# Surround-View Fisheye Camera Perception for Automated Driving: Overview, Survey & Challenges

Varun Ravi Kumar, Ciarán Eising, *Member, IEEE*, Christian Witt, and Senthil Yogamani

***Abstract*— Surround-view fisheye cameras are commonly used for near-field sensing in automated driving. Four fisheye cameras on four sides of the vehicle are sufficient to cover 360° around the vehicle capturing the entire near-field region. Some primary use cases are automated parking, traffic jam assist, and urban driving. There are limited datasets and very little work on near-field perception tasks as the focus in automotive perception is on far-field perception. In contrast to far-field, surround-view perception poses additional challenges due to high precision object detection requirements of 10cm and partial visibility of objects. Due to the large radial distortion of fisheye cameras, standard algorithms cannot be extended easily to the surround-view use case. Thus, we are motivated to provide a self-contained reference for automotive fisheye camera perception for researchers and practitioners. Firstly, we provide a unified and taxonomic treatment of commonly used fisheye camera models. Secondly, we discuss various perception tasks and existing literature. Finally, we discuss the challenges and future direction.**

***Index Terms*— Automated driving, omnidirectional camera, fisheye camera, surround view perception, bird-eye's view perception, multi-task learning.**

## I. Introduction

SURROUND-VIEW systems use four sensors to form a network with overlap regions, sufficient to cover the near-field area around the car. Figure 1 shows the four views of a typical surround-view system, along with a representation of the typical parking use-case. Wide-angle views exceeding 180° are used for this near-field sensing. Any perception algorithm must consider the significant fisheye distortion inherent with such camera systems. This is a significant challenge, as most work in computer vision focuses on narrow field-of-view cameras with mild radial distortion. However, as such camera systems are more widely deployed, work has been completed in this area. It is the aim of this paper to give the reader an overview of surround view cameras (*e.g.*, image formation, configuration, and rectification), to survey the existing state of the art, and to provide insights into the current challenges in the area.

Manuscript received 26 May 2022; revised 26 October 2022; accepted 29 December 2022. Date of publication 10 January 2023; date of current version 29 March 2023. The Associate Editor for this article was Z. Duric. *(Varun Ravi Kumar and Ciarán Eising are co-first authors.) (Corresponding author: Ciarán Eising.)*

Varun Ravi Kumar, Christian Witt, and Senthil Yogamani are with Valeo, Tuam, Galway H54 Y276, Ireland.

Ciarán Eising is with the Department of Electronic and Computer Engineering, University of Limerick, Limerick, V94 T9PX Ireland (e-mail: ciaran.eising@ul.ie).

Digital Object Identifier 10.1109/TITS.2023.3235057

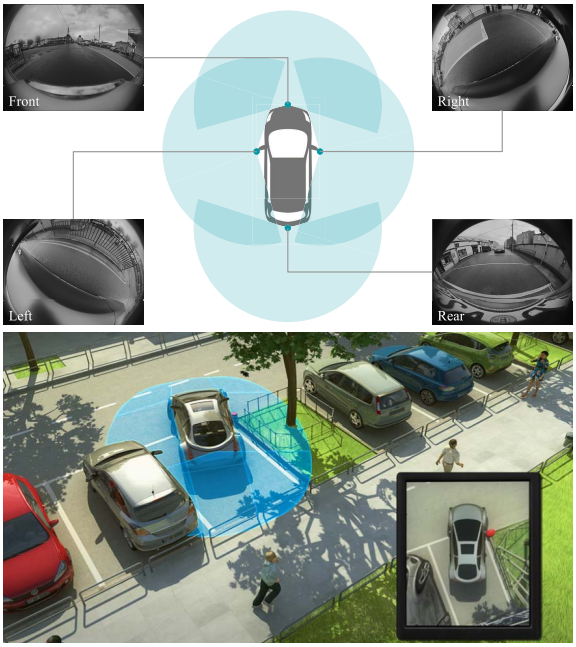



Fig. 1. **Illustration of a typical automotive surround-view system** consisting of four fisheye cameras located at the front, rear, and on each wing mirror (top). The bottom figure illustrates the near field covering the entire 360° around the vehicle. Surround visualization for the driver by stitching the four cameras is also illustrated within the smaller box.

In theory, the field-of-view of a pinhole camera is 180°. However, in practice, due to the practical limitations of the size of the aperture and imager, it is not easy to get over 80°, as illustrated in Figure 2 (top). Fisheye lenses are commonly used to effectively increase the field-of-view to 180° or more. It is interesting to note that the term *fisheye* is a bit of a misnomer, as illustrated in Figure 2 (bottom). Due to the bending of light rays due to refraction at the junction of water and air surface, a large field-of-view of nearly 180° is compressed to a smaller field-of-view of nearly 100°. A human swimmer would observe the same effect; it is nothing to do with the optics of fish's eye.

The development of fisheye cameras has a long history. Wood initially coined the term fisheye in 1908 and constructed a simple fisheye camera [1], a fact that is acknowledged in the naming of the recently released *WoodScape* dataset of automotive fisheye video [2]. This water-based lens was replaced with a hemispherical lens by Bond [3], and thus began the optical development of fisheye cameras. Miyamoto [4] provided early insight into the modelling of geometric distortion in fisheye cameras, suggesting the use of equidistant, stereographic, and

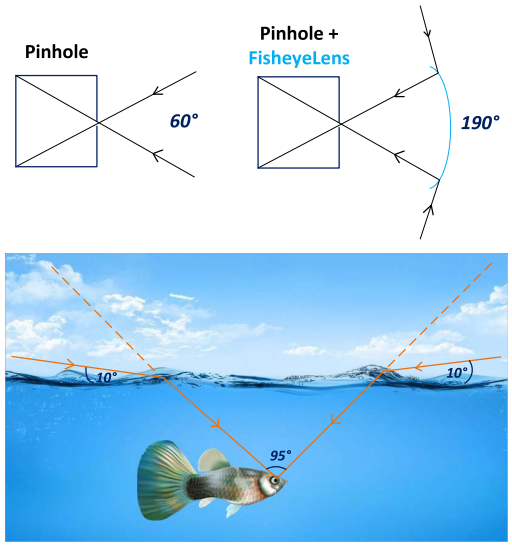



Fig. 2. **Illustration of fisheye perspective**. (top) Rays incident on a pinhole camera at wide angles cannot be imaged effectively beyond 60°. The addition of a fisheye lens dramatically increases the field-of-view to 190° due to refraction. (bottom) Refraction of light rays at the surface of water causes compression of the horizon into a smaller field-of-view.

equisolid models. These models were already known in the field of cartography (*e.g.*, [5] and many others).

### A. Applications

Fisheye cameras offer a significantly wider field-of-view than standard cameras, often with a 180° field-of-view or even greater. This can offer several advantages, not least that fewer cameras can be used to achieve complete coverage. The first successful commercial application of fisheye cameras was in photography, particularly in the entertainment industry, where the fisheye lens effect became a stylistic element. A video by Vox [6] provides an excellent overview of the history of its usage. The second successful area of application is video surveillance, where the hemispherical lens surface can be commonly seen in modern surveillance systems [7]. More recently, wide-angle lens cameras are commonly used in virtual reality headsets [8]. They are also commonly used in underwater robotics [9], and aerial robotics [10].

Automotive is one of the important application areas of fisheye cameras where more advanced visual perception is necessary. The first wide-angle rear-view camera and a TV display were deployed in General Motors' Buick Centurion concept model in 1956. In 2018, a rear-view fisheye camera was mandated in the United States to reduce accidents during reversing [11]. In 2008, surround-view cameras were deployed by BMW for park view [12]. Surround-view cameras have become a commonly used feature in many vehicles. They were subsequently used for computer vision applications like cross-traffic alerts [13], object detection [14], and automated parking [15]. Figure 1 (top) illustrates the position of the cameras and sample images of a surround-view system. Figure 1 (bottom) shows the near-field region, and it forms the primary sensor for 360° sensing around the vehicle. Surround visualization for the driver by stitching the four cameras is also illustrated within the smaller box.


Fig. 3. **Standard bounding box is not a good object representation for fisheye images**. (a) Red pixels within the bounding box show a large area that does not contain the object. Oriented box (b) and curved bounding box (c) are better representations [14].

Fisheye cameras have several challenges, however. The most obvious is that they exhibit a strong radial distortion that cannot be corrected without disadvantages, including reduced field-of-view and resampling distortion artifacts at the periphery [16]. Appearance variations of objects are larger due to the spatially variant distortion, particularly for close-by objects. This increases the learning complexity of a Convolutional Neural Network (CNN), which uses translation invariance as an inductive bias and increases sample complexity as the model must learn the appearance of all the distorted versions of an object. In addition, the commonly used application of object detection using a bounding box becomes more complex as the bounding box does not provide an optimal fit for fisheye distorted objects, as illustrated in Figure 3. More sophisticated representations instead of a rectangular box, such as a curved bounding box exploiting the known radial distortion of fisheye cameras, were explored in [14]. Fisheye perception is a challenging task, and despite its prevalence, it is comparatively less explored than pinhole cameras.

In the case of cameras without significant fisheye distortion, there is a very common geometry associated with them, being the *pinhole* model. One may first consider the intersection of a ray with a single planar surface at some fixed distance from the projection center. All models of the distortion due to the lens for such cameras then are simply designed to shift the intersection point position radially from the projection center on the plane. In a way, fisheye algorithm development has been complicated by the lack of a unifying geometry. Many models use different properties to describe fisheye projection. One of the aims of this paper is to examine common models and demonstrate that several of the models are highly related to one another. Several models can be seen as specific cases of the General Perspective Mapping or Ellipsoidal General Perspective Mapping, both of which have been known for many decades in other fields of science [17]. We show that a few of the presented models are even re-derivations of existing models. Thus we attempt to map a path through the many proposed models and consider them in several classes. For example, we could consider a class of *on-image* models,

in which the fisheye projection is measured as a deviation from pinhole projection, *e.g.*, [18] and [19]. Alternatively, we could consider a model in which the ray projection angle is manipulated at the projection center (*e.g.*, [2], [20]). Others still propose the use of a series of projections onto different surfaces to model fisheye distortion, for example, [21], [22], and [23], which we can refer to as *spherical models*.

### B. Relation to Other Sensors

Automated parking systems are typically designed using fisheye cameras and sonar [15]. Sonar is typically used in the front and rear, and it is very reliable to detect near-field obstacles accurately [24]. However, its range is typically limited to about 5 meters. Additionally, the information is very sparse, making it impossible to obtain richer information about the scene. Typically, a classical late fusion approach combines the perception output of the fisheye camera, and sonar [15], [25]. More recently, an array of short-range radars (SRR) providing 360° coverage, which is used for urban driving applications, are being reused for near-field sensing applications like parking. They are significantly denser than sonar and have a range of 30 meters. However, they do not cover the entire near-field, and there are some blind spots. Radar is additionally limited in that it cannot detect road markings and has limited performance in object classification [26]. Parking space detection using SRR is discussed in more detail in [27]. Fusion of fisheye camera and SRR is typically performed in a classical dynamic occupancy grid fusion framework [28]. CNN-based fusion approaches are emerging as well [29]. LiDAR is a far-field sensor with a range of over 200 meters, and thus it is typically not combined with near-field fisheye cameras. Varga et al. [30] have attempted to combine fisheye camera and LiDAR to provide a unified 360° environmental model, but there are blind spots in the near field. Classification of objects in LiDAR has extremely limited performance [26]. To summarize, other near-field sensors like radar and sonar capture limited information about the scene, and thus they cannot operate independently to perform near-field perception.

This paper is intended to be a broad overview and survey complementing our previous work [31] which is relatively a narrow discussion of our concrete architecture and implementation of surround-view perception. We list few other review papers which are related to our paper. In [15], a brief survey of computer vision for the specific use case of automated parking was provided. In [12], an early survey is provided on surround-view monitoring, though no perception tasks are discussed. Finally, [32] provides a comprehensive review of vision tasks, but not specifically for automotive surround-view systems.

The paper is organized as follows. In Section II, we discuss some of the commonly used models and build a taxonomy of these methods establishing equivalences and specialization. In Section III, we introduce the automotive setup of four fisheye cameras forming a near-field surround view system and discuss basic constructs like calibration, rectification, and geometric primitives. Section IV covers a detailed survey of visual perception tasks on surround-view cameras. Section V discusses future research directions to be explored by the community. Section VI provides concluding remarks.

## II. Fisheye Camera Models

In this section, we provide a survey of several of the more popular fisheye camera models. The aim is to provide a comprehensive list of possible models using a unified notation. For a developer, this could be seen as a tool to guide the choice of model for a given application. One could attempt to use the simpler, more specialized models, and, depending on the specific application, extend the development to one of the more general models in the case that errors remain high for a given camera following calibration.

### A. Notation and Terminology

Matrices are denoted by $A \in \mathbb{R}^{m\times n}$. The usual notation for ordinary vectors $\mathbf{v} \in \mathbb{R}^n$ will be used, represented as $n$-tuples. Specifically, points in $\mathbb{R}^3$ will be denoted as $\mathbf{X} = (X, Y, Z)^\mathsf{T}$, and a point in the set of image points $I^2$ is denoted as $\mathbf{u} = (u, v)^\mathsf{T}$. The unit sphere is defined by $S^2 = \left\{\mathbf{s} \in \mathbb{R}^3 \mid \|\mathbf{s}\| = 1\right\}$, and points on the unit sphere are represented as 3-vectors, i.e., $\mathbf{s} = (x, y, z)^\mathsf{T}$.

We can define a mapping from $C^3 \subseteq \mathbb{R}^3$ to the image as

$$\pi : C^3 \to I^2$$

where $C^3$ denotes the set of points for which the projection $\pi$ is defined. $I^2 \subseteq \mathbb{R}^2$ denotes the image following projection from $C^3$. $\theta$ (usually in radians) is used to denote the field-angle (angle against the $Z$-axis) of the imaged point, and $\theta_{max}$ indicates the maximum field-angle of the model.

A true inverse of $\pi$ is naturally not possible. However, we can define an unprojection function mapping from the image domain to the unit central projective sphere

$$\pi^{-1} : I^2 \to S^2$$

In some cases, the analytic unprojection $\pi^{-1}(\mathbf{u})$ does not exist or has singularities. Figure 4 demonstrates the relationship between the image points and the unit sphere.

We also use $\|\mathbf{u}\| = \varrho(\theta)$ to denote the radial form of the projection function. That is, this is a function that maps the field angle to a radial distance on the image plane (from the distortion center). The radial unprojection function is denoted $\theta = \varrho^{-1}(\|\mathbf{u}\|)$. The radial to incident angle unprojection is a true inverse, unlike the unprojection to the image sphere. Occasionally, we will have the need to refer to two image points, a distorted and an undistorted point. In this case, we will use the subscript $d$ and $u$ to distinguish (*e.g.*, $\mathbf{u}_d$ and $\mathbf{u}_u$). On-image mappings radially warp an image from its distorted point to the undistorted point (*i.e.*, from $\|\mathbf{u}_d\|$ to $\|\mathbf{u}_u\|$) on the image. We denote this mapping as $\|\mathbf{u}_u\| = \tau(\|\mathbf{u}_d\|)$, and its inverse $\|\mathbf{u}_d\| = \tau^{-1}(\|\mathbf{u}_u\|)$.

When discussing the models below, we use subscripts to denote the parameters and functions for each of the different models. Specifically, we use subscript $p$ for the pinhole model, $e$ for the equidistant, $s$ for the stereographic, $o$ for the orthographic, $eo$ for the extended orthographic, $div$ for division, $fov$ for field-of-view, $ucm$ for Unified Camera Model and $ds$ for Double-Sphere.

