Optimizing the Lens Selection Process for Multi-Focus Plenoptic Cameras and Numerical Evaluation

Luca Palmieri Reinhard Koch
Department of Computer Science, Kiel University, Kiel, Germany
{lpa, rk}@informatik.uni-kiel.de

Abstract

The last years have seen a quick rise of digital photography. Plenoptic cameras provide extended capabilities with respect to previous models. Multi-focus cameras enlarge the depth-of-field of the pictures using different focal lengths in the lens composing the array, but questions still arise on how to select and use these lenses. In this work a further insight on the lens selection was made, and a novel method was developed in order to choose the best available lens combination for the disparity estimation. We test different lens combinations, ranking them based on the error and the number of different lenses used, creating a mapping function that relates the virtual depth with the combination that achieves the best result. The results are then organized in a look up table that can be tuned to trade off between performances and accuracy. This allows for fast and accurate lens selection. Moreover, new synthetic images with respective ground truth are provided, in order to confirm that this work performs better than the current state of the art in efficiency and accuracy of the results.

1. Introduction

The idea of plenoptic imaging was introduced by Lippmann in 1908 in [13], but only recent developments have made it possible to actually build devices that are capable of capturing the so-called plenoptic function. In recent years the interest towards such devices is growing, and many different approaches are appearing. The micro-lens array based cameras, that are in fact equivalent to an array of cameras, as proved in [6], have mounted an array of micro lenses between the main lens and the sensor extending the capacity of the device to capture the light field in only one shot.

A specific subset of these cameras is characterized by the use of different micro lenses, particularly with different focal lengths, and they exploit this aspect obtaining for example a wider depth of field. Multi-focus plenoptic cameras were recently introduced in 2012 by Georgiev and Lumsdaine in [5] and a detailed technical explanation of the multi-focus properties is present in [7], but they quickly attract interests in scientific research as shown by the publications addressing these specific cameras.

Different topics were tackled, like a robust automated calibration in [11] and [7], the lens selection and cost function description using semi global matching in [4], a faster feature matching approach in [3] and the whole pipeline until the 3D rendering in [9], but also in industrial application, as shown by the fast growth of companies exploiting the technology.

Such cameras can be used for entertainment as Lytro [14] is doing, or for inspection and modelling, like Raytrix [17] is doing, but also for photography related taks, like the most recent approach brought by the Light company [12], consisting of a pocket-size camera that emulates the performance of a DSLR camera using multiple lenses with different focal lengths.

Our approach targets one of the mostly used models that accounts for three different types of lenses (with three different focal lengths) provided by Raytrix, but has the advantage of being quite flexible and could be applied to all devices that use different lens types and need a strategy to accurately and efficiently select each time which one to use.

1.1. Structure of the Paper

Section 2 reviews related works, to give the reader an overview of the state of the art techniques related to the topic; in Section 3 we show the specific case of the plenoptic cameras that we are using, and we make an assumption about the disparity estimation with a detailed motivation; in Section 4 we go through the first step of the proposed approach, that uses ground truth generated data to evaluate different combinations; in Section 5 we combine the results into a specialized structure that allows an easy and efficient execution of the lens selection algorithm; finally, Section 6 compares the results obtained against the previously known techniques both with synthetic and real data (acquired with a Raytrix camera) and Section 7 provides a conclusion and

some possible future developments. Appendix A is used to give the reader further insight on how synthetic and real data are generated.

2. Related Work

Many approaches have been proposed to address the challenge of creating accurate disparity maps from image captured with a plenoptic camera: we focus on the aspects that make this camera unique in his genre, the micro lenses array and more specifically the lenses and the characteristics of their usage.

To reconstruct the image from the light field as captured by the lens array, one needs to select multiple adjacent lenses and compute depth (or disparity) from them. The depth is needed to collect the correct light rays for the sharp real image.

The lens selection problem remains an open challenge of high importance, because it addresses the very nature of the cameras: the micro-lenses array that allows the camera to capture the light field in only one shot and controls the trade-off between lateral and spatial resolution, which adapts each camera for specific purposes like refocusing or estimating the disparity map and reconstructing the three dimensional geometry of the scene.

Previously proposed methods about lens selection always assume some geometrical information about the pixel, whose disparity or depth has to be computed, and are used to refine the estimation: they can be divided in two categories:

1. Using the geometrical information, limit the lenses range and check on every lens the amount of defocus blur and the minimum overlapping in order to understand if they could positively affect the estimation.

2. Divide the world into slices on the z-direction and assign at every slice a certain range of lenses.

The first approach was proposed by Fleischmann and Koch in [4], with an adaptive strategy that uses a first estimate of the disparity to select lenses, discarding the ones where the overlapping was without a certain threshold: their first estimation is efficient, but the adaptive strategy always involves some computational effort and does not reach the highest precision in the lens selection.

As an example of the second approach we pick the most recent paper on the topic from Ferreira and Goncalves [3], where they divided the space into four quartiles. When a point seemed to belong to a certain sector, they assigned the lenses range and a predetermined combination: the idea of dividing the space into slices is functional in terms of performances, since it does not involve further computations, but it lacks in accuracy, since they use only four different areas and three different combinations, and they discard many lenses just because of the difference in the focal lengths, while those still may contain useful information.

The proposed approaches for the lens selection seem not to reach the optimal solution, lacking in terms of either accuracy or performances, mainly because of two issues that are common to this kind of data:

1. To predict which lenses should be used for a point or a lens, some geometric information about the position of that point should be known, and the accuracy of this information greatly affects the final result.

2. It's challenging to capture images with ground truth to evaluate different methods, due to the particularity of the cameras.

We address both problems and propose our solution in the following.

3. Initial Lens Selection

In this paper we deal with a very specific version of the multi-focus plenoptic cameras: the approach was developed using the Raytrix cameras with three different focal types and the lenses arranged in the hexagonal grid as shown in [4] and in Fig. 1.

