics – are not always truly representative of perceptual quality. It is important to include qualitative assessments with each of five criteria in mind, alongside the quantitative metrics.

Though this applies to each of the evaluation criteria, a good example to illustrate the shortcomings of qualitative metrics are the problems with the Naturalness Quality criterion. In addition to requiring the \downarrow CIEDE2000 to quantify colour differences, the NQ criterion requires a metric that is able to quantify the distortions introduced by a dehazing process, that result in an unnatural appearing image. For the purposes of detecting naturalness and distortions, there are three commonly use natural image quality evaluators \downarrow NIQE [64] , \downarrow BRISQUE [66], and \downarrow PIQUE [67]. However, all these evaluators fail to identify a commonly occurring dehazing distortion as unnatural. Many dehazing systems, particularly those relying on the DCP, have difficulty in dealing with large objects in the image that are brighter than the atmospheric light. Attempting to dehaze these objects results in large grey blob-like distortions. An example of this is highlighted in Figure 11.



Figure 11: A natural clear image (top) and a failure case of a DCP dehazing system (bottom). Note the large grey blob-like distortion on the left hand side of the image.

Table 10: Quality metrics for the images in Figure 11.

Image	↓FADE	↓PIQUE	↓NIQE	↓BRISQUE
Clear Image (Top of Figure 11)	1.775	38.97	2.588	28.36
Failure Case (Bottom of Figure 11)	1.708	38.58	2.265	18.35

Despite this image appearing visually jarring and unnatural due to the large grey blob on the left-hand side, it achieves better ↓NIQE, ↓BRISQUE, and ↓PIQUE, scores than the original images (Table 10). This is true for similar such distortions – henceforth referred to simply as “blob distortions” – tested on different sequences. This is also not a phenomenon unique to the evaluation dataset either, it occurs with any hazy images that feature large white regions. In each case, when large objects brighter than the atmospheric light are present in the image, many dehazing methods introduce these blob distortions into the dehazed image and the ↓PIQE, ↓NIQE, ↓BRISQUE scores are either better or only insignificantly worse for the distorted dehazed image than the natural original images when they should be much worse. This a clear failure of the evaluators to properly ascribe ‘naturalness’ in the presence of blob distortions. While not a rigorous explanation of why these three evaluation metrics fail in these cases, one can conjecture the following: All three methods extract the same natural scene statistics (NSS) features based on the work of Ruderman [68], known as mean subtracted contrast normalised (MSCN) coefficients. These features are derived from local patches and it’s possible that due to the large spatial frequency of the blob distortions that they have little to no impact on the extracted features. This seems unlikely however, as all three methods employ multiscale strategies to capture a wide gamut of distortions of varying scale. Additionally, ↓BRISQUE uses a supervised model trained on natural and distorted images with an accompanying mean opinion score provided by human testing. If the extracted NSS features used do not inherently ignore these blob distortions, it is highly likely they were not featured in training data and thus cannot be dealt with in the final model predictor. Retraining the ↓BRISQUE model with scored instances of the blob distortion as training data, may alleviate the issue. ↓NIQE on the other hand uses an unsupervised model only trained on natural images, however ↓NIQE specifically only evaluates in patches with a high sharpness.

Despite the failure in these cases, ↓NIQE is still a useful metric as part of the evaluation criteria, as it is still capable of quantifying the distortions it is designed for, predominantly noise, compression artefacts, and (global) blurring.

The failure of metrics in certain cases is not exclusive to the Naturalness Quality criterion. \downarrow FADE is a metric for the Visibility Improvement criterion that can also give counterintuitive scores as a result of these blob distortions. \downarrow FADE fails to recognise the haze present in the centre of the bottom image in Figure 11, and gives it a better score than the true clear image. As with \downarrow NIQE, \downarrow FADE is still useful as it performs suitably in most cases and despite the problems, is currently the best available measure for quantifying haze.

Attempts to find or devise a quality metric capable of identifying and quantifying this distortion have proved to be in vain. However, these blob distortions are just one of possibly many distortions or perceptual inaccuracies unaccounted for in the combined collection of quantitative evaluation metrics. In order to capture a full picture of a dehazing systems performance it is necessary to include a subjective evaluation of the five quality criteria to surmount the shortcomings of the existing metrics. Due to limitations of the project and the length of footage required to review; it is impractical to establish a sufficient experimental setup and sample the opinions of multiple participants. Instead, the discussions below will include qualitative analyses based on the videos produced by each dehazing system, with these videos being made available for further review⁵. Each video features all sequences in descending order of visibility, with one video for each dehazing system plus videos for the clear and hazy sequences. To improve the efficiency of viewing, the videos are played at 20 FPS instead of the 10 FPS of the original sequences.

⁵ The videos for each method are available from <https://essexuniversity.box.com/s/fvkjw944g7prhm01inweaern07pl1llf>

4.4 Discussion

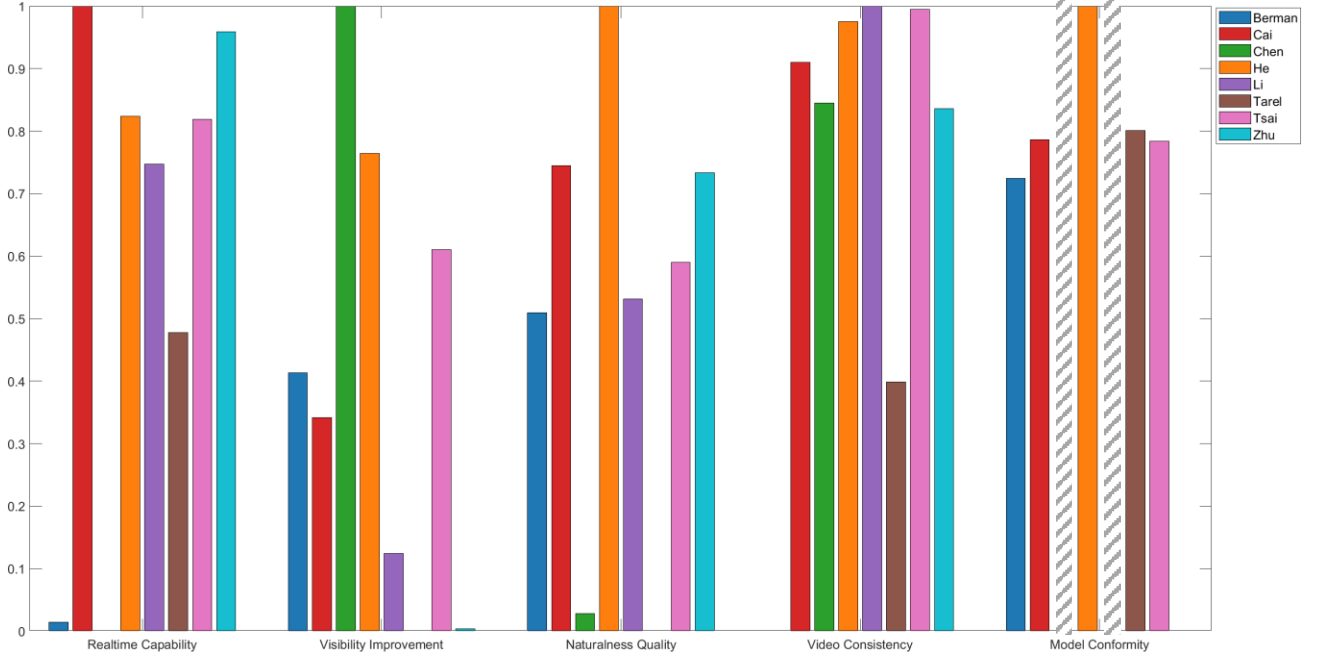


Figure 12: Normalised criteria scores for the eight dehazing systems. Hashed lines indicate unavailable data. Other missing bars indicate the lowest value for that criterion. See appendix for normalisation process.

BermanDehazer

The **BermanDehazer** is considerably slow, achieving less than 1 megapixel per second (Table 5). Unlike other methods the majority of its time is spent on atmospheric light estimation. On average the **BermanDehazer** spent 1.46 seconds on the airlight estimation and 1.19 seconds on the dehazing process. Given that this atmospheric light estimation method does not perform better than other approaches (Table 9), this dehazing system would do better with an alternate method, however it likely still wouldn't improve speeds enough to compete with the other dehazing systems. Though it removes haze moderately well, **BermanDehazer** vastly oversaturates the image as well as introducing unusual colours and significant flashing into the haze that remains. This can be seen in its video consistency and CIEDE200 results (Table 7, Table 8, Figure 12). It is also demonstrated somewhat in Figure 13 (row A). However, this effect is hugely understated in the metrics, the video results show constant flashing of random hues despite having some of best results for \downarrow NIQE (Table 7) and good results for SpEED-QA (Table 8). That being said, with all metrics normalised and averaged together Figure 12 does accurately reflect the qualitative review of the video results.

CaiDehazer

CaiDehazer achieves the best Real-time Capability scores (Table 5), running at roughly 4.5 megapixels a second. Though it's worth bearing in mind that the **CaiDehazer** has an irreducible delay of one frame. Despite being the fastest system, this is still an unsatisfactory result for an ADAS, even when taking into consideration the lack of optimisation all methods share. Despite its video considerations it is only fourth in Video Consistency (Figure 12), performing worse than to two frame-by-frame methods. It does reduce some of the flickering that occurs in the sky regions by the **HeDehazer** and other methods, however the flickering becomes more prominent around high contrast regions.

ChenDehazer

The **ChenDehazer** achieves the best Visibility Improvement however it is incredibly slow (Table 5) at just over 0.1 \uparrow MPPS. While other methods feature flickering and blob distortions in the sky regions and remaining haze, the **ChenDehazer** features general flickering through its video. It also fares the worst on naturalness quality measures, though it must be said that subjectively **ChenDehazer** did not have the least natural video results when compared with those of **BermanDehazer** and **TarelDehazer**. This can be seen in Figure 13 (row C), where the only unnaturalness is slight oversaturation and darkening of regions. Nevertheless, its slow speed makes this method very unsuitable for ADAS dehazing.

HeDehazer

Despite being one of the oldest methods, the **HeDehazer**, performs best on Naturalness Quality and Model Conformity, as well as being within the top three for all criteria (Figure 12). **HeDehazer** achieves the excellent results in both transmission and atmospheric light error (Table 9), beating **TsaiDehazer** and **CaiDehazer** despite them using similar but claimed improvements to atmospheric light estimation. Overall, the **HeDehazer** is the best performing dehazing system. However, as with the other DCP methods, it features blob distortions and flickering in sky regions. The flickering and perceptibility of the blob distortions is greatly lessened in frames where the atmospheric light is predicted accurately. Despite achieving the best atmospheric light errors (Table 9), if **HeDehazer** was able to use the true atmospheric light there would be likely be no perceptible blob distortions and sky region flickering, which is unlikely to be the case for other methods.

