Jesse Berent and Pier Luigi Dragotti

Plenoptic Manifolds

Exploiting structure and coherence in multiview images

The availability of multiple views of a scene makes possible new and exciting applications ranging from 3-D and free-viewpoint television to robust scene interpretation and object tracking. The hardware for multicamera systems is developing fast and is already being deployed for multimedia, security, and industrial applications. However, there are still some challenging issues in terms of processing, primarily due to the sheer amount of data involved when the number of cameras becomes very large. It is therefore a primordial point to understand how the information is structured and how to take advantage of the inherent redundancy that results when the cameras are looking at the same scene.

This article provides insights on the nature of the data in multiview imaging systems, particularly in terms of structure and coherence. Using this structure, we derive a multidimensional variational framework for the extraction of coherent regions and occlusion boundaries, which is an important issue in numerous multiview image processing applications such as view interpolation, compression, and scene understanding.

SEEING IN SEVEN DIMENSIONS

Our visual perception sense (i.e., our eyes) enables us to view the world in three dimensions. One might also say that time is a fourth dimension we are able to perceive. One way to understand why this is the case is to say that an eye captures two spatial dimensions describing where

Digital Object Identifier 10.1109/MSP.2007.905883

1053-5888/07/$25.00©2007IEEE

it is looking and another dimension for time. Our second eye provides the fourth dimension for the location of the viewing point. In the case of a camera array however, the number of "eyes" and their position is unlimited.

Studying the data in multiview camera systems from an image processing point of view means adding more degrees of freedom to the problem, and this leads to new difficulties, not the least that the data have more dimensions than most of us are able to visualize. In fact, the number of dimensions goes up to seven when all the degrees of freedom are taken into account. Indeed, the visual information captured depends on the viewing position (V_x, V_y, V_z), the viewing direction (θ, ϕ), the wavelength λ and the time t if dynamic scenes are considered. In [1], Adelson and Bergen gather all these parameters into a single function $P = P_7(\theta, \phi, \lambda, t, V_x, V_y, V_z)$ called the *plenoptic function*. Usually, it is represented with the cartesian coordinates used in numerous computer vision and image processing algorithms. It therefore becomes

$$P = P_7(x, y, \lambda, t, V_x, V_y, V_z)\,, \tag{1}$$

where x and y are analogous to the coordinates on the image plane.

It is far from trivial to deal with the seven dimensions of the plenoptic function. However, there are some solutions to overcome this obstacle. First, several assumptions can be made to reduce the dimensionality. As we will see in the next section, these assumptions include dropping the wavelength, considering static scenes, or constraining the camera locations. Second, the plenoptic function in all its parameterizations has a high degree of regularity under the assumption of photoconsistency. Clearly, the analysis of multiview images calls for multidimensional signal processing algorithms that take advantage of this inherent regularity for compression, interpolation, and interpretation.

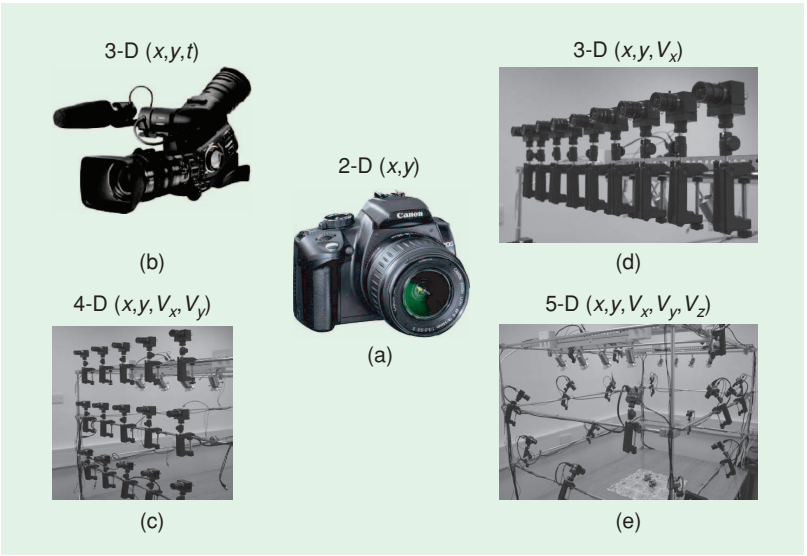
In order to understand the regularity involved in plenoptic data, consider the more easily visualized case of the video. Looking at the full space-time volume such as the one illustrated in Figure 3(a) reveals that an object or a layer moving around in the scene carves out a 3-D volume or an *object tunnel* [2]. Since this volume is constructed with images of the same object, the information inside it is highly regular. Similarly, consider a dense set of multibaseline stereo images that are collated such that they form a 3-D data set (also known as the epipolar plane image (EPI) volume [3]). Again, looking at the whole volume of data, such as that represented in Figure 3(b), shows that *EPI-tubes* [4] are carved out by objects at different depths in very much the same way as in the video. The purpose of this article is to emphasize that, just like the tunnels carved out by objects in space-time or the tubes in EPI volumes, hypervolumes are carved out in the plenoptic function.

