FisheyeYOLO: Object Detection on Fisheye Cameras for Autonomous Driving

Hazem Rashed[*1], **Eslam Mohamed**[*1], **Ganesh Sistu**[*2], **Varun Ravi Kumar**[3], **Ciarán Eising**[4], **Ahmad El-Sallab**[1] and **Senthil Yogamani**[2]
[*]Equal contribution [1]Valeo R&D, Egypt [2]Valeo Vision Systems, Ireland
[3]Valeo DAR Kronach, Germany [4]University of Limerick, Ireland

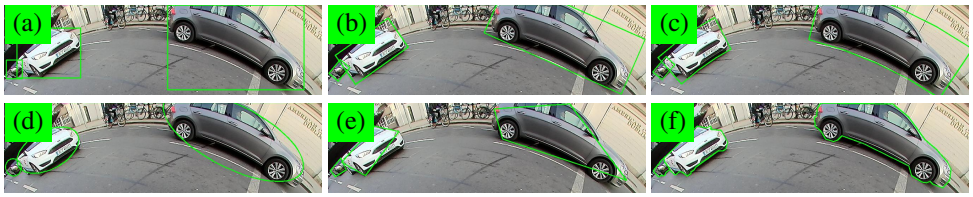

Figure 1: **Various 2D object detection representations on fisheye camera images.** (a) Standard Box, (b) Oriented Box, (c) Curved Box, (d) Ellipse, (e) 4-sided Polygon and (f) 24-sided Polygon.

Abstract

In this paper, we tackle Object Detection in fisheye images and study different object representations. The standard bounding box fails in fisheye cameras due to the strong radial distortion, particularly in the image's periphery. We explore better representations like oriented bounding box, ellipse, and generic polygon. Moreover, we design a novel curved bounding box model that has optimal properties for fisheye distortion models. Overall, the proposed polygon model improves mIoU relative accuracy by 40.3%. We present the first detailed study on object detection on fisheye cameras for autonomous driving scenarios to the best of our knowledge. The dataset[1] comprising of 10,000 images along with all the object representations ground truth will be made public to encourage further research. We summarize our work in a short video with qualitative results at https://youtu.be/4GLF4dz5CYc.

1 *Introduction*

Surround-view coverage is critical for low-speed maneuvering autonomous driving applications like automated parking [7, 6]. Four surround-view fisheye cameras are typically part of this sensor suite, enabling a dense 360° near field perception. The wide field of view of the fisheye image comes with the side effect of strong radial distortion. A common practice is to rectify distortions in the image using a 4^{th} order polynomial model or unified camera model [1]. However, undistortion comes with resampling distortion artifacts, especially at the periphery, reduced field of view, and non-rectangular image due to invalid pixels. Thus, we aim to perform object detection on distorted fisheye images. Although semantic segmentation is an easier solution on fisheye images, object detection annotation costs are much lower [17]. In general, there is limited work on fisheye perception [12, 22, 20, 11, 13].

One of the main issues is the lack of a public dataset, particularly for autonomous driving scenarios. The recent fisheye object detection paper FisheyeDet [14] emphasizes the lack of a useful dataset, and they create a simulated fisheye dataset by applying distortions to the Pascal VOC dataset [5].

[1]This dataset is an extension of our WoodScape dataset [23].

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

FisheyeDet makes use of a 4-sided polygon representation aided by distortion shape matching. SphereNet [3] and its variants [15, 18, 10] formulate CNNs on spherical surfaces. However, fisheye images do not follow spherical projection models, as seen by non-uniform distortion in horizontal and vertical directions.

The rectangular bounding box fails to be a good representation due to the massive distortion in the scene. As demonstrated in Figure 1. Instance segmentation can help obtain accurate object contours. However, it is a different task which is computationally complex and more expensive to annotate. Our objective is to present a more detailed study of various techniques for fisheye object detection in autonomous driving scenes. Our main contributions include:

- Exploration of seven different object representations for fisheye object detection.
- Design of novel representations, including the curved box and adaptive step polygon.
- Release of a dataset of 10,000 images with annotations for all the object representations.
- Empirical study of FisheyeYOLO baseline with different output representations.

2 *Object Representations*

2.1 Adaptation of Box representations

Standard Box Representation The rectangular bounding box is the most common representation for object detection. They are represented by four parameters ($\hat{x}$, $\hat{y}$, $\hat{w}$, $\hat{h}$), namely the box center, width and height. It may capture a large non-object area within the box for complex shapes, which is the case for fisheye distorted images, as shown in Figure 1 (a). ***Oriented Box Representation*** The oriented box is a simple extension of the standard box with an additional parameter $\hat{\theta}$ to capture the rotation angle of the box, in the range of (-90°to +90°) with respect to the x-axis. For this study, we used instance segmentation contours to estimate the optimally oriented box as a minimum enclosing rectangle. ***Ellipse Representation*** Ellipse is closely related to an oriented box and can be represented using the same parameter set with the ellipse's major and minor axes. In contrast to an oriented box, the ellipse has a smaller area at the edge, and thus it is better for representing overlapping objects as shown for the objects at the left end in Figure 1. We created our ground truth by fitting a minimum enclosing ellipse to the ground truth instance segmentation contours.