LiDehazer

For a neural network, the **LiDehazer** performs adequately in Real-time Capability relative to the other systems (Table 5). Visually the **LiDehazer** does perform better at removing haze than the metrics suggest (Figure 13) however it oversaturates and darkens some sequences which appears to be correlated inversely with the intensity of the true atmospheric light. In other words, it excels at dehazing sequences with near-white coloured haze but performs poorly otherwise. It does however have best Video Consistency (Figure 12) score despite being a frame-by-frame method. It is likely that the relatively poor scores in \downarrow SpEED-QA compared to the excellent scores in \uparrow MIC (Table 8) is due to the spatial component of \downarrow SpEED-QA.

Tare1Dehazer

The **Tare1Dehazer** performs very poorly overall (Figure 12) generally oversaturating the image, darkening the haze, and introducing flickering grey distortions (Figure 13, columns 1 & 2). Despite this it achieves good \downarrow FADE scores for dense and moderate haze, that do not align at all with qualitative assessment and other metrics. This is likely due to its effect of ‘flattening’ haze (Figure 13, column 3), removing its gradient causing it to appear like a solid – but incorrectly located and coloured – sky region. However, it’s worth noting that thanks to the planar assumption, the **Tare1Dehazer** restores light coloured roads better than other methods that tend to introduce too much contrast and saturation. This can be seen in Figure 13 row F, column 3, where despite the rest of the image being poorly dehazed, the road surface up to a certain point is almost identical to the clear image. Due to the planar assumption only imposing a lower bound on the road and having looked through the transmission maps, it is not a contributing factor to the systems poor performance and only prevents the scores from being worse than they currently are.

TsaiDehazer

Despite the video considerations of this method and its good scores in \uparrow MIC, suffers from oscillating between a too bright and too dark image over the course of a few seconds, this is likely due to a combination of its poor atmospheric light estimation method (Table 9) and its gradual updating procedure. As a trade-off however it features far less flickering in sky regions.

ZhuDehazer

The **ZhuDehazer** achieves good Real-time Capability (Table 5) however it fails to remove haze or preserve the naturalness in moderate and dense haze. On light haze however, it achieves the best \uparrow PSNR, \uparrow MS-SSIM (Table 6) and \downarrow CIEDE2000, \downarrow NIQE (Table 7), and

\uparrow MIC scores. Overall, though this still equates to poor general performance (Figure 12). The most likely explanation for its poor performance on moderate and dense haze is due to its fixed assumption of $\beta = 1$ which is unsuitable in these cases despite its recommendation good general value in [39]. Regardless, it appears this parameters needs to be adjusted on a case-by-case basis which is infeasible for an ADAS.



Figure 13: Representative frames. The first column shows a light haze sequence, the second a moderate haze sequence, and the third a dense haze sequence. (J) is the true clear image. (I) is the hazy image presented to the dehazing systems. The remaining rows are the output of the dehazing systems; (A) BermanDehazer, (B) CaiDehazer, (C) ChenDehazer, (D) HeDehazer, (E) LiDehazer, (F) TarelDehazer, (G) TsaiDehazer, (H) ZhuDehazer

[Full size images can be found here](#)

4.5 Conclusions

The results of the evaluation can be summarised in three key points.

- The dark channel prior-based dehazing systems perform well overall, with He's original method incorporating the guided filter performing better in general than the derived methods of Cai and Tsai.
- Current video considerations do not significantly improve Temporal Consistency, dehazing solely frame-by-frame is a perfectly valid approach for video. The flickering of the DCP method without video considerations, that occurs mostly in the sky, is mitigated by having an accurate estimation of atmospheric light.
- The fastest performing dehazing systems are still too slow to be considered for the real-time applications of an ADAS.
- All methods are plagued with different visual errors, no method is able to consistently provide a safe view

5 Proposed Smooth-Extremum Dehazing System

A contribution of this study is a new dehazing method developed from the three key conclusions of chapter 4. In keeping with the naming convention for the evaluated dehazing systems; the proposed approach is implemented as the `TilburyDehazer` dehazing system. The proposed system is based on the DCP process but foregoes the need for refinement through the use of novel smooth-extremum filters. It incorporates the successful aspects of previous methods, such as the atmospheric light estimation of He et al. [17] with an additional parameter, and the lower bound planar assumption of Tarel et al. [9].

5.1 Smooth-Extremum Filters

The smooth-extremum filters are image filters based around the smooth approximations of the maximum and minimum function [69-72]. The smooth maximum function is defined as:

$$S_{\alpha}(\mathbf{V}) = \frac{\sum_{i=1}^N V_i e^{\alpha V_i}}{\sum_{i=1}^N e^{\alpha V_i}} \quad (17)$$

Where \mathbf{V} is a vector of length N and α is a scaling factor.

Though called the smooth ‘maximum’, it is a family of functions that can approximate both the minimum and maximum function depending on the value of α . When $\alpha = 0$ it is equivalent to the mean of \mathbf{V} . It approximates the `min` function as $\alpha \rightarrow -\infty$, and approximates the `max` function as $\alpha \rightarrow \infty$. As such a more fitting name for the general form of S_{α} is the smooth-extremum.

The smooth-extremum is related to the `softargmax`⁶, often used in neural networks to provide a differentiable form of the `argmax` function. The `softargmax` is a vector-valued function, defined for an element i as:

⁶Unintuitively, `softargmax` is more commonly known simply as `softmax`, despite it approximating the `argmax` function and not the `max` function. For the sake of clarity and to avoid confusion with the smooth-maximum, `softmax` is referred to as `softargmax`.

$$\sigma_{\alpha}(\mathbf{V})_i = \frac{e^{\alpha V_i}}{\sum_{j=1}^N e^{\alpha V_j}} \quad (18)$$

Intuitively, softargmax ascribes to each index of \mathbf{V} a probability that the element of \mathbf{V} at that index is the maximum. The softargmax is related to the smooth-extremum in the following way:

$$S_{\alpha}(\mathbf{V}) = \sum_{i=1}^N V_i \sigma_{\alpha}(\mathbf{V})_i \quad (19)$$

Though the softargmax is widely used [73] and has become ubiquitous as a neural network output activation, the smooth-extremum rarely finds applications.

The smooth-extremum can be extended to images by reducing the whole image:

$$S_{\alpha}(\mathbf{I}) = \frac{\sum_{\mathbf{x} \in \mathbf{P}} I(\mathbf{x}) e^{\alpha I(\mathbf{x})}}{\sum_{\mathbf{x} \in \mathbf{P}} e^{\alpha I(\mathbf{x})}} \quad (20)$$

Where \mathbf{P} is the set of all pixel indices in the image \mathbf{I} .

A far more useful and novel application however is applying the smooth-extremum as a windowed function, applied to each patch of the image where $\Omega(\mathbf{x})$ is a local patch centred around \mathbf{x} .

$$M_{\alpha}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \Omega(\mathbf{x})} I(\mathbf{y}) e^{\alpha I(\mathbf{y})}}{\sum_{\mathbf{y} \in \Omega(\mathbf{x})} e^{\alpha I(\mathbf{y})}} \quad (21)$$

This describes a set of image filters parameterised by α that is equivalent to a mean filter (box blur) when $\alpha = 0$, approximates an erosion filter (patch-minimum) as $\alpha \rightarrow -\infty$, and approximates a dilation filter (patch-maximum) as $\alpha \rightarrow \infty$. If α is treated as a parameter, this is the Smooth-Extremum Filter (SEF) and can be computed very efficiently using sum filters (Algorithm 1). This filter provides some inherent benefits when used in place of an erosion or dilation with a suitable α , namely it is far more resilient to locally peaking noise (Figure 14).

Algorithm 1: Smooth-Extremum Filter (SEF)

Input: Greyscale Image – $I \in [0, 1]$
 Peak Factor – $\alpha \in [-\infty, \infty]$ // Though values with a very large magnitude are likely to cause overflow or underflow errors
 Filter Radius – $r \in \mathbb{Z} \cap [3, \infty]$ // Similarly, very large values can cause problems

Begin
 $E \leftarrow \exp(I \cdot \alpha);$
 $D \leftarrow \text{SumFilter}(E, r);$
 $M \leftarrow \text{SumFilter}(E \cdot I, r) \div D;$
return $M;$
End

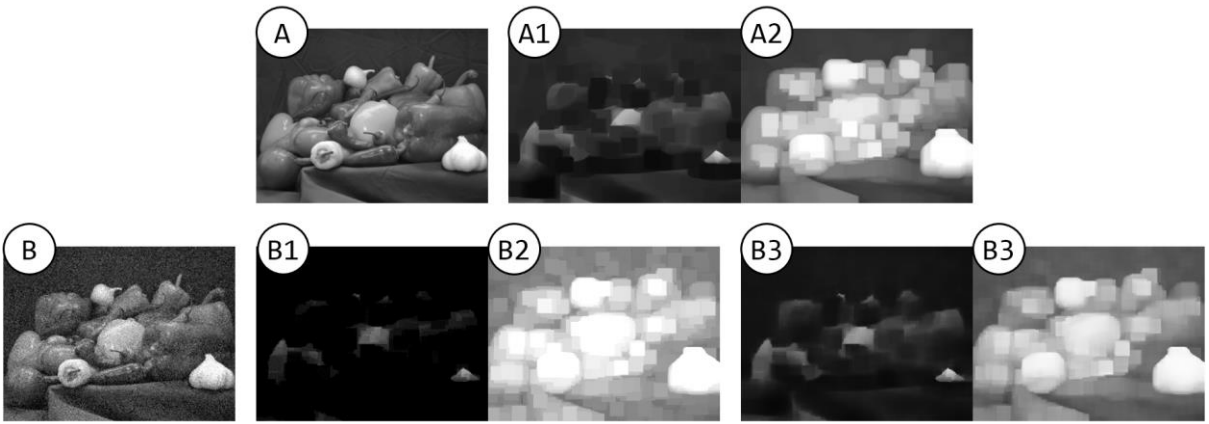


Figure 14: A comparison of the erosion and dilation operations with the smooth-extremum filter for radius 15 (patch size 31). (A) shows the original image, (A1) and (A2) show the result of applying erosion and dilation operations to (A). (B) shows the original image with gaussian noise added. (B1) and (B2) show the result of applying erosion and dilation operation to (B). (B3) and (B4) show the result of applying the SEF filter to (B) with an α of -10 and 10 respectively. Both appear closer to the erosion and dilation on the noise-free image than the true erosion and dilation operations applied on the noisy image.

