Gelman *et al. EURASIP Journal on Advances in Signal Processing* 2012, **2012**:61
http://asp.eurasipjournals.com/content/2012/1/61

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Layer-based sparse representation of multiview images

Andriy Gelman[1*], Jesse Berent[2] and Pier Luigi Dragotti[1]

Abstact

This article presents a novel method to obtain a sparse representation of multiview images. The method is based on the fact that multiview data is composed of epipolar-plane image lines which are highly redundant. We extend this principle to obtain the layer-based representation, which partitions a multiview image dataset into redundant regions (which we call layers) each related to a constant depth in the observed scene. The layers are extracted using a general segmentation framework which takes into account the camera setup and occlusion constraints. To obtain a sparse representation, the extracted layers are further decomposed using a multidimensional discrete wavelet transform (DWT), first across the view domain followed by a two-dimensional (2D) DWT applied to the image dimensions. We modify the viewpoint DWT to take into account occlusions and scene depth variations. Simulation results based on nonlinear approximation show that the sparsity of our representation is superior to the multi-dimensional DWT without disparity compensation. In addition we demonstrate that the constant depth model of the representation can be used to synthesise novel viewpoints for immersive viewing applications and also de-noise multiview images.

1 Introduction

The notion of sparsity, namely the idea that the essential information contained in a signal can be represented with a small number of significant components, is widespread in signal processing and data analysis in general. Sparse signal representations are at the heart of many successful signal processing applications, such as signal compression and de-noising. In the case of images, successful new representations have been developed on the assumption that the data is well modelled by smooth regions separated by edges or regular contours. Besides wavelets, which have been successful for image compression [1], other examples of dictionaries that provide sparse image representations are curvelets [2], contourlets [3], ridgelets [4], directionlets [5], bandlets [6,7] and complex wavelets [8,9]. We refer the reader to a recent overview article [10] for a more comprehensive review on the theory of sparse signal representation.

In parallel and somewhat independently to these developments, there has been a growing interest in the capture and processing of multiview images. The popularity of this approach has been driven by the advent of novel exciting applications such as immersive communication [11] or free-viewpoint and three-dimensional (3D) TV [12]. At the heart of these applications is the idea that a novel arbitrary photorealistic view of a real scene can be obtained by proper interpolation of existing views. The problem of synthesising a novel image from a set of multiview images is known as image-based rendering (IBR) [13].

Multiview data sets are inherently multi-dimensional. In the most general case multiview images can be parameterised using a single 7D function called the plenoptic function [14]. The dimensions, however, can be reduced by making some simplifying assumptions as discussed in the next section. In particular, the assumption that a camera can move only along two directions leads to the 4D light field parameterisation [15]. If the camera moves only along a straight line the 3D epipolar-plane image (EPI) volume is obtained. We will discuss and use these two parameterisations throughout the article. Intuitively, in the case of a multi-view image array which captures the same scene from different locations, a significantly more sparse representation can be obtained than the independent analysis of each image. When dealing with multiview images, however, the data

* Correspondence: andriy.gelman@imperial.ac.uk
[1]Communications and Signal Processing Group, Imperial College London, London SW7 2AZ, UK
Full list of author information is available at the end of the article

model must take into account appearing (disocclusions) and disappearing (occlusions) objects. This nonlinear property means that finding a sparse representation is inherently more difficult than in the two-dimensional (2D) case. For this reason, in this article we propose a hybrid method to obtain a sparse representation of multiview images. The fundamental component of the algorithm is the layer-based representation. In many situations, it is possible to divide the observed scene into a small number of depth layers that are parallel to the direction of camera motion. The layer-based representation partitions the multiview images into a set of layers each related to a constant depth in the observed scene. See also Figure 1 for a visual example of the partition. We present a novel method to extract these regions, which takes into account the structure of multiview data to achieve accurate results. In the case of the 4D light field, the sparse representation of the data is then obtained by taking a 4D discrete wavelet transform (DWT) of each depth layer. First we take a view compensated DWT along the two view directions, then the 2D separable spatial DWT is taken. This new representation is more effective than a standard separable DWT and this is shown using nonlinear approximation results. In addition, we present IBR and de-noising applications based on the extracted layers.

The article is organised as follows. Next we review the structure of multiview data, discuss the layer-based representation and present a high-level overview of our proposed method. In Section 3 we present the layer extraction algorithm. The multi-dimensional DWT is discussed in Section 4. We finally evaluate the proposed sparse representation in Section 5 and conclude in Section 6.

2 Multiview data structure

We start by introducing the plenoptic function and the structure of multiview data. In addition we present a layer-based representation that exploits the multiview structure to partition the data into volumes each related to a constant depth in the scene.

2.1 Plenoptic function

In the IBR framework, multiview images form samples of a multi-dimensional structure called the *plenoptic function* [14]. Introduced by Adelson and Bergen, this function parameterises each light ray with a 3D point in space (V_x, V_y, V_z) and its direction of arrival (θ,φ). Two further variables λ and t are used to specify the wavelength and time, respectively. In total the plenoptic function is therefore seven dimensional:

$$I = P_7(V_x, V_y, V_z, \theta, \phi, \lambda, t), \tag{1}$$

where I corresponds to the light ray intensity.

In practise, however, it is not feasible to store, transmit or capture the 7D function. A number of simplifications are therefore applied to reduce its dimensionality. Firstly, it is common to drop the λ parameter and instead deal with either the monochromatic intensity or the red, green, blue (RGB) channels separately. Secondly, the light rays can be recorded at a specific moment in time, thus dropping the t parameter. This simplification can for example be applied when viewing a stationary scene. The resulting object is a 5D function.

A popular parameterisation of the plenoptic function, known as the *light field* [15] defines each light ray by its intersection with a camera plane and a focal plane:

$$I = P_4(V_x, V_y, x, y), \tag{2}$$

where as illustrated in Figure 2, (V_x, V_y) and (x, y) correspond to the coordinates of the camera and the focal plane, respectively. Observe that the dataset can be analysed as a 2D array of images, where each image is formed by the light rays which pass through a specific point on the camera plane. In Figure 3 we illustrate an

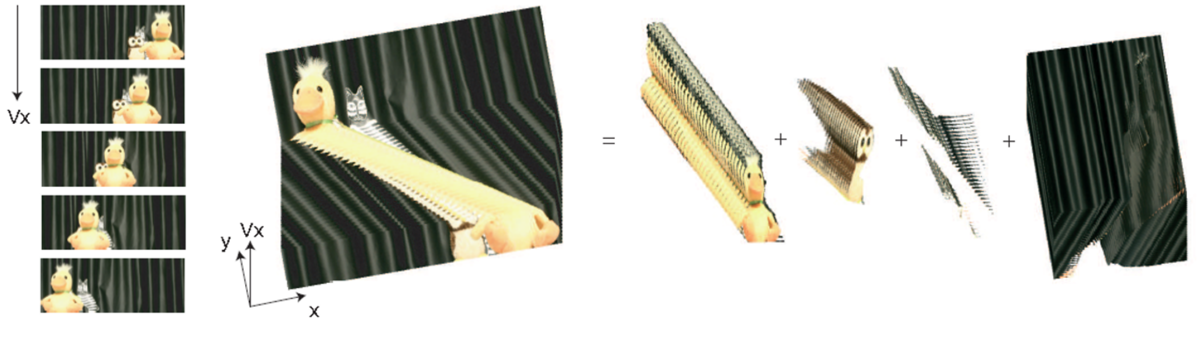


Figure 1 Animal Farm layer-based representation [34]. The dataset can be divided into a set of volumes where each one is related to a constant depth in the scene. Observe that the layer contours at each viewpoint remain constant, unless there is an intersection with another layer which is modelled by a smaller depth.

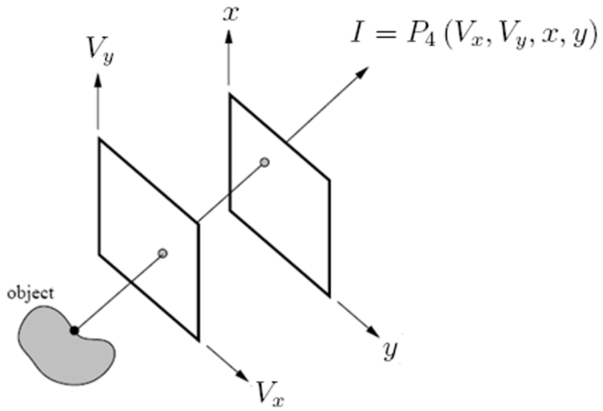


Figure 2 Light field parameterisation. Each light ray is defined by its intersection with a camera plane (V_x, V_y) and a focal plane (x, y) [36].

example of a light field with 16-camera locations. The camera positions are evenly spaced on a 2D grid (V_x, V_y).

The light field can be further simplified by setting the 2D camera plane to a line. This is also known as the EPI volume [16]:

$$I = P_3(V_x, x, y). \quad (3)$$

In comparison to the light field, the EPI is easier to visualise and in the following sections we use it to present a number of concepts. All of the properties are however easily generalised to the light field. Next, we review the EPI and light field structure and present the layer-based representation.

