Pre-print of article that will appear in 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 1621-1625, doi: 10.1109/ICIP40778.2020.9191168.

The published article is available on https://ieeexplore.ieee.org/document/9191168

PARALLAX MOTION EFFECT GENERATION THROUGH INSTANCE SEGMENTATION AND DEPTH ESTIMATION

Allan Pinto[1] *Manuel A. Córdova*[1] *Luis G. L. Decker*[1] *Jose L. Flores-Campana*[1] *Marcos R. Souza*[1]
Andreza A. dos Santos[1] *Jhonatas S. Conceição*[1] *Henrique F. Gagliardi*[2]
Diogo C. Luvizon[2] *Ricardo da S. Torres*[3] *Helio Pedrini*[1]

[1]Institute of Computing, University of Campinas (UNICAMP), Campinas, SP, Brazil, 13083-852
[2]AI R&D Lab, Samsung R&D Institute Brazil, Campinas, SP, 13097-160, Brazil
[3]NTNU – Norwegian University of Science and Technology, Ålesund, Norway.

ABSTRACT

Stereo vision is a growing topic in computer vision due to the innumerable opportunities and applications this technology offers for the development of modern solutions, such as virtual and augmented reality applications. To enhance the user's experience in three-dimensional virtual environments, the motion parallax estimation is a promising technique to achieve this objective. In this paper, we propose an algorithm for generating parallax motion effects from a single image, taking advantage of state-of-the-art instance segmentation and depth estimation approaches. This work also presents a comparison against such algorithms to investigate the trade-off between efficiency and quality of the parallax motion effects, taking into consideration a multi-task learning network capable of estimating instance segmentation and depth estimation at once. Experimental results and visual quality assessment indicate that the PyD-Net network (depth estimation) combined with Mask R-CNN or FBNet networks (instance segmentation) can produce parallax motion effects with good visual quality.

Index Terms— Parallax Motion Effect; Instance Segmentation; Depth Estimation; Inpainting; Deep Learning

1. INTRODUCTION

Stereo vision [1, 2] is a growing topic in computer vision (CV) due to the innumerable opportunities this technology offers for developing modern applications, such as virtual and augmented reality systems [3, 4], entertainment [5], autonomous robot navigation [6], and medicine [7]. While common tasks and problems (e.g., image classification) in computer vision are concerned with the development of algorithms for identifying, understanding and analyzing 2D images, the stereo vision and 3D reconstruction tasks aim the design of models and algorithms able to infer 3D properties from objects presented in a scene and then reconstruct the spatial relationship between them.

Although several advances have been reported in the literature for 3D reconstruction and stereo vision problems, the understanding of 3D information in a scene from images is still an open challenge due mainly to inherent ill-posing nature of estimating depth information from pixels based on their intensity values [8]. To overcome these limitations, the use of machine learning techniques and data from 3D sensors has been proposed to minimize the errors in the inference related to ambiguity issues during 3D reconstruction [8, 9, 10]. In this context, CV research community has spent efforts to provide good quality data from 3D imaging sensor to enable accurate CV-based machine learning models for some specific tasks, such as autonomous driving. Examples of good quality data for this task include Cityscapes and KITTI datasets [11, 12].

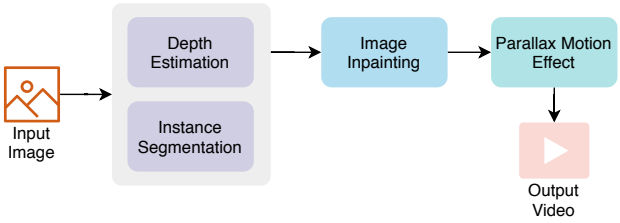

Fig. 1: Overview of the proposed methodology for generating parallax motion effect from images.

Among the techniques available for 3D reconstruction and visualization, depth estimation in binocular vision systems, disparity estimation from stereo images and motion parallax estimation from a sequence of images are certainly the most promising techniques for achieving this objective [13, 14, 15]. In particular, the spatial perception stimulus generated by motion parallax has propelled several theoretical studies in the areas of visual perception and psychology that seek physiological explanations in humans, towards establishing the neurological bases of our visual ability [16, 17, 18].

