Deep 3D Geometric Saliency Estimation from Light Field Images

Konstantinos Ntogkas, Gerasimos Arvanitis, Konstantinos Moustakas
Department of Electrical and Computer Engineering, University of Patras, Patra, Greece
Email: up1053748@upnet.gr, arvanitis@ece.upatras.gr, moustakas@upatras.gr

***Abstract*—In recent years, Light Field (LF) technology has shown remarkable progress in various computer vision tasks such as depth estimation and salient object extraction. One significant factor contributing to this advancement is the availability of commercial LF cameras that are becoming more affordable and sophisticated. Geometric saliency extraction has also played a crucial role in many computer vision problems by focusing on the most informative aspects of an image or video. In this study, we investigate the feasibility of extracting geometric salient directly from LF images. However, the task is challenging due to the lack of an extensive LF dataset that could provide rich information for image analysis. Therefore, we propose to bridge the gap by creating a synthetic dataset of LF images, which can be used for saliency map estimation. We further train a popular neural network model, called EpiNET, commonly used for depth estimation, to extract salient maps. Experimental results demonstrate that the proposed method effectively extracts salient maps with a 10-20% error on a custom metric. This finding not only confirms the feasibility of the task but also paves the way for further research in this area.**

I. Introduction

One challenging problem in 3D computer vision is to define the visual saliency of an image or an object. Visual saliency is a metric highlighting the significance of an area in a scene based on human visual perception. A salient part of the scene should differ from its surrounding region due to a difference in characteristics. More specifically, the saliency of the 3D object can be calculated by colour changes in its texture [1], geometric differences on its surface [2]–[5], or even by semantic [6]–[8] and behavior supported visual saliency [9]. Although traditional saliency extraction methods are computationally heavy, Convolutional Neural Networks (CNNs) have emerged as a computationally lighter approach for saliency mapping. However, training a CNN for this task is difficult due to the unclear definition of saliency and the lack of saliency map datasets for meshes. Recently, LF image processing has been gaining popularity in the computer vision community [10]–[13]. LF cameras capture the direction of the light and create an array of images with a small baseline difference, making them ideal for depth map [14], [15] and saliency map estimation [16]–[18].

In this work, we tackle the phenomenaly ill-posed problem of estimating the geometric saliency (i.e. 3D shape frequencies) from LF images. While LFs have been used for depth estimation, the result has low resolution in the depth dimension thus being incapable to capture geometric saliency. We argue that light-fields may inherently encode much more than sub-pixel disparities and thus may be capable to capture geometric saliency.

To achieve our aim, since no ground truth datasets exists for geometric saliency of LF images, we propose a methodology for creating a synthetic LF saliency map dataset using 3D meshes. We also demonstrate the feasibility of directly estimating saliency maps using a LF image. Our approach has the potential to provide a lightweight and efficient solution for saliency mapping in computer vision tasks. The key contributions of this work can be summarized as follows:

- A methodology for creating a LF saliency map dataset coupled to a synthetic dataset of LF images.
- A method for saliency map extraction of LFs using convolutional neural networks.
- Initial evidence that geometric saliency can be captured using deep architectures and an end-to-end pipeline for further experimentation.

The rest of this paper is organized as follows: In Section 2, we discuss the related state-of-the-art work in this field. Section 3 presents the necessary preliminaries. In Section 4, we present in detail each step of our proposed method. Section 5 shows the experimental results and in Section 6 we draw the conclusions.

II. Related work

A. LF depth estimation. Due to the very small baseline difference between the LF images, there is a lot of information that could be used for depth estimation, similar to a classic stereo matching scenario. A lot of methods have been suggested for performing the depth estimation task with LF images, both conventional and with the usage of machine learning. Wang et al. [19] propose a method which uses estimates the occlusion on LF images and exploits it in order to get the depth map is a scene. Shin et al. [20] propose an end-to-end CNN which can quickly extract the depth map of a scene from LF images. In order to deal with the lack of datasets, they also suggest a data augmentation technique which can be used for patch wise training on LF images.

B. LF salient object estimation. Salient object estimation is the task of separating an object which is considered important in a scene from the background. Li et al. [21] suggest a methodology by which we can pinpoint the background and foreground of an image using conventional methods for the extraction of the salient object. Wang et al. [22], propose a

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

neural network which is trained on a LF dataset that they captured with LF cameras. Their network uses as input the different focal stacks that we acquire with LF cameras.

