Super-rays for Efficient Light Field Processing

Matthieu Hog, Neus Sabater, Christine Guillemot

Super-rays for Efficient Light Field Processing

Matthieu Hog, Neus Sabater, Christine Guillemot

***Abstract*—Light field acquisition devices allow capturing scenes with unmatched post-processing possibilities. However, the huge amount of high dimensional data poses challenging problems to light field processing in interactive time. In order to enable light field processing with a tractable complexity, in this paper, we address the problem of light field over-segmentation. We introduce the concept of super-ray, which is a grouping of rays within *and* across views, as a key component of a light field processing pipeline. The proposed approach is simple, fast, accurate, easily parallelisable, and does not need a dense depth estimation. We demonstrate experimentally the efficiency of the proposed approach on real and synthetic datasets, for sparsely and densely sampled light fields. As super-rays capture a coarse scene geometry information, we also present how they can be used for real time light field segmentation and correcting refocusing angular aliasing.**

I. Introduction

OVER the past few years, light field capturing devices have gained in popularity, both at the research and the industrial level. These devices can be broadly classified into the following categories. Plenoptic cameras use an array of microlenses placed in front of the sensor of a classical camera in order to capture light fields [1], leading to trade spatial resolution for angular resolution. Although this trade-off can be controlled [2], the captured light field is always angularly dense (the disparity is of the order of the pixel). Plenoptic cameras typically target consumer photography via their refocusing feature (e.g. Lytro Illum[1]), but also the industrial market thanks to the capabilities they offer for passive, accurate, monocular depth estimation (e.g. Raytrix[2]). Light fields can also be captured by camera arrays [3], [4]. While being significantly more expensive and harder to use, hence dedicated to professionals, they offer the best spatial resolution (but low angular resolution) and a much larger baseline, making them suitable for applications requiring long range depth estimation, change of viewpoint and view synthesis, such as AR content capture (e.g. Lytro Immerge[3]) or movie post production (e.g. Light-field Technology at Fraunhofer [5][4]). A third category of acquisition devices is the camera gantry (e.g. the Stanford Lego Gantry[5]), where a camera is moved along a plane and shots are taken at regular intervals. These devices are easy to use and implement, and can in theory provide high spatial and angular resolution. However, this type of device is limited to the capture of static light fields. Finally, recent and emerging smart phones are equipped either with several cameras (e.g Iphone 7, Amazon Fire Phone), or with a single specialised sensor (e.g Pelican Imaging [5], ProFUSION-25C [5], or with a Wafer-level-optics camera array [6]) which could, to some extent, capture light fields, not as angularly dense as plenoptic cameras. Because they need to be compact, they also have a much smaller baseline and resolution than camera arrays.

The volume of data inherent to light fields, for all capturing devices, is a real issue for user interaction that requires near real-time processing, potentially on devices having limited computational power. This becomes even more critical for light field videos. A second issue is the ease of use. While users are familiar with 2D image editing, light field editing boils down to edit a 3D capture embedded into a 4D signal [7]. One objective of the paper is to enable user interaction with the whole light field while entering inputs on one view only. State of the art light field editing methods either only deal with densely sampled light fields or use a dense depth estimation to perform user-guided segmentation [8]–[10] or to propagate user inputs [11]–[15]. Despite the latest advances [16]–[19] in light field depth estimation, these methods use computationally expensive regularisation to obtain satisfactory depth maps.

Our goal is instead to propose a solution for unsupervised light field over-segmentation which would in addition be angular sampling agnostic and rely less on depth estimation. Our approach is motivated by the observation that, for most editing applications, it might be more important to have accurate object boundaries, even with coarse depth information, than having a refined depth map. We show in Section IV, as already noticed by the authors in [7], that a dense and accurate (in terms of disparity values) depth estimation is not needed for some typical light field applications.

To treat the aforementioned problems, we introduce in this paper the concept of *super-ray* which is the counterpart of super-pixels [20] for light fields. The major difference with conventional super-pixels and super-voxels is that super-rays group perceptually similar *and* corresponding pixels across several views. In other words, super-rays are groups of rays of similar color coming from the same scene area. We then propose what we believe to be the first light field over-segmentation algorithm. It is inspired by SLIC [21], a state of the art super-pixel generation method with good properties in terms of accuracy over complexity and parallelism. The major difference is that centroids, initialised on a reference view, are shared by all the views with a unique spatial and color coordinate and an attached depth information. Then, the assignment step is simultaneously performed by projecting the centroids onto each view and the update is done by re-projecting all the rays assigned to the super-ray onto the reference view, using a fronto planar assumption. The approach is fast, free of any strong scene geometry prior, easy to understand and to implement, and it gives satisfactory results. Moreover, it does not require a dense depth map estimation, making it a suitable

[1] http://www.lytro.com/imaging

[2] http://www.raytrix.de

[3] http://www.lytro.com/immerge

[4] http://www.iis.fraunhofer.de/en/ff/bsy/tech/lichtfeld.html

[5] http://lightfield.stanford.edu/lfs.html

candidate for a first step of a light field processing pipeline.

A new metric is then introduced to evaluate an important feature of super-rays: the view-consistency of the over-segmentation. We also quantitatively and qualitatively test our approach on synthetic and real light field data-sets having different angular and spatial resolutions, using standard super-pixel metrics. Finally, in [7], [12] the authors observe that an accurate dense depth map is not crucial for editing. As a follow-up of this work, we show that super-rays enable light field processing tasks which are typically done using a dense depth map, such as light field segmentation and correcting the angular aliasing that occurs when refocusing sparsely sampled light fields.

II. Related Work

In this section we discuss the topics related to our work. We first give a brief review of light field editing and then present existing over-segmentation methods for video, multi-view stereo and RGB-D data.

Light field editing: So far, most of light field research has focused on typical light field applications: refocusing [22], depth estimation [23], [24], super-resolution [25], [26]. Light field editing has only been recently addressed with methods targeting either automatic propagation of user inputs from one view to the others [11], or object segmentation.

In the first category, the authors in [11] describe an approach using a 3D voxel-based model of the scene with an associated radiance function to propagate pixel edits and illumination changes in a consistent manner from one view to the other views of the light field. The 3D voxels are then carved away to enforce consistency with an a-priory specified scene radiance model. The voxels whose image projections do not conform well to the local scene radiance model are carved away. Thus, propagating a pixel edit requires determining the voxels that correspond to that pixel and modifying their radiance functions, the change is then propagated by projecting the voxel into each image. The authors in [12] extend the stroke-based 2D image edit propagation method of [27] to light fields. To overcome the computational burden inherent to light field data, the edits are propagated in a downsampled version of the light field. The downsampling step can be seen as some pixel clustering based on an affinity metric defined in the 4D light field space. The pixels are first projected into the affinity space which is then subdivided into clusters, in a way similar to the bounding volume hierarchy. Once the edits have been propagated in the low resolution version of the light field, an upsampling is performed guided by the full resolution data as in the Joint Bilateral Upsampling technique [28]. In [7], the authors present a study on two ways users can interact with light fields. Experiments have been carried out where subjects are asked to perform different typical light field editing tasks, using a ground truth depth map. It is shown that the same tasks can be performed without having a perfect depth map. Using an estimated depth map they observe only a few differences on the capacity of the users to perform the editing. Assuming that light field data can be well approximated by a fixed number of scene layers at different depth, a depth-layer-aware image synthesis method is proposed in [14] for edit propagation.

Perhaps more linked to over-segmentation, the second class of approaches aims at providing object segmentation masks on all views. This can be done using level sets [29]–[31], but this assumes each segmented object to be fronto-planar to the camera. An alternative is to segment each ray using the spatial and angular neighbourhood. In [8], [9], a random forest and SVM technique are used respectively to learn a color and depth model using user scribbles on the central view. A regularisation is then run to obtain the segmentation masks using respectively depth aware and raw angular neighbourhood. To decrease the computational load of the regularisation, it is possible to merge several rays coming from the same scene point in the regularisation [10], but this relies on a relatively accurate dense depth estimation and still scales badly with the spatial light field resolution.

The common point of all these approaches is that they all rely, at some stage, on pixel-level representations of light fields, yielding a high computational load, especially when global regularisation is involved, which motivated the proposed concept of super-rays and can be seen as an extension of super-pixels to light fields.

Super-pixel algorithms: Superpixels have been introduced to circumvent the computational complexity issue in traditional image processing. The term *super-pixel*, first coined in [20] is often described as the partitioning (or clustering) of image pixels into a set of perceptually uniform regions. Ideally super-pixels should be compact (uniform in size), adhere well to the boundaries of objects and be fast to compute.

Because of these properties, super-pixels efficiently represent the image content and are often used as an alternative to pixel representations. Super-pixels allow reducing the computational complexity of many image processing tasks such as object segmentation or object tracking, while providing useful region-based information (e.g texture description or guided regularisation).

