https://doi.org/10.2352/ISSN.2470-1173.2020.14.COIMG-378

Sky Segmentation for Enhanced Depth Reconstruction and Bokeh Rendering with Efficient Architectures

Tyler Nuanes [*†] ***Matt Elsey*** [†] ***Radek Grzeszczuk*** [†] ***John Paul Shen*** [*]
tnuanes@andrew.cmu.edu ***matt.elsey@light.co*** ***radek@light.co*** ***jpshen@andrew.cmu.edu***
[*] ***Carnegie Mellon University, Pittsburgh PA, USA***
[†] ***Light Labs Inc., Redwood City CA, USA***

Abstract

We present a high-quality sky segmentation model for depth refinement and investigate residual architecture performance to inform optimally shrinking the network. We describe a model that runs in near real-time on mobile device, present a new, high-quality dataset, and detail a unique weighing to trade off false positives and false negatives in binary classifiers. We show how the optimizations improve bokeh rendering by correcting stereo depth misprediction in sky regions. We detail techniques used to preserve edges, reject false positives, and ensure generalization to the diversity of sky scenes. Finally, we present a compact model and compare performance of four popular residual architectures (ShuffleNet, MobileNetV2, Resnet-101, and Resnet-34-like) at constant computational cost.

key words: sky, segmentation, residual, depth, bokeh, deep learning, computer vision

Introduction

In this paper, we present two parallel stories: the training of a high-quality sky segmentation model to improve depth prediction for bokeh rendering and a comparison of residual block performance to inform how best to shrink the network.

Sky pixels may be some of the most easily recognizable for humans. Despite this, sky shape, color, and texture is quite diverse and accurate sky segmentation remains an active research problem. Developing a robust solution is important in enabling a number of applications: replacing sky in images [1, 2], performing general image color enhancement [3, 4, 5], dehazing [6], and color appearance modeling [7]. We demonstrate that sky segmentation can also improve depth of field rendering.

The depth of field effect, also known as bokeh, is an artistic photographic tool, often used to draw attention to the subject and blur out a distracting background. However, standard cell phone cameras cannot produce bokeh photos optically and must instead synthesize them computationally. Typically, this is done by estimating the depth of the captured scene and blurring the background content. Recently, machine learning approaches are used to segment the foreground from the background and blur the background without explicitly reconstructing depth. While promising, these approaches are still brittle and often work for only specific scenes, e.g., the photo must have a person in the shot. We demonstrate that our sky mask can effectively push sky to infinity in a stereo algorithm, allowing for improvements in computational bokeh rendering of many outdoor scenes.

It is known that multi-view stereo algorithms do not perform well on images that contain large texture-less areas, like sky, and instead produce unreliable or incomplete reconstructions. A number of deep learning approaches have been proposed for improving depth prediction [8, 9, 10, 11, 12]. While promising, these models are training-data dependent and thus may not achieve as generalized a quality of reconstruction as can be obtained with multi-view stereo techniques. Although Zhong *et al.* [13] work to develop a generalized deep learning model, their results may still have artifacts in the sky region. More traditional, geometry-based approaches [14] produce high-quality results but tend to be slow, often taking minutes, or even hours, to compute. We generate dense depth maps using the semi-global matching algorithm [15], a traditional algorithm that computes in seconds, but sky segmentation could also be applied as a feature in a deep learning approach to improve depth in those regions. The deep learning solution developed in this paper produces high-quality results in close to real time—inference takes only $\sim$500ms on a mobile device CPU.

In the deep learning community, sky has been included as a class in many publicly available datasets and papers. For instance, Paszke [16] reports a 95% accuracy on sky pixel classification, despite his average across 11 classes being 68%. This raises the question of whether we can train a much smaller network specifically for the task of sky segmentation? If so, which residual block architecture offers the best performance at a given computational cost?

Originally, networks hundreds of layers deep were difficult to train and degraded performance until He *et al.* introduced the residual block architecture, with Resnet-34 and Resnet-101 wherein the output of each block is $F(x)+x$. This technique enabled successful training of hundreds of nonlinear layers [17]. With the popularization of residual networks, model size has exploded to deeper architectures, resulting in the desire to reduce compute. Thus, several studies investigated efficiency in neural networks. At least two such studies, MobileNetV2 [18] and ShuffleNet [19] argued to offer competitive performance with ResNet at lower computational cost.