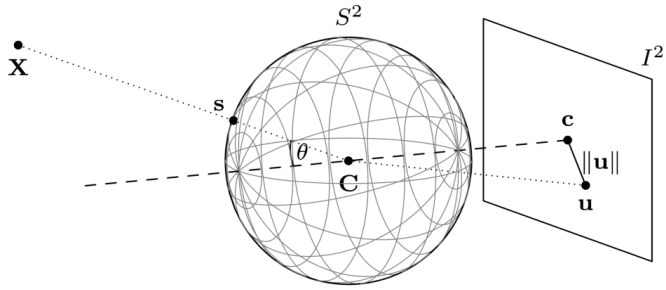



Fig. 4. Relationship between fisheye image point **u** and and its equivalent point **s** on the unit sphere, with **s** lying on the same ray as **X**.

### B. Pinhole Camera Model

The pinhole camera model is the standard projection function used in many areas of computer vision and robotics when the research is limited to considering standard field-of-view cameras. The pinhole model is given by

$$\pi_p(\mathbf{X}) = \frac{f}{Z}\begin{pmatrix} X \\ Y \end{pmatrix} \tag{1}$$

or, if we consider it as a radial function

$$\varrho_p(\theta) = f \tan\theta \tag{2}$$

where $\theta$ is the field angle of the projected ray. Note that the parameter $f$ is sometimes referred to as the focal length.

The unprojection functions are

$$\varrho_p^{-1}(\|\mathbf{u}\|) = \operatorname{atan}\left(\frac{\|\mathbf{u}\|}{f}\right), \quad \pi_p^{-1}(\mathbf{u}) = \frac{(u, v, f)^\mathsf{T}}{\|(u, v, f)^\mathsf{T}\|} \tag{3}$$

The pinhole model is defined for the set of points $C^3 = \{\mathbf{X} \in \mathbb{R}^3 \mid Z > 0\}$. The points map to the entire image plane, *i.e.*, $I^2 = \mathbb{R}^2$, and $\theta_{max} = \pi/2$. In practice, however, even when radial distortion is considered, the pinhole model is rarely of use for points with field-angle $\theta > 60°$.

### C. Classical Geometric Models

We refer to the models discussed in this section as *classical*, as they have been researched for at least six decades [4].

*1) Equidistant Projection:* In the *equidistant fisheye model*, the projected radius $\varrho_e(\theta)$ is related to the field angle $\theta$ through the simple scaling by the equidistant parameter $f$ (see Figure 5a). That is

$$\varrho_e(\theta) = f\theta, \quad \pi_e(\mathbf{X}) = \frac{f\theta}{d}\begin{pmatrix} X \\ Y \end{pmatrix} \tag{4}$$

where

$$d = \sqrt{X^2 + Y^2}, \quad \theta = \operatorname{acos}\left(\frac{Z}{\|\mathbf{X}\|}\right) \tag{5}$$

The unprojection functions are

$$\varrho_e^{-1}(\|\mathbf{u}\|) = \frac{\|\mathbf{u}\|}{f}, \quad \pi_e^{-1}(\mathbf{u}) = \begin{pmatrix} \frac{u}{\|\mathbf{u}\|} \cdot \sin\left(\frac{\|\mathbf{u}\|}{f}\right) \\ \frac{v}{\|\mathbf{u}\|} \cdot \sin\left(\frac{\|\mathbf{u}\|}{f}\right) \\ \cos\left(\frac{\|\mathbf{u}\|}{f}\right) \end{pmatrix} \tag{6}$$

The equidistant projection is valid for the points $C^3 = \mathbb{R}^3 \setminus (0,0,0)^\mathsf{T}$, $I^2 = \{\mathbf{u} \in \mathbb{R}^2 \mid \|\mathbf{u}\| \leq f\pi\}$, and $\theta_{max} = \pi$.

*2) Stereographic Projection:* As with the equidistant model, in *stereographic projection*, the center of projection of **X** to the projection sphere is **C** (Figure 5b). The stereographic projection is therefore described by

$$\varrho_s(\theta) = 2f \tan\left(\frac{\theta}{2}\right), \quad \pi_s(\mathbf{X}) = \frac{2f}{Z + \|\mathbf{X}\|}\begin{pmatrix} X \\ Y \end{pmatrix} \tag{7}$$

The unprojection functions, which we shall need later, are

$$\varrho^{-1}(\|\mathbf{u}\|) = 2\operatorname{atan}\left(\frac{\|\mathbf{u}\|}{2f}\right) \tag{8}$$

$$\pi_s^{-1}(\mathbf{u}) = \frac{1}{4f^2 + \|\mathbf{u}\|^2}\begin{pmatrix} 4fu \\ 4fv \\ 4f^2 - \|\mathbf{u}\|^2 \end{pmatrix} \tag{9}$$

The stereographic projection is valid for the points $C^3 = \mathbb{R}^3 \setminus (0,0,0)^\mathsf{T}$, and maps these points to the entire image plane, *i.e.*, $I^2 = \mathbb{R}^2$. As such, the maximum field-angle is $\theta_{max} = \pi$.

*3) Orthographic Projection:* Similar to the previous projections models, the *orthographic projection* begins with a projection to the sphere (Figure 5c). This is followed by an orthogonal projection to the plane. The orthographic projection is therefore described by

$$\varrho_o(\theta) = f \sin\theta, \quad \pi_o(\mathbf{X}) = \frac{f}{\|\mathbf{X}\|}\begin{pmatrix} X \\ Y \end{pmatrix} \tag{10}$$

The unprojection functions are

$$\varrho_o^{-1}(\|\mathbf{u}\|) = \operatorname{asin}\left(\frac{\|\mathbf{u}\|}{f}\right)$$
$$\pi_o^{-1}(\mathbf{u}) = \frac{1}{f}\begin{pmatrix} u \\ v \\ \sqrt{f_o^2 - \|\mathbf{u}\|^2} \end{pmatrix} \tag{11}$$

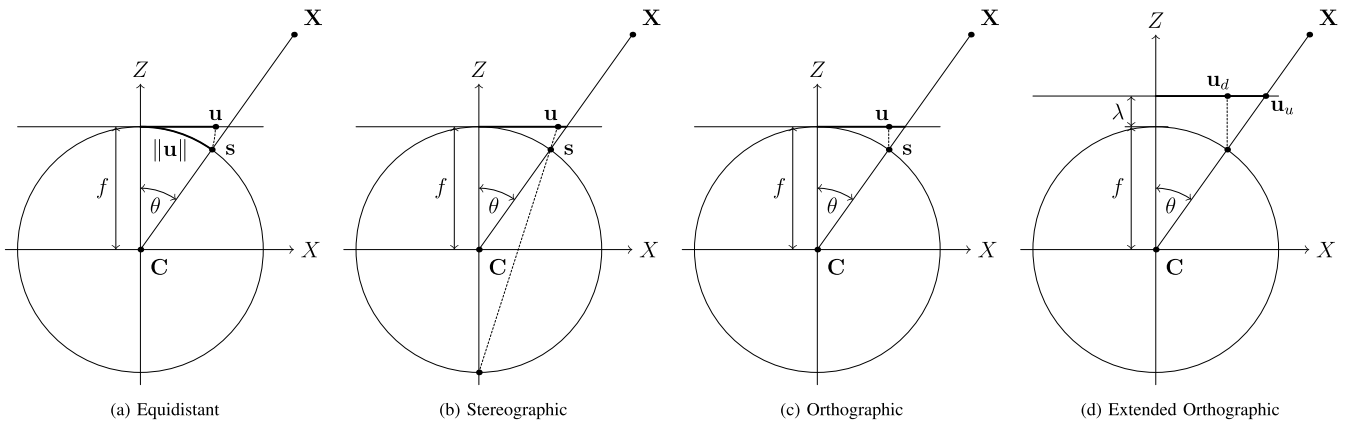
Here, $I^2 = \mathbb{R}^2$, $C^3 = \{\mathbf{X} \in \mathbb{R}^3 \mid Z > 0\}$, and $\theta_{max} = \pi/2$. These unprojection functions are well defined, as $f \geq \|\mathbf{u}\|$, which is enforced by the original projection (10).

*4) Extended Orthographic Model:* The *Extended Orthographic Model* [33], as demonstrated by Figure 5d, extends the classical orthographic model by freeing the projection plane from being tangential to the projection sphere, allowing an offset $\lambda$. The distorted projection remains the same as equations (10). However, the relationship between the distorted and undistorted radial distances and its inverse is given by

$$\tau_{eom}(\|\mathbf{u}_d\|) = \frac{(\lambda+f)\|\mathbf{u}_d\|}{\sqrt{f^2 - \|\mathbf{u}_d\|^2}} \tag{12}$$

$$\tau_{eom}^{-1}(\|\mathbf{u}_u\|) = \frac{f\|\mathbf{u}_d\|}{\sqrt{(\lambda+f)^2 + \|\mathbf{u}_u\|^2}} \tag{13}$$

This is slightly simplified representation to that presented in [33], and assumes that $f$ and $(\lambda + f)$ are positive, which is entirely practical constraints. The extended orthographic model has the same domain and co-domain as the standard orthographic model.




(a) Equidistant (b) Stereographic (c) Orthographic (d) Extended Orthographic

Fig. 5. Classical Geometric models.

*5) Extended Equidistant Model:* In fact, the extended orthographic model is simply a conversion from a projection to an on-image map. Many models can be converted to on-image mappings in the same manner as the extended orthographic model. We give just one example of the equidistant model.

Rearranging (4) such that $\theta = \|\mathbf{u}_s\|/f$, substituting into (2), and letting the focal length of (2) be $f+\lambda$, we get the on-image mapping for the equidistant model. Following similar steps, we can also obtain the inverse.

$$\tau_e(\|\mathbf{u}_d\|) = (f+\lambda)\tan\left(\frac{\|\mathbf{u}_d\|}{f}\right) \tag{14}$$

$$\tau_e^{-1}(\|\mathbf{u}_u\|) = f\,\text{atan}\left(\frac{\|\mathbf{u}_u\|}{f+\lambda}\right) \tag{15}$$

This is described in [34], albeit without the additional scaling parameter $\lambda$. We could follow the same steps above to obtain an *Extended Stereographic Model* as well. $C^3$, $I^2$ and $\theta_{max}$ for these extended models are the same as for the orthographic model.

### D. Algebraic Models

We provide a short discussion on algebraic models of fisheye cameras, specifically polynomial models, and the division model. The polynomial model discussion we provide for completeness, though we concentrate on the geometric models for the remainder of the paper.

*1) Polynomial Models:* The classical *Brown–Conrady model* of distortion for non-fisheye cameras [35], [36] uses an odd-termed polynomial, $\|\mathbf{u}_d\| = P_n(\|\mathbf{u}_u\|)$, to describe the radial distortion on the image (*i.e.* mapping $\|\mathbf{u}_u\|$ to $\|\mathbf{u}_d\|$), where $P_n$ represents some arbitrary $n$th order polynomial. Despite its age, the Brown-Conrady model is the standard distortion model in software implementations for non-fisheye cameras [37], [38]. To account for fisheye distortion, an on-image polynomial model known as the *Polynomial Fisheye Transform* (PFET), was proposed in [18]. The difference between the PFET and the Brown-Conrady model is that the PFET allows both odd and even exponents to account for the added distortion encountered in fisheye cameras.

A class of polynomial fisheye models exist, in which the mapping of the field angle to the image plane is via a polynomial, i.e $\varrho_P(\theta) = P_n(\theta)$, using the angle of incidence instead of the undistorted radius. For example, Kannala-Brandt [20] (and as implemented in the popular *OpenCV* software [37]) propose an polynomial model of order $n = 5$, or more, with odd exponents only. In [2], an $n = 4$ polynomial containing both even and odd exponents is proposed. Neither model used a constant coefficient term in the polynomial, as doing so would lead to an undefined area in the center of the image. In [39] a fifth order polynomial is proposed, but they reduce it to four independent parameters if the fisheye radius and the field-of-view are known. All the above could be interpreted as generalization of the equidistant model, which is a first order polynomial. In this case, the projection sphere is replaced by some surface defined by the given polynomial. However, this is forcing a geometric interpretation with little utility.

The *MATLAB Computer Vision Toolbox* [40] and the NVidias DriveWorks SDK [41] include implementations of a polynomial-based fisheye model provided in [42]. In this case, polynomials are used to model both the projection and unprojection, negating the need for a numerical approach to invert a projection (which is a major computational problem for polynomial-based models). Note, that both polynomials are not the inverse of each other, but two different functions. These polynomials are calibrated independently, which can make it unusable if iterative approaches that project and unproject points for several times.

*2) Division Model:* The *division model* [17] of radial distortion gained some popularity due to the nice property that, at least for the single parameter variant, straight lines project to circles in the image [43], [44], [45], and for many lenses, the single parameter variant performs very well [46]. The model and its inverse are given by

$$\tau_{div}(\|\mathbf{u}_d\|) = \frac{\|\mathbf{u}_d\|}{1 - a\|\mathbf{u}_d\|^2} \tag{16}$$

$$\tau_{div}^{-1}(\|\mathbf{u}_u\|) = \frac{\sqrt{1+4\,a\|\mathbf{u}_u\|^2}-1}{2a\|\mathbf{u}_u\|} \tag{17}$$

This was extended in [34] by adding an additional scaling parameter, which improved the modeling performance for certain types of fisheye lens. While the division model

was originally presented as an *on-image* mapping, it can be expressed as the projection function

$$\varrho_{div}(\theta) = \frac{\sqrt{1+4\ af\tan^2\theta}-1}{2af\tan\theta}$$
$$\pi_{div}(\mathbf{X}) = \frac{fr'_d}{Zr'_u}\begin{pmatrix}X\\Y\end{pmatrix}$$
$$r'_u = \frac{\sqrt{X^2+Y^2}}{Z},\quad r'_d = \frac{\sqrt{1+4br'^2_u}-1}{2br'_u} \tag{18}$$

The radial projection function $\varrho_{div}(\theta)$ is simply obtained by substituting the pinhole model (2) into (16). $f$, in this case, can be thought of as the parameter of the pinhole model once distortion has been addressed by the division model. The unprojection of the division model is

$$\varrho^{-1}_{div}(\|\mathbf{u}\|) = \text{atan2}\left(\|\mathbf{u}\|,\ f\left(1-a\|\mathbf{u}\|^2\right)\right)$$
$$\pi^{-1}_{div}(\mathbf{u}) = \frac{(u',v',f)^\mathsf{T}}{\|(u',v',f)^\mathsf{T}\|},\quad \mathbf{u}' = \frac{1}{1-a\|\mathbf{u}\|^2}\mathbf{u}$$

The projection functions and the on-image mapping have the same domain, $C^3 = \left\{\mathbf{X}\in\mathbb{R}^3 \mid Z > 0\right\}$, $I^2 = \mathbb{R}^2$, and $\theta_{max} = \pi/2$.

### E. Spherical Models

A set of more recent (at least, from the last couple of decades) fisheye models are also considered, based on the projection of the point to a unit sphere (or its affine generalisation).