Many super-pixel approaches have been proposed and they can be classified into two main categories (see [32] for a recent overview). The first type of methods concerns graph based approaches [33]–[37]. While these methods offer a good accuracy, they either do not provide control on the shape of super-pixels, are very computationally expensive or are not parallelisable, hence not suitable for our applications. The second category of approaches, usually faster than graph-based solutions, aims at growing or evolving existing super-pixels. This category includes a variety of methods such as the multi-scale watershed segmentation approach proposed in [38], the turbopixels segmenting the image into a lattice-like structure of compact regions by dilating seeds [39], and the quick shift clustering technique [40]. In the latter category, one also finds the SLIC [21] and the SEEDS [32] methods for super-pixels on which we focus in the sequel.

Simple Linear Iterative Clustering (SLIC) super-pixels [21] rely on a reformulation of Loyd's algorithm for the k-means problem with two novelties. First, the distance metric is a weighted sum of the CIELab color distance and the Euclidean pixel distance. Second a search window around the centroid is used to reduce the complexity of the algorithm. This approach has been extended in [41] to take into account a geodesic distance between pixels. Color and spatial information are integrated along the shortest path between two pixels in order to guarantee compactness, color consistency and connectivity.

SEEDS super-pixels [32] take quite the opposite approach. The method starts from a regular, coarse grid segmentation, and iteratively updates blocks of pixels at the edge for the current segmentation. That update is done such that each block can change its super-pixel labelling if it decreases a total energy function of the color distribution of the super-pixels. The block size is reduced along the iterations at a given rate.

Despite the fact that the two last approaches provide better results, as they enforce continuity between super-pixels, in SLIC, the computation for the assignment and update steps can be done for each pixel independently. This is not the case for all approaches relying on any *stack formulation*, which makes the methods awkward to implement in parallel. To the best of our knowledge, only SLIC has been successfully implemented on GPUs [42] to provide results in real time.

Multiview segmentation: Since a light field can be seen as a (possibly dense) collection of views, the proposed work raises issues one can also find when dealing with multiview segmentation. One issue is in particular the possibility of simultaneously perform segmentation and depth estimation.

Super-rays imply establishing correspondences between rays corresponding to different views during the segmentation. The correspondences are found with the help of sparse depth information. While it is possible to simultaneously estimate depth and perform the segmentation [43] or the over-segmentation [44], this can understandably be achieved at the cost of a very high computational cost.

An alternative approach is to first compute super-pixels independently for each view, and then find correspondences between them. In [45], a graph is constructed connecting relatively small super-pixels computed on each view separately. The super-pixels form the vertexes of the graph, while the edges connect super-pixels in all the neighbouring images that satisfy the epipolar constraint. To enforce spatial consistency, the weights on edges are given by the color-consistency of two connected super-pixels. Using a foreground and background color model, the authors are able to extract a foreground object directly. The goal being object segmentation, the super-pixels are not explicitly grouped in a consistent manner across views, as targeted here with the proposed super-rays. In [46], super-pixels are computed on one view only, and are assigned a normal and depth measure using photo-consistency with the other views. The authors are more interested in estimating depth information in a reference view rather than by the correspondence established by view segments.

Video over-segmentation: Over-segmentation has also been studied for reducing the complexity of video analysis tasks. Two main categories of approaches exist for video over-segmentation, either considering a set of consecutive frames as a volume or processing each image separately and updating super-pixels as a new frame arrives.

In [47], super-voxels are computed for a set of consecutive frames. The authors assess five super-voxel algorithms in terms of spatio-temporal coherence, object and region boundary detection, region compression and parsimony. Redundancy between frames in the temporal dimension is hence exploited to construct the super-voxels as it is in the inter-view dimension for multiview data. The authors in [48]–[52] instead try to compute temporally consistent super-pixels. The approaches hence consist in updating the super-pixels as each frame arrives, either by deleting, creating or updating super-pixels to account for the scene motion. Equally, optical flow is often used as an additional clustering information, but also when large displacements are involved.

Although applicable to densely sampled light fields, the first kind of approaches is likely to fail in the case of sparsely sampled light fields as they usually fail for videos in the case of large object displacements. The second type of approaches applies to light fields but does not exploit the fact that object displacements from one view to the other is, due to the scene geometry, uniform.

RGB-D clustering: To be complete, one should also mention the work focusing on RGB-D over-segmentation. However, this problem differs from ours in the sense that the goal is to segment a point cloud rather than pixels on several views. Nevertheless, one paper [53] interestingly uses a modified version of SLIC, using seeds defined from the 3D map, and performs the assignment step using the image distance from the centroid projection in order to circumvent the errors in depth estimation. Note that this work differs from ours as we do not assume dense depth information to be available and, in addition, we target view segmentation rather than point cloud segmentation.

III. Super-ray Light Field Over-Segmentation

Let r be a light ray of the light field LF, and (s,t,x,y) its coordinates using the two plane parametrisation [54], where (s,t) and (x,y) are the angular (view) and spatial (pixel) coordinates respectively. Besides, each light ray has an associated $CIELab$ color value Lab_r. In this work, we note $(x',y') := \mathcal{P}^d_{s',t'}(x,y) \in \mathbb{R}^2$ the spatial pixel position in view (s',t') imaging the same scene point, at a distance d, as (x,y) in view (s,t). This is, (x,y) and $\mathcal{P}^d_{s',t'}(x,y)$ are corresponding points imaging the same scene point in different views. In particular, in the case of a uniformly sampled light field we have

$$\mathcal{P}^d_{s',t'}(x,y) = \big(d\,(s-s') + x,\ d\,(t-t') + y\big). \qquad (1)$$

However, if the light field has been acquired with a camera array, $\mathcal{P}$ should take into account the extrinsic and intrinsic matrices of each camera, and allow us to estimate the pixel correspondences in this particular setting. Using this notation, $r \sim r'$ are corresponding rays imaging the same scene point, where $r' := (s',t',\mathcal{P}^d_{s',t'}(x,y))$.

Now, given a light field, our goal is to group in the so-called *super-rays*, all perceptually similar rays corresponding to the same scene area. Formally, we aim to compute the mapping $A\colon LF \subset \mathbb{Z}^4 \to \mathbb{Z}$, such that each light ray r of the light field is assigned with a super-ray label c. We define SR_c the set of rays r such that $A(r) = c$. Each super-ray SR_c is characterised by a centroid ray r_c. By definition, the angular coordinates of r_c correspond to the fixed *reference view* (s_c,t_c). Besides, each centroid ray has a depth d_c associated to it.

A. Method description

Initialisation and depth estimation for centroids: First of all, the spatial positions (x_c, y_c) of the centroid rays are initialized on a regular grid of step S in the reference view. The corresponding $CIELab$ color values on such positions are the initial color values of the centroid rays Lab_{r_c}.

Then, a depth d_c is estimated for each centroid ray r_c. As this step is important for the rest of the algorithm the depth estimation needs to be robust. Thus, inspired by the recent works on light field depth estimation [19], we consider a multi-baseline block-matching strategy with angular patches in order to be more robust to occlusions and fattening errors. Let Ω be the set of angular patches where each patch $o \in \Omega$ is defined such that $o(s,t)$ is 1 if a ray is visible on the view (s,t), and 0 otherwise. Each angular patch can be seen as a visibility mask. In practice, we define Ω as a predefined set of angular patches, one patch that corresponds to the full view visibility and eight patches corresponding to different half view visibilities (top-bottom, right-left and diagonals). See an example for a 3×3 light field in Fig. 1. Hence, the depth for the centroid c is estimated by minimizing the color distance in the RGB color space using the different angular patches

$$d_c = \arg\min_{d} \left\{ \min_{o \in \Omega} \sum_{s',t'} o(s',t')\ \Delta^{B}_{RGB}(r_c, r'_c) \right\}, \quad (2)$$

where $r'_c = (s', t', \mathcal{P}^{d}_{s',t'}(x_c, y_c))$ and

$$\Delta^{B}_{RGB}(r_c, r'_c) = \sum_{(i,j) \in [-B,B]^2} \left(RGB_{r_c}(i,j) - RGB_{r'_c}(i,j) \right)^2, \quad (3)$$

is the patch color distance between the patch in the reference view (s_c, t_c) and the patch in $(s', t') \neq (s_c, t_c)$. In particular, $RGB_{r_c}(i,j)$ is the RGB-color value of the ray $(s_c, t_c, x_c + i, y_c + j)$. In this work, we fix $B = 3$ and we consider 9 angular patches (their size being equal to the number of views in the light field). Since the depth is estimated for a few points (the centroids), this choice is acceptable for low complexity applications.

Assignment step: At each iteration, each light ray $r(s,t,x,y)$ of the light field is assigned a super-ray label. First, the depth estimation in the previous step is used to compute the corresponding rays of r_c. Formally, we compute $r'_c = (s', t', \mathcal{P}^{d_c}_{s',t'}(x_c, y_c))$ such that $r_c \sim r'_c$. Then, each ray in a neighbourhood $N_S(r'_c)$ of size S around r'_c, is assigned to the super-ray SR_c if it minimizes the color and spatial distances:

$$A(r) = \arg\min_{c} \left(\Delta_{Lab}(r, r_c) + m\, \Delta_{xy}(r, r'_c) \right), \quad (4)$$

where

$$\Delta_{Lab}(r, r_c) = ||Lab_r - Lab_{r_c}||^2, \quad (5)$$

$$\Delta_{xy}(r, r'_c) = ||(x,y) - \mathcal{P}^{d_c}_{s',t'}(x_c, y_c)||^2; \quad (6)$$

and m is the parameter weighting the color and spatial distances. A visual explanation can be found in Fig 2. Note that, when r belongs to the reference view, $r_c = r'_c$ in Eq. 4 and our assignment step is equivalent to the SLIC assignment step. However, our approach allows to coherently assign a label to all rays in the other light field views.