Motion parallax [19,20,21,22,23] provides an important monocular depth cue raised from the relative velocity between the objects and the observer. In motion parallax, objects near the observer move faster than objects that are farther away. Since this motion is considered a rich source of 3D information [24, 25], several computer vision studies have recently proposed the use of motion parallax to enrich depth perception in tasks involving 3D scene reconstructions [5, 26, 27].

Based on evidences that motion parallax can potentially enrich human depth visual perception, this research aims to devise algorithms and methods to automatically generate motion parallax effect from a single image, in order to provide a 3D immersion experience to the user with devices equipped with a general-purpose RGB cam-

We thank Samsung R&D Institute Brazil for the financial support. This work was funded by Samsung Eletrônica da Amazônia Ltda., through the project "Parallax Effect", within the scope of the Informatics Law No. 8248/91. Authors are grateful to Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES (Finance Code 001), National Council for Scientific and Technological Development – CNPq (grant #309330/2018-1) and São Paulo Research Foundation – FAPESP (grants #2016/50250-1, #2017/12646-3 and #2019/16253-1).

(a) Depth map. (b) Nearest object. (c) Farthest object.

Fig. 2: Example of detected objects (squirrel and stone) sorted according to the average depth value around their center of mass.

era. Since there is no good quality dataset available for this task, this work aims to answer the following research questions: (i) Could CV-based machine learning models, originally proposed for depth estimation problem, be adapted to generate motion parallax effects, with a good visual quality? (ii) Are the state-of-the-art methods for instance segmentation able to generalize enough to enable their use in scenarios whose image acquisition is different from those considered in the training time? To answer these questions, we proposed a method for parallax motion effect that takes advantage of recent developments for instance segmentation and depth estimation problems, as illustrated in Fig. 1.

The remaining of this text is organized as follows. Section 2 introduces the proposed method for motion parallax effect generation. Section 3 presents and discusses the achieved results. Finally, Section 4 provides our conclusions and future research venues.

2. PROPOSED METHOD

This section presents the proposed method to generate a video considering the use of parallax motion concepts to move objects in an image. The proposed method was designed to produce parallax motions, considering three types of movements: zoom in, the left and right. Regardless of the movement type considered, we propose the use of a simple speed model to determine the relative position of the foreground and background components at a given instant t. The following sections discuss the main steps of our method.

Merging the Results of Instance Segmentation and Depth Estimations Networks. This step aimed to join the results from instance segmentation and depth estimation methods to capture the scene semantic context associated with spatial relations among the objects in the scene. First, we used an instance segmentation algorithm to find the boundary of objects in an image I. Next, we applied a depth estimation method to find the position of these objects on the z-axis. Finally, we sorted them to get the nearest object to the camera.

To sort the objects according to their z-axis positions, firstly, we computed a binary mask for all segmented objects. Next, we used these masks to compute the center of mass of the objects. Finally, we averaged the depth values considering a 5×5 kernel size around the center of mass, which were used to sort the objects' masks (see Fig. 2). The mask with the highest disparity value (nearest to the camera) was used to: (i) isolate the nearest object, which was clipped and pasted into a new image with a transparent background, hereafter named as foreground component; and (ii) to remove the nearest object from the original image to produce a new image without the foreground object, hereafter named as background component.

Refinement of Background and Foreground Components and Image Inpainting. After finding the *nearest mask*, which is used to produce the background and foreground components, the next step aims to perform a post-processing upon this mask to remove erroneous pixels in both background and foreground components, caused by segmentation errors. In summary, this step is essential: (i) to prevent that the inpainting method fills out the holes in the background image using the objects' pixels left in the image, after the object removal; and (ii) to enhance the boundary of the objects that comprise the foreground component by removing pixels belonging to background.

