https://doi.org/10.2352/EI.2022.34.10.IPAS-356

Volumetric Segmentation for Integral Microscopy with Fourier Plane Recording

Sergio Moreschini, Robert Bregovic, Atanas Gotchev; Tampere University; Tampere, Finland

Abstract

Light Field (LF) microscopy has emerged as a fast growing field of interest in the last two decades for its undoubted capacity of capturing in-vivo samples from multiple perspectives. In this work we present a framework for Volumetric Segmentation of LF images created following the setup of a Fourier Integral Microscope (FIMic). In the proposed framework, we convert the FIMic-captured LF into a three-dimensional Focal Stack (FS) to be used as an input to machine learning models with the aim to get the 3D locations of the specimen of interest. Using a synthetic dataset generated in Blender, we train three neural networks based on the U-Net architecture and merge their outputs to achieve the desired volumetric segmentation. In our main test results we achieve a precision of more than 95%, while in the related tests we still achieve a value higher than 80%.

Introduction

LF is a formal representation of the bunch of directional rays in space, parameterized most often by the ray coordinates on two parallel planes [1]. LF capture principles can be traced back to the original work on integral cameras [2], aimed at multiplexing angular and spatial ray information on a 2D sensor. Such LF images are instrumental for computational post-processing, such as refocusing and depth estimation. Therefore, LF has been considered an attractive 3D microscopy alternative for imaging in-vivo samples in reduced time and lower phototoxicity. The first LF microscopy setup proposed in [3] comprises a microlens array located at the intermediate image plane of the objective followed by a sensor at the rear focal plane of the microlenses [3]. The part of the sensor image behind a microlens, referred to as an Elemental Image (EI), captures a different perspective of the specimen of interest. The compound image of all EIs multiplexes angular and spatial information in a two-plane parameterized LF [1].

An alternative of this setup is the Fourier Integral Microscope (FIMic) [4], where the microlens array is placed at the aperture stop of the microscope objective. This results in EIs with extended depth of field and enhanced lateral resolution for the price of reduced angular resolution. While high lateral resolution is required for observing small object details, a high angular resolution is desired for resolving dense depth planes trough computational refocusing [5]. The goal of refocusing in LF microscopy is to compute the image that would be visible if a microscope would be focused at a specific depth. Such an image shows both the sharp objects in focus (i.e. at the targeted depth) and the blurred objects out of focus (being at different depths). However, there might be applications demanding segmentation of only objects at a particular depth and excluding the ones at other depths.

In classical (confocal) microscopy [6], [7], images are captured by selectively exciting a 3D sample located at the specific depth plane by fluorescence, and changing the focal lengths accordingly. When stacked together, the images form the so-called ZStack (or Z-Scan) [8] [9]. While instrumental for excluding the undesired blurred image regions, the process is slow and phototoxic for in-vivo samples.

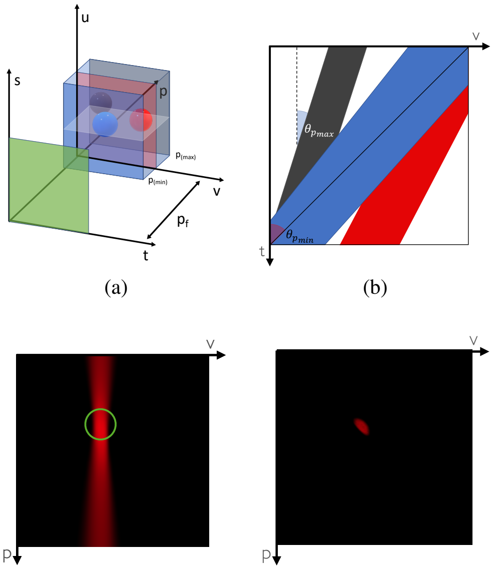
In this work we aim at describing a specimen volume through computationally segmenting the ZStack into object pixels of interest, making use of our previous work on refocusing [5]. The possibility of fully reconstructing a volume from an LF microscope image has been investigated in [10] and [11]. Both works utilize single-shot LF microscope images with high angular resolution. On the contrary, in our work, we aim at exploiting the computationally reconstructed densely-sampled LF as an input to a volumetric segmentation machine learning model.

