A Novel View Direct and Indirect Image Synthesis Method

Xiaonong Sun, and Anagh Malik

Abstract—We propose a method that achieves high resolution separation of direct and indirect images of a scene with minimum frames, which can allow us to recreate the scene and the surrounding landscape in 3D. We simulate the 200 transient images of a scene captured at different camera angles to approximate the images typically captured by emergent image sensors like the Single-Photon Avalanche Diode (SPAD) with the renderer software Mitsuba. From this, we introduce a novel heuristic to separate out the indirect and direct lighting from a transient image. We then synthesize a Neural Radiance Field with the 100 indirect and direct images. Our method showcases, that despite the complicated image formation model of our captured images, NeRF is able to reconstruct a 3D scene using them to perform free-viewoint rendering. Our algorithm and flow shows we can generate direct and indirect images This work can potentially allow new and novel applications in the field of surveillance, environment monitoring and defence equipment.

Index Terms—Computational Photography, Transient Imaging, Direct and Indirect Imaging

1 Introduction

When we take a conventional image of a scene, we are summing up all the photons that hit the sensor over the exposure time. However, this process discards the arrival trajectory and the paths taken by each individual photon, which means useful information is lost. Recently, emerging sensors utilizing new hardware processes like the Single-Photon Avalanche Diode (SPAD) cameras allow the viewing of a trillion frames per second and have allowed new imaging techniques like transient imaging. Transient imaging is a set of techniques that captures the propagation of photons through 3D space and creates a response map that contains information in 3 dimensions (x, y and time). The accuracy of this method can even facilitate viewing single photon arrivals. In the active imaging regime, transient imaging could potentially allow the disambiguation between the direct light (which contains photons which have bounced off at most 1 surface) and indirect light (which has bounced off at least 2 surfaces). By differentiating between these photons, we can separate out images of direct and indirect light. Separating out direct and indirect light could open up potential applications like higher resolution time-of-flight imaging, recreating the 3D environment outside the camera's field of view, and seeing behind walls and objects.

However, capturing single-photon image data is computationally and memory expensive, which makes this unfeasible for many mass market memory-limited applications like consumer cameras and surveillance. Recent developments in 3D Computer Vision have however led to algorithms for Novel View Synthesis, which would allow us to recover indirect/direct light images from just a few multi-view observations that would significantly cut down on the acquisition time required. For this project, we propose to extract a direct light and indirect light neural radiance field of a scene from only 100 transient images taken from different camera angles using NeRF, and then through integrating the neural radiance field get the direct and indirect image of the scene. This paper seeks to demonstrate the feasibility of extracting high-resolution direct and indirect images from only a small number of transient images. Section 2 gives a brief overview of the work currently done in this area, section 3 goes into more detail about the proposed method to extract direct and indirect light, and section 4 talks about the experimental results.

- *Xiaonong Sun is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. E-mail: frankxns.sun@mail.utoronto.ca*
- *Anagh Malik is with the Department of Computer Science, University of Toronto, Toronto, ON, Canada.*

2 Related Work

Recent work has been done on 3D Scene Reconstruction as well as direct and indirect light separation. However, no method has been forumalted to do Novel View Synthesis in the direct or indirect image domain separately.

2.1 Direct and Indirect Light Separation

One of the papers that sparked the interest in direct and indirect separation was published in 2006 [1]. The papers proposes to use high frequency light sources and multiview images to get light separation. Since then there has been a lot of follow up work [2], [3], however none have attempted to reconstruct novel indirect/direct light views.