In this work, we offer an alternative perspective. On a small network, if we modify channels so each residual block has similar compute, which residual block offers the best performance? We introduce a small network that can be used for sky segmentation and evaluate performance of each residual block: ResNet-34-like, ResNet-101, MobileNetV2, and ShuffleNet.

We make the following contributions: a new, high-quality Light Sky dataset of 548 mobile-device images (typically 4032×3016 resolution), a weighing function to trade-off false positive and false-negative rates in binary classifiers, an application of sky

segmentation to improve dense depth reconstruction and improve bokeh rendering artifacts near sky, as well as a performance comparison of four residual block architectures.

Background

Highly-accurate sky segmentation is an area of active research within machine learning. In a sky replacement pipeline, Ta *et al.* [1] use three separate random forests to segment sky from scenes classified as "Blue", "Cloudy", or "Sunset". They identify the quality of sky segmentation as a key bottleneck. Shang *et al.* [20] use SVM classification of SLIC superpixels refined by conditional random fields. While their results are impressive, superpixels result in mis-classifications near borders and fine features. Hazirbas *et al.* [21] use depth from RGB-D sensors and Stone *et al.* [22] use a UV light sensor, both of which prove effective at improving sky segmentation quality.

Researchers at the University of Nevada developed a sky segmentation method for NASA's Mars rovers by fusing the results of k-means clustering with pixelwise segmentation from a neural network to segment grayscale images with a fine boundary. Their fusion process merges sky pixels, resulting in a single sky region [23]. Similarly, Merabet *et al.* [24] focus on fish-eye lenses with a large, continuous sky region. After generating a set of clusters classified by statistical similarity to a database of sky and non-sky regions, nearby clusters are merged, which largely excludes isolated sky regions from the final mask. Many deep learning models include sky in semantic image segmentation [25, 26, 16]. While sky segmentation accuracy is near the top of the various classes, performance can be further generalized by including challenging imagery [27]. Additionally, many pixels are often misclassified by these models since ground truth sky masks are themselves misclassified around irregular objects, openings, and thin structures.

Residual Blocks

We chose to investigate the performance of 4 popular residual blocks: ResNet-34, ResNet-101, MobileNetV2, and ShuffleNet. Our implementation of each is shown in Fig. 1.

ResNet-34 uses a stack of two 3×3 convolutions [17]. In our work, we add an extra 3×3 convolution to the Resnet-34 architecture in order to keep the number of nonlinear layers equal in each of the networks.

ResNet-101 uses 1×1 convolutions to decrease the feature space before the compute-expensive 3×3 convolution. It follows with a 1×1 convolution up to the original number of feature. The ResNet paper describes these blocks as "bottlenecks" due to their feature contraction [17].

MobileNetV2 uses separable convolutions to reduce compute. In doing so, they also introduce the concept of an "inverted bottleneck" architecture, wherein the number of features is increased before the 3×3 convolution and decreased before the residual connection [18].

ShuffleNet uses grouping over channels to reduce the computational cost of MobileNetV2 residual block. They add a shuffle operation, implemented as a transpose and reshape, in order to transfer information between the groups [19].

Datasets

According to research by Zlatesky *et al.* [28], training on both coarsely- and finely-labeled data results in performance equal to or greater than training only on a smaller dataset of finely-labelled data. We chose to follow their methodology, and train on several public datasets while also creating our own relatively small high-quality dataset.