In an effort to generalize the notion and inspired by Adelson and Bergen [1], we introduce, with a slight abuse of terminology, the concept of *plenoptic manifolds*. (See the section "Capturing the Plenoptic Function.") With the term plenoptic manifold, we mean the hypervolume carved out by an object in the plenoptic domain. Since these manifolds capture the coherence of the plenoptic function in all dimensions, their extraction is a very useful step in numerous multiview imaging applications such as layer-based representations [5], [6], MPEG-4–like object-based coding [7], disparity-compensated and shape-adaptive wavelet coding [8], and image-based rendering (IBR) [9], [10], especially in the case of occlusions and large depth variations. Other applications include scene interpretation and understanding [11]. All these applications make it very attractive to develop methods that are able to extract such manifolds.

In this article, we go through some common parameterizations of the plenoptic function and recall the shape constraints imposed on the plenoptic manifolds in some simple camera setups. Then, we focus mainly on the light field parameterization [12] and derive a global multidimensional variational framework based on [13], [14] for the extraction of these plenoptic manifolds. Finally, we demonstrate some experimental results and applications in IBR.

CAPTURING THE PLENOPTIC FUNCTION

The plenoptic function was introduced by Adelson and Bergen [1] in order to describe the visual information available from any point space. It is characterized by seven dimensions, namely, the viewing position (V_x, V_y, V_z), the viewing direction (x, y), the time t and the wavelength λ. Usually the wavelength is omitted by considering separate channels for color images or one channel for grayscale images. There are many different ways to capture the plenoptic function and most of the popular sensing devices, some of which are illustrated in Figure 1, do



[FIG1] Capturing the plenoptic function. From the still image camera to the video camera or multiview imaging systems, all the sensing devices illustrated sample the plenoptic function with a varying number of degrees of freedom.

not necessarily sample all the dimensions. The still image camera, for instance, fixes the viewing point and the time. Only the (x, y) dimensions remain. The video camera is able to capture images at different times and therefore captures the (x, y, t) dimensions. Another case of a three-dimensional plenoptic function can be obtained by giving one degree of freedom to the camera location such that (x, y, V_x) is sampled. Higher-dimensional cases add more degrees of freedom to the viewing position such that (x, y, V_x, V_y) or even (x, y, V_x, V_y, V_z) can be captured.

PLENOPTIC TRAJECTORIES AND ASSOCIATED MANIFOLDS

Given a known camera setup, a point in space is projected onto the images in a particular fashion dictated by the geometry of the array of cameras and the movement of the objects. The captured plenoptic function therefore has a structure that depends on the camera setup sampling it. In this section, we illustrate more precisely the properties of the plenoptic function for some simple camera setups starting from the most basic and going on to the higher-dimensional cases (see also the section "Capturing the Plenoptic Function").

Before going through some common representations of the plenoptic function and their associated plenoptic manifolds, we lay out the assumptions made in this article. First, we assume that the cameras follow the basic pinhole camera model as illustrated in Figure 2, and we will use the cartesian coordinate system for the plenoptic function as in (1). Second, the wavelength parameter λ is dropped by considering grayscale images or separate red, green, and blue channels for color data. Finally, we assume that the scenes are made of opaque Lambertian surfaces.

SINGLE-VIEW CAMERAS

A single still image camera samples the plenoptic function where the viewing position and the time are fixed (e.g., in $V_x = V_y = V_z = t = 0$). Only the x and y dimensions remain, which are the image coordinates. The pinhole camera model says that points in the world coordinates $\vec{X} = (X, Y, Z)$ are mapped onto the image plane (x, y) in the point where the line connecting $\vec{X}$ and the camera center intersects with the image plane [15]. The focal length f measures the distance separating the camera center and the image plane. By using similar triangles, it can be shown that the mapping is given by

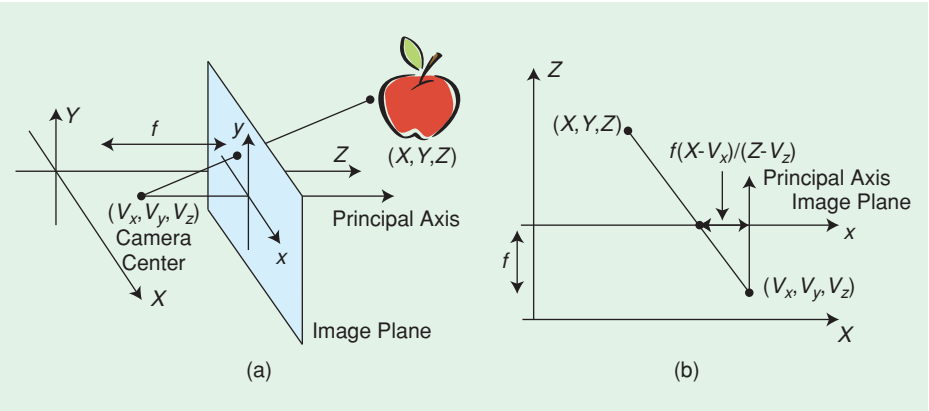
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix},$$

where we assume that the focal length f is unity and that the principal point is located at the origin. The extraction of coherent regions in this case can based on color or texture (i.e., strictly spatial coherence in the x and y dimensions), and, although this problem is extremely interesting, it is not the point we wish to put forward in this article. Rather, we wish to portray the coherence involved when several images of the same objects at different locations or different times are available.