2.2 Scene Reconstruction

There exists a wide range of methods for RGB and RGB-D based 3D reconstruction. Most of RGB-D reliant methods are based on [4], where multiple depth measurements are fused using a signed distance function (SDF) which is stored in a uniform 3D grid. An example of such work is KinectFusion [5] combines such representation with real-time tracking to reconstruct objects and small scenes in real-time. An

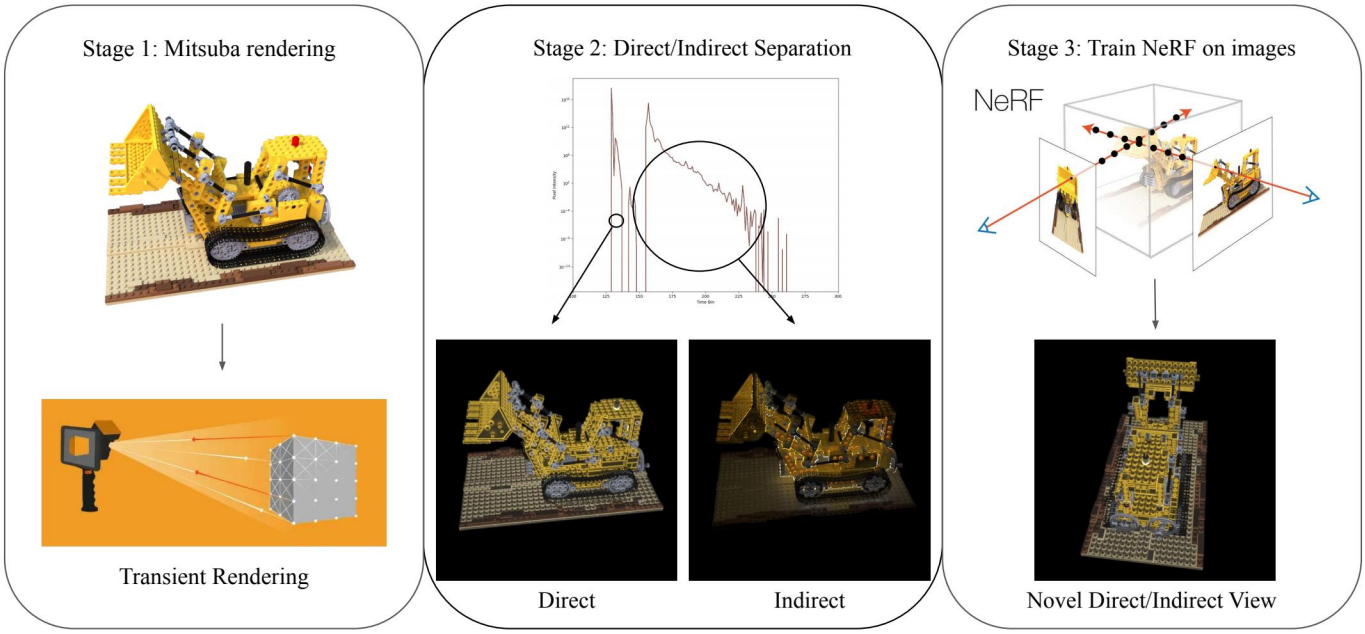

Fig. 1. The process of our direct and indirect image free viewpoint rendering. We first generate transient images from mitsuba. In the second step we separate the direct and indirect components using a simple heuristic. Finally we train NeRF to reconstruct the scene from just these two components separately.

example of a method reliant on just RGB images is Single View MPI [6], which learns to generate multiplane images given one or more images with known viewpoints.

Most recently coordinate-based multi layer perceptrons (MLP) have become a popular representation of the 3D scene [7]. As input the MLP takes a 3D location in the model space and outputs for example, occupancy, density or colour. There has been a lot of work using this simple idea in multiple applications like SLAM [8], [9], for novel-view synthesis [7], [10].

The ubiquitously used, MLP based method called NeRF [7] uses a 5D neural radiance field to represent the scene i.e. the scene is represented with its volume density and directional emitted radiance at any point in space. To render the color of any ray passing through the scene, they propose to parametrize an MLP with a scene coordinate and viewing direction **d**, to output colour **c**(x) and volume density σ(**x, d**).

The expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f is then calcualted using quadrature as:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) \mathbf{c}_i$$

$$\text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$$

The original NeRF implementation was however very slow and there have been many methods since then to speed it up [11], [12]. We will however use an implementation of Instant-NGP [13], which uses multi-resolution hashed feature grids to store representation of points which are then then decoded into the same network outputs as for NeRF. The implementation we will be using is from the popular library NeRFAcc [14].

3 Proposed Method

For the project we will be working in simulation, specifically, we use the Mitsuba renderer [15] to render transient images of the scene. We then use a heuristic to separate direct/indirect photons, which can then be added up to give the indirect and direct component images. We then use Nerf [7] to be able to render novel indirect and direct images. To demonstrate this process, two images were processed using our flow (car and lego). The flow is visualized in Figure 1.

3.1 Capturing Transient Images

To generate the 200 transient images, we take the Mitsuba file of the target scene and then generate 200 new camera locations and angles, in the process generating 200 Mitsuba files each with its unique camera location. To expedite the process of generating the 200 camera angles, the Mitsuba file is first exported to Blender to determine the x,y and z location of the target object in the scene, as well as its size. Then a script was written based on NeRF's synthetic

dataset to generate 200 camera locations that sweep around an object in 360 degrees. The angle could be changed to change the sweep range (0 to 360 degrees), as well as the camera pivot origin, camera x, y and z distance to the object, and camera start and end orientation. The values are then manually inputted based on the scene shown in Blender. This method allows us to quickly generate all the camera locations while providing greater versatility when it comes to the camera movement through the scene, which would make this method feasible for different varieties of scenes. The camera location in Mitsuba files is represented as a 15-element vector that contains information on camera location, orientation and angle.

This is different from the output of the Blender code, to use the generated camera locations in Mitsuba, we have written code that performs the transformation between the spatial representations. After the Mitsuba file is generated from the original image, Mitsuba was run on each of the 200 xml files to generate 200 transient images.

We use a derivative of Mitsuba, called Non-line-of-sight Mitsuba [16], since Mitsuba does not directly support transient rendering. The code was specifically adapted for transient rendering, however we use a slight reimplmentation of it to support also confocal transient imaging.

The two scenes we generated the dataset from can be found in [17]. To be exact we used a dataset of a pontiac car and a legoloader.

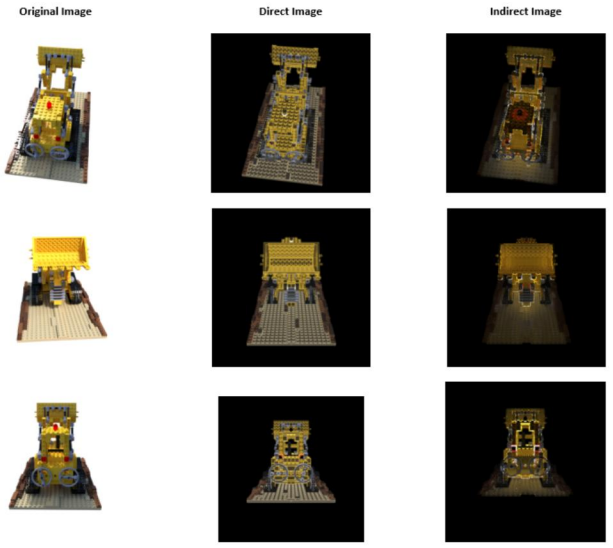

Fig. 2. The direct and indirect separated images from our method, put together with the original images of the scene.

3.2 Separation of direct and indirect images

From the transient images, the time-intensity graph was extracted, this is shown by the graph in Stage 2. of Figure 1. From this, the two predominant peaks were identified and the values within the two buckets around the peaks were integrated over to obtain the direct and indirect image.

More specifically we use the square-fall off property of light to conclude that the direct peak of the light will be dominant in the transient. From this we can conclude that for a given pixel the highest peak in the time dimension will be representing the direct component of light (since it has the shortest distance). We can then by a simple heuristic take 5 timesteps after this peak to represent the direct components of light. We assume that the remaining components of the transient compose the indirect reflections. Summing over these two components in the time dimension gives us the two desired images.