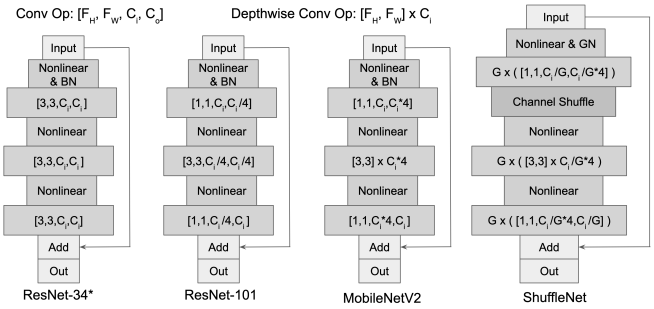
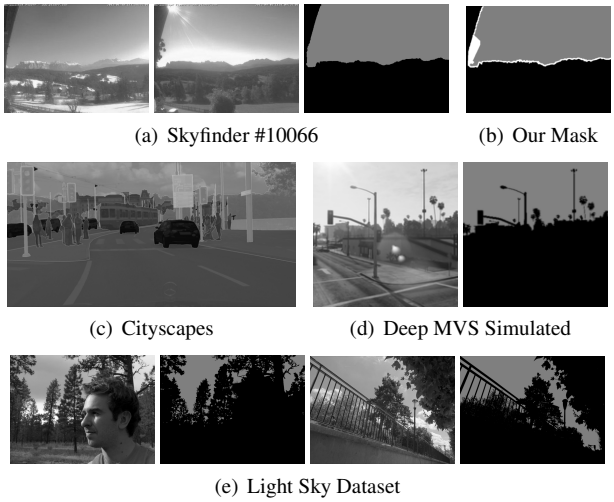


Figure 1: Our implementation of each of the 4 blocks



(a) Skyfinder #10066 (b) Our Mask

(c) Cityscapes (d) Deep MVS Simulated

(e) Light Sky Dataset

Figure 2: Sky Segmentation Datasets

While our binary model only predicts "sky" or "not sky", our ground truth has 3 classes, also including "unclassified." The "unclassified" pixels have a weight of 0 in the loss function to prevent backpropagation regardless of prediction. This is to permit our model to ignore outputs in regions where it is difficult for a human to classify between sky and not-sky, so that it can better focus on matching human performance on the task.

Public Datasets

Skyfinder [29] is comprised of 90,000 scenes from 53 static cameras [30]. Unfortunately, the provided masks are not robust to temporal variation (Figure 2(a)), such as vegetation growth or camera motion; additionally, there are thousands of images without clear imagery, due to either fog, snow, night, saturation, or clouds obscuring distant mountains. We remove these unclear scenes, resulting in a reduced dataset of about 52,000 images. We create new sky masks to include an unclassified region between the binary classes to ignore temporal artifacts (Figure 2(b)). Skyfinder typically has low resolution (often < 600 pix), poor sensor quality, and JPEG artifacts. Thus, while fine-tuning on this dataset is unwise, Skyfinder's sheer volume can improve model generalization.

Cityscapes [25] is comprised of 2048×1024 images taken in several European cities from a car's hood. They provide two

Table 1: Experiment: Including Indoor Images

Metric	548 Sky Set	1210 Full Set
Accuracy	98.7%	99.1%
mIOU	95.0%	91.0%
FN Rate	5.6%	8.5%
FP Rate	0.10 %	0.12%

datasets, a detail-oriented Fine dataset and a Coarse dataset with large unclassified regions. We convert the masks from multi-class to binary, before applying edge detection and dilation to produce an unclassified boundary. Cityscapes masks often have poor classification of skies around phone lines and foliage edges, as shown in Figure 2(c).

As shown by Richter *et al.* [31], augmenting real-world data with computer-rendered images can significantly increase accuracy. Therefore, we used RGBD scenes from the MVS-Synth dataset generated in the game *Grand Theft Auto* to produce extremely accurate labels for sky as in Figure 2(d). While this dataset has limitations in photorealism and scene diversity, the accuracy of the masks can improve model performance along fine features and object boundaries [32].

Our Light Sky Dataset

We developed the Light Sky dataset of 548 images. As we are only segmenting sky, images are segmented by hand at full resolution of 4032×3016 pixels, which allows us to more easily mask thin structures and supports creative downsampling according to training and/or model requirements (Fig. 2(e)). When downsampling is uncertain about whether a pixel is "sky" or "not sky," we set the pixel class to "unclassified." This makes a large difference in training, although doing it too aggressively results in poorer edge prediction.