Background

Consider a semitransparent specimen within a volume $V(u,v,p)$, where p is the depth axis. The ensemble of 2D sections $V(u,v,p_k)$ at different depths p_k within a range $[p_{min}, p_{max}]$ is referred to as *ZStack* $Z(u,v,p_k)$ as shown in Figure 1 (a)).

The LF formalism is applicable for sensing the specimen within the volume $V(u,v,p)$. Consider an LF $L(u,v,s,t)$ parameterized by two parallel planes (s,t) and (u,v) at a focal distance p_f, as illustrated in Figure 1 (a) [1]. By locating a matrix of pinhole cameras on the *camera plane* (s,t), one gets EIs from different perspectives on the *image plane* (u,v). The 4D LF representation $L(u,v,s,t)$ can be sliced by fixing e.g. $u = u_0$ and $s = s_0$. The new representation $E(v,t) = L(u_0,v,s_0,t)$, referred to as Epipolar Plane Image (EPI) [12], is characterized by stripes picturing the evolution of objects along camera motion (perspective) with slopes θ corresponding to the objects' depths within the range $[\theta_{p_{min}}, \theta_{p_{max}}]$ (Figure 1 (b)). Elements in focus appear with vertical slopes. Through shearing with particular θ_{p_k}, one can get different planes p_k in focus and stack them to get the so-called *Focal Stack* (FS) $FS(u,v,p_k)$ [13]. The resharing process is problematic when the EIs are taken at sparse discrete perspectives that is equivalent to having broken epipolar lines in the EPI representation. To tackle this issue, in our previous work, we have reconstructied the FS from a representation referred to as densely-sampled LF (DSLF), comprising EIs with less than 1 pixel maximum disparity between adjacent images [5].

While the ZStack includes only sections of objects at particular depths, the FS includes images of all objects, both those in-focus, which appear sharp, and out-of-focus, which appear blurred. In this work, our goal is to use a reconstructed FS, in order to find the objects of interest within the ZStack.



(a) (b)

(c) (d)

Figure 1: (a) Two-plane LF parameterization. (b) Epipolar Plane Image. (c) 2D slice of an FS. (d) 2D slice of a ZStack.

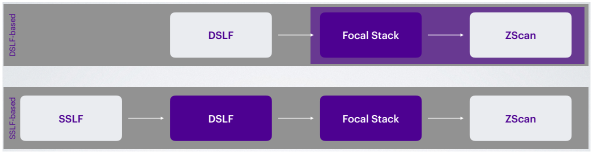


Figure 2: Pipeline of the approach.

Methodology

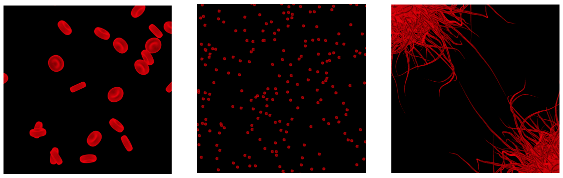
We aim at segmenting a specimen within a volume from an LF captured by a FIMic. In our approach, we reconstruct a discrete FS first to make use of the in-focus sections there for segmentng the corresponding ZStack.

Consider a 2D slice of the FS by fixing $u = u_0$, $EFS(v, p) = FS(u_0, v, p)$. The object's evolution along p is pictured as *hourglass-shaped* changing-intensity stripe, with an intensity peak at the depth plane where the object is in focus, which is also the tiniest stripe area, since the energy is the same for all ps (c.f. the green circle in Figure 1 (c)). The effect is caused by the way the FS is obtained, generally by averaging.

The counterpart ZStack slice looks both similar and different. One can find a match between the slices only where a featured object does exist in the ZStack and thus appears in focus in the FS (c.f. Figure 1 (d)). We therefore need a tool to analyze the FS as an input and segment the areas where the intensity is confined and maximized.

We make use of a deep learning model, based on the U-Net architecture [14]. Such network architecture has been widely used for image segmentation tasks especially in the field of biomedical images. Some of its main advantages are the minimal amount of data required for training and the high segmentation precision.

Our ultimate goal is to be able to segment the volume in the form of ZStack from captured EIs, the so-called sparsely-sampled LF (SSLF), as illustrated in the bottom half of Figure 2. Our hypothesis is that the DSLF is the instrumental intermediary for generating FS as the machine learning model input and has to be reconstructed from SSLF first. To test this hypothesis, in this work we utilize synthetic imagery. Specifically, we use Blender, to render both the Ground Truth (GT) ZScans of specimens and the corresponding DSLFs (top half of Figure 2).



