https://doi.org/10.2352/ISSN.2470-1173.2021.17.AVM-181

FisheyeDistanceNet++: Self-Supervised Fisheye Distance Estimation with Self-Attention, Robust Loss Function and Camera View Generalization

Varun Ravi Kumar[1,3]***, Senthil Yogamani***[2]***, Stefan Milz***[3]***, Patrick Mäder***[3]
[1] ***Valeo DAR Kronach, Germany***
[2] ***Valeo Vision Systems, Ireland***
[3] ***Technische Universität Ilmenau, Germany***

Abstract

FisheyeDistanceNet [1] proposed a self-supervised monocular depth estimation method for fisheye cameras with a large field of view ($> 180°$). To achieve scale-invariant depth estimation, FisheyeDistanceNet supervises depth map predictions over multiple scales during training. To overcome this bottleneck, we incorporate self-attention layers and robust loss function [2] to FisheyeDistanceNet. A general adaptive robust loss function helps obtain sharp depth maps without a need to train over multiple scales and allows us to learn hyperparameters in loss function to aid in better optimization in terms of convergence speed and accuracy. We also ablate the importance of Instance Normalization over Batch Normalization in the network architecture. Finally, we generalize the network to be invariant to camera views by training multiple perspectives using front, rear, and side cameras. Proposed algorithm improvements, FisheyeDistanceNet++*, result in 30% relative improvement in RMSE while reducing the training time by 25% on the WoodScape dataset. We also obtain state-of-the-art results on the KITTI dataset, in comparison to other self-supervised monocular methods.*

Introduction

Depth estimation is an important task in autonomous driving as it is used to avoid obstacles and plan trajectories. Almost all approaches for depth estimation [3], [4], [5], [6], [7] have primarily focused on traditional pinhole camera images. Surround view cameras have become a standard sensor in automated driving and recently there is a lot of progress in various visual perception tasks such as semantic segmentation [8], moving object detection [9], re-localisation [10], soiling detection [11, 12], etc. The standard depth estimations methods do not work out of the box for fisheye or omnidirectional cameras, which have a strong radial distortion. *FisheyeDistanceNet* [1] is the first proposed end-to-end self-supervised monocular scale-aware training framework for fisheye cameras with a large field of view ($> 180°$) to regress a Euclidean distance map. Other relevant work on depth estimation include [13, 14, 15].

There is a strong trend of using automated optimization techniques to find the best neural network hyperparameters and architecture topology [16]. Significant gains have been achieved in accuracy through these techniques. However, loss function optimization has been relatively less explored. The default and virtually only choice of photometric loss employed in [4], [3], [17], [1] is ℓ_1. Inspired by the recent work of Barron [2], we explore using a family of loss functions and find an optimal one for our self supervised depth estimation task on fisheye images. We improve upon the baseline *FisheyeDistanceNet* by making the following contributions:

Figure 1: **Distance estimation on a single fisheye image.** Our self-supervised model, **FisheyeDistanceNet++**, generates superior quality distance maps.

- We mainly bring attention to the photometric loss's design choice for image restoration in the self-supervision training regime of depth. The significance of the error metric used to train neural networks for image processing is demonstrated: ℓ_1 loss is still the *de facto* standard despite its well-known drawbacks.
- We study the generalized loss function [2] which is more robust than the standard ℓ_1. This loss function adapts itself during training without requiring any manual parameter tuning. We demonstrate significant improvement in accuracy and faster training time without changing the baseline architecture.
- We introduce a novel distance estimation network architecture using a self-attention based encoder to enhance the fea-

tures fed to the distance decoder.

- We depict the importance of Instance Normalization over Batch Normalization in the entire architecture.
- We train out the network on multiple cameras and present a generalized network invariant to different camera views.

Background

Supervised distance estimation methods [18], [19], [20], [21] require dense pixel-wise labels often constructed from stereo maps or sparse LiDAR point-clouds. Supervised distance estimation is a difficult problem in fisheye cameras due to several challenges. Pixel-wise distance labels generated by projecting LiDAR point-clouds to fisheye camera images often contain significant motion distortions. Moreover, common LiDAR sensors do not provide coverage in the entire near range of ego vehicle's to generate a dense depth map in the fisheye camera view. This is usually addressed by mounting additional LiDAR sensors or by approximation of distance in the blind zones.

To overcome this challenge, FisheyeDistanceNet [1] proposed self-supervised methods to estimate distance in fisheye camera images efficiently. Self-supervised methods [22], [23], [24] typically supervise photometric loss between a geometrically consistent target image and the reconstructed image obtained using monocular depth and ego-motion predictors.