Initially, we attempted to include indoor scenes in our dataset. However, performance degraded as in Table 1 where the model was trained on the 548 sky-only training set vs on a 1210 full training set (with 662 no-sky images). Textureless walls often mimic sky scenes and are thus challenging to differentiate for the network. For our application, correctly classifying walls is less important that correctly identifying small sky regions (textureless regions are invariant to blur), so we chose to include only the 548 images containing sky, which still results in an imbalanced dataset of roughly 3 non-sky : 1 sky pixels.

Network

Bokeh Model: E-Net

Our bokeh network is a slight variation of the E-Net (Efficient Net) architecture by Paszke [16], which was developed as an high-quality, compute-efficient model for generalized scene segmentation, and achieves 95% classification accuracy of sky on the CamVid dataset [33]. We add long-range skip connections at each resolution to keep high-frequency detail. Additionally, we performed a simple max pooling for downsampling and used transposed convolutions for upsampling instead of the indexing methods recommended in the original work.

Compact Comparison Net

Our compact comparison network is displayed in Fig. 3. It has has 45 activations and uses dilated convolutionsat the lowest resolution so that the full context for that stage is 33×33 pixels.

Figure 3: Our compact architecture. "Resid" stands for "residual blocks." "DR" stands for "dilation rate."

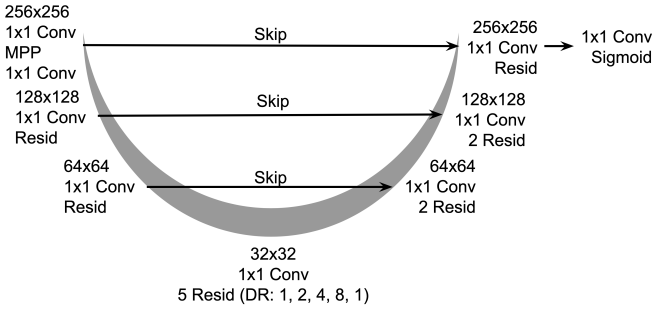


Table 2: Experiment: Median Frequency Balancing

Metric	Modified MFB	Original MFB
Accuracy	99.4%	99.7%
mIOU	97.4%	97.7%
FN Rate	2.7%	0.27%
FP Rate	0.017%	0.28%

Each 3×3 convolution is padded and each 1×1 convolution (except the final one) is followed by a prelu activation, which helps prevent dead weights and boost performance at the risk of overfitting. Each group in the ShuffleNet implementation has a separate prelu slope parameter. As shown in Fig. 1, the start of each residual block has a nonlinear activation followed by a batch normalization, as proposed by He *et al.* [34]. In ShuffleNet where a GroupNorm replaces BatchNorm.

Finally, we chose to place a Mean Pyramid Pooling (MPP) [35] layer at the start of the network to increase context at each pixel. We downsample the image in stages down to a resolution of 4×4, bilinearly interpolate each downsampled image up to full resolution, and concatenate them together. A 1×1 convolution compresses the features. We intend for this layer to provide context to the network at full resolution.

Loss Weights

Bokeh Model

For bokeh rendering, an extremely low false positive rate is important to prevent pushing foreground objects, such as a blue t-shirt, to infinity. In contrast, a moderately higher false negative rate can be tolerated as depth will simply rely on the stereo algorithm and should not degrade from the baseline.

To trade off between false positives and false negatives, we chose to modify Median Frequency Balancing (MFB), a technique used to train with unbalanced classes wherein the weight for each class is the median number of pixels in all classes over the number of pixels in that class [36]. In order to satisfy both our desire to balance the classes and favor false negatives over false positives, we choose loss weights for four classes: True Positive (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Specifically, the weight for TP is $NC/(TP+1)$ where NC is the number of classified pixels (the $+1$ is for numerical stability) and the weight for FN is $NC/(TN+1)$. The weight for FN pixels is $NC/(TP+FN+1)$ so these pixels will be weighted less than TP pixels. Finally, the weight for FP pixels is $NC/(FP+FN+1)$ so that there is a tradeoff between the number of false positives and false negatives. We feed these weights

Table 3: Filters, Computational Cost, & Parameters
MFLOP refers only to convolution compute
E-Net reports full network compute.