The approach we propose is flexible, and it can be adapted to all multi-focus plenoptic cameras, where the lenses of different focal types need to be selected for the disparity mapping or any other application.

The method is mainly divided into two steps:

- First, we compute a virtual data set consisting of different known depth planes. Using this calibration data we test lens combinations that are optimized with respect to efficiency and accuracy.

- Next, we create a look up table structure where data is stored and can be used efficiently during runtime.

We will discuss this in detail respectively in Section 4 and 5.

The mentioned calibration process is highly time-consuming, but it has to be performed only once and then the results will be stored in order to be used in every successive execution, in a similar way to a calibration process, allowing an efficient computation at runtime.

Before continuing, we briefly describe two concepts that are important for the rest of the paper.

Virtual Depth

The concept of *virtual depth*, introduced in [16], is defined as the number of different lenses in a row that image a point, so a point with virtual depth N, would be imaged by N different lenses belonging to the same line, i.e. N different horizontal viewpoints.

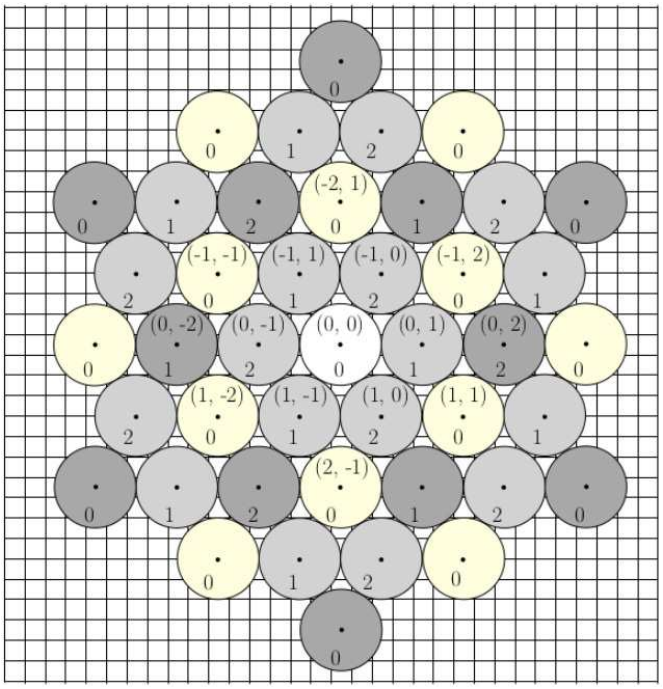


Figure 1: The lens grid of a multi-focus plenoptic camera: in this case a Raytrix camera with three different focal lengths is depicted. The numbers $0, 1, 2$ indicate the focal type of each lens, and the coordinates (x, y) are relative to the central lens. Picture taken from [4]

.

Intuitively, the virtual depth is inversely proportional to the disparity: we can express that in mathematical terms in Eq. 1

$$VD = \frac{K}{d} \propto \frac{1}{d} \tag{1}$$

Where VD stands for the virtual depth, d for the disparity and K for a constant factor that is related to the metric calibration parameters of the lens array.

Disparity Estimation

Many techniques that compute the disparity are available. Based on the literature, we choose to use semi global block matching algorithm first introduced in [8] and used in [4], since it achieves better results as compared to the feature matching approach implemented in [3].

More sophisticated approaches exploiting the multi-view nature of these images will be inspected in future research.

3.1. Initial Virtual Depth Creation

Our method is based on the relation between a disparity and the combination of the lenses (that will lead to a refined version of the disparity), hence we need to compute a first hypothesis on the position of the point in space, which does not have to be completely accurate. Since at this point we can trade accuracy for computational speed, we choose Fleischmann and Koch's [4] idea, making an initial guess using a small number of lenses and a block matching approach.

Nevertheless we reviewed different possibilities for this task: without changing the estimation method, we can select different combinations of lenses to reach a better results without loss of performances, as seen in Fig. 2:

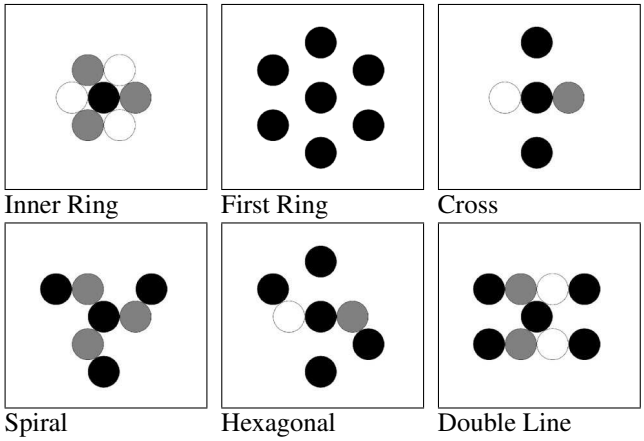


Figure 2: Different combinations of lenses that could be used for the initial virtual depth creation. They are chosen because of their structure around the central lens, and the gray colors represent their focal type: lenses with same color belong to same focal type.

The choices were made to tackle some particular characteristics of the lens grid: inner and first ring are most likely to be used because of the smaller baseline, being able to estimate disparity for both close and far objects.

Since the inner ring consists in lenses that have different focal lengths, they are not reliable, but necessary for close objects, whose virtual depth is small. The first ring, that contains only lenses with same focal type, thus with the same amount of defocus blur, should be more reliable.

We evaluated the different combinations on two datasets with respective ground truth, the first one also used in [4] to evaluate the results and the second one introduced to have more structure in the scene, in order to obtain both visual and numerical data to support our choice.

The visual feedback of both datasets is in line with the theoretical assumptions: the disparity computed with the inner ring is quite accurate for close points, but highly noisy for background points.