2.2 EPI and light field structure

In this section we show that an EPI volume and a light field are structured datasets. By structure we mean that the fundamental component of multiview images are lines along which the intensity of the pixels is constant. This concept is shown in Figure 4c. This illustration is obtained by stacking an array of images into a volume and taking a cross section through the dataset. It can be clearly observed that pixels are redundant along lines of varying gradients. These pixels along which the intensity of the volume is constant are also known as an *EPI line*.

In order to demonstrate why the fundamental component of multiview images are EPI lines, consider the setup in Figure 4a. Here we show a simplified version of the scene: the horizontal axis corresponds to the camera location line; the line parallel to it defines the focal plane of each camera[a]; and the vertical axis defines the depth of the scene. The curved line thus corresponds to the surface of the object.

Given this setup consider a point in space with coordinates (X, Y, Z). Assuming a Lam-bertian scene[b] this point will appear in each of the images with coordinates

$$x = \frac{fX}{Z} - \frac{fV_x}{Z}, \quad (4)$$

$$y = \frac{fY}{Z}, \quad (5)$$

where f is the focal length. As illustrated in Figure 4b, x is linearly related to the camera location V_x. The rate of change in the pixel location, also known as the

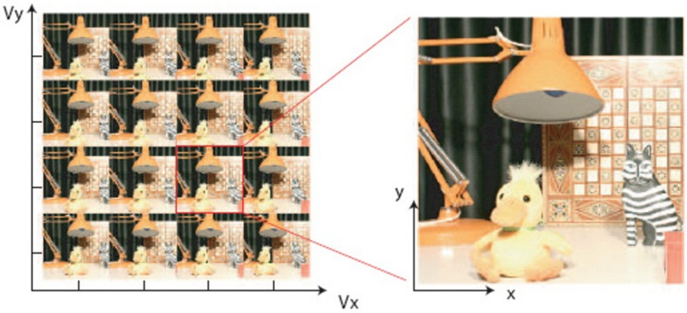
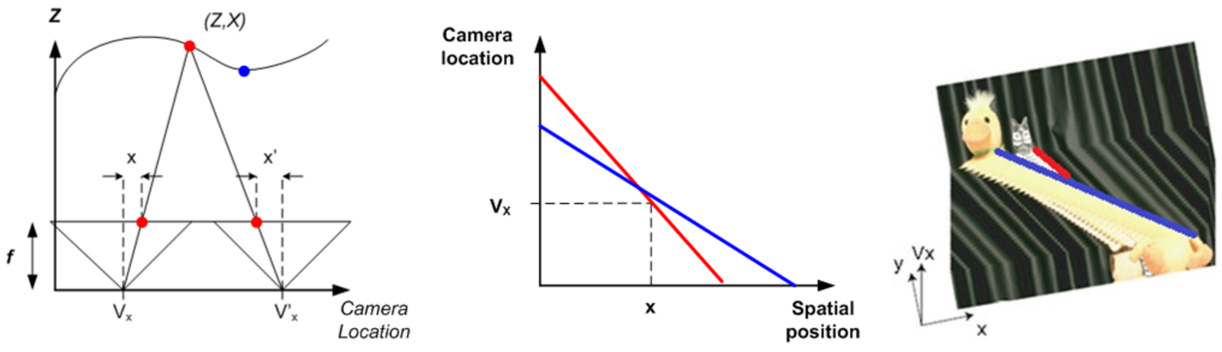


Figure 3 Captured light field [34]. Dataset can be analysed as a 2D array of images.



(a) (b) (c)

Figure 4 EPI volume structure. **(a)** Camera setup. The sampling camera moves along a straight line; the direction of the camera is perpendicular to the camera location line. **(b)** Each point in space maps to a line in the EPI volume. Observe that the blue object is closer to the focal plane and therefore occludes the red object. It can be shown using (4) and (5) that a data sample (x, y, V_x) can be mapped onto a different viewpoint $V_x{}'$ with spatial coordinates $x' = x - \frac{f(V_x' - V_x)}{Z}$ and $y' = y$. **(c)** Shows a cross section of the EPI volume. This figure is obtained by stacking a 1D array of images into a volume and taking a cross section of the dataset. Two EPI lines which correspond to two points in space are illustrated.

disparity $\Delta p = \frac{f}{Z}$, is inversely related to the depth of the object. Thus, a point in space maps to a line in the EPI volume.

However, the above analysis does not take into account occlusions. Clearly when two lines intersect, the EPI line corresponding to a smaller depth (larger disparity) will occlude all the EPI lines which are related to a larger depth (smaller disparity) in the scene. This principle is illustrated in Figure 4c.

The above concepts can also be extended to the light field, where the camera is allowed to move along two dimensions (V_x, V_y). In this case, a point (X, Y, Z) maps onto a 2D plane as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ V_x \\ V_y \end{pmatrix} = \begin{pmatrix} (X - V_x)f/Z \\ (Y - V_y)f/Z \\ V_x \\ V_y \end{pmatrix}. \qquad (6)$$

2.3 Layer-based representation

The layer-based representation is an extension of the EPI line concept. The representation partitions the multiview data into homogenous regions, where each layer is a collection of EPI lines modelled by a constant depth plane. An example of a layer-based representation is shown in Figure 1.

Consider a set of EPI lines modelled by a constant disparity Δp_k as shown in Figure 5a. We define the layer carved out by the EPI lines with $\mathcal{H}_k$ and the boundary which delimits the region with Γ_k Assuming there are no occlusions, observe that using (4) and (5), the boundary Γ_k can be defined by a contour on one of the viewpoints projected to the remaining frames. More specifically, if we define the contour γ_k (s) = [x (s), y (s)] to be the boundary on the viewpoint (V_x = 0), we obtain the relationship

$$\Gamma_k(s, V_x) = \begin{pmatrix} x(s) - \Delta p_k V_x \\ y(s) \\ V_x \end{pmatrix}, \qquad (7)$$

where s parameterises the contour γ_k (s).

In order to take into account occlusions, we can use the same principles as in the case of EPI lines; a layer will be occluded when it intersects with other layers which are related to a smaller depth in the scene. We illustrate this in Figure 5, which shows that when two layers intersect we obtain their visible representations[c] $\mathcal{H}_{k-1}^{\mathcal{V}}$ and $\mathcal{H}_k^{\mathcal{V}}$. In this example the layers are ordered in terms of increasing depth (i.e., $\mathcal{H}_k$ corresponds to a larger depth than $\mathcal{H}_{k-1}$). In general, the visible regions of a layer can be defined as

$$\mathcal{H}_k^{\mathcal{V}} = \mathcal{H}_k \cap \overline{\left(\bigcup_{j=1}^{k-1} \mathcal{H}_j\right)}. \qquad (8)$$

We illustrate the layers $\mathcal{H}_k$ and $\mathcal{H}_k^{\mathcal{V}}$ from the Animal Farm dataset in Figure 6.

There are a number of advantages to segmenting a multiview dataset into layers. Firstly, each layer is highly redundant in the direction of the disparity Δp. This is due to the fact that each layer consists of EPI lines modelled by a constant depth. Secondly, any occluded regions are explicitly defined by the representation. These regions correspond to artificial boundaries, and

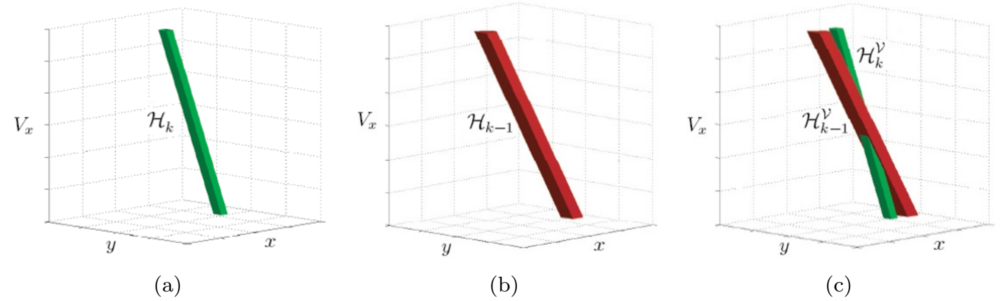


Figure 5 Comparison between two layers $\mathcal{H}_{k-1}$, $\mathcal{H}_k$ and their intersection. The layers are ordered in terms of depth (i.e., $\mathcal{H}_{k-1}$ corresponds to a smaller depth than $\mathcal{H}_k$). **(a)** A set of EPI lines related to constant disparity Δp_k. The collection of EPI lines carve out a layer $\mathcal{H}_k$. Observe that the complete segmentation of the layer can be defined by a boundary on one viewpoint projected to the remaining frames. **(b)** $\mathcal{H}_{k-1}$ modelled by a constant disparity Δp_{k-1}. **(c)** When the two layers intersect, $\mathcal{H}_{k-1}$ will occlude $\mathcal{H}_k$ as it is modelled by a smaller depth. We define the visible volumes with $\mathcal{H}_{k-1}^{\mathcal{V}}$ and $\mathcal{H}_k^{\mathcal{V}}$.

their specific locations can be used to design a transform which takes them into account. Thirdly, the boundary of each layer can be efficiently defined by a contour on one viewpoint $\gamma(s)$ and its disparity Δp. This is important if a compression algorithm based on the sparse representation is to be implemented, where the segmentation of each layer must also be transmitted.