Res.	RN-34	RN-101	MNV2	SN	E-Net
256x256	4	8	4	8	16
128x128	4	16	4	8	64
64x64	4	16	8	8	128
32x32	4	32	8	8	—
MFLOP	72	66	69	81	1090*
Params	5,794	10,657	9,386	7,027	370k

to the sigmoid cross entropy loss function.

We compare the observed training results in Table 2 with our modification to MFB. Naïvely, similar statistical results to our loss function can be achieved by raising the threshold after softmax on the original MFB model. However, raising the threshold simply leads to poorer classification along boundaries where uncertainty is generally highest, whereas high-confidence false positives remain misclassified.

Comparison Model

In the comparison model, we merely weigh the FP and FN classes $2\times$ heavier than TP and TN classes, using the equations descried above. We do this as a naïve way to favor classification accuracy over entropy, and since having a low false positive rate is inconsequential for a general comparison study.

For each residual block architecture, we modify the number of filters at each resolution in our network in order to keep the floating point operations roughly constant between all networks. These details are recorded in Table 3. ShuffleNet has 4 groups, 2 channels per group.

Training

Bokeh Model

We utilize images from the Skyfinder, Cityscapes, MVS-Synth, and Light Sky datasets to create our training dataset. We initially downsample our images to 300×300 and 600×600 before training. To augment the data, we performs random rotations in the interval of $\pm 10^{\circ}$ and random crops. We further modify the images by hue, brightness, saturation, and contrast. We do this at 256×256 and 512×512 resolutions to allow our network to generalize to different zooms. This is done per epoch.

Training takes part in three stages. Initially, we train on images from all 256×256 samples for 175 epochs. We next train on all 512×512 samples for 120 epochs in order to encourage greater generalization. Finally, we fine-tune on only the 256×256 images from our Light Sky dataset for 50 epochs. Learning rate is kept at 10^{-3}, loss is weighted sigmoid cross-entropy, and the optimizer is ADAM.

Comparison Model

For the residual block study, we do not worry about generalization performance to different real-world scenes than in our Light Sky dataset, so we choose to only train and test on our dataset. We perform an 80%/20% split, resulting in 109 test images. Since we do not have a validation set, we do not tune any model parameters.

During training, we randomly augment the image hue, brightness, saturation, and contrast. Additionally, we geometrically transform the data through random rotation, horizontal flips, and vertical flips. To increase performance on different magnifications, we downsample the images to 1024×1024, take a random crop larger than 256×256, and then downsample that crop to 256×256. This is done per epoch. Learning rate, optimizer, and loss are the same as above.

Figure 4: Sky segmentation on a test set

Metric	Fine (Train)	Coarse (Test)	Light (Train)
Accuracy	99.89%	99.90%	99.4%
mIOU	94.5%	95.0%	97.5%
FN Rate	5.1%	3.0%	2.8%
FP Rate	0.011%	0.039 %	0.045%

Table 4: Inference on Cityscapes and Light Sky

Results

Bokeh Model

Sky Segmentation

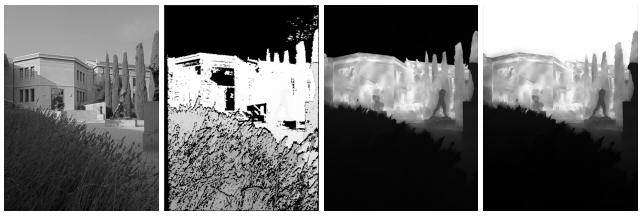
Due to the small size of our dataset, we evaluate our generalization performance using Cityscapes Coarse as a test set after training on Cityscapes Fine, as shown in Table 4. We report training statistics on our own dataset for comparison. We include performance visualization of Light Sky test images in Fig. 4.

Bokeh Rendering

We apply our sky segmentation model to improve the depth maps produced by the dense depth reconstruction algorithm. Figure 5 shows the comparison of the depth maps generated by the public COLMAP algorithm [37] and by Light Lab's dense stereo reconstruction algorithm with, and without, sky segmentation. As can be seen in the figure, using sky segmentation corrects the depth map by pushing sky pixel depth to the background.