(a) (b) (c)

Figure 3: Preview of the scenes used for test: (a) Cells. (b) Korona Particles. (c) Pili.

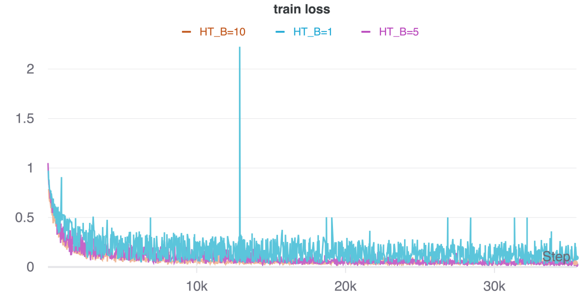


Figure 4: Train Loss for H Training based on different Batch size.

An FS, reconstructed from the corresponding DSLF, is used as input for three ML models based on the U-Net architecture. The Frontal model uses the FS slices (u, v) along p axis, while the Horizontal and Vertical models use (v, p) slices along u, and (u, p) slices along v correspondingly. The three models give rise to three ZStacks, which are then merged into a single segmented volume.

While our main goal in this work is to test the DSLF potential for reconstructing ZStack from computed FS, in practice one would get an SSLF in the form of a limited number of EIs. In our simulations, SSLFs are obtained by decimating the DSLFs generated in Blender. The desired DSLFs are then reconstructed using the tool introduced in [15].

Experiments

The network has been trained using a synthetic dataset created in Blender that mimics the FIMic capture process. The dataset includes 8 scenes as described in [16]. For each scene, the GT ZStack is composed of 201 images and the DSLF, to be used for reconstructing the FS, is composed of 51x51 images. Three of the eight scenes have been used for testing and are presented in Figure 3. The *Cells* scene comprises multiple objects of medium thickness; the *Korona Particles* scene contains a high amount of small particles characteristic for coronaviruses; and the *Pili* scene has *Bacteria* objects with multiple tentacles.

The U-Net architecture follows the one in [17]. It is implemented in Pytorch and makes use of the Weight and Biases developer tool for ML monitoring [18].

The experiments have been set as follows:

- **Optimizer:** ADAMAX
- **Scheduler:** StepLR (step_size = 20, gamma = 0.1)
- **Loss:** CrossEntropy + Dice Coefficient

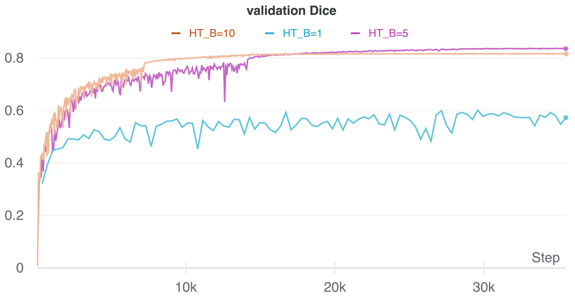


Figure 5: Validation Dice result for H training (first 35500 steps).

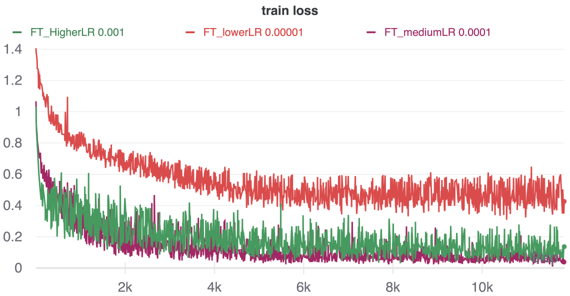


Figure 6: Train Loss for F Training.

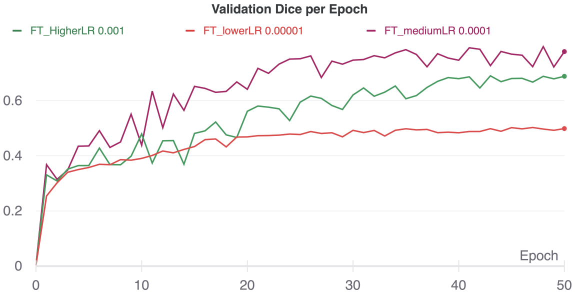


Figure 7: Validation dice per epoch for F Training.