The opposite happens for the first ring, that addresses the far point in the correct way, but misses the correspondence for points with small virtual depth, as clearly visible in Fig. 3 and Fig. 4 where the centers of the micro images referring

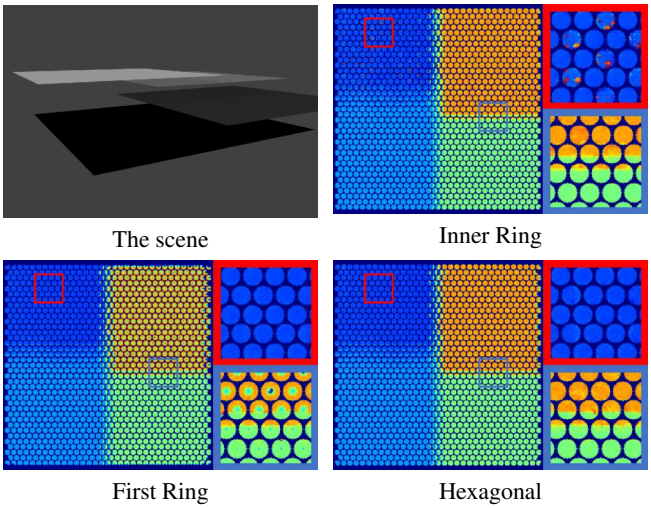


Figure 3: Evaluation on the first dataset consisting of four planes at different distances on the z-axis, to highlight errors at different disparities; only three of the combinations are shown here.

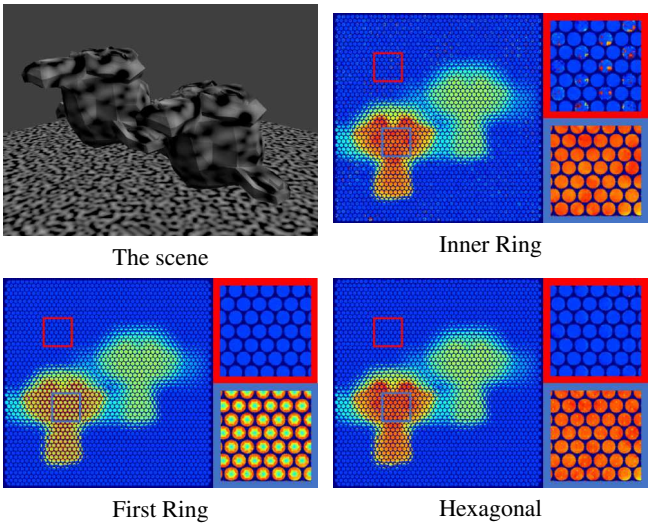


Figure 4: Evaluation on a more complex dataset where two objects show some structure, to see how each combination deals with border and objects close to each other.

to points close to the camera are not correctly computed, due to a large baseline.

We then see that the new proposed combinations use a mixed combination of both rings to address a larger depth of field, at a price of a lower accuracy.

Combinations with less lenses were tried with the scope of reducing the computational cost and increase the performances, but as shown the estimation with only four lenses is noisy and would affect negatively the final estimation.

Combinations with the same number of lenses show a better quality, particularly the best results are achieved for the *hexagonal* strategy, that uses six lenses as depicted in Fig. 2 (Hexagonal): two lenses with different focal lengths (white and gray) and four lenses of the same type around the central one. The *double line* combination achieves even more accurate outcomes, but at a price of a higher number of lenses.

Combination	**Four Planes**		**Objects**		**Average**		**Lenses**
	Avg.	Std.	Avg.	Std.	Avg.	Std.	
Inner Ring	0.305	0.568	0.427	0.851	0.366	0.710	6
First Ring*	0.359	0.664	0.319	0.666	0.339	0.665	6
Cross	0.267	0.438	0.319	0.671	0.293	0.555	4
Spiral	0.254	0.402	0.284	0.562	0.269	0.481	6
Hexagonal	0.244	0.387	0.282	0.392	0.263	0.390	6
Double Line	0.237	0.364	0.273	0.570	0.255	0.467	8

Table 1: Combinations and respective errors. Values are expressed in pixels and the disparity range is [0.5, 12.5]. We report here more combinations, to show how different approaches would deal with the problem.
* = combination used in [4]

The errors reported in Table 1 are computed using a simple absolute difference function, taking into consideration only valid pixels: for every micro-image only the pixels contained in a circle with a diameter slightly smaller than the image side are taken into account, with the exact value of the diameter set during the calibration process to avoid vignetting errors.

We print both average and standard variation to give an idea of how the error is distributed: a smaller variance would translate into a more robust outcome, that would be preferable in our case, since large error could lead to a wrong lens selection and thus to a wrong final estimation.

Analyzing both visual and numerical results, we select the *hexagonal* combination, that seems to solve at the best the trade-off between performances and accuracy.

We use this selection in the rest of the paper, so that every time we refer to the initial disparity guess, we mean the value obtained as explained above with the *hexagonal* combination.

4. Optimizing the Lens Selection

Based on the previously discussed assumption, we are now ready to proceed to the second step: having an initial guess of the point position allows us to create a direct mapping from this information to the best possible relative com-

The disparity maps are shown in colored version for an easier visualization: red color means high disparity, and blue indicates lower disparity values. The two datasets are synthetically generated as explained in Appendix A.2.

bination of neighbouring lenses to fully exploit the characteristics of the micro lens array.

The novelty of this approach consists in the creation of a ground truth set of data by simulating the array of cameras: we set the position of our camera and we generate the images of a plane at a certain distance z from the camera, as if it was capture from a multi-focus plenoptic camera, using the right focal length for each micro-lens. To see the details about the creation of these synthetic planes, we refer to Appendix A.2.

Moving this synthetic plane from close to far with respect to the camera allows us to create a test set for all different positions that a point could assume in space and its corresponding ground truth, since we know the exact distance between plane and the virtual camera position.