Curved box representation Bräuer-Burchardt and Voss [2] show that if we assume that the first-order *division model* can accurately describe the fisheye distortion, then we may use circles in the image to model the projected straight lines. In [8], the authors adopt the division model slightly to include an additional scaling factor and prove that this does not impact the projection of line to a circle. They show that the division model is a correct replacement for the equidistant fisheye model. Thus, we propose a novel curved bounding box representation using circular arcs.

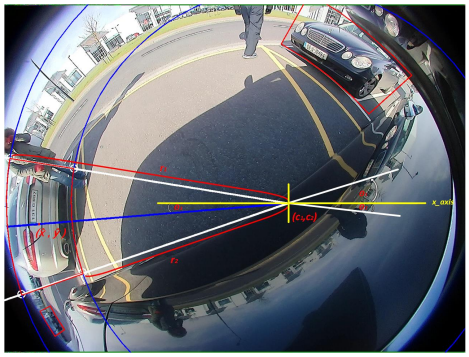

Figure 2: We propose the **Curved Bounding Box** using a circle with an arbitrary center and radius, as illustrated.

Figure 2 illustrates the details of the curved bounding box. The blue line represents the axis, and the white lines intersect with the circles creating starting and ending points of the polygon. This representation allows two sides of the box to be curved, giving the flexibility to adapt to image distortion in fisheye cameras. It can also specialize in an oriented bounding box when there is no distortion for the objects near the principal point. Additionally, it captures the radial distortion and obtains a better footpoint, which helps localize the vehicle in the 3D world.

We create an automatic process to generate the representation that takes an object contour as an input. First, we generate an oriented box from the output contour. We choose a point that lies on the oriented box's axis line to represent a circle center. From the center, we create two circles intersecting with the corner points of the bounding box. We construct the polygon based on the two circles and the intersection points. To find the best circle center, we iterate over the axis line and choose the circle

center, which forms a polygon with the minimum IoU with the instance mask. The output polygon can be represented by 6 parameters, namely, (c_1, c_2, r_1, r_2, θ_1, θ_2) representing the circle center, two radii and angles of the start and end points of the polygon relative to the horizontal x-axis. By simple algebraic manipulation, we can re-parameterize the curved box using the object center ($\hat{x}$, $\hat{y}$) following a typical box representation instead of the center of the circle.

2.2 Generic Polygon Representations

Polygon is a generic representation for any arbitrary shape and is typically used even, for instance segmentation annotation. Thus polygon output can be seen as a coarse segmentation. Polar representations have been studied in PolarMask [21] and PolyYOLO [9]. The object contour can be uniformly sampled in the range of $360°$ split into N equal polygon vertices, each represented by the radial distance r from the object's centroid. Polygon is finally represented by object center ($\hat{x}$, $\hat{y}$) and $\{r_i\}$. PolyYOLO [9] showed that it is better to learn polar representation of the vertices $\{(r_i, \theta_i)\}$ instead.

Curvature-adaptive Perimeter Sampling Uniform sampling cannot represent curvatures due to sampling losses. Thus, we propose an adaptive sampling based on the curvature of the local contour. We distribute the vertices non-uniformly in order to represent the object contour best. We adopt the algorithm in [19] to detect the dominant points in a given curved shape, which best represents the object. Then we reduce the set of points using the algorithm in [4] to get the most representative simplified curves.

3 Experimental Results

3.1 Dataset and Evaluation Metrics

Our dataset comprises 10,000 images captured in several European countries and the USA, with 1MPx resolution and $190°$ horizontal FOV, sampled roughly equally from the four views. The dataset comprises 4 classes, namely vehicles, pedestrians, bicyclists, and motorcyclists. For our experiments, we used only the vehicles' class. We divide our dataset into 60-10-30 split and train all the models using the same setting. We discuss the further details in our *WoodScape Dataset* paper.

Unlike conventional evaluation, our first objective is to provide better representation than a conventional bounding box. Therefore, we first evaluate our representations against the most accurate representation of the object, the ground-truth instance segmentation mask. We report mIoU between a representation and the ground-truth instance mask. Additionally, we qualitatively evaluate the representations in obtaining object intersection with the ground (footpoint). Finally, we report model speed in terms of frames-per-second (fps) as we focus on real-time performance.