2.4 Sparse representation method high-level overview

We use the above analysis to develop a new method that provides sparse representations of multiview images. The method is outlined in Figure 7. The first step of the method is to obtain a layer-based representation. As highlighted in Section 2.3 each layer is modelled by a constant depth plane and a contour on one of the image viewpoints. To extract these layers, we use a variational framework where the general segmentation results are modified to include the camera setup and the occlusion constraints.

In the following step we decompose the layers using a 4D DWT applied in a separable fashion across the viewpoint and the spatial dimensions. We modify the viewpoint transform to include disparity compensation and also efficiently deal with disoccluded regions. Additionally, the transform is implemented using the lifting scheme [17] to reduce the complexity and maintain invertibility.

In the following sections we describe the layer extraction and 4D DWT stages in more detail.

3 Layer-based segmentation

Data segmentation is the first stage of the proposed method. Here we introduce our segmentation algorithm which achieves accurate results by taking into account the structure of multiview data. We introduce the method by first describing a general segmentation problem and then showing how that solution can be adapted to extract layers from a light field dataset.

3.1 General region-based data segmentation

Consider a general segmentation problem shown in Figure 8. The aim is to partition an m-dimensional dataset $\mathcal{D} \subset \mathbb{R}^m$ into subsets $\mathcal{H}$ and $\bar{\mathcal{H}}$ where the boundary which delimits the two regions is defined by $\Gamma(\boldsymbol{\sigma})$ with $\boldsymbol{\sigma} \in \mathbb{R}^{m-1}$. This type of problem can be solved using an optimisation framework, where the boundary is obtained

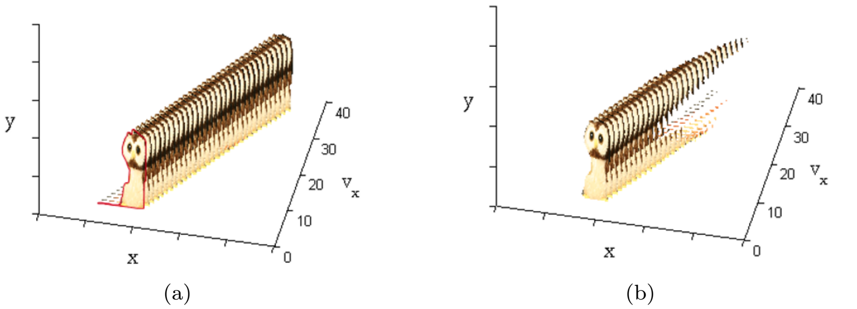


Figure 6 Layer from the animal farm dataset. **(a)** The unoccluded layer $\mathcal{H}_k$ can be defined using the contour $\gamma_k(s)$ on one viewpoint projected to the remaining frames. The 2D contour is denoted by the red curve on the first image. **(b)** Occluded layer $\mathcal{H}_k^{\mathcal{V}}$ can be inferred by removing the regions which intersect with other layers related to a smaller depth.

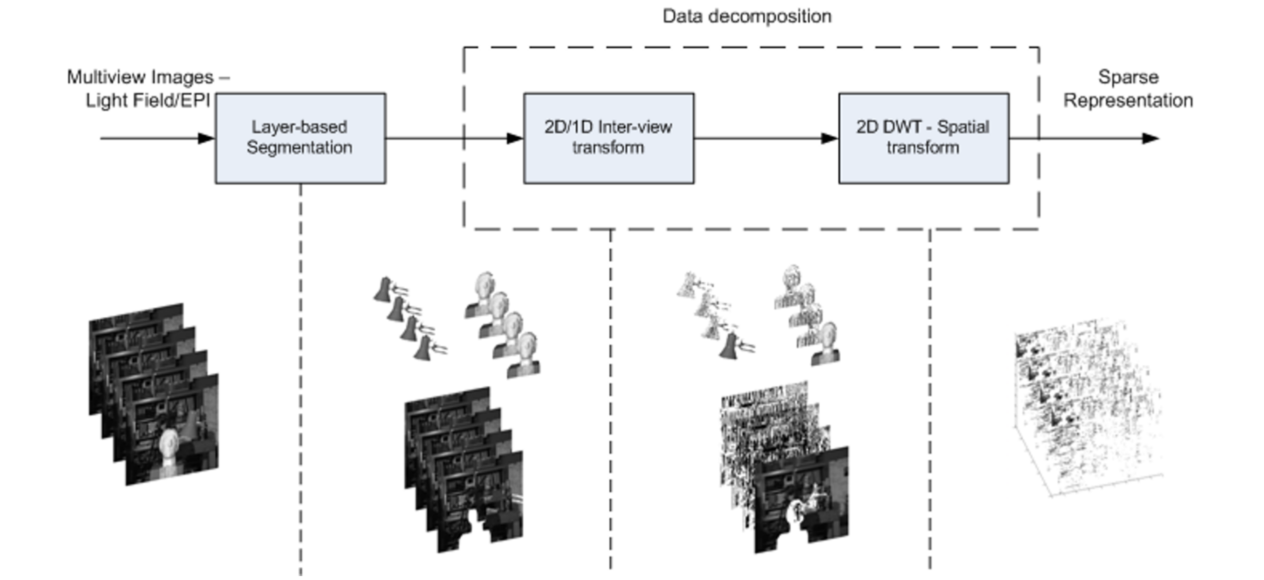

Data decomposition
Multiview Images – Light Field/EPI
Layer-based Segmentation
2D/1D Inter-view transform
2D DWT - Spatial transform
Sparse Representation

Figure 7 High-level block diagram. The data is initially segmented into layers where each volume is related to a constant depth in the scene. The obtained layers are then decomposed using a 4D DWT along the viewpoint and spatial dimensions. Additionally, we illustrate the obtained transform coefficients at each stage of the method.

by minimising an objective function *J*:

$$\Gamma = \arg\min\{J(\Gamma)\}. \tag{9}$$

The cost function in (9) can be defined using either a boundary or region-based approach. The boundary methods evaluate the cost only on Γ and, hence, they are influenced by local data properties and easily affected by noise. In contrast, the region-based methods evaluate the cost function over a complete region and are therefore more robust. A typical region-based cost function [18] can be defined as:

$$J(\Gamma) = \int_{\mathcal{H}} d(\mathbf{x}, \mathcal{H})d\mathbf{x} + \int_{\bar{\mathcal{H}}} d(\mathbf{x}, \bar{\mathcal{H}})d\mathbf{x} + \int_{\Gamma} \eta d\boldsymbol{\sigma}, \tag{10}$$

where the descriptor d (·) measures the homogeneity of each region and $\mathbf{x}$ ε $\mathbb{R}^m$. The descriptor can be designed such that when $\mathbf{x}$ belongs to the region $\mathcal{H}$, d $(\mathbf{x},\mathcal{H})$ tends to zero and vice versa. Note also that (10) has an additional regularisation term η, which acts to minimise the length of the boundary.

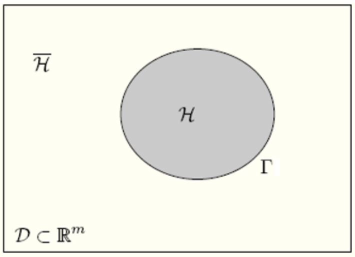


Figure 8 An *m* dimensional dataset $\mathcal{D}$ is partitioned into $\mathcal{H}$ and $\bar{\mathcal{H}}$. The boundary is a closed curve defined by Γ [34] and is an (*m*-1)-dimensional object.

The optimisation problem defined in (10) cannot be solved directly for Γ. An iterative solution can, however, be obtained by making the boundary a function of an evolution parameter τ. Consider modeling the boundary using a partial differential equation (PDE), also known as an active contour [19]:

$$\frac{\partial\Gamma(\boldsymbol{\sigma},\tau)}{\partial\tau} = \mathbf{v}(\boldsymbol{\sigma},\tau) = F(\boldsymbol{\sigma},\tau), \mathbf{n}(\boldsymbol{\sigma},\tau), \tag{11}$$

where $\mathbf{v}$ is a velocity vector, which can be expressed in terms of a scalar force F acting in the outward normal direction $\mathbf{n}$ to the boundary. The velocity vector $\mathbf{v}$ can be evaluated in terms of the descriptor $d(\cdot)$ by differentiating (10) with respect to τ. Applying the Eulerian framework [18], the derivative can be expressed in terms of boundary integrals:

$$\frac{\partial J(\Gamma(\tau))}{\partial\tau} = \int_{\Gamma(\tau)} [d(\mathbf{x},\mathcal{H}) - d(\mathbf{x},\bar{\mathcal{H}}) + \eta\kappa(\mathbf{x})](\mathbf{v}\cdot\mathbf{n})d\boldsymbol{\sigma}, \tag{12}$$

where κ is the curvature of the boundary Γ and · denotes the dot product. Observe that v and n correspond to the velocity and the normal vectors in (11), respectively.

The velocity vector, which evolves Γ in the steepest descent direction can hence be deduced using the

Cauchy-Schwarz inequality as:

$$\mathbf{v} = \left[d(\mathbf{x}, \bar{\mathcal{H}}) - d(\mathbf{x}, \mathcal{H}) - \eta\kappa(\mathbf{x})\right]\mathbf{n}. \tag{13}$$

The above framework is also known as 'competition-based' segmentation. This is clear from (13), where a point on the boundary will experience a positive force if it belongs to the region $\mathcal{H}$ and vice versa, hence evolving the contour in the correct direction. In conclusion, the general segmentation problem can be solved by modeling the boundary Γ as a PDE and evolving the contour in the direction of the velocity vector **v**.