*1) Field-of-View Model:* The field-of-view model [19] and its inverse is defined by

$$\tau_{fov}(\|\mathbf{u}_d\|) = \frac{\tan(\|\mathbf{u}_d\|\omega)}{2\tan\frac{\omega}{2}} \tag{19}$$

$$\tau^{-1}_{fov}(\|\mathbf{u}_u\|) = \frac{\text{atan}\left(2\|\mathbf{u}_u\|\tan\frac{\omega}{2}\right)}{\omega} \tag{20}$$

The parameter $\omega$ approximates the camera field-of-view, though not exactly [19]. This is an *on-image* model, like the Division Model, where $\|\mathbf{u}_u\|$ and $\|\mathbf{u}_d\|$ define undistorted and distorted radii on the image plane. Alternatively, it can be expressed as a projection function [23].

$$\varrho_{fov}(\theta) = \frac{\text{atan}\left(2f\tan\theta\tan\frac{\omega}{2}\right)}{\omega} \tag{21}$$
$$\pi_{fov}(\mathbf{X}) = \frac{fr'_d}{r'_u}\begin{pmatrix}X\\Y\end{pmatrix}$$
$$r'_u = \sqrt{X^2+Y^2},\quad r'_d = \frac{\text{atan2}(2r'_u\tan(\frac{\omega'}{2}), Z)}{\omega'}$$

The unprojection is given by

$$\varrho^{-1}_{fov}(\|\mathbf{u}\|) = \text{atan}\left(\frac{\tan\left(\omega'\|\mathbf{u}\|\right)}{2f\tan\frac{\omega'}{2}}\right) \tag{22}$$

$$\pi^{-1}_{fov}(\mathbf{u}) = \begin{pmatrix}\frac{u}{f}\cdot\frac{sin(r'_d\omega')}{2r'_d\tan(\frac{w}{2})}\\ \frac{v}{f}\cdot\frac{sin(r'_d\omega')}{2r'_d\tan(\frac{w}{2})}\\ \cos(r'_d\omega')\end{pmatrix},\quad r'_d = \frac{\|\mathbf{u}\|}{f} \tag{23}$$

A nice artefact of expressing the field-of-view model as a projection function is that the domain of the projection $\pi_{fov}(\mathbf{X})$ covers all of $C^3 = \mathbb{R}^3 \setminus (0,0,0)^\mathsf{T}$. In contrast, the on-image mapping form of the field-of-view model is restricted to mapping image points where $C^3 = \left\{\mathbf{X}\in\mathbb{R}^3 \mid Z > 0\right\}$, which is true of any on-image mapping. The set of imaged points is $I^2 = \left\{\mathbf{u}\in\mathbb{R}^2 \mid \|\mathbf{u}\| \le \frac{\pi}{\omega'}\right\}$, and $\theta_{max} = \pi$. We shall soon show that the field-of-view model is the equivalent of the equidistant model, and as such is a spherical projection.

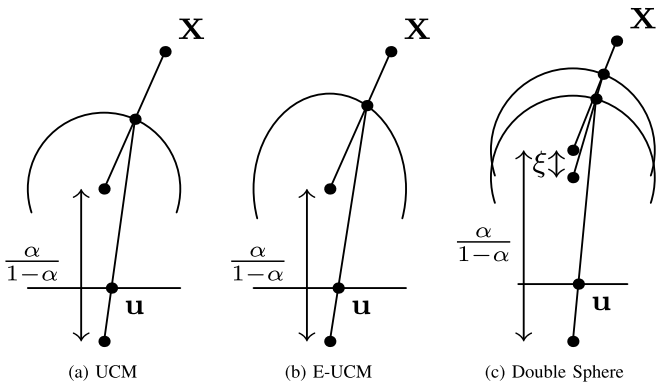



Fig. 6. **Spherical models.** The UCM (a) consists first of a projection to the unit sphere, followed by a perspective projection. The E-UCM replaced the sphere with an ellipsoid, with coefficient $\beta$. The DS model added a second unit sphere projection to the UCM, with the distance between the spheres being $\xi$.

*2) Unified Camera Model:* The UCM was initially used to model catadioptric cameras [21], and later was shown to be useful when modelling fisheye cameras [47], [48]. It has been shown to perform well across a range of lenses [46]. First, the point $\mathbf{X}$ is projected to a unit sphere, followed by a projection to a modeled pinhole camera (Figure 6a). We present the version with better numerical properties from [23].

$$\varrho_{ucm}(\theta) = \frac{f\sin\theta}{(1-\alpha)\cos\theta+\alpha} \tag{24}$$

$$\pi_{ucm}(\mathbf{X}) = \frac{f}{\alpha\|\mathbf{X}\|+(1-\alpha)Z}\begin{pmatrix}X\\Y\end{pmatrix} \tag{25}$$

The unprojection of the UCM is given by

$$\pi^{-1}_{ucm}(\mathbf{u}) = K\begin{pmatrix}u\\v\\\frac{f}{1-\alpha}\end{pmatrix} - \begin{pmatrix}0\\0\\\frac{\alpha}{1-\alpha}\end{pmatrix} \tag{26}$$

$$K = \frac{\alpha f+(1-\alpha)\sqrt{(1-2\alpha)\|\mathbf{u}\|^2+f^2}}{(1-\alpha)^2\|\mathbf{u}\|^2+f^2} \tag{27}$$

$\varrho^{-1}_{ucm}(\theta)$ is a complicated equation (more so than the above) and as such is not shown here. The domain of the projection and the radial function is given as

$$C^3 = \begin{cases}\{\mathbf{X}\in\mathbb{R}^3 \mid Z > \|\mathbf{X}\|\frac{\alpha}{\alpha-1}\}, & \text{if } \alpha \le 0.5\\ \{\mathbf{X}\in\mathbb{R}^3 \mid Z > \|\mathbf{X}\|\frac{\alpha-1}{\alpha}\}, & \text{if } \alpha > 0.5\end{cases} \tag{28}$$

$$\theta_{max} = \pi - \text{acos}\left(\frac{1-\alpha}{\alpha}\right) \tag{29}$$

$$I^2 = \begin{cases}\mathbb{R}^2, & \text{if } \alpha \le 0.5\\ \{\mathbf{u}\in\mathbb{R}^2 \mid \|\mathbf{u}\|^2 \le \frac{f^2}{2\alpha-1}\}, & \text{if } \alpha > 0.5\end{cases} \tag{30}$$

When $\alpha < 0.5$, the pinhole projection point is inside the unit sphere, outside when $\alpha > 0.5$, and on the sphere when $\alpha = 0.5$.

*3) Enhanced Unified Camera Model:* The UCM was extended by the Enhanced UCM [22] (Figure 6b), which generalizes the spherical projection with a projection to an ellipsoid (or, in fact, a general quadratic surface), and was able to demonstrate some accuracy gain. The E-UCM is given by

$$\varrho_{eucm}(\theta) = \frac{f \sin\theta}{\alpha\sqrt{\beta \sin^2\theta + \cos^2\theta} + (1-\alpha)\cos\theta} \tag{31}$$

$$\pi_{eucm}(\mathbf{X}) = \frac{f}{\alpha d + (1-\alpha)Z}\begin{pmatrix} X \\ Y \end{pmatrix} \tag{32}$$

where $d = \sqrt{\beta(X^2+Y^2)+Z^2}$, and $\beta$ is the ellipse coefficient. The unprojection function is not pretty for the EUCM, and the readers are referred instead to [22]. The set of valid points and angles is

$$C^3 = \begin{cases} \{\mathbf{X} \in \mathbb{R}^3 \mid Z > \|\mathbf{X}\| \frac{\beta\alpha}{\alpha-1}\}, & \text{if } \alpha \le 0.5 \\ \{\mathbf{X} \in \mathbb{R}^3 \mid Z > \|\mathbf{X}\| \frac{\alpha-1}{\beta\alpha}\}, & \text{if } \alpha > 0.5 \end{cases} \tag{33}$$

$$\theta_{max} = \pi - \operatorname{acos}\left(\frac{1-\alpha}{\beta\alpha}\right) \tag{34}$$

$$I^2 = \begin{cases} \mathbb{R}^2, & \text{if } \alpha \le 0.5 \\ \{\mathbf{u} \in \mathbb{R}^2 \mid \|\mathbf{u}\|^2 \le \frac{f^2}{\beta(2\alpha-1)}\}, & \text{if } \alpha > 0.5 \end{cases} \tag{35}$$

*4) Double-Sphere Model:* Later still, the UCM was extended again by the double-sphere (DS) model [23], which added a second unit sphere projection to enable more complex modeling (Figure 6c).

$$\begin{aligned} \varrho_{ds}(\theta) &= \frac{f \sin\theta}{\sqrt{1+\xi^2+2\xi\cos\theta}\left(\frac{\alpha}{1-\alpha} - \xi\right) + (\xi + \cos\theta)} \\ \pi_{ds}(\mathbf{X}) &= \frac{f}{\alpha d_2 + (1-\alpha)(\xi d_1 + Z)}\begin{pmatrix} X \\ Y \end{pmatrix} \\ d_1 &= \sqrt{x^2+y^2+z^2}, \quad d_2 = \sqrt{x^2+y^2+(\xi d_1 + Z)^2} \end{aligned} \tag{36}$$

Convincing results are presented in [23] to demonstrate the effectiveness of the double-sphere model. The unprojection functions of this model are

$$\begin{aligned} \varrho_{ds}^{-1}(\|\mathbf{u}\|) &= \operatorname{asin}(K\|\mathbf{u}\|) \\ \pi_{ds}^{-1}(\mathbf{u}) &= K\begin{pmatrix} u \\ v \\ \zeta \end{pmatrix} \\ K &= \frac{\zeta + \sqrt{\zeta^2 + (1-\xi^2)\|\mathbf{u}\|^2}}{\zeta^2 + \|\mathbf{u}\|^2} \\ \zeta &= \frac{f - \alpha\|\mathbf{u}\|^2}{\alpha\sqrt{1-(2\alpha-1)\frac{\|\mathbf{u}\|^2}{f^2}} + 1 - \alpha} \end{aligned} \tag{37}$$

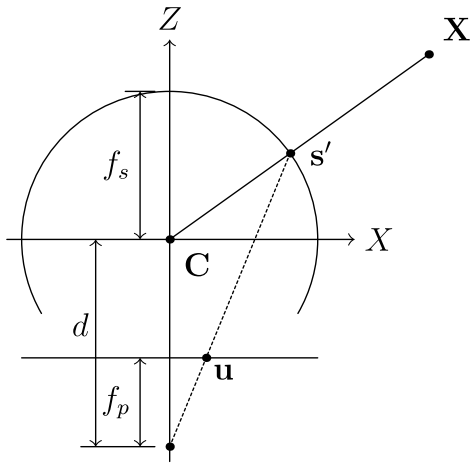



Fig. 7. **The general perspective mapping is defined by a projection of a point to the sphere of radius $f_s$, followed by a perspective projection to pinhole model with focal length $f_p$.** The two projection centers are offset by $d$. As with the classical models, it is known in cartography for many decades [53].

The valid ranges for projection and unprojection are

$$\begin{aligned} C^3 &= \{\mathbf{X} \in \mathbb{R}^3 \mid Z > -w_2\|\mathbf{X}\|\} \\ \theta_{max} &= \operatorname{acos}(-w_2) \\ I^2 &= \begin{cases} \mathbb{R}^2, & \text{if } \alpha \le 0.5 \\ \{\mathbf{u} \in \mathbb{R}^2 \mid \|\mathbf{u}\|^2 \le \frac{f^2}{2\alpha-1}\}, & \text{if } \alpha > 0.5 \end{cases} \\ w_2 &= \frac{w_1 + \xi}{\sqrt{2w_1\xi + \xi^2 + 1}}, \quad w_1 = \begin{cases} \frac{\alpha}{1-\alpha}, & \text{if } \alpha \le 0.5 \\ \frac{1-\alpha}{\alpha}, & \text{if } \alpha > 0.5 \end{cases} \end{aligned} \tag{38}$$

### F. Other Models

While we have discussed many of the more popular fisheye projection models, still, this is not exhaustive. We have omitted the details of some models that would seem a little less popular, for whatever reasons. For example, Bakstein and Pajdla [49] proposed two extensions to the classical models. A logarithm-based *Fisheye Transform* (FET) was also proposed in [18], though the accuracy was low compared to other models. The hyperbolic sin-based model proposed in [50], and later used for wide-angle cameras [51], is not discussed here, nor is the cascaded one-parameter division model [52].

### G. Unified Usage of Camera Models

With the proliferation of fisheye models, it is natural to wonder if there is a commonality between some of the models, or even if there has been repetition in development of the models.

*1) General Perspective Projection and Fisheye Models:* The unified camera model is in a class of general vertical perspective projections of a sphere, which is known in the fields of geodesy and cartography [53], [54], with the addition of the trivial step of central projection to the spherical surface. The stereographic and the orthographic projections belong to this class as well. The stereographic projection has the pinhole projection center on the surface of the sphere, while

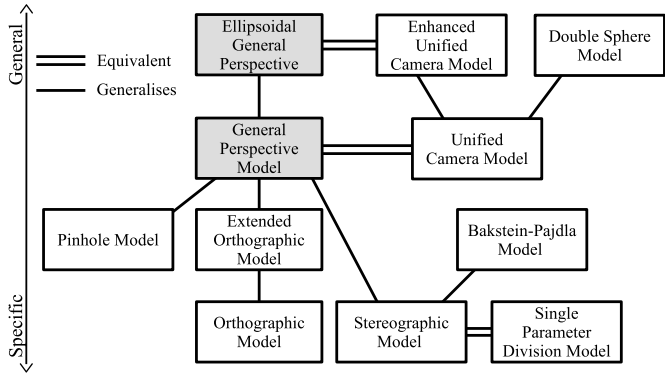



Fig. 8. **The relationship between the various fisheye models and the general perspective projections.** Double line indicates that two models are equivalent, and single line indicates a generalisation/specialisation.

the orthographic projection has an infinite focal length (hence the term orthographic). The link between the stereographic projection and the UCM is in fact described in [21].

Let us begin by examining the general vertical perspective projection, described by Figure 7. The pinhole camera is offset along the $Z$-axis by a distance $d$. The projection to the sphere is given by

$$\mathbf{s}' = f_s \frac{\mathbf{X}}{\|\mathbf{X}\|} \tag{39}$$

Here we use $\mathbf{s}' = (x', y', z')^\mathsf{T}$ for the point on the sphere of radius $f_s$, so as to distinguish it from $\mathbf{s}$ used previously to denote a point on the unit sphere. The point $\mathbf{u}$ is the pinhole projection of $\mathbf{s}'$

$$\pi(\mathbf{X}) = \frac{f_p}{z' + d} \begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{f_p}{Z + \frac{d}{f_s}\|\mathbf{X}\|} \begin{pmatrix} X \\ Y \end{pmatrix} \tag{40}$$

The $+d$ translates the point $\mathbf{s}'$ from the sphere to the pinhole coordinate system. Thus, with the two parameters $\gamma = f_p$ and the $\xi = d/f_s$, we have (24), the UCM. Additionally, if we constrain the pinhole camera plane to be on the surface of the sphere (*i.e.*, $d = f_s$), and make $f_p = 2f_s$, we get the stereographic equation (7).

The E-UCM [22] extended the UCM by projecting to an ellipsoid instead of a sphere. This type of projection is known in geodesy and cartography for a long time [53], [54] as *ellipsoidal* general perspective projections. We will not re-derive the equations here but would refer the reader to the source material. As mentioned, the DS model [23] extends the UCM by adding a second projection sphere to model more complex optics.

Thus, the UCM, the E-UCM and the DS models of fisheye lenses can be considered as generalizations of the stereographic camera model. It may be even more correct to say that they all (UCM, E-UCM, DS, division, and stereographic models) are part of a class of general perspective models. If we allow $f_s$ to approach infinity, then (40) becomes the pinhole projection model. If we allow $f_p$ (and thus also $d$) to go to infinity, then we get the orthographic projection.

