

# Hyperpixels: Flexible 4D Over-Segmentation for Dense and Sparse Light Fields

Maryam Hamad<sup>1</sup>, *Graduate Student Member, IEEE*, Caroline Conti<sup>2</sup>, *Member, IEEE*,  
Paulo Nunes<sup>1</sup>, *Member, IEEE*, and Luís Ducla Soares<sup>1</sup>, *Senior Member, IEEE*

**Abstract**—4D Light Field (LF) imaging, since it conveys both spatial and angular scene information, can facilitate computer vision tasks and generate immersive experiences for end-users. A key challenge in 4D LF imaging is to flexibly and adaptively represent the included spatio-angular information to facilitate subsequent computer vision applications. Recently, image over-segmentation into homogenous regions with perceptually meaningful information has been exploited to represent 4D LFs. However, existing methods assume densely sampled LFs and do not adequately deal with sparse LFs with large occlusions. Furthermore, the spatio-angular LF cues are not fully exploited in the existing methods. In this paper, the concept of hyperpixels is defined and a flexible, automatic, and adaptive representation for both dense and sparse 4D LFs is proposed. Initially, disparity maps are estimated for all views to enhance over-segmentation accuracy and consistency. Afterwards, a modified weighted *K*-means clustering using robust spatio-angular features is performed in 4D Euclidean space. Experimental results on several dense and sparse 4D LF datasets show competitive and outperforming performance in terms of over-segmentation accuracy, shape regularity and view consistency against state-of-the-art methods.

**Index Terms**—Light field over-segmentation, 4D *K*-means clustering, light field representation, superpixel, supervoxel.

## I. INTRODUCTION

THE required resolution (e.g., spatial, angular and temporal) and degrees of freedom in multimedia applications are growing rapidly. Consequently, the associated computational complexity for processing the data is also increasing significantly. 4D Light Fields (LFs) that capture the same scene from different perspectives are a clear example of what this trend is leading to [1]. To efficiently process the huge amount of data, one possible approach is to reduce the number of data units that need to be processed. This can be achieved by grouping the locally homogenous data units according to

Manuscript received 14 February 2022; revised 31 March 2023; accepted 13 June 2023. Date of publication 5 July 2023; date of current version 11 July 2023. This work was supported by the Fundação para a Ciência e a Tecnologia (FCT)/ Ministério da Ciência, Tecnologia e Ensino Superior (MCTES) through national funds under Project UIDB/50008/2020 and Project PTDC/EEL-COM/7096/2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yulan Guo. (*Corresponding author: Maryam Hamad.*)

The authors are with Instituto de Telecomunicações, Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisbon, Portugal (e-mail: maryam.hamad@lx.it.pt; caroline.conti@lx.it.pt; paulo.nunes@lx.it.pt; lds@lx.it.pt).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TIP.2023.3290523>, provided by the authors.

Digital Object Identifier 10.1109/TIP.2023.3290523

some criteria into larger ones. This approach is known as “image over-segmentation”. A recent trend in computer vision is to process 2D images and 3D volumes at a higher-level representation instead of at the pixel-level representation [2]. As an example, image over-segmentation can be used as a pre-processing step in image compression [3], [4], object tracking [5], object segmentation [6], [7], 3D semantic segmentation [8] and saliency detection [9]. Considering that image over-segmentation can be applied to 2D images and 3D volumes to facilitate subsequent applications, applying a similar approach to 4D LFs would also make sense.

4D LFs indirectly describe the distribution of light rays in free space by capturing the same scene from several points of view [1], [10]. Depending on the LF capturing approach, dense or sparse 4D LFs can be generated [1]. In dense LFs, most of the objects exist in all LF views and, therefore, LF processing or editing can be done on only a single LF view, or a small subset of LF views, and then propagated into all other LF views using, for example, LF view warping. In sparse LFs, however, such possibility is limited by largely occluded regions or regions that only appear in some LF views due to the viewing angle. To handle these specific issues of sparse LFs, all objects that appear in any LF view must be considered, and an adequate propagation method must be used to ensure accurate and angularly consistent LF processing or editing. In both cases, due to the existing similarities within LF views, LF over-segmentation can be exploited to group data units within and across LF views. Therefore, a significant reduction in the number of data units to be processed can be achieved to facilitate subsequent tasks [7], [11], [12], [13]. 4D LF over-segmentation should aim at not only spatial accuracy (i.e., adhering well to object boundaries and separating regions correctly), but also angular consistency (i.e., segmented regions not changing abruptly when the viewpoint changes). Currently, only a few methods for 4D LF over-segmentation are available in the literature. These methods can be classified as being either clustering-based methods [11], [14], [15], [16], [17] or graph-based methods [18], depending on the used approach. The clustering-based approach is adopted in this paper, since it is widely used due to the superior results in terms of accuracy and also due to the reduced computational complexity and memory usage, when compared to graph-based ones [2], [19]. Although the available methods that tackled 4D LF over-segmentation challenges have significantly improved over-segmentation angular consistency (compared to simply applying a 2D method to each view

independently), remaining limitations still need to be further investigated.

Firstly, existing methods consider dense LFs (i.e., captured with narrow baselines between views) and do not adequately deal with sparse LFs with large occlusions (i.e., captured with large baselines between views). For example, one reference view (e.g., the central view) or the structure of the central Epipolar Plane Image (EPI) (i.e., the unique 2D spatio-angular slice of the LF typically containing a regular structure with several oriented lines [20]) is used to perform 2D over-segmentation. After that, the obtained segments are propagated to other LF views. For this, it is assumed that each 2D segment in the central view should have a corresponding one in all other LF views (i.e., “full-sliced” property). This assumption, however, may not always hold, notably for sparse LFs. In the sparse LF case, some objects may not exist in all LF views, either because they are occluded in some LF views by foreground objects or because they fall outside the viewing angle of those views.

Secondly, the spatio-angular LF cues, including depth or disparity information (i.e., the displacement of a point between different views, which is inversely proportional to the depth), and 4D spatio-angular coordinates are not fully exploited in most existing methods. The used disparity information in some existing methods is either estimated for some pixels only (e.g., the clustering centroids) or for all pixels in one reference view only (e.g., the central view) [11], [15]. Moreover, disparity information in some methods is used to enforce a view consistent projection for the clustering centroids, but not as a discriminative feature to guide the over-segmentation (for instance, when color information is insufficient to separate different regions [17]). Additionally, all available clustering-based methods are still not 4D in nature, meaning that the clustering is applied using 2D Euclidean space without considering the angular dimensions, and the centroids are fixed in one angular location. Lastly, none of the previous methods (except in [17]) support adaptive clustering.

In this paper, a novel clustering-based 4D LF over-segmentation method that tackles these limitations is proposed. The contribution of this paper is four-fold:

- **The definition of 4D hyperpixels for dense and sparse LFs**– The “hyperpixels” definition is provided to have an entity that adequately reflects the high dimensional nature of the basic element of 4D LF over-segmentation, supporting flexible clustering/grouping criteria for both dense and sparse LFs. The provided definition extends the existing definitions in [11] and [15] as detailed in Section III.
- **Flexible, adaptive and consistent 4D over-segmentation method for dense and sparse LFs**– In this paper, LF over-segmentation is applied using a modified  $K$ -means clustering in the 4D hypercubic domain that is adapted to LF content and fully exploits the spatio-angular cues. As such, it is the only over-segmentation method for LFs that is truly 4D in nature. The differences between the proposed hyperpixel over-segmentation method and other methods is detailed in Section IV. Experimental results, including dynamic

results in the supplemental materials, show superior performance when compared to existing methods.

- **A 4D LF dataset of sparse LFs with a large absolute disparity range**– To validate our proposed method for sparse 4D LFs quantitatively, a dataset of 4D LFs including non-Lambertian objects and complex texture regions that mimic real images is generated. This is the first sparse 4D LFs dataset that includes ground truth segmentation label images, disparity, and depth maps for all LF views. It is publicly available and can be used to qualitatively and quantitatively evaluate 4D LFs for several LF applications.
- **Labeling-LF Angular Consistency (LLFAC) metric**– Existing LF view consistency metrics discard the large occlusions in off-central views when projected into the central view and, hence, may not fairly evaluate the view consistency in sparse LFs. In this paper, we highlight the importance of having metrics for sparse LFs that can consider local angular consistency. Therefore, we adapted the recently proposed metric that is applied for LF style transfer applications [21] to evaluate labeling LF angular consistency for dense and sparse LFs.

The remainder of the paper is organized as follows. Section II briefly reviews the related work on LF over-segmentation. Section III introduces the concept of hyperpixels in 4D space and explains the differences with respect to previous definitions. Section IV describes the proposed method in detail, while in Section V its performance is evaluated through a series of experiments. Finally, Section VI concludes the paper with final remarks and proposes directions for future work.

## II. RELATED WORK

Image over-segmentation aims at providing a more meaningful representation of an image and can reduce the number of processing data units. Ren and Malik [22] first defined a group of locally coherent pixels that share the same visual properties as “superpixels”. Subsequently, this concept has inspired many researchers to propose various 2D over-segmentation methods, of which a comprehensive review can be found in [2]. More recently, deep learning was exploited in 2D image over-segmentation, leading to a further improvement in accuracy [23], [24]. However, applying 2D over-segmentation methods to each LF view independently will not ensure LF angular consistency, which is crucial for many applications. The superpixel concept has also been extended to consider 3D volumes [25], videos [26] and higher-dimensional visual data, such as 4D LFs, where over-segmentation angular consistency is particularly important.

In this section, the few available 4D LF over-segmentation methods are briefly reviewed. Current 4D LF over-segmentation methods can be classified as clustering-based or graph-based, depending on the approach used to divide 4D LFs into homogeneous regions.

### A. Clustering-Based 4D LF Over-Segmentation

In this class, 4D LFs are divided into a certain number of homogeneous clusters of pixels with similar sizes using the

$K$ -means clustering technique. Currently, all available methods in this category start the clustering process by initializing the centroids only in the central view of the LF.

Hog et al. [11] proposed a fast method that groups light rays of similar color in an LF into what they defined as “superrays” using 2D  $K$ -means clustering. The angular consistency is enforced by projecting the superrays in the central view into all other views and vice versa, using the disparity values of their centroids. Notice that the disparity values are estimated only for the centroids in the central view in the initial position of the centroids to apply the projection step and are not included as a clustering feature. Therefore, a cleaning step is needed to correct wrongly labeled or unlabeled pixels due to inaccurate projection or clustering, especially in largely occluded regions. Later, the authors extended their work to handle LF videos by also considering the temporal dimension [27].

Zhu et al. [15] defined the concept of 4D LF SuperPixel (LFSP) and a metric for evaluating LFSP angular consistency (i.e., the self-similarity metric). The method proposed in [15] to generate LFSPs relies on segmenting the central view firstly with a 2D  $K$ -means clustering algorithm, assisted by the disparity feature only for the central view. After that, superpixels are projected to other views using the centroids disparity values. Finally, an optimization stage is needed to ensure the EPI space regularity. In this work, the “full-sliced” property is assumed, which can represent a significant limitation for sparse LFs.

Khan et al. [16] proposed a novel View-Consistent Light Field Superpixel (VCLFS) segmentation. Initially, the over-segmentation is applied in the EPI space for the central horizontal and central vertical EPIs independently, by considering that each pair of lines defines a 2D segment. After that, a 2D  $K$ -means clustering is applied after combining the horizontal and vertical EPIs into the central view. Labels are then propagated to all off-central LF views using per-pixel disparity. Although the disparity for all views is used during the clustering, relying on EPI regularity can limit the VCLFS method performance for sparse LFs (e.g., due to their irregular EPI structure).

Recently, Hamad et al. [17] proposed an adaptive LF Over-segmentation (ALFO) method based on modified 2D  $K$ -means clustering. In the ALFO method, the weights applied to the different features for clustering are adjusted adaptively based on the image content. Hence, the balance between regularity, compactness, and angular consistency is improved. In this method, per-pixel disparity is required as input and exploited during the clustering. Although ALFO has shown outperforming performance, it still does not fully exploit the spatio-angular cues, this fact will be further discussed in Section IV-F. Moreover, as in the previous methods, only the central view is used to initialize the centroids, which is not adequate for sparse LFs and largely occluded regions.