3.2 Multiview image segmentation

In the case of a light field, the goal is to extract N layers, where each volume is modelled by a constant depth Z_k or the associated disparity Δp_k. In the context of the previous section, this is equivalent to segmenting the data into 4D layers $\{\mathcal{H}_1, ..., \mathcal{H}_N\}$, where the boundary of each layer is defined by $\{\Gamma_1,..., \Gamma_N\}$ (the background volume $\mathcal{H}_N$ is assigned the residual regions which do not belong to any other layer).

In this setup, $\mathcal{H}_k$ corresponds to a layer which is defined by a contour on one viewpoint and a disparity as outlined in Section 2.3. However, due to occlusions the complete layer will not be visible in the dataset. Therefore, we define the cost function[d] in terms of the visible regions $\mathcal{H}_k^{\mathcal{V}}$, and this leads to the following:

$$\min_{\{\Gamma_1,...,\Gamma_N,\Delta p_1,...,\Delta p_N\}} \left(\sum_{k=1}^{N} \int_{\mathcal{H}_k^{\mathcal{V}}} d_k(\mathbf{x}, \Delta p_k) d\mathbf{x} \right), \tag{14}$$

where x = $[x, y, V_x, V_y]^T$ and $\mathcal{H}_k^{\mathcal{V}}$ correspond to the visible regions of each layer.

Recall that under our assumptions, the intensity along each EPI line is constant. We therefore choose the descriptor d_k $(\mathbf{x}, \Delta p_k)$

$$d_k(\mathbf{x}, \Delta p_k) = [I(\mathbf{x}) - \mu(\mathbf{x}, \Delta p_k)]^2, \tag{15}$$

where μ $(\mathbf{x}, \Delta p_k)$ is the mean of the EPI line which passes through a point **x** and has a disparity Δp_k.

The aim of the segmentation is then to obtain the layer boundaries Γ_k and the disparity values Δp_k for k = 1,..., N by minimising (14). Observe however that (14) has a large number of unknown variables. In order to minimise the function, we consider the problem of layer evolution and disparity estimation separately and then show how the problem is iteratively solved in Section 3.2.4.

Assuming the layer disparities are known, the minimisation can be simplified to

$$\min_{\{\Gamma_1,...,\Gamma_N\}} \left(\sum_{k=1}^{N} \int_{\mathcal{H}_k^{\mathcal{V}}} d_k(\mathbf{x}, \Delta p_k) d\mathbf{x} \right). \tag{16}$$

One way to minimise (16) is to evolve iteratively the boundary of each layer. For example assuming there are three volumes $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ and we choose to evolve the boundary of the first one, the energy function can be expressed as

$$J_1 = \int_{\mathcal{H}_1^{\mathcal{V}}} d_1(\mathbf{x}, \Delta p_1) d\mathbf{x} + \underbrace{\int_{\mathcal{H}_2^{\mathcal{V}}} d_2(\mathbf{x}, \Delta p_2) d\mathbf{x} + \int_{\mathcal{H}_3^{\mathcal{V}}} d_3(\mathbf{x}, \Delta p_3) d\mathbf{x}}_{\int \overline{H_1^{\mathcal{V}}} d_1^{\mathrm{out}}(\mathbf{x}) d\mathbf{x}} \tag{17}$$

$$= \int_{\mathcal{H}_1^{\mathcal{V}}} d_1(\mathbf{x}, \Delta p_1) d\mathbf{x} + \int_{\overline{\mathcal{H}_1^{\mathcal{V}}}} d_1^{\mathrm{out}}(\mathbf{x}) d\mathbf{x}, \tag{18}$$

where $d_1^{\mathrm{out}}(\mathbf{x}) = d_i(\mathbf{x}, \Delta p_i)$ when $\mathbf{x} \in \mathcal{H}_i^{\mathcal{V}}$ for i = 2, 3.

In general, when evolving the k-th layer, the cost function can be simplified to

$$J_k = \int_{\mathcal{H}_k^{\mathcal{V}}} d_k(\mathbf{x}, \Delta p_k) d\mathbf{x} + \int_{\overline{\mathcal{H}_k^{\mathcal{V}}}} d_k^{\mathrm{out}}(\mathbf{x}) d\mathbf{x}. \tag{19}$$

A possible solution would then be to evaluate the 4D velocity vector of the boundary corresponding to $\mathcal{H}_k^{\mathcal{V}}$ This approach, however, would not explicitly take into account the structure of multiview data in the minimisation. In the following we show how (19) is solved by imposing the camera setup and the occlusion constraints.

3.2.1 Imposing camera setup and occlusion constraints

Recall that the background layer corresponds to the object with the largest depth (smallest disparity). If the boundary of this layer increases, it will automatically be occluded by the remaining layers in the dataset. Therefore, the structure of the visible layers will remain unchanged, and hence the cost must also remain the same. When evolving the k-th layer, we model this by using the following indicator function:

$$\mathcal{I}_k(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \in \left(\bigcup_{j=1}^{k-1} \mathcal{H}_j \right) \\ 1, & \text{otherwise,} \end{cases} \tag{20}$$

where the layers $\{\mathcal{H}_1, ..., \mathcal{H}_N\}$ are ordered in terms of increasing depth. Incorporating this into (18) allows the cost to be expressed in terms of $\mathcal{H}_k$ as follows:

$$J_k = \int_{\mathcal{H}_k} d_k(\mathbf{x}, \Delta p_k)\mathcal{I}_k(\mathbf{x})d\mathbf{x} + \int_{\overline{\mathcal{H}_k}} d_1^{\text{out}}(\mathbf{x})\mathcal{I}_k(\mathbf{x})d\mathbf{x}. \tag{21}$$

Observe that the integration bounds $\mathcal{H}_k$ now correctly correspond to the layer boundary Γ_k . This, therefore, allows the derivative of the cost to be defined as

$$\frac{dJ_k}{d\tau} = \int_{\Gamma_k} [d_k(\mathbf{x}, \Delta p_k)\mathcal{I}_k(\mathbf{x}) - d_k^{\text{out}}(\mathbf{x})\mathcal{I}_k(\mathbf{x})](\mathbf{v}_{\Gamma_k} \bullet \mathrm{n}_{\Gamma_k})d\boldsymbol{\sigma}. \tag{22}$$

Additionally, recall that using the camera setup constraint, the boundary Γ_k can be parameterised by a 2D contour γ_k (s) = $[x$ (s),$y(s)]$ on the image viewpoint (V_x = 0). Substituting this into (22) we obtain

$$\frac{dJ_k}{d\tau} = \int_{\gamma_k} [D_k(s, \Delta p_k) - D_k^{\text{out}}(s)](\mathbf{v}_{\gamma_k} \bullet \mathbf{n}_{\gamma_k})ds, \tag{23}$$

where $\mathbf{v}_{\gamma k}$ and $\mathbf{n}_{\gamma k}$ now correspond to the velocity and the outward normal vector of the 2D boundary[e], respectively. In addition, the new objective functions D_k $(\cdot)$ and $D_k^{\text{out}}(\cdot)$ are defined as

$$D_k(s, \Delta p_k) = \iint d_k(\mathbf{x}, \Delta p_k)\mathcal{I}_k(\mathbf{x})dV_x dV_y \tag{24}$$

$$D_k^{\text{out}}(s) = \iint d_k^{\text{out}}(\mathbf{x})\mathcal{I}_k(\mathbf{x})dV_x dV_y, \tag{25}$$

where

$$\mathbf{x} = \begin{bmatrix} x(s) - \Delta p_k V_x \\ y(s) - \Delta p_k V_y \\ V_x \\ V_y \end{bmatrix}.$$

Note that the new descriptors $D_k^{\text{out}}(\cdot)$ and D_k $(\cdot)$ are simply the descriptors $d_k^{\text{out}}(\cdot)$ and d_k $(\cdot)$ integrated over the viewpoint dimensions.

The velocity vector which reduces the cost in the direction of steepest descent can therefore be chosen as[f]

$$\mathbf{v}_{\gamma k} = [D_k^{\text{out}}(s) - D_k(s, \Delta p_k)]\mathbf{n}_{\gamma k}. \tag{26}$$

There are two main advantages in simplifying the evolution from a 4D to a 2D contour. First, the approach ensures that the layer boundary remains consistent across the views. Secondly, the complexity is reduced from evolving a 4D hypersurface to a 2D contour. We show a comparison between an unconstrained and constrained boundary evolution in Figure 9. Observe that by imposing the camera setup and occlusion constraints in Figure 9b we obtain a segmentation which is consistent with the EPI structure. In conclusion, (26) defines a velocity vector, which evolves the layer boundary γ_k (s) towards the desired segmentation for each layer.