A single viewpoint imaging system can sample a 3-D plenoptic function if it is able to capture the scene at different times. This is the case of the video or moving image. The point in space $\vec{X}$ is free to move in time and its mapping onto the video data becomes

$$\begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ t \end{pmatrix} = \begin{pmatrix} X(t)/Z(t) \\ Y(t)/Z(t) \\ t \end{pmatrix}$$

which is the parameterization of a trajectory in the 3D plenoptic domain. Note that the intensity along the trajectory remains fairly constant if the radiance of the point does not change in time. Furthermore, assuming the scene is made of moving objects, neighboring points in space will generate similar neighboring trajectories in the video data. Hence, apart from the object boundaries, the information captured varies mainly in a smooth fashion. This observation has motivated the segmentation of videos into coherent regions such as the ones undergoing similar motion. Recent methods for segmentation and motion estimation have shown that added robustness is achieved by considering the whole space-time volume as opposed to one or a few consecutive frames (see [11] for a recent presentation of space-time video analysis). The analysis of video as a 3-D function enables to impose coherence throughout the stack of images and gain insights on long term effects such as occlusions. In [2], Ristivojevic and Konrad show that tunnels are carved out by objects in the data as illustrated in Figure 3(a). Note that in the general case, the volumes do not have much



[FIG2] The pinhole camera model. Light rays from points in real-world coordinates $\vec{X} = (X, Y, Z)$ generate intensities on the image plane in the point (x, y) where the line connecting $\vec{X}$ and the camera center (V_x, V_y, V_z) intersects the image plane. The focal length f measures the distance separating the camera center and the image plane.

structure. Indeed, there is no real prior constraining the shape of the tunnel carved out unless some assumptions are made on the movement and the rigidity of the objects. Nevertheless, in natural videos, assuming a certain degree of smoothness and temporal coherence is usually a valid assumption.

LINEAR MULTIVIEW CAMERA ARRAYS

A linear camera array gives one degree of freedom to the position of the camera (e.g., in V_x). That is, parallax information is available. In this case, the plenoptic function reduces to the Epipolar plane image (EPI) volume first introduced by Bolles et al. [3]. It can be acquired either by translating a camera along a rail or by a linear camera array (such as the one illustrated in Figure 1). According to the pinhole camera model and assuming $V_y = V_z = t = 0$, points in real-world coordinates are mapped onto the EPI volume as a function of V_x according to

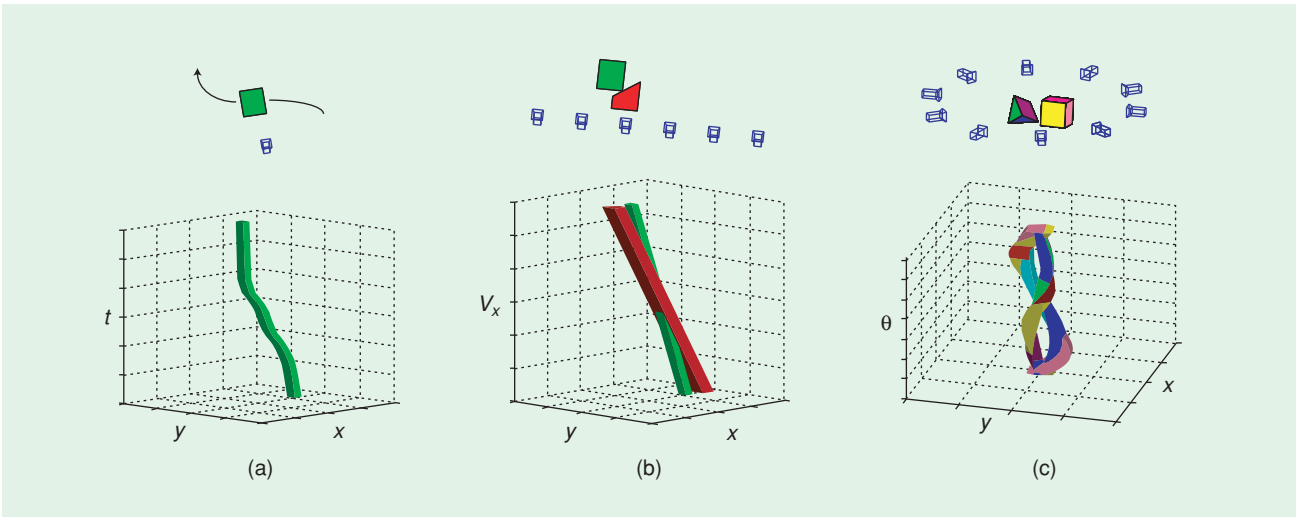
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ V_x \end{pmatrix} = \begin{pmatrix} X/Z - V_x/Z \\ Y/Z \\ V_x \end{pmatrix},$$

where we notice that a point in space generates a line. Furthermore, the slope of the line is inversely proportional to the depth of the point Z. Therefore, the data in this parameterization, as opposed to the video, have a very particular structure, which is noticeable in Figure 3(b). The occurrence of occlusions, for example, is predictable since a line with a larger slope will always occlude a line with a smaller slope. This property follows naturally from the fact that points closer to the image plane will occlude points that are further away. The example illustrated in Figure 4 portrays this property with natural images.