When sky pixels are pulled to the foreground by matching noise in the cost volume, harsh edges result as in 5. However, with the correct depth of sky pixels, the bokeh renderings more naturally mimics a pleasing depth of field (DOF).

Occasionally, our model fails in textureless regions, mainly



(a) In order: Image, COLMAP Depth, Depth w/o Sky, & Depth w/ Sky

(b) In order: DOF w/o Sky, DOF w/ Sky, Sky Mask

Figure 5: Bokeh results improve with sky segmentation.

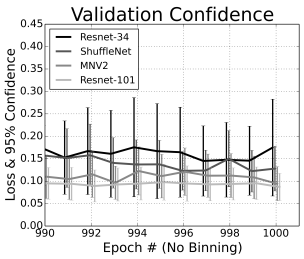


Figure 6: Validation Loss Confidence

on indoor walls. However, the impact is typically positive on bokeh as most walls are closer to "infinity" than foreground. When the wall is in the foreground, blurring of the textureless region results in a soft failure that will not degrade the experience for most users. False negative failures result in bokeh quality at the baseline.

Comparison Model

Confidence

Since we only validate on 109 images, our confidence is limited, as shown in Fig. 6. Unfortunately, all the confidences intervals overlap, limiting the conclusions we can draw.

Residual Block Comparison

The validation and training results are displayed in Fig. 7, averaged over a bin size of 10 epochs, which is why we have termed the mIOU plots as "Mean mIOU" on the y-axis. The purpose of averaging is simply to smooth the plots for visualization.. It appears that ShuffleNet trains faster and overfits to a greater extent than either MobileNetV2 or ResNet-101. MobileNetV2 has the greatest amount of noise during training and may benefit from a lower learning rate. ResNet-34 appears to be a relatively poor choice, at least for sky segmentation, compared to the other networks here.

Conclusion & Future Work

We introduce a new dataset with 548 high-quality sky masks, demonstrate its usefulness in generating a high-fidelity sky segmentation model, apply the model to improve depth reconstruction and enhance bokeh rendering, and investigate which residual block may be most appropriate for shrinking the model.

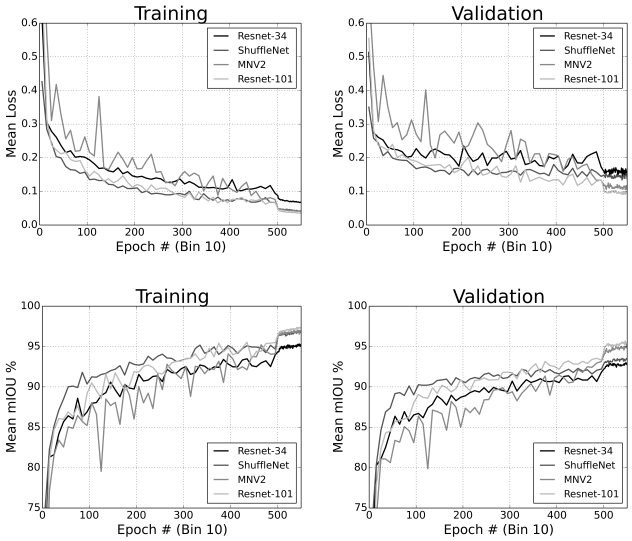


Figure 7: mIOU and Loss for Each Residual Block

In future work, it would be relevant to address two main limitations. First, as most of our dataset and the publicly available datasets were daytime outdoor scenes, we could include night images to improve performance on dark scenes. Second, it may benefit the community to establish more conclusive results on the residual block comparison. To do so, we may shrink the network further to degrade performance, and observe which residual block drops off first. As our model, in terms of context, is already near a minimal model, we may instead perform this same experiment on a more challenging segmentation task.

References

[1] Litian Tao, Lu Yuan, and Jian Sun, "Skyfinder: Attribute-based sky image search," *American Transactions on Graphics (TOG)*, vol. 28, no. 3, 2009.