3.2.2 Disparity and number of layers estimation

In the previous section we presented an approach to derive the velocity vector for each layer. However, the knowledge of the disparities is required in order to evaluate the correct evolution. We evaluate these parameters by assuming the 2D layer contours $\{\gamma_1 ..., \gamma_N\}$ are constant. In this case, the objective function can be simplified to:

$$\min_{\{\Delta p_1, \ldots, \Delta p_N\}} \sum_{k=1}^{N} \int_{\mathcal{H}_k^{\mathcal{V}}} d_k(\mathbf{x}, \Delta p_k)d\mathbf{x}. \tag{27}$$

In contrast to the optimisation of the layer contours, this problem is significantly simpler. A solution can be obtained in an iterative approach by estimating the disparity of each layer assuming the remaining disparities

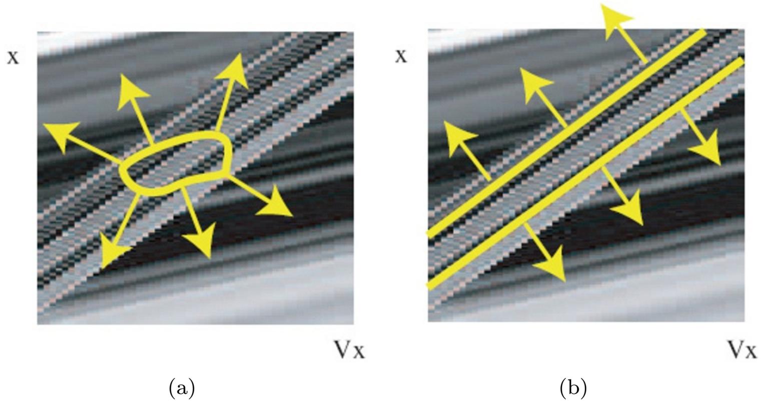


Figure 9 2D EPI volume cross section showing unconstrained and constrained boundary evolution. [34]. **(a)** Unconstrained boundary evolution. **(b)** Constrained boundary evolution. The segmentation is defined using a contour γ(*s*) on image viewpoint (V_x = 0) and a disparity Δ*p*.

are constant. Each parameter is then estimated using the MATLAB nonlinear optimisation toolbox.

In addition, observe that we require the knowledge of the number of layers N. In our approach we initialise this value using a stereo match algorithm [20]. Alternatively, one could estimate the number of layers using the spectral properties of the light field [21] as proposed in [22].

3.2.3 Level-set method for the boundary evolution

We have demonstrated an approach to derive the velocity vector for each boundary. We then implement the evolution of the active contours using the level-set method [23].

This method, instead of evolving directly the 2D boundary, implicitly models the curve using a higher dimensional surface $z = \varphi(x, y, \tau)$. The original boundary is then defined as the zero-level of the new function

$$\begin{aligned}\gamma(s,\tau) &= \arg\{\phi(x,y,\tau)\}\\ &\text{such that } \phi(x,y,\tau) = 0,\end{aligned} \tag{28}$$

where s parameterises the (x, y) coordinates on the 2D boundary.

The evolution equation of the surface can then be derived as follows. First, by implicitly differentiating φ (γ (s, τ), τ) = 0 with respect to τ we obtain

$$\frac{\partial\phi}{\partial\tau} + \frac{\partial\phi}{\partial x}\frac{\partial x}{\partial\tau} + \frac{\partial\phi}{\partial y}\frac{\partial y}{\partial\tau} = 0 \Leftrightarrow \frac{\partial\phi}{\partial\tau} + \nabla\phi(\gamma(s,\tau),\tau)\cdot\frac{\partial\gamma}{\partial\tau} = 0, \tag{29}$$

where ∇ is the gradient operator. Second, observe that the normal to the surface φ (x, y, τ) evaluated on the boundary γ (s, τ) corresponds to the outward normal vector of the boundary $\mathbf{n}_\gamma$. This implies that

$$\frac{\nabla\phi}{|\nabla\phi|} = \mathbf{n}_\gamma. \tag{30}$$

Combining (29), (30) with original boundary model $\frac{\partial\gamma}{\partial\tau} = F\mathbf{n}_\gamma$ we obtain the level-set evolution equation [23]

$$\frac{\partial\phi(x,y,\tau)}{\partial\tau} = -F(x,y)\left|\nabla\phi(x,y,\tau)\right|. \tag{31}$$

There are two main advantages to using the level-set method. First, the surface implicitly models any topological changes of the boundary. Second, unlike other parameterisation schemes, the approach does not suffer instability issues since (31) is evaluated on a fixed cartesian grid.

The evolution of the level-set method does however have a drawback in terms of increased complexity. To evolve the surface, the velocity vector must be evaluated at every position on the grid. In our approach, we deal with this problem by using the narrowband implementation [24], where only a region around the boundary is evolved instead of the complete surface.

3.2.4 Layer segmentation algorithm overview

An overview of the complete layer extraction algorithm is shown in Algorithm 1. First, the 2D contours and the disparity of each layer are initialised using a stereo matching algorithm [20]. The algorithm evaluates the disparity of each layer and then iteratively evolves the boundaries[g] using the proposed velocity vector in (26). This process continues for a certain number of iterations or until the change in the overall cost is below a predefined threshold.

An example of the extracted layers using the method outlined in Algorithm 1 is shown in Figure 1. In addition, in Figure 10 we show a comparison between an initialised layer-boundary using the stereo matching algorithm and the final layer contour.

To obtain a sparse representation, the obtained layers are decomposed using a 4D DWT as explained in the following section.

4 Data decomposition

In this stage, the redundancy of the texture in each layer shown in Figure 1 is reduced using a multi-dimensional wavelet transform. In the following, we present the inter-view and the spatial transforms in more detail.

Algorithm 1 Layer extraction algorithm

```
STEP 1: Initialise the 2D boundary of each layer $\{\gamma_1, \gamma_2,..., \gamma_N\}$ using a stereo matching algorithm (Algorithm [20] in our implementation).
STEP 2: Estimate the disparity of each layer $\{\Delta p_1, \Delta p_2,... \Delta p_N\}$ by minimising the squared error along the EPI lines.
STEP 3: Reorder the layers in terms of increasing depth.
STEP 4: Iteratively evolve the layer boundaries assuming the remaining layers are constant:
for $k$ = 1 to N-1 do
    Evaluate the velocity vector $\mathbf{v}_{\gamma k}$ of the $k$-th layer.
    Evolve the boundary $\gamma_k$ according to the velocity vector.
end for
STEP 5: Return to STEP 2 or exit algorithm if the change in the cost (14) is below a predefined threshold.
```

4.1 Inter-view 2D DWT

We implement the inter-view 2D DWT on each layer in two steps: first by applying a 1D disparity compensated DWT across the row images (V_y) followed by the column images (V_x) as illustrated in Figure 11. The process is iterated on the low-pass components to obtain a multiresolution decomposition.

In our implementation of the 1D DWT we use the disparity compensated Haar transform. This is

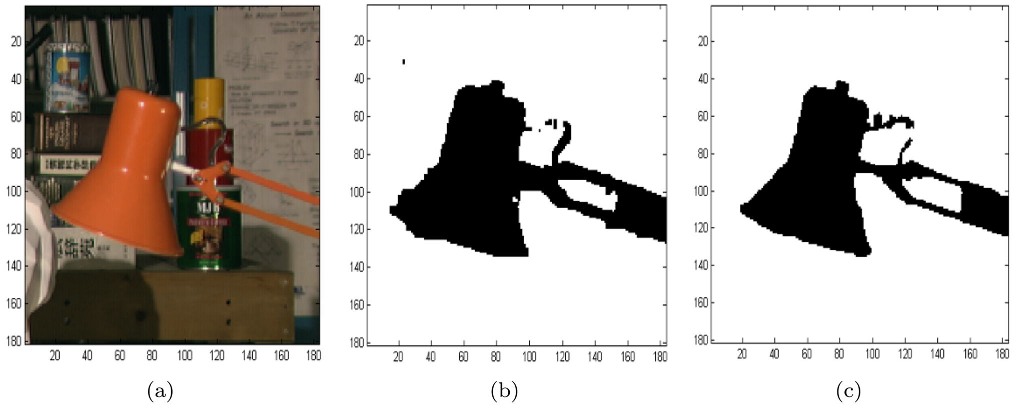


Figure 10 Comparison between an initialised and output layer contour. **(a)** Tsukuba dataset. **(b)** Initialised layer contour using a stereo matching algorithm [20]. **(c)** Layer contour after running Algorithm 1. The extraction algorithm improves the accuracy of the layer-based representation.

motivated by the fact that the light field intensity along the EPI lines is constant. Therefore, a wavelet with one vanishing moment is enough to obtain a sparse representation. It is applied by modifying the standard lifting equations [17] and including a warping operator $\mathcal{W}$ as follows:

$$\begin{aligned}\mathscr{L}_o[n] &= \frac{P_o[n] - \mathcal{W}\{P_e[n]\}}{2},\\ \mathscr{L}_e[n] &= P_e[n] + \mathcal{W}\{\mathscr{L}_o[n]\},\end{aligned}\tag{32}$$

where, P_o $[n]$ and P_e $[n]$ represent 2D images with spatial coordinates (x, y) located at odd $(2n + 1)$ and even $(2n)$ camera locations, respectively. Following (32), $\mathscr{L}_e[n]$ contains the 2D low-pass subband and $\mathscr{L}_o[n]$ the high-pass subband. Assuming that $\mathcal{W}$ is invertible and the images are spatially continuous, the above transform can be shown to be equivalent to the standard DWT applied along the motion trajectories [25].