*2) Stereographic and Division Models:* As discussed in [55], We can combine the pinhole projection (2) with the inverse of the stereographic model (8) to give

$$\tau_s(\|\mathbf{u}_d\|) = f \tan\left(2\text{atan}\left(\frac{\|\mathbf{u}_d\|}{2f}\right)\right) = \frac{\|\mathbf{u}_d\|}{1 - \frac{\|\mathbf{u}_d\|^2}{4f^2}} \tag{41}$$

Allow $a = 1/4f^2$, this is the same as the division model, (16). Thus, we can say that the division model is the on-image version of the stereographic projection.

*3) Equidistant and Field-of-View Models:* Consider the radial pinhole projection given by (2), and the equidistant fisheye projection model (4). Combining the two to a similar form as the field-of-view model (19)

$$\tau_e(\|\mathbf{u}_d\|) = f_p \tan\frac{\|\mathbf{u}_d\|}{f_e} \tag{42}$$

As $f_p$ and $f_e$ are free parameters, determined through calibration, we can set them to

$$f_p = \frac{1}{2\tan\frac{\omega}{2}} \quad \text{and} \quad f_e = \frac{1}{\omega} \tag{43}$$

Thus we see that (19) and (42) are equivalent mapping functions. The field-of-view model is the on-image version of the equidistant projection.

*4) Results:* To concretely demonstrate the equivalence of the Stereographic/Division and Equidistant/Field-Of-View model pairs, we provide a small set of results. Usefully, a set of parameters for five cameras is provided in [23], including parameters for the field-of-view model. Given the set of parameters $\omega$ for the field-of-view model from [23], we obtain the equidistant parameters through applying (43). See Table I. The difference between the two is at the level of machine precision, demonstrating the equivalence of the two models. The equivalence of the stereographic and division models is supported by the results presented in [46] (in particular, reference Table IV). We can see that there is zero residual when the stereographic model is compared to the division model. We can also see that there is zero residual when the UCM (called USM in [46]), or equivalently the General Perspective Mapping, is compared to the stereographic and the orthographic models.

*5) Discussion:* There is a great number of potential models for application with fisheye cameras. In this paper, we have mentioned *twenty* models, though for sure this is not exhaustive. However, we have shown that there is a strong relationship among many of the geometric models. At least seven of the models are related to or directly equivalent to the General Perspective Projection. In addition, we have shown that some of the more recently developed fisheye models are mathematically equivalent to the classical fisheye projection

TABLE I
FIELD-OF-VIEW PARAMETERS AND EQUIVALENT EQUIDISTANT PARAMETERS

| | Cam 1 | Cam 2 | Cam 3 | Cam 4 |
|---|---|---|---|---|
| $w$ | 0.93 | 0.92 | 0.95 | 0.90 |
| $f_p$ | 0.997 | 1.009 | 0.972 | 1.035 |
| $f_e$ | 1.075 | 1.087 | 1.053 | 1.111 |
| Max error ($\times 10^{-14}$) | 0.4 | 0.2 | 0.4 | 0.2 |


(a) 2D top view rendering.


(b) 3D bowl view rendering.

Fig. 9. Surround-View camera visualization.

functions, being the stereographic and the equidistant models proposed decades ago. In Figure 8, we provide a map of geometric fisheye models that are related to the General Perspective Projection.

## III. Surround View Camera System

In this section, we discuss the setup of Surround View Cameras (SVC) and its basic primitives which are necessary for perception. We start with the historical usage of SVC for visualization which provides an understanding of the automotive configuration. We then discuss the supporting modules such as calibration, rectification, and geometric primitives.

### A. Visualization

SVC have been historically used for display on the driver's dashboard for parking visualization. The first visualization application displayed a rear-view fisheye camera for reversing assist [12]. It was further improved by visualization of object detection like pedestrians and the driving tube path [56] and was subsequently enhanced into surround-view visualization using four fisheye cameras [57]. Initial systems were based on a 2D top view as shown in Figure 9 (a). This was mainly used for parking applications, but it could also be used for other low-speed maneuvering use cases like traffic jam assist. 2D top view assumed a flat ground, and thus it had artefacts when the ground surface had a non-flat profile. Other nearby objects, such as vehicles, were heavily distorted in this view. They were resolved by a 3D surround view that uses a bowl-like surface that is flat nearby and has an upward curvature towards the periphery, as shown in Figure 9 (b). In addition, depth estimation around the vehicle can be used to adapt the bowl shape for optimal viewing with lesser artifacts of nearby objects. For example, if a vehicle is nearby on one side, the bowl surface in that region is brought in front of the vehicle to avoid artifacts. Typically, the application provides a user interface to select a viewpoint needed by the driver dynamically. Surround-view visualization application is usually implemented as an OpenGL [58] rendering application using a graphics processing unit (GPU).

Classically, imaging pipelines for SVC systems are designed primarily for visualization. However, this is sub-optimal for computer vision and a dual image pipeline was proposed in [59] and [60]. The control loop part of the image pipeline such as auto-exposure and auto-gain control is typically tuned for visualization as they cannot be jointly tuned. As the four SVC point in four different directions, they may have different ambient lighting. For example, sun rays may be hitting on the front of the vehicle and the corresponding image has high sun glare and saturation. Whereas the rear camera has the corresponding shadows and is dark. To improve the visual quality, the image brightness and color are harmonized when they are stitched together [61]. This could affect computer vision if the harmonization is done in the common image pipeline for visualization and computer vision.

### B. Configuration

The main motivation for using fisheye cameras in an SVC system is to cover the entire 360° near-field region around the vehicle. This is achieved by four fisheye cameras with a large horizontal field-of-view (hFOV) of around 190° and a vertical field-of-view (vFOV) of around 150°. A fisheye camera has a very large angular volume coverage, but its angular resolution is relatively small, and it cannot perceive smaller objects at long range. Thus, it is primarily used as a near-field sensor. For comparison, a typical far-field front camera has hFOV of 120° and vFOV of 60°. The angular volume is significantly smaller, but it has a much higher angular resolution enabling it to perceive objects in far range. The large hFOV of fisheye cameras enables 360° coverage with only four fisheye cameras. The large vertical field-of-view enables capturing the region close to the vehicle, *e.g.*, detection of higher elevation objects like a traffic light when stopped at a junction.

Figure 1 shows the mounting positions and the field-of-views of a typical SVC system. Four cameras are placed on four sides of the car marked with a blue circle for their positions. The front camera is placed on the front grille of the car and the rear camera is typically on the boot door handle. Left and right-side cameras are placed under the wing mirrors. Together they cover the entire 360° region around the vehicle. The cameras are placed in such a way that the region very close to the vehicle is visible, which is crucial for parking scenarios. Because of this, a significant portion of the camera includes the ego vehicle. One can also notice the significant overlap of the field-of-view as seen in the intersecting regions. This can be exploited to resolve scale in structure from motion problems. However, this overlap is at the periphery which has the highest distortion, and it is hard to get algorithms to work accurately in this region. Figure 10 illustrates object detection and segmentation on a commercially deployed near-field perception system [31]

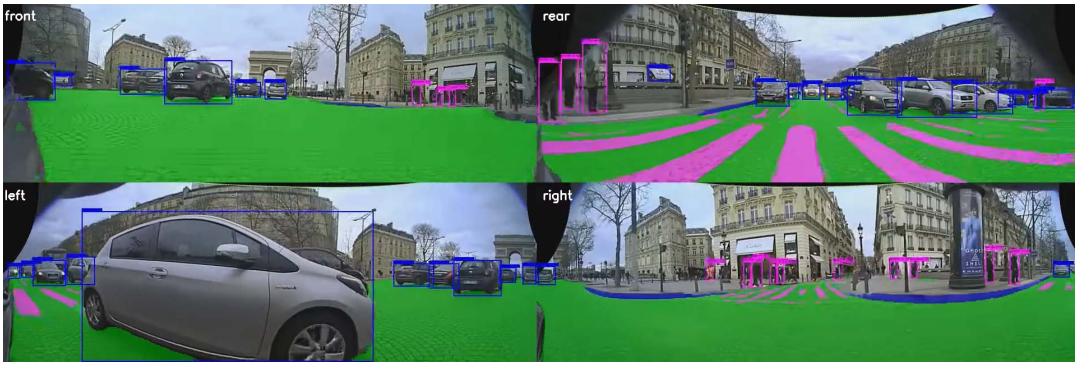



Fig. 10. **Illustration of object detection and segmentation on cylindrical rectified surround-view images from a commercially deployed system [31] (See https://youtu.be/ae8bCOF77uY for a full video).** Overlapping field-of-view can be observed by noticing the arched gate and the vehicle in front of it.

tested on a busy urban street in Paris. The overlapping field-of-view can be observed by noticing the arched gate which is seen in the center of the front camera and on the edges of the left and right cameras. The silver car in front of the gate is detected in all three cameras. Very wide-angle detections of vehicles (left end of left camera) and pedestrians (left end of rear camera) can also be observed.

### C. Calibration

Previously, we have discussed various models for fisheye cameras. Each of these models has a set of parameters (known as *intrinsic parameters*, that must be estimated through a calibration procedure. In addition, the *extrinsic parameters* of the camera should be estimated, being the position and orientation of the camera system in the vehicle coordinate system [62], [63]. A typical calibration process is that, first, image features are being detected (*e.g.*, corners in the check board pattern [64]) and secondly an algorithm will try to estimate the intrinsic and extrinsic parameters to project the detected features using the model of the calibration setup, by minimizing the reprojection error of the points. The reprojection error indicates hereby how well a model having a set of parameters can represent the projection function of the lens. Other photogrammetric approaches use vanishing Point Extraction and sets lines for estimating the calibration parameters [34], [44]. A widely known toolbox for calibration is implemented in the OpenCV library [37]. OpenCV also offers a version for fisheye camera models [20] (§II-D.1). Other known calibration toolboxes for fisheye cameras are OCamCalib [42], [65], [66] and Kalibr [67], [68], [69], [70]. Finally, in [71], a calibration process for multiple fisheye cameras on a vehicle is proposed (intrinsic and extrinsic), based on the extraction of checkerboard features and inter-camera correspondences. This is suitable for surround view camera systems, as it provides an accurate extrinsic calibration with respective to the vehicle is prerequisite for providing a seamless surround-view image. When the calibration patterns have a known position with respective to the vehicles coordinate system, the pose of the camera can be estimation like described above in off-line environment [72]. Over the lifetime of a vehicle, the camera's pose relative to the vehicle can drift, due to wear of mechanical parts. It is desirable for the camera system to update its calibration automatically, with a class of algorithms. To correct the camera poses' change in online environments, it is possible to minimize photometric errors between ground projections of adjacent cameras [73]. The approach of Choi et al. exploit corresponding lane markings captured and detected by adjacent cameras to refine an initial calibration [74]. In [75] Ouyang et al., a strategy to optimize the exterior orientations by taking the vehicle odometry into account is presented, by estimating the vehicle forward movement using geometry consistency and the vehicle direction using the vertical vanishing point estimates. Those algorithms are mostly used to correct geometric misalignment, but require an initial location obtained by an offline calibration. Friel et al. [76] describe a method of automatically extracting fisheye intrinsics from an automotive video sequence using, though it is limited to single parameter fisheye models (such as the equidistant model).

### D. Projection Geometry

In a pinhole camera, any set of parallel lines on a plane converge at a single *vanishing point*. Those can be used to estimate the intrinsic and extrinsic parameters. For pinhole camera models, geometric problems can often be formulated using linear algebra. In this case parallel lines can be detected using a Hough-Transformation [77]. The set of all vanishing points is the *horizon line* for that plane. In a real world camera system, the pinhole camera is a mathematical model of the camera, that has errors in the form of, *e.g.*, optical distortions. This is generally acceptable for narrow field-of-view cameras, where the distortion is mild. For wide field-of-view cameras, the distortion is too great for this to be a practical solution, and if the field-of-view (FOV) of the camera is greater than 180°, then there is not a one-to-one relationship of points in the original image to the corrected image plane. For fisheye cameras, a better model is the spherical projection surface [78], [79]. In the fisheye image, Hughes et al. describe in [34], how those parallel lines can be approximated and fitted as circles or conics for fisheye cameras to determine vanishing points or horizontal lines. These parallel lines correspond to great circles of the spherical surface. Correspondingly, straight lines imaged by a fisheye camera are approximately conic [55], and parallel lines imaged by a fisheye camera converge at two vanishing points (Figure 11).

Fig. 11. **Illustration of horizon line, vanishing points, and epipolar lines.** Lines in the fisheye image can be approximated as conics. Equivalent to how parallel straight lines in a perspective image converge at a single vanishing point, parallel straight lines in a fisheye image converge at two vanishing points. The two vanishing points, when raised to the unit sphere, are antipodal points on the sphere. Red and green denote the perspective of horizontal parallel lines (with blue as the associated horizon line) and vertical parallel lines respectively. The red and green dots denote the vanishing points with one of the vanishing points outside the image.

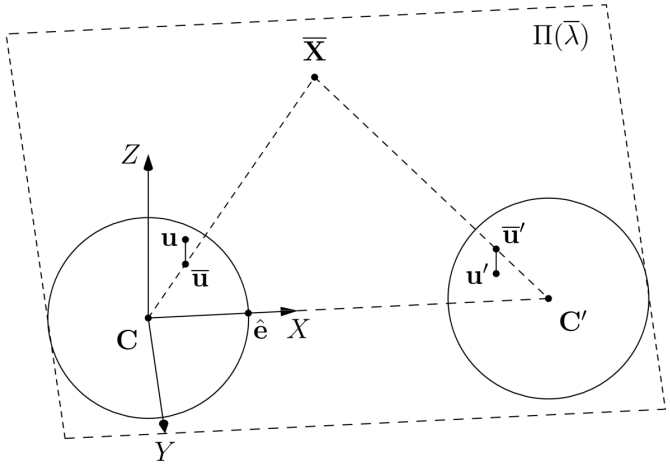



Fig. 12. **Spherical epipolar geometry.** The epipolar plane $\Pi$ is one of the pencil of planes about the epipole $\mathbf{e}$, defined by the camera centers $C$ and $C'$. Ideal observations $\overline{\mathbf{u}}$ and $\overline{\mathbf{u}}'$ will lie on the epipolar plane. However, actual observed points $\mathbf{u}$ and $\mathbf{u}'$, in the presence of noise, will have a non-zero distance to the epipolar plane.

### *E. Spherical Epipolar Geometry*

The geometric relations of stereo vision are described by *epipolar geometry*, which can be used for *depth estimation* and *structure from motion* approaches in combination with feature extractors. In pinhole camera models, the intersection of the line passing through the two camera optical centers and the image planes define special points called *epipoles*. This line is called the *baseline*. Each plane through the baseline defines matching epipolar lines in the two image planes. A point in one camera is located on an epipolar line on the other and vice versa. This reduces the search of a corresponding point (stereo matching) in a two-view camera setup to a 1D problem. For omnidirectional cameras, such as fisheye, where we employ spherical projection surfaces in place of planar, it is more intuitive to discuss epipolar planes instead of epipolar lines, as described in Figure 12. Ideal observations of a single 3D point from two cameras will lie on the same epipolar plane, in the same way that they lie on the epipolar lines in the pinhole case. It is important, however, to note that the cameras must be calibrated to raise image features to the projective sphere. In contrast, for narrow FOV cameras, epipolar geometry is defined for the uncalibrated case, via the fundamental matrix.