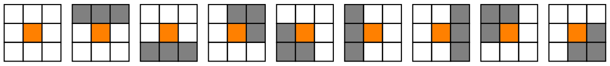


Fig. 1: Example of angular patches in Ω for a light field of 3×3 views. The orange color corresponds to the reference view (s_c, t_c) so the angular patches are equal to 1 at this position. White positions corresponds to visible rays, so its value is equal to 1, and grey positions are equal to 0. The leftmost patch assumes the ray is visible in all views. Other patches correspond to partial visibility.

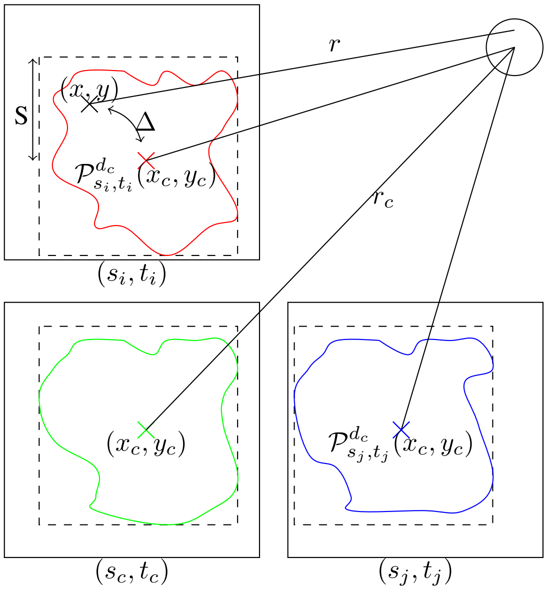

r
(x,y)
S
Δ
$\mathcal{P}^{d_c}_{s_i,t_i}(x_c, y_c)$
r_c
(s_i, t_i)
(x_c, y_c)
$\mathcal{P}^{d_c}_{s_j,t_j}(x_c, y_c)$
(s_c, t_c)
(s_j, t_j)

Fig. 2: Assignment step. r is a ray inside the search window of the super-ray SR_c, defined according to the projection of its centroid r_c, $P^{d_c}_{s',t'}(x_c, y_c)$ in the view where r lies. The color and spatial distances in Eq. 4 is denoted Δ.

In our assignment step we assume (as in [10]) that two light rays from the same view and close spatial coordinates are likely to image two close scene points. Therefore, a ray that is similar in appearance and close to a centroid light ray or close to one of its corresponding rays is considered likely to belong to the same scene object. Therefore, it should belong to the super-ray corresponding to this centroid.

Update step: In this step, the spatial coordinates of the ray centroid and its corresponding Lab values are updated. In particular, the new color value of r_c is the average of the color values of all rays in SR_c and the new spatial coordinates are the average coordinates of all light rays, $r = (s,t,x,y)$ in SR_c projected on the reference view using the depth d_c:

$$Lab_{r_c} = \frac{1}{|SR_c|} \sum_{r \in SR_c} Lab_r\,, \quad (7)$$

$$(x_c, y_c) = \frac{1}{|SR_c|} \sum_{r \in SR_c} \mathcal{P}^{d_c}_{s_c,t_c}(x,y)\,. \quad (8)$$

Note that the centroid rays are defined on a reference view so its angular coordinates (s_c, t_c) are not changed in our algorithm. On the contrary, the centroid spatial coordinates (x_c, y_c) are first initialized on a regular grid in $\mathbb{Z}^2$ and then updated in Eq. 8, which produces new coordinate values in $\mathbb{R}^2$. So, r_c is defined as a virtual light ray which is not necessarily one of the light rays captured in the light field. We summarize the update step in Fig. 3.

When updating the spatial coordinates we assume that rays inside the same super-rays are likely to have similar depth, so Eq. 8 is a good approximation with respect to the centroid position we would obtain using the true depth per ray.

Furthermore, Eq. 8 ensures that two corresponding rays, on two different views, have nearly the same spatial distance Δ_{xy} (as in Eq. 4) from a given centroid ray. This is not necessarily the case when seeding the centroids independently on all the views.

Cleanup step: Similarly to SLIC, our algorithm does not enforce super-ray spatial connectivity, so after our light ray grouping procedure some rays may remain isolated, specially when the spatial term in Eq. 4 has a low weight. For this reason, a simple post-processing is performed, that consists in re-labeling super-ray disconnected components (with a number of pixels $< \frac{1}{4}S^2$) with the closest super-ray label.

The entire algorithm proposed in this paper is described in Algorithm 1.

B. *Experiments*

In order to quantitatively evaluate the proposed approach, well-known super-pixel quality measures can be trivially extended considering all views, such as the Achievable Segmentation Accuracy (ASA), the Boundary Recall (BR) [55] or the Corrected Under-segmentation Error (CUE) [32]. However, these measures do not evaluate the coherence through the light field views. For this reason, we introduce a new evaluation measure called View Consistency (VC). This new measure assumes that the ground truth depth D is known and uses it to select the light rays to consider. Indeed, given a light ray r, our measure aims at evaluating the assignment consistency for the set of corresponding rays imaging the same scene point as r:

$$L'(r) = \Big\{ r' \in LF \;\; s.t. \;\; (x', y') = \mathcal{P}^{D(r)}_{s',t'}(x, y), \\ (x, y) = \mathcal{P}^{D(r')}_{s,t}(x', y'), \;\; (s', t') \neq (s, t) \Big\}. \quad (9)$$

Note that the re-projection check using the ground truth D takes into account the occlusions and guarantees that $L'(r)$ contains light rays imaging the same scene point. Therefore, we define

$$VC(A) = \frac{1}{|LF|} \sum_{r \in LF} \frac{1}{|L'(r)|} \sum_{r' \in L'(r)} \delta\big(A(r), A(r')\big), \quad (10)$$

where δ is the Kronecker delta. This metric is somehow related to the *Inter-Frame Label Consistency* [49] for super-pixel evaluation in the case of videos, but instead of computing the consistency from frame to frame using the ground truth optical-flow, we measure the consistency between all light field views simultaneously using the ground truth depth.

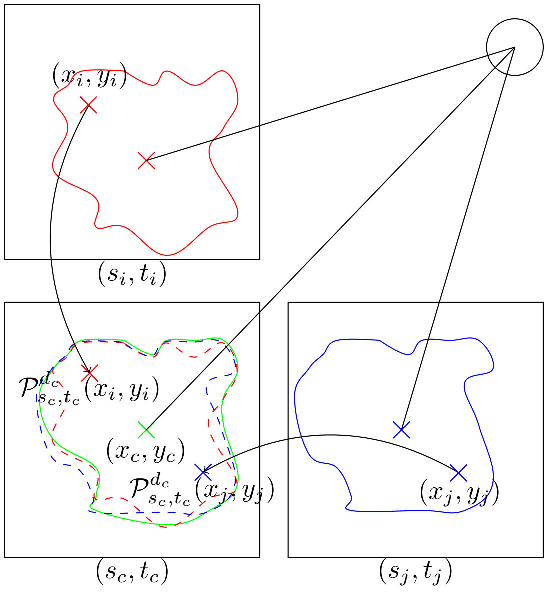

(x_i, y_i)
(s_i, t_i)
$\mathcal{P}^{d_c}_{s_c,t_c}(x_i, y_i)$
(x_c, y_c)
$\mathcal{P}^{d_c}_{s_c,t_c}(x_j, y_j)$
(x_j, y_j)
(s_c, t_c)
(s_j, t_j)

Fig. 3: Update step. Each ray of the super-ray SR_c is reprojected on the reference view using the depth of the super-ray. Here we show r_i and r_j being reprojected on the reference view (s_c, t_c). The projections are averaged, giving the new centroid ray position on the reference view (x_c, y_c).