3.2 FisheyYOLO network

We adapt YOLOv3 [16] model to output different representations, called FisheyeYOLO, discussed in Section 2. Our baseline bounding box model is the same as YOLOv3, except the Darknet53 encoder is replaced with ResNet18 encoder. The final loss is a combination of center $\mathcal{L}_{xy}$, width/height $\mathcal{L}_{wh}$, object $\mathcal{L}_{obj}$ and class $\mathcal{L}_{class}$ sub-losses as in YOLOv3. In the case of oriented box or ellipse prediction, we define an additional loss function based on ellipse angle or orientation of the box. The loss function for oriented box and ellipse is given by:

$$\mathcal{L}_{orn} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} [\theta_i - \hat{\theta}_i]^2, \qquad \mathcal{L}_{total} = \mathcal{L}_{xy} + \mathcal{L}_{wh} + \mathcal{L}_{obj} + \mathcal{L}_{class} + \mathcal{L}_{\mathbf{orn}} \quad (1)$$

The polar polygon regression loss is given by,

$$\mathcal{L}_{cods} = \sum_{i=0}^{S^2} \sum_{j=0}^{N} \hat{\alpha}_{ij} [(r_{i,j} - \hat{r}_{i,j})^2 + (\theta_{i,j} - \hat{\theta}_{i,j})^2], \quad \mathcal{L}_{mask} = -\sum_{i=0}^{S^2} \sum_{j=0}^{N} \alpha_{ij} log(\hat{\alpha}_{ij}) \quad (2)$$

$$\mathcal{L}_{total} = \mathcal{L}_{xy} + \mathcal{L}_{obj} + \mathcal{L}_{class} + \mathcal{L}_{\mathbf{cods}} + \mathcal{L}_{\mathbf{mask}} \quad (3)$$

where N corresponds to the number of sampling points, each point is sampled with a step size of $360/N$ angle in polar coordinates. Our polar loss is similar to PolyYOLO [9], where each polygon

point is represented using three parameters r, θ, and α. Hence the total required parameters for N sampling points are $3 \times N$. In case of curved box, $\mathcal{L}_{wh}$ is replaced by $\mathcal{L}_{cods}$ in equation (2). We further improve our predictions by adding our IoU loss function, which minimizes the area between the prediction and ground truth.

3.3 Results Analysis

We evaluate different representations, both quantitatively and qualitatively. For quantitative analysis, we first evaluate each representation against the instance segmentation mask ground truth as in Table 1. Also, we evaluate the prediction output from our FisheyeYOLO network for each representation, as shown in Table 2. Compared to the standard bounding box approach, the proposed oriented box and ellipse models improved mIoU score on the test set by 2% and 3.8%. Ellipse prediction provides slightly better accuracy than the oriented box as it has higher immunity to occlusions, as demonstrated in the qualitative results video. Our distortion-aware bounding box provides a significant improvement over the standard box in IoU with instance masks. However, it is slightly less than an oriented box because two circular sides of the box share the same circle center, which adds some area inside the polygon, decreasing the IoU. However, this representation provides better qualitative measures in terms of alignment with scene distortion. The object's footpoint is captured almost entirely, as observed in qualitative results, especially for the side cameras where distortion is maximized.

Representation	**mIoU**				**mIoU**	***No. of params***
	Front	Rear	Left	Right	All	
Standard Box	53.7	47.9	60.6	43.2	51.35	4
Curved Box	53.7	48.6	63.5	44.2	52.5	6
Oriented Box	55	50.2	64.8	45.9	53.9	5
Ellipse	56.5	51.7	66.5	47.5	55.5	5
24-sided Polygon	**87.2**	**87**	**86.2**	**86.1**	**86.6**	48

Table 1: **Evaluation of representation capacity of various representations.** We estimate the best fit for each representation using ground truth instance segmentation and then compute mIoU to evaluate capacity. We also list the number of parameters used for each representation to provide comparison of complexity.

The qualitative results video shows a visual evaluation of our proposed representations. Unlike boxes, ellipse allows a minimal representation of the object due to the absence of corners, which avoids incorrect occlusion with free parking slots. Polygon representation provides higher accuracy in terms of IoU with instance mask. We fix the number of points of polygon representation to 24 to represent each object. We observe no significant delay in fps due to increasing the number of parameters/objects where our models run at 56 fps on a standard NVIDIA TitanX GPU. It is due to the utilization of YoloV3 [16] architecture, which performs the prediction at each grid cell in a parallel mechanism.

Representation	**IoU**				**mIoU**
	Front	Rear	Left	Right	
YoloV3	32.5	32.1	34.2	27.8	31.6
Curved Box	33	32.7	35.4	28	32.3
Oriented Box	33.9	33.5	37.2	30.1	33.6
Ellipse	35.4	35.4	40.4	30.5	35.4
Polygon	**44.4**	**46.8**	**44.7**	**42.7**	**44.65**

Table 2: **Quantitative results of proposed model on different representations on our dataset.** The experiments are performed on the best performing model according to Table 1.