The results can be seen in Figure 3.1. This is not the best way to separate the indirect and direct components of light. We can see that the direct component looks a bit darker in some spots, this is likely due to some ""leakage" of the direct into the indirect images - since the choice of 5 additional timesteps is a bit arbitrary. A more principled way given the renderer would be to explicitly render only one bounce and then more bounces, to subtract the two images.

However we wanted to simulate a SPAD camera and see how this separation could be done with data obtained using a confocal SPAD measurement.

3.3 NeRF

For free viewpoint rendering of the indirect and direct lighting representations of the scene we train the NeRF as described previously, using the NeRFAcc implementation of Instant-NGP. Separately we train the network to render the colour of indirect/direct light components, using only the direct path integration of NeRF along rays. To train we use the groundtruth poses we get from Blender.

For the training setup, we separated the dataset of 200 images into two equal sized training and test sets. We used every second image as a training set. This means that we also had quite a dense representation of the scene beforehand.

4 Experimental Results

We obtained very good results for both the lego and car scenes. This can be better seen in 4, where we show the PSNR are SSIM metrics for both the scenes.

Firstly we can notice that the car scene has a much better PSNR than the lego images, this is probably since in the car scene the object is much smaller than in the lego scenes and the images are mostly represented by the background, hence the pixel based metrics for example squared error are quite - the network does not have to learn much to represent the image.

However notice that the car image, more specifically the image of the indirect lighting components 4 are not very well represented and the image contains some specularities.

What is more interesting however is to notice the difference between the PSNRs for the direct and indirect images. The PSNRs for the direct images are consistently higher as should be expected. This is since even though NeRF has a view direction component to it, it does poorly with data with changing colours from one image to another one, which happens more so in indirect images, since th light source can be thought of as being different from image to image. This is somewhat the case with the direct component as well (lighting is changing as we have a confocal

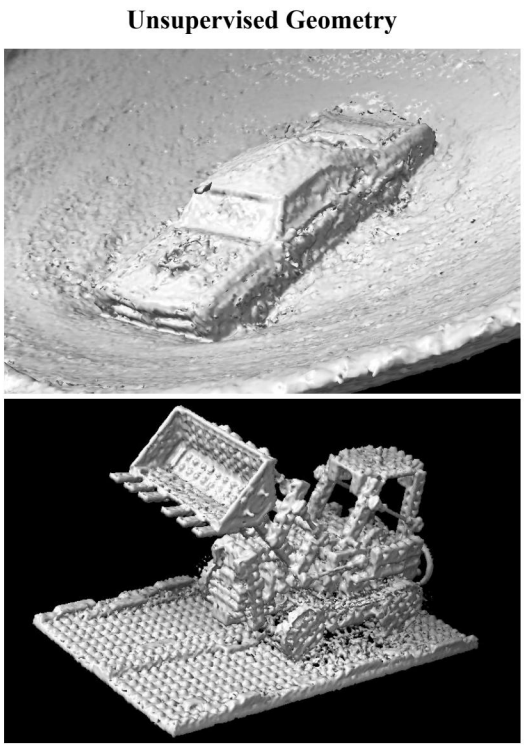

Fig. 3. The geometry extracted from our direct NeRFs. As we can see the network is able to capture quite a lot of details in the geometry, for example small ridges on the lego ground or the wheels of the car. We call the geometry unsupervised, since apart from camera positions there is no regularization on it, and it emerges from the network.

setup), however the direct images does not take these more complicated light paths into equation.

Nonetheless it is very surprising to see that NeRF can represent these indirect images as all, it is not at all obvious. As mentioned previously we could think of these indirect images as if the light is coming from the second bounce position, this means that the exact lighting conditions for ex very pixel are extremely complicated. NeRF however is able to deal with this by just having a viewing direction conditioned colour. This might also say something about indirect reflections itself, we can deduce that indirect reflections are principled enough for them to be representable.