Algorithm 1: Our super-ray algorithm

```
Data: Input Light Field $LF$
Result: Super-ray assignments $A$
Initialize centroids on reference view;
while not (converged $OR$ max. iteration reached) do
    \\ Assignment step
    for each centroid $c$ do
        for each view $(s', t')$ do
            Compute $(x'_c, y'_c) = \mathcal{P}^{d_c}_{s',t'}(x_c, y_c)$;
            for each ray $r$ in $\mathcal{N}_S(r'_c)$ do
                $A(r) = c$ ($c$ minimizing Eq. 4);
    \\ Update step
    for each centroid $c$ do
        Compute $\mathcal{P}^{d_c}_{s_c,t_c}(x, y), \;\; \forall r \in SR_c$;
        Update $Lab_c$ and $(x_c, y_c)$ (Eqs. 7 & 8);
Cleanup step
```

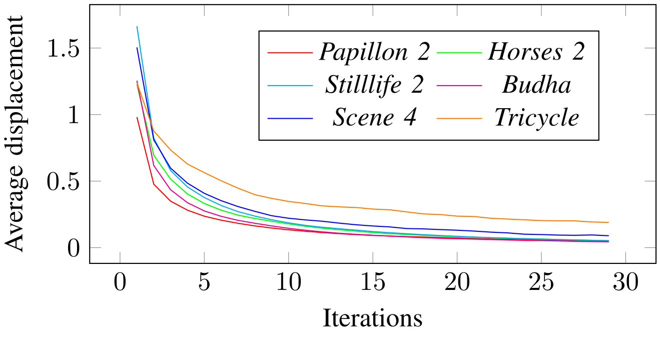
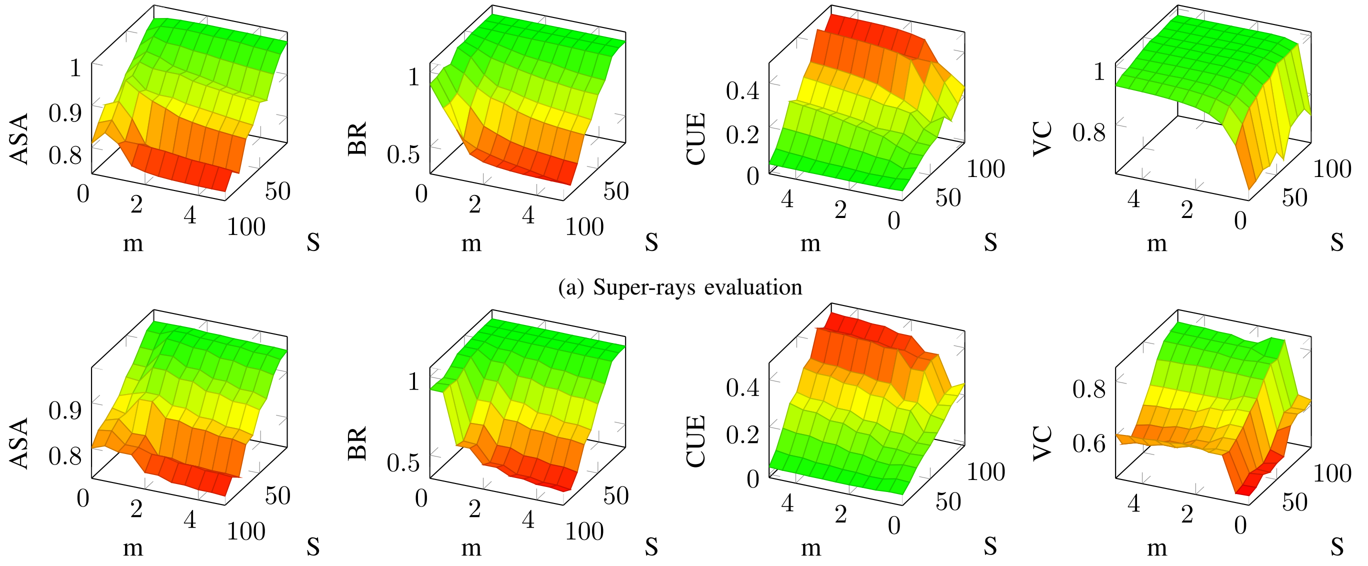


Fig. 4: Average displacement in pixels of the centroid spatial coordinates with respect to the number of iterations.



(a) Super-rays evaluation

(b) Evaluation of independent super-pixel estimation and *a posteriori* merging.

Fig. 5: Different evaluation metrics across different parameters for (*Scene 4*). Green means good score while red represent bad score. For the sake of readability, the axes are flipped differently for each metric.

Our quantitative evaluation is performed on synthetic datasets, with segmentation and depth ground truth. We use the dataset in [24], that is composed of 9×9 densely sampled views of 768×768 pixels (*Papillon 2, Horses 2, Stilllife 2, Budha*). We also use the dataset in [10], which is a 4×4 sparsely sampled light field views of 640×360 pixels (*Scene 4*). Finally, we propose a new dataset of 5×5 views of 640×480 pixels (*Tricycle*), which is rather sparse.

First, we observe that our approach converges in 10-15 iterations, similarly to SLIC, as shown in Fig. 4, for both dense and sparse light fields.

We compare the proposed super-rays construction method with what we would obtain by separately computing super-pixels on each view, and then merging super-pixels having the highest number of corresponding rays across views. In particular, we use the ground truth depth in the synthetic datasets to re-project rays onto the central view, and we then merge the super-pixels of different views with the super-pixels on the central view having the highest number of re-projected rays, i.e such that VC is maximised. Fig. 5 and Fig. 6 show the superiority of our strategy compared to the merging of independent super-rays.

Fig. 5 shows the behaviour of the four quality metrics (ASA, BR, CUE, VC), when varying the different parameters, i.e., the size S of the super-rays and the compactness parameter m, for the dataset *Scene 4*. We observe in Fig. 5a that when increasing the values of S and m, the super-rays do not segment correctly the objects in the scene, as it was observed with SLIC super-pixels. We also remark that ASA, BR and CUE have similar behaviours with similar numerical values when varying S and m, but the proposed metric VC has an opposite behaviour. Indeed, decreasing S and m decreases the view consistency. This can be explained by the fact that decreasing S and m increases the number of super-rays, hence of super-ray edges near which rays are more prone to labelling errors. So, the view consistency decreases. Fig. 5b shows that merging independent super-pixels has no impact on the super-pixel metrics (ASA, BR, CUE) as one may expect, but view consistency is severely deteriorated.

The same observations generalises to the rest of our test set. Fig. 6 shows how the two approaches compare when fixing one of the parameters. To be able to compare light fields of different spatial resolutions, we use k, the number of visible super-rays per view. Once again, we observe very close results when changing k and the spatial weighting parameter m for the ASA, BR and CUE. However, enforcing the super-rays to have the same centroid ray imaging the same scene point, yields super-pixel consistency across views. Our approach allows the segmentation consistency to be independent of the parameters k and m, whereas when computing super-pixels on each view, one super-pixel on a view can be described at two (or more) disjoint pieces of super-pixels on another view, depending mostly on the initial seeding. The other thing we notice is the significant difference in terms of over-segmentation performance between densely sampled and sparsely sampled light fields. The over-segmentation of sparsely sampled light fields is less consistent across views, and usually slightly less accurate than for dense light fields. This can be explained by errors in the initial depth estimation, leading to some inconsistent super-rays.

Fig. 7 and 8 show the super-rays constructed by the proposed algorithm, with $m = 1$ for the smoothness parameter and $S = 15$ and $S = 20$ for the super-pixels size respectively. Note that we only display 3×3 views for the sake of readability. Each super-ray is reassigned a random color, the projection of each centroid ray on the different views is represented with a small cross. At a first glance, the super-rays on each view look like regular super-pixels, but the main