### *F. Rectification*

It is possible to remove radial distortion in fisheye cameras and re-use standard perception algorithms. Although it is a rapid way to start fisheye camera perception development, there are several problems associated with rectification. Firstly, it is theoretically impossible to rectify a fisheye image to a rectilinear viewport as the horizontal field-of-view is greater than 180°, thus there are rays that are incident on the lens behind the camera which doesn't work for pinhole setup. It is counterproductive to use a fisheye lens with a large field-of-view and then lose some of it because of rectification. The second significant problem is resampling distortion which is more practical in nature. It is a particular manifestation of interpolation artifacts, wherein for fisheye images, a small patch (particularly at the periphery where the distortion is high) is expanded to a very large region in the rectified image leading to high noise. In particular, the negative impact on computer vision due to the introduction of spurious frequency components by resampling is discussed in [80]. Additionally, the warping step is needed at inference time, which consumes significant computing power and memory bandwidth. It creates a non-rectangular image with invalid pixels which reduces computational efficiency further.

Commonly used rectification methods for fisheye are shown in Figure 13. Figure 13 (a) shows the standard rectilinear correction. Significant loss of near field can be observed from the missing horizontal white line. Regions at the left and right edges are missing as well. Although there is a significant loss, this enables the usage of standard camera algorithms. Figure 13 (b) shows a cubic approximation where the fisheye lens manifold surface is approximate by an open cube. It can be interpreted as a piecewise linear approximation of the fisheye projection surface. Each plane is a rectilinear correction and hence standard algorithms can be used within each block. However, the distortion across two surfaces of the cube has a large distortion and it will be difficult to detect objects which are split across the two regions. One can also notice the strong perspective distortion and blurriness due to re-sampling artifacts at the periphery.

Practically, a common rectification process is to use a cylindrical surface as illustrated in Figure 13 (c). It can be interpreted as a quasi-linear approximation as it is linear in the vertical direction and the surface has a quadratic curvature in the horizontal direction. It covers a significantly larger field-of-view relative to a rectilinear viewport. The main advantage is

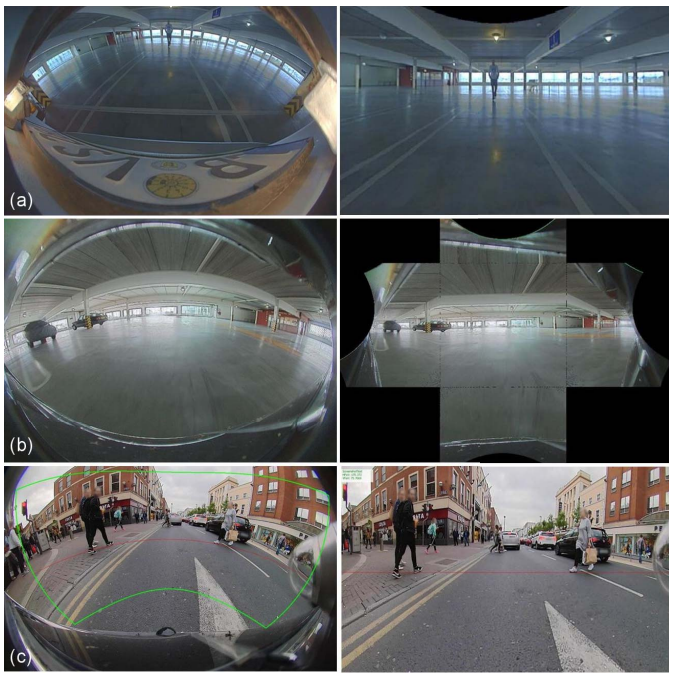



Fig. 13. **Undistortion of fisheye images:** (a) Rectilinear correction; (b) Piecewise linear correction; (c) Cylindrical correction. Left: raw image; Right: undistorted image.

that the vertical objects remain vertical as observed by vertical lines on the building [81]. Thus, scanlines are preserved for performing searches horizontally for stereo algorithms between two consecutive fisheye images (motion stereo) or between a fisheye and a narrow field-of-view camera (asymmetric stereo). The main disadvantage is its inherent inability to capture the near field region close to the vehicle. This can be fixed by using an additional smooth surface covering the near-field region. There is also an increased distortion of nearby objects.

## IV. Perception Tasks

There is relatively less literature on perception tasks for fisheye images as there are limited datasets. We split the perception tasks into semantic, geometric, and temporal tasks. Finally, we discuss joint multi-task models.

### A. Semantic Tasks

In this section, we discuss semantic tasks which are based on appearance-based pattern recognition.

*1) Semantic Segmentation:* It is the process of assigning a class label to each pixel in an image such as a pedestrian, road, or curb as shown in Figure 14 ($2^{nd}$ column). CNN-based approaches have recently been very successful compared to classical computer vision-based methods on semantic segmentation employed on a pinhole front camera [82]. Although, in urban traffic scenarios, autonomous cars require a wider field-of-view to perceive what is around them, particularly at intersections. An Overlapping Pyramid Pooling module (OPP-Net) was presented by Deng et al. [83] by employing multiple focal lengths to generate various fisheye images with their respective annotations. The OPP-Net was trained and evaluated on an existing urban traffic scene semantic segmentation dataset on fisheye images. Furthermore, to improve the model's generalization performance [83] proposed a novel zoom augmentation technique to augment the data specifically designed for fisheye images. Extensive experiments indicated the effectiveness of the zoom augmentation technique and the OPP-net performed well in urban traffic scenarios. Saez et al. [84] introduced a real-time semantic segmentation technique which is an adaptation of Efficient Residual Factorized Network (ERFNet) [85] to fisheye road sequences and generated a new semantic segmentation dataset for fisheye cameras based on CityScapes [86]. The tests were performed on authentic fisheye sequences, although only qualitative results were revealed as there is no ground truth.

Deng et al. [87] uses surround-view cameras to tackle 360° road scene segmentation as they are widely adopted in production vehicles. To deal with the distortion issues in fisheye images, Restricted Deformable Convolution (RDC) was proposed. They allow effective geometric transformation modeling by learning the shape of the convolutional filter based on the input feature map. Also, the authors presented a zoom augmentation technique for converting perspective images into fisheye images. This facilitates in the creation of a large-scale training set of surround-view camera images. An RDC-based semantic segmentation model is also developed. A multi-task learning (MTL) architecture is used to train for real-world surround-view camera images by combining real-world and transformed images. These models were trained on Cityscapes [86], FisheyeCityScapes [88] and SYNTHIA [89] datasets and tested on authentic fisheye images.

Clément et al. [90] focuses on showing that deformable convolutions can be used on top of an existing CNN without varying its pre-trained weights. This helps systems that bank on multiple image modalities as each model can now be modified reliably without retraining them from scratch. They also demonstrate that the deformable components can be trained independently (although if finetuning, it is recommended to use batch normalization). Authors also say that the need for large datasets of labeled fisheye images is eliminated. After learning the deformable components, finetuning network weights are not necessary for achieving high performance.

Yaozu et al. [88] presented a 7-degrees-of-freedom (DoF) augmentation technique for converting rectilinear perspective images into fisheye images. It includes a spatial relationship between the world and the fisheye coordinate system (6-DoF), as well as the virtual fisheye camera's focal length variations (1-DoF). During the training phase, rectilinear perspective images are turned into fisheye images in 7-DoF to replicate fisheye images taken by cameras with various locations, orientations, and focal lengths. This improves the model's accuracy and robustness while dealing with distorted fisheye data. The 7-DoF augmentation provides a generic solution for semantic segmentation for fisheye cameras as well as provides definite parameter settings for augmentation of autonomous driving and created the FisheyeCityScapes [88] dataset.


Fig. 14. **Qualitative results of raw fisheye images from the OmniDet framework on the WoodScape dataset [2].** The 1st column indicates the input images from Front, Left, Right and Rear cameras, 2nd column indicate distance estimates, 3rd column semantic segmentation maps, 4th indicate generalized object detection representations and 5th indicate the motion segmentation. For more qualitative results at a higher resolution, check this video: https://youtu.be/xbSjZ5OfPes.

*2) Object Detection:* Object detection suffers the most from radial distortion in fisheye images. Due to inherent distortion in the fisheye image formation, objects at different angles from the optical axis appear very different, making object detection difficult (see Figure 14). The rectangular bounding boxes tend to be not the best representation of the size of the object sometimes is half the standard bounding box whereas the box itself would be twice the object of interest. Instance segmentation provides precise contours of the objects, but they are much more expensive to annotate and require a bounding box estimation step. Rectification provides a significant benefit but it also suffers from the side effects discussed in Section III.

FisheyeDet [91] underlines the need for a useful dataset. They create a simulated fisheye dataset by applying distortions to the Pascal VOC dataset [92]. 4-sided polygon representation along with distortion shape matching helps FisheyeDet. The No-prior Fisheye Representation Method (NPFRM) was proposed to extract adaptive distortion features without using lens patterns and calibration patterns. Also, the Distortion Shape Matching (DSM) strategy was put forward to localize objects tightly and robustly in fisheye images. They use improper quadrilateral bounding boxes formed from the contour of distorted objects. An end-to-end network detector is created by combining it with the NPFRM and DSM.

SphereNet [93] and its variants [94], [95], [96] formulate CNNs on spherical surfaces and also explicitly encode invariances against the distortions. SphereNet accomplishes this by reversing distortions by adapting the sampling locations of the convolutional filters and wrapping them around the sphere. Existing perspective CNN models can be transferred to the omnidirectional scenario using SphereNet, which is modeled on normal convolutions. Moreover, quasi distortion in both horizontal and vertical directions indicates that fisheye images do not conform to spherical projection models. The outcomes of several detection algorithms that utilize equirectangular projection (ERP) sequences as direct input data were compared by Yang et al. [97], revealing that the CNN produces only a certain accuracy without projecting ERP sequences into normal 2D images.

FisheyeYOLO [14], [98] investigates various representations such as orientated bounding box, ellipse, and generic polygon. Using the intersection-over-union (IoU) metric and accurate instance segmentation ground truth, they compare these representations. They suggest a new curved bounding box approach with the best features for fisheye distortion camera models, as well as a curvature adaptive perimeter sampling methodology for generating polygon vertices that enhances relative mAP score by 4.9% over uniform sampling. Overall, the suggested polygon CNN model improves mean IoU relative accuracy by 40.3%.

*3) Soiling:* Surround view cameras are directly exposed to the external environment and is susceptible to soiling. For comparison, a front camera is placed behind the windshield, and it is less susceptible. This task was first formally defined in [99]. There are two types of soiled areas: opaque (mud, dust, snow) and transparent (water, oil, and grease) (water). Transparent soiling might be difficult to identify due to the limited visibility of the background. Soiling can cause significant degradation of perception accuracy, thus cleaning systems using a spray of water or more advanced ultrasonic based cleaning is employed for higher level of automated driving. Even if the camera is not cleaned, soiling detection is required to enhance the uncertainty of vision algorithms in degraded areas. As it is difficult to collect soiled data, DirtyGAN [100] proposed to use generative adversarial networks (GANs) to artificially generate different soiling patterns inpainted on real scenes. Boundaries of soiling are blurry and not well defined; thus, the manual annotation can be subjective and noisy. Das et al. [101] proposed tile level soiling classification to handled noisy annotations and to improve computational efficiency. Uricar et al. [102] proposed to use an ensemble based semi-supervised learning of pseudo labels to refine the noisy annotations automatically.

From a perception perspective, there are two ways to handle soiling. One way is to include robustness measures to improve the perception algorithm. For *e.g.*, Sakaridis et al. [103] proposed a foggy scene aware semantic segmentation. The other way is to restore the soiled region. Mud or water droplets are generally static or occasionally have low-frequency dynamics of moving water droplets. Thus, it is more effective to use video-based restoration techniques. Porav et al. [104] explored transparent soiling by using a stereo camera in conjunction with a dripping water source to mimic raindrops on the camera lens. This was done to automatically annotate the rainy pixels and they trained a CNN to restore the rainy regions. A desoiling dataset benchmark for surround view cameras was provided by Uricar et al. [105]. They use three cameras in proximity with various levels of soiling and a fourth camera with no soiling which acts as ground truth. They implemented a multi-frame baseline which can profit from the visibility of soiling occluded zones as time passes.

Sun glare detection is a closely related task of soiling. In manual and automatic driving, glare from the sun is a typical issue. Overexposure in the image is caused by sun glare, which substantially impacts visual perception algorithms. It is critical for higher levels of autonomous driving for the system to recognize that there is sun glare, which can degrade the system. The literature on detecting sun glare for automated driving is scarce. It is essentially based on image processing algorithms to detect saturated brightness areas and extract sections. A highly resilient algorithm is required from the perspective of a safety system. As a result, Yahiaoui et al. [106] created two complementary algorithms that use traditional image processing techniques and CNN to learn global context.

*4) Chargepad Assist:* Electric vehicles are becoming more widespread, and inductive chargepads are a practical and effective way to charge them. However, because drivers are usually poor at accurately aligning their vehicles for optimal inductive charging, a desirable solution would be to perform an automated alignment of the charging plates. The usage of surround-view cameras is ideal as it's a near-field perception task and implemented as an extension of the automated parking system. Dahal et al. [107] proposes a methodology premised on a surround-view camera framework that automatically identifies, localizes, and aligns the vehicle with the inductive chargepad. The visual design of the chargepads is not consistent and is often not recognized ahead of time. As a result, employing a system that depends on offline training would occasionally fail. Henceforth, they propose a self-supervised online learning technique that learns a classifier to auto-annotate the chargepad in the video sequences for further training by leveraging the driver's actions when manually aligning the car with the chargepad along with the weakly supervised semantic segmentation and depth predictions. When confronted with a previously undetected chargepad, the driver would have to align the car once manually as the chargepad lying on the ground is flat and is not easy to see and spot from afar. To achieve alignment from a more extensive range, they propose employing a Visual Simultaneous Localization and Mapping (SLAM) framework to learn landmarks relative to the chargepad.

*5) Trailer Assist:* Trailers are frequently used to move products and recreation equipment. Maneuvers with trailers, particularly reversing, can be tricky and unpleasant even for seasoned drivers. As a result, driver assistance systems come in handy in these situations. A single rear-view fisheye camera perception algorithm is usually used to achieve these. There is relatively little academic research on the subject because there is no publicly available dataset for this challenge. This prompted Dahal et al. [108] to detail all the trailer assist use cases and suggest a CNN-based solution to trailer perception issues. Using deep learning, they built a dataset for trailer detection and articulation angle estimation tasks. They developed and obtained high accuracy by detecting and tracking the trailer and its angle with an efficient CNN and long short-term memory (LSTM) model.

### B. Geometric Tasks

*1) Depth Estimation:* It involves estimating the distance to an object (or any plane) at a pixel level, as shown in Figure 14. Calculating distance relative to a camera plane is still very challenging. Currently, most of the works are on the rectified KITTI [109] sequences where barrel distortion is removed. In the case of a pinhole camera, depth is defined as the perpendicular distance from the camera plane, namely $z$. Previous structure-from-motion (SfM) approaches [110], [111], estimated inverse depth by parameterizing the network's disparity predictions into depth for the unprojection operation during the view synthesis step. This parameterization does not work well for fisheye cameras as they undergo large distortions which result in obtaining obtain angular disparities on the epipolar curves compared to the epipolar lines in the pinhole camera. To apply the same approach as pinhole, we would need to rectify the fisheye images which would result in a loss in field-of-view as described in Section III-F. However, the same multi-view geometry [112] principles that apply to pinhole projection model cameras also apply to fisheye images. By observing the scene from differing viewpoints and establishing correspondences between them, the underlying geometrical structure can be estimated. It is noteworthy to consider the CNN to output norm values than angular disparities for fisheye cameras when *SfM* approach is employed as it would make it difficult to parameterize the angular disparities to distance for the view synthesis operation. Furthermore, the value of $z$ can be (close to) zero or negative for field-of-views greater than 180°, which also leads to numerical issues as the models typically have some direct or indirect division by $z$ computation. Instead, it is useful to estimate the radial distance *i.e.* norm $\sqrt{x^2 + y^2 + z^2}$ instead of $z$. The norm is always positive and non-zero (except for $x, y, z = 0$) and allows a more numerical stable implementation.