Photometric loss

We incorporate ℓ_1 pixel-wise loss and Structural Similarity (SSIM) [25] loss terms for the photometric error between the target image I_t and the reconstructed target image $\hat{I}_{t'\to t}$. The photometric loss $\mathscr{L}_p$ is:

$$\tilde{\mathscr{L}}_p(I_t,\hat{I}_{t'\to t}) = \omega\,\frac{1-\text{SSIM}(I_t,\hat{I}_{t'\to t})}{2} + (1-\omega)\,\left\|(I_t-\hat{I}_{t'\to t})\right\|_{l^1} \tag{1}$$

where $\omega = 0.85$ is a weighting factor between both loss terms. The final per-pixel minimum reconstruction loss $\mathscr{L}_r$ [3] is then calculated over all the source images

$$\mathscr{L}_p = \min_{t'\in\{t+1,t-1\}} \tilde{\mathscr{L}}_p(I_t,\hat{I}_{t'\to t}) \tag{2}$$

Our focus in this paper mainly lies on the ℓ_1 loss term and we use the common regression loss function $\xi(.)$ given by:

$$\arg\min_{\theta}\,\xi\left(f_\theta(x)-y\right) \tag{3}$$

Robust loss function

Recently, a general and more robust loss function is proposed by Barron [2] which is a generalization of many common losses such as ℓ_1 or ℓ_2 functions. It can also represent the Geman-McClure, Welsch/Leclerc, Cauchy/Lorentzian, Welsch/Leclerc and Charbonnier ℓ_1-ℓ_2 loss functions. In this loss, robustness is introduced as a continuous parameter and it can be optimized within the loss function to improve the performance of regression tasks. The general form of the loss function is:

$$f_{\text{rob}}(\zeta,\rho,c) = \frac{|\rho-2|}{\rho}\left(\left(\frac{(\zeta/c)^2}{|\rho-2|}+1\right)^{\rho/2}-1\right) \tag{4}$$

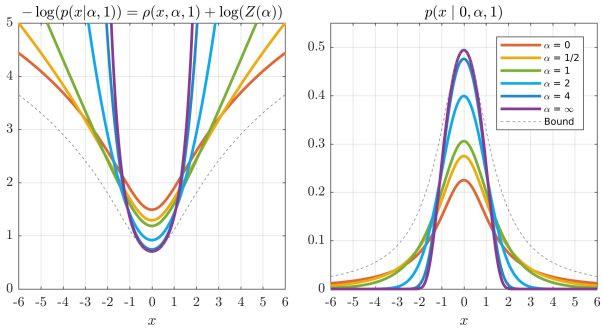


Figure 2 Figure is reproduced from [2]. The negative log-likelihoods (left) and probability densities (right) of the distribution relating to Barron's [2] loss function when it is defined ($\rho \geq 0$). A log partition function shifts the NLLs losses.

The free parameters in this loss function can be automatically adapted to any particular problem via the data driven optimization. To induce ρ as a trainable parameter Barron [2] encapsulates the loss into a probability density function given by:

$$p(\zeta\mid\mu,\rho,c) = \frac{1}{cZ(\rho)}\exp\left(-\rho\left(\zeta-\mu,\rho,c\right)\right) \tag{5}$$

$$Z(\rho) = \int_{-\infty}^{\infty}\exp\left(-\rho\left(\zeta,\rho,1\right)\right) \tag{6}$$

where $p(\zeta\mid\mu,\rho,c)$ is only defined if $\rho\geq 0$, as $Z(\rho)$ is divergent when $\rho < 0$. Then the optimization function reduces to:

$$\arg\min_{\theta,\rho} -log(p(\zeta|\rho) = \rho\left(\zeta,\rho\right)+log(Z(\rho)) \tag{7}$$

where $log(Z(\rho))$ is an analytical function which is approximated with a cubic spline function. $Z(\rho)$ is an important factor in the loss function as it reduces the cost of outliers. The loss of outliers decreases with the reduction of ρ. Correspondingly, the loss of inliers will increase.

The main properties of the robust loss function are summarized below:

1. It is monotonic with respect to its inputs $|\zeta|$ and ρ which is useful for graduated non-convexity.
2. It is smooth respect to its inputs ζ and ρ (i.e in C^∞).
3. It has bounded first and second derivatives (no exploding gradients and easier pre-conditioning).

Network Architecture

A novel self-attention based encoder model coupled with norm decoder and skip connections is implemented to handle the view synthesis. Fig. 3 illustrates the proposed *FisheyeDistanceNet++* architecture. Multiple camera images from four sides of the vehicle are used to train a single model to make the network invariant to the perspective. We have kept the encoder-decoder architecture simplistic for easy extension to multi-task learning [26] and fusion with other sensors like ultrasonics [27] and Lidar [28, 29].

Self-Attention Encoder

Previous depth estimation networks [3, 4] use normal convolutions for capturing the local information in an image, but the convolutions' receptive field is quite small. Inspired by [30], who