4 Conclusion

In this paper, we studied various representations of fisheye object detection. At a high level, we can split them into bounding box extensions and generic polygon representations. We explored oriented bounding box, ellipse, and designed a curved bounding box with optimal fisheye distortion properties for the former. We proposed a curvature adaptive sampling method for polygon representations, which improves significantly over uniform sampling methods. Overall, the proposed models improve the relative mIoU accuracy significantly by 40.3% compared to a YOLOv3 baseline. We consider our method to be a baseline for further research into this area. We will make the dataset with ground truth annotation for various representations publicly available. We hope this encourages further research in this area leading to mature object detection on undistorted fisheye images.

References

[1] Joao P Barreto. Unifying image plane liftings for central catadioptric and dioptric cameras. In *Imaging Beyond the Pinhole Camera*. Springer, 2006. 1

[2] Christian Bräuer-Burchardt and Klaus Voss. A new algorithm to correct fish-eye and strong wide-angle-lens-distortion from single images. In *IEEE International Conference on Image Processing*, 2001. 2

[3] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[4] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In *Cartographica: the international journal for geographic information and geovisualization*. University of Toronto Press, 1973. 3

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. In *International Journal of Computer Vision*. Springer, 2010. 1

[6] Markus Heimberger, Jonathan Horgan, Ciarán Hughes, John McDonald, and Senthil Yogamani. Computer vision in automated parking systems: Design, implementation and challenges. *Image and Vision Computing*, 68:88–101, 2017. 1

[7] Jonathan Horgan, Ciarán Hughes, John McDonald, and Senthil Yogamani. Vision-based driver assistance systems: Survey, taxonomy and advances. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2032–2039. IEEE, 2015. 1

[8] Ciaran Hughes, Robert McFeely, Patrick Denny, Martin Glavin, and Edward Jones. Equidistant fish-eye perspective with application in distortion centre estimation. In *Image and Vision Computing*, 2010. 2

[9] Petr Hurtik, Vojtech Molek, Jan Hula, Marek Vajgl, Pavel Vlasanek, and Tomas Nejezchleba. Poly-yolo: higher speed, more precise detection and instance segmentation for yolov3, 2020. 3

[10] Chiyu Jiang, Jingwei Huang, Karthik Kashinath, Philip Marcus, Matthias Niessner, et al. Spherical cnns on unstructured grids. In *arXiv preprint arXiv:1901.02039*, 2019. 2

[11] Varun Ravi Kumar, Sandesh Athni Hiremath, Markus Bach, Stefan Milz, Christian Witt, Clément Pinard, Senthil Yogamani, and Patrick Mäder. Fisheyedistancenet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 574–581. IEEE, 2020. 1

[12] Varun Ravi Kumar, Stefan Milz, Christian Witt, Martin Simon, Karl Amende, Johannes Petzold, Senthil Yogamani, and Timo Pech. Monocular fisheye camera depth estimation using sparse lidar supervision. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018. 1

[13] Varun Ravi Kumar, Senthil Yogamani, Markus Bach, Christian Witt, Stefan Milz, and Patrick Mader. Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. *arXiv preprint arXiv:2007.06676*, 2020. 1

[14] Tangwei Li, Guanjun Tong, Hongying Tang, Baoqing Li, and Bo Chen. Fisheyedet: A self-study and contour-based object detector in fisheye images. In *IEEE Access*. Ieee, 2020. 1

[15] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. In *Astronomy and Computing*. Elsevier, 2019. 2

[16] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. In *arXiv preprint arXiv:1804.02767*, 2018. 3, 4

[17] Mennatullah Siam, Sara Elkerdawy, Martin Jagersand, and Senthil Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pages 1–8. IEEE, 2017. 1

[18] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[19] C-H Teh and Roland T. Chin. On the detection of dominant points on digital curves. In *IEEE Transactions on pattern analysis and machine intelligence*. Ieee, 1989. 3

[20] Michal Uřičář, Pavel Křížek, Ganesh Sistu, and Senthil Yogamani. Soilingnet: Soiling detection on automotive surround-view cameras. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 67–72. IEEE, 2019. 1

[21] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[22] Marie Yahiaoui, Hazem Rashed, Letizia Mariotti, Ganesh Sistu, Ian Clancy, Lucie Yahiaoui, Varun Ravi Kumar, and Senthil Yogamani. Fisheyemodnet: Moving object detection on surround-view cameras for autonomous driving. *arXiv preprint arXiv:1908.11789*, 2019. 1

[23] Senthil Yogamani, Ciaran Hughes, Jonathan Horgan, Ganesh Sistu, Sumanth Chennupati, Michal Uricar, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, and et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Ieee, 2019. 1