Our training has been coping with the limited amount of data we have. Following approaches originally used for time series analysis, we leave as less data as possible for testing, and use the rest of data for training and validation [19]. More specifically, each time we leave one scene for testing and use the data of the remaining seven scenes for training and validation in a corresponding proportion 80% to 20%. We train three U-NET based machine learning models, for three data arrangements denoted as Horizontal (H), Vertical (V) and Frontal (F) training. We repeat the training for leaving different test scenes in order to test the consistency and stability of the trained models.

Most U-Net based models have been trained with a batch size equal to one. We found that this selection is suboptimal for our 3D-structure case and performed some tests to find out the best batch size. We illustrate our findings with the H training and the *Cells* dataset for testing. Grouping samples and increasing the batch size leads to less training steps. We compare both the training loss and the validation loss based on a maximum value of 35500 steps. As seen in Figure 4, a batch size of one yields an unstable training loss, while increasing the batch size reduces the fluctuations. The curve of the validation process shows that a training performed with a batch size of 1 stays below the score of 0.6 for the Validation Dice (Figure 5) while a higher batch size brings a score higher that 0.8 and with much lower fluctuations.

To optimize the learning rate (LR), a stepLR optimizer set to 20 has been used. We illustrate the empirical observations for the case of F training and *Cells* test dataset for the first 50 epochs. The 3 starting values adopted for the different LR are: 0.001, 0.0001, and 0.00001. A low starting LR produces a train loss which remains above 0.4 score (Figure 6). Using a high starting LR produces too many fluctuations and therefore does not stabilize the training. The results are reflected also in the validation curve (Figure 7) where the lower LR quickly stabilizes to a very low validation dice score, while the medium and higher value are still more unstable in the first 50 values but reach a higher result. An initial, rather moderate, LR of 0.0001 should be favored.

Results

We present results in terms of several metrics aimed at quantifying the segmentation quality. The True Positive *(TP)* metric represents the ratio between the number of correctly marked pixels (marked by 1) and the total number of pixels. Its counterpart is the True Negative *(TN)*, which represents the ratio between the correct no-object pixels (marked by 0) and the total number of pixels. The sum of *TP* and *TN*, denoted by *TruePixels*, is the total number of all correctly marked pixels (both object and no object). The False Positive *(FP)* metric represents the ratio between the total number of predicted 1, when the actual value should be 0, and the total number of pixels. Likewise, wrongly predicted 0's versus all pixels is denoted by False Negative *(FN)*. The sum of *FP* and *FN*, denoted by *FalsePixels*, gives the total of all incorrectly predicted pixels versus all pixels: $FalsePixels = 1 - TruePixels$. Precision and Recall (True Positive Rate (TPR)) metrics are defined as follows: $Precision = \frac{TP}{TP+FP}$; $Recall = TruePositiveRate(TPR) = \frac{TP}{P}$. Subsequently, the True Negative Rate *(TNR)* is $\frac{TN}{N}$, the False Positive Rate *(FPR)* is $\frac{FP}{N}$, and the False Negative Rate *(FNR)* is $\frac{FN}{P}$.

The results for the test scene *Cells* are summarized in Table 1. Each row lists results per neural network and their combination. One can notice that the H (or V) networks perform better in terms of detecting 'positive' pixels. However, relying on either H or V is very much scene specific, as it depends on the scene composition. This justifies the use of both networks. On the other hand, the F network is more capable of detecting 'negative' pixels and reaches the best values for the TNR (and by consequence the lowest for FPR). This is especially important for removing false positives detected by the other networks. We illustrate this qualitatively in Figure 9, which shows a fragment of the segmentation result for the scene *Cells*. While the H network detects correctly the bulk of 'positive' pixels, it also falsely marks pixels of no objects as object pixels. The F network fails to detect all object pixels however there are no false positives. The merged result from the three networks is a compromise with effectively removed outliers and correctly detected object pixels.