Additionally, smoothing map can be used in conjunction with α as a scaling parameter. Let \mathbf{Q} be a map of the same dimensions as \mathbf{I} with values in the range $[-1, 1]$.

$$M_{\alpha}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \Omega(\mathbf{x})} I(\mathbf{y}) e^{\alpha Q(\mathbf{x}) I(\mathbf{y})}}{\sum_{\mathbf{y} \in \Omega(\mathbf{x})} e^{\alpha Q(\mathbf{x}) I(\mathbf{y})}} \quad (22)$$

This formulation provides local control over how the filter behaves, allowing an image to have smooth transitions between eroded, blurred, and dilated regions. This filter is dubbed the Smooth-Extremum Local Filter (SELF) and to the best of my knowledge and research, this is the first description of the smooth-extremum being applied to patches as image filter, and having a locally varying α . SEF is a special case of SELF with a constant \mathbf{Q} .

To avoid having to calculate a separate exponential denominator for each individual patch, which requires an additional loop, an approximation can be used:

$$M_{\alpha}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \Omega(\mathbf{x})} I(\mathbf{y}) e^{\alpha Q(\mathbf{y}) I(\mathbf{y})}}{\sum_{\mathbf{y} \in \Omega(\mathbf{x})} e^{\alpha Q(\mathbf{y}) I(\mathbf{y})}} \quad (23)$$

This function is appropriately dubbed the Approximate Smooth-Extremum Local Filter (ASELF). As with the SELF, ASELF is a generalisation of SEF. The behaviour of the ASELF for a given pixel is not independently controlled by the \mathbf{Q} value for that pixel, it is instead influenced by the values of the whole patch. Importantly, this precludes the use of combining positive and negative values in \mathbf{Q} , as rather than producing locally separated erosion and dilation, the values will cancel out resulting in purely blur-like behaviour. However, the advantage is that unlike the SELF, ASELF can be very efficiently calculated (Algorithm 2).

Algorithm 2: Approximate Smooth-Extremum Local Filter (ASELF)

Input: Greyscale Image – $I \in [0, 1]$
Smoothing Map – $Q \in [0, 1]$
Peak Factor – $\alpha \in [-\infty, \infty]$ // Though values with a very large magnitude are likely to cause overflow or underflow errors
Filter Radius – $r \in \mathbb{Z} \cap [3, \infty]$ // Similarly, very large values can cause problems

Begin

$E \leftarrow \exp(I \cdot Q \cdot \alpha);$
 $D \leftarrow \text{SumFilter}(E, r);$
 $M \leftarrow \text{SumFilter}(E \cdot I, r) \div D;$
return $M;$

End

The SEF, and ASELF are proposed as solutions to remove the need for a computationally costly refinement step as part of the DCP dehazing process. This refinement step is generally needed to remove the obtrusive blocky edges caused by the erosion filter, however using the SEF allows the necessary information regarding local minima to be extracted while maintaining a smooth transmission map.

Though the SELF is not used in the proposed dehazing system due to a lack of efficient implementation, it arguably has the most potential for future applications and analysis.

5.2 Smooth-Extremum-Based Dehazing

The `TilburyDehazer` incorporates the smooth-extremum filter to estimate transmission without the need for refinement. The full process of transmission estimation is shown in Figure 15. `TilburyDehazer` is implemented and tested with three different methods for calculating a smooth-extremum dark channel (the blue step in Figure 15). These methods are named ‘Local’, ‘Global’ and ‘Opening’.

Due to the success of the He et al.’s atmospheric light estimation method, achieving the lowest atmospheric light error for the whole dataset – this method is incorporated into the `TilburyDehazer`. However, in an attempt to achieve a more accurate atmospheric light colour to reduce the sky issues of the `HeDehazer`, rather than using a single patch size / filter radius for transmission estimation and atmospheric light estimation, a separate parameter r_A is included that controls the patch size of the dark channel used for atmospheric light estimation.

Though the `Tare1Dehazer` performed poorly in general, its planar assumption bound prevents lightly coloured roads from becoming oversaturated. Due to this, it is incorporated into the proposed `TilburyDehazer`.

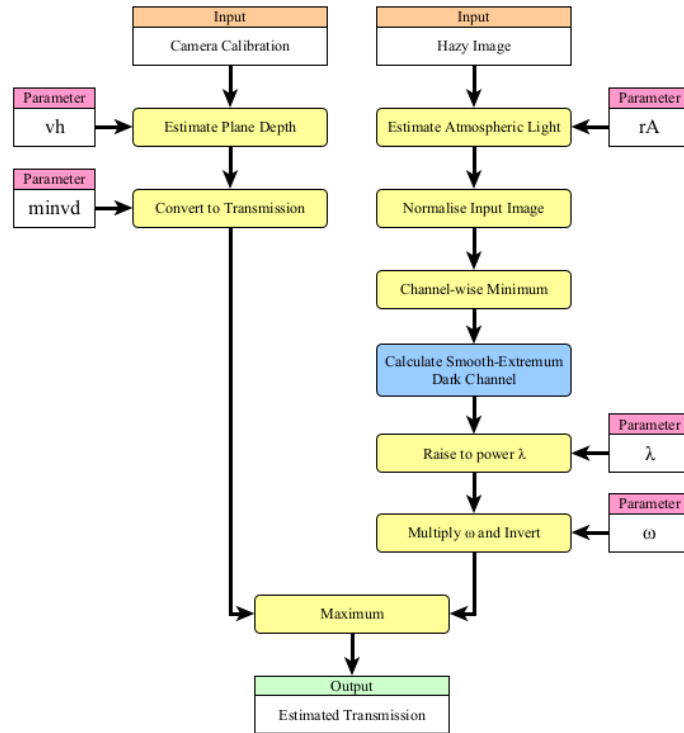


Figure 15: A diagram showing the process of the proposed approach. See Figure 16, Figure 17, and Figure 18 for diagrams of the three implementations for the blue highlighted step.

The ‘Local’ type **TilburyDehazer** uses the ASELF filter with the morphological gradient as the smoothing map. This choice of map is based on the problems with the coarse transmission map (Figure 6). The morphological gradient (dilation minus erosion) is large in the regions where the coarse transmission would produce blocky edges, the assumption is that if the ASELF acts more like a mean filter in these regions it will reduce the blocky edges. The process is shown in Figure 16.

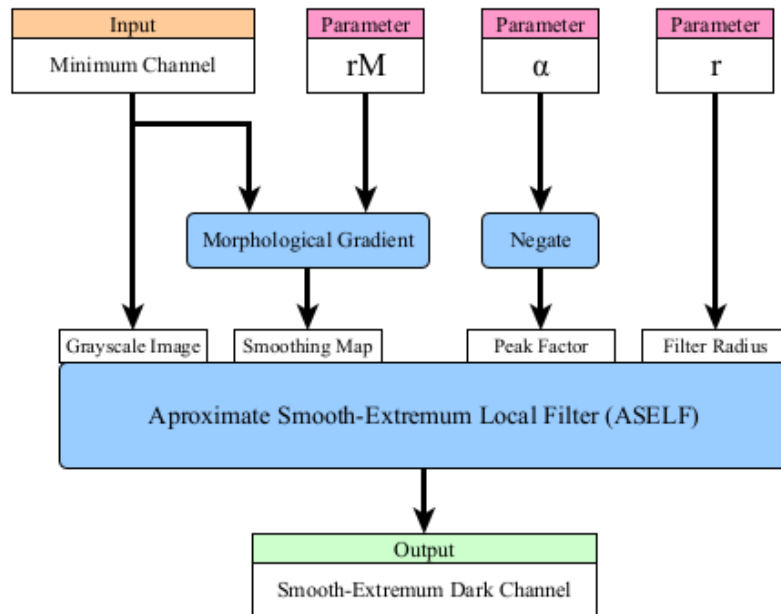


Figure 16: ‘Local’ method of smooth-extremum dark channel calculation.

The ‘Global’ type **TilburyDehazer** uses a basic form SEF with a negative α . The process is shown in Figure 17.

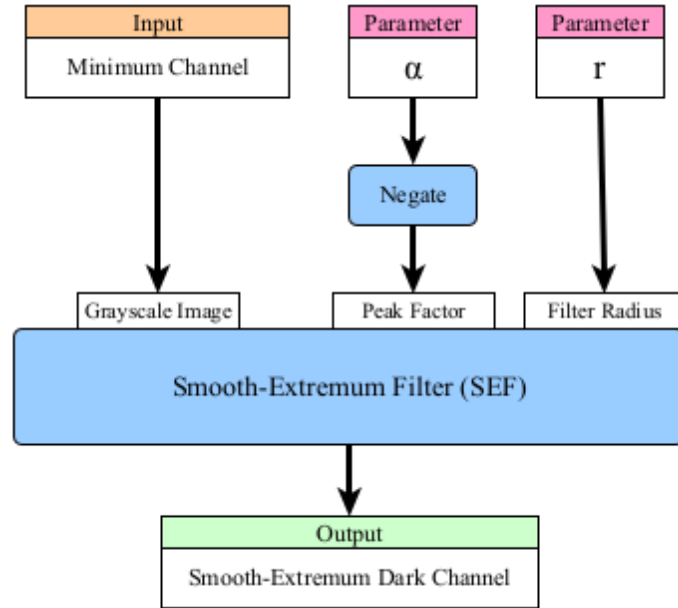


Figure 17: 'Global' method of smooth-extremum dark channel calculation.

'Opening' uses the SEF with a negative α , followed by the SEF with a positive α . In the same way that SEF with a negative α is a smooth approximation of the erosion filter, and SEF with a positive α is a smooth approximation of the dilation filter. Applying a negative α SEF followed by a positive α SEF is a smooth approximation of the morphological opening operation. The process is shown in Figure 18.