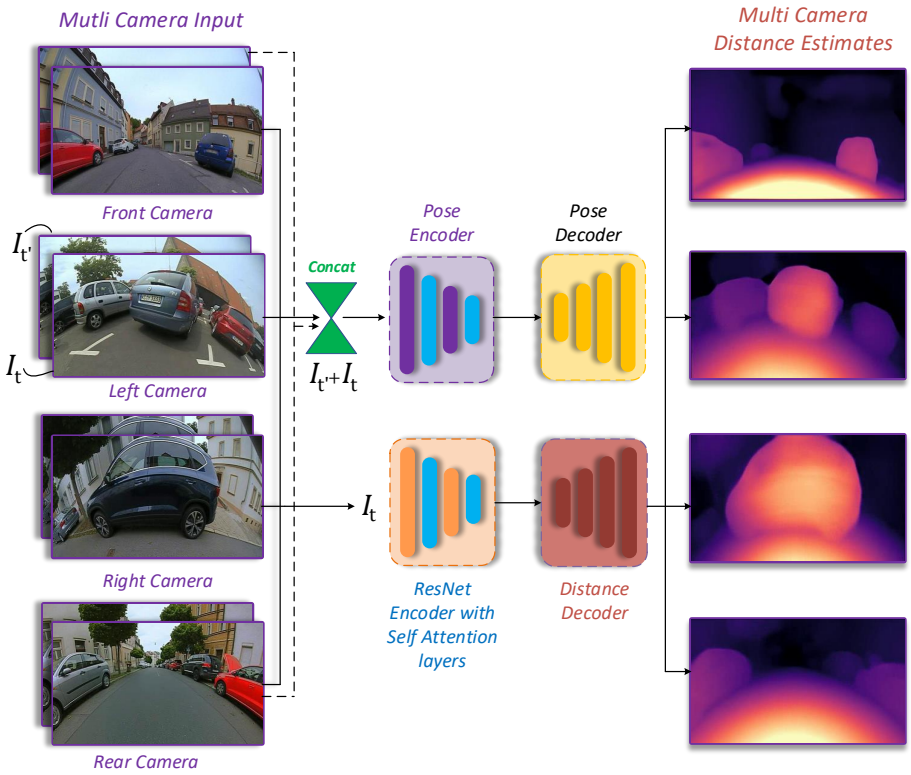


Figure 3 **Distance estimation on a multiple fisheye cameras.** Our self-supervised model, **FisheyeDistanceNet++**, generalizes to multiple view-points and estimates superior quality distance maps.

took self-attention in CNNs even further by using stand-alone self-attention blocks instead of only enhancing convolutional layers. The authors present a self-attention layer which may replace convolution while reducing the number of parameters. Similar to a convolution, given a pixel $x_{ij} \in \mathbb{R}^{d_{in}}$ inside a feature map, the local region of pixels defined by positions $ab \in \mathcal{N}_k(ij)$ with spatial extent k centered around x_{ij} are extracted initially which is referred to as a memory block. For every memory block, the single-headed attention for computing the pixel output $z_{ij} \in \mathbb{R}^{d_{out}}$ is then calculated by:

$$z_{ij} = \sum_{ab \in \mathcal{N}_k(ij)} \texttt{softmax}_{ab}\left(q_{ij}^{\top} k_{ab}\right) v_{ab} \tag{8}$$

where $q_{ij} = W_Q x_{ij}$ are the *queries*, *keys* $k_{ab} = W_K x_{ab}$, and *values* $v_{ab} = W_V x_{ab}$ are linear transformations of the pixel in position ij and the neighborhood pixels. The learned transformations are denoted by the matrices W. $\texttt{softmax}_{ab}$ defines a softmax applied to all logits computed in the neighborhood of ij. $W_Q, W_K, W_V \in \mathbb{R}^{d_{out} \times d_{in}}$ are trainable transformation weights. There exists an issue in the above-discussed approach, as there is no positional information encoded in the attention block. Thus the Eq. is invariant to permutations of the individual pixels. For perception tasks, it is typically helpful to consider spatial information in the pixel domain. For example, the detection of a pedestrian is composed of spotting faces and legs in a proper relative localization. The main advantage of using self-attention layers in the encoder is that it induces a synergy between geometric and semantic features for distance estimation and semantic segmentation tasks. In [31] sinusoidal embeddings are used to produce the absolute positional information. Following [30] instead of attention with 2D relative position embeddings, we incorporate relative attention due to their better accuracy for computer vision tasks. The relative distances of the position ij to every neighborhood pixel (a,b) is calculated to obtain the relative embeddings. The calculated distances are split up into row and column distances r_{a-i} and r_{b-j} and the embeddings are concatenated to form $r_{a-i,b-j}$ and multiplied by the query q_{ij} given by:

$$z_{ij} = \sum_{ab \in \mathcal{N}_k(ij)} \texttt{softmax}_{ab}\left(q_{ij}^{\top} k_{ab} + q_{ij}^{\top} r_{a-i,b-j}\right) v_{ab} \tag{9}$$

It ensures the weights calculated by the softmax function are modulated by both the relative distance and content of the key from the query. Instead of focusing on the whole feature map, the attention layer only focuses on the memory block.

Additional Considerations and Final Objective Loss

We incorporate the same protocols from FisheyeDistanceNet [1], to perform view synthesis using the polynomial projection model. We follow the same technique in [1] to recover the scale of the predicted distance estimates on fisheye and pinhole camera models. The photometric loss values are clipped to improve the process of distance estimation in homogeneous and occluded areas. We also added the backward sequences to the view-synthesis to resolve unknown estimates in the border region of the images. The self-supervised final loss comprises of a photometric term $\mathscr{L}_p$ that is computed between the reconstructed $\hat{I}_{t' \to t}$ and original I_t target images, and an inverse depth or distance