C. LF datasets. With the rise of LF image processing, a new need was created for more datasets of LF images. Because of an increase in the use of LF cameras, due to their commercial availability. The LF datasets that have been proposed are separated into two main categories, which are multi-view image datasets and focal stack datasets. The first category takes into account the ability of a LF camera to capture the central view with different focus points. The second category uses the LF cameras micro lens array in order to capture the multiple LF images which have a very small baseline difference between them. Most of the well-known datasets [21]–[25] have as ground truth, the depth of the central view or the salient object of an image which can be separated from the background. As far as we know, there is no dataset which has as ground truth the geometrical salient areas of a mesh.

III. Preliminaries and Notation

In this section, we present the basic definitions and preliminaries which are necessary for the complete understanding of our assumptions and also the ground truth saliency estimation approach, the results of which will be used for the training of our method that will be described in Section IV.

A. Basic Definitions of 3D Meshes

In this work, we focus on triangle meshes $\mathcal{M}$ consisting of n vertices $\mathbf{v}$ and n_f faces f. Each i vertex is represented by Cartesian coordinates, denoted by $\mathbf{v}_i = [x_i,\ y_i,\ z_i]^T,\ \forall\ i = 1, \cdots, n$. Each f_j face constitutes a triangle that can be represented by its centroid $\mathbf{c}_j = (\mathbf{v}_{j1} + \mathbf{v}_{j2} + \mathbf{v}_{j3})/3$ and its outward unit normal $\mathbf{n}_{c_i} = \frac{(\mathbf{v}_{j2}-\mathbf{v}_{j1})\times(\mathbf{v}_{j3}-\mathbf{v}_{i1})}{\|(\mathbf{v}_{j2}-\mathbf{v}_{j1})\times(\mathbf{v}_{j3}-\mathbf{v}_{j1})\|}$, where $\mathbf{v}_{j1}$, $\mathbf{v}_{j2}$ and $\mathbf{v}_{j3}$ are the position of the vertices that define face $f_j = \{\mathbf{v}_{j1},\ \mathbf{v}_{j2},\ \mathbf{v}_{j3}\},\quad \forall\ j = 1, \cdots, n_f$.

B. Spectral-Based Saliency Estimation

Firstly, the whole mesh is separated into n small, overlapped and equal-sized patches (i.e., one patch per vertex), similar to [2]. For each i vertex $\mathbf{v}_i$ of the mesh, a patch of $k+1$ vertices $\mathcal{P}_i = \{\mathbf{v}_i, \mathbf{v}_{i_1}, \mathbf{v}_{i_2},\ \cdots,\ \mathbf{v}_{i_k}\}$ is created consisting of the k geometrical nearest vertices of vertex $\mathbf{v}_i$, estimated by using the k nearest neighbors (k-nn) algorithm (typically, we set $k = 25$). These patches are utilized to create n matrices $\mathbf{N}_i \in \mathbb{R}^{(k+1)\times 3}$, consisting of the $k+1$ corresponding normals [26]:

$$\mathbf{N}_i = [\mathbf{n}_{ci},\ \mathbf{n}_{ci_1},\ \mathbf{n}_{ci_2},\ \cdots,\ \mathbf{n}_{ci_k}]^T\ \forall\ i = 1, \cdots, n \quad (1)$$

Then, the covariance matrix $\mathbf{R}_i = \mathbf{N}_i^T\mathbf{N}_i \in \mathbb{R}^{3\times 3}$ is estimated for each matrix $\mathbf{N}_i$ and it is decomposed: $\mathbf{R}_i = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, to a matrix $\mathbf{U}$ with the eigenvectors and a diagonal matrix $\mathbf{\Lambda} = \text{diag}(\lambda_{i1}, \lambda_{i2}, \lambda_{i3})$ with the corresponding eigenvalues $\lambda_{ij},\ \forall\ j = 1-3$. Finally, the spectral saliency s_{1i} of a vertex $\mathbf{v}_i$ is defined as the value given by the inverse norm 2 of the corresponding eigenvalues:

$$s_{1i} = \frac{1}{\sqrt{\lambda_{i1}^2 + \lambda_{i2}^2 + \lambda_{i3}^2}}\ \forall\ i = 1, \cdots, n \quad (2)$$

We also normalize the values in order to be in the range of [0-1], according to:

$$\bar{s}_{1i} = \frac{s_{1i} - \min(s_{1i})}{\max(s_{1i}) - \min(s_{1i})}\ \forall\ i = 1, \cdots, n \quad (3)$$