In both the prediction and update steps in (32), the warping operator $\mathcal{W}$ is chosen to maximise the inter-image correlation. This is achieved by using a projective operation that maps one image onto the same viewpoint as its odd/even complement in the lifting step. Using (4) and the fact that the layers are modelled by a constant disparity, we define the warping operation from viewpoint n_1 to n_2 along the V_x dimension as:

$$\mathcal{W}_{n_1 \to n_2}\{P[n_1]\}(x, y) = P[n_1](x + \Delta p(n_2 - n_1), y), \tag{33}$$

where Δp is the layer disparity.

Note that in the case of an occlusion, the DWT leads to filtering across an artificial boundary and, thus, results in a reduced sparsity efficiency. To prevent this, we use the concept proposed in [26] to create a shape-adaptive transform in the view domain. The transform in (32) is modified whenever a pixel at an even or odd location is occluded such that

$$\mathscr{L}_e[n] = \begin{cases} P_e[n], & \text{occlusion at } 2n + 1 \\ \widehat{\mathcal{W}}\{P_o[n]\}, & \text{occlusion at } 2n \end{cases}, \tag{34}$$

and the high pass coefficient in $\mathcal{L}_o[n]$ is set to zero. In (34), the warping operator $\widehat{\mathcal{W}}$ is set to an integer pixel precision to ensure invertibility and is set to be the *ceiling* of the disparity in (33).

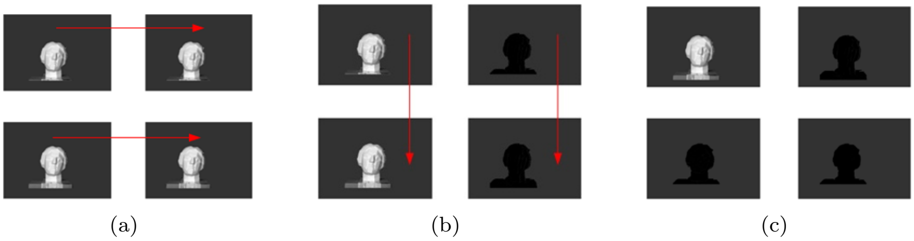


Figure 11 Inter-view 2D DWT is implemented in a separable approach by filtering the image rows followed by the image columns by using the 1D disparity compensated DWT. The red arrow shows the direction of the 1D DWT. **(a)** Extracted layer: 2 × 2 light field. **(b)** Transform coefficients following 1D disparity compensated DWT across each row. **(c)** Transform coefficients following 1D disparity compensated DWT across each column. Note that the background has been labeled grey and is outside the boundary of the layer.

4.2 Spatial shape-adaptive 2D DWT

Following the inter-view transform we reduce the intra-view redundancy by using a 2D DWT. However, prior to applying the 2D DWT on each image, we recombine the transform coefficients into a single layer. This is done to increase the number of decompositions which can be applied by the spatial transform. A comparison between the original and recombined layers is illustrated in Figure 12. Note that due to occlusions and the way in which the inter-view transform is implemented, two or more layers may overlap in each subband. In this case, we apply a separate spatial transform to the overlapped pixels.

Note that the overlapped pixels are commonly bounded by an irregular (non-rectangular) shape. For that reason, the standard 2D DWT applied to the entire spatial domain is inefficient due to the boundary effect. We therefore use the shape-adaptive DWT [26] within arbitrarily shaped objects. The method reduces the magnitude of the high pass coefficients by symmetrically extending the texture whenever the wavelet filter is crossing the boundary. The 2D DWT is built as a separable transform with linear-phase symmetric wavelet filters (9/7 or 5/3 [27]), which, together with the symmetric signal extensions, leads to critically sampled transform subbands.

5 Evaluation

In this section we evaluate the performance of the proposed sparse representation using its nonlinear approximation properties. In addition, we demonstrate de-noising and IBR applications based on the proposed decomposition.

5.1 Nonlinear approximation

We evaluate the sparseness of the representation using its N-term nonlinear approximation. To implement the nonlinear approximation, we keep the N largest coefficients in the transform domain, reconstruct the data and evaluate the data fidelity in terms of PSNR.

Our results show that the proposed layer-based representation offers superior approximation properties compared to a typical multi-dimensional DWT[h]. We demonstrate this in Figure 13 on three datasets: Tsukuba light field [272 × 368 × 4 × 4], Teddy EPI [368 × 352 × 4] and Doll EPI [368 × 352 × 4] (all from [28]), which vary in terms of scene complexity, number of images and spatial resolution. We show that in each case our approach achieves a sparser representation across the complete range of retained coefficients, with PSNR gains of up to 7 dB on the Tsukuba light field. The Tsukuba light field has a larger PSNR improvement than the respective Teddy and Doll EPI volumes due to the additional viewing dimension. This means that there exists more redundant information and this is fully exploited by our representation. We also show that the PSNR curves correspond to a subjective improvement in Figure 14.

We note that the nonlinear approximation metric is also a good indicator of the compression capability of the representation. In practice, the issue of compression is more complicated due to the additional problem of encoding the locations of the significant coefficients and also to the rate allocation. These issues are beyond the scope of this paper, however, we refer the reader to [29] where these problems are addressed and a complete multiview image compression method is presented.

5.2 De-noising

Here we demonstrate de-noising results based on the proposed sparse representation in the presence of additive white Gaussian noise (AWGN). We implement the de-noising by soft thresholding the wavelet coefficients in each subband. For each subband, the threshold is

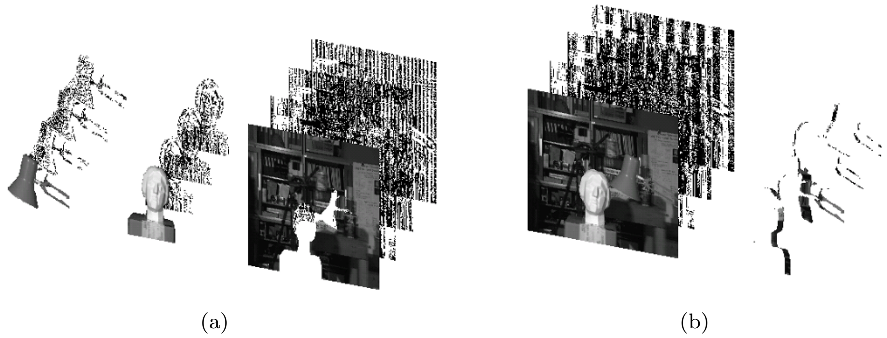
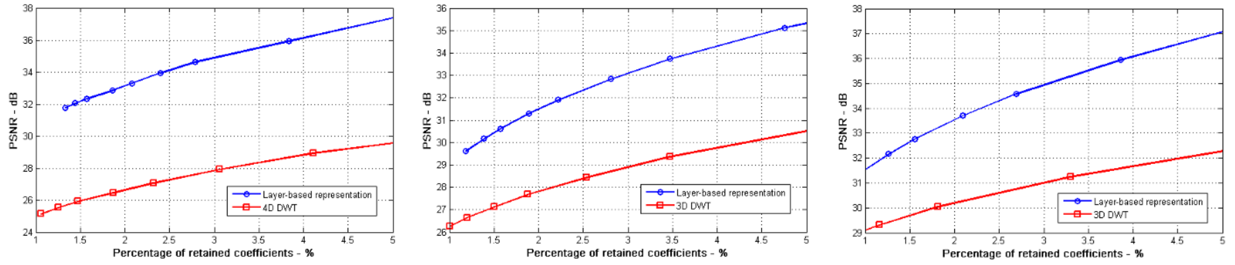


Figure 12 Merging of the inter-view transform coefficients. **(a)** Tsukuba transform coefficients following the inter-view transform. Each of the three transformed layers is composed of one low-pass subband and three high frequency images. **(b)** Recombined layers. The view subbands from each layer are grouped into a single image to increase the number of decompositions that can be applied by the spatial transform. In each subband two or more layers may overlap. We apply a separate shape-adaptive 2D DWT to the overlapped pixels.



(a) Tsukuba light field. (b) Teddy EPI. (c) Doll EPI.

Figure 13 N-term nonlinear approximation of the layer-based representation in comparison to a standard multi-dimensional DWT. The percentage of retained coefficients is evaluated as $\frac{100N_r}{N_d}$, where N_r is the total number of retained coefficients and N_d is the total number of pixels in the dataset. **(a)** Tsukuba light field. **(b)** Teddy EPI. **(c)** Doll EPI.

chosen by minimising the Stein's Unbiased Risk Estimate (SURE) of the mean squared error (MSE) [30].

Note that the aim of this section is not to compare the results to the state-of-the-art in multiview de-noising techniques but to demonstrate that the sparse representation can be used for de-noising applications. In Figure 15 we compare our algorithm to the competitive SURE-LET OWT de-noising method [31] applied to each image independently. In this setup, we assume the noise is added to the extracted layers. Analysing the Tsukuba light field, Teddy EPI and Doll EPI datasets, our approach corresponds to a PSNR improvement of up to 2 dB. The light field has the most significant gain due to a sparser representation, which results from the larger number of images in the dataset.

The subjective results are illustrated in Figure 16 and these clearly show that the proposed sparse representation attains more visually pleasing results than the SURE-LET OWT method.