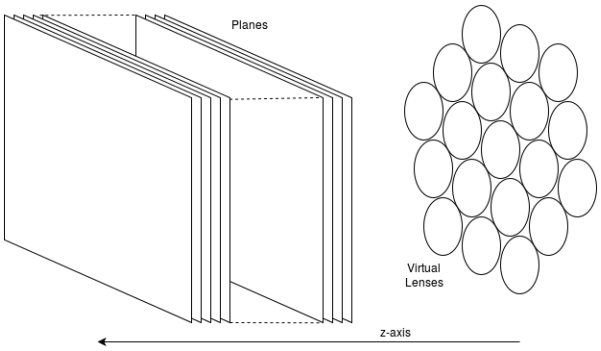


Figure 5: Representation of the planes and the lenses.

At this point we also want to stress that our specific cameras use lenses with three different focal lengths, so we need to evaluate the error and the combinations per each lens type, i.e. taking into account the amount of defocus blur of each particular lens.

An important parameter that has to be tuned is the distance between each plane: evaluating our combinations on planes that are too close to each other will result in erroneous selection, since our initial disparity guess cannot be particularly accurate, and using planes too far away will use the same combination for points that could benefit from a different one.

Our choice is to use the virtual depth measurement, as explained above, which has the advantage of being scale-independent, and extends the flexibility of our approach, also leaving space for particular application when the range of the scene has specific constraints and we are not interested in the whole depth of field of the camera.

We have chosen to use textured planes because of their spatial structure, being able to divide the world space in slices, and since the disparity estimation is not based on the structure, but just on the intensity of the pixel, textured planes fit our requirement for this task.

4.1. Comparisons among Combinations

The planes dataset was used to run all the disparity estimation, and every different combination was evaluated using the same error function explained above, absolute difference with respect to the ground truth, discarding the border of the lens.

We evaluated several combinations through the whole range of the scene, and here we report the ones that gave the most significant outcomes.

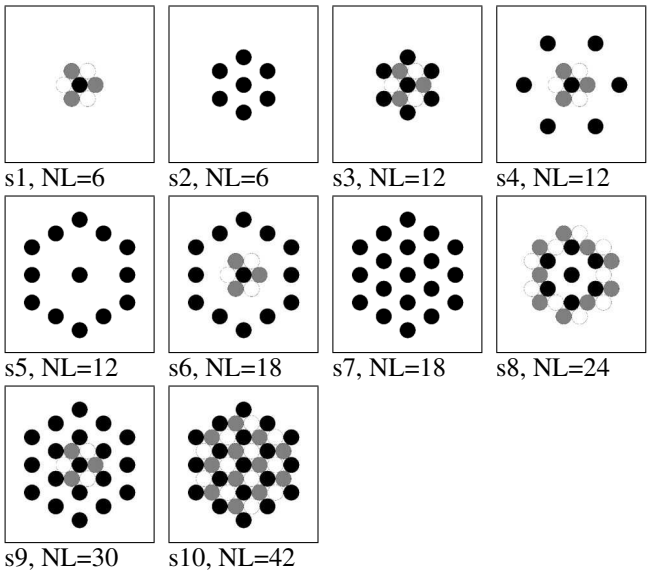


Figure 6: Different combinations: central lens is always shown in black, and for the other lenses every color represents a different focal type.
The number of lenses (NL) is also reported.

As expected, the combinations that use adjacent lenses report better results on the close range (from 2 to 4 virtual depths) but show a large error when the distance from the camera increases, resulting useless in those cases; the opposite happens for the combinations that use lenses with a larger baseline, as they need at least between 4 or 5 virtual depths to start working properly.

Once we calculated the results, we can sort them based on the average of the error in the disparity image and pick the best combination for every value of virtual depth. One can see that different combinations obtain similar results in the central area, where many different combinations are possible and many correspondences are available, thus increasing the difficulty of a correct choice.

Our idea is to develop two possible combinations for each slice: one which would give the most accurate result and one which would use the lowest possible number of lenses given a certain error threshold, in order to boost performances keeping a low error.

	Best Combinations		Perform.		Accuracy	
VD	**LT**	**CBs**	**CB**	**NL**	**CB**	**NL**
2	0	s1	s1	6	s1	6
	1	s1	s1	6	s1	6
	2	s1	s1	6	s1	6
3	0	s1	s1	6	s1	6
	1	s1	s1	6	s1	6
	2	s1	s1	6	s1	6
4	0	s2, s3	s2	6	s3	12
	1	s2, s3, s8	s2	6	s3	12
	2	s2, s3	s2	6	s3	12
5	0	s2	s2	6	s2	6
	1	s2, s3	s2	6	s3	12
	2	s2, s3, s8	s2	6	s3	12
6	0	s5, s2, s7	s2	6	s5	12
	1	s4,s6,s2,s3	s2	6	s3	12
	2	s10, s3, s8, s9	s3	12	s8	24
7	0	s5, s2, s7	s2	6	s5	12
	1	s2, s3, s7	s2	6	s3	12
	2	s2, s3, s8	s2	6	s8	24
8	0	s5, s2, s7	s2	6	s5	12
	1	s5, s2, s7, s9	s2	6	s9	30
	2	s2, s8	s2	6	s8	24
9	0	s5, s7	s2	6	s7	18
	1	s10, s2, s7	s2	6	s7	30
	2	s10, s2, s3, s8	s2	6	s8	24
10	0	s2, s7	s2	6	s7	18
	1	s5, s6, s3, s9	s3	12	s9	30
	2	s2, s3, s8, s9	s2	6	s7	18
11	0	s2, s7	s2	6	s7	18
	1	s3, s10	s3	12	s3	12
	2	s2, s3, s7, s8	s2	6	s8	24
12	0	s6, s9	s6	18	s9	30
	1	s3, s9	s3	12	s9	30
	2	s2, s3, s8, s9	s2	6	s9	30
13	0	s6, s9	s6	18	s9	30
	1	s3	s3	12	s3	12
	2	s2, s3, s8	s2	6	s3	12
14	0	s6, s2, s7, s9	s2	6	s9	30
	1	s3	s3	12	s3	12
	2	s2, s3, s8	s2	6	s3	12

Table 2: Outcomes of the lens selection step: the best combinations that satisfy Eq. (2) are grouped in the third column; choices for best performance and accuracy are shown in the columns on the right side.