Table 2 summarizes the results for two other test scenes, namely *Korona Particles* and *Pili*. The results for the former are similar to these for *Cells*. However, the results for the latter are considerably worse. The scene contains fairly wide areas with connected object pixels, which is in contrast with the other test scenes characterized by small objects at different depths. Such wide areas do not manifest as bright (in-focus) spots in the focal stack slices and correspondignly are not detected accurately. The

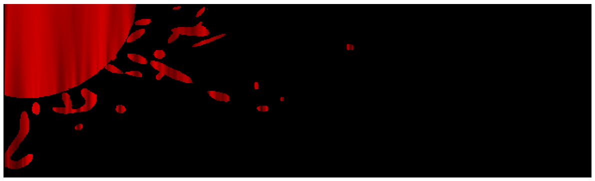


Figure 8: Overlap of GT area over training image.

effect is illustrated in Figure 8. It shows the GT slice overlapped by the slice to be segmented. The latter contains 'shadowed' areas (c.f. the upper left corner of the figure), which confuse the ML model. The results is also visible in Figure 10 (f) and (g).

The so-far presented results are for DSLFs directly rendered from the synthetic scenes, i.e. 'ideal' DSLFs. In practice, the DSLF cannot be sensed due to the requirement for less than 1 pixel disparity between neighbouring views, which in turn requires very high angular sampling. The DSLF has to be reconstructed from sparsely sampled angular views instead. Consider the example of the *Cells* scene. Its DSLF is represented by 51x51 views, while an FIMic would be set by much lower number of angular views. To quantify the performance of our segmentation approach for the real case, we downsampled the true DSLF to 5x5 views and subsequently reconstructed the DSLF and got the FS out of it [5]. The reconstructed FS was then used as input to the three NNs. The results are given in the last row of Table 2 and in Figure 10 (b).

Conclusion

In this paper, we presented an approach for volumetric segmentation of 3D samples captured by an LF microscope with the FIMic optical setting. By means of simulations with synthetic scenes, we investigated the possibility to segment the ZStack given the FS as reconstructed from the DSLF of the sensed scene. We made use of three difference 2D slice directions of the FS and trained the corresponding ML models. We further tested the proposed framework on an SSLF, assuming this would be the real case of FIMic-based LF sensing. Our results are quite encouraging especially in terms of Precision. While the Recall values are not equally high, we believe this can be improved by extending the training dataset.

References

[1] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.

[2] G. Lippmann. Epreuves reversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1):821–825, 1908.

[3] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. In *ACM SIGGRAPH 2006 Papers*, pages 924–934. 2006.

[4] G. Scrofani, J. Sola-Pikabea, A Llavador, E. Sanchez-Ortiga, J. Barreiro, G. Saavedra, J. Garcia-Sucerquia, and M. Martínez-Corral. Fimic: design for ultimate 3d-integral microscopy of in-vivo biological samples. *Biomedical optics express*, 9(1):335–346, 2018.

[5] S. Moreschini, G. Scrofani, R. Bregovic, G. Saavedra, and A. Gotchev. Continuous refocusing for integral microscopy with fourier plane recording. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 216–220. IEEE, 2018.

[6] A. Nwaneshiudu, C. Kuschal, F. Sakamoto, R. Anderson, K. Schwarzenberger, and R. Young. Introduction to confocal microscopy. *Journal of Investigative Dermatology*, 132(12):1–5, 2012.

[7] D. Semwogerere and E. Weeks. Confocal microscopy. *Encyclopedia of biomaterials and biomedical engineering*, 23:1–10, 2005.

[8] K. Schilling, V. Janve, Y. Gao, I. Stepniewska, B. Landman, and A. Anderson. Comparison of 3d orientation distribution functions measured with confocal microscopy and diffusion mri. *Neuroimage*, 129:185–197, 2016.

[9] M. Sheik-Bahae, A. Said, and E. Van Stryland. High-sensitivity, single-beam n 2 measurements. *Optics letters*, 14(17):955–957, 1989.

[10] N. Wagner, F. Beuttenmueller, N. Norlin, J. Gierten, J. C. Boffi, J. Wittbrodt, M. Weigert, L. Hufnagel, R. Prevedel, and A. Kreshuk. Deep learning-enhanced light-field imaging with continuous validation. *Nature Methods*, 18(5):557–563, 2021.

[11] J. Page, F. Saltarin, Y. Belyaev, R. Lyck, and P. Favaro. Learning to reconstruct confocal microscopy stacks from single light field images. *arXiv preprint arXiv:2003.11004*, 2020.

[12] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International journal of computer vision*, 1(1):7–55, 1987.

[13] A. Levin and F. Durand. Linear view synthesis using a dimensionality gap light field prior. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1831–1838. IEEE, 2010.