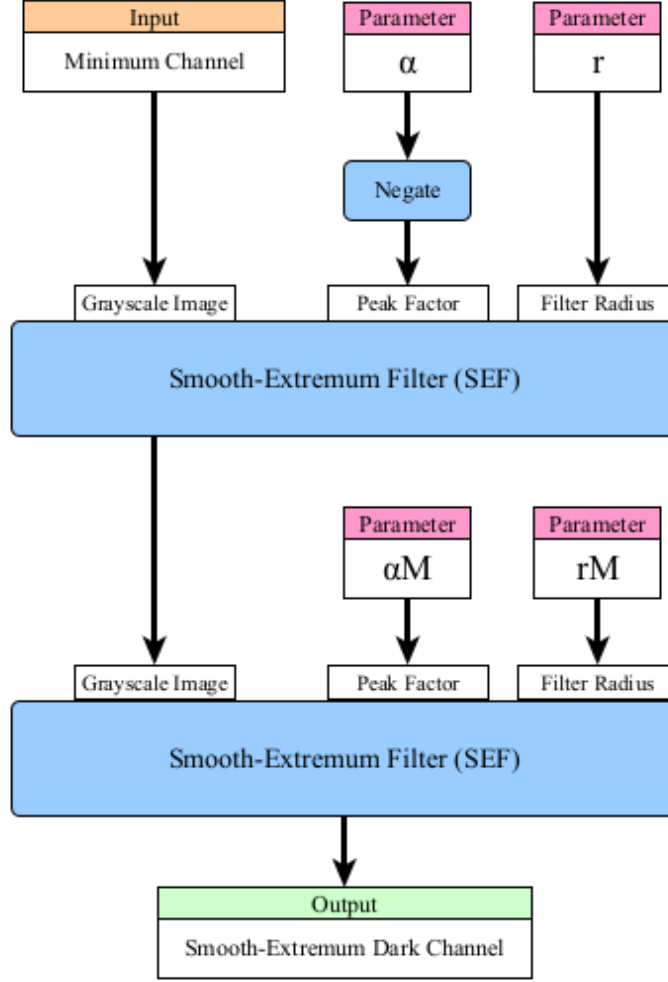


Figure 18: ‘Opening’ method of smooth-extremum dark channel calculation.

5.3 Parameter Optimisation

This section details the efforts to optimise the parameters of the three `TilburyDehazer` methods. ‘Optimisation’ as discussed here refers only to finding the best parameter values, `TilburyDehazer` is implemented in MATLAB with the same degree of software and hardware optimisation shared by the other dehazing systems, as described in 4.2.

The initial parameter values for the `TilburyDehazer` are based on the values used for the `HeDehazer`. Parameters pertaining to the planar assumption are kept as they are in `Tare1Dehazer`. Each method was ran with using Bayesian optimisation for 100 iterations. The initial and final values for the parameters are presented in Table 11, alongside descriptions of each parameter. The optimisation process is further described in the following sections.

Table 11: Initial and final optimised parameter values for the three methods

Parameter	Variable Name	Initial Value	Optimised Values			Description
			Local	Global	Opening	
α	alpha	20	10.20	7.212	23.86	Peak factor for the SEF or ASELF. A higher value results in the filters acting more like an erosion/dilation filter.
γ	gamma	1.25	1.285	1.281	1.277	Transmission gamma correction factor.
ω	omega	0.95	0.8511	0.9099	0.9576	Proportion of haze to remove from the image, leaving some in for visual cues
t_0	t0	0.1	0.00005415	0.1601	0.1832	Lower bound for final transmission map
r	r	15	42	40	37	Radius of the SEF or ASELF filter
r_A	rA	30	39	43	46	Radius used in calculating the atmospheric light.
r_M	rM	15	40	-	41	Using local method: Radius of morphological gradient Using opening method: Radius of smooth maximum
α_M	alphaM	20	-	-	5.593	Using opening method: Peak factor for the smooth-dilation.
Objective Value		0.06574	0.05312	0.05330	0.04917	

5.3.1 Optimisation Dataset

The haze dataset described in chapter 3 is used for the evaluation of all dehazing systems. This data cannot be used to optimise the parameters of **TilburyDehazer** or any other dehazing system as it will introduce bias into its respective evaluation results. To overcome this, a second smaller dataset has been developed referred to for the rest of this paper as the optimisation dataset. Following the same procedure as in chapter 3, the optimisation dataset uses 18 different KITTI sequences from that of the evaluation dataset. Only 20 frames from each sequence are included and used for the optimisation, totalling 360 frames. The properties of each sequence are described in Table 12.

Table 12: Overview of the optimisation dataset.

Sequence	KITTI Category	A Intensity	Visibility (m)	Haze Category
2011_09_26_drive_0002	City	0.8687	571	Moderate
2011_09_26_drive_0011	City	0.8039	929	Light
2011_09_26_drive_0019	Residential	0.7703	167	Dense
2011_09_26_drive_0020	Residential	0.9820	574	Moderate
2011_09_26_drive_0023	Residential	0.9334	147	Dense
2011_09_26_drive_0027	Road	0.8242	890	Light
2011_09_26_drive_0029	Road	0.9019	625	Moderate
2011_09_26_drive_0032	Road	0.7744	641	Moderate
2011_09_26_drive_0039	Residential	0.9700	228	Dense
2011_09_26_drive_0046	Residential	0.8287	678	Moderate
2011_09_26_drive_0048	City	0.7585	473	Moderate
2011_09_26_drive_0052	Road	0.8281	806	Light
2011_09_26_drive_0079	Residential	0.7579	552	Moderate
2011_09_26_drive_0095	City	0.8879	920	Light
2011_09_26_drive_0113	City	0.8541	378	Dense
2011_09_29_drive_0004	Road	0.8005	204	Dense
2011_09_29_drive_0026	City	0.7761	949	Light
2011_09_30_drive_0016	Road	0.7143	153	Dense

5.3.2 Bayesian Optimisation

The specific optimisation algorithm used is Bayesian optimisation [74, 75]. Bayesian optimisation is a global optimisation method designed for black-box objective functions. A major advantage of Bayesian optimisation over other methods is that it can optimise while minimising the number of times the objective function is executed. Making it ideal for costly to evaluate objective functions, such as running a dehazing system on the whole optimisation dataset. The specific implementation used – included within MATLAB’s Statistics and Machine Learning Toolbox – also has the benefit of accepting mixed-type input. Meaning it can optimise both real and integer parameters. This is required due to the filter widths and radii (r , r_A , and r_M) being integers.

It’s worth reiterating that this optimisation as it is being discussed here solely refers to finding the best parameters of the method. **TilburyDehazer** is implemented with the same level of software and hardware optimisation shared by all other dehazing systems as described in 4.2

5.3.3 Objective Function

In general, optimisation attempts to minimise an objective function. The function must be designed to encompass the goals of the optimisation into quantifiable values. The objective function chosen to optimise the parameters of a dehazing system involves evaluating the dehazing system on the entirety of the optimisation dataset. Inside the objective function, multiple calls to a separate loss function are used to calculate the error of a dehazing system for a given a frame or sequence. The full objective function is the mean of the loss function applied to every frame and sequence in the optimisation dataset.

There are plenty of metrics comprising the evaluation criteria described in chapter 4 that could be considered for use as the loss function. However, this in itself poses a problem, as no single metric in the evaluation criteria has sole importance for the performance of a dehazing system. Bayesian optimisation is not multi-objective, meaning it cannot minimise multiple objective functions simultaneously. In order to consider all the evaluation criteria, they would need to be combined into a single metric. This process is known as scalarizing [76], however there exists problems due to the very different derivatives these functions have with respect to each other and the parameters to be optimised, even the normalised scores only have relevant scaling in the context of the eight existing systems evaluated.

Unfortunately, there is no rigorous solution or elegant choice in this case. In the qualitative assessment of dehazing systems, the \uparrow MS-SSIM correlated most with subjective opinion. Therefore, the loss function used in the objective function is $\downarrow(1-\text{MS-SSIM})$.

Table 13: \uparrow MS-SSIM results for the proposed dehazing systems, evaluated on the full evaluation dataset and on the optimisation dataset

	\uparrow MS-SSIM (Evaluation Dataset)	\uparrow MS-SSIM (Optimisation Dataset)
TilburyLocal	0.9500	0.9469
TilburyGlobal	0.9459	0.9467
TilburyOpening	0.9540	0.9508

After finding the best parameters the methods were evaluated in the same manner as the existing methods on the evaluation dataset. Though these results are presented and discussed in much more detail in the next section, it's worth highlighting here the \uparrow MS-SSIM scores as this was used in the loss function. It can be seen that the TilburyDehazer shows very little difference on the optimisation dataset and the evaluation dataset (Table 13). This

indicates that no overfitting occurred in the learning of parameters – despite the optimisation dataset’s small size compared to the evaluation dataset.

5.4 Results & Discussion

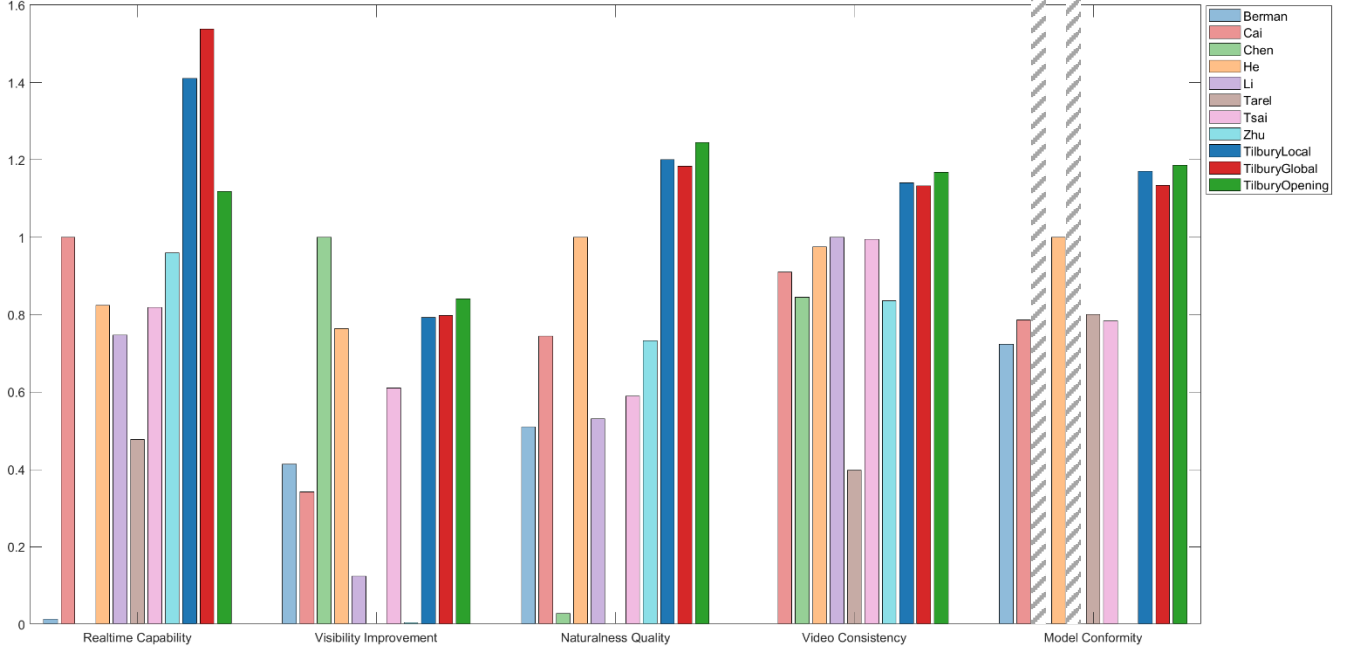


Figure 19: Normalised criteria scores for the eight original dehazing systems, and the three proposed systems. Values are scaled using the same normalisation scaling in Figure 12. Therefore, values greater than 1 indicate better performance in that criteria than the previous best scoring dehazing system. Hashed lines indicate unavailable data. Other missing bars indicate the lowest value for that criterion. See appendix for normalisation process.