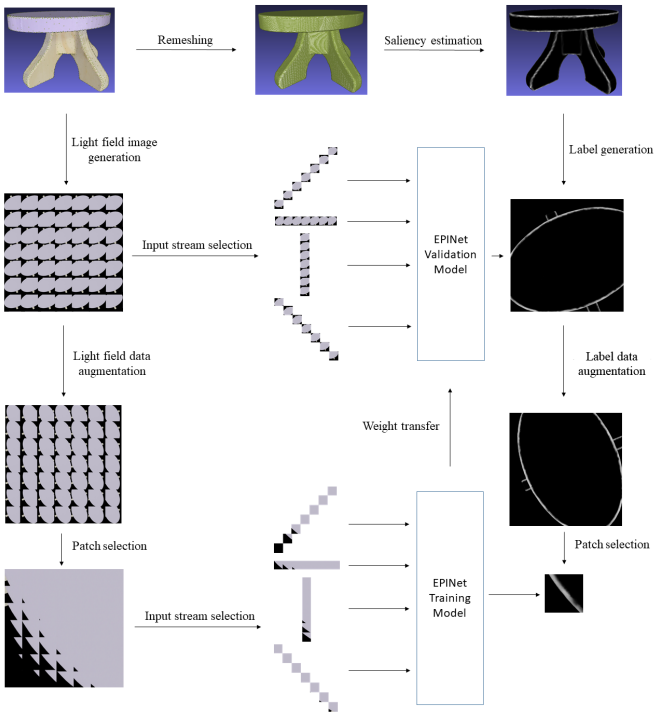

Fig. 1: Framework of the proposed method.

IV. Method

A. Dataset creation

The basic architecture of the proposed method is briefly presented in Fig. 1. More details about each component will be presented in the following sections.

1) Mesh dataset: A lot of 3D objects datasets [27]–[29] were investigated for potential utilization, however, the Google Scanned Objects (GSO) dataset [30] was finally used since a uniform distribution of the vertices is necessary for the geometrical saliency map extraction. The objects of the GSO dataset were scanned using photogrammetry techniques and processed to create high-resolution, textured 3D models. Then, we use the Blenders' Remesh modifier which performs a form of isotropic surface remeshing, to create an optimal mesh with unified vertex distribution. Afterwards, the methodology, presented in Section III, is used to extract the ground truth geometrical salient features of each 3D mesh.

2) Creating the synthetic LF images dataset: To create a dataset of LF images that are generated from 3D virtual objects, we first need to replicate the functionality of the LF camera microlens array. In order to do this, we utilize the 3D computer graphics software tool called Blender which has a Python API scripting tool that allows us to easily create add-ons. Initially, we created a script based on [31], which is

used to transform a typical Blender camera into an LF camera. This script enables us to manage various parameters that can be found in the design specifications of a real LF camera, such as the focal length of the lens, the image resolution of the sensor chip, the f-stop of the cameras, the number of the microlens cameras, the baseline distance between the microlens cameras and the focal distance of the cameras. We then designed a very simple scene in Blender with our LF camera that consistently points towards the objects we load. To avoid any interference with the neural network, we maintain a black background. Afterwards, we sequentially load each object in the GSO dataset, complete with their corresponding textures and ground truth saliency maps. We capture 81 LF images for 9x9 input views which have a size of 512x512 pixels. Finally, we capture the saliency map of the central view of the LF camera. By following this approach, we were able to create a robust and reliable dataset of LF images that are sourced from 3D virtual objects.

B. EPINet Neural Network

The EPINet [20] is a CNN used for a fast and accurate LF depth estimation. Despite other methods, facing the challenge of the created noise by the very small baseline between the microlens camera images, EPINet has four separate and identical processing streams for four angular directions: horizontal, vertical, left, and right. Later, the results from each stream are combined to produce the final output. Before commencing the training process, we selected 990 objects and split them into a train-test set with an 80-20% ratio, respectively. During the training of the model, we used two different NN models. The first model is utilized for patch-wise training while the second model is responsible for validating the full image.