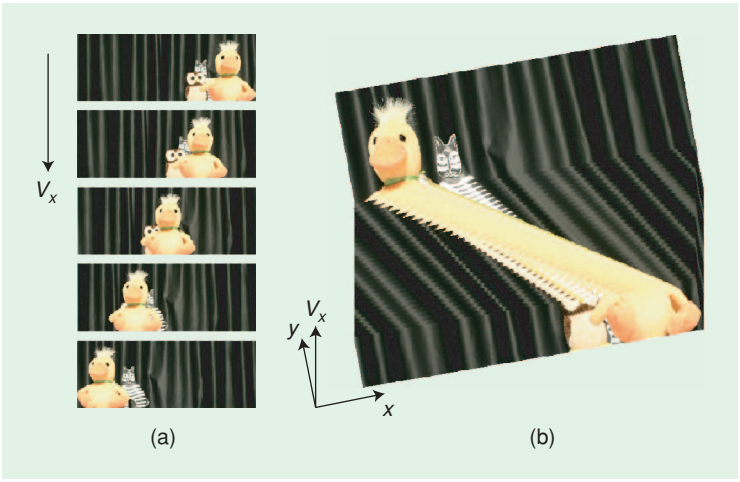
In the original EPI analysis paper [3], this particular structure is used to infer depth information in a scene by finding the slopes of the lines in the EPI volume. It is emphasized that by looking at the problem in this manner, all the images are taken into account simultaneously. However, the problem of dense segmentation was not dealt with. This problem was studied much later by Criminisi et al. [4] where horizontal slices of the EPI volume are analyzed in order to gather lines with similar slopes. This segmentation generates coherent volumes that are called EPI-tubes for their obvious tube-like appearance (see Figure 3(b)). As opposed to the method in [4], which analyses the data slice by slice, the method presented in [13], [14], which we describe in more detail below in "Extracting Plenoptic Manifolds in Multiview Data," exploits coherence in the three dimensions; that is, the whole stack of images is analyzed in a global manner.

THIS ARTICLE PROVIDES INSIGHTS ON THE NATURE OF THE DATA IN MULTIVIEW IMAGING SYSTEMS, PARTICULARLY IN TERMS OF STRUCTURE AND COHERENCE.

The concept of EPI analysis is not necessarily restricted to the case of cameras placed along a line, and has been extended by Feldmann et al. [16] with image cube trajectories (ICTs). The authors show that other one-dimensional camera setups, such as the circular case illustrated in Figure 3(c), generate particular trajectories in the plenoptic domain and occlusion-compatible orders can be defined. While the image cubes, EPI volumes, and videos are all three-dimensional, more dimensions can be added. For example, the case where the sensors are video cameras along a line leads to a four-dimensional parameterization that includes the time dimension.



[FIG3] Illustration of some 3-D plenoptic manifolds. (a) The volume carved out by a flat object in the space-time volume. (b) The volumes carved out by two flat objects in a linear camera array, where V_x denotes the position of the cameras along a line. (c) The volumes generated by two objects in the case of a circular camera array, where θ denotes the angle of the camera position around the circle. Note that the shape of the volume in (a) depends on the movement of the objects which means it is not necessarily structured. In both the other cases (linear and circular still image camera arrays), the shape of the manifold is constrained by the camera setup and occlusion events can be predicted.



[FIG4] The Epipolar Plane Image (EPI) volume. Cameras are constrained to a line resulting in a 3-D plenoptic function where x and y are the image coordinates and V_x denotes the position of the camera. Points in space are projected onto lines where the slope of the line is inversely proportional to the depth of the point. The volume illustrated is sliced in order to show the particular structure of the data.

PLANAR AND UNCONSTRAINED CAMERA ARRAYS

Planar camera arrays such as the one illustrated in Figure 1 give two degrees of freedom to camera locations (e.g., in V_x and V_y). The structure that governs the light rays in this case has been very well explored in the popular light field [12] or lumigraph [17] parameterizations introduced by Levoy and Hanrahan and by Gortler et al., respectively. In the planar case and assuming $V_z = t = 0$, a point in space is mapped onto a four-dimensional trajectory according to

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ V_x \\ V_y \end{pmatrix} = \begin{pmatrix} X/Z - V_x/Z \\ Y/Z - V_y/Z \\ V_x \\ V_y \end{pmatrix}, \qquad (2)$$

where V_x and V_y are variable. By extension of the EPI volume, 4-D hypervolumes are carved out by objects at different depths in the scene and, just as in the EPI, their shape is constrained. In a similar spirit to the plenoptic manifolds, the pop-up light field method [18] makes use of the segmentation of the data into coherent regions (or layers) for view interpolation purposes. The contours of layers are semimanually extracted on one image and propagated to the other views (i.e., the two other dimensions V_x and V_y) by applying a user-defined depth map. By performing the segmentation in this manner, the coherence of the layers is enforced in all the views.