The proposed **TilburyDehazer** methods are evaluated in the haze video dataset in the same manner as the other methods, as described in 4.3. All three methods outperform the existing dehazing systems in the evaluation criteria (Figure 19), being beaten only the **ChenDehazer** in Visibility improvement. Notably the **TilburyDehazer** methods are much faster (Figure 20, Table 14, Table 15, and Table 16) than the existing methods, with Local and Global achieving more than 6 ↑MPPS on average. There is very little variation in performance between the three methods in terms of qualitative metrics however the distinction is made clearer in the video results, where the opening method is much more clearly superior in avoiding ‘glowing edges’ (this can be seen slightly in Figure 21). From a model perspective, these methods achieve the best results in atmospheric light and transmission estimation (Table 14, Table 15, and Table 16) with the sole exception of the Global method for moderate haze. Furthermore, the planar assumption manages to preserve the road in many cases with very little detriment to performance. There does not appear to be any significant difference in performance between different densities of haze, beyond the trends experienced by all dehazing

systems. The only thing of note in regard to the haze categories is that the methods fare far better on dense and light haze for atmospheric light estimation (Table 14, Table 15), where the errors are less than half that of the next best method, but for moderate haze there is not quite such a gap (Table 16) .

Despite the excellent scores, the methods still suffer from haloed edges and flickering. Though the blob distortions and flickering in sky regions is far less than that of **HeDehazer**, flickering around edges is more common. The dehazing system fares worst on \downarrow FADE and \downarrow NIQE, this is reflected also in the qualitative video results (Figure 21). I would say, despite the scores suggesting otherwise, that the **TilburyDehazer** methods do not appear visually superior to all other methods, nor are they an acceptably safe visual representation of the road. **ChenDehazer** in general appears much more realistic and haze free, with other methods occasionally appearing better for certain sequences. However, the results generally appear better than those of the **HeDehazer** upon which it is based, and it does so with the major advantage of faster processing.

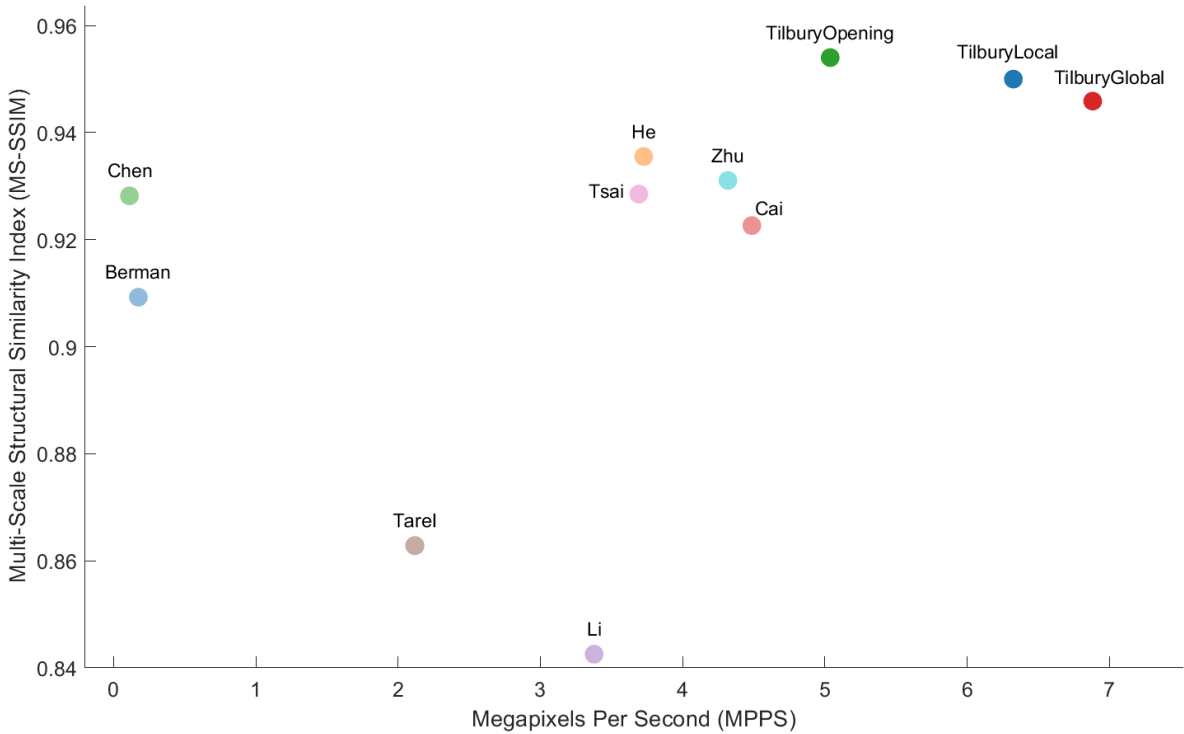


Figure 20: Comparison of \uparrow MS-SSIM scores against \uparrow MPPS

The results for the **TilburyDehazer** methods in the tables below also include the best, and second best results of the eight existing dehazing systems. **TilburyDehazer** results highlighted in green are better than the best result of the existing methods, and results highlighted in yellow are better than the second best result of the existing methods.

Table 14: Results for the three proposed methods on dense haze.

	Dense Haze									
	↑MPPS	↑PSNR	↑MS-SSIM	↓FADE	↓CIEDE2000	↓NIQE	↑MIC	↓SpEED-QA	↓TError	↓AError
TilburyLocal	6.297	20.70	0.9311	2.012	5.884	2.594	0.8543	3.540	0.1267	0.02608
TilburyGlobal	6.850	20.53	0.9267	1.923	5.879	2.605	0.8578	3.633	0.1337	0.02591
TilburyOpening	5.009	21.06	0.9357	2.038	5.449	2.609	0.8758	3.701	0.1207	0.02560
Previous Best	4.494 Cai	19.19 Chen	0.9222 He	1.358 Tarel	7.658 Chen	2.584 Berman	0.8573 Li	3.537 He	0.1483 He	0.07590 He
Previous 2 nd Best	4.324 Zhu	18.34 He	0.9121 Chen	1.416 Chen	7.675 He	2.633 He	0.8148 Tsai	4.938 Chen	0.1710 Tarel	0.09127 Berman

Table 15: Results for the three proposed methods on moderate haze.

	Moderate Haze									
	↑MPPS	↑PSNR	↑MS-SSIM	↓FADE	↓CIEDE2000	↓NIQE	↑MIC	↓SpEED-QA	↓TError	↓AError
TilburyLocal	6.364	20.93	0.9339	2.100	5.188	2.774	0.9208	3.662	0.1354	0.01832
TilburyGlobal	6.916	20.50	0.9283	1.993	5.435	2.790	0.9140	4.493	0.1553	0.02022
TilburyOpening	5.053	21.28	0.9390	2.051	4.851	2.803	0.9204	3.827	0.1230	0.02137
Previous Best	4.502 Cai	19.77 He	0.9312 He	1.309 Chen	6.044 He	2.707 Berman	0.9240 Li	2.979 He	0.1401 He	0.02635 He
Previous 2 nd Best	4.316 Zhu	18.41 Chen	0.9207 Tsai	1.343 Tarel	7.239 Zhu	2.768 He	0.9016 Tsai	5.146 Berman	0.1552 Tarel	0.04277 Cai

Table 16: Results for the three proposed methods on light haze

	Light Haze									
	↑MPPS	↑PSNR	↑MS-SSIM	↓FADE	↓CIEDE2000	↓NIQE	↑MIC	↓SpEED-QA	↓TError	↓AError
TilburyLocal	6.315	24.81	0.9707	1.714	3.509	2.583	0.9511	0.7764	0.1414	0.06237
TilburyGlobal	6.879	24.34	0.9677	1.639	3.740	2.608	0.9501	0.9280	0.1519	0.06234
TilburyOpening	5.044	25.39	0.9738	1.755	3.230	2.596	0.9526	0.7687	0.1441	0.06291
Previous Best	4.472 Cai	23.69 Zhu	0.9692 Zhu	0.9705 Li	3.875 Zhu	2.573 Tarel	0.9643 Zhu	0.9920 He	0.1905 Tarel	0.1307 He
Previous 2 nd Best	4.316 Zhu	22.51 Chen	0.9534 Chen	1.091 He	5.664 Chen	2.612 Zhu	0.9505 Li	1.183 Berman	0.1985 He	0.1472 Berman



Figure 21: Representative frames. The first column shows a light haze sequence, the second a moderate haze sequence, and the third a dense haze sequence. Rows (1), (2), and (3) are the outputs for the *TilburyDehazer* using ‘Local’, ‘Global’, and ‘Opening’ methods respectively. (J) is the true clear image. (I) is the hazy image presented to the dehazing systems. The remaining rows are the output of the dehazing systems; (A) *BermanDehazer*, (B) *CaiDehazer*, (C) *ChenDehazer*, (D) *HeDehazer*, (E) *LiDehazer*, (F) *TareLDehazer*, (G) *TsaiDehazer*, (H) *ZhuDehazer*
[Fullsize images can be found here](#)

6 Conclusion

This study has laid the groundwork for the future development of dehazing ADAS by analysing the appropriateness of different dehazing algorithms and techniques under the situations and criteria that would be expected of such a system. It has introduced a novel haze video dataset and evaluation framework, filling a gap in the current research of ADAS dehazing. A new dehazing system has been developed that improves upon its predecessors; the introduction of the smooth-extremum filter to the dark channel prior dehazing process has eliminated the need for a refinement step, and in doing so greatly improved processing speed. All three versions of the proposed **TilburyDehazer** outperform the techniques upon which they are based in all five criteria. The methods are only let down by their \downarrow FADE and \downarrow NIQE scores, however the results appear visually better than most existing methods and are able to be achieved at a much faster rate than the other methods. However, all methods, including the **TilburyDehazer**, still have many visual errors that need to be addressed before implementation in an ADAS. Though **TilburyDehazer** makes great strides in improving efficiency, faster methods are still needed to keep up with the real-time needs of an ADAS.

6.1 Future Work

Though much has been accomplished throughout this study, many avenues have been explored that present further opportunities for experimentation and development. A few of these are detailed below.