[2] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, and Ming-Hsuan Yang, "Sky is not the limit: Semantic-aware sky replacement," *ACM Transactions on Graphics (Proceeds SIGGRAPH)*, vol. 35, no. 4, 2016.

[3] R. Bergman and H. Nachlieli, "Perceptual segmentation: Combining image segmentation with object tagging," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1668–1681, 2011.

[4] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Quian, and James Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, vol. 33, no. 4, 2014.

[5] Bahaman Zafarifar and Peter H.N. de With, "Blue sky detection for content-based television picture quality enhancement," *Digest of Technical Papers International Conference on Consumer Electronics*, 2007.

[6] Yingchao Song, Haibo Luo, Junkai Ma, Bin Hui, and Zheng Chang, "Sky detection in hazy images," *Sensors*, vol. 18, no. 4, 2018.

[7] Yulia Gryaditskaya, Tania Pouli, Erik Reinhard, and Hans-Peter Seidel, "Sky based light metering for high dynamic range images," *Computer Graphics Forum*, vol. 33, no. 7, pp. 61–69, 2014.

[8] Fatma Guney and Andreas Geiger, "Displets: Resolving stereo ambiguities using object knowledge," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Baschrach, and Adam Bry, "End-to-end learning of geometry and context for deep stereo regression," *International Conference on Computer Vision (ICCV)*, 2017.

[10] Richard Szeliski and Ramin Zabih, "An experimental comparison of stereo algorithms," *Lecture Notes in Computer Science*, vol. 1883, 2000.

[11] Q. Yang, C. Engels, and A. Akbarzadeh, "Near real-time stereo for weakly-textured scenes," *Proceedings of the British Machine Vision Conference*, pp. 72.1–72.10, 2008.

[12] Yi Zhang, Weichao Qiu, Qi Chen, Xiaolin Hu, and Alan Yuille, "Unrealstereo: Controlling hazaradous factors to analyze stereo vision," *International Conference on 3D Vision (3DV)*, 2018.

[13] Yiran Zhong, Yuchao Dai, and Hongdong Li, "Self-supervised learning for stereo with self-improving ability," *arXiv:1709.00930*, 2017.

[14] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf, "Casual 3d photography," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 234:1–234:15, 2017.

[15] Heiko Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[16] Adam Paszke, Abbhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv:1606.02147v1*, 2016.

[17] Shaoqing Ren aiming He, Xiangyu Zhang and Jian Sun, "Deep residual learning for image recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen Mark Sandler, Andrew Howard, "Mobilenetv2: Inverted residuals and linear bottlenecks," *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[19] Mengxiao Lin Jian Sun Xiangyu Zhang, Xinyu Zhou, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *Computer Vision and Pattern Recognition (CVPR)*.

[20] Yuanyuan Shang, Ge Li, Zhong Luan, Xiuzhuang Zhou, and Guodong Guo, "Sky detection by effective context inference," *Neurocomputing*, vol. 208, pp. 238–248, 2016.

[21] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers, "Fusenet: Incorproating depth into semantic segmentation via fusion-based cnn architecture," *Asian Conference on Computer Vision (ACCV)*, 2016.

[22] Thomas Stone, Michael Mangan, Paul Ardin, and Barbara Webb, "Sky segmentation with ultraviolet images can be used for navigation," *Robotics: Science and Systems*, pp. 395–404, 2014.

[23] Ali Pour Yazdanpanah, Emma E. Regentovam, Ajay Kumar Mandava, Touqeer Ahmad, and George Bebis, "Sky segmentation by fusing clustering with neural networks," *International Symposium on Visual Computing (ISVC)*, pp. 663–672, 2013.

[24] Houssam Bourr, Youssef El Merabet, Rochdi Messoussi, Ibtissam Benmiloud, and Yassine Ruichek, "An efficient sky detection algorithm from fisheye image based on region classification and segmentation analysis," *Transactions on Machine Learning and Artifical Intelligence (TMLIA)*, vol. 5, no. 4, pp. 714–724, 2017.