On LiDAR distance measurements, such as KITTI, depth prediction models can be learned in a supervised manner. Ravi Kumar et al. [113] took a similar method, demonstrating the ability to predict distance maps employing LiDAR ground truth for training on fisheye images. Although, LiDAR data is very sparse and expensive to set up with good calibration. To overcome this problem, FisheyeDistanceNet [114] focused on solving one of the most challenging geometric

problems, *i.e.*, distance estimation on raw fisheye cameras using image-based reconstruction techniques, which is a challenging task, as the mapping between 2D images to 3D surfaces is an under-constrained problem. Depth estimation is also an ill-posed problem because there are several potential erroneous depths per pixel, which could also replicate the novel view. UnRectDepthNet [16] introduced a generic end-to-end self-supervised training framework for estimating monocular depth maps on raw distorted images for different camera models. The authors demonstrated the results of the framework work on raw KITTI and WoodScape datasets.

SynDistNet [115] learned semantic-aware geometric representations that could disambiguate photometric ambiguities in a self-supervised learning *SfM* context. They incorporated a generalized robust loss function [116], which significantly improved performance while eliminating the necessity of hyperparameter tuning with the photometric loss. They employed a semantic masking approach to reduce the artifacts due to the dynamic objects that violated static world assumptions. SynDistNet considerably enhanced the root mean squared error (RMSE) when compared to prior methods [16], [114] on fisheye images, reducing it by 25%. Most current depth estimation methodologies rely on a single camera, that cannot be seamlessly generalized to multiple fisheye cameras. Furthermore, the model must be implemented over several various-sized car lines with differing camera geometries. Even within a single-car line, intrinsics differ due to the manufacturing tolerances. Deep neural networks do seem to be sensitive to these changes, and training and testing each camera instance is nearly impossible. As a result, SVDistNet [117] proposed an innovative camera-geometry adaptive multi-scale convolutions that use the camera parameters as a conditional input, allowing the network to generalize to previously unknown fisheye cameras.

*2) Visual Odometry:* Liu et al. [118] describes a conventional direct visual odometry technique for a fisheye stereo camera. The technique does both camera motion estimation and semi-dense reconstruction at the same time. There are two threads in the pipeline: one for tracking and one for mapping. They estimate the camera posture using semi-dense direct image alignment in the tracking thread. To circumvent the epipolar curve problem, the plane-sweeping stereo algorithm is used for stereo matching and to initialize the depth. Cui et al. [119] demonstrated a large-scale, real-time dense geometric mapping technique using fisheye cameras. The camera poses were obtained from a global navigation satellite system/inertial navigation system (GNSS/INS) but they also propose that they can be retrieved from the visual-inertial odometry (VIO) framework. The depth map fusion uses the camera postures retrieved by these approaches. Heng et al. [120] described a semi-direct visual odometry algorithm for a fisheye stereo camera. In a tracking thread, they track-oriented patches while estimating camera poses; in a mapping thread, they estimate the coordinates and surface normal for every new patch to be tracked. Surface normal estimation allows us to track patches from distinct viewpoints. They do not employ descriptors or strong descriptors matching in their technique to detect patch correspondences. Instead, they employ photoconsistency-based approaches to find patch correspondences. Numerous visual odometry approaches for fisheye cameras, including [121] and [122], have recently been presented. In addition, Geppert et al. [121] used a multi-camera visual-inertial odometry framework to extend the visual-inertial localization technique for large-scale environments, resulting in a system that allows for accurate and drift-free pose estimation. Ravi Kumar et al. [123] employed CNNs for the visual odometry task, which acts as an auxiliary task in the monocular distance estimation framework.

*3) Motion Segmentation:* It is defined as the task of identifying the independently moving objects (pixels) such as vehicles and persons in a pair of sequences and separating them from the static background as shown in Figure 14. It is used as an appearance agnostic way to detect arbitrary moving objects using motion cues that are not common like rare animals (*e.g.*, kangaroo or a moose). It was first explored in MODNet [124] for autonomous driving. Recently, instance-level motion segmentation was defined and explored in InstanceMotSeg [125]. FisheyeMODNet [126] extends it to fisheye cameras without rectification. There was no explicit motion compensation, but it was mentioned as future work. Mariotti et al. [79] uses a classical approach to accomplishing this task based on vehicle odometry [127]. Spherical coordinate transformation of optical flow was performed and the positive height, depth, and epipolar constraints were adapted to work in this setup. They additionally propose anti-parallel constraint to remove motion parallax ambiguity which commonly occurs when a car is moving parallel to the ego-vehicle.

### C. Temporal Tasks

Although geometric tasks like depth and motion can use multiple frames for training and inference, the output is defined only on one frame. We define temporal tasks to be one whose output is defined on multiple frames. It typically requires multi-frame sequential annotation.

*1) Tracking:* Object tracking is the common temporal task where an object has to be associated across multiple frames. Detection and tracking of moving objects were explored in [128] for surround-view cameras. They use a classical optical flow-based approach for tracking. WEPDTOF [129] is a recently released dataset for pedestrian detection and tracking on fisheye cameras in an overhead surveillance setup. Although it is not an automotive dataset, it captures the challenges necessary for developing a tracking system on fisheye cameras. Trajectory prediction is closely related to tracking where the location of the object of interest must be predicted for the next set of frames. In the case of autonomous driving, it is particularly done in 3D bird's eye view space. PLOP algorithm [130] explored doing trajectory prediction of vehicles on a fisheye front camera after applying cylindrical rectification.

*2) Re-Identification:* Re-identification (Re-ID) is the association of detected objects across the cameras. It could also include association over time across cameras. Wu et al. [131] propose to perform vehicle Re-ID on the surround view cameras and highlight the two significant challenges: Firstly, due

to fisheye distortion, occlusion, truncation, and other factors, it is difficult to detect the same vehicle from previous image frames in a single-camera view. Secondly, in a multi-camera perspective, the appearance of the identical vehicle changes dramatically depending on which camera is used. They offer a new quality evaluation mechanism to counteract the effects of tracking box drift and target consistency. They employ a Re-ID network based on an attention mechanism, which is then paired with a spatial constraint method to improve the performance of diverse cameras.

Zhao et al. [132] proposes a pedestrian Re-ID algorithm. It consists of a single camera detection and tracking module and a two-camera ReID module applied to multi-camera views. It includes a single camera detection and tracking module as well as a two-camera ReID module for multi-camera views. Using a YOLOv3 [133], the detection module recognizes pedestrians in single camera view videos. To track pedestrians and issue an ID to each identified pedestrian, the tracking model integrates OSnet [134] with DeepSORT [135]. Both models were adapted to the fisheye images using transfer learning procedures.

*3) SLAM:* Feature correspondence comprises keypoint detection, description, and matching and it is the primary step in SLAM systems. FisheyeSuperPoint [136] introduces a unique training and evaluation methodology for fisheye images. As a starting point, they employ SuperPoint [137], a self-supervised keypoint detector and descriptor that has generated state-of-the-art homography prediction results. They present a fisheye adaption framework for training on undistorted fisheye images; fisheye warping is employed for self-supervised training on fisheye images. Through an intermediary projection phase to a unit sphere, the fisheye image is translated to a new, distorted image. The virtual posture of the camera can be changed in 6-Dof. Tripathi et al. [138] where they explored the problem of relocalization using surround-view fisheye cameras using an ORB SLAM pipeline. The goal was to perform mapping of a private area like an apartment complex and relocalize with respect to this map to assist automated parking. Feature detection was performed on raw fisheye images, and a comparison of different feature correspondence algorithms on raw fisheye cameras was analyzed.

### D. Mulitask Models

Multi-task learning (MTL) is carried out by learning commonly shared representations from multi-task supervisory signals. Since the introduction of deep learning, many dense prediction tasks, *i.e.*, tasks that generate pixel-level predictions, have witnessed significant performance increases. These tasks are typically learned one at a time, with each task requiring the training of its own neural network. Recent MTL approaches [149], [150], on the other hand, have shown promising outcomes in terms of performance, computational complexity, and memory footprint by jointly handling many tasks via a learned shared representation.

For fisheye cameras, Sistu et al. [151] proposed a joint MTL model for learning object detection and semantic segmentation. The primary objective was to achieve real-time performance on a low-power embedded system on a chip using the same encoder for both tasks. They use a simple ResNet10-like encoder shared by both decoders to build an efficient architecture. Object detection employs the YOLO v2 decoder, whereas semantic segmentation employs the FCN8 decoder. Leang et al. explored different task weighting methods for the two-task setup on fisheye cameras [152]. FisheyeMultiNet [153] discusses the design and implementation of an automated parking system from the perspective of camera-based deep learning algorithms. On a low-power embedded system, FisheyeMultiNet is a real-time multi-task deep learning network that recognizes all the necessary objects for parking. The setup is a four-camera system that runs at 15fps and performs three tasks: object detection, semantic segmentation, and soiling detection.

Finally, a holistic real-time scene understanding for the near-field perception of the environment *using cameras only* was presented in OmniDet [123]. They build a near-field perception system that constitutes a *Level* 3 autonomous stack as shown in Figure 14. With this framework's help, we can jointly understand and reason about geometry, semantics, motion, localization, and soiling from a single deep learning model comprising of six tasks at 60fps on embedded systems. Motivated by Rashed et al. [154] who demonstrated that the geometric tasks like depth and motion can aid semantic segmentation, synergized cross links across tasks were implemented. Camera calibration was converted to a pixel-wise tensor and fed into the model such that it can adapt to various camera intrinsics. Sobh et al. [155] studied the effect of adversarial attacks in a multi-task setup using OmniDet, which is important for safety-critical applications. The tests addressed both white and black box attacks for targeted and untargeted cases and the effect of using a simple defense strategy while attacking a task and analyzing the effect on the others.

## V. Public Datasets and Research Directions

### A. Datasets

Building an automotive dataset is costly and time consuming [156], it is currently the main bottleneck in progressing research in fisheye perception. In Table II, we summarize the published fisheye camera datasets. WoodScape is a comprehensive dataset for 360° sensing around ego vehicle using four fisheye cameras. It is intended to complement the current automotive datasets where only narrow FOV images are available. Of these, KITTI [157] was the groundbreaking dataset with different types of tasks. This is the first comprehensive fisheye automotive dataset to evaluate computer vision algorithms like fisheye image segmentation, object detection, and motion segmentation in detail [158]. The synthetic variant of the surround-view dataset Woodscape is SynWoodScape [139]. Many of its flaws are covered and extended. The authors of WoodScape were unable to collect ground truth for pixel-level optical flow and depth because all four cameras were not available at the same time to sample different frames. This means that multicamera algorithms, which are conceivable in SynWoodScape, cannot be implemented in WoodScape.

TABLE II
SUMMARY OF VARIOUS AUTONOMOUS DRIVING DATASETS CONTAINING FISHEYE IMAGES

| Datasets | *Automotive* | *Real* | *Synthetic* | *numFisheyeCameras* | *Year* | *Resolution* | *Tasks* |
|---|---|---|---|---|---|---|---|
| WoodScape [2] | ✓ | ✓ | ✗ | 4 | 2021 | 1280×966 | Semantic/Instance Seg.<br>Motion Seg.<br>2D Bounding Boxes |
| SynWoodScape [139] | ✓ | ✗ | ✓ | 4 | 2022 | 1280×966 | Semantic/Instance Seg.<br>Motion Seg.<br>2D/3D Bounding Boxes<br>Depth/Flow Estimation<br>Event Camera<br>LiDAR/IMU/GNSS |
| KITTI 360 [140] | ✓ | ✓ | ✗ | 2 | 2021 | 1400×1400 | 2D/3D Semantic Seg.<br>2D/3D Instance Seg.<br>3D Bounding Boxes |
| FisheyeCityScapes [88] | ✓ | ✓ | ✗ | 1 | 2020 | 600×600 | Semantic Seg. |
| Oxford Robot Car [141] | ✓ | ✓ | ✗ | 3 | 2016 | 1024×1024 | LiDAR/IMU/GNSS |
| THEODORE [142] | ✗ | ✗ | ✓ | 1 | 2020 | 1024×1024 | Semantic/Instance Seg.<br>2D Bounding Boxes |
| OmniScape [143] | ✗ | ✗ | ✓ | 2 | 2020 | 1024×1024 | Semantic/Instance Seg.<br>2D/3D Bounding Boxes<br>Depth Estimation<br>LiDAR/IMU/GNSS |
| PIROPO [144] | ✗ | ✓ | ✗ | 3 | 2020 | 800 × 600 | People Indoor Localization |
| Go Stanford [145] | ✗ | ✓ | ✗ | 2 | 2018 | 128×128 | Traversability Estimation |
| Mo2Cap2 [146] | ✗ | ✓ | ✗ | 1 | 2018 | 256×256 | Human Pose Estimation |
| LMS Fisheye [147] | ✗ | ✓ | ✓ | 1 | 2016 | 1088 × 1088 | Motion Estimation |
| EgoCap [148] | ✗ | ✓ | ✗ | 2 | 2016 | 1280×1024 | Human Tracking<br>Human Pose Estimation |
| LSD SLAM [122] | ✗ | ✓ | ✗ | 1 | 2015 | 640×480 | Tracking/Mapping |

It contains 80k images with annotations from the synthetic dataset.

KITTI 360° is a suburban dataset with a broader input modality, extensive semantic instance annotations, and precise localization to aid study in the visual, computing, and robotics fields. Compared to WoodScape, KITTI 360° differs in that it provides temporally coherent semantic instance annotations, 3D laser scans, and 3D annotations for inference in perspective and omnidirectional images. FisheyeCityScapes [88] has proposed a seven-DoF extension, which is a virtual Fisheye data augmentation method. This method uses a radial distortion model to convert a rectilinear dataset to a fisheye dataset. It synthesizes fisheye images captured by the camera in various orientations, positions, and $f$ values, greatly improving the generalized performance of fisheye semantic segmentation. Oxford RobotCar [141] is a large-scale dataset focused on autonomous vehicles' long-term autonomy. Localization and mapping are the primary tasks of this dataset, which enables study into continuous learning for autonomous vehicles and mobile robotics.

THEODORE [142] is a large non-automotive synthetic dataset for indoor scenes containing 100,000 high resolution and 16 classes of diverse fisheye images in top-view. To achieve this, they create a 3D virtual environment of the living room, various human characters, and interior textures. The authors construct annotations for semantic segmentation, instance masks, and bounding boxes for object detection in addition to recording fisheye images from virtual environments. OmniScape [143] dataset contains two front fisheye and catadioptric stereo RGB images mounted on the motorcycle with semantic segmentation, depth sequences, and vehicle dynamics captured by velocity, angular velocity, acceleration, and orientation. It also contains over 10,000 frames and data recorded by GTA V and CARLA that could also be extended to other simulators. Sequences were recorded in two distinct rooms utilizing both omnidirectional and perspective cameras for the PIROPO [144] (People in Indoor Rooms with Perspective and Omnidirectional cameras). The sequence depicts people in various positions, such as walking, standing, and sitting. Ground truth is point-based, and both annotated and non-annotated sequences are provided (each person in the scene is represented by a point in the center of the head). Over 100,000 annotated frames are accessible in total.