5.3 Image-based rendering

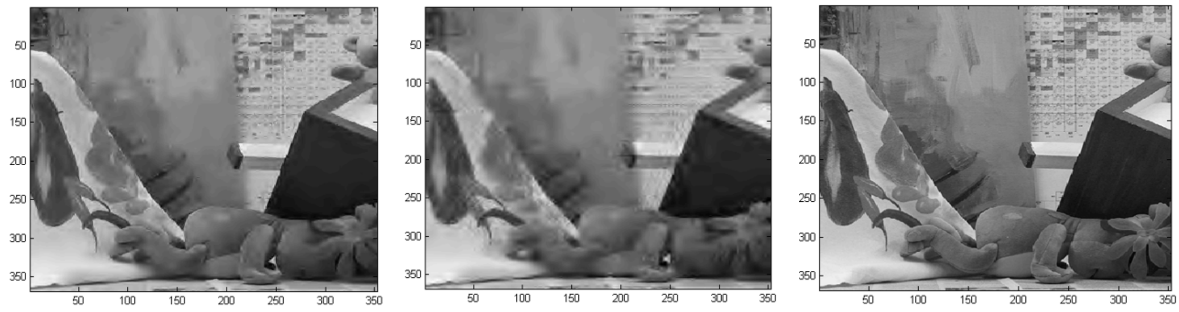
In this section we present viewpoint interpolation results based on the layer-based representation shown in Figure 1.

To render an image at an arbitrary viewpoint, we linearly interpolate the closest available images. Recall that the data pixels are highly correlated in the direction of the disparity. We take this into account by modifying the support of the rendering kernel according to the disparity of each layer. Additionally, we modify the interpolation in the presence of occlusions to further improve the results. In this case, only pixels that belong to the layer are used in the rendering process.

In order to obtain an objective evaluation, we use the leave-one-out approach. In this case the images located at odd camera viewpoint locations are removed and synthesised using the scene model[i].

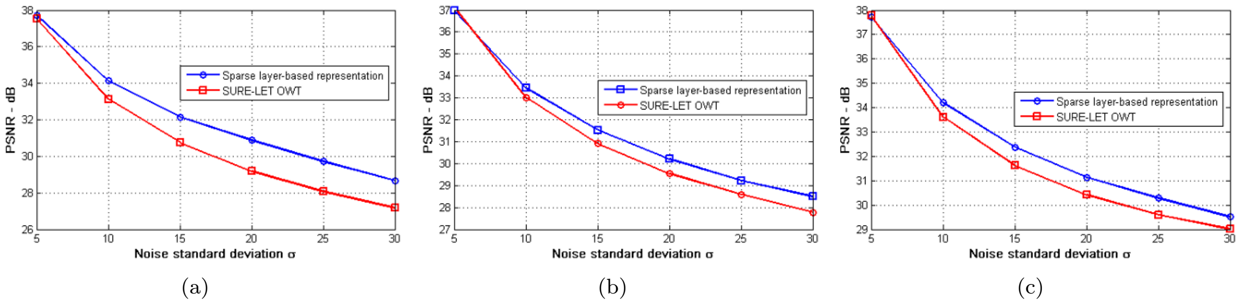
We compare our results to a state-of-the-art stereo matching algorithm [32] and an EPI tubes extraction method [33]. These methods specify the structure of the EPI lines, and the interpolation is implemented using the same approach as in the proposed algorithm.

In Table 1 we show a comparison on four datasets: Dwarves EPI [555 × 695 × 7] [28], Lobby EPI [800 × 800 × 5], Desk light field [500 × 500 × 4 × 4] and Animal Farm EPI [235 × 625 × 32 (last three from [34]). Observe that the layer-based representation achieves an



(a) Proposed method - PSNR 29.62dB with 1.18% of coefficients retained. (b) Standard 3D DWT - PSNR 26.6dB with 1.20% of coefficients retained. (c) Original dataset.

Figure 14 Nonlinear approximation subjective evaluation. **(a)** Sparse layer-based representation (PSNR 29.62dB) with 1.18% of coefficients retained. **(b)** Standard 3D DWT (PSNR 26.6dB) with 1.20% of coefficients retained. **(c)** Original Teddy EPI dataset.



(a)

(b)

(c)

Figure 15 De-noising comparison between the proposed sparse representation and the SURE-LET OWT method [31]. In the proposed approach, the de-noising is implemented by soft thresholding the transform coefficients in each subband. The threshold step-size is chosen by minimising the SURE estimate of the MSE. **(a)** Tsukuba light field. **(b)** Teddy EPI. **(c)** Doll EPI.

improved SNR in comparison to both the stereo matching and the EPI tubes extraction method on all datasets.

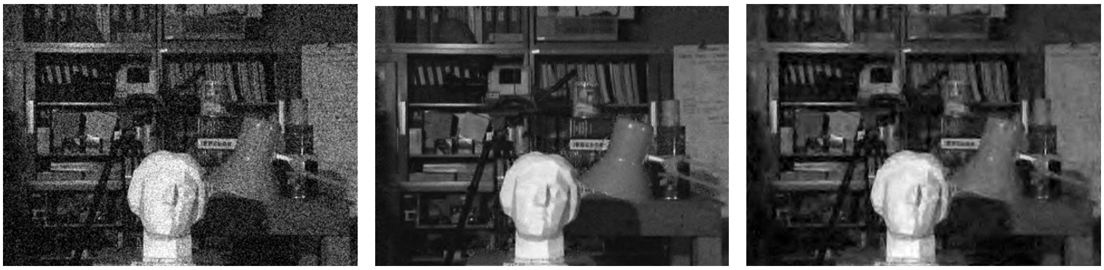
In addition to the quantitative evaluation, we show a subjective comparison in Figure 17. This shows that interpolation using the proposed layer-based representation achieves significantly improved results. We note that using our method we obtain artifact free and photo-realistic images. In comparison, aliasing artifacts are present in the stereo matching and EPI tube extraction methods. This is due to incorrect compensation of the interpolation kernel, which stems from inaccurate depth correspondence in the scene.

6 Conclusion

We presented a novel method to obtain a sparse representation of multiview images. The fundamental component of the algorithm is the layer-based representation, which partitions the multiview images into a set of layers each related to a constant depth in the scene. We presented a novel method to obtain the layer-based representation using a general segmentation framework which takes into account the structure of multiview data to achieve accurate results. The obtained layers are then decomposed using a 4D DWT applied in a separable approach, first across the camera viewpoint and then the image dimensions. We modify the viewpoint transform to efficiently deal with occlusions and depth variations. Simulation results based on nonlinear approximation have shown that the sparsity of our representation is superior to a multi-dimensional DWT with the same decomposition structure without disparity compensation. In addition, we have shown that the proposed representation can be used to efficiently synthesise novel viewpoints for IBR applications and also de-noise multiview images in the presence of AWGN.

Endnotes

[a]Each camera in the setup is modelled by the pinhole model [35]. [b]Light ray intensity is constant when an object is observed from a different angle. [c]By visible regions we mean the EPI line segments which are present in the EPI volume. [d]We have not included the regularisation terms for the sake of clarity. [e]It can be shown that given a fronto-parallel depth plane, the inner product of $\mathbf{v}_{\gamma k} \cdot \mathrm{n}_{\gamma k}$ is equal to $\mathbf{v}_{\Gamma k} \cdot \mathrm{n}_{\Gamma k}$. [f]Note that in practise we also include a regularisation term to constrain the evolution according to the curvature of the boundary. [g]Note that the background layer $\mathcal{H}_N^{\mathcal{V}}$ is



(a) Input (PSNR 18.6 dB).

(b) Proposed sparse representation (28.65 dB).

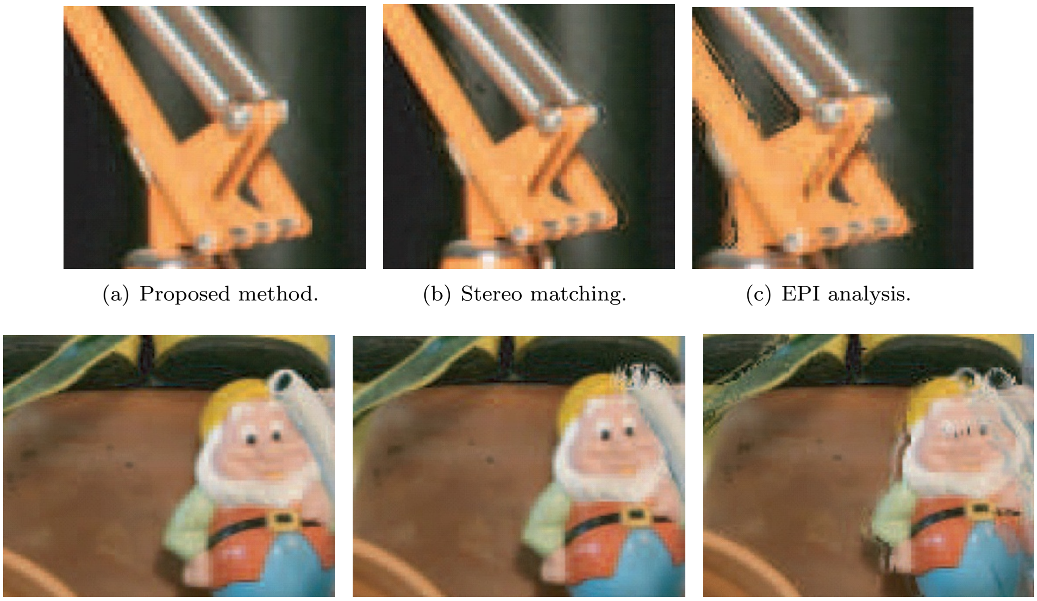
(c) SURE-LET OWT (PSNR 27.16 dB)

Figure 16 De-noising subjective evaluation. **(a)** Tsukuba light field corrupted with AWGN (PSNR 18.6 dB). **(b)** De-noised dataset using proposed sparse representation (PSNR 28.65 dB). **(c)** De-noised dataset using SURE-LET OWT applied to each image independently (PSNR 27.16 dB).