	Method	Resolution	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
			lower is better				higher is better		
	KITTI								
Original [32]	SfMLeaner [4]	416 x 128	0.183	1.595	6.709	0.270	0.734	0.902	0.959
	Vid2depth [33]	416 x 128	0.163	1.240	6.220	0.250	0.762	0.916	0.968
	DDVO [6]	416 x 128	0.151	1.257	5.583	0.228	0.810	0.936	0.974
	Struct2Depth [34]	416 x 128	0.141	1.026	5.291	0.215	0.816	0.945	0.979
	EPC++ [35]	640 x 192	0.141	1.029	5.350	0.216	0.816	0.941	0.976
	Monodepth2 [3]	640 x 192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	FisheyeDistanceNet [1]	640 x 192	0.117	0.867	4.739	0.190	0.869	0.960	0.982
	PackNet-SfM [17]	640 x 192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	FisheyeDistanceNet++	640 x 192	**0.107**	**0.721**	**4.564**	**0.178**	**0.894**	**0.971**	**0.986**
	Monodepth2 [3]	1024 x 320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
	FisheyeDistanceNet [1]	1024 x 320	0.109	0.788	4.669	0.185	0.889	0.964	0.982
	FisheyeDistanceNet++	1024 x 320	**0.103**	**0.705**	**4.386**	**0.164**	**0.897**	**0.980**	**0.989**
Improved [36]	SfMLeaner [4]	416 x 128	0.176	1.532	6.129	0.244	0.758	0.921	0.971
	Vid2Depth [33]	416 x 128	0.134	0.983	5.501	0.203	0.827	0.944	0.981
	DDVO [6]	416 x 128	0.126	0.866	4.932	0.185	0.851	0.958	0.986
	EPC++ [35]	640 x 192	0.120	0.789	4.755	0.177	0.856	0.961	0.987
	Monodepth2 [3]	640 x 192	0.090	0.545	3.942	0.137	0.914	0.983	0.995
	PackNet-SfM [17]	640 x 192	**0.078**	0.420	3.485	0.121	**0.931**	0.986	**0.996**
	FisheyeDistanceNet++	640 x 192	0.081	**0.414**	**3.412**	**0.117**	0.926	**0.987**	**0.996**
	WoodScape								
	FisheyeDistanceNet [1]	512 x 256	0.152	0.768	2.723	0.210	0.812	0.954	0.974
	FisheyeDistanceNet++	512 x 256	**0.102**	**0.396**	**1.869**	**0.123**	**0.890**	**0.988**	**0.994**

Table 1: **Quantitative performance comparison of FisheyeDistanceNet++** for depths up to 80 m for KITTI and 40 m for FisheyeDistanceNet++. *Original* incorporates raw depth maps from [32] for evaluation, and *Improved* uses annotated depth maps from [36]. All the methods listed in the table are self-supervised on monocular video sequences. Excluding FisheyeDistanceNet, FisheyeDistanceNet++ and PackNet-SfM rest of the methods scale the depth estimates using median ground-truth LiDAR during inference. We generalized the previous model FisheyeDistanceNet in our new training framework and added additional features which improved results in WoodScape [37].

regularization term $\mathscr{L}_s$ introduced in [5] that ensures edge-aware smoothing in the distance estimates $\hat{D}_t$. Finally, we apply a cross-sequence distance consistency loss $\mathscr{L}_{dc}$ derived from the chain of frames in the training sequence from [1]. The final objective loss $\mathscr{L}_{tot}$ is averaged per pixel, scale and image batch is:

$$\mathscr{L}_{tot} = \mathscr{L}_p(I_t, \hat{I}_{t'\to t}) + \beta\, \mathscr{L}_s(\hat{D}_t) + \gamma\, \mathscr{L}_{dc}(\hat{D}_t, \hat{D}_{t'}) \qquad (10)$$

where β and γ are weight terms between the reconstruction loss $\mathscr{L}_r$, the distance regularization loss and the cross-sequence distance consistency loss $\mathscr{L}_s$, respectively.

Implementation Details

We base our model on FisheyeDistanceNet [1], an *encoder-decoder* network with skip connections. We prefer ResNet18 [38] as the encoder since it produces an efficient depth prediction and enhances in higher complexity encoders is incremental after testing different variants of ResNet family. We could leverage the usage of a more robust loss function over ℓ_1 to reduce training times on ResNet18 and ResNet50 as shown in Table 2 by carrying out a single scale image depth prediction instead of multi-scale in [1]. We incorporate Pytorch [39] and the final training objective function can be minimized using Ranger (RAdam [40] + LookAhead [41]) optimizer than the previously employed Adam [42]. To adjust the adaptive momentum of Adam, RAdam leverages a dynamic rectifier based on the variance and effectively creates an automated warm-up which ensures a solid start to the training on the custom dataset. LookAhead "lessens the need for extensive hyperparameter tuning" while accomplishing "faster convergence across different deep learning tasks with minimal computational overhead". Breakthroughs can be achieved from RAdam and LookAhead in various aspects of deep learning optimization, and the combination is highly synergistic, possibly providing the best of both improvements for the results.