The Go Stanford [145] dataset consists of about 24 hours of video from over 25 indoor environments. The experiment focuses on estimating traversability indoors using fisheye images. The Mo2Cap2 [146] dataset is used to estimate egocentric 3D human poses in a variety of unconstrained daily activities. This dataset aims to answer the challenge of mobile 3D posture estimation in a variety of activities such as walking, cycling, cooking, sports, and office work that take place in unrestricted real-world scenarios. Sports, animation, healthcare action recognition, motion control, and performance analysis can all benefit from these 3D postures. LMS Fisheye [147] dataset aims to provide researchers with video sequences for developing and testing motion estimation algorithms developed for the fisheye camera. Both the synthetic sequence generated by Blender and the actual sequence recorded by the fisheye camera are provided.

EgoCap [148] is a markerless, egocentric, real-time motion capture dataset for full-body skeletal pose estimation from a lightweight stereo pair fisheye camera mounted on a helmet

or virtual reality headset — optical inside-in method. The LSD-SLAM [122] dataset is derived from a new real-time monocular SLAM approach. It is completely direct (*i.e.*, it does not use key points/features) and creates large-scale, semi-dense maps in real-time on a laptop. Researchers can use this dataset to work on tracking (direct image alignment) and mapping. (Pixel-wise Distance Filtering) directly enables a unified omnidirectional model capable of modeling a central imaging device with an FoV of more than 180°.

### B. Research Directions

*1) Distortion Aware CNNs:* CNNs naturally exploit the translation invariance in the image grid, and it is broken in fisheye images due to spatially variant distortion. Spherical CNNs [93], [159] have been proposed, which can be directly used for spherical radial distortion models. However, automotive lenses are more complex, and the spherical model is unsuitable. It would be an interesting direction to generalize Spherical CNNs to a more complex fisheye manifold surface. Kernel transformer networks [95] efficiently transfer convolution operators from perspective to equirectangular projections of an omnidirectional image, and it is more suitable to generalize to a fisheye image.

*2) Handling Temporal Variation:* As we discussed before, the sample complexity of an object detector is increased for a fisheye camera due to larger variability in appearance due to radial distortion. This is further exacerbated for temporal tasks, which require matching features across two frames, which could have two different distortions. For example, object tracking and reidentification are significantly more challenging in the case of fisheye cameras. Tracking a pedestrian moving from left to right of a static camera would require handling large radial distorted appearance variation. Similarly, for a static pedestrian, the horizontal and vertical motion of the camera causes large variations. It is also a challenge for the point feature correspondence problem, like tracking. One solution could be to explicitly embed the radial distortion in the feature vector, which can be leveraged for matching.

*3) Bird-eye's View Perception:* In automated driving, it is essential to lift the detections on the image to 3D. It is typically achieved by inverse perspective mapping (IPM) [160], assuming a flat ground surface. It can also be enhanced by using depth estimation or fusion with 3D sensors [161]. There is a recent trend of outputting directly in 3D using the IPM implicitly in the network [162], [163]. It is typically achieved by transforming the abstract encoder features using a learnable rectification layer as an alternative to performing IPM at the input level. As CNNs have more context information and a learnable transformation can be more flexible, it works better than a pixel-wise IPM [163]. In the case of pinhole cameras, IPM is a linear transform, and it is relatively easy to design the spatial transformer of encoder features. However, for fisheye cameras, IPM is a complex non-linear operator, and it remains an open problem to directly output in bird's eye view space.

*4) Multi-Camera Modeling:* Most of the current work in surround-view cameras treats each of the four cameras independently and performs perception algorithms. It might be more optimal to model all four surround-view cameras jointly. Firstly, it will aid detection of large vehicles (*e.g.*, transportation trucks) visible across two or three cameras (front, left, and rear). Secondly, it eliminates the re-identification of objects seen in multiple cameras (see Figure 10) and post-processing of individual detections to form a unified output like the lane model. A multi-camera model would more efficiently aggregate information and produce more optimal outputs. [164] developed a classical geometric approach of treating multiple cameras as single cameras. However, there is some recent work that makes use of multiple cameras as input to a single perception model [163], [165]. They make use of pinhole cameras with minimal overlapping field-of-view. It is significantly more challenging to model this for surround-view cameras.

Fig. 15. Illustration of near and far-field front camera images forming an asymmetric stereo pair.

*5) Unified Modeling of Near and Far-Field Cameras:* A typical configuration for a next-generation automated driving system comprises full 360° coverage of near-field using four surround-view cameras and six far-field cameras (one front, one rear, two on each side) [166]. As discussed in §III-B, they have drastically different fields-of-view and ranges. Thus, it is challenging to perform unified modeling of all the cameras extending the multi-camera modeling discussed above. Figure 15 illustrates the near and far-field images of the front region. They form an asymmetric stereo pair where depth could be easily computed instead of the more challenging monocular depth, which has fundamental ambiguities. Currently, there are no public datasets containing both near and far-field cameras to enable this research.

## VI. Conclusion

Fisheye cameras are one of the most common sensors in autonomous driving systems. Despite its prevalence, there is limited understanding of it in the automotive community as it's a specialized camera sensor, and standard algorithms do not generalize to it. This work provided a detailed account of getting started with surround-view fisheye camera development. The paper is part tutorial describing the fisheye geometry and models in detail and part survey discussing the perception algorithms developed on fisheye. We finally provide future directions to be explored.

## Acknowledgment

The authors would like to thank Balaji Sankar Balachandaran, Jacob Roll, and Louis Kerofsky from Qualcomm for providing detailed review comments.

## REFERENCES


[1] R. Wood, "Fish-eye views, and vision under water," *Phil. Mag.*, vol. 12, no. 6, pp. 159–162, 1908.

[2] S. Yogamani et al., "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9308–9318.

[3] W. N. Bond, "A wide angle lens for cloud recording," *Phil. Mag.*, vol. 44, no. 263, pp. 999–1001, 1922.

[4] K. Miyamoto, "Fish eye lens," *J. Opt. Soc. Amer.*, vol. 54, no. 8, pp. 1060–1061, Aug. 1964.

[5] P. D. Thomas, *Conformal Projections in Geodesy and Cartography*. Washington, DC, USA: Government Printing Office, 1952.

[6] (Dec. 2019). *How the Fisheye Lens Took Over Music*. [Online]. Available: https://youtu.be/YNQzqEAphjs

[7] H. Kim, J. Jung, and J. Paik, "Fisheye lens camera based surveillance system for wide field of view monitoring," *Optik*, vol. 127, no. 14, pp. 5636–5646, Jul. 2016.

[8] T. A. DeFanti et al., "The StarCAVE, a third-generation CAVE and virtual reality OptIPortal," *Future Gener. Comput. Syst.*, vol. 25, no. 2, pp. 169–178, Feb. 2009.

[9] L. Meng, T. Hirayama, and S. Oyanagi, "Underwater-drone with panoramic camera for automatic fish recognition based on deep learning," *IEEE Access*, vol. 6, pp. 17880–17886, 2018.

[10] K. Qiu, T. Liu, and S. Shen, "Model-based global localization for aerial robots using edge alignment," in *Proc. Int. Conf. Robot. Autom.*, vol. 2, 2017, pp. 1256–1263.

[11] C. R. Sunstein, "Rear visibility and some unresolved problems for economic analysis (with notes on experience goods)," *J. Benefit-Cost Anal.*, vol. 10, no. 3, pp. 317–350, 2019.

[12] C. Hughes, M. Glavin, E. Jones, and P. Denny, "Wide-angle camera technology for automotive applications: A review," *IET Intell. Transp. Syst.*, vol. 3, no. 1, pp. 19–31, Mar. 2009.

[13] M. Bandyopadhyay, A. Kumar, and A. Pathayapurakkal, "Fish-eye image based cross traffic alert system," in *Computer Vision, Pattern Recognition, Image Processing, and Graphics*, R. V. Babu, M. Prasanna, and V. P. Namboodiri, Eds. Singapore: Springer, 2020, pp. 43–52.

[14] H. Rashed et al., "Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2272–2280.

[15] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Computer vision in automated parking systems: Design, implementation and challenges," *Image Vis. Comput.*, vol. 68, pp. 88–101, Dec. 2017.

[16] V. R. Kumar, S. Yogamani, M. Bach, C. Witt, S. Milz, and P. Mader, "UnRectDepthNet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 8177–8183.

[17] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proc. CVPR*, vol. 1, 2001, pp. 125–132.

[18] A. Basu and S. Licardie, "Alternative models for fish-eye lenses," *Pattern Recognit. Lett.*, vol. 16, no. 4, pp. 433–441, 1995.

[19] F. Devernay and O. Faugeras, "Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured enviroments," *Mach. Vis. Appl.*, vol. 13, no. 1, pp. 14–24, 2001.

[20] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.

[21] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical applications," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 445–461.

[22] B. Khomutenko, G. Garcia, and P. Martinet, "An enhanced unified camera model," *IEEE Robot. Automat. Lett.*, vol. 1, no. 1, pp. 137–144, Jan. 2016.

[23] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 552–560.

[24] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 1039–1044.

[25] J. K. Suhr and H. G. Jung, "Sensor fusion-based vacant parking slot detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 21–36, Feb. 2014.

[26] X. Tang, Z. Zhang, and Y. Qin, "On-road object detection and tracking based on radar and vision fusion: A review," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 5, pp. 103–128, Sep./Oct. 2022.

[27] A. Loeffler, J. Ronczka, and T. Fechner, "Parking lot measurement with 24 GHz short range automotive radar," in *Proc. 16th Int. Radar Symp. (IRS)*, Jun. 2015, pp. 137–142.

[28] M. R. Schmid, S. Ates, J. Dickmann, F. Von Hundelshausen, and H.-J. Wuensche, "Parking space detection with hierarchical dynamic occupancy grids," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 254–259.

[29] V. Lekic and Z. Babic, "Automotive radar and camera fusion using generative adversarial networks," *Comput. Vis. Image Understand.*, vol. 184, pp. 1–8, Jul. 2019.

[30] R. Varga, A. Costea, H. Florea, I. Giosan, and S. Nedevschi, "Super-sensor for 360-degree environment perception: Point cloud segmentation using image features," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[31] C. Eising, J. Horgan, and S. Yogamani, "Near-field perception for low-speed vehicle automation using surround-view fisheye cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 13976–13993, Sep. 2022.

[32] J. Malik et al., "The three R's of computer vision: Recognition, reconstruction and reorganization," *Pattern Recognit. Lett.*, vol. 72, pp. 4–14, Mar. 2016.

[33] D. Kim, J. Park, and J. Paik, "Extended fisheye lens model for practical geometric correction and image enhancement," *Opt. Lett.*, vol. 39, no. 21, pp. 6261–6264, 2014.

[34] C. Hughes, P. Denny, M. Glavin, and E. Jones, "Equidistant fish-eye calibration and rectification by vanishing point extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2289–2296, Dec. 2010.

[35] D. C. Brown, "Alternative models for fish-eye lenses," *Photogramm. Eng.*, vol. 32, no. 2, pp. 444–462, 1966.

[36] A. E. Conrady, "Decentred lens-systems," *Monthly Notices Roy. Astronomical Soc.*, vol. 79, no. 5, pp. 384–390, Mar. 1919.

[37] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–123, 2000.

[38] *MATLAB, (R2021a)*. The MathWorks Inc., Natick, MA, USA, 2021.

[39] X. Ying, Z. Hu, and H. Zha, "Fisheye lenses calibration using straight-line spherical perspective projection constraint," in *Proc. Asian Conf. Comput. Vis.*, 2006, pp. 61–70.

[40] The Mathworks. (2020). *MATLAB R2020b: What Is Camera Calibration?* Accessed: Dec. 16, 2020. [Online]. Available: https://www.mathworks.com/help/vision/ug/camera-calibration.html

[41] NVidia. (2020). *DriveWrorks 3.0 SDK: Camera Model*. Accessed: Apr. 22, 2022. [Online]. Available: https://docs.nvidia.com/drive/archive/driveworks-3.0/group__cameramodel__group.html

[42] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proc. 4th IEEE Int. Conf. Comput. Vis. Syst. (ICVS)*, Jan. 2006, p. 45.

[43] H. Wildenauer and B. Micusík, "Closed form solution for radial distortion estimation from a single vanishing point," in *Proc. Brit. Mach. Vis. Conf.*, 2013, p. 2.

[44] M. Antunes, J. P. Barreto, D. Aouada, and B. Ottersten, "Unsupervised vanishing point detection and camera calibration from a single Manhattan image with radial distortion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4288–4296.

[45] F. Bukhari and M. N. Dailey, "Automatic radial distortion estimation from a single image," *J. Math. Imag. Vis.*, vol. 45, no. 1, pp. 31–45, 2013.

[46] J. Courbon, Y. Mezouar, and P. Martinet, "Evaluation of the unified model of the sphere for fisheye cameras in robotic applications," *Adv. Robot.*, vol. 26, nos. 8–9, pp. 947–967, May 2012.

[47] X. Ying and Z. Hu, "Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2004, pp. 442–455.

[48] J. Courbon, Y. Mezouar, L. Eckt, and P. Martinet, "A generic fisheye camera model for robotic applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 1683–1688.

[49] H. Bakstein and T. Pajdla, "Panoramic mosaicing with a 180° field of view lens," in *Proc. IEEE Workshop Omnidirectional Vis. Held Conjunct (ECCV)*, Jun. 2002, pp. 60–67.

[50] J. Perš and S. Kovacic, "Nonparametric, model-based radial lens distortion correction using tilted camera assumption," in *Proc. Comput. Vis. Winter Workshop*, vol. 1, 2002, pp. pp–286.

[51] G. Klančar, M. Kristan, and R. Karba, "Wide-angle camera distortions and non-uniform illumination in mobile robot tracking," *Robot. Auto. Syst.*, vol. 46, no. 2, pp. 125–133, Feb. 2004.

[52] X. Mei, S. Yang, J. Rong, X. Ying, S. Huang, and H. Zha, "Radial lens distortion correction using cascaded one-parameter division model," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3615–3619.

[53] A. L. Laubscher, "A basic investigation of perspective map projections," thesis, Dept. Geodetic Sci., Ohio State Univ., Columbus, OH, USA, 1965.

[54] J. P. Snyder, *Map Projections: A Working Manual*, vol. 1395. Washington, DC, USA: US Geological Survey Professional Paper, 1987.

[55] C. Hughes, P. Denny, E. Jones, and M. Glavin, "Accuracy of fish-eye lens models," *Appl. Opt.*, vol. 49, no. 17, pp. 3338–3347, Jun. 2010.

[56] C. Calefato et al., "The HMI of preventing warning systems: The DESERVE approach," in *Towards a Common Software/Hardware Methodology for Future Advanced Driver Assistance Systems: The DESERVE Approach*, G. Payá-Vayá and H. Blume, Eds. Denmark: River Publishers, 2017, ch. 10, pp. 227–250.

[57] S. Dabral, S. Kamath, V. Appia, M. Mody, B. Zhang, and U. Batur, "Trends in camera based automotive driver assistance systems (ADAS)," in *Proc. IEEE 57th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2014, pp. 1110–1115.

[58] M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*. Reading, MA, USA: Addison-Wesley Longman Publishing, 1999.

[59] L. Yahiaoui et al., "Optimization of ISP parameters for object detection algorithms," *Electron. Imag.*, vol. 2019, no. 15, pp. 1–44, Jan. 2019.

[60] L. Yahiaoui, J. Horgan, B. Deegan, S. Yogamani, C. Hughes, and P. Denny, "Overview and empirical analysis of ISP parameter tuning for visual perception in autonomous driving," *J. Imag.*, vol. 5, no. 10, p. 78, Sep. 2019.

[61] V. Zlokolica, B. Deegan, P. Denny, M. Griffin, and B. Dever, "Free-view multi-camera visualization and harmonization for automotive systems," *Electron. Imag.*, vol. 29, no. 19, pp. 12–17, Jan. 2017.