Finally we also visualize the geometry produced by our images in Figure 4. The geometry is very much comparable to the results that can be obtained from normal NeRF, in fact after some optimization of parameters we should expect the direct components geometry to be better than NerF, since the network no longer has to use the geometry to explain away more complicated lighting effects.

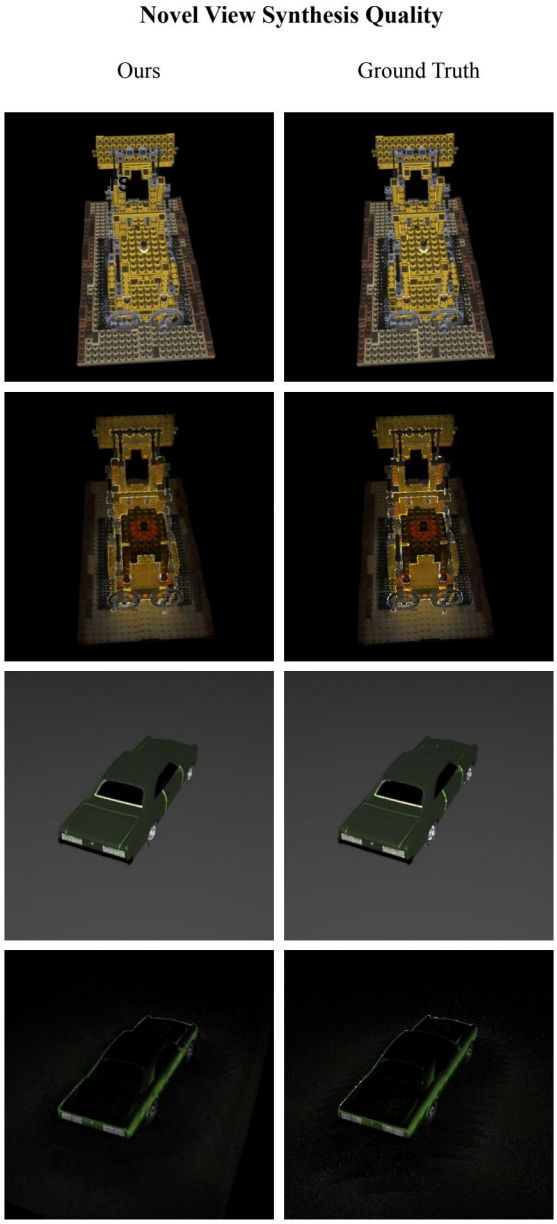

Fig. 4. Test viewpoint renders from out direct and indirect NeRFs. As we can see despite a few inconsistencies on the indirect images of the car, the network is able to reconstruct the images quite well.

5 Discussion

From the experimental data, NeRF's simple rendering equation is able to represent complex lighting effects, such as indirect reflections. Nonetheless, results are better for the reconstruction of the scene using the direct component of light. This is probably due to the image formation process being closer to the rendering equation. We are able to recover geometry from the images. For future work, it would be interesting to see how the reconstructed geometry compares to the geometry reconstructed with normal images. It would also be interesting to see these reconstructions for more complicated surfaces, for example with sub-surface scattering.

Another possibly interesting avenue for exploration would be to use a volumetric framework to perform more sophisticated rendering. We could use the direct and indirect images we get to supervise a model whose rendering equations contain indirect light paths. More specifically by rendering a say 2 or 3 light bounce model, we could generate estimated direct and indirect images from the network,

Scene	Component	PSNR	SSIM
Lego	Direct	33.64	0.98
Lego	Indirect	28.50	0.95
Car	Direct	38.54	0.98
Car	Indirect	33.01	0.90

Fig. 5. PSNR are SSIM results of our reconstructed direct and indirect images on the car and lego scene. The average results are reported across 100 test views of the scene.

which could then be compared with groundtruth data to train a volumetric representation of the scene.