### B. Graph-Based 4D LF Over-Segmentation

In this class, LF over-segmentation is considered as a graph-partitioning task. More precisely, an undirected graph is created from a 4D LF by considering every single pixel

in a 4D LF as a graph node. Afterwards, according to the edge weights between adjacent nodes, the graph is cut into sub-graphs with each sub-graph representing a 4D segment. Generally, applying graph optimization on a huge number of pixels, may require a long execution time and extensive consumption of resources.

Li et al. [18] proposed a Hierarchical and View-invariant LF Segmentation (HVLFS) method. By creating a weighted undirected 4D graph from a 4D LF, the over-segmentation is achieved by maximizing the graph entropy in the 4D LF domain. In this method, different features are used to guide the over-segmentation, such as color, depth and texture. The entropy rate for the over-segmentation is measured in the EPI space to ensure angular consistency. This method generates subgraphs with different sizes according to the user input. However, some limitations remain regarding the need for proper normalization of the used weights for optimization to robustly fit different LF datasets. Moreover, since angular consistency is handled by tracking the EPI structure, the method has been shown to fail when applied to sparse 4D LFs [18].

## III. HYPERPIXELS DEFINITION

A pixel (short for “picture element”) is the fundamental unit in 2D images. Similarly, the fundamental unit of 3D volumes is called a voxel (short for “volume element”). Given the fact that these low-level representations do not necessarily have a perceptual meaning [22], a more compact and natural representation is desired. Therefore, locally coherent pixels/voxels in 2D/3D space can be grouped into superpixels/supervoxels [25], respectively, according to some criteria. The main objective is to provide a more meaningful representation and to reduce the number of processing data units. Recently, a froxel was defined to describe an element of a frustum-aligned voxel grid, by using depth and camera-setup-dependent discretization of the view frustum [28].

For 4D LFs, the concepts of superray [11] and LFSP [15] were proposed. These concepts, however, still have some limitations that prevent them from being truly analogous to the superpixel and supervoxel ideas but extended for 4D LFs.

In this paper, we try to overcome such limitations by introducing the concept of “hyperpixel”, simply defined as “a group of similar pixels in the discrete 4D LF space”. The criteria used to define what are similar pixels will depend on the specifics of the over-segmentation method adopted. The differences with respect to superrays and LFSP are described as follows.

The authors in [11] defined superrays as “groups of rays of similar color coming from the same scene area”. This definition implies a representation in the continuous 3D scene space, although the authors used it interchangeably to refer to its corresponding projection in the discrete 4D LF space  $(x, y, u, v)$ . Moreover, in this definition, the authors impose the following constraint on the grouping of rays: the rays in each superray must have a similar color. The goal of our proposal is to have an entity defined purely in the discrete 4D LF space without imposing any constraint on the similarity criteria used for grouping. With the proposed definition of hyperpixels,





Fig. 1. Examples of regions only visible in some views. The fire extinguisher is occluded by the blue car in view (5, 9). The blue car is not visible in view (5, 1) because it is outside the viewing angle of this view. This scene is one of the generated sparse 4D LFs in our dataset.

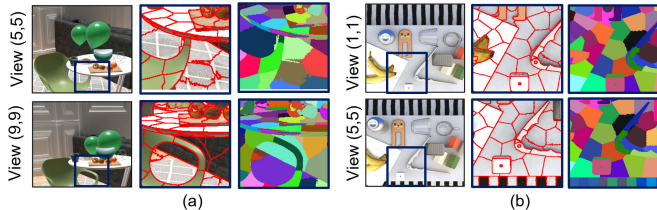


Fig. 2. Visualization of non-existent or occluded regions in the central view, i.e., view (5, 5), that are visible in other LF views and vice versa. a) The part of the sofa that can be seen through the hole of the chair armrest in view (9, 9) is occluded in view (5, 5); b) The bottom part of the black and white carpet appears in view (5, 5) but is not visible in view (1, 1) because it falls outside the viewing angle of this view. These scenes are from our sparse 4D LF dataset.

pixel grouping can be performed using a variety of pixel features (e.g., texture, depth, 4D spatial-angular coordinates, etc.). Obviously, the pixel grouping can still be performed using only the color feature, as is the case of superrays. The choice of grouping criteria to be used depends on the specifics of the over-segmentation method adopted.

According to [15], “*LFSP is a light ray set which contains all rays emitted from a proximate, similar and continuous surface in 3D space*”. This definition also implies a representation in the continuous 3D scene space, although the authors of [15] also used it interchangeably to refer to its corresponding projection in the discrete 4D LF space  $(x, y, u, v)$ . Moreover, in this definition, the authors impose the following constraint on LFSPs: “*there are 2D slices of LFSP in all views of light field in free space (i.e., without occlusion)*”. On the other hand, hyperpixels are not required to have 2D slices in all LF views, even for objects in free space (i.e., without occlusion). This is particularly important when considering sparse LFs, where it is possible that some objects in free space are only visible in some views and large occlusions can exist (see Fig. 1). Obviously, our definition would also support the case in which a given object in free space is visible in all LF views; in that situation, a 2D slice would exist in all views, as in LFSP. In Fig. 2, an example is shown of how hyperpixels can have slices in some views and not be present in other views if no corresponding pixels exist in those views.

To sum up, we consider that the hyperpixel concept reflects adequately the high dimensional nature of the basic element of 4D LF over-segmentation and it is sufficiently generic and flexible to comprise the 4D projections of both existing superrays and LFSPs definitions.

#### IV. PROPOSED 4D LIGHT FIELD OVER-SEGMENTATION

This paper proposes a flexible, adaptive, and view-consistent 4D over-segmentation method for dense and sparse static

TABLE I  
MAIN NOTATIONS USED IN THIS PAPER

Symbol	Definition
$I(x, y, u, v)$	A static 4D light field with $x, y$ spatial coordinates and $u, v$ angular coordinates
$K$	Number of hyperpixels
$H_{size}$	Grid step size (a.k.a., hyperpixel size)
$\Omega_i$	Searching window centered at centroid $c$ , where $i \in \{1, \dots, K\}$
$ A $	Cardinality of set $A$
$\mathbf{p}$	A pixel in 4D space with $(x_p, y_p, u_p, v_p)$ coordinates
$\mathbf{c}$	A centroid in 4D space with $(x_c, y_c, u_c, v_c)$ coordinates before being projected into other views
$\mathbf{c}'$	A projected centroid in 4D space into $(u', v')$ view
$d_{hor,p}^{(u,v) \rightarrow (u',v')}$ , $d_{ver,p}^{(u,v) \rightarrow (u',v')}$	Horizontal and vertical disparities, respectively, of pixel $\mathbf{p}$ from view $(u, v)$ to view $(u', v')$
$H_i$	A hyperpixel represented by a centroid with index $i$
$D_f(\mathbf{p}, \mathbf{c})$	Distance between pixel $\mathbf{p}$ and centroid $\mathbf{c}$ according to feature $f$ , $f \in \{p, l, a, b, d\}$
$WV_f$	Within-cluster variance of feature $f$
$w_f$	Clustering feature weight of feature $f$
$D_{GT}$	Ground truth disparity maps
$L_{GT}$	Ground truth segmentation label images

LFs. According to the hyperpixel definition, our proposed LF over-segmentation method aims at grouping similar pixels in 4D space into hyperpixels. For grouping, several features are considered (i.e., 4D position, color and disparity values). To achieve that,  $K$ -means clustering is applied in 4D space. In summary, given a 4D LF scene, disparity maps for all LF views and the hyperpixel size, the proposed method undergoes four main steps (see Fig. 3), where each step is detailed in the following subsections:

1. Initial clustering centroids (i.e., the hyperpixel center of mass in 4D space) are first selected by considering the central view and largely occluded regions from other views. Each centroid is characterized by several features.
2.  $K$ -means clustering is applied in 4D LF space and all pixels are labeled iteratively to minimize the within-hyperpixel variance.
3. Centroids color, 4D position and disparity features are adjusted at each iteration during the clustering.
4. Clustering weights are adapted after each iteration.

Steps 2, 3 and 4 are repeated until convergence is reached.

In this paper, we assume a regular arrangement of cameras with a parallel optical axis and uniform camera baseline and focal length. However, the proposed method can also be extended and applied to other camera arrangements by adjusting the used equations accordingly. The main notations used in this paper are listed in TABLE I.

##### A. Occlusion-Aware Centroids Initialization

The first step in the proposed hyperpixels over-segmentation method is to select initial centroids to guide the 4D clustering process. Different from other available clustering-based LF over-segmentation methods, where the centroids are initialized in a pre-defined reference view (e.g., the central view), the proposed method enables occlusion-aware centroids initialization. Initializing centroids only in the central view



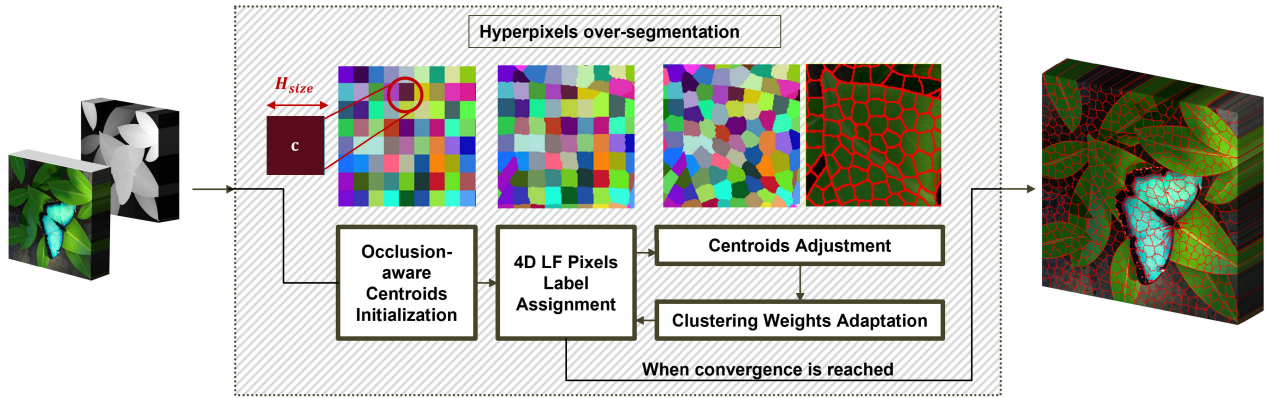


Fig. 3. The main steps of the proposed 4D LF over-segmentation method. Given a 4D LF and the corresponding disparity maps for all views, initial centroids characterized by distinct features are assigned in the reference view/views. Next, hyperpixels are generated by iteratively applying 4D K-means clustering, including pixel labeling, centroids adjustment and clustering weights adaptation, until convergence is reached.

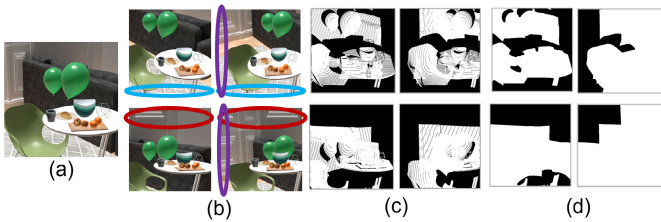


Fig. 4. Example of occluded regions in sparsely sampled LFs. a) The central view; b) 4 reference corner views; c) Occluded regions (black regions) in each view; d) Visibility masks for corner views after redundancy removal. Only the central view and the black regions in the corner views as in (d) will be used to initialize unique centroids to consider the largely occluded regions.

may generate inaccurate over-segmentation for occluded or non-existent regions in the reference view due to different view perspectives; this can be critical due to largely occluded regions in sparse LFs. Therefore, to handle this problem, the four corner views are considered along with the central view for centroid initialization. These extreme corner views are selected since they typically contain all LF information.