Other planar camera setups include the dynamic light fields that are 5-D since the time dimension is captured as well. The concentric mosaic [19] introduced by Shum and He also gives two degrees of freedom to the camera locations. In that paper, a 1-D camera (i.e., capturing slit images) is free to move along a circle with variable radius. The data are therefore parameterized with three dimensions, namely the rotation angle, the radius, and the y axis of the image.

The case when the camera locations are unconstrained (i.e., free to move in V_x, V_y, and V_z) gives rise to the 5-D plenoptic modeling function [20] first introduced by McMillan and Bishop. An even more general case including the wavelength and the time dimension has been called the surface plenoptic function [10] by Zhang and Chen. It contains six dimensions since it is assumed that radiance along a light ray does not change unless it is occluded. While it is more difficult to visualize, a point in space generates a particular trajectory in these parameterizations and objects generate multidimensional hypervolumes.

PLENOPTIC MANIFOLDS

As we saw in the previous sections, points in space are mapped onto trajectories in the plenoptic function. Objects that are made of neighboring points in space are therefore mapped onto volumes (or more generally hypervolumes) that are made of neighboring trajectories. This collection of trajectories generates a multidimensional manifold $\mathcal{M}$ which we will call plenoptic manifold. Note that the concept of plenoptic manifold shares many ideas with the coherent layers in [18] and the IBR objects in [7].

There are two important elements to retain from the parameterizations described above. First, the multidimensional trajectories are constrained by the camera setup. This is illustrated by the way points in space are mapped onto the plenoptic domain. In the following, we will refer to this prior as the *geometry constraint*. Second, there is a well-defined occlusion ordering. Points at different depths generate different trajectories and will intersect in the event of an occlusion. The study of these trajectories determines which point will occlude the other. We will refer to this prior as the *occlusion constraint*. There are several benefits in considering the extraction of the whole manifold carved out by objects. The procedure enables a global vision of the problem and operates on the entire available data. That is, all the images are taken into account simultaneously and the segmentation is consistent throughout all the views, which increases robustness. As pointed out in [18], this consistency is also beneficial for applications such as view interpolation.

EXTRACTING PLENOPTIC MANIFOLDS IN MULTIVIEW DATA

The extraction of coherent regions in plenoptic data is essentially a segmentation problem and closely related to that of extracting layers. Semiautomatic schemes such as those proposed in [18], [7] can be very accurate but necessitate the user's input. Other methods are unsupervised, in which case the end result is usually obtained by initializing a set of regions and using an iterative method that converges toward the desired segmentation.

Some layer extraction methods include k-means clustering [5] and linear subspace approaches [21]. Other common methodologies use the Expectation-Maximization (EM) algorithm or energy minimization with graph-based methods such as Graph Cuts [22]. These methods can be very efficient but are sometimes difficult to formulate. Alternative approaches such as active contours [23] are based on the computation of the gradient of an energy functional and use a steepest-descent methodology to converge toward a minimum. As we will see in the next section, we set the problem of extracting plenoptic manifolds in the variational framework, since it scales naturally to multidimensional signals and is flexible in terms of functionals to minimize.

A VARIATIONAL APPROACH

Since their introduction in the late 1980s, active contours (a.k.a. snakes) [23], along with active surfaces and variational frameworks in general, have enjoyed a huge popularity in numerous computer vision and image processing algorithms. Part of their success is due to the introduction of the level set method [24] (see also box "The Level Set Method Basics"), which solved some of the issues such as numerical stability and topology dependence. Another key advantage lies in the formulation of the energy minimization, which can be extended to any number of dimensions [25], [26]. Applications of the 3D case include volumetric data segmentation [22] and space-time video segmentation [11]. In [25], dynamic 3D modeling is performed by using a 4D framework in order to impose coherence in the time dimension as well. The trend is definitely going toward higher dimensional analysis, and these methods are capable of dealing with it in an elegant fashion. Since the plenoptic function has seven dimensions, there is a role to play for variational methods in the analysis of multiview data.

The problem is formulated as follows: Start with a surface $\vec{\Gamma}(\vec{\sigma}) = (x(\vec{\sigma}), y(\vec{\sigma}), \dots) \subset \mathbb{R}^N$ for points $\vec{\sigma} \in \mathbb{R}^{N-1}$ and make it evolve until it converges to the boundaries of the sought-after region. In order to do this, the surface must be made dependent on an evolution parameter τ such that $\vec{\Gamma}(\vec{\sigma}, \tau)$ and assigned a speed function $F(\vec{\Gamma}(\vec{\sigma}, \tau))$ in order to evolve according to the following partial differential equation [29], [28]

$$\frac{\partial \vec{\Gamma}(\vec{\sigma}, \tau)}{\partial \tau} = F(\vec{\Gamma}(\vec{\sigma}, \tau))\vec{n}_\Gamma(\vec{\sigma}, \tau), \tag{3}$$

where $\partial\vec{\Gamma}/\partial\tau = \vec{v}_\Gamma$ is the velocity, $\vec{n}_\Gamma$ is the outward normal vector and the initial condition $\vec{\Gamma}(\vec{\sigma}, 0)$ is the starting point defined by the initialization of the algorithm. The velocity function F will be chosen such that the surface converges toward the desired segmentation when $\tau \to \infty$. In practice, F can be arbitrarily designed or it can be derived from an energy functional to minimize. In the latter case, the optimal speed in a steepest descent sense can be determined.