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

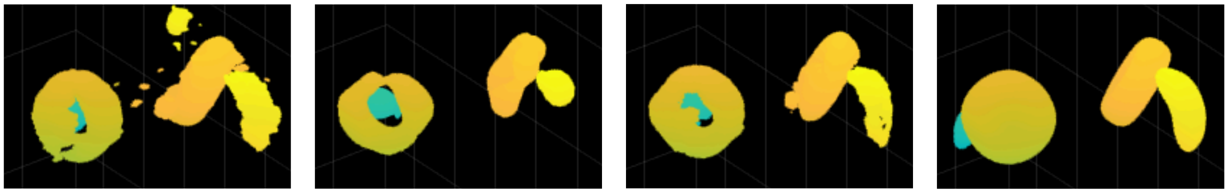
[15] S. Moreschini, R. Bregovic, and A. Gotchev. Shearlet-based light field reconstruction of scenes with non-lambertian properties. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 140–145, 2019.

[16] S. Moreschini, R. Bregovic, and A. Gotchev. CIVIT Dataset: Integral Microscopy with Fourier Plane Recording. *Submitted to Data in Brief*, 2022.

[17] milesial. U-net: Semantic segmentation with pytorch. `https://github.com/milesial/Pytorch-UNet`, November 2021.

[18] L. Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

[19] F. Lomio, E. Skenderi, D. Mohamadi, J. Collin, R. Ghabcheloo, and H. Huttunen. Surface type classification for autonomous robot indoor navigation. *arXiv preprint arXiv:1905.00252*, 2019.



(a) (b) (c) (d)

Figure 9: Fragment of segmentation results for scene *Cells*. (a) H-network output; (b) F-network output; (c) merged HVF output; (d) GT.

	TP	TN	FP	FN	TruePixels	FalsePixels	Precision	Recall	TNR	FNR	FPR
H	0.4856	99.2486	0.0619	0.2038	99.7343	0.2657	0.8869	0.7044	0.9994	0.2956	0.0006
V	0.4344	99.2609	0.0496	0.2550	99.6953	0.3047	0.8975	0.6301	0.9995	0.3699	0.0005
F	0.3254	99.3002	0.0103	0.3640	99.6256	0.3744	0.9692	0.4720	0.9999	0.5280	0.0001
H+V+F	0.4342	99.2944	0.0161	0.2552	99.7286	0.2714	0.9642	0.6298	0.9998	0.3702	0.0002

Table 1: Results for test related to scene Cells

	TP	TN	FP	FN	TruePixels	FalsePixels	Precision	Recall	TNR	FNR	FPR
Cells	0.4342	99.2944	0.0161	0.2552	99.7286	0.2714	0.9642	0.6298	0.9998	0.3702	0.0002
Korona P.	0.1184	99.7913	0.0131	0.0773	99.9096	0.0904	0.9006	0.6050	0.9999	0.3950	0.0001
Pili	0.5642	97.5746	0.1318	1.7294	98.1388	1.8612	0.8106	0.2460	0.9987	0.7540	0.0013
Cells Rec.	0.3328	99.2958	0.0147	0.3566	99.6256	0.3714	0.9576	0.4827	0.9999	0.5173	0.0001