To detect the small color differences, before initializing the centroids, the input LF views are converted to the CIELAB color space, which is widely used for image segmentation since it mimics human visual perception. To avoid biased clustering, the LF views and the disparity maps are normalized according to the min-max normalization method as in [17]. Given the normalized inputs, the centroids are initially distributed in the central view over a uniform 2D square grid with step size,  $H_{size}$  (a.k.a. hyperpixel size). Afterwards, to detect the occluded or non-existent regions in the central view that are visible in any corner view, the central view is warped to the corner views by using its disparity map. All the occluded regions in each corner view are represented by a binary visibility mask where the occluded regions are assigned the value 0 (black pixels in Fig. 4c).

To avoid redundancy, when initializing new centroids in the corner views, the regions that represent the same occluded 3D points in more than one corner view are kept only in one corner view and discarded from others (see, for example, the ovals with similar color in Fig. 4b). To achieve that, each corner

view is iteratively warped into other corner views using its disparity map. Afterwards, pixels in the current corner view that overlap with the projected pixels from other corner views are kept only in the visibility mask of the current corner view and discarded from the visibility masks of other corner views. Moreover, the connected pixels (with 8-direction connectivity) in the occluded regions that are smaller than  $H_{size}$ , are also discarded. Finally, new centroids are initialized uniformly only in the remaining regions in the corner views that do not have corresponding centroids in the central view as applied earlier to the central view. After initializing the centroids in the central and corner views, that represent the hyperpixels, each pixel in 4D space will be clustered to the appropriate hyperpixel as explained in the next step.

### B. 4D LF Pixels Label Assignment

In this step, each pixel in the 4D LF is labeled and assigned to the corresponding hyperpixel based on the similarity in the used clustering features. To exploit LF cues during the clustering, each pixel is characterized by a feature vector  $[x, y, u, v, l, a, b, d]$  according to its position in the 4D space, where  $(x, y)$  are the spatial coordinates,  $(u, v)$  are the angular coordinates,  $(l, a, b)$  are the color components in the CIELAB color space, and  $d$  is the disparity value. To assign labels for all pixels in 4D LF, a modified version of the  $K$ -means clustering algorithm is used by considering an adaptive weighted clustering in 4D space.

In 4D LFs, considering cameras with a parallel optical axis, the scene is captured from different angular perspectives hence, views with spatial shifts are generated. These shifts lead to the appearance of slanted lines in the EPI space, as can be seen in Fig. 5 where the EPI slices with yellow and red borders are generated by first stacking the central horizontal and vertical LF views, respectively. Different from voxels in 3D space, the corresponding pixels that represent the same 3D point in 4D space have a spatial shift across views, horizontally and vertically, according to the disparity of each object in the scene.

Therefore, to support truly 4D clustering, the centroids are projected into each LF view to enforce the cross-view

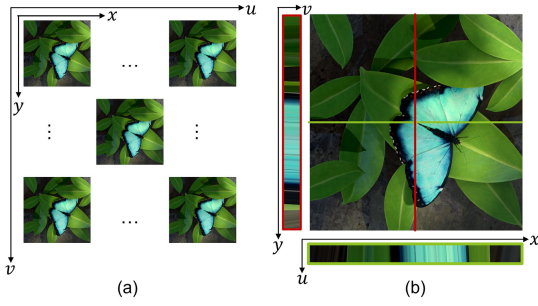


Fig. 5. In 4D LFs, each LF view (i.e., a slice of 4D LF in a particular angular plane  $(u, v)$ ) captures the scene from a different view perspective, resulting in shifted light rays across views as can be seen in the yellow and red bordered EPIs shown below and to the left of the central view.

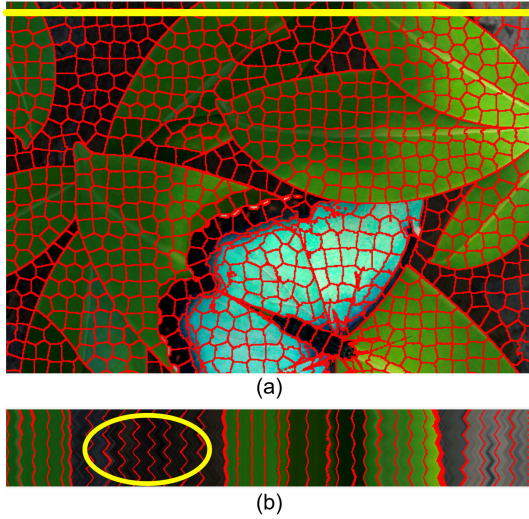


Fig. 6. To ensure consistency with respect to the EPI slanted nature, centroids are projected spatially during the 4D clustering. a) 2D view overlaid with hyperpixel borders; b) A stack of horizontal EPIs when projecting the centroids into each view.

consistency according to the slanted nature of the EPIs as in Fig. 6. Notice that the EPIs in Fig. 6 are generated by stacking the 4D LF views in serpentine order (to maintain connectivity in the EPI lines for better visualization), resulting in 2D horizontal EPI slices. Due to the differences in sampling the angular and spatial dimensions (especially for sparse LFs), a sampling compensation is needed. This can be achieved here by shifting the LF views using their disparity maps during the clustering to make the corresponding pixels aligned as described below.

More precisely, the 4D K-means clustering is applied in each view by spatially projecting the centroids, using their disparities, from their current angular position into each view without changing their angular dimensions, as in (1):

$$\begin{aligned} x_{\mathbf{c}'} &= x_{\mathbf{c}}^{(u', v')} = x_{\mathbf{c}}^{(u, v)} + d_{hor, \mathbf{c}}^{(u, v) \rightarrow (u', v')}, \\ y_{\mathbf{c}'} &= y_{\mathbf{c}}^{(u', v')} = y_{\mathbf{c}}^{(u, v)} + d_{ver, \mathbf{c}}^{(u, v) \rightarrow (u', v')}, \end{aligned} \quad (1)$$

where  $(x_{\mathbf{c}'}, y_{\mathbf{c}'})$  are the spatial coordinates of the projected centroid,  $\mathbf{c}'$ , using the disparity of the centroid located in  $(u, v)$  view, and  $d_{hor, \mathbf{c}}^{(u, v) \rightarrow (u', v')}$  and  $d_{ver, \mathbf{c}}^{(u, v) \rightarrow (u', v')}$  are, respectively, the horizontal and vertical disparities from  $(u, v)$  view to  $(u', v')$

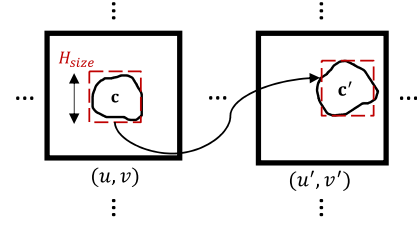


Fig. 7. Example of spatial projection of a hyperpixel centroid from view  $(u, v)$  into view  $(u', v')$  using the horizontal and vertical disparity values.

view. Considering that the used disparity estimation methods for densely and sparsely sampled 4D LFs generate per-pixel disparities from each view to its right horizontal adjacent view [11], [15], [17], the disparity value is here computed as in (2):

$$\begin{aligned} d_{hor, \mathbf{c}}^{(u, v) \rightarrow (u', v')} &= d_{\mathbf{c}} \times (u' - u), \\ d_{ver, \mathbf{c}}^{(u, v) \rightarrow (u', v')} &= d_{\mathbf{c}} \times (v' - v), \end{aligned} \quad (2)$$

where  $d_{\mathbf{c}}$  is the disparity of the centroid,  $\mathbf{c}$ , from each view to its right horizontal adjacent view and  $(u, v)$  are the angular coordinates where the centroid is located. Notice that in (2) a uniformly sampled camera setup is considered. However, if the camera baselines are different for horizontal and vertical directions, then a consideration of camera parameters is needed [15]. When centroids are projected into other views, their spatial position  $(x_{\mathbf{c}'}, y_{\mathbf{c}'})$  may belong to  $\mathbb{R}^2$ , however, color and disparity values in the used datasets are only available for integer positions. Therefore, the coordinates of the projected centroids are rounded to ensure integer indexing belonging to  $\mathbb{Z}^2$ . More precisely, for projection, unnormalized position and disparity values are used. However, during 4D clustering and weights adaptation steps, the normalized unrounded values are used.

Due to the high dimensionality of 4D LFs and since most hyperpixels usually have a local slice in each view, the searching of the nearest centroid is applied, as proposed for 2D images [29], in a small searching window,  $\Omega_i$ , around each centroid in each view as defined in (3):

$$\Omega_i = (4 \times H_{size})^2, \quad (3)$$

where  $i \in \{1, \dots, K\}$ ,  $H_{size}$  is the hyperpixel size as in Fig. 7.

Let  $H = \{H_1, \dots, H_K\}$  represent the set of all hyperpixels where  $K$  is the number of hyperpixels. This way, the over-segmentation can be considered as an energy minimization problem in (4):

$$E = \arg \min_H \sum_{i=1}^K \sum_{\mathbf{p} \in H_i} D_w(\mathbf{p}, \mathbf{c}_i), \quad (4)$$

where  $\mathbf{p}$  is a pixel in 4D space that belongs to hyperpixel  $H_i$ ,  $D_w$  is the weighted distance, and  $\mathbf{c}_i$  is the centroid of  $H_i$  in 4D space. In this step, each pixel in  $\Omega_i$  is assigned to the “nearest” hyperpixel based on,  $D_w$ , as in (5)-(10):

$$\begin{aligned} D_w(\mathbf{p}, \mathbf{c}) &= w_p \times D_p^2 + w_l \times D_l^2 \\ &\quad + w_a \times D_a^2 + w_b \times D_b^2 + w_d \times D_d, \end{aligned} \quad (5)$$

where  $w_p$  is the position clustering weight,  $w_l$ ,  $w_a$ ,  $w_b$  are the color clustering weights,  $w_d$  is the disparity clustering weight and  $D_p$ ,  $D_l$ ,  $D_a$ ,  $D_b$ ,  $D_d$  are the position, color and disparity distances between each pixel  $\mathbf{p}$  and a target centroid  $\mathbf{c}$ , respectively,  $D_d$  here is not squared to impose a larger penalty on the disparity feature as in [17]. The distances in this paper are computed as follows:

$$D_p(\mathbf{p}, \mathbf{c}) = \sqrt{\frac{(x_{\mathbf{p}} - x_{\mathbf{c}'})^2 + (y_{\mathbf{p}} - y_{\mathbf{c}'})^2 + (u_{\mathbf{p}} - u_{\mathbf{c}})^2 + (v_{\mathbf{p}} - v_{\mathbf{c}})^2}{8 \times H_{size}^2 + (N_u - 1)^2 + (N_v - 1)^2}}, \quad (6)$$

$$D_l(\mathbf{p}, \mathbf{c}) = \sqrt{(l_{\mathbf{p}} - l_{\mathbf{c}})^2}, \quad (7)$$

$$D_a(\mathbf{p}, \mathbf{c}) = \sqrt{(a_{\mathbf{p}} - a_{\mathbf{c}})^2}, \quad (8)$$

$$D_b(\mathbf{p}, \mathbf{c}) = \sqrt{(b_{\mathbf{p}} - b_{\mathbf{c}})^2}, \quad (9)$$

$$D_d(\mathbf{p}, \mathbf{c}) = \sqrt{(d_{\mathbf{p}} - d_{\mathbf{c}})^2}, \quad (10)$$

where  $\mathbf{p}$  represents each pixel in 4D space that belongs to the searching window centered on centroid  $\mathbf{c}$ . Furthermore,  $x_{\mathbf{c}'}$ ,  $y_{\mathbf{c}'}$  are the spatial coordinates of centroid  $\mathbf{c}$  when projected into the view of  $\mathbf{p}$  with angular coordinates  $(u_{\mathbf{p}}, v_{\mathbf{p}})$ . Additionally,  $(u_{\mathbf{c}}, v_{\mathbf{c}})$  is the original angular coordinate of centroid  $\mathbf{c}$  without projection and  $N_u$ ,  $N_v$  are the horizontal and vertical angular dimensions, respectively. The projected spatial position is used here to enforce cross-view consistency by considering the disparity between views and to compensate for the difference in sampling spatial and angular dimensions. To normalize the position feature,  $D_p$  is divided by  $(8 \times H_{size}^2 + (N_u - 1)^2 + (N_v - 1)^2)$ , by considering the minimum distance to be zero and  $\sqrt{8 \times H_{size}^2 + (N_u - 1)^2 + (N_v - 1)^2}$  is the maximum distance in 4D space. In the first iteration, all the weights are initialized with the same value, equal to  $1/|W|$ , where  $W$  is the set of clustering weights  $\{w_p, w_l, w_a, w_b, w_d\}$  and  $|W|$  is the number of the used clustering weights. As shown in [17], the values of the initial weights do not significantly impact the final clustering weights. Notice that the used weights must be in the  $(0, 1)$  range, and  $\sum w_{f \in \{p, l, a, b, d\}} = 1$ , in each iteration.