THE LEVEL SET METHOD BASICS

Consider the problem of evolving a boundary $\vec{\gamma}(s, \tau) = (x(s, \tau), y(s, \tau)) \subset \mathbb{R}^2$ with a speed F in its outward normal direction $\vec{n}$. This evolution can be described with the following partial differential equation

$$\frac{\partial \vec{\gamma}(s, \tau)}{\partial \tau} = F(\vec{\gamma}(s, \tau))\vec{n}(s, \tau),$$

where the initial condition is the curve in $\vec{\gamma}(s, 0)$. A natural way to implement the evolution is to discretize the curve with a set of connected points and compute the displacement for each point according to the speed F. While this approach seems natural, it has some drawbacks. First, the points may move in such a way that they are closer and closer together or farther and farther away, which can lead to numerical instabilities in the computation of derivatives. Second, a curve may be separated into two regions by the speed function or, inversely, two curves could merge (i.e., topological changes). These cases are difficult to deal with, since they require reparameterizing the curve. This procedure becomes even more problematic as the number of dimensions increases. The level set method [24] addresses these issues by embedding the curve $\vec{\gamma}$ as the zero level of a higher-dimensional surface $z = \phi(x, y, \tau) \subset \mathbb{R}^3$ and evolving the surface as opposed to the curve itself. In order to derive an evolution equation for the surface that will solve the original problem, $\phi(\vec{\gamma}(s, \tau), \tau) = 0$ needs to hold for all s and at all iterations τ. In other terms, the partial derivatives of $\phi(\vec{\gamma}(s, \tau), \tau)$ with respect to s and τ must be zero since the function is constant. The application of the chain rule for both cases leads to

$$\begin{cases} \frac{\partial \phi}{\partial \tau} + \vec{\nabla}\phi(\vec{\gamma}(s, \tau), \tau) \cdot \frac{\partial \vec{\gamma}}{\partial \tau} = 0 \\ \frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|} = -\vec{n} \end{cases}$$

where $\vec{\nabla}$ is the gradient operator. Putting the two together along with the definition of the speed of the original contour $\partial\vec{\gamma}/\partial\tau = F\,\vec{n}$ enables us to write the level set equation

$$\frac{\partial \phi(x, y, \tau)}{\partial \tau} = F(x, y)|\vec{\nabla}\phi(x, y, \tau)|.$$

The level set surface ϕ is free to expand, shrink, rise, and fall in order to generate the deformations of the original curve, and topological changes are naturally handled. Moreover, since this evolution equation is defined over the whole domain, there is no need to parameterize the curve with individual points. The numerical computations are performed by using finite differences on a fixed cartesian grid (thus solving the stability issue). Furthermore, the methodology extends naturally to evolving surfaces and generally hypersurfaces in any number of dimensions. These advantages, however, clearly come at the cost of computational complexity, since the gradients and the speed functions need to be computed for all the levels of ϕ. Some solutions to reduce the number of computations have been developed, such as the fast marching and narrowband methods [24].

The energy functional to minimize is usually written as a function of the surface

$$E_{\text{tot}}(\vec{\Gamma}) = E_{\text{data}}(\vec{\Gamma}) + E_{\text{smooth}}(\vec{\Gamma}), \tag{4}$$

where the first term E_{data} measures the consistency of the segmentation with the data and the second term E_{smooth} ensures smooth surfaces in order to compute derivatives and reject outliers. These terms are also referred to as the external and internal energies [23], since the former is measured by the data and the latter is derived from the properties of the curve itself. Assuming that the surface $\Gamma = \partial\mathcal{M}$ is the boundary of a region $\mathcal{M}$, the data contribution can be written as a region competition term [29]

THERE ARE STILL SOME CHALLENGING ISSUES IN TERMS OF PROCESSING, PRIMARILY DUE TO THE SHEER AMOUNT OF DATA INVOLVED WHEN THE NUMBER OF CAMERAS BECOMES VERY LARGE.

$$E_{\text{data}} = \int_{\mathcal{M}(\tau)} d_{\text{in}}(\vec{x})d\vec{x} + \int_{\overline{\mathcal{M}}(\tau)} d_{\text{out}}(\vec{x})d\vec{x},$$

where $\vec{x} \in \mathbb{R}^N$, $\overline{\mathcal{M}}$ is the outside of $\mathcal{M}$ and $d_{\text{in}}(\vec{x})$ and $d_{\text{out}}(\vec{x})$ are descriptors measuring the consistency with their respective regions. The smoothness constraint is written as a boundary-based term