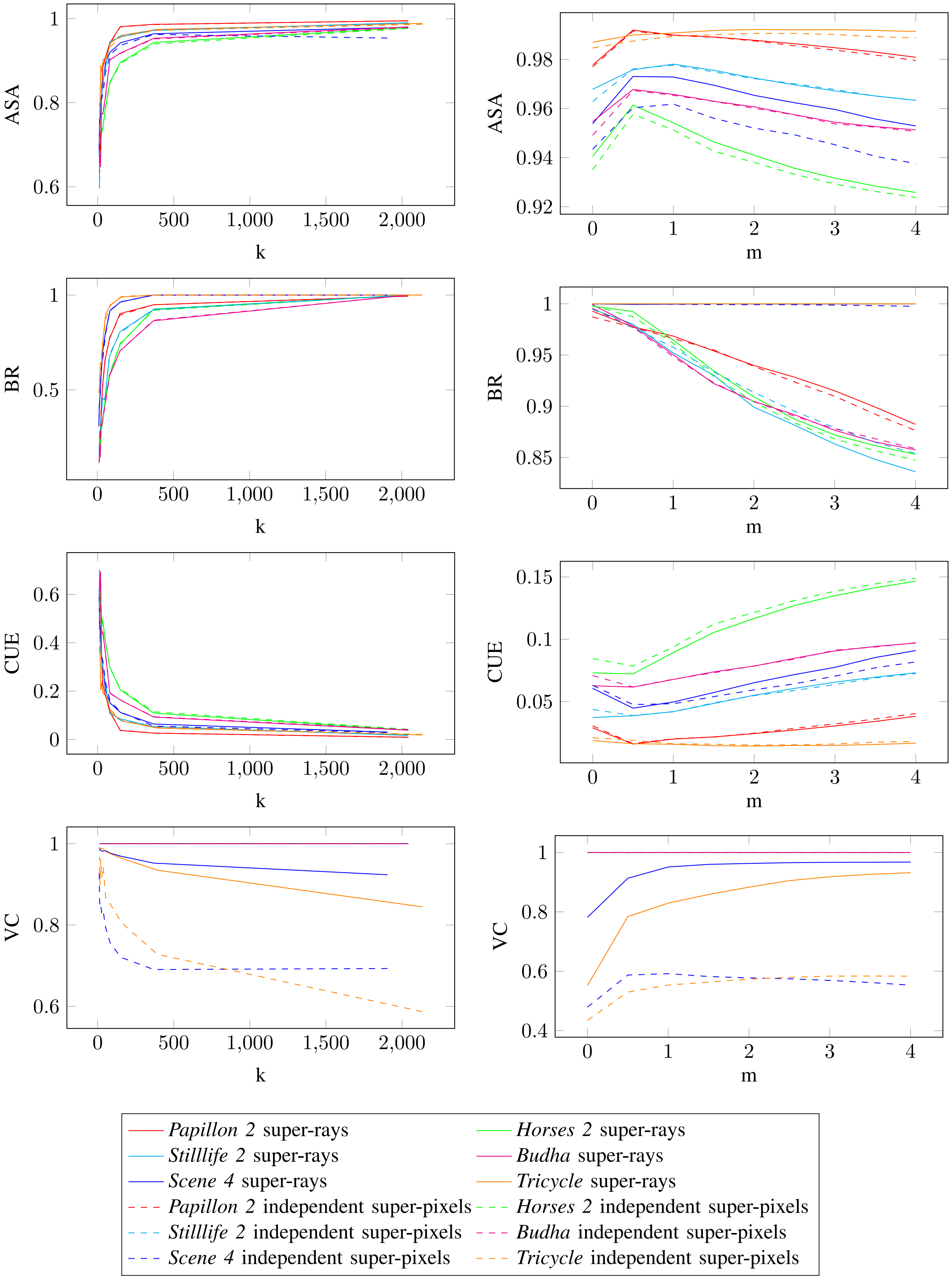
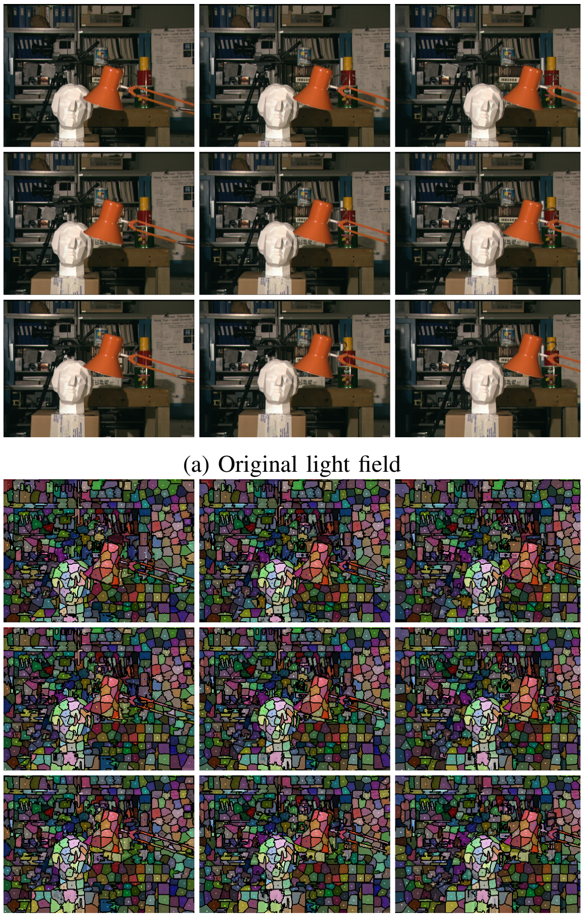


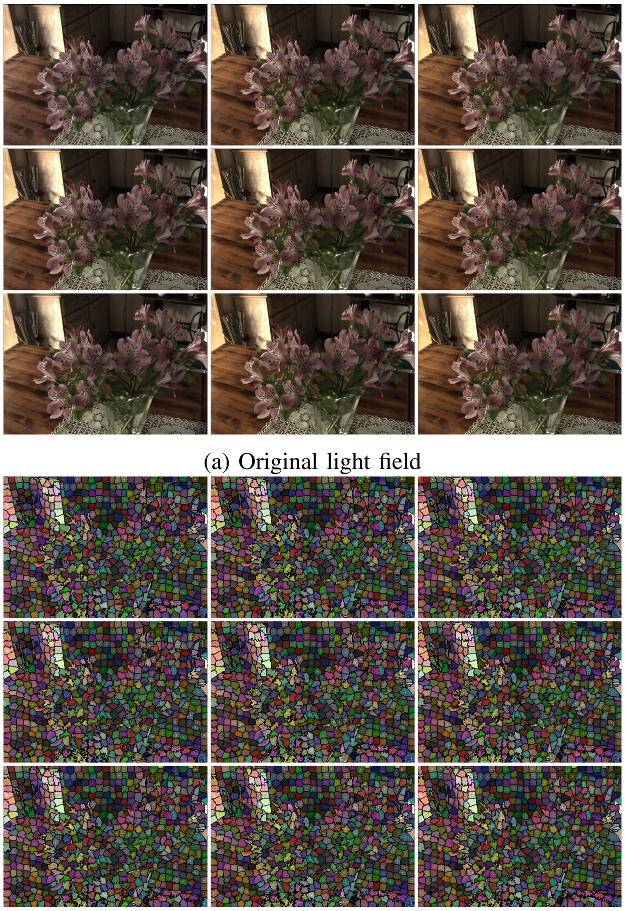
Fig. 6: Comparison of super-rays versus merged independent super-pixels, when varying k, the number of super-rays visible on each view with $m = 1$ fixed (left column) and when varying m the compactness parameter, with $k = 500$ fixed (right column).



(a) Original light field

(b) Color-coded super-rays

Fig. 7: Super-rays for the sparsely sampled light field in the *Tsukuba* dataset [56].



(a) Original light field

(b) Color-coded super-rays

Fig. 8: Super-rays for the densely sampled light field we have captured with the Lytro Illum and decoded using the toolbox in [57], [58].

difference is that super-rays are overall consistent from a view to another, despite occlusions. We invite the reader to zoom in to see the details.

Regarding the running time, we currently have two implementations, one on CPU using C++ and the other one on GPU using Python and Opencl. None of these implementation is optimised (the GPU implementation uses global memory and atomic operations) but still give low run-time on our laptop equipped with an $Intel\ i7-5600U$ and a $Radeon\ R7-M260X$. On the *Tsukuba* dataset, the super-pixels are computed in 6s and 0.3s on GPU. For the dataset of [24], we have a run-time of 80s on CPU and 4.2s on GPU[6]. For Lytro Illum light field, it takes 57s on CPU and 3s on GPU.

We invite the reader to consult our website[7] for more detailed results.

C. Discussion

The main interest of super-rays is to give a compact mutli-view representation of the scene, without relying on dense depth estimation. This representation can be used directly or with depth information associated to each centroid ray, as we show in the next section. One main challenge of super-rays is to be as computationally efficient as super-pixels. While it would be possible to integrate some photo-consistency prior in the assignment step (Eq. 4), this would come with a huge computational cost, since the photo-consistency would either need to be pre-computed for each ray and for each candidate disparity value, or to be computed at each iteration. Moreover, the K-means strategy applied here relies on some strong assumptions on the data (e.g. spherical distribution variance or uniform cluster size), that get easily violated when dealing with other quantities such as color and spatial information. Instead, our approach only uses the disparity of centroid rays, and lets the spatial distance of the reprojected rays do the grouping. In that sense, the geometric information given by the light field is not fully exploited, but on the other hand, as long as two objects have sufficiently different colors, our approach is still sufficient to yield a good segmentation.

The first obvious limitation of this method is that it relies heavily on the centroid depth initialisation. Even if we propose a method to robustify this initial search, errors may have negative consequences on the output segmentation, rays being assigned to the wrong super-ray. Precisely, this is a problem

[6] we used 7x7 views on GPU because the entire light field overflowed the maximum allowed buffer size in our implementation

[7] http://www.irisa.fr/temics/demos/Superrays/index.html

when the disparity error is greater than the super-ray size, as the centroid would potentially fall outside an object during projection. This being said, light-field depth estimation is an active research topic and our depth estimator could be replaced in the future with another sparse and more accurate method.

The second limitation is related to occlusions. Indeed, because the projected centroids coordinates are not placed regularly, but rather warped according to the scene depth, zones with large occlusion have few, or no nearby centroids projections. If an occlusion is bigger than the search window of the super-ray, rays of a view might not be assigned to any super-rays.

IV. Super-ray Applications

In this section we propose two examples of editing applications that exploit the super-rays presented in this paper. On the one hand, we present a fast light field segmentation algorithm based on super-rays. On the other hand, we present a novel algorithm for correcting angular aliasing for sparsely sampled light fields.

A. *Real Time Interactive Segmentation*

Light field segmentation has a high computational complexity [10] [9] [8] and using super-rays is a good strategy for decreasing it, similarly to temporal super-pixels for video segmentation.