After assigning labels to all the pixels in 4D LFs, centroids are adjusted in terms of their features according to the current iteration as described in the next step.

### C. Centroids Adjustment

In this step, the clustering features vector of each centroid  $\mathbf{c}$  is adjusted iteratively until convergence is reached. After each iteration, the color feature values,  $l_{\mathbf{c}}$ ,  $a_{\mathbf{c}}$ ,  $b_{\mathbf{c}}$ , and the 4D position features,  $x_{\mathbf{c}}$ ,  $y_{\mathbf{c}}$ ,  $u_{\mathbf{c}}$ ,  $v_{\mathbf{c}}$ , of each centroid are adjusted by the mean values of all pixels that belong to the corresponding hyperpixel,  $H_i$ , where  $i \in \{1, \dots, K\}$  as (11):

$$t_{\mathbf{c}} = \frac{1}{|H_i|} \sum_{\mathbf{p} \in H_i} t_{\mathbf{p}}, \quad (11)$$

where  $t_{\mathbf{p}}$  is the feature value of a pixel,  $\mathbf{p}$ , in 4D space, and  $t \in \{x, y, u, v, l, a, b\}$ . Notice that, different than the

existing LF over-segmentation methods, the proposed method also adjusts the angular coordinates. This is useful especially for the objects that exist only in some LF views and are occluded (partially or completely) or non-existent in other views.

Finally, to ensure robust centroid projection in the next iteration, and similar to [17], the disparity value of each centroid,  $d_{\mathbf{c}}$ , is updated using the actual disparity value of the centroid updated 4D position (rounded to integer positions) from the input disparity maps,  $d$ , as in (12):

$$d_{\mathbf{c}} = d(x_{\mathbf{c}}, y_{\mathbf{c}}, u_{\mathbf{c}}, v_{\mathbf{c}}). \quad (12)$$

After adjusting the centroids, the clustering weights still need to be adapted according to the current iteration; to avoid biased or non-optimal over-segmentation as explained in the next step.

### D. Clustering Weights Adaptation

As the last step in each iteration and after the centroids are adjusted, the clustering weights are adapted according to the LF content and the current iteration. This step is beneficial especially when the features differ in their ranges. Moreover, selecting certain fixed values for clustering weights that suit different datasets without considering their content is a challenging, time-consuming task and may generate non-optimal over-segmentations. Since the use of adaptive weights has been shown to improve over-segmentation performance in [17] and [30], a similar technique is exploited here.

As in [30], the feature discriminability principle states that the features with the smaller within-cluster variances (i.e., the total sum of the feature distances from each pixel to its centroid in all hyperpixels) are more discriminative. Hence, it is beneficial to assign a larger weight to these features to properly influence the over-segmentation. The discriminability of each clustering feature can be computed by finding the normalized within-cluster variance for each feature,  $f$ , as in (13):

$$WV_f = \sum_{i=1}^K \sum_{\mathbf{p} \in H_i} D_f(\mathbf{p}, \mathbf{c}_i)^2, \quad (13)$$

where  $K$  is the number of hyperpixels,  $\mathbf{p}$  is a pixel in 4D space that belongs to hyperpixel  $H_i$ ,  $\mathbf{c}_i$  is the centroid of  $H_i$  in 4D space,  $D_f$  is the feature distance from each pixel,  $\mathbf{p}$ , and the centroid,  $\mathbf{c}_i$ , and  $f \in \{p, l, a, b, d\}$ . Unlike the technique in [30], but similar to [17], in this paper, the input 4D LF image and disparity maps are normalized before clustering. Therefore, we did not divide  $WV_f$  by the feature ranges, which is needed in [30] to normalize  $WV_f$ . After computing  $WV_f$  for each feature, the clustering weights are updated by assigning higher weight values to the features with smaller  $WV_f$  values using (14):

$$w_f = \frac{1}{\sum_{j \in \{p, l, a, b, d\}} (WV_j / WV_j)^{\frac{1}{|W|-1}}}, \quad (14)$$

where  $j$  represents each clustering feature and  $|W|$  is the number of the used clustering weights.



### E. Convergence Criterion

After applying the above steps, the iterative 4D clustering will continue until convergence or the maximum number of iterations is reached. To check for convergence, after each iteration, the average displacement of all centroids,  $D_{avg}$ , is computed by finding the 4D Euclidean distance between the previous centroid position in 4D space and the current 4D position. In this paper, we set the maximum number of iterations to 20 as will be discussed in the following section. Additionally, to improve the performance (in terms of the needed number of iterations), we considered a convergence threshold for  $D_{avg}$  of 0.7% of  $H_{size}$  (this value has been determined empirically after exhaustive testing). By choosing this threshold, we noticed, especially in dense 4D LFs, that the over-segmentation can converge before reaching the maximum number of iterations without a significant difference in accuracy.

### F. 4D Versus 2D K-Means

In this section, the differences between the proposed 4D  $K$ -means clustering method and the 2D  $K$ -means clustering used in most of the available 4D LF over-segmentation clustering-based methods are briefly explained.

In the proposed method the centroids are initialized, before clustering, in the central view and in occluded regions in off-central views, as explained in Section IV-A. Other methods initialize centroids only in the central view, e.g., [11] and [17].

Besides the color feature, in the proposed method the 4D pixel position and disparity features are also considered during the clustering for all LF views. Other methods, either do not use disparity information as a clustering feature but merely for enforcing consistent centroids projection [11], or do not exploit the angular dimensions during the clustering [11], [15], [16], [17].

During the clustering, the centroids positions can be adjusted not only spatially but also angularly. In all other available methods [11], [15], [16], [17] the centroids are fixed angularly. Moreover, in the proposed method, disparity values are adjusted from the input disparity maps for each centroid after updating its 4D position. However, in most available methods, centroid disparity values are either never adjusted even when a centroid changes its position [11], or are adjusted to the mean disparity value of all pixels in the LF segment [15], [16].



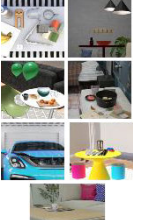
The proposed energy minimization function considers clustering weights for each feature to either penalize or increase its importance, with the weights being adapted to the LF content, similar to ALFO [17], which does not happen in other methods that rely on fixed values for clustering weights.

Consequently, the proposed method is truly 4D in nature and the creation of hyperpixels is based on grouping similar pixels in the 4D LF space. All other LF over-segmentation methods rely on projecting 2D superpixels in the center view to other LF views and then applying a final optimization.

## V. EXPERIMENTAL RESULTS

To evaluate the proposed 4D LF over-segmentation method, from here on simply called hyperpixels method, in various

TABLE II  
IMAGE DATASETS USED IN THE EXPERIMENTAL RESULTS

4D LF dataset	View resolution ( $N_x \times N_y$ ) pixels	Number of views ( $N_u \times N_v$ )	Thumbnails
HCI dataset [31]: Buddha, Papillon, Horses, and StillLife	768×768 except for Horses: 1024×576	9×9	
MMSPG dataset [32]: Sphynx, Bikes, and Sophie	625×434	15×15	
Our generated dataset for sparse LFs: Kitchen, Room, Balloons, Antique, Car, Chess and Leisure	512×512 except for Leisure: 1280×720	9×9	

aspects, both dense and sparse, synthetic and real world LF datasets are used. Additionally, to validate the results, qualitative and quantitative comparisons with state-of-the-art methods are presented. In the following sub-sections, the used 4D LF datasets, benchmark methods to compare with and the used evaluation metrics are detailed. To clearly notice cross-view consistency, we highly encourage the reader to see the extended results on entire LFs in the supplemental materials for dynamic visualizations available online (please note that not all LF views are presented in this paper but can be found in the supplemental materials).<sup>1</sup>

### A. Used 4D LF Datasets and Experimental Setup

In this paper, three different datasets are used to generate hyperpixels for densely and sparsely sampled LFs as shown in Table II. In the case of dense LFs, the synthetic HCI 4D LF dataset [31], which contains Ground Truth (GT) disparity maps and 4D LF segmentation labels, is used. Moreover, only the central  $11 \times 11$  views of the real world EPFL MMSPG dataset captured with a Lytro Illum camera [32] are used to eliminate the vignetting effects in corner LF views (i.e., darkening of the edges of the captured microimages).

For sparse 4D LFs, there is currently no available 4D LF sparse dataset with GT segmentation labels, GT disparity and depth maps for all LF views, which are needed for quantitative evaluation. For this reason, by using Blender software with Cycles rendering [33], LF Blender tools proposed by Honauer et al. [34], and some publicly available 3D models in [35], [36], and [37], we generated a new synthetic dataset accompanied by GT disparity maps, depth maps and segmentation labels, in order to enable the numerical evaluation. Our dataset has disparity values between adjacent views within the range  $[-125, 125]$  and consists of 11 4D

<sup>1</sup>Higher quality versions at <https://github.com/MaryamHamad/Hyperpixels>

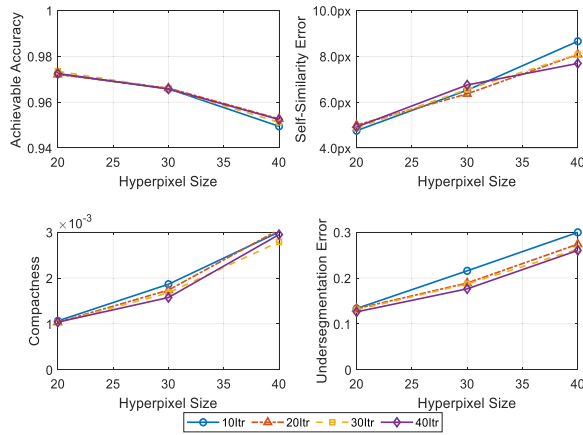


Fig. 8. Average quantitative evaluation of used test 4D LFs with different hyperpixel sizes and number of iterations.

LFs with  $(9 \times 9)$  angular resolution and either  $(512 \times 512)$  or  $(1280 \times 720)$  spatial resolution. Our dataset contains several objects and challenging regions for segmentation, for example, non-Lambertian objects (e.g., glass and metal), complex textures, uneven lighting and overlapping objects with similar colors. As such, it can be used to evaluate various LF applications (the IT-4DLF dataset is available for download at: <http://www.img.lx.it.pt/IT-4DLF/>). In this paper, 7 challenging sparse 4D LFs and 7 dense 4D LFs from other commonly used datasets are used.

It is worth noting that our proposed method relies neither on any experimentally set clustering weights nor on any post-processing step. Most existing methods require cleaning or optimization as a post-processing step to fill unlabeled pixels due to inaccurate over-segmentation or to regularize the over-segmentation results across views. Like in existing clustering-based LF over-segmentation methods, the hyperpixel size is assigned by the user according to the desired application. It was observed that using adaptive 4D clustering enhances over-segmentation convergence [17]. The proposed hyperpixels method converges most of the time within 10 iterations. However, the maximum number of iterations was chosen to be 20 to ensure accurate labeling even for complex scenes. This value was selected after comparing the average performance for the used test images generated after 10, 20, 30 and 40 iterations. Since there was no significant improvement in the performance after 20 iterations, as shown in Fig. 8, this value was chosen as a convergence criterion. Our implementation is not optimized yet, but it has been shown in the literature [2], [10], [16] that  $K$ -means clustering can be parallelized for fast over-segmentation, which may be required for some applications.