$$E_{\text{smooth}} = \int_{\partial\mathcal{M}(\tau)} \mu d\vec{\sigma},$$

where μ is a constant weighting factor determining the influence of E_{smooth}. Minimizing the total energy thus involves computing the derivative of the total functional with respect to τ and evolving the boundary in a steepest descent fashion such that the energy converges to a minimum. It is possible to show that the gradient of the energy E_{tot} is given by [29], [26]

$$\frac{dE_{\text{tot}}(\tau)}{d\tau} = \int_{\partial\mathcal{M}(\tau)} [d_{\text{in}}(\vec{x}) - d_{\text{out}}(\vec{x}) + \mu\kappa(\vec{x})](\vec{v}_\Gamma \cdot \vec{n}_\Gamma)d\vec{\sigma},$$

where κ is the mean curvature of $\vec{\Gamma}$ and $\cdot$ denotes the scalar product. From this equation, we deduce that the steepest descent of the energy yields the following partial differential equation:

$$\vec{v}_\Gamma = [d_{\text{out}}(\vec{x}) - d_{\text{in}}(\vec{x}) - \mu\kappa(\vec{x})]\vec{n}_\Gamma, \tag{5}$$

where, by comparing (5) with (3), we now have an explicit form for the speed F of the evolving interface. Note that the so-called competition formulation is now clear. A point the belongs to the inside of the sought-after region has a small d_{in} and a large d_{out}, thus resulting in a positive speed. The point will therefore be incorporated. Inversely, a point belonging to the outside has a small d_{out} and a large d_{in}, resulting in a negative speed thus causing the point to be rejected. The curvature term helps to smooth the contour by straightening the curve in places where the curvature is large. On the basis of photoconsistency, some common descriptors used to extract coherent regions are related to the intensity differences between two frames or views and in a more global way, the variance along a plenoptic trajectory [29], [2].

While the equation in (5) driving the evolution of the hypersurfaces is valid in the general case, it does not take into account the geometry and occlusion constraints that are inherent to the plenoptic function. In the following sections, we study the cases of the EPI and the light field parameterizations (see "Linear Multiview Camera Arrays" and "Planar and Unconstrained Multiview Camera Arrays," above). These representations are popular, easier to visualize, and enable a clear imposition of the constraints. Some extensions to other camera setups are possible. Recall that, under these parameterizations, points in space are mapped onto lines in the plenoptic domain and the slopes of the lines are inversely proportional to the depth of the points. The next sections show how the evolution of the surfaces or hypersurfaces can be modified in order to take into account these constraints.

IMPOSING THE GEOMETRY CONSTRAINT

Consider a scene with a single object or layer that carves out a plenoptic manifold $\mathcal{M}$ in a light field. The problem of extracting the plenoptic manifold consists in finding the hypersurface $\vec{\Gamma} \subset \mathbb{R}^4$ that delimits the contour of the object on all the views. Let $\vec{\gamma}(s, \tau) = (x_0(s, \tau), y_0(s, \tau))$ be the 2D contour defined by the intersection of the hypersurface and the image plane in $V_x = V_y = 0$. That is, it represents the contour of the object on a single image. For simplicity, we assume that the depth map of the object is strictly frontoparallel, hence the depth $Z = Z_0$ is constant. According to (2), the boundary plenoptic manifold under these assumptions can be parameterized as

$$\vec{\Gamma}(s, V_x, V_y, \tau) = \begin{pmatrix} x_0(s, \tau) - V_x/Z_0 \\ y_0(s, \tau) - V_y/Z_0 \\ V_x \\ V_y \end{pmatrix}$$

and is completely determined by the curve $\vec{\gamma}(s, \tau)$ if we assume that Z_0 is known. It is therefore possible to propagate the position and the shape of $\vec{\gamma}$ on all the other images. That is, the shape variations in the hypersurface $\vec{\Gamma}$ are completely determined by the shape variations of the curve $\vec{\gamma}$ in the two-dimensional subspace. An explicit derivation of the normal and velocity vectors shows that the projection of $\vec{v}_\Gamma \cdot \vec{n}_\Gamma$ onto the subspace is equal to $\vec{v}_\gamma \cdot \vec{n}_\gamma$ and therefore does not depend on the location of the camera (V_x, V_y). The intuition behind this property is easily grasped. Given the fact that the depth map is constant, the layer's contour will simply be a translated version

of itself on all the other images. Hence the gradient of the data term can be rewritten in the form

$$\frac{dE_{\text{data}}(\tau)}{d\tau} = \int_{\gamma} [D_{\text{in}}(s) - D_{\text{out}}(s)](\vec{v}_{\gamma} \cdot \vec{n}_{\gamma}) ds,$$

where $D_{\text{in}}(s)$ and $D_{\text{out}}(s)$ are the original descriptors integrated over the plenoptic trajectories. Therefore the optimal velocity vector driving the evolution of the contour $\vec{\gamma}$ in the subspace becomes

$$\vec{v}_{\gamma} = [D_{\text{out}}(s) - D_{\text{in}}(s) - \mu\kappa(s)]\vec{n}_{\gamma}, \tag{6}$$

where a smoothness term is added to insure that the 2D contour stays regular. In the case of regions that are not frontoparallel, the same intuition holds; however, a weighting factor must be introduced in order to compensate for the shape changes between the views. In a more general sense, the relation between the normal velocity of original boundary $\vec{\Gamma}$ and the normal velocity of the contour $\vec{\gamma}$ becomes $\vec{v}_{\Gamma} \cdot \vec{n}_{\Gamma} = \alpha(\vec{\sigma})(\vec{v}_{\gamma} \cdot \vec{n}_{\gamma})$, where $\alpha(\vec{\sigma})$ is the weighting function depending on the depth map and the camera setup.