[25] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Scene parsing through ade20k dataset," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] Cecilia La Place, Aisha Urooj Khan, and Ali Borji, "Segmenting sky pixels in images," *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] Aleksandar Zlateski, Ronnachai Jaroensri, Prafull Sharma, and Frdo Durand, "On the importance of label quality for semantic segmentation," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1479–1487, 2018.

[29] Radu P. Mihail, Scott Workmanm, Zach Bessinger, and Nathan Jacobs, "Sky segmentation in the wild: An empirical study," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[30] Nathan Jacobs, Walker Burgin, Nick Fridrich, Austin Abrams, Kylia Miskell, Bobby H. Braswell, Andrew D. Richardson, and Robert Pless, "The global network of outdoor webcams: Properties and applications," *International Conference on Advances in Geographic Information Systems (ACM/SIGSPATIAL GIS)*, pp. 111–120, 2009.

[31] Stephan R. Richter, Vibhav Vineet andStefan Roth, and Vladlen Koltun, "Playing for data: Ground truth from computer games," *European Conference on Computer Vision (ECCV)*, 2016.

[32] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang, "Deepmvs: Learning multi-view stereopsis," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[33] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla, "Segmentation and recognition using structure from motion point clouds," *European Conference on Computer Vision (ECCV)*, pp. 44–57, 2008.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," *European Conference on Computer Vision (EECV)*, 2016.

[35] Shaoqing Ren Kaiming He, Xiangyu Zhang and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

[36] David Eigen and Rob Fergus, "Predicting depth, surface normals, and semantic labels with common multi-scale convolutional architecture," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[37] Johannes Lutz Schönberger and Jan-Michael Frahm, "Structure-from-Motion Revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

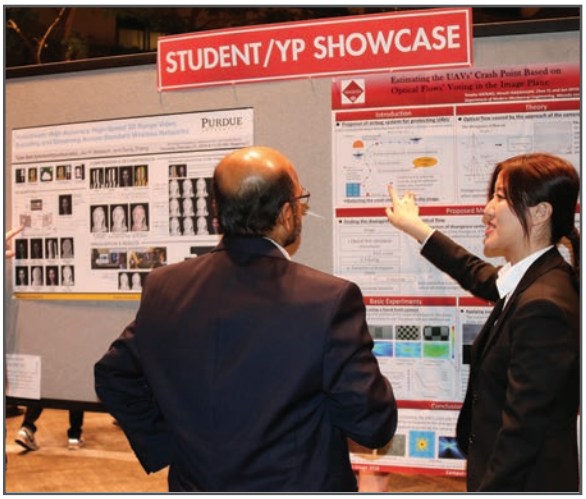
JOIN US AT THE NEXT EI!

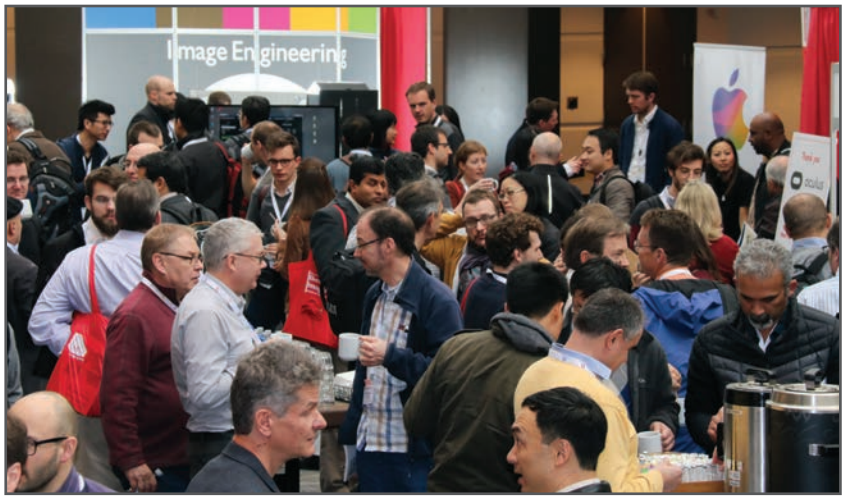
IS&T International Symposium on

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