Method	*Robust loss*	*Self Attn.*	*Instance Norm*	*Batch Norm*	Abs Rel	Sq Rel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
					lower is better				higher is better		
FisheyeDistanceNet [1]	✗	✗	✗	✓	0.152	0.768	2.723	0.210	0.812	0.954	0.974
FisheyeDistanceNet++ (ResNet-18)	✓	✗	✗	✓	0.147	0.572	2.341	0.176	0.869	0.966	0.980
	✓	✗	✗	✓	0.147	0.572	2.341	0.176	0.869	0.966	0.980
	✓	✗	✓	✗	0.119	0.497	2.268	0.156	0.886	0.970	0.984
	✓	✓	✗	✗	0.111	0.429	2.028	0.155	0.875	0.980	0.990
	✓	✓	✓	✗	0.109	0.421	1.989	0.147	0.875	0.982	0.990
	✓	✓	✓	✓	**0.102**	**0.396**	**1.869**	**0.123**	**0.890**	**0.988**	**0.994**
FisheyeDistanceNet++ (ResNet-50)	✓	✗	✓	✗	0.143	0.566	2.310	0.169	0.872	0.969	0.981
	✓	✗	✓	✗	0.109	0.485	2.197	0.147	0.892	0.974	0.988
	✓	✓	✗	✗	0.105	0.411	1.978	0.132	0.881	0.984	0.992
	✓	✓	✓	✗	0.101	0.394	1.918	0.135	0.880	0.984	0.994
	✓	✓	✓	✓	**0.088**	**0.345**	**1.785**	**0.111**	**0.894**	**0.991**	**0.996**

Table 2: **Ablative analysis** showing the effect of each of our contributions using the Fisheye WoodScape dataset [37]. The input resolution is 512×256 pixels and distances are capped at $40m$. We start with FisheyeDistanceNet [1] baseline and incrementally add Robust loss, self-attention based encoder, Instance Normalization and Batch Normalization.

Cams	Abs Rel	Sq Rel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
Front	0.102	0.396	1.869	0.123	0.890	0.988	0.994
Rear	0.105	0.401	1.885	0.131	0.891	0.986	0.992
Left	0.102	0.398	1.874	0.126	0.886	0.986	0.994
Right	0.107	0.405	1.876	0.128	0.884	0.983	0.990

Table 3 **Ablation study on multiple cameras** using the Fisheye WoodScape dataset [37].

Evaluation

We evaluate FisheyeDistanceNet++'s depth and distance estimation results using the metrics proposed by Eigen et al. [32] to facilitate comparison. The quantitative shown in the Table 1 illustrate that the improved scale-aware self-supervised approach outperforms all the state-of-the-art monocular approaches. More specifically, we improve the baseline *FisheyeDistanceNet* with the usage of a general and adaptive loss function [2]. Due to its usage, we could leverage a deeper architecture ResNet50 than the previously used ResNet18. We could not leverage the Cityscapes dataset into our training regime to benchmark our scale-aware framework, due to the absence of odometry data. In contrast to PackNet-SfM [17], which presumably uses a superior architecture compared to our super-resolution ResNet18, with the capability of estimating scale-aware depths with their velocity supervision loss, we could achieve higher accuracy with subtle improvements to the standard ResNet18 and the training framework. In Fig. 4, we show a few qualitative results of the failure cases having artifacts such as holes or merging of thin objects like poles with the background. Finally, in Fig. 5, we showcase qualitative results on the KITTI Eigen split. In Table3, we evaluate the results on multiple cameras mounted around the car. Although the distortion around the side cameras is significantly high compared to the front camera, we could obtain similar results, and the model generalizes well to different viewpoints. Using a single model for multiple views allows us to perform 3D surround-view tasks and this information can also be leveraged in SLAM systems.

Fisheye ablation study on variants of robust loss function strategy

The ℓ_1 loss is replaced with different variants of the robust general loss [2] and showcase that usage of adaptive or annealed variants of the loss can significantly improve the performance. The shape parameter ρ is varied, keeping the scale fixed with a general distribution than a fixed Laplacian distribution. Instead of RGB representation, following [2] YUV wavelet representations are used to model the images with the robust loss function. The loss is applied on a YUV wavelet decomposition. The multi-scale training as the reconstruction loss in FisheyeDistanceNet [1] is dropped which induces the sum of the means of the losses imposed at each scale in a D-level pyramid of side prediction since [2] is a D level normalized wavelet decomposition. Compared to [2] we retained the edge smoothness loss from FisheyeDistanceNet [1] as it gave better results which can be seen in Table 4. The fixed scale assumption is matched by setting the loss's scale c fixed to 0.01, which also roughly matches the shape of its ℓ_1 loss. The loss is multiplied by c to avoid exploding gradients which bounds the gradient magnitudes by residual magnitudes.

For the fixed scale models in Table 4, we used a constant value for ρ for all wavelet coefficients. We observe that there is an improvement in the performance, and there is no single value of ρ, which is optimal. In the adaptive $\rho \in (0,2)$ variant, ρ is made a free parameter and is allowed to be optimized along with the network weights during training. The adaptive plan of action outperforms the fixed strategies, which showcases the importance of allowing the model to regulate the robustness of its loss during training adaptively. Comparison of the adaptive model's performance with the fixed models indicates that no single setting of ρ is optimal for all wavelet coefficients.

Method	ρ	Abs Rel	Sq Rel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
FisheyeDistanceNet [1]	✗	0.152	0.768	2.723	0.210	0.812	0.954	0.974
FisheyeDistanceNet++	1	0.148	0.721	2.615	0.202	0.837	0.961	0.979
	0	0.136	0.648	2.482	0.183	0.854	0.963	0.981
	2	0.125	0.549	2.338	0.175	0.855	0.970	0.981
	$(0,2)$	**0.119**	**0.497**	**2.268**	**0.156**	**0.886**	**0.970**	**0.984**

Table 4: **Ablation study on different variants of our FisheyeDistanceNet++** using the Fisheye WoodScape dataset [37]. Distances are capped at $40m$.