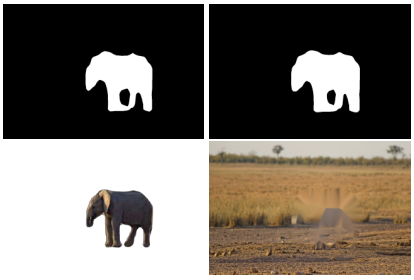

Fig. 3: Example of refined masks (first row) and background and foreground components (second row). The top-left image illustrates a refined mask used to produce a foreground component (bottom left), while the bottom-left image shows a refined mask used to produce the background component (bottom right).

To refine the foreground component, first, we applied a Gaussian blur, considering a kernel size of 7×7 upon the *nearest mask*. Next, we threshold the smoothed mask to come up with a new one, which was used to produce the refined foreground component. On the other hand, to refine the background component, we performed a dilation of the nearest mask considering a kernel size of 11×11 towards enlarging the region of interest coded into the binary mask, and thus come up with a coarse object's delimitation to ensure removal of all pixels that belong to the object. Finally, we used the Telea's [28] inpainting algorithm to fill out the hole left in the image after the object removal (see Fig. 3).

Speed Model for Background and Foreground Components. The parallax motion is simulated through a simple and efficient method to compute the movement of the background and foreground components. According to the concepts of motion parallax, the object near to the camera moves faster than objects far from the camera. In this initial solution, we simulated this effect by considering the use of finite arithmetic sequences, with n elements, for both components but with different constant terms, as shown in Eq. 1:

$$\begin{aligned} fore_n &= fore_1 + (n-1) \times c_{fore} \\ back_n &= back_1 + (n-1) \times c_{back} \end{aligned} \quad (1)$$

where $fore_1$ and $back_1$ are the foreground and background components used to produce the 1-st frame of a video containing parallax motion effects, $fore_n$ and $back_n$ are the foreground and background components, respectively, used to produce the n-th frame, and the coefficients c_{fore} and c_{back} are constant terms that defines the speed movement. In this context, each value of these sequences is used as a sum factor to compute the 2D geometric transformation of the background and foreground component, regardless the movement type. In such circumstances, small constant terms produce movements slower than movements produced with larger constant terms. As a result of this process, we ended up with n background and n foreground images, which were blended to generate a video clip containing the parallax motion effect.

Enhancing the Quality of Parallax Motion Generation. We adopted three strategies to enhance the visual quality of parallax motion effects, as follows:

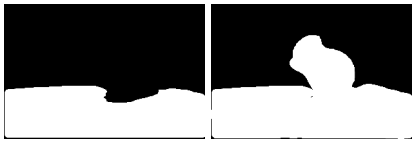

Fig. 4: Example of a background image without any post-processing (left image) and its refined version (right image).

- **Small object filtering.** This step aims to filter out small objects that are irrelevant to the parallax motion effect generation. The criterion adopted to define the minimum size of the objects in the image corresponds to relative area of objects, compared to the area of the largest object in the image. All objects with a relative area smaller than 5% are added to the background layer.
- **Joining near objects.** To mitigate the effect of possible depth estimation errors, we devised an algorithm to join near objects considering a relative tolerance between their distance. After computing the average of depth values for each segmented objects, we sorted the objects according to their distances and then we joined pair of objects with a relative distance up to 20%. This strategy is useful to generate parallax motion effect for images without a clear object of interest.
- **Two-layered scene.** We also proposed a procedure to join objects from different classes, but that should be in the foreground component. Fig. 4 illustrates an example in which the squirrel and stone should be considered as a foreground component. However, due to bad depth estimation, both "objects" are far apart from each other. To overcome this problem, we slice the scene into two layers, according to the median value of depth values. In this context, an object is classified as being of the background layer if the average of their depth value is smaller than median value of the whole depth map. Otherwise, the object is classified as being of the foreground layer.

3. EXPERIMENTS AND RESULTS

This section presents the datasets and evaluation protocols used to validate the proposed method. We report the quality of obtained results considering metrics adopted in each category of algorithms used in this work, i.e., instance segmentation and depth estimation.