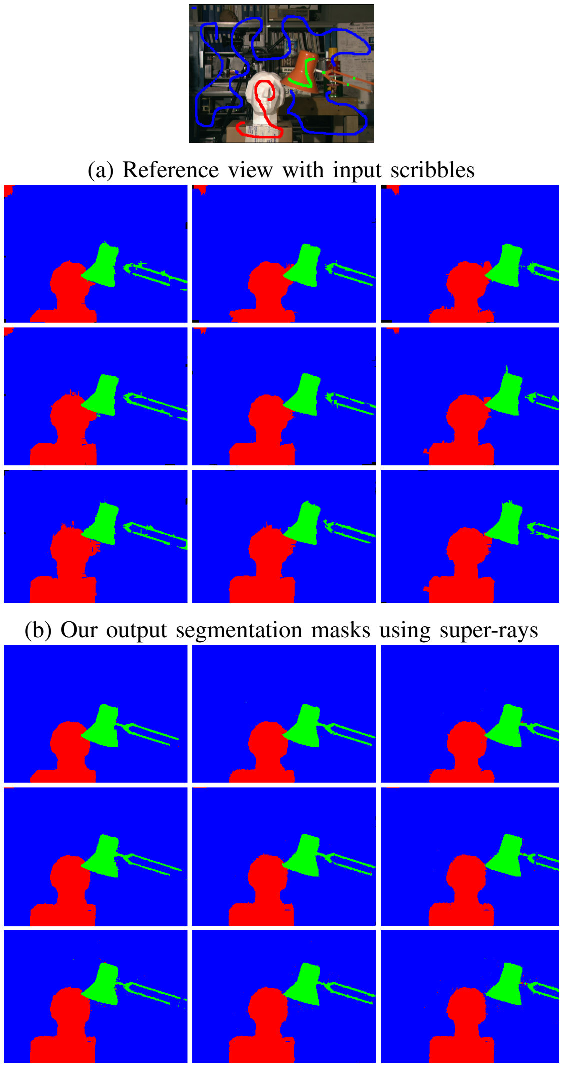
We use the same graph structure introduced in [10] in which rays of the light field are packed in ray bundles in order to decrease the number of nodes of the graph since there is a node per ray bundle instead of a node per light ray. However, it requires a dense depth estimation per view to determine the ray bundles. On the contrary, our proposed strategy uses the super-rays and the centroid depth estimates to create the graph. This is, we build a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where each node in $\mathcal{V}$ corresponds to one super-ray SR_c and we set an edge in $\mathcal{E}$ between two nodes if the two super-rays share a common border in at least one view (adjacent super-rays). Then, we define the energy

$$\varphi = \sum_{c_i} U(c_i) + s \sum_{\substack{c_i \\ c_j \in \mathcal{N}(c_i)}} P(c_i, c_j) \ , \tag{11}$$

where s is the smoothness parameter, $\mathcal{N}$ the function that indicates if two super-rays are adjacent and U and P the unary and pairwise terms respectively. U uses the super-ray average color probability, defined from a Gaussian mixture learnt from the user input and P is defined as a summation of the view conventional pairwise relationship.

In order to minimize Eq. 11 the multi-label Graph-Cuts algorithm in [59], [60] is applied, using an available implementation[8]. Fig. 9 shows the final segmentation for the *Tsukuba* dataset along with the input scribbles in the reference view and the results in [10] for comparison.

The main interest of using super-rays as a base for segmentation is the gain in running time as summarized in Tab.I. For



(a) Reference view with input scribbles

(b) Our output segmentation masks using super-rays

(c) Output segmentation masks in [10]

Fig. 9: Graph-cut segmentation of the dataset *Tsukuba* using our super-rays. Our strategy allows a significant reduction of the running time compared to the state of the art light field segmentation.

Dataset Approach	Tsukuba	[8] dataset
[8] (GPU)	N/S	393.6 to 633.6s
[10] (CPU)	12.2s	156.2s
Ours (CPU)	8.1s	94s
Ours (GPU)	2.4s	14s

TABLE I: Comparison of the total segmentation running times, including depth estimation if any. Results of [8] were computed on a more powerful machine than the others.

the Tsukuba data set, the graph cut applied on super-rays only takes 1 ms on the same machine as in Sec. III-B (a laptop with an $Intel\ i7-5600U$ and a $Radeon\ R7M260X$). The complete segmentation algorithm takes 8.1s on CPU with the following running times for each step: 6s to compute the super-rays, 2.1s to build the graph (i.e to compute the models, unary and pairwise costs), 1ms to do the graph cut. With super-rays computed on GPU, the segmentation only only takes 2.4s with

[8] http://vision.csd.uwo.ca/code/

0.3s for the super-rays computation, 2.1s to build the graph, and 1ms to do the graph cut. On the dataset in [8] the approach takes 94s (80s for the super-rays, 14s to build the graph, 4ms for the graph cut) on CPU and 18.2s (4.2s for the super-rays, 14s to build the graph, 4ms for the graph cut) on GPU.

These running times are significantly reduced compared with the state-of-the-art methods in [8] and [10]. The authors in [8] report a segmentation time between 6 and 10min, including 1 to 5min to learn the models and 5min to perform the optimization on a more powerful desktop GPU ($Nvidia$ $GTX-580$). The approach in [10] which uses depth to reduce the number of nodes in the graph takes 9.3s on the Tsukuba dataset with the same hardware. It takes 6.3s to build the graph and 3s to do the segmentation. On the dataset in [8] and using real depth (in contrast to the ground truth as described in the paper), the approach in [10] takes 122.6s (82s to build the graph, 40s to do the segmentation).

The above figures for the reference methods in [8] and [10] do not take into account the time needed to compute the dense depth map on each view, while our super-ray construction method includes the coarse depth estimation. For a fair comparison, the time needed to estimate the dense depth maps should be added to the above timing reported for [8] and [10] which respectively takes 2.9s and 33.6s for the Tsukuba data set and the data set in [8] using the approach in [61] (implemented as pre-processing on a desktop GPU using C++/OpenCL).

In conclusion, in the context of interactive segmentation, we see that avoiding to compute dense depth maps and running the segmentation on a much more compact light field representation allow a significant reduction of the segmentation running time on CPU, and thanks to the parallelizable nature of super-rays, a dramatic reduction when using GPU. We are confident that more implementation efforts will lead the super-rays computation to be near real-time.

This gain comes at the expense of losing precision, but the obtained accuracy is sufficient for many real-time applications requiring a fast (rather than accurate) segmentation. We illustrate this loss of accuracy in Tab II. We see that super-rays introduce errors with a rate of 0.1% to 2% in the best case scenario, when the algorithm used to assign a label to each super-ray does not do any error. Using the graph structure and the data and pairwise terms in [10], the accuracy drops, with an error rate of 1% to 3%, mostly due to the limited data cost model. Using a more complex model, using textural information provided by super rays will solve this issue.

B. Correcting Angular Aliasing

One of the major new light field applications compared to conventional cameras is the post-capture image refocusing. In fact, light fields captured with Plenoptic type 1.0 cameras provide quite impressive refocusing results [1] but angular aliasing appears when refocusing light fields from camera rigs or refocused plenoptic cameras. This is due to the poor angular sampling of such acquisition systems. Angular aliasing is particularly visible when the simple *shift-and-add* algorithm is used for refocusing, whereas other solutions such as the adaptive splatting [62] or the rendering of novel views [63] [6] decrease the artifacts.

TABLE II: Segmentation accuracy (ratio of pixel assigned to the right labels according to the ground truth) with a synthetic dataset. The parameters were fixed to $k = 2000$ and $m = 1$. We first show our results with respect to the dominant ground truth label assignment for the segmentation (as when computing the ASA metric). This allows us to measure the loss in segmentation accuracy solely introduced by the super-rays (i.e assuming the segmenting process does not introduce additional errors). We also give the results using the super-rays followed by the graph cut approach of [10].

Dataset:	*Still life 2*	*Papillon 2*	*Horses 2*	*Budha*
Result in [8]:	99.3	99.4	99.3	98.6
Result in [10]:	99.2	99.5	99.1	99.1
Our results (GT Segmentation)	99.0	99.5	97.0	98.1
Our results (Graph cut)	98.8	99.1	96.1	96.6

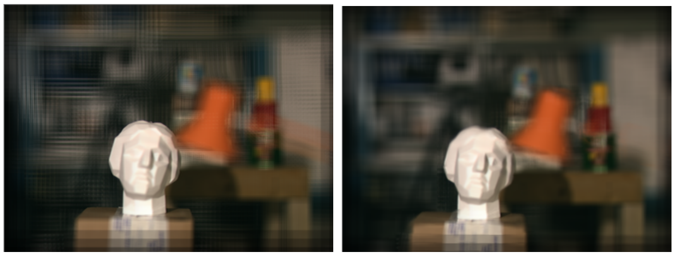


Fig. 10: Refocusing the sparsely sampled *Tsukuba* dataset. Left: Shift-and-add algorithm which creates annoying angular aliasing artifacts. Right: Our strategy using real and virtual views. The virtual views have been rendered with our super-rays. We invite the reader to zoom-in to see the details.