[62] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Sep. 1999, pp. 666–673.

[63] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 1997, pp. 1106–1112.

[64] A. Duda and U. Frese, "Accurate detection and localization of checkerboard corners for calibration," in *Proc. BMVC*, 2018, p. 126.

[65] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 5695–5701.

[66] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3121–3126.

[67] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending *kalibr*: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4304–4311.

[68] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1280–1286.

[69] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2088–2095.

[70] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 473–480.

[71] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1793–1800.

[72] X. Shao, X. Liu, L. Zhang, S. Zhao, Y. Shen, and Y. Yang, "Revisit surround-view camera system calibration," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1486–1491.

[73] X. Liu et al., "Online camera pose optimization for the surround-view system," in *Proc. 27th ACM Int. Conf. Multimedia*, New York, NY, USA, 2019, pp. 383–391.

[74] K. Choi, H. Jung, and J. Suhr, "Automatic calibration of an around view monitor system exploiting lane markings," *Sensors*, vol. 18, no. 9, p. 2956, Sep. 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/9/2956

[75] Z. Ouyang, L. Hu, Y. Lu, Z. Wang, X. Peng, and L. Kneip, "Online calibration of exterior orientations of a vehicle-mounted surround-view camera system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4990–4996.

[76] M. Friel, C. Hughes, P. Denny, E. Jones, and M. Glavin, "Automatic calibration of fish-eye cameras from automotive video sequences," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 136–148, 2010.

[77] N. Aggarwal and W. C. Karl, "Line detection in images through regularized Hough transform," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 582–591, Mar. 2006.

[78] W. Förstner and B. P. Wrobel, *Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction*, 1st ed. Cham, Switzerland: Springer, 2016.

[79] L. Mariotti and C. Eising, "Spherical formulation of geometric motion segmentation constraints in fisheye cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4201–4211, May 2022.

[80] M. Lourenço, J. P. Barreto, and F. Vasconcelos, "SRD-SIFT: Keypoint detection and matching in images with radial distortion," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 752–760, Jun. 2012.

[81] E. Plaut, E. B. Yaacov, and B. El Shlomo, "3D object detection from a single fisheye image without a single fisheye training image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3659–3667.

[82] A. Das, S. Kandan, S. Yogamani, and P. Křížek, "Design of real-time semantic segmentation decoder for automated driving," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 1–8.

[83] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, "CNN based semantic segmentation for urban traffic scenes using fisheye camera," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 231–236.

[84] Á. Sáez et al., "Real-time semantic segmentation for fisheye urban driving images based on ERFNet," *Sensors*, vol. 19, no. 1, p. 503, 2019.

[85] E. Romera et al., "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Oct. 2017.

[86] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. Comput. Vis. Pattern Recognit. Conf.*, 2016, pp. 3213–3223.

[87] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang, "Restricted deformable convolution-based road scene semantic segmentation using surround view cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4350–4362, Oct. 2020.

[88] Y. Ye, K. Yang, K. Xiang, J. Wang, and K. Wang, "Universal semantic segmentation for fisheye urban driving images," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 648–655.

[89] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3234–3243.

[90] C. Playout, O. Ahmad, F. Lecue, and F. Cheriet, "Adaptable deformable convolutions for semantic segmentation of fisheye images in autonomous driving systems," 2021, *arXiv:2102.10191*.

[91] T. Li, G. Tong, H. Tang, B. Li, and B. Chen, "FisheyeDet: A self-study and contour-based object detector in fisheye images," *IEEE Access*, vol. 8, pp. 71739–71751, 2020.

[92] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[93] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 518–533.

[94] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications," *Astron. Comput.*, vol. 27, pp. 130–146, Apr. 2019.

[95] Y.-C. Su and K. Grauman, "Kernel transformer networks for compact spherical convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9442–9451.

[96] C. Jiang et al., "Spherical CNNs on unstructured grids," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–16.

[97] W. Yang, Y. Qian, J.-K. Kamarainen, F. Cricri, and L. Fan, "Object detection in equirectangular panorama," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2190–2195.

[98] H. Rashed et al., "FisheyeYOLO: Object detection on fisheye cameras for autonomous driving," in *Proc. Mach. Learn. Auto. Driving NeurIPSW*, 2020, pp. 1–5.

[99] M. Uřičář, P. Křížek, G. Sistu, and S. Yogamani, "SoilingNet: Soiling detection on automotive surround-view cameras," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 67–72.

[100] M. Uřičář et al., "Let's get dirty: GAN based data augmentation for camera lens soiling detection in autonomous driving," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 766–775.

[101] A. Das et al., "TiledSoilingNet: Tile-level soiling detection on automotive surround-view cameras using coverage metric," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[102] M. Uřičář, G. Sistu, L. Yahiaoui, and S. Yogamani, "Ensemble-based semi-supervised learning to improve noisy soiling annotations in autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2925–2930.

[103] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 973–992, Sep. 2018.

[104] H. Porav, T. Bruls, and P. Newman, "I can see clearly now: Image restoration via de-raining," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7087–7093.

[105] M. Uřičář et al., "Desoiling dataset: Restoring soiled areas on automotive fisheye cameras," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4273–4279.

[106] L. Yahiaoui, M. Uřičář, A. Das, and S. Yogamani, "Let the sunshine in: Sun glare detection on automotive surround-view cameras," *Electron. Imag.*, vol. 2020, no. 1, pp. 1–80, 2020.

[107] A. Dahal, V. R. Kumar, S. Yogamani, and C. Eising, "An online learning system for wireless charging alignment using surround-view fisheye cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20553–20562, Nov. 2022.

[108] A. Dahal et al., "DeepTrailerAssist: Deep learning based trailer detection, tracking and articulation angle estimation on automotive rear-view camera," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2339–2346.

[109] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[110] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[111] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3828–3838.

[112] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, vol. 2. Cambridge, U.K.: Cambridge Univ., 2003.

[113] V. R. Kumar et al., "Near-field depth estimation using monocular fisheye camera: A semi-supervised learning approach using sparse LiDAR data," in *Proc. Comput. Vis. Pattern Recognit. Conf. Workshops*, vol. 7, 2018, p. 2.

[114] V. R. Kumar et al., "FisheyeDistanceNet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 574–581.

[115] V. R. Kumar, M. Klingner, S. Yogamani, S. Milz, T. Fingscheidt, and P. Mader, "SynDistNet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 61–71.

[116] J. T. Barron, "A general and adaptive robust loss function," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4331–4339.

[117] V. R. Kumar et al., "SVDistNet: Self-supervised near-field distance estimation on surround view fisheye cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10252–10261, Aug. 2022.

[118] P. Liu, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Direct visual odometry for a fisheye-stereo camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1746–1752.

[119] Z. Cui, L. Heng, Y. C. Yeo, A. Geiger, M. Pollefeys, and T. Sattler, "Real-time dense mapping for self-driving vehicles using fisheye cameras," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6087–6093.

[120] L. Heng and B. Choi, "Semi-direct visual odometry for a fisheye-stereo camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4077–4084.

[121] M. Geppert, P. Liu, Z. Cui, M. Pollefeys, and T. Sattler, "Efficient 2D–3D matching for multi-camera visual localization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5972–5978.

[122] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 141–148.

[123] V. R. Kumar et al., "Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2830–2837, Feb. 2021.

[124] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, "MODNet: Motion and appearance based moving object detection network for autonomous driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2859–2864.

[125] E. Mohamed et al., "Monocular instance motion segmentation for autonomous driving: KITTI InstanceMotSeg dataset and multi-task baseline," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 114–121.

[126] M. Yahiaoui et al., "FisheyeMODNet: Moving object detection on surround-view cameras for autonomous driving," in *Proc. Irish Mach. Vis. Image Process.*, 2019, pp. 1–4.

[127] C. Eising, L. Pereira, J. Horgan, A. Selvaraju, J. McDonald, and P. Moran, "2.5 D vehicle odometry estimation," *IET Intell. Transp. Syst.*, vol. 16, no. 3, pp. 292–308, Mar. 2022.

[128] I. Baek, A. Davies, G. Yan, and R. R. Rajkumar, "Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 447–452.

[129] M. O. Tezcan, Z. Duan, M. Cokbas, P. Ishwar, and J. Konrad, "WEPDTOF: A dataset and benchmark algorithms for in-the-wild people detection and tracking from overhead fisheye cameras," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 503–512.

[130] T. Buhet et al., "PLOP: Probabilistic polynomial objects trajectory prediction for autonomous driving," in *Proc. Conf. Robot Learn.* 2021, pp. 329–338.

[131] Z. Wu, M. Wang, L. Yin, W. Sun, J. Wang, and H. Wu, "Vehicle re-ID for surround-view camera system," in *Proc. CVPR Workshop Scalability Auto. Driving*, 2020, pp. 1–8.

[132] Z. Zhao, Z. Zhao, S. Wang, P. Watta, and Y. Lu Murphey, "Pedestrian re-identification using a surround-view fisheye camera system," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9533301.

[133] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[134] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3702–3712.

[135] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[136] A. Konrad, C. Eising, G. Sistu, J. McDonald, R. Villing, and S. Yogamani, "FisheyeSuperPoint: Keypoint detection and description network for fisheye images," in *Proc. 17th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2022, pp. 1–8.

[137] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 224–236.

[138] N. Tripathi and S. Yogamani, "Trained trajectory based automated parking system using Visual SLAM," in *Proc. Comput. Vis. Pattern Recognit. Conf. Workshops*, 2021, pp. 1–8.

[139] A. R. Sekkat et al., "SynWoodScape: Synthetic surround-view fisheye camera dataset for autonomous driving," 2022, *arXiv:2203.05056*.

[140] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," 2021, *arXiv:2109.13410*.

[141] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *IJ Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2016.

[142] T. Scheck, R. Seidel, and G. Hirtz, "Learning from THEODORE: A synthetic omnidirectional top-view indoor dataset for deep transfer learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 943–952.

[143] A. R. Sekkat, Y. Dupuis, P. Vasseur, and P. Honeine, "The OmniScape dataset," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1603–1608.
[144] C. R. Del-Blanco, P. Carballeira, F. Jaureguizar, and N. García, "Robust people indoor localization with omnidirectional cameras using a grid of spatial-aware classifiers," *Signal Process., Image Commun.*, vol. 93, Apr. 2021, Art. no. 116135.
[145] N. Hirose, A. Sadeghian, M. Vazquez, P. Goebel, and S. Savarese, "GONet: A semi-supervised deep learning approach for traversability estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3044–3051.
[146] W. Xu et al., "$Mo^2Cap^2$: Real-time *Mobile* 3D *Motion Capture* with a *Cap*-mounted fisheye camera," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 5, pp. 2093–2101, May 2019.
[147] A. Eichenseer and A. Kaup, "A data set providing synthetic and real-world fisheye video sequences," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1541–1545.
[148] H. Rhodin et al., "EgoCap: Egocentric marker-less motion capture with two fisheye cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–11, Nov. 2016.
[149] G. Sistu et al., "NeurAll: Towards a unified visual perception model for automated driving," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 796–803.
[150] S. Chennupati, G. Sistu, S. Yogamani, and S. Rawashdeh, "AuxNet: Auxiliary tasks enhanced semantic segmentation for automated driving," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 645–652.
[151] G. Sistu, I. Leang, and S. Yogamani, "Real-time joint object detection and semantic segmentation network for automated driving," in *Proc. NeurIPSW ML Phone Other Consum. Devices*, 2018, pp. 1–5.
[152] I. Leang, G. Sistu, F. Burger, A. Bursuc, and S. Yogamani, "Dynamic task weighting methods for multi-task networks in autonomous driving systems," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.
[153] P. Maddu et al., "FisheyeMultiNet: Real-time multi-task learning architecture for surround-view automated parking system," in *Proc. Irish Mach. Vis. Image Process. Conf.*, 2019, pp. 1–8.
[154] H. Rashed, A. El Sallab, S. Yogamani, and M. ElHelw, "Motion and depth augmented semantic segmentation for autonomous navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 364–370.
[155] I. Sobh, A. Hamed, V. R. Kumar, and S. Yogamani, "Adversarial attacks on multi-task visual perception for autonomous driving," *J. Imag. Sci. Technol.*, vol. 65, no. 6, p. 60408, Nov. 2021.
[156] M. Uřičář, D. Hurych, P. Křížek, and S. Yogamani, "Challenges in designing datasets and validation for autonomous driving," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 1–7.
[157] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[158] S. Ramachandran, G. Sistu, J. McDonald, and S. Yogamani, "Woodscape fisheye semantic segmentation for autonomous driving—CVPR 2021 OmniCV workshop challenge," 2021, *arXiv:2107.08246*.
[159] M. Eder, M. Shvets, J. Lim, and J.-M. Frahm, "Tangent images for mitigating spherical distortion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12426–12434.
[160] A. Muad, A. Hussain, S. Samad, M. Mustaffa, and B. Majlis, "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," in *Proc. IEEE Region 10 Conf. (TENCON)*, vol. 1, Nov. 2004, pp. 207–210.
[161] S. Mohapatra, S. Yogamani, H. Gotzig, S. Milz, and P. Mader, "BEVDetNet: Bird's eye view LiDAR point cloud based real-time 3D object detection for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2809–2815.
[162] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11138–11147.
[163] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 194–210.
[164] R. Pless, "Using many cameras as one," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, pp. 1–7.
[165] Y. Wang et al., "DETR3D: 3D object detection from multi-view images via 3D-to-2D queries," in *Proc. Conf. Robot Learn.*, Jan. 2022, pp. 180–191.
[166] M. Bojarski et al., "The NVIDIA PilotNet experiments," 2020, *arXiv:2010.08776*.

**Varun Ravi Kumar** received the M.Sc. degree from TU Chemnitz, Germany, in 2017, and the Ph.D. degree in artificial intelligence from TU Ilmenau in 2021. His Ph.D. thesis builds a first-ever six task multi-task learning near-field perception system that constitutes the necessary modules for a level3 autonomous stack using surround-view fisheye cameras. His research mainly focuses on designing self-supervised perception algorithms using neural networks for self-driving cars. His diverse expertise lies in geometric, semantic tasks, 2D and 3D object detection, point-cloud processing, and multi-task modeling. He was awarded the Deutschlandstipendium for Top-Class International Talent. He has 23 publications and two patents in top robotics and vision conferences.

**Ciarán Eising** (Member, IEEE) received the Ph.D. degree in computer engineering from NUI Galway in 2010. From 2009 to 2020, he was a Computer Vision Architect and a Senior Expert at Valeo, Ireland. In 2020, he joined the University of Limerick as a Lecturer in computer vision and artificial intelligence.

**Christian Witt** received the master's degree in physics from the Technical University of Dresden in 2017. Since 2017, he has been working in driving assistance research at Valeo, where he also held the title of an Expert. Since 2021, he has been the Team Lead of Data Recording and Tooling.

**Senthil Yogamani** is currently an Artificial Intelligence Architect and holds the director level technical leader position at Valeo, Ireland. He leads the research and design of AI algorithms for various modules of autonomous driving systems. He has over 17 years of experience in computer vision and machine learning, including 14 years of experience in industrial automotive systems. He is the author of more than 100 publications with more than 3000 citations and more than 100 inventions with 75 filed patent families. He serves on the Editorial Board of various leading IEEE automotive conferences, including ITSC and IV and an Advisory Board of various industry consortia, including Khronos, Cognitive Vehicles, and IS Auto. He was a recipient of the Best Associate Editor Award at ITSC 2015 and Best Paper Award at ITST 2012.