Table 1 Image-based rendering evaluation

Dataset	Proposed method (dB)	Stereo (dB) [32]	EPI analysis (dB) [33]
Dwarves EPI [28]	**27.08**	26.24	19.56
Lobby EPI [34]	**17.65**	17.21	11.69
Desk light field [34]	**25.57**	25.46	19.79
Animal farm EPI [34]	**23.04**	17.47	19.02

The proposed layer-based representation is compared to a state-of-the-art stereo matching algorithm [32] and an EPI tubes extraction method [33]. The results are obtained using the leave-one-out scenario. In this case the odd images are removed and then synthesised using the available data. The image fidelity is expressed in terms of SNR, which is evaluated in terms of the ground truth and the rendered image. The bold values indicate the results with the highest SNR.



(a) Proposed method. (b) Stereo matching. (c) EPI analysis.

(d) Proposed method. (e) Stereo matching. (f) EPI analysis.

Figure 17 IBR subjective evaluation. **(a-c)** Desk light field interpolation using **(a)** proposed layer-based representation, **(b)** stereo matching algorithm [32] and **(c)** EPI tubes extraction method [33]. **(d-f)** Dwarves EPI interpolated images in the same order. Note that the layer extraction algorithm has been implemented on the grayscale images. The obtained layers were then used to interpolate each of the RGB channels to show colour results.

automatically assigned all of the regions which do not belong to the remaining layers and is therefore not evolved. [h]This multi-dimensional DWT has the same decomposition structure as our method, however no disparity compensation. iThe extracted layers are obtained using the dataset with the removed images.

Acknowledgements
We would like to thank the anonymous reviewers whose constructive comments led to an improved manuscript.

Author details
[1]Communications and Signal Processing Group, Imperial College London, London SW7 2AZ, UK [2]Google Inc., Brandschenkestrasse 110, Zurich 8002, Switzerland

Competing interests
The authors declare that they have no competing interests.

Received: 14 July 2011 Accepted: 9 March 2012
Published: 9 March 2012

References

1. D Taubman, M Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, (Kluwer Academic Publishers, Boston, 2004)
2. EJ Candés, DL Donoho, *Curvelets: a surprisingly effective nonadaptive representation of objects with edges*, (Curve and Surface Fitting, University Press, Saint-Malo, 2000)
3. MN Do, M Vetterli, The contourlet transform: an efficient directional multiresolution image representation. IEEE Trans Image Process. **14**(12), 2091–2106 (2005)
4. MN Do, M Vetterli, The finite ridgelet transform for image representation. IEEE Trans Image Process. **12**(1), 16–28 (2003). doi:10.1109/TIP.2002.806252
5. V Velisavljevic, B Beferull-Lozano, M Vetterli, PL Dragotti, Directionlets: anisotropic multidirectional representation with separable filtering. IEEE Trans Image Process. **5**(7), 1916–1933 (2006)
6. E Le Pennec, S Mallat, Sparse geometric image representations with bandelets. IEEE Trans Image Process. **14**(4), 423–438 (2005)
7. E Le Pennec, S Mallat, Bandelet image approximation and compression. SIAM J Multiscale Model Simul. **4**(3), 992–1039 (2005). doi:10.1137/040619454

8. I Bayram, IW Selesnick, On the dual-tree complex wavelet packet and *M*-band transforms. IEEE Trans Signal Process. **56**(6), 2298–2310 (2008)
9. IW Selesnick, RG Baraniuk, NG Kingsbury, The dual-tree complex wavelet transform. IEEE Signal Process Mag. **22**(6), 123–151 (2005)
10. AM Bruckstein, DL Donoho, M Elad, From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. **51**, 34–81 (2009). doi:10.1137/060657704
11. MN Do, QH Nguyen, HT Nguyen, D Kubacki, SJ Patel, Immersive visual communication. IEEE Signal Process Mag. **28**(1), 58–66 (2011)
12. A Kubota, A Smolic, M Magnor, M Tanimoto, T Chen, C Zhang, Multiview imaging and 3DTV. IEEE Signal Process Mag. **24**(6), 10–21 (2007)
13. C Zhang, T Chen, A survey on image-based rendering-representation, sampling and compression. Signal Process Image Commun. **19**, 1–28 (2004). doi:10.1016/j.image.2003.07.001
14. EH Adelson, JR Bergen, *The Plenoptic Function and the Elements of Early Vision Computational Models of Visual Processing*, (MIT Press, Cambrige, 1991), pp. 3–20
15. M Levoy, P Hanrahan, Light field rendering, in *Proceedings of Computer Graphics (SIGGRAPH)*, New Orleans, Louisiana, pp. 31–42 (1996)
16. R Bolles, H Baker, D Marimont, Epipolar-plane image analysis: an approach to determining structure from motion. Int J Comput Vis. **1**(1), 7–55 (1987). doi:10.1007/BF00128525
17. I Daubechies, W Sweldens, Factoring wavelet transforms into lifting steps. J Fourier Anal Appl. **4**(3), 247–269 (1998). doi:10.1007/BF02476026
18. S Jehan-Besson, M Barlaud, G Aubert, DREAM2S: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. Int J Comput Vis. **53**, 45–70 (2003). doi:10.1023/A:1023031708305
19. M Kass, A Witkin, D Terzopoulos, Snakes: Active contour models. Int J Comput Vis. **1**(4), 321–331 (1988). doi:10.1007/BF00133570
20. V Kolmogorov, R Zabih, Multi-camera scene reconstruction via graph cuts, in *Proceedings of the 7th European Conference on Computer Vision-Part III (ECCV)*, Springer-Verlag, Copenhagen, Denmark, pp. 82–96 (2002)
21. JX Chai, X Tong, SC Chan, HY Shum, Plenoptic sampling, in *Proceedings of Computer Graphics (SIGGRAPH)*, ACM Press/Addison-Wesley Publishing Co., New York, pp. 307–318 (2000)
22. J Berent, PL Dragotti, M Brookes, Adaptive layer extraction for image based rendering, in *Proceedings of IEEE Workshop on Multimedia Signal Processing (MMSP)*, Rio De Janeiro, Brazil, pp. 266–271 (2009)
23. JA Sethian, *Level Set Methods*, (Cambridge University Press, Cambridge, 1996)
24. M Hötter, Object-oriented analysis-synthesis coding based on moving two-dimensional objects. Signal Process Image Commun. **2**(4), 409–428 (1990). doi:10.1016/0923-5965(90)90027-F
25. A Secker, D Taubman, Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression. IEEE Trans Image Process. **12**(12), 1530–1542 (2003). doi:10.1109/TIP.2003.819433
26. S Li, W Li, Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. IEEE Trans Circ Syst Video Technol. **10**(5), 725–743 (2000). doi:10.1109/76.856450
27. M Unser, T Blu, Mathematical properties of the JPEG2000 wavelet filters. IEEE Trans Image Process. **12**(9), 1080–1090 (2003). doi:10.1109/TIP.2003.812329
28. D Scharstein, R Szeliski, Middlebury datasets http://www.vision.middlebury.edu/stereo/data
29. A Gelman, PL Dragotti, V Velisavljevi'c, Multiview image compression using a layer-based representation, in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Hong Kong, China, pp. 13–16 (2010)
30. D Donoho, IM Johnstone, Adapting to unknown smoothness via wavelet shrinkage. J Am Stat Assoc. **90**, 1200–1224 (1995). doi:10.2307/2291512
31. T Blu, F Luisier, The SURE-LET approach to image denoising. IEEE Trans Image Process. **16**(11), 2778–2786 (2007)
32. AS Ogale, Y Aloimonos, Shape and the stereo correspondence problem. Int J Comput Vis. **65**(3), 147–162 (2005). doi:10.1007/s11263-005-3672-3
33. A Criminisi, SB Kang, R Swaminathan, R Szeliski, P Anandan, *Extracting layers and analyzing their specular properties using epipolar-plane-image analysis*, (Microsoft Research Technical Report MSR-TR-2002-19, 2002)
34. J Berent, in Coherent multi-dimensional segmentation of multi-view images using a vari-ational framework and applications to image based rendering, (PhD Thesis, Imperial College, 2008) http://www.commsp.ee.ic.ac.uk/~pld/group/Thesis_Berent_08.pdf
35. RI Hartley, A Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn. (Cambridge University Press, Cambridge, 2004). ISBN:0521540518
36. HY Shum, SB Kang, A review of image-based rendering techniques. IEEE SPIE Vis Commun Image Process (VCIP). **213**, 1–12 (2000)

doi:10.1186/1687-6180-2012-61
Cite this article as: Gelman *et al.*: **Layer-based sparse representation of multiview images.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:61.