However, these solutions are depth-based methods which makes them unsuitable for fast editing applications. Here we propose to refocus using our super-rays to avoid angular aliasing while maintaining a low complexity. Our goal is not to render new views but our philosophy is similar in the sense that we use approximate intermediate views computed with our super-rays. Thus, refocusing is performed via *shift-and-add* using both original and approximate intermediate views that we call *virtual views*.

With our notations, the *shift-and-add* method to compute the refocused image at depth d from the viewpoint (s_c, t_c) is defined as

$$I^d_{s_c,t_c}(x,y) = \frac{1}{N^2} \sum_{s,t} RGB(s,t,\mathcal{P}^d_{s,t}(x,y))\,, \qquad (12)$$

where the original angular coordinates $s,t = 0,\cdots,N-1$; correspond to the original image views available in the light field. Now, we define the virtual positions $(u,v) \in \mathbb{R}^2$ as

$$(u,v) = (s,t) + \frac{1}{\Delta_d}(m,n)\,, \quad m,n = 1,\cdots,\Delta_d - 1\,, \quad (13)$$

where the number of virtual views between two original views is $\Delta_d = [|(d_c - d)|]$ the integer part of the absolute value of the depth difference between the refocusing depth d and the centroid depth d_c, c being the centroid label to which (s_c, t_c, x, y) belongs. Therefore, the corrected refocused image is defined as the sum of the original images and the virtual views contribution:

$$I'^{d}_{s_c,t_c}(x,y) = \frac{1}{(N \cdot \Delta_d)^2} \sum_{s,t} \Big\{ RGB(s,t,\mathcal{P}^d_{s,t}(x,y)) + \\ + \sum_{u,v} RGB(u,v,\mathcal{P}^d_{u,v}(x,y)) \Big\}. \quad (14)$$

Note that Eq. 14 requires to interpolate the virtual views, which can be quite memory and time consuming. However, in our strategy we do not explicitly compute them but we use the super-rays depth d_c to approximate ray colors in virtual views:

$$RGB(u,v,\mathcal{P}^d_{u,v}(x,y)) \simeq RGB([u],[v],\mathcal{P}^{d_c}_{[u],[v]}(\mathcal{P}^d_{u,v}(x,y))), \quad (15)$$

where $([u],[v])$ is the closest original view.

Considering d_c as the depth of all rays of a super-ray is a coarse approximation but it has few or no consequences on the final result since we only approximate for the blurred, out of focus areas. However, the high frequencies on the blur due to the poor angular sampling are successfully removed. Moreover, using the approximate depth information is sufficient to prevent out of focus rays coming from the background to be mixed with occluding in-focus rays which may create artifacts.

Fig. 10 shows the angular aliasing correction using the dataset from [56] when refocusing at the sculpture bust. Our method successfully decreases the angular aliasing on the background. Note that we simulate a squared aperture which provides a squared *bokeh*.

Finally, we compare our approach with the one presented in [63], which presents a Convolutional Neural Network approach to learn view synthesis for light fields. It provides excellent results, with quite big computational cost both for training but also for generating the views (the authors report $12s$ to generate a single 541x376 view). Their training and test set consist of Lytro Illum light fields. Only the four corner views are kept, giving a sparse light field with a very small number of views, and the 60 views in between these images are generated to extrapolate an 8x8 light field which is compared to the captured Lytro Illum ground truth. We use our approach on the 2x2 corner views to generate the other views the same way. The parameters used were $k = 1000$ and $m = 1$. The computation of the super-rays takes $0.36s$ ($0.17s$ per iteration) and generating a view takes $0.007s$. Then, we compare on Fig. 11 the refocused image obtained with the *shift-and-add* algorithm using the 2x2 views, the synthesized views from [63] and our synthesised views using the super-rays. We also show the refocused ground truth using the 8x8 Lytro Illum real views.

Quantitatively, on the test set, the average SSIM (Structural Similarity) and PSNR (Peak Signal to Noise Ratio) between the ground truth refocused image and our approach is respectively 0.98 and 39.0 versus 0.99 and 42.6 using [63] and 0.92 and 34.5 for the image refocused using only the 2x2 views. Note that we had to crop a 22 pixels border on the reference and our refocused images to comply with the output of [63] that looses the border pixels. Overall, our results are of lower quality because of depth estimates, but in the zones where depth is correct, we see no or few differences between our result and the ground truth. However, our approach has a significantly lighter complexity compared to [63] (the authors report 10s to generate a single view using Matlab and CuDNN, while our approach runs in the order of $\frac{1}{100}s$) which is the main interest of our super-rays.

V. Conclusion

We have introduced the concept of super-ray and we have proposed an implementation that is suitable for sparsely and densely sampled light fields. Our solution is fast, easy to implement, and suitable for GPU implementations. Besides, our super-rays are view consistent, which is a major advantage for light field editing. In particular, we have shown the interest of super-rays for light field segmentation and refocusing without angular aliasing. In the future, the proposed approach could be used for other light field editing tasks, such as intrinsic decomposition [64] or video light field processing.

References

[1] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report*, vol. 2, no. 11, pp. 1–11, 2005.

[2] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *ICCP*. IEEE, 2009, pp. 1–8.

[3] C. Zhang and T. Chen, "A self-reconfigurable camera array," in *SIGGRAPH Sketches*. ACM, 2004, p. 151.

[4] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz, "High-speed videography using a dense camera array," in *CVPR*, vol. 2. IEEE, 2004, pp. II–294.

[5] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar, "Picam: An ultra-thin high performance monolithic camera array," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 166, 2013.

[6] C.-T. Huang, J. Chin, H.-H. Chen, Y.-W. Wang, and L.-G. Chen, "Fast realistic refocusing for sparse light fields," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1176–1180.

[7] A. Jarabo, B. Masia, A. Bousseau, F. Pellacini, and D. Gutierrez, "How do people edit light fields?" *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, vol. 33, no. 4, 2014. [Online]. Available: http://www-sop.inria.fr/reves/Basilic/2014/JMBPG14

[8] S. Wanner, C. Straehle, and B. Goldluecke, "Globally consistent multi-label assignment on the ray space of 4d light fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1011–1018.

[9] H. Mihara, T. Funatomi, K. Tanaka, H. Kubo, H. Nagahara, and Y. Mukaigawa, "4d light-field segmentation with spatial and angular consistencies," in *Proceedings of IEEE International Conference on Computational Photography (ICCP)*, 2016.

[10] M. Hog, N. Sabater, and C. Guillemot, "Light field segmentation using a ray-based graph structure," in *European Conference on Computer Vision*. Springer, 2016, pp. 35–50.

[11] S. M. Seitz and K. N. Kutulakos, "Plenoptic image editing," *International Journal of Computer Vision*, vol. 48, no. 2, pp. 115–129, 2002.

[12] A. Jarabo, B. Masia, and D. Gutierrez, "Efficient propagation of light field edits," *Proceedings of the SIACG*, 2011.

[13] H. Ao, Y. Zhang, A. Jarabo, B. Masia, Y. Liu, D. Gutierrez, and Q. Dai, "Light field editing based on reparameterization," in *Pacific Rim Conference on Multimedia*. Springer, 2015, pp. 601–610.

[14] F.-L. Zhang, J. Wang, E. Shechtman, Z.-Y. Zhou, J.-X. Shi, and S.-M. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *Transactions on Visualization and Computer Graphics*, 2016.

[15] K. W. Shon, I. K. Park *et al.*, "Spatio-angular consistent editing framework for 4d light field images," *Multimedia Tools and Applications*, pp. 1–17, 2016.