1) Training model: The training model is designed to incorporate the patch-wise training method (Fig. 1). In each epoch, we load 82 out of 792 images, from which we randomly select a batch of 8 images. Based on experimental results, we found that using a 7x7 angular view for our model performs better than other configurations. For the patch-wise training, the first model receives a patch of pixels generated through the generator, with a size of 25x25 pixels. The corresponding label for this input patch is a 3x3 pixel patch, and both of these patches are generated using the training generator. Additionally, the patch undergoes data augmentation processing. To perform stochastic gradient descent, we use a batch size of 8. By doing so, the NN can potentially learn more from each step by paying close attention to each example separately. We begin the training with a learning rate of 1e-5 which is later reduced to 1e-6. We perform 100 steps for each epoch to utilize our training augmentation technique further.

2) Validation model: The validation model is designed to run at the end of every epoch using the weights of the training model (Fig. 1). Our objective is to achieve a low Bad Pixel Metric (BPM) value, so we will use the model that has the lowest BPM value. For the second model, we use full 512x512 LF images with a padded label of 482x482 pixels as input. We use a batch size of 1 and 7x7 angular views. We load all of the test object images used for training into this model. After predicting the output of the model, we calculate the BPM between the result of the neural network and the labels.

V. Experimental Evaluation

A. Experimental Setup and Implementations

The training was carried out using an AMD® Ryzen 5 3600x 6-core processor and NVIDIA GeForce RTX 2080 Ti PC with 24 GB of RAM.

B. Experimental Setup and training loss

As we can see in Fig. 2, the training loss of the model is characterized by oscillations that can be attributed to two possible factors. One could be that the learning rate is too high. However, we can easily dismiss this idea since we do not see any change in the fluctuations after changing the learning rate on epoch 100. The second factor is due to the stochastic gradient descent. It is very possible that the model is getting stuck on local minima in every iteration. We can confirm this because the validation BPM metric does not change significantly after a certain point. The train loss function is fluctuating between the values of 0.04 and 0.06.

Fig. 2: The train loss plot for our model.

C. BPM metric

We use a metric called the BPM to evaluate our results. The BPM measures the average number of pixels for which the absolute difference between the ground truth and predicted pixels exceeds a certain threshold. Our threshold for training is set at 0.07. One of the reasons we use the BPM is to account for small errors between the predicted and actual values that can be safely ignored. Fig. 3 shows that the BPM metric starts at high values, and after a point, it fluctuates until it gets stuck in the 10-20% value range. It can be explained considering that the model takes time to adjust its parameters to ignore the black background. After learning to ignore the background, we dont see any significant improvement in the BPM value up to 500 epochs.

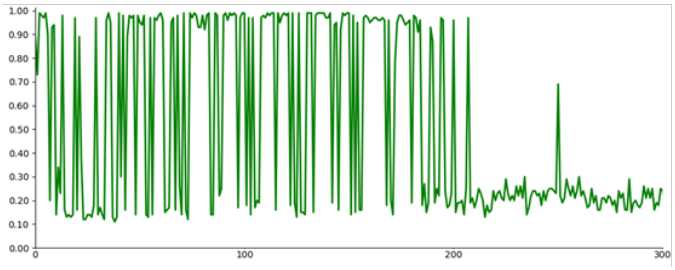

Fig. 3: The 0.07 BPM metric plot for our model.