3.1. Datasets and Metrics

In this section, we briefly describe the datasets and evaluation protocols adopted in this work to validate our method.

COCO 2017 Dataset. This dataset was proposed to be used in three tasks in the COCO 2017 Place Challenge: scene parsing, scene instance segmentation, and semantic boundary detection [29]. In this work, we used the data available for the scene instance segmentation, which aims to segment an image into object instances.

KITTI 2015 Dataset. The KITTI dataset [12] was built considering an autonomous driving platform equipped with several acquisition sensors for collecting a wide gamma of information including stereo images (grayscale and color), optical flow estimations, visual odometry, 3D points estimations, geographic localization, among others.

Parallax60 Dataset. This dataset contains sixty images collected over Internet, which comprises high-quality and ultra-high-definition (UHD) images (from $3,840 \times 2,160$ to $8,192 \times 5,461$) with different backgrounds (see Fig. 5). Most of the images are natural scenes with various types of vegetation, which makes this dataset the hardest one to generate parallax motion effect.

Fig. 5: Examples of images from the Parallax60 dataset.

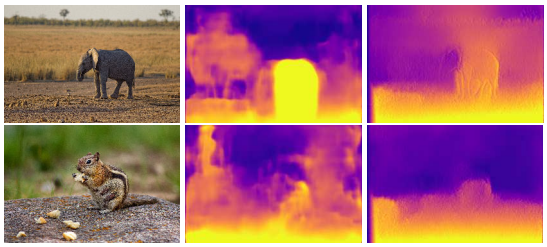

Fig. 6: Comparison between methods for depth estimation. The second and third columns illustrate the results obtained by Semantic-Monodepth and PyD-Net networks, respectively.

Evaluation Metrics. To measure efficiency aspects of our method, we consider both the processing time and the disk usage (in MB). We used the Linux *time* command for measuring processing time since this tool can be applied to all evaluated methods, regardless the programming language. Regarding the efficacy aspects, we performed a visual inspection to measure the quality of a video containing a parallax motion effect due to inherent subjectivity present in this task.[1]

3.2. Comparison of Methods for Depth Estimation and Instance Segmentation

This section presents the performance results for the PyD-Net and Semantic-Monodepth networks considering the use of models provided by the authors. We measured the effectiveness of these models upon the KITTI dataset, with confirmed the results reported by the authors [30]. Considering efficiency aspects, the Semantic-Monodepth network spent 21.53 sec./image, whereas the PyD-Net network spent 12.5 sec./image. Fig. 6 presents a comparison among depth maps obtained with Semantic-Monodepth and PyD-Net networks, from which we could observe that both networks were able to detect the object of interest as a foreground object, but also produced depth maps with several inconsistencies.

In the context of parallax motion effect generation, segmentation methods also play a crucial role in the overall quality of parallax videos. We investigated three networks for instance and semantic segmentation that have different requirements in terms of processing system requirements (see Table 1).

Fig. 7 shows visual results achieved with the segmentation methods evaluated in this work. From this experiment, we observed that

[1] A supplementary material with more examples and videos containing parallax motion effects generated by our method can be found in `https://allansp84.github.io/motion-parallax/` (As of May 2020).

Fig. 7: Examples of segmentation results achieved by segmentation methods. First three rows present the results obtained by the Mask R-CNN (first row), Semantic-Monodepth (second row), and FBNet (third row) networks. The fourh and fifth rows present the results achieved by the Semantic-Monodepth and FBNet networks, respectively. In this example, the Mask R-CNN network was not able to segment the fox.

Fig. 8: Example of a parallax motion effect before (top row) and after (bottom row) joining near objects considering their relative distance.

Table 1: Model size (in MB) and latency (in sec./image) of segmentation methods upon the Parallax60 dataset.