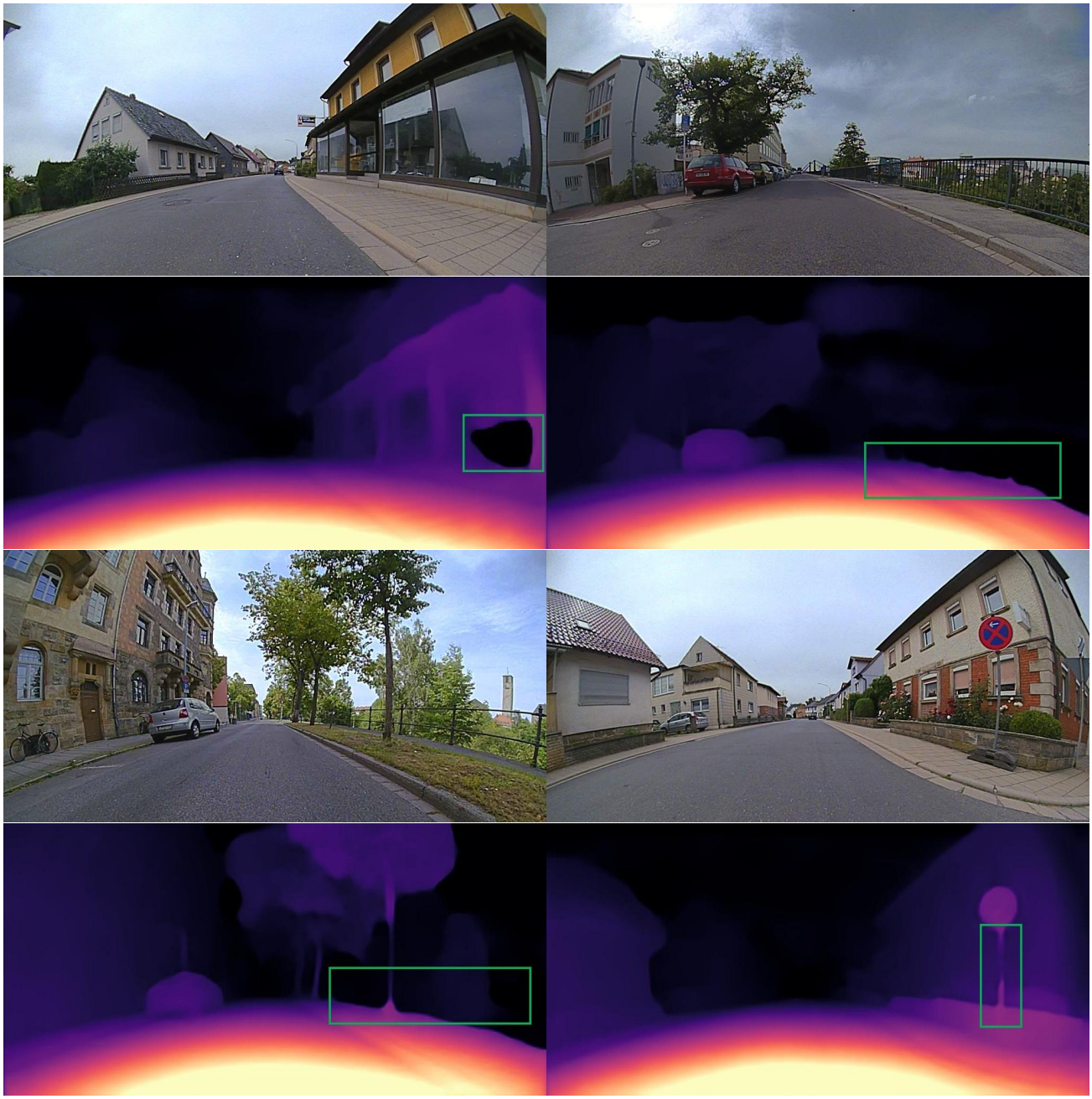


Figure 4 **Failure Cases on the Fisheye WoodScape [37] dataset.** For reflective regions, the photometric loss fails to estimate correct distances which can be seen in the 1st figure. In the following figures shown above, where boundaries are unclear the model fails to predict accurately.

Figure 5 **Qualitative results on the KITTI dataset.** Our model resolves the low textured areas such as sky i.e. infinite depth and provides sharper transition in the boundaries of objects.

Ablation study on different features

We perform an ablation study to understand the significance of different features used and tabulate in Table 2. By incorporating a generic parameterized loss function coupled with Instance Normalization [43] we could achieve significant improvements in

Method	Dataset	Encoder head	Network Resolution	Training time (hrs)
FisheyeDistanceNet [1]	K	ResNet18	640 x 192	11
	K		1024 x 320	19
	WS		512 x 256	10
FisheyeDistanceNet++	K	ResNet18	640 x 192	08
	K		1024 x 320	15
	WS		512 x 256	07
	WS	ResNet50	512 x 256	11

Table 5: **Ablation study on training times** of FisheyeDistanceNet++ on the KITTI (K) and WoodScape (WS) dataset.

accuracy over using ℓ_1 loss and Batch Normalization [44]. With our novel, self-attention layers introduced to the encoder could boost the performance of feature extraction in the ResNet18 head, which inherently helps the norm decoder to produce accurate distance estimates. We found that no single setting of normalization techniques was optimal. We could achieve state-of-the-art results with the combination of Instance Normalization layers in the encoder head and retaining the Batch Normalization in the decoder. In Table 5 we ablate the training times of our model with FisheyeDistanceNet. To provide a fair comparison to FisheyeDistanceNet, training times are reported without the contribution of self-attention layers in the encoder. All the results are reported by training on a single Titan RTX GPU. We can see the improvement by using the robust loss function as described in Section over the ℓ_1 loss.