Legend	
VD	Virtual Depth
LT	Lens Type
CB	Combination
NL	Number of Lenses

Table 3: Legend for Table 2

To retrieve such combinations, the outcomes are ranked for accuracy, then all combinations that satisfy Eq. (2) are grouped and sorted this time based on performance, using the total number of lenses used for the estimation.

$$\mu_{f_j,i} < 1.5\mu_{min,i} \tag{2}$$

Where $\mu_{f_j,i}$ is the mean of the error for the j-th combination and for virtual depth i and $\mu_{min,i}$ is the minimum error for virtual depth i achieved by any of the combinations.

The results reported in Table 2 show the difference between the lens types and their need of different combinations in order to reach the highest accuracy.

The reported best combinations have similar outputs, so other choices are also possible, based on particular scenes or different parameters when acquiring or generating the scene, or also introducing different constraints on errors or maximum number of lenses. However, in our case we found these combinations to be the best.

The idea we want to highlight here is the fact that every lens, based on his focal type, could benefit a different patch of lenses for the disparity estimation: moreover, as the virtual depth increases, combinations that exploit lenses with higher baseline seem to achieve more accurate estimations.

Finally, it can be noticed that the different amount of blur highly affects the estimation of disparity images: as also found in [3], most of the lenses that will actually be used belong to the same lens type of the central lens.

5. Storing the Lens Selection

Once we gather the results of our simulation, we need to store them in a way that allows us to use it in every next execution of the software: many structures could be used for this purpose, but our choice has fallen on a structure that resemble the characteristics of a look up table, that allows us to retrieve the best possible combination for each lens and lens type.

5.1. Choice of the Structure

The look up table structure seems to be the best solution is this case for different reasons, namely:

- **Efficiency**: once we have computed the initial guess, the computational effort needed to retrieve the relative position of the lenses of a particular combination from a virtual depth value is only a fetching instruction
- **Flexibility:** using virtual depth we have a measure that is scale-independent and the same structure can be used for all subsequent acquisitions.
- **Different choices available:** depending on the specific task, the user may want a different outcome; if quality of the results is the primary concern, the best combination of available lenses is selected, but if the operation to be performed has more speed constraint and needs a quicker execution, a parameter controlling

the choice, changes the selection towards the quickest combination (i.e. the combination that uses lowest number of lenses and still achieves an error lower than a certain threshold)

5.2. Creation of the Structure

Based on these ideas, we created a structure that can manage the trade-off between accuracy and performance, storing both combinations at the same time, and giving as output only one of them when needed

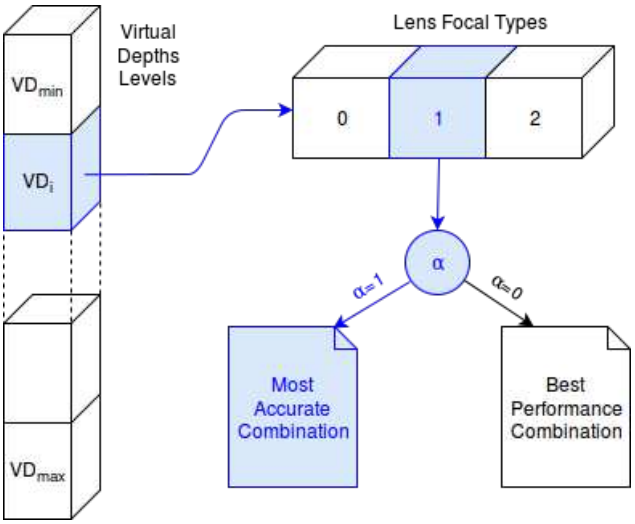


Figure 7: Look up table structure, with an example of a point with an initial guess of VD_i, belonging to lens with focal type 1, with parameter α optimized for accuracy.

The table has D levels, where D is the number of *slices* in which we divide the space (in our case $D = 13$) that have LT entries each, where LT is the number of lens types present in the camera (in our case $LT = 3$): those entries contain the relative combination of the lenses to be used for the estimation, and a parameter α is used to select which combination (most accurate or best performance) will be used, as depicted in Fig. 7.

If this approach had to be adapted to a different type of plenoptic camera, a simple change of the parameters would be enough to use the same structure in any other case.

5.3. Runtime Execution

Assuming we have calibrated our camera and created the look up table, we also create our internal mapping from disparity to virtual depth values, knowing that the virtual depth (VD) is proportional to the inverse of the disparity (d) as shown in Eq. (1).

The execution at runtime is controlled by a simple lookup: we feed as input three values, namely the virtual depth (VD), the lens type (lt) and the parameter controlling the trade-off between accuracy and performances.

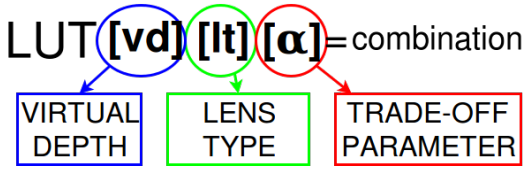


Figure 8: Graphic representation of the function that controls the look up table.

The output of such a function is the relative position of the lenses that should be used to achieve the best results during disparity estimation.

6. Results

We show some results to evaluate the proposed approach with respect to the state of the art technique; we use both synthetic images with generated ground truth to get some numerical value as an objective evaluation and real images without ground truth as a visual subjective evaluation.