While the evaluation criteria aims to be a complete framework for the assessment of dehazing ADAS, the metrics presented feature many flaws. Better quality metrics are required that to account for the distortions mentioned in 4.3.6. Potential options include; comparing gradient magnitudes to \mathbf{J} ; quantifying changes in gradient directions that deviate from the acceptable rotation of the gradient directions between \mathbf{I} and \mathbf{J} ; and directly computing errors in frequency space. Furthermore, although CIEDE2000 generally seems to work well, it also fails correctly quantify some cases of oversaturation. A potential method that was unable to be included into the evaluator due to time constraints is a “colour blowout” measure that is calculated as follows:

$$\downarrow CB(\tilde{\mathbf{I}}, \mathbf{J}^*) = \sqrt{\sum_{\mathbf{x} \in \mathbf{P}} \left(I_V(\mathbf{x}) \max(I_S(\mathbf{x}) - J_S^*(\mathbf{x}), 0) \right)^2}$$

Where I_S and I_V are the saturation and value components of the estimated dehazed image $\tilde{\mathbf{I}}$

The smooth-extremum filters present the most interesting opportunities for future development and applications. Only one type of smoothing map was used, and it is possible there are better methods for generating a smoothing map. The smooth-extremum filters could have other applications where morphological operations are needed. A useful modification would be to modify the filter to behave as the identity function rather than a mean filter at $\alpha = 0$. The smooth-extremum filter could also be applied in the calculation of the atmospheric light. Furthermore, although it was not used in any dehazing system due to a lack of efficient algorithm, the SELF filter has potential to be the most useful by allowing per-pixel control. Future work will look into finding an efficient algorithm for this filter.

As mentioned in 3.2, the haze dataset – while novel – is plagued by the lighting conditions of the original KITTI sequences. Based on the merits and downsides of the proposed haze dataset; it is highly recommended that future haze-synthetic video datasets use the procedure involving sky masks in the synthesis and evaluation process, and that they use non-hazy but overcast sequences as a base.

Bayesian optimisation can handle stochastic objective functions, so a better approach in the optimisation of dehazing system parameters would be for the objective function to evaluate the loss on a random sample from a much larger optimisation dataset. The objective function could also generate the hazy images itself using randomly chosen \mathbf{A} and β providing a much more general set of conditions to avoid overfitting and getting caught in local minima.

Finally, based on the success of the *TilburyDehazer*, work can begin on a hardware efficient field programmable gate array (FPGA) implementation as the next step to fully realising a dehazing ADAS. The boost in speed will allow for additional processing to recover a visible and fully safe view of the road.

References

- [1] Y. Cheng, W. Niu, and Z. Zhai, "Video dehazing for surveillance unmanned aerial vehicle," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1-5: IEEE.
- [2] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 41-50, 2014.
- [3] D. Singh and V. Kumar, "A Comprehensive Review of Computational Dehazing Techniques," (in En), *Archives of Computational Methods in Engineering*, OriginalPaper vol. 26, no. 5, pp. 1395-1413, 2018-09-18 2018.

- [4] Y. Xu, J. Wen, L. Fei, and Z. Zhang, "Review of video and image defogging algorithms and related studies on image restoration and enhancement," *Ieee Access*, vol. 4, pp. 165-188, 2015.
- [5] G. Ma, M. Dwivedi, R. Li, C. Sun, and A. Kummert, "A real-time rear view camera based obstacle detection," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1-6: IEEE.
- [6] S. Schleicher and C. Gelau, "The influence of Cruise Control and Adaptive Cruise Control on driving behaviour—A driving simulator study," *Accident Analysis & Prevention*, vol. 43, no. 3, pp. 1134-1139, 2011.
- [7] R. Ono, W. Ike, and Y. Fukaya, "Pre-collision system for toyota safety sense," SAE Technical Paper0148-7191, 2016.
- [8] N. Hautière, J.-P. Tarel, and D. Aubert, "Towards fog-free in-vehicle vision systems through contrast restoration," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8: IEEE.
- [9] N. Hautière, J.-P. Tarel, J. Lavenant, and D. Aubert, "Automatic fog detection and estimation of visibility distance through use of an onboard camera," *Machine Vision and Applications*, vol. 17, no. 1, pp. 8-20, 2006.
- [10] J.-P. Tarel and N. Hautiere, "Fast visibility restoration from a single color or gray level image," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 2201-2208: IEEE, 1369.
- [11] J.-P. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer, "Vision enhancement in homogeneous and heterogeneous fog," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 2, pp. 6-20, 2012.
- [12] J.-P. Tarel, N. Hautiere, A. Cord, D. Gruyer, and H. Halmaoui, "Improved visibility of road scene images under heterogeneous fog," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 478-485: IEEE.
- [13] C.-C. Tsai, C.-Y. Lin, and J.-I. Guo, "Dark channel prior based video dehazing algorithm with sky preservation and its embedded system realization for ADAS applications," *Optics express*, vol. 27, no. 9, pp. 11877-11901, Apr 29 2019.
- [14] H. Koschmieder, "Theorie der horizontalen Sichtweite," *Beitrage zur Physik der freien Atmosphere*, pp. 33-53, 1924.
- [15] J. McCartney, *Optics of the Atmosphere: Scattering by Molecules and Particles*. Wiley, 1976.
- [16] R. T. Tan, "Visibility in bad weather from a single image," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8: IEEE, 2141.
- [17] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341-2353, 2010.
- [18] R. Fattal, "Single image dehazing," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, pp. 1-9, 2008.
- [19] S. K. Nayar and S. G. Narasimhan, "Vision in bad weather," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 820-827: IEEE.
- [20] Y. Pei, Y. Huang, Q. Zou, Y. Lu, and S. Wang, "Does haze removal help cnn-based image classification?," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 682-697.
- [21] D. Guo, Y. Pei, K. Zheng, H. Yu, Y. Lu, and S. Wang, "Degraded image semantic segmentation with dense-gram networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 782-795, 2019.

- [22] H. S. M. Lim and A. Taeihagh, "Algorithmic decision-making in AVs: Understanding ethical and technical concerns for smart cities," *Sustainability*, vol. 11, no. 20, p. 5791, 2019.
- [23] J. Li, X. Zhao, M.-J. Cho, W. Ju, and B. F. Malle, "From trolley to autonomous vehicle: Perceptions of responsibility and moral norms in traffic accidents with self-driving cars," *SAE Technical paper*, vol. 10, pp. 2016-01, 2016.
- [24] J. S. Brodsky, "Autonomous vehicle regulation: How an uncertain legal landscape may hit the brakes on self-driving cars," *Berkeley Technology Law Journal*, vol. 31, no. 2, pp. 851-878, 2016.
- [25] *Standard No. 111; Rear visibility*, C. o. F. Regulations 111, 2018.
- [26] *Rear Visibility Systems*, C. R. o. Canada 111, 2021.
- [27] J.-Y. Na and K.-J. Yoon, "Stereo Vision aided Image Dehazing using Deep Neural Network," in *Proceedings of the 1st Workshop and Challenge on Comprehensive Video Understanding in the Wild*, 2018, pp. 15-19.
- [28] S. G. Narasimhan and S. K. Nayar, "Chromatic framework for vision in bad weather," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, 2000, vol. 1, pp. 598-605: IEEE.
- [29] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 713-724, 2003.
- [30] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. I-I: IEEE, 905.
- [31] S. Shwartz, E. Namer, and Y. Y. Schechner, "Blind haze separation," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2, pp. 1984-1991: IEEE, 677.
- [32] S. G. Narasimhan and S. K. Nayar, "Interactive (de) weathering of an image using physical models," in *IEEE Workshop on color and photometric Methods in computer Vision*, 2003, vol. 6, no. 6.4, p. 1: France.
- [33] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397-1409, 2012.
- [34] K. He and J. Sun, "Fast guided filter," *arXiv preprint arXiv:1505.00996*, 2015.
- [35] S. Lee, S. Yun, J.-H. Nam, C. S. Won, and S.-W. Jung, "A review on dark channel prior based image dehazing algorithms," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, pp. 1-23, 2016.
- [36] H. Xu, J. Guo, Q. Liu, and L. Ye, "Fast image dehazing using improved dark channel prior," in *2012 IEEE International Conference on Information Science and Technology*, 2012, pp. 663-667, Bilateral DCP: IEEE.
- [37] D. Berman, S. Avidan, and T. Treibitz, "Non-local image dehazing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1674-1682.
- [38] D. Berman, S. Avidan, and T. Treibitz, "Air-light estimation using haze-lines," *2017 IEEE International Conference on Computational Photography (ICCP)*, pp. 1-9, 2017.
- [39] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE transactions on image processing*, vol. 24, no. 11, pp. 3522-3533, 2015.
- [40] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187-5198, 2016.

- [41] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4770-4778.
- [42] W.-T. Chen, H.-Y. Fang, J.-J. Ding, and S.-Y. Kuo, "PMHLD: Patch Map-Based Hybrid Learning DehazeNet for Single Image Haze Removal," *IEEE Transactions on Image Processing*, vol. 29, pp. 6773-6788, 2020.
- [43] T. Dong, G. Zhao, J. Wu, Y. Ye, and Y. Shen, "Efficient traffic video dehazing using adaptive dark channel prior and spatial-temporal correlations," *Sensors*, vol. 19, no. 7, p. 1593, Apr 2 2019.
- [44] B. Cai, X. Xu, and D. Tao, "Real-time video dehazing based on spatio-temporal mrf," in *Pacific Rim conference on multimedia*, 2016, pp. 315-325: Springer.
- [45] Y.-F. Hsu, "A generalization of Piéron's law to include background intensity and latency distribution," *Journal of Mathematical Psychology*, vol. 49, no. 6, pp. 450-463, 2005.
- [46] B. Li *et al.*, "Reside: A benchmark for single image dehazing," *arXiv preprint arXiv:1712.04143*, vol. 1, 2017.
- [47] Y. Zhang, L. Ding, and G. Sharma, "Hazerd: an outdoor scene dataset and benchmark for single image dehazing," in *2017 IEEE international conference on image processing (ICIP)*, 2017, pp. 3205-3209: IEEE.
- [48] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973-992, 2018.
- [49] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer, "O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 754-762.
- [50] C. O. Ancuti, C. Ancuti, M. Sbert, and R. Timofte, "Dense-Haze: A benchmark for image dehazing with dense-haze and haze-free images," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1014-1018: IEEE.
- [51] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213-3223.
- [52] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231-1237, 2013.
- [53] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3828-3838.
- [54] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth," pp. 1164-1174, 2021/6//.
- [55] X. Cheng *et al.*, "Hierarchical Neural Architecture Search for Deep Stereo Matching," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22158-22169, 2020.
- [56] Z. Zou, "Castle in the Sky: Dynamic Sky Replacement and Harmonization in Videos," *arXiv preprint arXiv:2010.11800*, 2020.
- [57] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer London, 2010.
- [58] L. Guan, Y. He, and S.-Y. Kung, *Multimedia Image and Video Processing*. Baton Rouge, UNITED STATES: Taylor & Francis Group, 2012.
- [59] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, 2003, vol. 2, pp. 1398-1402: Ieee.
- [60] A. C. Bovik, *The Essential Guide to Image Processing*. Elsevier Science, 2009.