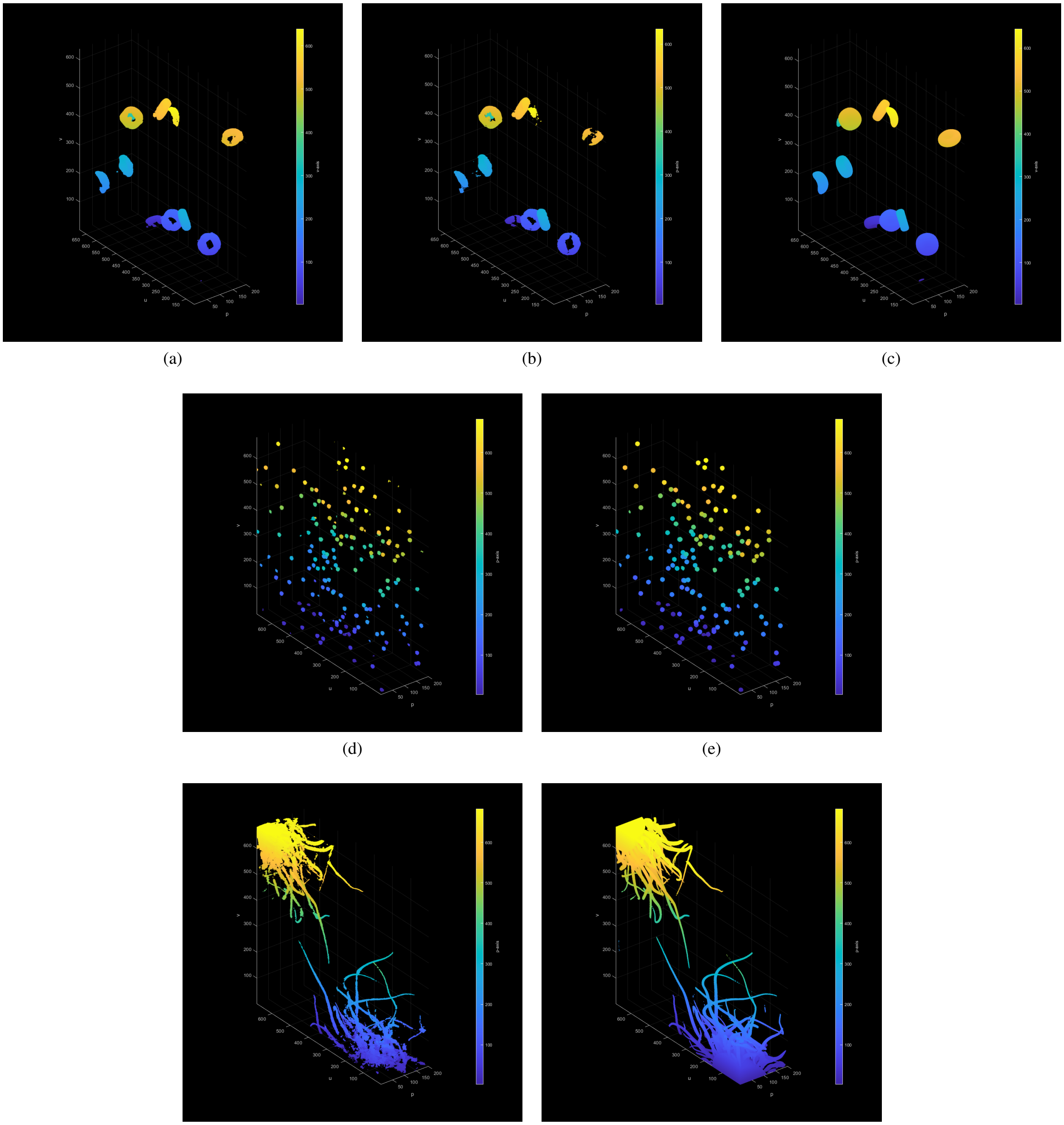
Table 2: H+V+F results comparison for different scenes.

Author Biography

Sergio Moreschini is a Ph.D. candidate in the Faculty of Information Technology and Communication Sciences, Tampere University, Finland. He received his Master's degree in Information and Communication Technology Engineering from University of Roma Tre, Italy (2016). His main research interest focuses on light field analysis and reconstruction, and application of light field analysis for capturing and synthesis systems. He also contributes actively to the domains of empirical software engineering, open-source software quality, data-driven software engineering.

Robert Bregović received the MSc degree in electrical engineering from the University of Zagreb (1998), Croatia, and the Dr. Sc. (Tech.) degree in information technology from the Tampere University of Technology (2003), Finland. Since 1998, he is with Tampere University (former Tampere University of Technology), where he currently works as a Project Manager. His research interests include the design and implementation of digital filters and filterbanks, multirate signal processing, and topics related to acquisition, processing, modeling, and visualization of 3D content.

Atanas Gotchev received the M.Sc. degrees in radio and television engineering (1990) and applied mathematics (1992), the Ph.D. degree in telecommunications from the Technical University of Sofia (1996), and the D.Sc.(Tech.) degree in information technologies from the Tampere University of Technology (2003). He is currently Professor of Signal Processing with the Tampere University. His recent work concentrates on developing methods for multi-sensor 3D scene capture, and light field imaging and display.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 10: Volumetric reconstruction comparison for different scenes. (a) H+V+F test for Cells. (b) H+V+F test for Cells after reconstruction. (c) GT for Cells. (d) H+V+F test for Korona Particles. (e) GT for Korona Particles. (f) H+V+F for Pili. (g) GT for Pili.