### B. Benchmark Methods and Experimental Parameters

In this paper, we compared our results with all the existing 4D LF over-segmentation methods listed in Section II namely: the Superray [11]; LFSP [15]; VCLFS [16]; HVLFS [18]; and ALFO [17] methods. The used software for these methods was obtained and used as detailed in [17]. To generate the superrays in [11], numerous parameters are required as input, such as

the disparity range between adjacent views, and compactness weight (e.g., a weight that controls superrays compactness). The disparity range is obtained from the estimated disparity in [38] and [39] (as used for our method), for each test image independently and the compactness weight is set to 10, as it shows superior performance in [29], for different superrays sizes. As input to the LFSP method [14], [15], different methods are used by the authors of the LFSP method for estimating only the central disparity map without significantly affecting the performance, such as [40] and [41]. In this paper, the input disparity map of the central view that is used for the LFSP method is the same as the one used for our hyperpixels method for dense and sparse LFs. For the VCLFS [16], the maximum disparity parameter is merely changed according to each LF and this value is set using the same disparity maps that are used for our method. For the HVLFS method [18], we only have results provided by the author for dense synthetic LFs and superpixel size belonging to [20, 45]; hence, we could not compare this method with sparse LFs or compute its execution time. For ALFO method [17], disparity maps for all 4D LF views are required as input. Therefore, the used disparity maps for our method are also used for ALFO method.

Regarding the input hyperpixel size (a.k.a. cluster/segment size),  $H_{size}$ , for dense and sparse LFs, several sizes were tested on the HCI and our generated datasets (i.e., 20, 25, 30, 35, 40). For the MMSPG dataset, since there is no labeling GT available, only  $H_{size} = 20$  is presented.

### C. Evaluation Metrics

To generate the quantitative results, the evaluation metrics comprehensively described for 4D LF in [17] are used. Namely, the Achievable Accuracy (AA), Boundary Recall (BR), Under-segmentation Error (UE), Compactness (CP), Self-Similarity error (SS), and number of Labels per Pixel (LP). Notice that the existing consistency metrics used in [17] do not adequately consider regions that exist in other views but are occluded or non-existent in the central view, especially in sparse LFs. To overcome this limitation, the recently proposed LF Angular Consistency (LFAC) metric for style transfer applications [21] is adapted and modified to compute the consistency of sparse LF over-segmentation more accurately. Different from LFAC [21], where the consistency of RGB stylized LFs (i.e., composed LF in the style of another image) is compared with an original one and where the estimated disparity of the original image is used, in this paper, a labeled 4D LF is used to compute the angular consistency assisted with the GT disparity maps and segmentation label images for all LF views.

**Labeling-LF Angular Consistency (LLFAC)**– Given a GT 4D LF disparity map,  $D_{GT}$ , and GT segmentation label images,  $L_{GT}$ , the angular consistency is computed by initially grouping the hyperpixels into object-level using  $L_{GT}$ . To achieve that, each hyperpixel in the hyperpixel labeled image,  $L$ , is assigned to the label of the segment in  $L_{GT}$  that has the largest overlap with the current hyperpixel. Afterwards, the local angular variance map,  $\sigma^2(L)$ , is initially computed

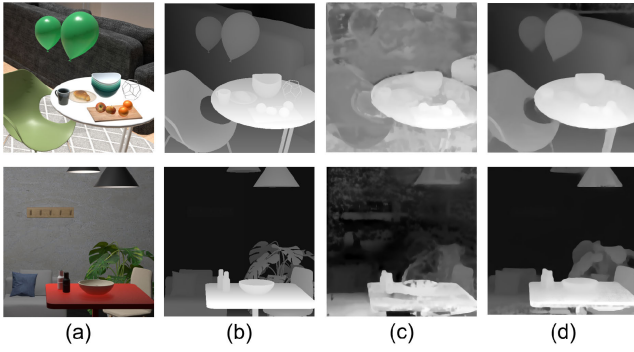


Fig. 9. Estimated disparity for sparse LFs: a) The central LF view for which the disparity estimated; b) GT disparity with range  $[-35.3, 8.7]$ ; c) Results by using the deep learning based method in [39]; d) Results by applying our proposed modification on [39] to improve the accuracy and angular local consistency.

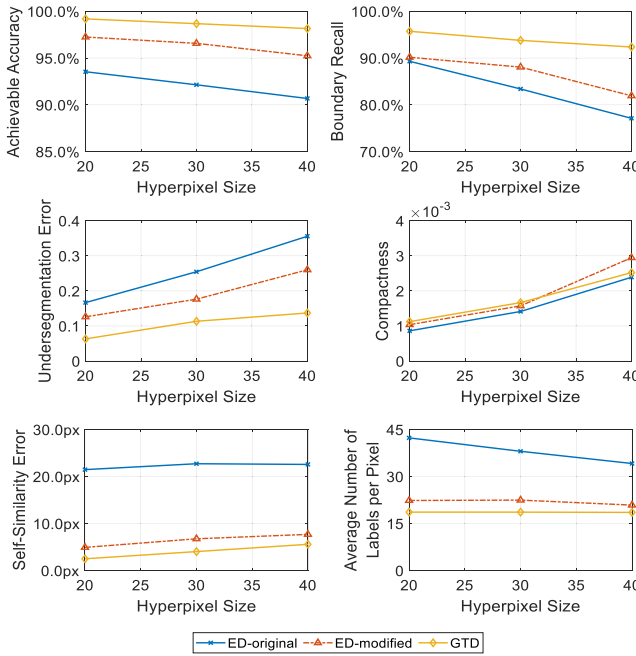


Fig. 10. Quantitative comparison on our proposed method using different estimated disparities namely, Estimated Disparity using (ED-original) [39]; modified Estimated Disparity (ED-modified); Ground Truth Disparity (GTD). Better disparity maps can significantly improve the hyperpixels performance.

as follows [21]:

$$\sigma^2(L) = \frac{1}{N_u \times N_v} \sum_{u,v} \frac{1}{|C_{u,v}|} \times \left\{ \sum_{u',v' \in C_{u,v}} occ_{u',v'}^{u,v} \left( w_{u',v'}^{u,v} (L_{u',v'}) - \overline{L_{u,v}} \right)^2 \right\},$$

$$LAC(L) = 10 \log_{10} \left( r^2 / \sigma^2(L) \right), \quad (15)$$

where  $N_u, N_v$  are the number of horizontal and vertical views,  $C_{u,v}$  is the closest 8 neighboring views of labeled view  $L_{u,v}$ ,  $occ_{u',v'}^{u,v}$  represents per-pixel weights where occluded regions between two adjacent views are set to 1 and 0 elsewhere,  $w_{u',v'}^{u,v}$  represents the warping function as explained in [21], to warp a given view using a disparity map between view

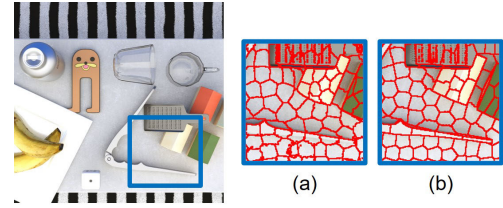


Fig. 11. Example of inaccurate over-segmentation of a non-Lambertian region of the nutcracker using different disparity maps during the clustering: a) Using estimated disparity; b) Using ground truth disparity. Accurate disparity maps can improve the over-segmentation performance.

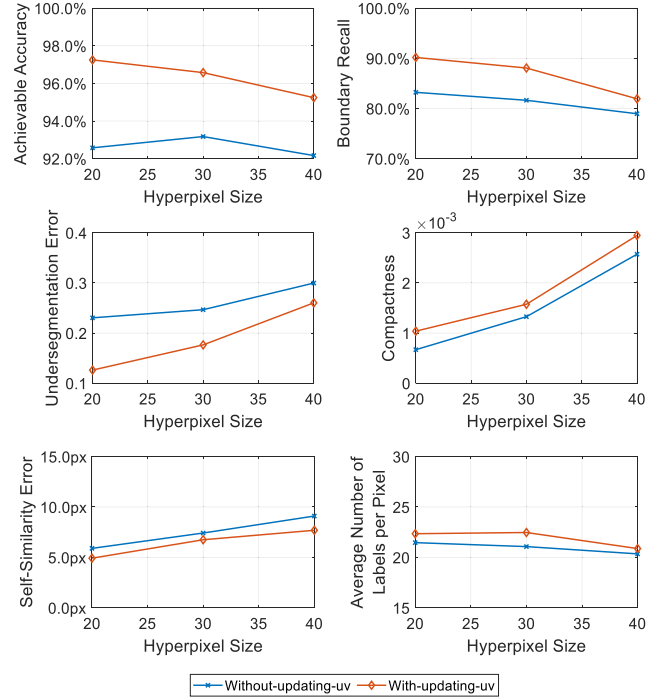


Fig. 12. Quantitative comparison of our proposed method with and without adjusting the centroids angular location during the clustering for sparse LFs.

$L_{u',v'}$  and  $L_{u,v}$ ,  $\overline{L_{u,v}}$ , is the mean of all the LF neighboring views warped into view  $L_{u,v}$ ,  $r$  is the pixels values range, and  $\sigma^2(L)$  is the mean of  $\sigma^2(L)$ . A higher  $LAC$  indicates better angular consistency.

#### D. Disparity Maps Estimation

As input, the proposed hyperpixels method requires disparity maps for all 4D LF views, to fully exploit LF cues during the 4D clustering. In the case of dense LFs, the recently proposed view-consistent depth estimation method in [38] is used. This method [38] relies heavily on the EPI structure and is designed only for dense LFs. In the case of sparse LFs, to the best of the authors' knowledge, only the deep learning based disparity estimation method proposed in [39] can estimate disparity (for all dense and sparse LF views, considering and ensuring cross-view consistency), with promising performance and has an open-source software. This method relies on initially estimating the corner views using a fine-tuned Flow Net 2.0 [42], [39]. Afterwards, the inner views disparity maps are synthesized and propagated using an occlusion-aware soft 3D reconstruction method proposed in [43] based on the corner



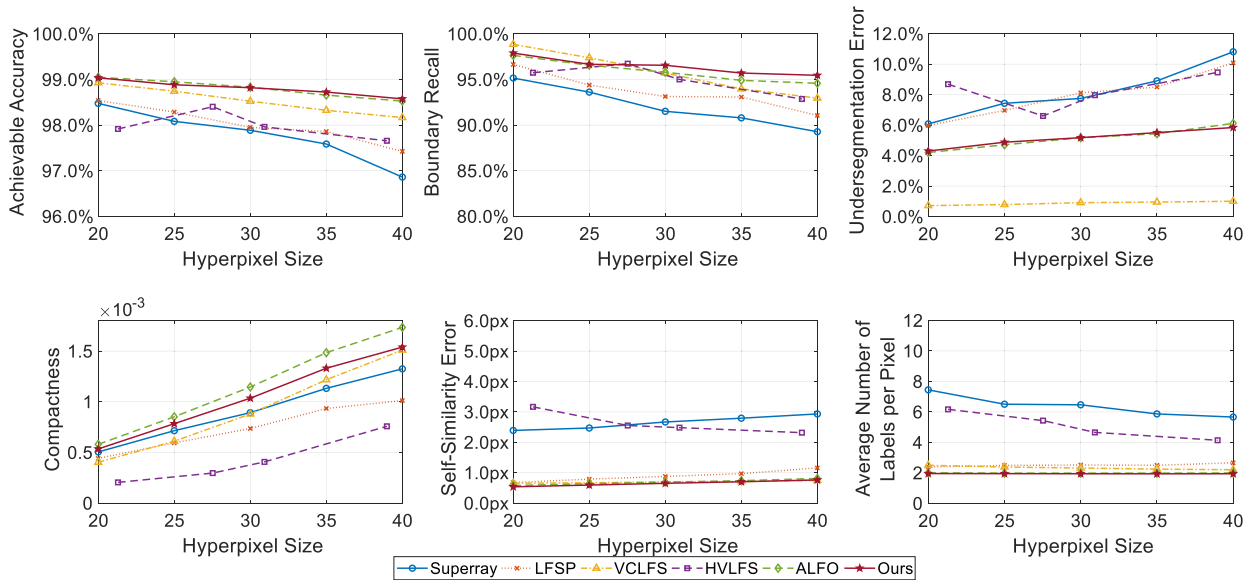


Fig. 13. Average quantitative evaluation on all 4D LFs of the dense HCI 4D LF dataset listed in Table II for different 4D LF over-segmentation methods.