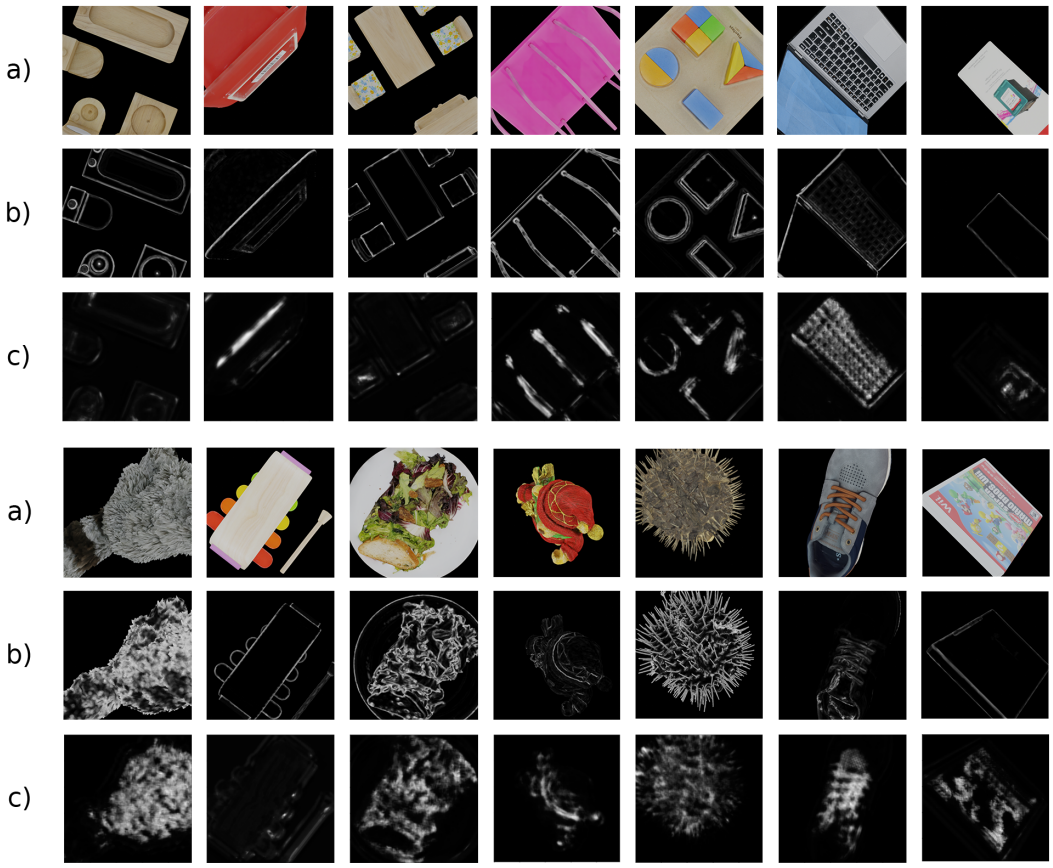

Fig. 4: (a) The center LF image, (b) the ground truth saliency map, and (c) the predictions of our NN, applied on the test dataset and on some random 3D models found on the internet.

D. Results in the test dataset samples

The proposed method was applied both on the test data of the GSO dataset (as described in subsection IV.B) as well as on random 3D models found on the internet. Some indicative results are presented in Fig. 4. The prediction values of our trained neural network are very low, and as a result, some of the details can be missed in the prediction image. In order to deal with this issue, we normalize the values of the prediction between 0 and 1. By performing the normalization process we can see that the most important geometric features, which we have used as labels, have been recognized, even though their values were very low on the prediction outcome. We can also notice that the neural network is able to detect saliency even when multiple objects are in the scene. This can be attributed to the patch wise training that we performed.

VI. Conclusion, Limitations and Future Work

The proposed framework and study concludes that it is indeed possible to estimate an index of geometric saliency directly from light field images, even if this is directly not possible due to the low resolution in the depth dimension.

One of the biggest limitations faced, was finding an optimal way to keep track of the network's progress. The BPM value, which was suggested for the original EpiNET neural network, can give a decent estimation of the performance, but the fluctuations do not allow for straightforward optimizations. Also, while the neural network generalizes impressively in some areas like the keyboard in Fig. 4, it may ignore more fine geometric differences. As far as the dataset is concerned, uniform sample distribution over the surface is something that has to be considered. Moreover, more sophisticated data augmentation, before rendering the LF images in Blender, like texturing, illumination or scaling, could further improve performance.

Future plans involve enhancing the neural network by exploring different options in terms of network structure depth and hyper-parameters.

Acknowledgement

This work has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 101092875 - DIDYMOS-XR: Digital DynaMic and responsible twinS for XR.

The authors would like to thank George Papoulias, a researcher from the University of Patras, for his input in the functionality of EpiNET and for his assistance in reviewing the dataset creation process.

REFERENCES

[1] Y. Nehm, M. Abid, G. Lavou, M. P. D. Silva, and P. L. Callet, "Cmdm-vac: Improving a perceptual quality metric for 3d graphics by integrating a visual attention complexity measure," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3368–3372.

[2] G. Arvanitis, A. S. Lalos, and K. Moustakas, "Robust and fast 3-d saliency mapping for industrial modeling applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1307–1317, 2021.

[3] ——, "Saliency mapping for processing 3d meshes in industrial modeling applications," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 683–686.

[4] K. Lamicchane, P. Mazumdar, and M. Carli, "Geometric feature based approach for 360 image saliency estimation," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019, pp. 228–233.

[5] G. Tinchev, A. Penate-Sanchez, and M. Fallon, "Skd: Keypoint detection for point clouds using saliency estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3785–3792, 2021.