Conclusion

In this paper, we explore the usage of a generic parameterized loss function to improve fisheye depth estimation. We demonstrate significant improvements in accuracy and training time using the fisheye dataset WoodScape [37]. We demonstrate the results on four cameras mounted around the car. We take into account the variance in the style of an image for view synthesis and ablate the importance of instance normalization over batch normalization in the training distribution. When the robust adaptive loss is paired with image representations in which variable degrees of heavy-tailed behavior occurs, such as wavelets, this adaptive training approach allows us to improve the image synthesis and neural networks self-supervised monocular depth estimation. We also test the model on KITTI dataset and obtain the state of the art results among self-supervised methods. Our motivation is to show that the loss function has to be chosen in a data-driven way instead of using the standard ℓ_1 loss. In future work, we aim to jointly optimize the loss function along with other training parameters and network topology.

Acknowledgements We want to acknowledge Valeo, particularly DAR Kronach, Germany, and Valeo Vision Systems, Ireland, for assisting the making of the WoodScape dataset. We want to thank Sumanth Chennupathi (Veoneer) for providing a detailed review.

References

[1] V. R. Kumar, S. A. Hiremath, M. Bach, S. Milz, C. Witt, C. Pinard, S. Yogamani, and P. 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)*, pp. 574–581, IEEE, 2020.

[2] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

[3] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," *arXiv preprint arXiv:1806.01260*, 2018.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.

[5] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[6] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] M. Hassaballah and A. I. Awad, *Deep learning in computer vision: principles and applications*. CRC Press, 2020.

[8] A. Briot, P. Viswanath, and S. Yogamani, "Analysis of efficient cnn design techniques for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 663–672, 2018.

[9] M. Yahiaoui, H. Rashed, L. Mariotti, G. Sistu, I. Clancy, L. Yahiaoui, V. R. Kumar, and S. Yogamani, "Fisheyemodnet: Moving object detection on surround-view cameras for autonomous driving," *arXiv preprint arXiv:1908.11789*, 2019.

[10] N. Tripathi, G. Sistu, and S. Yogamani, "Trained trajectory based automated parking system using visual slam," *arXiv preprint arXiv:2001.02161*, 2020.

[11] M. Uřičář, P. Křížek, G. Sistu, and S. Yogamani, "Soilingnet: Soiling detection on automotive surround-view cameras," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 67–72, IEEE, 2019.

[12] M. Uricár, J. Ulicny, G. Sistu, H. Rashed, P. Krizek, D. Hurych, A. Vobecky, and S. Yogamani, "Desoiling dataset: Restoring soiled areas on automotive fisheye cameras," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[13] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Near-field depth estimation using monocular fisheye camera: A semi-supervised learning approach using sparse lidar data," in *CVPR Workshop*, vol. 7, 2018.

[14] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Monocular fisheye camera depth estimation using sparse lidar supervision," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2853–2858, IEEE, 2018.

[15] 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," *arXiv preprint arXiv:2007.06676*, 2020.

[16] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning*. Springer, 2019.

[17] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3d packing for self-supervised monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust

monocular depth estimation for obstacle detection with fully convolutional networks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4296–4303, IEEE, 2016.

[19] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, 2018.

[20] S. Chennupati, G. Sistu., S. Yogamani., and S. Rawashdeh., "Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving," in *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pp. 645–652, INSTICC, SciTePress, 2019.

[21] S. Chennupati, G. Sistu, S. Yogamani, and S. A Rawashdeh, "Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[22] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 340–349, 2018.

[23] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Advances in neural information processing systems*, pp. 35–45, 2019.

[24] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE international conference on computer vision*, pp. 3828–3838, 2019.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[26] G. Sistu, I. Leang, S. Chennupati, S. Yogamani, C. Hughes, S. Milz, and S. Rawashdeh, "Neurall: Towards a unified visual perception model for automated driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 796–803, IEEE, 2019.

[27] M. Pöpperli, R. Gulagundi, S. Yogamani, and S. Milz, "Capsule neural network based height classification using low-cost automotive ultrasonic sensors," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 661–666, IEEE, 2019.

[28] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, "Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[29] K. El Madawi, H. Rashed, A. El Sallab, O. Nasr, H. Kamel, and S. Yogamani, "Rgb and lidar fusion based 3d semantic segmentation for autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 7–12, IEEE, 2019.

[30] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of NIPS*, (Long Beach, CA, USA), pp. 5998–6008, Dec. 2017.

[32] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *CoRR*, vol. abs/1406.2283, 2014.

[33] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5667–5675, 2018.

[34] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8001–8008, 2019.

[35] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *arXiv preprint arXiv:1810.06125*, 2018.

[36] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 International Conference on 3D Vision (3DV)*, pp. 11–20, IEEE, 2017.