- [61] H. R. Wu and K. R. Rao, *Digital Video Image Quality and Perceptual Coding*. CRC Press, 2017.
- [62] L. K. Choi, J. You, and A. C. Bovik, "Referenceless prediction of perceptual fog density and perceptual image defogging," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3888-3901, 2015.
- [63] M. R. Luo, G. Cui, and B. Rigg, "The development of the CIE 2000 colour-difference formula: CIEDE2000," *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, vol. 26, no. 5, pp. 340-350, 2001.
- [64] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209-212, 2012.
- [65] C. G. Bampis, P. Gupta, R. Soundararajan, and A. C. Bovik, "SpEED-QA: Spatial efficient entropic differencing for image and video quality," *IEEE signal processing letters*, vol. 24, no. 9, pp. 1333-1337, 2017.
- [66] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Blind/referenceless image spatial quality evaluator," in *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, 2011, pp. 723-727: IEEE.
- [67] N. Venkatanath, D. Praneeth, M. C. Bh, S. S. Channappayya, and S. S. Medasani, "Blind image quality evaluation using perception based features," in *2015 Twenty First National Conference on Communications (NCC)*, 2015, pp. 1-6: IEEE.
- [68] D. L. Ruderman, "The statistics of natural images," *Network: computation in neural systems*, vol. 5, no. 4, p. 517, 1994.
- [69] T. Bansal, M. Das, and C. Bhattacharyya, "Content driven user profiling for comment-worthy recommendations of news and blog articles," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 195-202.
- [70] T. Bansal, P. Verga, N. Choudhary, and A. McCallum, "Simultaneously linking entities and extracting relations from biomedical text without mention-level supervision," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 05, pp. 7407-7414.
- [71] M. Lange, D. Zühlke, O. Holz, T. Villmann, and S.-G. Mittweida, "Applications of lp-Norms and their Smooth Approximations for Gradient Based Learning Vector Quantization," in *ESANN*, 2014.
- [72] G. Takacs, "Smooth maximum based algorithms for classification, regression, and collaborative filtering," *Acta Technica Jaurinensis*, vol. 3, no. 1, pp. 27-63, 2010.
- [73] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [74] J. Močkus, "On Bayesian methods for seeking the extremum," in *Optimization techniques IFIP technical conference*, 1975, pp. 400-404: Springer.
- [75] J. Mockus, *Bayesian approach to global optimization: theory and applications*. Springer Science & Business Media, 2012.
- [76] C.-L. Hwang and A. S. M. Masud, *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Springer Science & Business Media, 2012.
- [77] C. G. Bampis, P. Gupta, R. Soundararajan, and A. C. Bovik. (2017, Accessed: 26/08/2021). *SpEED-QA Software Release*. Available: https://github.com/christosbampis/SpEED-QA_release
- [78] L. K. Choi, J. You, and A. C. Bovik. (2015, Accessed: 26/08/2021). *FADE Software Release*. Available: http://live.ece.utexas.edu/research/fog/FADE_release.zip
- [79] K. He. (2015, Accessed: 26/08/2021). *Fast Guided Filter MATLAB Code*. Available: <http://kaiminghe.com/eccv10/fast-guided-filter-code-v1.rar>

- [80] M. Singal. (2018, Accessed: 26/08/2021). *PyTorch-Image-Dehazing*. Available: <https://github.com/MayankSingal/PyTorch-Image-Dehazing>
- [81] W.-T. Chen, H.-Y. Feng, J.-J. Ding, and S.-Y. Kuo. (2020, Accessed: 26/08/2021). *PMHLD-Patch-Map-Based-Hybrid-Learning-DehazeNet-for-Single-Image-Haze-Removal*. Available: <https://github.com/weitingchen83/Dehazing-PMHLD-Patch-Map-Based-Hybrid-Learning-DehazeNet-for-Single-Image-Haze-Removal-TIP-2020>
- [82] D. Berman. (2017, Accessed: 26/08/2021). *non-local-dehazing*. Available: <https://github.com/danaberman/non-local-dehazing>
- [83] B. Cai, X. Xu, and D. Tao. (2016, Accessed: 26/08/2021). *Real-time Video Dehazing based on Spatio-temporal MRF Software Release*. Available: <https://github.com/caibolun/ST-MRF/>
- [84] S. Tierney. (2014, Accessed: 26/08/2021). *MATLAB implementation of "Single Image Haze Removal Using Dark Channel Prior"*. Available: <https://github.com/sjtrny/Dark-Channel-Haze-Removal>
- [85] J.-P. Tarel. (2013, Accessed: 26/08/2021). *Vision Enhancement in Homogeneous and Heterogeneous Fog MATLAB Source Code*. Available: <http://perso.lcpc.fr/tarel.jean-philippe/visibility/visibresto2.zip>
- [86] Q. Zhu, J. Mai, and L. Shao. (2015, Accessed: 26/08/2021). *Color-Attenuation-Prior-Dehazing*. Available: <https://github.com/JiamingMai/Color-Attenuation-Prior-Dehazing>
- [87] pean1128. (2018, Accessed: 26/08/2021). *KITTI-download-scipt* [sic]. Available: <https://github.com/pean1128/KITTI-download-scipt>

Appendix: Code Overview

The repository for this project can be found at

https://cseegit.essex.ac.uk/2020_ce901/ce901_tilbury_benjamin_m_j/-/tree/master/

Below is an overview of its structure, including references to incorporated, adapted and third-party code. All third-party code is used and modified in accordance with their respective licenses, which are included with the files or folders where appropriate.

Entries highlighted in green indicated wholly original code.

Entries highlighted in yellow indicate original code with some (modified) third-party functions included.

Entries highlighted in orange indicate code that has been heavily modified and rewritten from an initial third-party source.

Ref	Structure	Description
	dehaze-evaluation	Folder containing code for the evaluation of dehazing systems
	3rd-party	Folder containing 3 rd party tools for evaluation.
[65, 77]	SpEED-QA	3 rd -Party code for the SpEED-QA evaluator
[62, 78]	FADE	3 rd -Party code for the FADE evaluator
[52]	loadCalibrationCamToCam.m	3 rd -Party code for reading KITTI calibration data
	old	Folder containing various old, currently unused code. Kept for attribution of effort
	example-results	Folder containing small sample of image results for each dehazing
	empiricalComplexity.m	Plots the runtime of dehazers for various size inputs.
	evaluateDehazers.m	Code to evaluate dehazing systems.
	groupedResults.mat	Results on the evaluated dehazing systems, averaged and grouped by haze category.
	groupResults.m	Function to group results by category.
	imageCollage.m	Function to create a collage of dehazed images for given dehazing systems
	labelScatter.m	Function to plot labelled scatter graphs.
	normalisedScore.m	Function to normalise results.
	plotScoring.m	Function to normalise and plot results.
	results.mat	Full results on the evaluated dehazing systems, grouped by sequence
	saveVideos.m	Function to save the video of a given dehazing system's results on the entire dataset.
	dehaze-systems	Folder containing code for the implemented dehazing systems
	abstracts	Folder containing abstract base classes for dehazing systems
[34, 79]	BaseDehazer.m	Abstract base class for all dehazing system. Includes guided filter and window sum filter code.
	ExternalPythonDehazer.m	Abstract base class for external python dehazing systems. (Windows Only)
	neural-nets	Folder containing code for the neural network methods
[41, 80]	AOD	Folder containing model definitions for AOD neural network
[42, 81]	PMHLD	Folder containing model definitions for PMHLD neural network
	pyinterfacer.py	Python program for interfacing with MATLAB to run the two neural network methods
	old	Folder containing contains various old, currently unused code. Kept for attribution of effort
	optimise-params	Folder containing code to optimise the TilburyDehazer
	optimise-tilbury.m	Code for running Bayesian optimisation on the TilburyDehazer
	optims.mat	Results of the Bayesian optimisations
[37, 38, 82]	BermanDehazer.m	Dehazing system implementing Berman et al.'s method
[44, 83]	CaiDehazer.m	Dehazing system implementing Cai et al.'s method
	ChenDehazer.m	Dehazing system wrapper for Chen et al.'s PMHLD method (Requires TensorFlow) (Windows Only)
[17, 84]	HeDehazer.m	Dehazing system implementing He et al.'s DCP method using the guided filter.
	LiDehazer.m	Dehazing system wrapper for Li et al.'s AOD neural network (Requires PyTorch) (Windows Only)
[11, 85]	TarelDehazer.m	Dehazing system implementing Tarel et al.'s method
	TilburyDehazer.m	Proposed dehazing system containing different implementations of the smooth-extremum filter
	TsaiDehazer.m	Dehazing system implementing Tsai et al.'s method
[39, 86]	ZhuDehazer.m	Dehazing system implementing Zhu et al.'s CAP method
	UnrefinedDehazer.m	Dehazing system implementing He et al.'s DCP method without refinement.
	haze-video-dataset	Folder containing code for generating the haze video dataset.
	dataset	Default location for the haze dataset and KITTI data to be downloaded to.
[87]	download_kitti.sh	3 rd -Party script to download the required raw KITTI sequences automatically (Requires Bash shell)
	prep-tools	Default location to put repositories of components needed in the generation of the dataset.
	simplewrappers	Folder containing wrappers to interface with the 3 rd -Party components.
	simpleleastereo.py	Wrapper for running LEAStereo to generate depth maps
	simplemanydepth.py	Wrapper for running ManyDepth to generate depth maps
	simplemonodepth2.py	Wrapper for running MonoDepth2 to generate depth maps
	simpleskyar.py	Wrapper for running SkyAR to generate sky masks
	hazeutils.py	Code to generate hazy images
	hazegenerator.mlap	MATLAB app to preview hazy sequences
	kittititerator.py	A multifunctional program to iterate through the KITTI data. Generates the haze dataset.
	figures	Folder containing figures used in the dissertation paper.
	haze-subset.txt	List of KITTI sequences in the haze dataset. By default, any other sequences encountered are ignored.