[6] Z.-J. Zha, C. Wang, D. Liu, H. Xie, and Y. Zhang, "Robust deep co-saliency detection with group semantic and pyramid attention," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2398–2408, 2020.

[7] G. Ma, S. Li, C. Chen, A. Hao, and H. Qin, "Rethinking image salient object detection: Object-level semantic saliency reranking first, pixelwise saliency refinement later," *IEEE Transactions on Image Processing*, vol. 30, pp. 4238–4252, 2021.

[8] X. Zhu, J. Wu, and L. Zhu, "Rgb-d saliency detection based on cross-modal and multi-scale feature fusion," in *2022 34th Chinese Control and Decision Conference (CCDC)*, 2022, pp. 6154–6160.

[9] Y. Zhu, G. Zhai, Y. Yang, H. Duan, X. Min, and X. Yang, "Viewing behavior supported visual saliency predictor for 360 degree videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4188–4201, 2022.

[10] Z. Liang, P. Wang, K. Xu, P. Zhang, and R. W. H. Lau, "Weakly-supervised salient object detection on light fields," *IEEE Transactions on Image Processing*, vol. 31, pp. 6295–6305, 2022.

[11] H.-J. Park, J. Shin, H. Kim, and Y. J. Koh, "Light field image super-resolution based on multilevel structures," *IEEE Access*, vol. 10, pp. 59 135–59 144, 2022.

[12] C. Meng, P. An, X. Huang, C. Yang, L. Shen, and B. Wang, "Objective quality assessment of lenslet light field image based on focus stack," *IEEE Transactions on Multimedia*, vol. 24, pp. 3193–3207, 2022.

[13] J. M. Santos, L. A. Thomaz, P. A. A. Assuno, L. A. d. S. Cruz, L. Tvora, and S. M. M. de Faria, "Lossless coding of light fields based on 4d minimum rate predictors," *IEEE Transactions on Image Processing*, vol. 31, pp. 1708–1722, 2022.

[14] L. Han, Z. Shi, S. Zheng, X. Huang, and M. Xu, "Light-field depth estimation using rnn and crf," in *2022 7th International Conference on Image, Vision and Computing (ICIVC)*, 2022, pp. 725–729.

[15] V. Van Duong, T. N. Huu, J. Yim, and B. Jeon, "Lfdenet: Light field depth estimation network based on hybrid data representation," in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2022, pp. 1–4.

[16] P. Prathap and J. J, "Saliency detection from 4d light field images using an optimized cnn-based hybrid model," in *2022 IEEE 19th India Council International Conference (INDICON)*, 2022, pp. 1–5.

[17] M. Zhang, S. Xu, Y. Piao, and H. Lu, "Exploring spatial correlation for light field saliency detection: Expansion from a single view," *IEEE Transactions on Image Processing*, vol. 31, pp. 6152–6163, 2022.

[18] M. Feng, K. Liu, L. Zhang, H. Yu, Y. Wang, and A. Mian, "Learning from pixel-level noisy label : A new perspective for light field saliency detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1746–1756.

[19] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[20] C. Shin, H.-G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, "Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images," 2018. [Online]. Available: https://arxiv.org/abs/1804.02379

[21] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[22] T. Wang, Y. Piao, H. Lu, X. Li, and L. Zhang, "Deep learning for light field saliency detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8837–8847.

[23] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," 03 2017, pp. 19–34.

[24] J. Zhang, Y. Liu, P. R. Zhang, Shengping, and M. Wang, "Light field saliency detection with deep convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 4421–4434.

[25] D. G. Dansereau, B. Girod, and G. Wetzstein, "LiFF: Light field features in scale and depth," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019. [Online]. Available: http://dgd.vision/Papers/dansereau2019liff.pdf

[26] S. Nousias, G. Arvanitis, A. S. Lalos, and K. Moustakas, "Mesh saliency detection using convolutional neural networks," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[27] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015. [Online]. Available: http://arxiv.org/abs/1512.03012

[28] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Scenenet: Understanding real world indoor scenes with synthetic data," *CoRR*, vol. abs/1511.07041, 2015. [Online]. Available: http://arxiv.org/abs/1511.07041

[29] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, "3d-front: 3d furnished rooms with layouts and semantics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 933–10 942.

[30] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," 2022. [Online]. Available: https://arxiv.org/abs/2204.11918

[31] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Asian Conference on Computer Vision*. Springer, 2016.