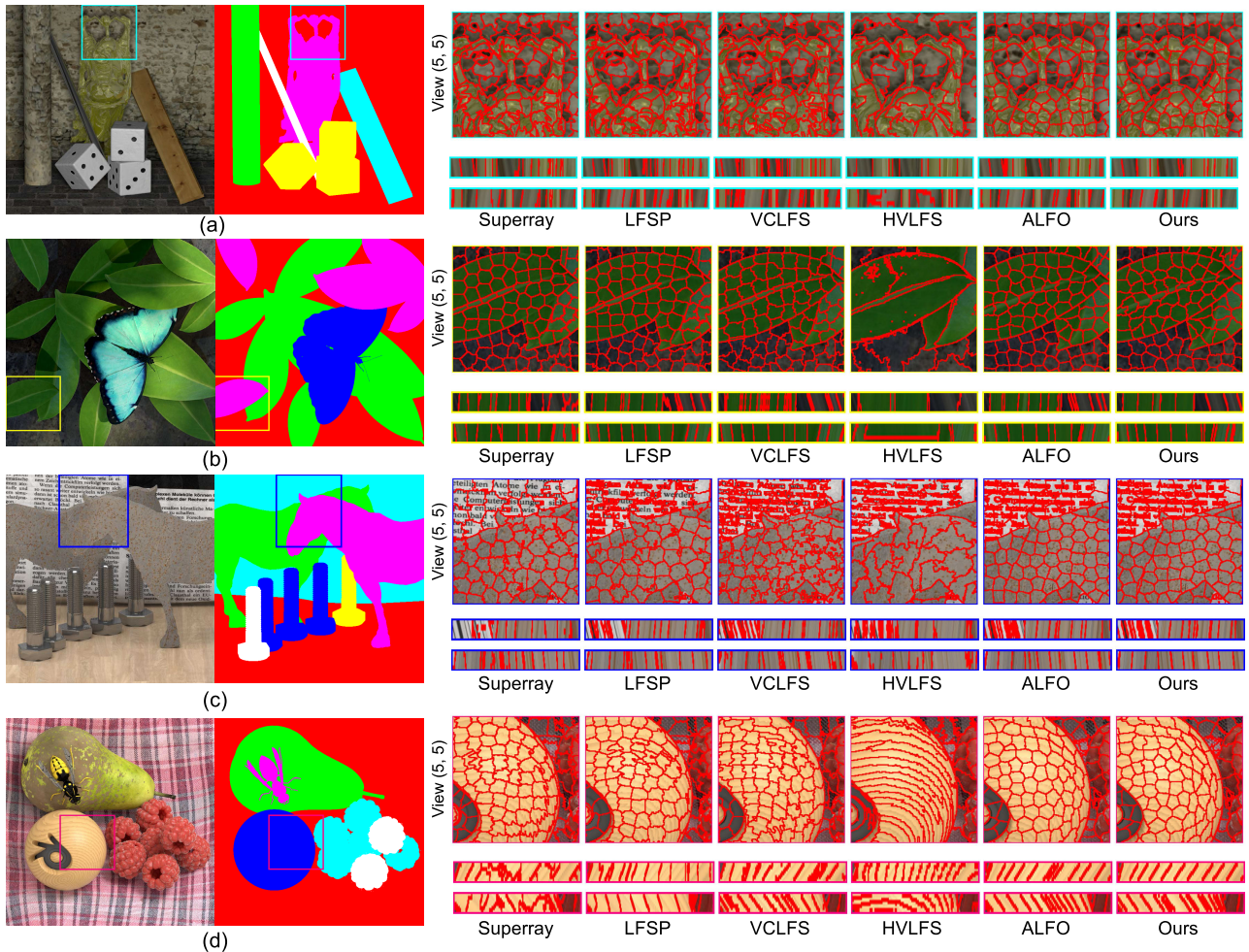


Fig. 14. Qualitative results using the densely sampled HCI 4D LF dataset. Challenging regions are selected to evaluate the robust balancing between spatial accuracy, compactness and cross-view consistency. For each LF, the central view, the vertical and horizontal EPIs are presented, respectively. As can be seen, our results adhere well to object boundaries and can accurately segment overlapping objects as in (b) and (c) and maintain compact and consistent across all views (as can be seen in the supplemental dynamic results).  $H_{size} = 20$ .

views. This method can generate accurate disparity maps for LFs with limited disparity ranges. However, the accuracy of the estimated disparity is significantly negatively affected

when large displacements exist between the corner views, especially for sparse LFs, which can dramatically affect the over-segmentation results. The authors extended this method

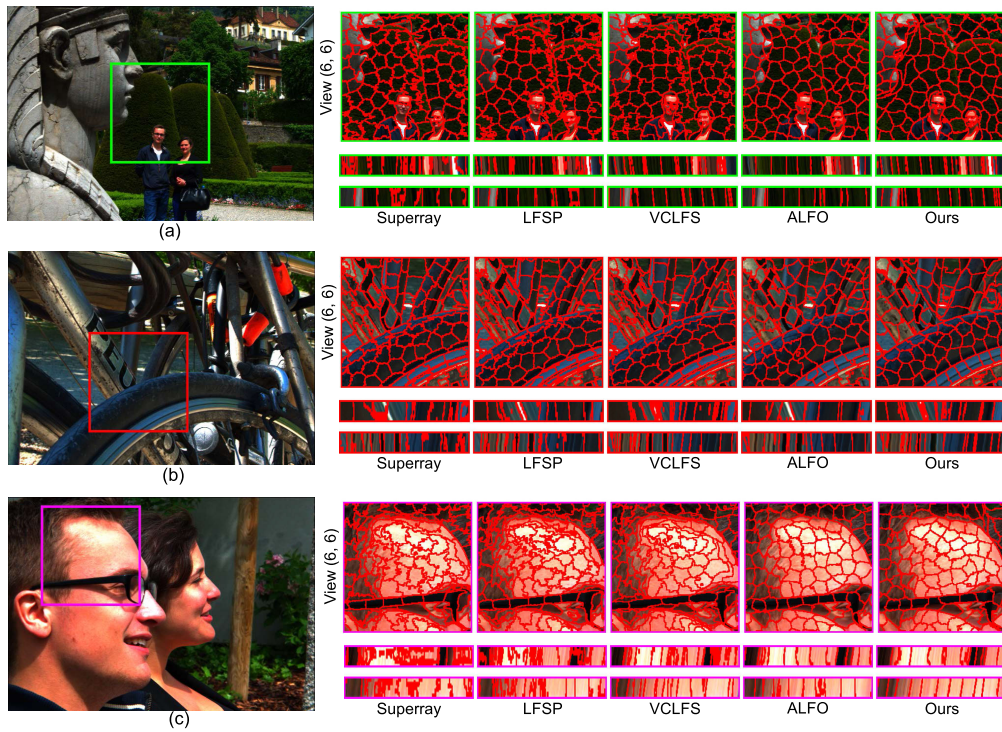


Fig. 15. Qualitative results using the densely sampled MMSPG 4D LF dataset. For each LF, the central view, the vertical and horizontal EPIs are presented, respectively. Regardless of the noise that exists in real LF views and non-even lighting, our results can adhere to object boundaries and can accurately segment challenging cases such as non-even lighting with complex texture and non-Lambertian regions and preserve compact and consistent across all views.  $H_{size} = 20$ .

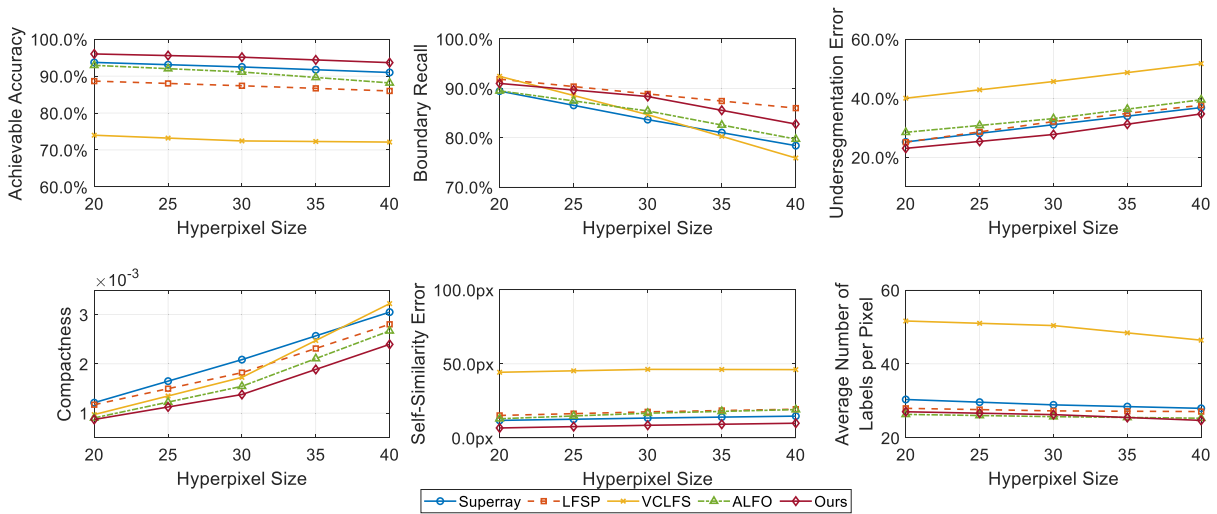


Fig. 16. Average quantitative evaluation on all 4D LFs of our sparse 4D LF dataset listed in Table II for different 4D LF over-segmentation methods.

in [44] to flexibly select any anchor views (e.g., not only corner views), but the disparity for only one target view can be estimated, hence no local or global angular consistency is considered when applying it for all LF views.

Therefore, the method in [39] is adopted in our experiment and the improved disparity estimation is used for all methods for sparse LFs. To ensure accuracy and local consistency in sparse LFs, instead of estimating the disparity for corner views and then propagating it to inner views that may include large-occluded regions, we estimate the disparity maps for every 4 adjacent views (e.g.,  $2 \times 2$ ) with step size equals to 2. This

way, there is no need for propagation using 3D reconstruction as in [43], and a significant improvement in disparity estimation accuracy is achieved, as can be seen in Fig. 9. Consequently, our over-segmentation performance is further improved in terms of hyperpixel accuracy, compactness, and cross-view consistency as shown in Fig. 10 and as discussed in the following section.

In conclusion, inaccurate disparity estimation can affect the hyperpixels results, as shown in Fig. 11, and the proposed hyperpixels method is positively affected by using more accurate disparity maps.



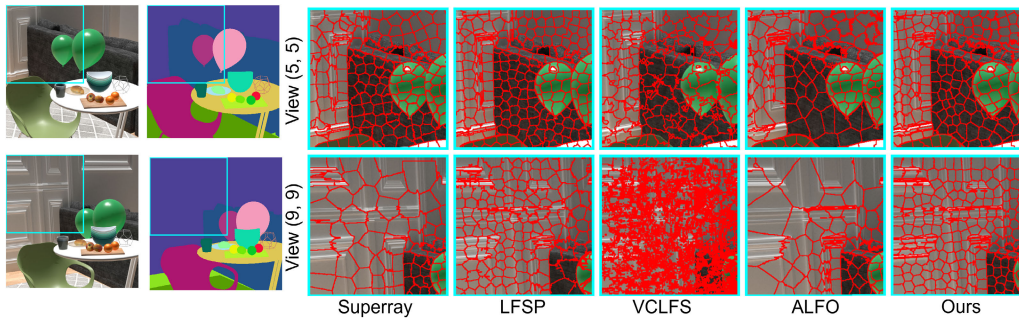


Fig. 17. Example of LF over-segmentation behavior for several methods for regions that do not exist in the central view. As can be seen inside the cyan square, a portion of the white region in view (9, 9) does not exist in the central view, i.e., view (5, 5). Our proposed method initializes centroids for these regions in 4D space before clustering. Therefore, hyperpixels remain with regular and similar sizes in all LF views and the accuracy and consistency are considered during the clustering for those regions.