Appendix: Parameter Tables

BermanDehazer

Published Parameter	Code Variable	Value	Value Source	Notes
λ	lambda	0.1	[37]	
—	t0	0.1	[82]	Not mentioned in either paper but present in the author's source code. This parameter is used in the code equivalently to the t0 parameters in DCP methods.
—	gamma	1	[82]	Not mentioned in either paper but present in the author's source code. It's optimal value is dependent on the gamma correction of input images. KITTI data is assumed to be linear and so a value of 1 is appropriate.
—	omega	1.05	[82]	Not mentioned in either paper but present in the author's source code. Though applied differently, this value functions similar to the omega parameter in DCP methods to retain a small degree of haze.
N	N	1000	[38]	
K	K	40	[38]	
ΔA	spacing	0.02	[38]	
τ_0	thres	0.02	[38]	
—	Amin	[0,0.05,0.1]	[82]	Not mentioned in the paper but present in the author's source code. Controls the minimum value of the atmospheric light search for each colour channel.
—	Amax	[1,1,1]	[82]	Not mentioned in the paper but present in the author's source code. Controls the maximum value of the atmospheric light search for each colour channel.

CaiDehazer

Published Parameter	Code Variable	Value	Value Source	Notes
ω	omega	0.7	[44]	
f	f	1	[44]	
λ_r	lambda	[0.25, 0.5, 0.25]	[44]	Hanning window of size $3f + 2$ truncated to remove zeros.
ρ	rho	0.1	[44]	
r	r	20	[83]	Mentioned but not given a value in the paper. Value derived from author's source code.
—	rA	15	[83]	Not mentioned in the paper but present in the author's source code. This parameter controls the window size for atmospheric light estimation.
—	t0	0.1	[83]	Not mentioned in the paper but present in the author's source code. This parameter is equivalent to its use in other DCP methods
—	eps	0.0001	[83]	Not mentioned in the paper but present in the author's source code. This parameter is equivalent in its use to other guided filter DCP methods.

HeDehazer

Published Parameter	Code Variable	Value	Value Source	Notes
<i>patch size</i>	winSize	15	[17]	
t_0	t0	0.1	[17]	
κ	omega	0.95	[17]	Most other DCP papers since refer to this parameter as omega (ω)
ϵ	eps	0.001	[33]	
r	rGF	20	[33]	

TarelDehazer

Published Parameter	Code Variable	Value	Value Source	Notes
s_v	winSize	11	[85]	Though a different value is provided in [10], the newer paper this dehazing system is based on [11], does not provide a value. However the code this value is taken from is the source code for the newer paper.
p	omega	0.95	[11]	
—	balance	-1	[85]	Briefly mentioned in both papers but not given a value. Value derived from author's source code.
—	gamma	1	[85]	Not mentioned in either paper, included in the code as a gamma correction factor to be based on the gamma of the input images. 1 is appropriate as the KITTI data is assumed to be linear.
d_{min}	minvd	50	[85]	This value differs from the paper but was featured in the author's code. Appears to work better than 60 (as in the paper)
λ	rc	1.65	Modified	Custom value based on KITTI data to match properties described in the paper. Is the height of the KITTI camera in meters.
v_h	vh	175	Modified	Custom value based on KITTI data to match properties described in the paper. General estimate for the pixel height of the horizon line for KITTI sequences.

TsaiDehazer

Published Parameter	Code Variable	Value	Value Source	Notes
ω	omega	0.95	[13]	
r_{min}	rMin	7	[13]	
r_{mean}	rGF	28	[13]	
ϵ	eps	0.001	Inferred	Mentioned but not given a value in the paper, uses the same value as the DCP method upon which this is based.
t_0	t0	0.1	Inferred	Mentioned but not given a value in the paper. The same value is used as in the other methods that use this paramter in the same way.
$\theta_{airlight}$	AThresh	0.85	[13]	
θ_s	skyThresh	0.6	[13]	
t_{sky}	skyLim	0.3	[13]	
θ_{diff}	ADiffThresh	0.01961	[13]	Converted from value of 5 in uint8 range [0,255] to float image range [0, 1]
λ	AUpdate	0.95	[13]	

ZhuDehazer

Published Parameter	Code Variable	Value	Value Source	Notes
β	beta	1	[39]	
r	r	15	[39]	
θ_0	theta(1)	0.121779	[39]	
θ_1	theta(2)	0.959710	[39]	
θ_2	theta(3)	-0.780245	[39]	
—	t0	0.1	[39]	No parameter name given in paper, the values are simply stated
—	t1	0.9	[39]	
—	eps	0.001	[86]	Not mentioned in paper but present in the author's code. Is the epsilon value for the colour guided filter
σ	sigma	0.041337	[39]	

Appendix: Normalisation Process

In order to provide a single quantitative score for the dehazing system results on each quality criterion, the metrics require normalisation based on the data obtained. This is because all the metrics feature different scales, and some do not have well defined bounds, so it is not possible to map each metric into the same scale without using the results themselves. A direct consequence of this is that the normalised scores only have usefulness as a comparative tool. The scores provided in Figure 12 and Figure 19 are scaled using only the results of the initial systems (Chapter 4). This was done so that the scores for the initial systems would be independent of the proposed system results, a new scale was not used for Figure 19 to maintain consistency so that the proposed method results would be easy to compare against the existing method results. The full pseudocode for generating the criteria scores is below:

Algorithm: Normalised Criteria Score Calculation

Input : Array of results \mathbf{R} of size $|M| \times |F| \times |D|$, where
 $M = \{MPPS, PSNR, MS-SSIM, FADE, CIEDE2000, NIQE, MIC, SpEED-QA, TError, AError\}$ is the set of all metrics,
 $D = D^I \cup D^N$ is the set of all dehazing systems.
 $D^I = \{Berman, Cai, Chen, He, Li, Tarel, Tsai, Zhu\}$ is the set of initial dehazing systems,
 $D^N = \{TilburyLocal, TilburyGlobal, TilburyOpening\}$ is the set of proposed dehazing systems,
 $F = \{0, 1, \dots, 15, 718\}$ all frames in the dataset.
Output: Scores a $5 \times |D|$ array of normalised and averaged scores for all dehazing systems on the 5 evaluation criteria. Scores are scaled based on the initial methods to provide a consistent comparison.

Begin

```
// Start by normalising the individual metrics
NormedMetrics  $\leftarrow$  NormaliseMetrics ( $\mathbf{R}$ )
// Now the metrics are all in the same scale, they can be combined
into the criteria scores
RawScores  $\leftarrow$  CriteriaScores (NormedMetrics)
// Because they now fall outside of the [0,1] range, each score needs to
be rescaled again.
FullScores  $\leftarrow$  NormaliseScores (RawScores)
// Finally average the results over the dataset to provide a single
score for each dehazing on each criteria
Scores  $\leftarrow$  AverageScores (FullScores)
```

end

Function: NormaliseMetrics

Input : Array of results \mathbf{R} of size $|M| \times |F| \times |D|$, as defined above.
Output: NormedMetrics an array of size $|M| \times |F| \times |D|$ of results rescaled by metric
Begin

```
Initialize NormedMetrics as empty  $|M| \times |F| \times |D|$  array
for  $m \in M$  do
    // Convert the error type metrics into score type so that for all
    metrics, a higher value is better
    UpMetric  $\leftarrow \begin{cases} -\mathbf{R}[m] & m \in \{FADE, CIEDE2000, NIQE, SpEED-QA, TError, AError\} \\ \mathbf{R}[m] & \text{otherwise} \end{cases}$ 
    // Find the minimum and maximum values for this metric out of the
    results for the initial dehazing systems  $d \in D^I$ 
    MetricMin  $\leftarrow \min_{f \in F, d \in D^I} (UpMetric[f, d])$ 
    MetricMax  $\leftarrow \max_{f \in F, d \in D^I} (UpMetric[f, d])$ 
    for  $d \in D$  do
        for  $f \in F$  do
            NormedMetrics[ $m, f, d$ ]  $\leftarrow \frac{UpMetric[f, d] - MetricMin}{MetricMax - MetricMin}$  // Scale the metrics
        end
    end
end
return NormedMetrics
```

end

Function: CriteriaScores

Input : Array of normalised metrics *NormedMetrics* of size $|M| \times |F| \times |D|$, as defined above.

Output: *Scores* an array of size $5 \times |F| \times |D|$ of criteria scores

Begin

```
Initialize Scores as empty  $5 \times |F| \times |D|$  array
// Real-time Capability
Scores[RC] = NormedMetrics[MPPS]
// Visibility Improvement
Scores[VI] =  $\frac{\text{NormedMetrics}[\text{PSNR}] + \text{NormedMetrics}[\text{MS-SSIM}] + \text{NormedMetrics}[\text{FADE}]}{3}$ 
// Naturalness Quality
Scores[NQ] =  $\frac{\text{NormedMetrics}[\text{CIEDE2000}] + \text{NormedMetrics}[\text{NIQE}]}{2}$ 
// Video Consistency
Scores[VC] =  $\frac{\text{NormedMetrics}[\text{MIC}] + \text{NormedMetrics}[\text{SpEED-QA}]}{2}$ 
// Model Conformity
Scores[MC] =  $\frac{\text{NormedMetrics}[\text{TError}] + \text{NormedMetrics}[\text{AError}]}{2}$ 
```

end

Function: NormaliseScores

Input : Array of criteria scores *RawScores* of size $5 \times |F| \times |D|$, as defined above.

Output: *Scores* an array of size $5 \times |F| \times |D|$ of normalised criteria scores

Begin

```
Initialize Scores as empty  $5 \times |F| \times |D|$  array
for  $c \in \{RC, VI, NQ, VC, MC\}$  do
    // Find the minimum and maximum values for this criteria score out
    // of the initial dehazing systems  $d \in D^I$ 
    ScoreMin  $\leftarrow \min_{f \in F, d \in D^I} (\text{RawScores}[c, f, d])$ 
    ScoreMax  $\leftarrow \max_{f \in F, d \in D^I} (\text{RawScores}[c, f, d])$ 
    for  $d \in D$  do
        for  $f \in F$  do
            Scores[ $c, f, d$ ]  $\leftarrow \frac{\text{RawScores}[c, f, d] - \text{ScoreMin}}{\text{ScoreMax} - \text{ScoreMin}}$  // Scale the scores
        end
    end
end
```

end

Function: AverageScores

Input : Array of criteria scores *FullScores* of size $5 \times |F| \times |D|$, as defined above.

Output: *Scores* an array of size $5 \times |D|$ of normalised criteria scores

Begin

```
Initialize Scores as empty  $5 \times |D|$  array
for  $c \in \{RC, VI, NQ, VC, MC\}$  do
    for  $d \in D$  do
        Scores[ $c, d$ ]  $\leftarrow \sum_{f \in F} \frac{\text{Scores}[c, f, d]}{|F|}$ 
    end
end
```

end