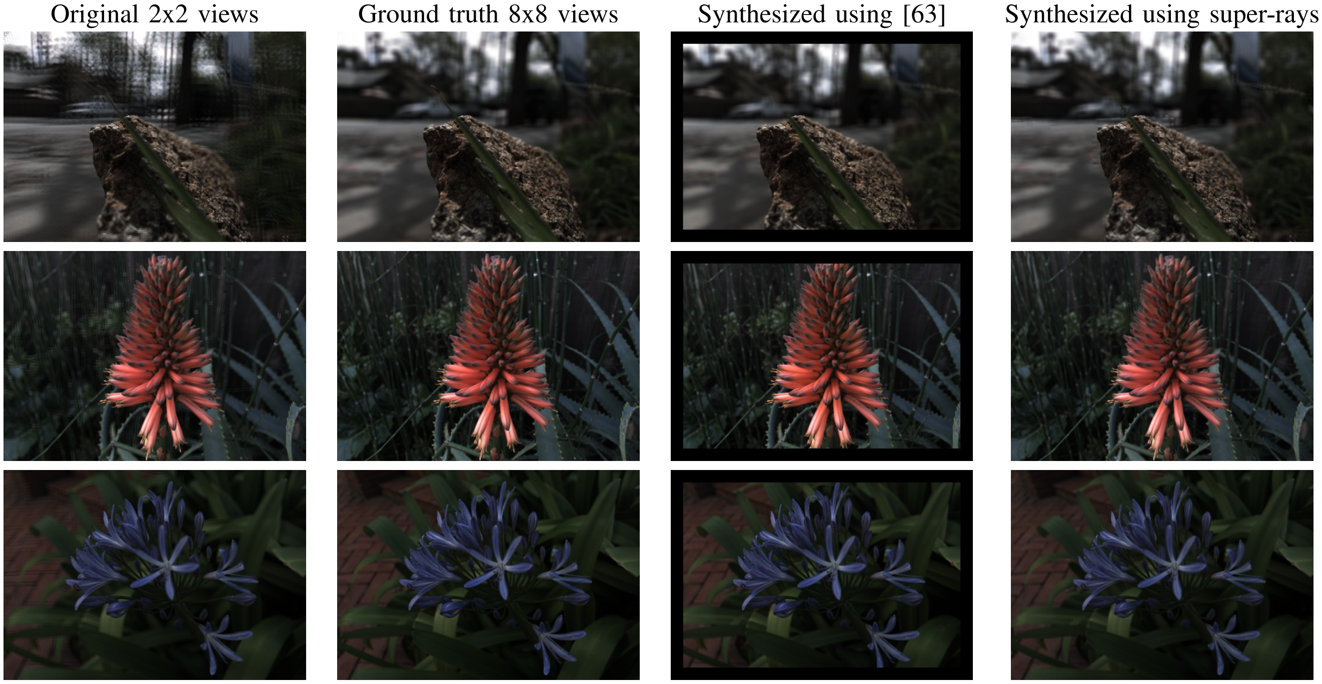


Fig. 11: Angular aliasing compassion. We show the images refocused with the 2x2 corner views in the test set in [63], the full 8x8 light field (ground truth), the 8x8 synthesized light field with the approach in [63] and the 8x8 light field synthesized with our approach. We use $k = 1000$ and $m = 1$.

[16] C. Chen, H. Lin, Z. Yu, S. Bing Kang, and J. Yu, "Light field stereo matching using bilateral statistics of surface cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1518–1525.

[17] H. Lin, C. Chen, S. Bing Kang, and J. Yu, "Depth recovery from light field using focal stack symmetry," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3451–3459.

[18] W. Williem and I. Kyu Park, "Robust light field depth estimation for noisy scene with occlusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4396–4404.

[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*, 2015, pp. 3487–3495.

[20] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV*. IEEE, 2003, pp. 10–17.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[22] R. Ng, "Fourier slice photography," in *TOG*, vol. 24. ACM, 2005, pp. 735–744.

[23] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *ICCV*, December 2013.

[24] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4d light fields," in *CVPR*. IEEE, 2012, pp. 41–48.

[25] T. E. Bishop, S. Zanetti, and P. Favaro, "Light field superresolution," in *ICCP*. IEEE, 2009, pp. 1–9.

[26] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *PAMI*, vol. 36, no. 3, pp. 606–619, 2014.

[27] X. An and F. Pellacini, "Appprop: all-pairs appearance-space edit propagation," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 40.

[28] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 96.

[29] J. Berent and P. L. Dragotti, "Unsupervised extraction of coherent regions for image based rendering." in *BMVC*, 2007, pp. 1–10.

[30] P. L. Dragotti and M. Brookes, "Efficient segmentation and representation of multi-view images," in *SEAS-DTC workshop*. SEAS-DTC workshop, Edinburgh, 2007.

[31] J. Berent and P. L. Dragotti, "Plenoptic manifolds–exploiting structure and coherence in multiview images," *Signal Processing Magazine*, 2007.

[32] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 298–314, 2015.

[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[34] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[35] A. P. Moore, S. J. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[36] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *European conference on Computer vision*. Springer, 2010, pp. 211–224.

[37] Y. Zhang, R. Hartley, J. Mashford, and S. Burn, "Superpixels via pseudo-boolean optimization," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1387–1394.

[38] F. Meyer and P. Maragos, "Multiscale morphological segmentations based on watershed, flooding, and eikonal pde," in *International Conference on Scale-Space Theories in Computer Vision*. Springer, 1999, pp. 351–362.

[39] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[40] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *European Conference on Computer Vision*. Springer, 2008, pp. 705–718.

[41] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha, "Structure-sensitive superpixels via geodesic distance," *International journal of computer vision*, vol. 103, no. 1, pp. 1–21, 2013.

[42] R. Birkus, "Accelerated gslic for superpixel generation used in object segmentation," *Proc. of CESCG*, vol. 15, 2015.

[43] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereojoint stereo matching and object segmentation," in *Computer*

Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 3081–3088.

[44] Y. Taguchi, B. Wilburn, and C. L. Zitnick, "Stereo reconstruction with mixed pixels using adaptive over-segmentation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[45] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Automatic object segmentation from calibrated images," in *Visual Media Production (CVMP), 2011 Conference for*. IEEE, 2011, pp. 126–137.

[46] B. Mičušík and J. Košecká, "Multi-view superpixel stereo in urban environments," *International journal of computer vision*, vol. 89, no. 1, pp. 106–119, 2010.

[47] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1202–1209.

[48] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Spatiotemporal closure," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 369–382.

[49] J. Chang, D. Wei, and J. W. Fisher, "A video representation using temporal superpixels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2051–2058.

[50] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Temporally consistent superpixels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 385–392.

[51] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool, "Online video seeds for temporal window objectness," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 377–384.

[52] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Fast label propagation for real-time superpixels for video content," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 902–906.

[53] J. Yang, Z. Gan, K. Li, and C. Hou, "Graph-based segmentation for rgb-d data using 3-d geometry enhanced superpixels," *IEEE transactions on cybernetics*, vol. 45, no. 5, pp. 927–940, 2015.

[54] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 43–54.

[55] P. Neubert and P. Protzel, "Superpixel benchmark and comparison," in *Proc. Forum Bildverarbeitung*, 2012, pp. 1–12.

[56] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[57] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, Jun 2013.

[58] ——, "Linear volumetric focus for light field cameras," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 2, Feb. 2015.

[59] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

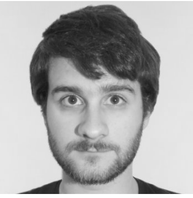
[60] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[61] V. Drazic and N. Sabater, "A precise real-time stereo algorithm," in *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, ser. IVCNZ '12. New York, NY, USA: ACM, 2012, pp. 138–143. [Online]. Available: http://doi.acm.org/10.1145/2425836.2425867

[62] J. Fiss, B. Curless, and R. Szeliski, "Refocusing plenoptic images using depth-adaptive splatting," in *Computational Photography (ICCP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–9.

[63] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, vol. 35, no. 6, 2016.

[64] E. Garces, J. I. Echevarria, W. Zhang, H. Wu, K. Zhou, and D. Gutierrez, "Intrinsic light field images," in *Computer Graphics Forum*. Wiley Online Library, 2017.



Matthieu Hog received the BSc in 2013 from the Université de Saint-Etienne, France and the MSc in 2015 from the Université de Saint-Etienne and the Gøjik University College, Norway. He is currently pursuing his PhD studies in INRIA (Institut National de Recherche en Informatique et en Automatique) and Technicolor Research & Innovation in Rennes, France. His research interests are image processing, computer vision and computational imaging.

Neus Sabater received the BSc degree in 2005 from the Universitat de Barcelona, Spain, and the MSc and PhD degrees in 2006 and 2009, respectively, in image processing from the Ecole Normale Supérieure de Cachan, France. She was a postdoctoral researcher at the California Institute of Technology before being appointed at Technicolor Research & Innovation in 2011 where she is currently a Senior Scientist. Her research interest include image processing, computer vision and computational photography.

Christine Guillemot is currently Director of Research at INRIA (Institut National de Recherche en Informatique et Automatique) in France. She holds a PhD degree from ENST (Ecole Nationale Supérieure des Telecommunications) Paris (1992). From 1985 to 1997, she has been with France Télécom in the areas of image and video compression for multimedia and digital television. From 1990 to mid 1991, she has worked as visiting scientist at Bellcore Bell Communication research) in the USA. Her research interests are signal and image processing, and in particular 2D and 3D image and video coding, joint source and channel coding for video transmission over the Internet and over wireless networks, and distributed source coding. She has served as Associate Editor for IEEE Trans. on Image Processing (from 2000 to 2003), for IEEE Trans. on Circuits and Systems for Video Technology (from 2004 to 2006), and for IEEE Trans. on Signal Processing (2007-2009). She is currently associate editor of the Eurasip journal on image communication (since 2010), for the IEEE Trans. on Image Processing (2014-2016), and for the IEEE journal on selected topics in signal processing (since 2013). She has been a member of the IEEE IMDSP (2002-2007) and IEEE MMSP (2005-2008) technical committees. She is currently a member of the IEEE IVMSP-Image Video Multimedia Signal Processing-technical committee (since 2013). She is the co-inventor of 24 patents, she has co-authored 9 book chapters, 62 international journal publications and around 150 articles in peer-reviewed international conferences. She is IEEE fellow since January 2013.