### E. Qualitative and Quantitative Results

In this section, our results are presented and compared to the benchmark methods for several dense and sparse 4D LF datasets. All the results in Fig. 10 – Fig. 18 are generated using estimated disparity maps as explained in Section V-D and not the GT ones. The GT disparity maps are only used for computing the quantitative evaluations.

Before comparing our results with the existing methods, it is worth showing the effect of updating the centroids angular location during the clustering. In the case of dense LFs, the disparity range is narrow and in our experiments the disparity ranges were always less than the  $H_{size}$ . Therefore, almost all hyperpixels have a 2D slice in all LF views. Consequently, the over-segmentation performance is not significantly affected by adjusting the centroids angular locations. However, in the case of sparse LFs, not all hyperpixels have a slice in all LF views; hence, the effect of updating the centroids angular location can be noticed. The importance of updating the centroids angular location during the clustering is shown in Fig. 12 for sparse LFs. In Fig. 12, the average performance in terms of accuracy, compactness and angular consistency is notably improved.

The performance evaluation of our method compared with other existing methods presented in (Fig. 13 – Fig. 18, where hyperpixel size is the same as cluster/segment size in other methods) can be summarized based on each metric as follows:

- **Achievable Accuracy ( $\uparrow$ )**– This metric shows that using accurate disparity maps can affect the accuracy as seen in Fig. 10, where GT and different estimated disparity maps are used during the over-segmentation. As can be seen in Fig. 13 – Fig. 18, the hyperpixels method achieves outperforming accuracy by using adaptive 4D clustering along with hybrid spatio-angular features, for both dense and sparse LFs. The significance of exploiting disparity information as a clustering feature becomes apparent in challenging cases, such as overlapping objects with low color difference but at different depths. In Fig. 14b and Fig. 14c, overlapping leaves and the horses’ heads are examples of this type of challenging regions.
- **Boundary Recall ( $\uparrow$ )**– Our results robustly preserve the boundaries in dense and sparse LFs even in challenging regions, such as the horse heads in Fig. 14c, and non-Lambertian objects, as the glass cup in Fig. 18a. However,

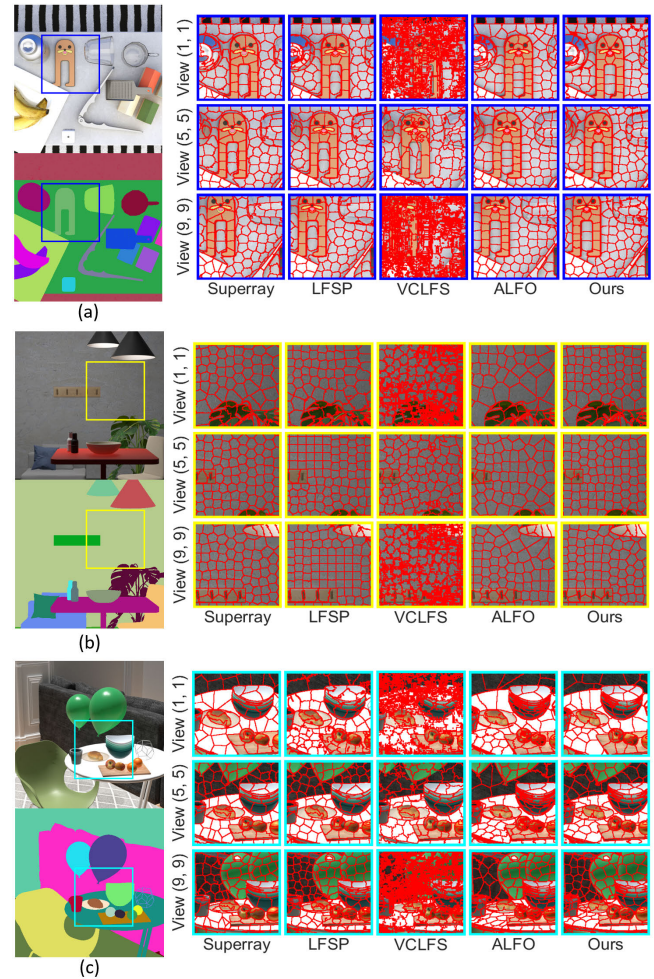


Fig. 18. Qualitative results using our generated sparse 4D LF dataset. Challenging regions are selected to evaluate the robust balancing between spatial accuracy, compactness, and cross-view consistency such as transparent glass, objects and large untextured regions as in the wall. Regardless of the wide disparity range in this dataset, the proposed hyperpixels are robust and consistent across views.  $H_{size} = 20$ .

if inaccurate disparity values are estimated, the  $BR$  results can be negatively affected as clearly presented in Fig. 10.

- **Under-segmentation Error ( $\downarrow$ )**– The proposed hyperpixels method balances the tradeoff between accuracy, shape



TABLE III  
LABELING-LF ANGULAR CONSISTENCY (LLFAC) FOR DENSE AND SPARSE LIGHT FIELDS ( $\uparrow$ )

4D LF	Disparity range	$H_{size}$														
		20					30					40				
		Superray [11]	LFSP [15]	VCLFS [16]	ALFO [17]	Ours	Superray [11]	LFSP [15]	VCLFS [16]	ALFO [17]	Ours	Superray [11]	LFSP [15]	VCLFS [16]	ALFO [17]	Ours
<b>Buddha</b>	[-8.5, 1.5]	39.00	<b>39.53</b>	39.19	39.52	39.42	38.53	39.28	38.92	<b>39.47</b>	39.21	38.03	<b>39.20</b>	38.91	39.01	38.94
<b>Papillon</b>	[-1.2, 0.9]	36.63	36.71	36.38	36.86	<b>36.94</b>	36.61	36.68	36.53	36.92	<b>36.99</b>	36.58	36.62	36.65	36.95	<b>37.02</b>
<b>Horses</b>	[-1.4, 0.9]	36.67	37.11	37.86	37.91	<b>38.12</b>	36.71	37.11	<b>37.43</b>	37.19	37.40	36.31	36.66	37.08	<b>37.24</b>	36.89
<b>StillLife</b>	[-2.7, 2.6]	35.76	36.47	37.03	<b>37.20</b>	37.13	35.30	36.34	36.78	37.11	<b>37.15</b>	33.74	36.01	36.93	37.12	<b>37.25</b>
<b>Kitchen</b>	[3.1, 13.5]	<b>36.13</b>	35.57	26.99	35.24	35.27	<b>35.56</b>	34.98	27.07	34.81	34.67	<b>35.08</b>	34.99	27.60	34.29	33.52
<b>Room</b>	[-18.3, 8.9]	30.04	28.86	27.24	29.85	<b>30.62</b>	29.97	29.34	27.50	29.94	<b>30.64</b>	30.36	29.74	27.78	29.99	<b>30.78</b>
<b>Balloons</b>	[-35.3, 3.2]	25.06	30.85	27.90	32.22	<b>32.51</b>	24.88	31.10	28.06	31.84	<b>32.38</b>	25.50	30.51	27.95	31.89	<b>32.36</b>
<b>Antique</b>	[-5.44, 1.17]	40.94	<b>41.16</b>	37.64	39.67	39.15	<b>40.70</b>	40.03	37.90	38.98	38.73	<b>40.94</b>	39.13	35.72	37.65	37.94
<b>Car</b>	[-1.55, 70.31]	29.44	28.58	27.17	29.73	<b>31.01</b>	29.03	28.71	25.98	29.63	<b>30.70</b>	28.53	28.51	26.10	29.29	<b>30.92</b>
<b>Chess</b>	[-15.8, 9.8]	<b>31.73</b>	29.90	28.12	30.99	31.61	<b>31.58</b>	30.03	28.54	30.83	31.46	31.23	30.22	27.92	30.02	<b>31.38</b>
<b>Leisure</b>	[-34.28, 123.34]	26.12	25.95	23.52	26.31	<b>26.90</b>	26.10	26.08	23.77	26.45	<b>26.95</b>	26.25	26.28	24.18	26.55	<b>26.98</b>
<b>Average</b>		33.41	33.70	31.73	34.14	<b>34.43</b>	33.18	33.61	31.68	33.92	<b>34.21</b>	32.96	33.44	31.53	33.64	<b>34.00</b>

TABLE IV

AVERAGE CPU TIME FOR VARIOUS CLUSTERING-BASED METHODS OVER SEVERAL LF DATASETS AND SIZES (IN SECONDS FOR ALL LF VIEWS)

$H_{size}$	4D LF dataset	Superray [11]	LFSP [15]	VCLFS [16]	ALFO [17]	Ours
20	HCI dataset	109.35	237.19	1837.95	443.56	630.93
	MMSPG dataset	76.84	153.19	1075.98	252.26	404.91
	Our generated dataset for sparse LFs	48.47	134.83	5753.73	169.94	279.96
40	HCI dataset	85.46	175.61	1663.29	453.74	745.32
	MMSPG dataset	61.37	122.99	1009.83	274.15	468.15
	Our generated dataset for sparse LFs	37.55	95.36	5721.17	250.48	340.22

regularity (i.e., compactness) and consistency by using the clustering weights adaptation. Hyperpixels results generate competitive  $UE$  in dense LFs and outperform the benchmark methods for sparse LFs, as shown in Fig. 13 and Fig. 16. Using accurate disparity maps can reduce  $UE$  as in Fig. 10. While the LFSP and VCLFS methods lead to lower under-segmentation errors for dense LFs, this is not necessarily true in terms of accuracy or consistency metric performance, as in Fig. 13.

- **Compactness ( $\uparrow$ )**– This metric reflects over-segmentation shape regularity that can be controlled during the clustering weights adaptation step. In most benchmark methods, the compactness parameter is either an input set by the user or is empirically set to a fixed value. However, in this paper, the clustering weight that affects the compactness is automatically adapted according to the LF content. The proposed method achieves competitive  $CP$  when compared to other benchmark methods for dense LFs. However, due to the new centroids creation in off-central views, we noticed that the benchmark methods achieve better  $CP$ . In some of the benchmark methods, when a region lacks a centroid projection, pixels in that region are grouped to the nearest segment, resulting in larger and more compact segments in off-central views as in Fig. 17. This situation increases the average compactness results and may affect the  $AA$  and  $UE$  performance. Hyperpixels compactness can be improved by using accurate disparity maps as in Fig. 10.
- **Consistency metrics:  $SS$  ( $\downarrow$ ),  $LP$  ( $\downarrow$ ),  $LLFAC$  ( $\uparrow$ )**– LF Over-segmentation consistency is an essential prop-

erty that can drastically affect subsequent editing tasks. The state-of-the-art methods have different techniques to ensure consistency, such as enforcing the continuity in the EPI space or using graph optimization. In this paper, we exploit per-pixel disparity to effectively project centroids across views and achieve cross-view consistency. As can be seen in Fig. 13, Fig. 16 and Table III, the proposed method achieves outperforming results in terms of  $SS$  and  $LP$  in dense and sparse LFs. Since there are no GT maps for the real LFs, angular consistency is not evaluated numerically. However, Fig. 14 and Fig. 15 show the angular consistency through the EPIs. Moreover, the angular consistency can be clearly noticed in the videos of the supplemental materials. Given the fact that in  $SS$  and  $LP$  metrics, the consistency is computed after warping the views into the central one and discarding the largely occluded region, we computed the  $LLFAC$  to fairly evaluate the labeling angular consistency for sparse 4D LFs. As seen in Table III and Fig. 16, the proposed method achieves the best angular consistency for sparse LFs. Additionally, as in Fig. 10, using better disparity estimation can improve cross-view consistency due to the accurate centroids projections.