Method	Model Size	Latency
Mask R-CNN (ResNet101)	483.0	26.11
Semantic-Monodepth	823.8	18.03
FBNet	26.70	13.85

Mask R-CNN was not able to find any object, for several input images. In total, Mask R-CNN was able to produce at least one mask for $39/60$ images. In turn, the Semantic-Monodepth and FBNet networks produced masks for all images on the Parallax60 dataset. In terms of quality of parallax videos, in general, both Mask R-CNN and FBNet produced better parallax motion effects, in comparison to Semantic-Monodepth network.

3.3. Improving Parallax Motion Effects

This section presents two ideas to improve parallax motion effects. The first strategy concerns with joining near objects, according to their relative distance. For all experiments, we considered a maximum relative distance, for merging two objects, up to 20%. From the experimental results and visual quality assessments, we observed that poor quality achieved by the segmentation methods is due to the lack of a clear object of interest. In general, these errors occur in images such as natural, landscape, and indoor images. Fig. 8 shows examples in which the refinement of foreground and background components (see Sec. 2) improved the visual quality of parallax motion effects significantly.

4. CONCLUSIONS

This work presented a method for parallax motion effect generation, considering the use of instance segmentation and depth estimation methods. The methods were evaluated in terms of their ability to segment instances towards delimiting objects, and infer distances between objects in the scene, considering landscape and natural images. For the depth estimation task, achieved results suggest that the PyD-Net network provides good depth estimations at an affordable computational cost, in comparison to Semantic-Monodepth network. For the instance segmentation task, the Mask R-CNN presented better qualitative results than all those networks evaluated in this work. However, this network is time consuming and requires about 0.5GB of storage. A low-cost alternative for this task is the FBNet network, which presented similar results at low computational costs, in terms of storage footprints, requiring 30MB of storage. Finally, some future research venues include the combination of efficient depth and instance segmentation networks in a unified architecture in order to have a fast and lightweight model.

5. REFERENCES

[1] Sangwon Kim, Jaeyeal Nam, and Byoungchul Ko, "Fast depth estimation in a single image using lightweight efficient neural network," *Sensors*, vol. 19, no. 20, pp. 4434, 2019.

[2] Peiliang Li, Tong Qin, et al., "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 646–661.

[3] F. Okura, Y. Nishizaki, T. Sato, N. Kawai, and N. Yokoya, "Motion Parallax Representation for Indirect Augmented Reality," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, Sep. 2016, pp. 105–106.

[4] S. Pathak, A. Moro, H. Fujii, A. Yamashita, and H. Asama, "Virtual reality with motion parallax by dense optical flow-based depth generation from two spherical images," in *2017 IEEE/SICE International Symposium on System Integration (SII)*, 2017, pp. 887–892.

[5] J. Thatte, J. Boin, H. Lakshman, and B. Girod, "Depth Augmented Stereo Panorama for Cinematic Virtual Reality with Head-Motion Parallax," in *IEEE International Conference on Multimedia and Expo (ICME)*, July 2016, pp. 1–6.

[6] Tanapol Prucksakorn, Sungmoon Jeong, and Nak Young Chong, "A Self-Trainable Depth Perception Method from Eye Pursuit and Motion Parallax," *Robotics and Autonomous Systems*, vol. 109, pp. 27 – 37, 2018.

[7] H. Liao, T. Inomata, I. Sakuma, and T. Dohi, "3-D Augmented Reality for MRI-Guided Surgery Using Integral Videography Autostereoscopic Image Overlay," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 6, pp. 1476–1486, June 2010.

[8] S. Choi, D. Min, B. Ham, Y. Kim, C. Oh, and K. Sohn, "Depth Analogy: Data-Driven Approach for Single Image Depth Estimation Using Gradient Samples," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5953–5966, Dec 2015.

[9] Ali Shahnewaz and Ajay K Pandey, "Color and Depth Sensing Sensor Technologies for Robotics and Machine Vision," in *Machine Vision and Navigation*, pp. 59–86. Springer, 2020.

[10] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards Real-Time Unsupervised Monocular Depth Estimation on CPU," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 5848–5854.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3213–3223.

[12] Moritz Menze, Christian Heipke, and Andreas Geiger, "Object Scene Flow," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 60–76, 2018, Geospatial Computer Vision.