The achieved results show an improvement of the accuracy of the estimation in some problematic areas, namely border of irregularly shaped objects, smooth textureless areas and fine structures, that are challenging for the task of disparity estimation.

We moreover stress the fact that this improvement is achieved without any additional computational efforts: while the calibration step and the creation of look up table is costly and time-consuming (but it has to be done only once), at runtime execution the algorithm does not need to compute any calculation and is able to choose the lenses to be used with only a simple fetching instruction in the look up table, resulting in an efficient approach.

Generally the selected combination uses a high number of comparisons because we are looking for an accurate disparity map, but due to the flexibility of the proposed approach we can also change the orientation towards a more performant approach changing the threshold value that creates the different combination in the look up table, allowing the targeting of different applications.

6.1. Real scenes

The reported images are excerpts of the full scenes from Raytrix [17] and focus on certain details that we want to focus on; we compare them with the previous method implemented in [4] and with the results extracted from the Raytrix RxLive software [18]: the parameters used to obtain such disparity map per lens are tuned towards a more dense result, that is not necessarily the best result, but it's important in our comparisons to highlight the areas where the estimation is most challenging, without the successive filling algorithm.

The first scene, shown in Fig. 9, is the widely used Watch

	Computed with [4]		Ours	
Scene	Avg.	Std.	Avg.	Std.
Four Planes				
Lens type 0	0.29	0.56	0.27	0.48
Lens type 1	0.26	0.44	0.23	0.33
Lens type 2	0.26	0.49	0.23	0.35
Platonic				
Lens type 0	0.47	0.41	0.46	0.41
Lens type 1	0.44	0.40	0.42	0.40
Lens type 2	0.41	0.40	0.40	0.40
Tomb				
Lens type 0	0.28	0.27	0.24	0.25
Lens type 1	0.27	0.25	0.23	0.23
Lens type 2	0.28	0.23	0.22	0.19

Table 4: Mean Error and Standard Deviation for the three synthetic scenes. Values are expressed in pixels.

scene and gives us a good example of how a different selection of the lenses can affect the final estimation of challenging areas, like the textureless background surface and the textured plane, exhibiting a high level of noise that is reduced in our implementation, obtaining a more robust and smooth estimation.

Second scene, Fig. 10, consists in a more challenging outdoor scene, with a zoomed area relative to the left hand of the girl in the front, where the improvement with respect to the previous approach is quite small, but is possible to notice how the small details that cannot be reconstructed with enough reliability from the RxLive software are computed with a high accuracy and detail.

Fig. 11 finally highlights the smoothness and robustness of the estimation for detailed and highly textured areas, where the precision is raised and the noise in the estimation is almost removed.

6.2. Synthetic Scenes

The synthetic images consist in a fundamental step for this kind of disparity per lens images: up to our knowledge, no other methods were proposed to produce a numerical output to measure accuracy of the final estimation.

The images are more trivial and do not yet reach the complexity of a real scene: this is a task that we are currently addressing for the future to extend our qualitative results, but are still very helpful for evaluation purposes at the moment.

The scenes are part of the dataset developed in [10] and available at the 4D Light field Benchmark website [15], but due to our settings, they have different point of view and disparity ranges.

The difference between the two estimations are not large and can appear unclear at a first glance, but as is visible from Table 4 our approach slightly reduces the error and obtains a lower standard deviation, meaning the estimation is more robust.

7. Conclusion

In this paper we tackled an issue common to multi-focus plenoptic cameras that represent a still open problem: our contribution is not only important in terms of positive quality of the results, but also in terms of the characteristics of the approach.

It takes from the idea of training the camera to achieve higher accuracy in the final outcomes, trading a computationally expensive lens selection phase to be done once allowing a multiple times more efficient runtime execution, that up to our knowledge was not proposed yet in terms of lenses estimation.

This approach works in this specific environment due to our assumption relative to an initial disparity guess and due to the nature of the problem: the lens selection is performed on the estimated distance in the z-axis of the point, without relying on texture or color information, therefore the training step can be done on planes while the execution process will most likely be done on different shapes without changing the outcomes.

Since in the last years these kind of cameras are developing a high potential for different number of applications, as pointed out by the recent introduction of the L16 Light camera that exploit many different lenses with three different focal lengths for enhanced photography, hence the lens selection process is worth a further insight to exploit the full potential of the cameras.

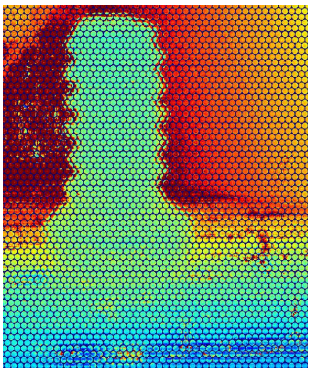
Secondly, we were able to provide some light field images for a numerical evaluation, a missing element in the disparity per lens estimation field, where, as shown in [3] and [4], apart from a really basic scene, only a visual evaluation was possible.

We start to introduce new images and we look forward to building a small dataset to allow different estimation techniques to be compared.

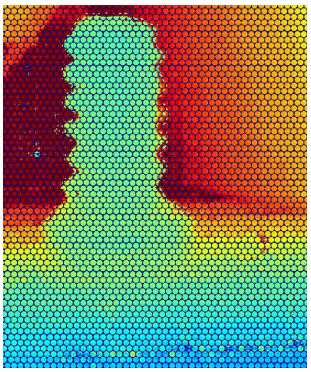
Next step would be to focus on the comparison between those lenses, trying to evaluate which would be the optimal similarity measure or methodology to be used for that purpose.