To sum up, the proposed method achieves a robust balance between all the metrics for all tested 4D LFs without using any post-processing step to correct labeling the hyperpixels. For sparse LFs, we noticed that the methods that rely on post-processing optimization, such as the superray and LFSP methods, can generate compact and accurate over-segmentation for sparse LFs but are not necessarily

consistent across views. Moreover, a significant reduction is noticed in the VCLFS method performance when sparse LFs are used. Since the VCLFS method relies on the EPI structure and cannot adequately handle the irregular EPI structure in sparse LFs.

Finally, existing limitations in this proposed method, in some 4D LFs where the disparity is not accurately estimated (e.g., in real world 4D LF scenes and when non-Lambertian objects exist), inconsistent or inaccurate hyperpixels may be generated. As an example, Fig. 11 shows a failure case in a part of the metallic object that has inaccurate disparity values. To avoid that, using better disparity maps can significantly improve the final results. Finally, the current implementation is not optimized since this was out of this paper scope. Nevertheless, the average CPU time required by each method to over-segment 4D LFs using several LF datasets is shown in Table IV.

## VI. CONCLUSION

In this paper, the concept of hyperpixel for 4D LFs is initially defined. After that, a 4D LF over-segmentation method based on 4D  $K$ -means clustering is proposed to be used for sparse and dense 4D LFs. Moreover, our proposed method initializes the centroids in an occlusion-aware manner and uses an adaptive weighted 4D  $K$ -means clustering based on hybrid features.

The proposed hyperpixels method can be used as a pre-processing step for sparse and dense LF processing and editing, such as semantic segmentation and saliency detection. Quantitative and qualitative results show outperforming over-segmentation performance for dense and sparse 4D LFs.

In the future, we will further investigate how to exploit the non-linearities in the EPI space for sparse LFs and non-Lambertian objects, to enforce hyperpixel consistency across views. Additionally, we will consider further extending our method to generate hyperpixels for 5D LF videos.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Mira Rizkallah for providing her re-implementation of the Superray software and Dr. Rui Li for providing his generated labels from his method for them to compare with. They would also like to thank Mr. Numair Khan for publishing the software of all the used evaluation metrics that facilitated their comparisons.

## REFERENCES

- [1] G. Wu et al., "Light field image processing: An overview," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 926–954, Oct. 2017.
- [2] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," *Comput. Vis. Image Understand.*, vol. 166, pp. 1–27, Jan. 2018.
- [3] X. Luo, "Image compression via K-means and SLIC superpixel approaches," in *Proc. 4th Int. Conf. Machinery, Mater. Inf. Technol. Appl.*, Paris, France, Jan. 2016, pp. 1008–1012.
- [4] C. Conti, L. D. Soares, and P. Nunes, "Dense light field coding: A survey," *IEEE Access*, vol. 8, pp. 49244–49284, Mar. 2020.
- [5] D. Yeo, J. Son, B. Han, and J. H. Han, "Superpixel-based tracking-by-segmentation using Markov chains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 511–520.
- [6] X. Xie, G. Xie, X. Xu, L. Cui, and J. Ren, "Automatic image segmentation with superpixels and image-level labels," *IEEE Access*, vol. 7, pp. 10999–11009, Jan. 2019.
- [7] M. Hamad, C. Conti, A. M. de Almeida, P. Nunes, and L. D. Soares, "SLFS: Semi-supervised light-field foreground-background segmentation," in *Proc. Telecoms Conf. (ConfTELE)*, Leiria, Portugal, Feb. 2021, pp. 1–6.
- [8] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, "Supervoxel convolution for online 3D semantic segmentation," *ACM Trans. Graph.*, vol. 40, no. 3, pp. 1–15, Jun. 2021.
- [9] Y. Yan and J. Zhu, "Saliency detection based on superpixel correlation and cosine window filtering," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21205–21221, Aug. 2019.
- [10] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 1996, pp. 31–42.
- [11] M. Hog, N. Sabater, and C. Guillemot, "Superrays for efficient light field processing," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1187–1199, Oct. 2017.
- [12] X. Lv, X. Wang, Q. Wang, and J. Yu, "4D light field segmentation from light field super-pixel hypergraph representation," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 9, pp. 3597–3610, Sep. 2021.
- [13] M. Hamad, C. Conti, P. Nunes, and L. D. Soares, "View-consistent 4D light field style transfer using neural networks and over-segmentation," in *Proc. IEEE 14th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Nafplio, Greece, Jun. 2022, pp. 1–5.
- [14] H. Zhu, Q. Zhang, and Q. Wang, "4D light field superpixel and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6709–6717.
- [15] H. Zhu, Q. Zhang, Q. Wang, and H. Li, "4D light field superpixel and segmentation," *IEEE Trans. Image Process.*, vol. 29, pp. 85–99, 2020.
- [16] N. Khan, Q. Zhang, L. Kasser, H. Stone, M. H. Kim, and J. Tompkin, "View-consistent 4D light field superpixel segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, (South) Korea, Oct. 2019, pp. 7810–7818.
- [17] M. Hamad, C. Conti, P. Nunes, and L. D. Soares, "ALFO: Adaptive light field over-segmentation," *IEEE Access*, vol. 9, pp. 131147–131165, Sep. 2021.
- [18] R. Li and W. Heidrich, "Hierarchical and view-invariant light field segmentation by maximizing entropy rate on 4D ray graphs," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–15, Nov. 2019.
- [19] M. Wang, X. Liu, Y. Gao, X. Ma, and N. Q. Soomro, "Superpixel segmentation: A benchmark," *Signal Process., Image Commun.*, vol. 56, pp. 28–39, Aug. 2017.
- [20] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 7–55, 1987.
- [21] D. Egan, M. Alain, and A. Smolic, "Light field style transfer with local angular consistency," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 2300–2304.
- [22] R. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 10–17.
- [23] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 13961–13970.
- [24] P. Li and W. Ma, "OverSegNet: A convolutional encoder–decoder network for image over-segmentation," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108610.
- [25] C. Xu and J. J. Corso, "LIBSVX: A supervoxel library and benchmark for early video processing," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 272–290, Sep. 2016.
- [26] S.-H. Lee, W.-D. Jang, and C.-S. Kim, "Superpixels for image and video processing based on proximity-weighted patch matching," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 13811–13839, May 2020.
- [27] M. Hog, N. Sabater, and C. Guillemot, "Dynamic super-rays for efficient light field video processing," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Newcastle, U.K., Sep. 2018, pp. 1–12.
- [28] T. Herfet, K. Chelli, T. Lange, and R. Kremer, "Fristograms: Revealing and exploiting light field internals," 2021, *arXiv:2107.10563*.
- [29] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [30] X. Xiao, Y. Zhou, and Y. Gong, "Content-adaptive superpixel segmentation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2883–2896, Jun. 2018.

- [31] S. Wanner, S. Meister, and B. Goldlücke, "Datasets and benchmarks for densely sampled 4D light fields," *Vis., Model. Vis.*, vol. 13, pp. 225–226, Sep. 2013.
- [32] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proc. 8th Int. Conf. Quality Multimedia Exper. (QoMEX)*, Lisbon, Portugal, Jun. 2016, pp. 1–2.
- [33] *Blender—A 3D Modelling and Rendering Package*. Accessed: Dec. 9, 2021. [Online]. Available: <http://www.blender.org>
- [34] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldlücke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Taipei, Taiwan, Nov. 2016, pp. 19–34.
- [35] *Chocofur Main Page*. Accessed: Dec. 28, 2021. [Online]. Available: <https://chocofur.com/>
- [36] *BlenderKit—Get Free 3D Models, Materials & More Directly in Blender*. Accessed: Dec. 9, 2021. [Online]. Available: <https://www.blenderkit.com/>
- [37] *3D Models for Professionals: TurboSquid*. Accessed: Dec. 9, 2021. [Online]. Available: <https://www.turbosquid.com/>
- [38] N. Khan, M. H. Kim, and J. Tompkin, "View-consistent 4D light field depth estimation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2020, pp. 1–13.
- [39] X. Jiang, J. Shi, and C. Guillemot, "A learning based depth estimation framework for 4D densely and sparsely sampled light fields," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 2257–2261.
- [40] T. Wang, A. A. Efros, and R. Ramamoorthi, "Depth estimation with occlusion modeling using light-field cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2170–2181, Nov. 2016.
- [41] H. Zhu, Q. Wang, and J. Yu, "Occlusion-model guided antiocclusion depth estimation in light field," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 965–978, Oct. 2017.
- [42] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1647–1655.
- [43] E. Penner and L. Zhang, "Soft 3D reconstruction for view synthesis," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–11, Nov. 2017.
- [44] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5867–5880, Dec. 2019.



**Maryam Hamad** (Graduate Student Member, IEEE) received the B.E. degree in computer systems engineering (CSE) from Palestine Technical University-Kadoorie (PTUK), Palestine, in 2018, covered by an excellence scholarship. She is currently pursuing the fully granted Ph.D. degree with the Instituto Universitário de Lisboa (ISCTE-IUL), Portugal. During her degree, she spent one semester as an Exchange Student with Middle East Technical University (METU) with ERASMUS+ Program, Turkey. She completed her professional internship in information science

and technology with IAESTE Program as a Researcher with the Multimedia Signal Processing Group, Instituto de Telecomunicações, Portugal, where she is also a Researcher. Her current research interests include immersive visual technologies, such as light field imaging, digital image processing, and computer vision. She is a member of the IEEE Women in Engineering Society, the IEEE Signal Processing Society, and the IEEE Young Professionals Group. She acts as a reviewer for IEEE ACCESS journal.



**Caroline Conti** (Member, IEEE) received the B.Sc. degree in electrical engineering from Universidade de São Paulo (USP), Brazil, in 2010, and the Ph.D. degree in information science and technology from Instituto Universitário de Lisboa (ISCTE-IUL), Portugal, in 2017. She is currently a Senior Researcher with the Multimedia Signal Processing Group, Instituto de Telecomunicações, and an Assistant Professor with the Department of Information Science and Technology, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal. Her research interests include immersive visual technologies and image and video processing, including light field processing and coding. She has contributed more than 25 papers to international journals and conferences in these areas. She serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING. She has been a Guest Editor for *Signal Processing: Image Communication* (Elsevier). She actively participates as a reviewer for various IEEE and EURASIP journals and conferences.



**Paulo Nunes** (Member, IEEE) received the degree in electrical and computers engineering from Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 1992, and the M.Sc. and Ph.D. degrees in electrical and computers engineering from IST in 1996 and 2007, respectively. He is currently a Senior Researcher with the Multimedia Signal Processing Group, Instituto de Telecomunicações, Portugal. He is also an Associate Professor with the Department of Information Science and Technology, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal.

He has coordinated and participated in various national and international (EU) funded projects and has acted as a Project Evaluator for the European Commission. His current research interests include 2D/3D image and video processing and coding, namely light field image and video processing and coding. He acts often as a reviewer for various ACM, EURASIP/Elsevier, IEEE, IET, MDPI, SPIE, and Springer conferences and journals and a member of the program and organizing committees of various international conferences. He has contributed more than 70 papers to international journals and conferences in these areas.



**Luís Ducla Soares** (Senior Member, IEEE) received the Licenciatura and Ph.D. degrees in electrical and computer engineering from Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 1996 and 2004, respectively. He is currently a Senior Researcher with the Multimedia Signal Processing Group, Instituto de Telecomunicações, Portugal. He is also an Associate Professor with the Department of Information Science and Technology, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal. His research interests include image and video

coding/processing, including light field coding and processing as well as biometric recognition. He has contributed more than 70 papers to international journals and conferences in these areas. In addition, he has participated in the development of the MPEG-4 Visual standard, as well as in several national and international projects. He is a member of the editorial board of the *EURASIP Signal Processing* (Elsevier). In parallel, he acts as a reviewer for several IEEE, IET, and EURASIP journals and conferences.