[13] Zewei Cai, Xiaoli Liu, Giancarlo Pedrini, Wolfgang Osten, and Xiang Peng, "Light-Field Depth Estimation Considering Plenoptic Imaging Distortion," *Optics Express*, vol. 28, no. 3, pp. 4156–4168, 2020.

[14] Seongwook Yoon, Taehyeon Choi, and Sanghoon Sull, "Depth Estimation from Stereo Cameras through a Curved Transparent Medium," *Pattern Recognition Letters*, vol. 129, pp. 101–107, 2020.

[15] Rostam Affendi Hamzah, MGY Wei, NSN Anwar, SF Abd Gani, AF Kadmin, and KAA Aziz, "Depth Estimation Based on Stereo Image Using Passive Sensor," in *Advances in Electronics Engineering*, pp. 127–136. Springer, 2020.

[16] Brian Rogers and Maureen Graham, "Motion Parallax as an Independent Cue for Depth Perception," *Perception*, vol. 8, no. 2, pp. 125–134, 1979, PMID: 471676.

[17] Mika E Ono, Josée Rivest, and Hiroshi Ono, "Depth Perception as a Function of Motion Parallax and Absolute-Distance Information," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 12, no. 3, pp. 331, 1986.

[18] Keith Stroyan and Mark Nawrot, "Visual Depth from Motion Parallax and Eye Pursuit," *Journal of Mathematical Biology*, vol. 64, no. 7, pp. 1157–1188, Jun 2012.

[19] Mostafa Mansour, Pavel Davidson, Oleg Stepanov, and Robert Piché, "Relative Importance of Binocular Disparity and Motion Parallax for Depth Estimation: A Computer Vision Approach," *Remote Sensing*, vol. 11, no. 17, pp. 1990, 2019.

[20] HyunGoo R Kim, Dora E Angelaki, and Gregory C DeAngelis, "The neural basis of depth perception from motion parallax," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 371, no. 1697, pp. 20150256, 2016.

[21] Miao Zhang, Yu Zhang, Yongri Piao, Jie Liu, Xinxin Ji, and Yukun Zhang, "Parallax based Motion Estimation in Integral Imaging," in *Digital Holography and Three-Dimensional Imaging*. Optical Society of America, 2019, pp. W3A–3.

[22] Oliver W Layton and Brett R Fajen, "Computational Mechanisms for Perceptual Stability using Disparity and Motion Parallax," *Journal of Neuroscience*, vol. 40, no. 5, pp. 996–1014, 2020.

[23] Ana Serrano, Incheol Kim, Zhili Chen, Stephen DiVerdi, Diego Gutierrez, Aaron Hertzmann, and Belen Masia, "Motion Parallax for 360 RGBD Video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 1817–1827, 2019.

[24] Brian Rogers, "Revisiting Motion Parallax as a Source of 3-D Information," *Perception*, vol. 45, no. 11, pp. 1267–1278, 2016, PMID: 27343185.

[25] Andreas Schindler and Andreas Bartels, "Motion Parallax Links Visual Motion Areas and Scene Regions," *NeuroImage*, vol. 125, pp. 803 – 812, 2016.

[26] A. Jones, J. E. Swan, G. Singh, and E. Kolstad, "The Effects of Virtual Reality, Augmented Reality, and Motion Parallax on Egocentric Depth Perception," in *IEEE Virtual Reality Conference*, March 2008, pp. 267–268.

[27] Petr Kellnhofer, Piotr Didyk, Tobias Ritschel, Belen Masia, Karol Myszkowski, and Hans-Peter Seidel, "Motion Parallax in Stereo 3D: Model and Applications," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2016)*, vol. 35, no. 6, 2016.

[28] Alexandru Telea, "An image inpainting technique based on the fast marching method," *J. Graphics, GPU, & Game Tools*, vol. 9, pp. 23–34, 2004.

[29] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Scene Parsing through ADE20K Dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[30] M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3061–3070.