6.1. Real Scenes

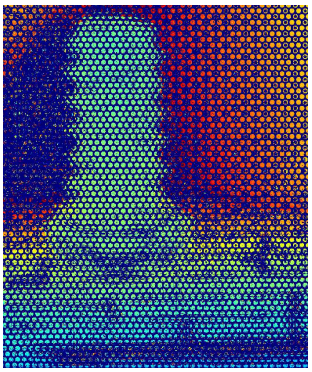
3D Model - Watch



Disparity computed with previous approach [4]



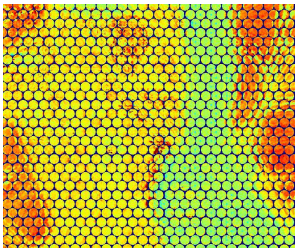
Disparity computed with our approach



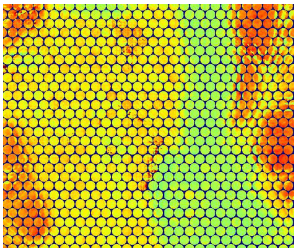
Disparity computed with Raytrix software

Figure 9: Part of the Watch scene.

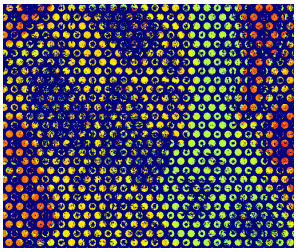
3D Model - Forest



Disparity computed with previous approach [4]



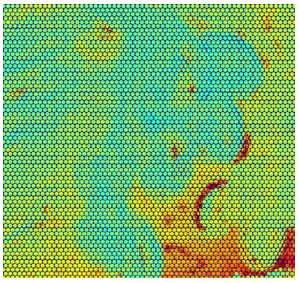
Disparity computed with our approach



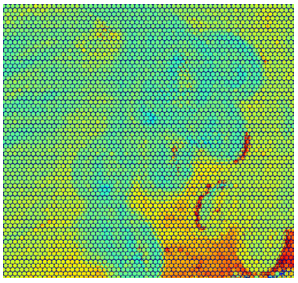
Disparity computed with Raytrix software

Figure 10: Part of the Forest scene.

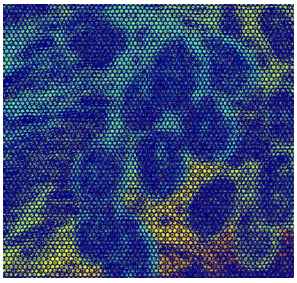
3D Model - Obstkorb



Disparity computed with previous approach [4]



Disparity computed with our approach

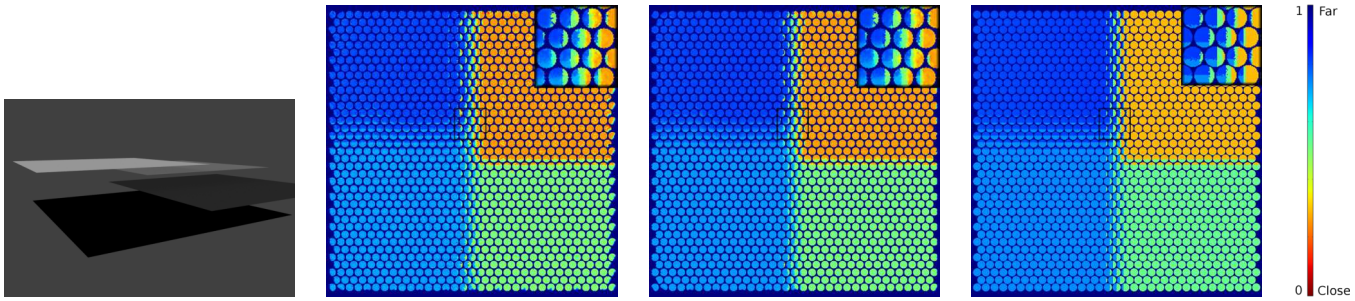


Disparity computed with Raytrix software

Figure 11: Part of the Obstkorb scene.

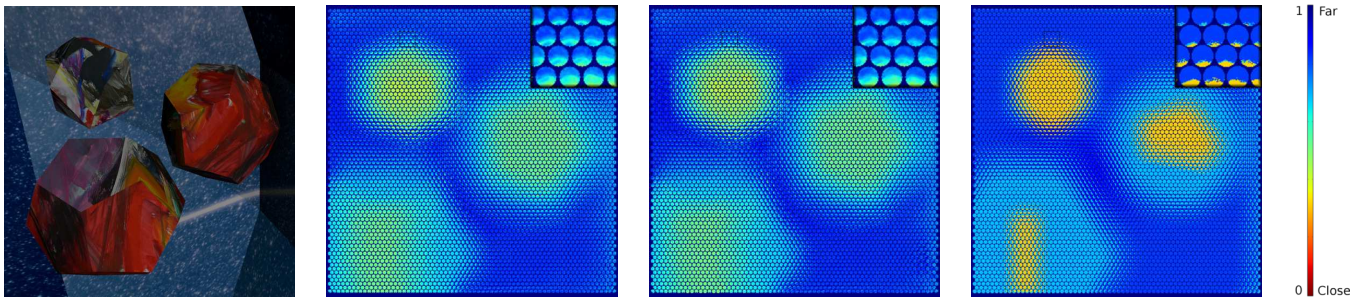
Scenes are provided by Raytrix [17].