[37] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende, *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9308–9318, 2019.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[40] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[41] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems*, pp. 9593–9604, 2019.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456, JMLR.org, 2015.

Author Biography

Varun Ravi Kumar *received a B.E. degree in 2015 and an M.Sc. degree in 2017 from TU Chemnitz, Germany. He is currently a Ph.D. student in Deep Learning for autonomous driving affiliated to TU Ilmenau and is currently working at Valeo. His research is mainly focused on the design of self-supervised perception algorithms using neural networks for self-driving cars. His expertise lies in depth and flow estimation for fisheye images and multi-task modeling. His focus also lies in semantic, motion segmentation, 2D and 3D object detection, and point cloud processing. He was awarded the* Deutschlandstipendium *for top-class international talent. He was also part of Udacity's first cohort of Self-Driving-Car Nanodegree in 2017.*

Senthil Yogamani *is an Artificial Intelligence architect and*

technical leader at Valeo. He leads the research and design of AI algorithms for various modules of autonomous driving systems. He has over 14 years of experience in computer vision and machine learning including 12 years of experience in industrial automotive systems. He is an author of over 90 publications and 60 patents with 1300+ citations. He serves in the editorial board of various leading IEEE automotive conferences including ITSC, IV and ICVES and advisory board of various industry consortia including Khronos, Cognitive Vehicles and IS Auto. He is a recipient of best associate editor award at ITSC 2015 and best paper award at ITST 2012.

Stefan Milz *received his Ph.D. degree in Physics from the Technical University of Munich. He has a strong history in professional software development and automotive. He is Managing Director of Spleenlab.ai, a self-founded machine learning company focusing on safety-critical computer vision applications (UAV, Automated Driving, Air-Taxis) deploying SLAM, sensor-fusion, perception functions into the real world. Besides, he is also a research fellow at the TU-Ilmenau. Stefan Milz is the author and co-author of more than 60 patents and more than 60 publications. He is a co-organizer of the SAIAD, OMNIVCV at CVPR, such as 3D-DLAD at ITSC and IV. In 2019 his team achieved the third rank at the NeuRIPS "Game of Drones Competition."*

Patrick Mäder *is a Professor at the Technische Universität Ilmenau, Germany and heading the endowed chair on Software Engineering for Critical Systems. His research interests include software engineering focusing on requirements traceability, requirements engineering, object-oriented analysis and design, and development of safety-critical systems. Mäder received a Diploma degree in industrial engineering and a Ph.D. degree (Distinction) in computer science from the Technische Universität Ilmenau in 2003 and 2009, respectively. He worked as a consultant for the EXTESSY AG, Wolfsburg, as a postdoctoral fellow at the Institute for Systems Engineering and Automation (SEA) of the Johannes Kepler University Linz, Austria and as postdoctoral researcher at the Software and Requirements Engineering Center at the DePaul University Chicago, USA.*

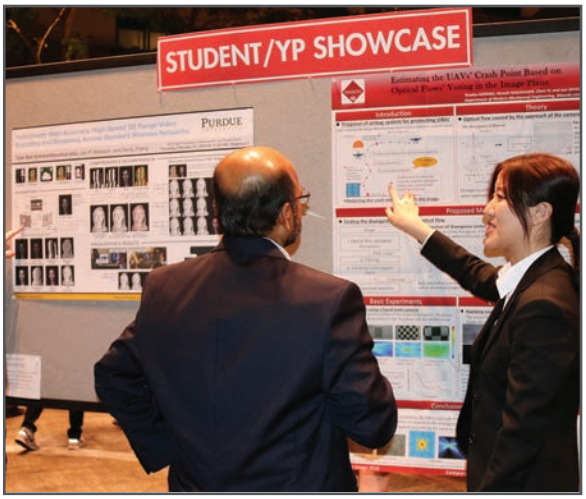
JOIN US AT THE NEXT EI!

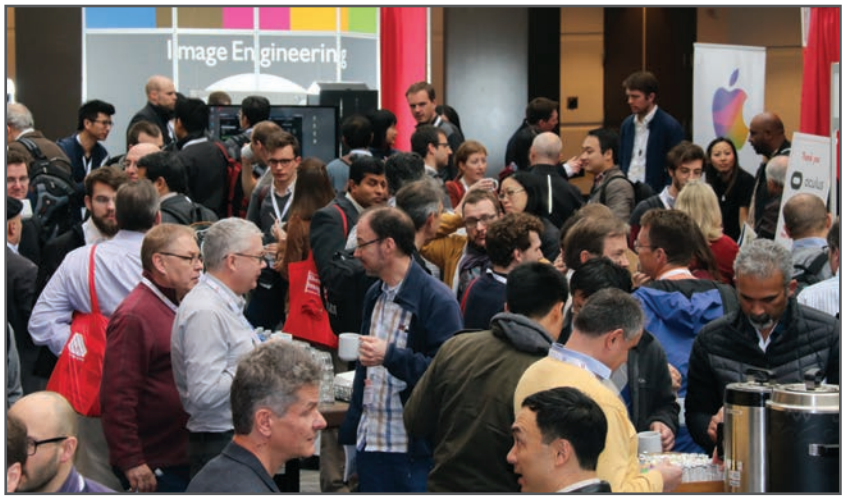
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org