This might unlock a plethora of applications, which have already been showcased with transient data, for example non-line of sight imaging or scene relighting.

6 Conclusion

Our work introduces an alternative to traditional reconstruction techniques generated from direct and indirect lights. This provides new opportunities We demonstrated the effectiveness of utilizing Neural Radiance fields generated from the direct and indirect images of only a few camera positions to construct a scene with minimum space requirements.

Despite our best efforts, we have not been able to show an exponential result or reason to separate indirect and direct components of light. Despite the applications of transient imaging being plentiful, the use of these intermediate representations (between normal and transient images) i.e. direct and indirect images, is still not fully clear. We expect more work and results in the future to showcase the usefulness of this intermediate representation.

We expect interest to grow as more camera sensors are developed that include direct and indirect image sensing, for example, the T6 camera [18].

Acknowledgments

The authors would like to thank Professor David Lindell for his guidance and support in the conception, implementation and refinement of this paper. We would also like to express our gratitude to the CSC2529 teaching assistant team for their help with the logistics and technical support.

References

[1] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, "Fast separation of direct and global components of a scene using high frequency illumination," *ACM Trans. Graph.*, vol. 25, no. 3, p. 935–944, jul 2006. [Online]. Available: https://doi.org/10.1145/1141911.1141977

[2] O. Nasu, S. Hiura, and K. Sato, "Analysis of light transport based on the separation of direct and indirect components," 06 2007.

[3] M. O'Toole, F. Heide, L. Xiao, M. B. Hullin, W. Heidrich, and K. N. Kutulakos, "Temporal frequency probing for 5d transient analysis of global light transport," *ACM Trans. Graph.*, vol. 33, no. 4, jul 2014. [Online]. Available: https://doi.org/10.1145/2601097.2601103

[4] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 303–312. [Online]. Available: https://doi.org/10.1145/237170.237269

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[6] R. Tucker and N. Snavely, "Single-view view synthesis with multiplane images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[8] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," *CoRR*, vol. abs/2103.12352, 2021. [Online]. Available: https://arxiv.org/abs/2103.12352

[9] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: neural implicit scalable encoding for SLAM," *CoRR*, vol. abs/2112.12130, 2021. [Online]. Available: https://arxiv.org/abs/2112.12130

[10] D. Verbin, P. Hedman, B. Mildenhall, T. E. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," *CoRR*, vol. abs/2112.03907, 2021. [Online]. Available: https://arxiv.org/abs/2112.03907

[11] D. B. Lindell, J. N. P. Martel, and G. Wetzstein, "Autoint: Automatic integration for fast neural volume rendering," in *Proc. CVPR*, 2021.

[12] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. P. C. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," *CoRR*, vol. abs/2103.10380, 2021. [Online]. Available: https://arxiv.org/abs/2103.10380

[13] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[14] R. Li, M. Tancik, and A. Kanazawa, "Nerfacc: A general nerf accleration toolbox." *arXiv preprint arXiv:2210.04847*, 2022.

[15] W. Jakob, S. Speierer, N. Roussel, and D. Vicini, "Dr.jit: A just-in-time compiler for differentiable rendering," *Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 41, no. 4, Jul. 2022.

[16] D. Royo, J. García, A. Muñoz, and A. Jarabo, "Non-line-of-sight transient rendering," *Computers Graphics*, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849322001200

[17] B. Bitterli, "Rendering resources," 2016, https://benedikt-bitterli.me/resources/.

[18] R. Gulve, N. Sarhangnejad, G. Dutta, M. Sakr, D. Nguyen, R. Rangel, W. Chen, Z. Xia, M. Wei, N. Gusev, E. Y. H. Lin, X. Sun, L. Hanxu, N. Katic, A. Abdelhadi, A. Moshovos, K. N. Kutulakos, and R. Genov, "A 39,000 subexposures/s cmos image sensor with dual-tap coded-exposure data-memory pixel for adaptive single-shot computational imaging," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, 2022, pp. 78–79.