6.2. Synthetic Scenes



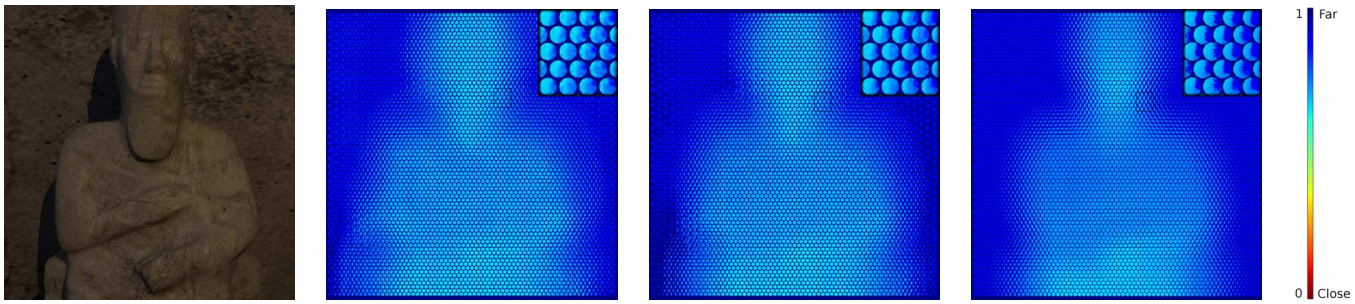
3D Scene - Four Planes | Disparity computed with previous approach [4] | Disparity computed with our approach | Ground Truth

Figure 12: The Four Planes scene.



3D Scene - Platonic | Disparity computed with previous approach [4] | Disparity computed with our approach | Ground Truth

Figure 13: Part of the Platonic Scene.



3D Scene - Tomb | Disparity computed with previous approach [4] | Disparity computed with our approach | Ground Truth

Figure 14: Part of the Tomb Scene.

The first scene (Four Planes) was produced by O.Fleischmann in [4].

The last two scenes (Platonic and Tomb) are part of the 4D Light Field Dataset [15] and the 3D models were used to generate the images with the respective ground truth.

A. Data Generation

Here we report more details about the generation of real and synthetic scenes with ground truth data.

A.1. Real Scenes

The real scenes were captured using Raytrix cameras. The 3D models and the disparity maps were extracted using the RxLive4.0 software provided by Raytrix [17].

A.2. Generating Synthetic Data with Ground Truth

The generation of the ground truth is a more complex process: since it's not yet publicly available any dataset with ground truth of disparity map per lens, evaluations is a challenging issue.

We provide ground truth images that simulate the image acquisition, even though they are not completely the same: the imaging process of Raytrix camera consists in projecting the real world scene through the main lens to an intermediate image in a virtual space, that stretches the relative depths and makes it easier to estimate larger disparities values. The particularity of their technique is that the intermediate image is virtually projected behind the micro lenses array, in a counter intuitive way not not possible to reproduce with other cameras.

Our simulated images use the same idea, recreating the situation where the a virtual micro lenses array see the intermediate image, but this image is in fact in front of the array of micro lenses. The virtual depth of an image is thus inverted, meaning that an object with a large virtual depth would be close to the camera in a real image, and far away in a synthetic generated image, but since we focus on the mapping from virtual depth to lenses combination, these images fit perfectly our needs.

This idea was already exploited in [4] for a numerical evaluation, we extended this to scenes that consist in real benchmark for light field disparity estimation by recreating them with our synthetic generation pipeline.

Acknowledgment

The work in this paper was funded from the European Unions Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 676401, European Training Network on Full Parallax Imaging.

Moreover the authors would like to thank O. Johannsen (University of Konstanz) for sharing some of the scenes of the benchmark for light field evaluation, A. Petersen (former Raytrix Gmbh) for sharing scenes captured with Raytrix cameras and O. Fleischmann (former University of Kiel) for the contribution in the generation of the synthetic scenes.

References

[1] M. Damghanian, R. Olsson, M. Sjostrom, A. Erdmann, and C. Perwass. Spatial resolution in a multi-focus plenoptic camera. In *IEEE International Conference on Image Processing (ICIP)*, 2014.

[2] J. R. Bergen E. H. Adelson. *The Plenoptic Function and the Elements of Early Vision*. Cambridge, 1991.

[3] R. Ferreira and N. Goncalves. Fast and accurate micro lenses depth maps for multi-focus light field cameras. In *German Conference on Pattern Recognition (GCPR)*, 2016. 1, 2, 3, 6, 8

[4] Oliver Fleischmann and Reinhard Koch. *Lens-Based Depth Estimation for Multi-focus Plenoptic Cameras*, pages 410–420. Springer International Publishing, 2014. 1, 2, 3, 4, 7, 8, 9, 10, 11

[5] T. Georgiev and A. Lumsdaine. The multi-focus plenoptic camera. In *SPIE Electronic Imaging*, January 2012. 1

[6] T. Georgiev, A. Lumsdaine, and S. Goma. Plenoptic principal planes. In *Imaging Systems and Applications (IS), OSA Topical Meeting*, July 2011. 1

[7] C. Heinze, S. Spyropoulos, S. Hussmann, and C. Perwaß. Automated robust metric calibration algorithm for multifocus plenoptic cameras. *IEEE Transactions on Instrumentation and Measurement*, 65 (5), May 2016. 1

[8] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341, February 2008. 3

[9] M. Hog, N. Sabater, B. Vandame, and V. Drazic. An image rendering pipeline for focused plenoptic cameras. *Hal*, 2016. 1

[10] 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 (ACCV)*, 2016. 8

[11] O. Johannsen, C. Heinze, B. Goldluecke, and C. Perwass. On the calibration of focused plenoptic cameras. In *GCPR Workshop on Imaging New Modalities*, 2013. 1

[12] Light. https://www.light.co/. 1

[13] G. Lippmann. Epreuves réversibles. photographies intégrales. *Académie des sciences*, pages pp. 446–451, 1908. 1

[14] Lytro. https://www.lytro.com. 1

[15] University of Konstanz and Heidelberg Collaboratory for Image Processing. http://hci-lightfield.iwr.uni-heidelberg.de/, May 2017. 8, 10

[16] C. Perwaß and L. Wietzke. Single lens 3d-camera with extended depth-of-field. In *Proceedings of SPIE - The International Society for Optical Engineering*. Addison-Wesley, February 2012. 2

[17] Raytrix. https://www.raytrix.de. 1, 7, 9, 11

[18] RxLive Software. https://www.raytrix.de/downloads/. 7