THE PLENOPTIC FUNCTION WAS INTRODUCED BY ADELSON AND BERGEN TO DESCRIBE THE VISUAL INFORMATION AVAILABLE FROM ANY POINT SPACE.

There are several advantages to using evolution equation (6) as opposed to (5). First, it constrains the shape of the manifold according to the camera setup. Second, it is implemented as an active contour instead of an active hypersurface, which reduces the computational complexity. However, it comes at the cost of having to compute the geometry of the object (i.e., the slope of the lines or in general the parameters of the plenoptic trajectories).

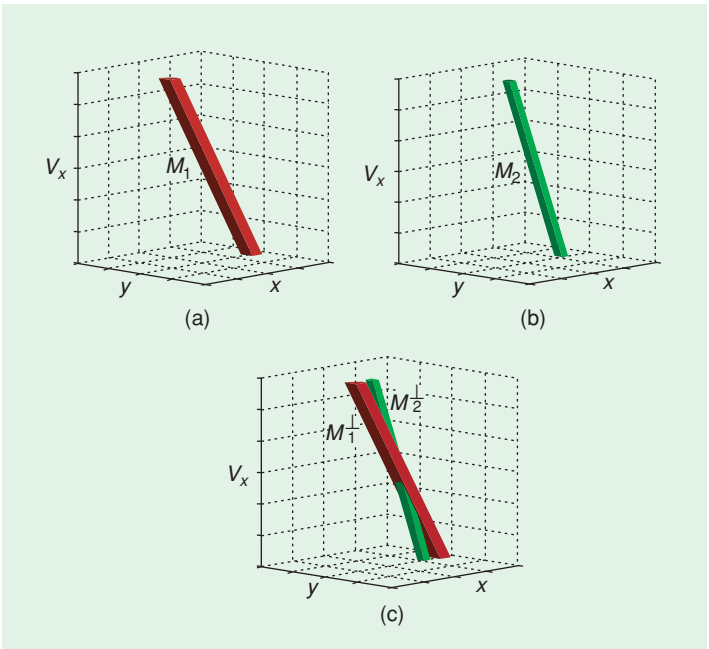
Inspired by stereo computer vision methods [30], we model the depth map as a linear combination of bicubic splines and use classical nonlinear optimization methods to find such parameters. The advantage of this model lies in the great variety of smooth depth maps that can be estimated. In addition, the bicubic splines can be forced to model simplified geometry if an accurate depth reconstruction is not necessary. For instance, frontoparallel regions can be extracted by imposing that all the weights of the linear combination are the same. The weights are estimated by minimizing the same functional (4) where the shape of the curves are kept constant. The overall optimization is done by an iterative approach in which the contours are estimated while filxing the depth maps and the depths are estimated while fixing the contours until there is no significant decrease in energy.

IMPOSING THE OCCLUSION CONSTRAINT

In the previous section, we have seen how to constrain the shape of the plenoptic manifolds according to the geometry of the camera setup. The second main factor to consider is occlusions and occlusion ordering. That is, sometimes the full manifold is not available, since it is occluded in some of the views. To account for this case, we denote as $\mathcal{M}_n$ the full manifold (as if it were not occluded) and $\mathcal{M}_n^{\perp}$ as the available manifold (i.e., excluding the occluded regions). Assuming the camera centers lie on a line or a plane (as in the EPI or light field parameterizations), this occlusion ordering stays constant throughout the views. Therefore, if the $\mathcal{M}_n$'s are ordered from front ($n = 1$) to back ($n = N$), the occlusion constraint [13], [14] can be written as

$$\mathcal{M}_n^{\perp} = \mathcal{M}_n \cap \sum_{i=1}^{n-1} \overline{\mathcal{M}_i^{\perp}}, \tag{7}$$

where the superscript *perp* denotes that the plenoptic manifold has been geometrically orthogonalized such that the occluding manifolds carve through the background ones (see Figure 5). A commonly used approach to deal with occlusions in EPI analysis is to start by extracting the frontmost regions (or lines) and removing them from further consideration [16], [4]. That is, when each $\mathcal{M}_n$ is extracted, the $\sum_{i=1}^{n-1} \overline{\mathcal{M}_i^{\perp}}$ is known. While this approach is straightforward, it has some drawbacks. First, the extraction of occluded objects will depend on how well the occluding objects were extracted. Second, it does not enable a proper competition formulation since the background regions are not known at the time of the extraction of the front ones.



[FIG5] The occlusion constraint with two 3-D plenoptic manifolds under the EPI parameterization. When put together, plenoptic manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ become $\mathcal{M}_1^{\perp}$ and $\mathcal{M}_2^{\perp}$. The occlusion constraint says that $\mathcal{M}_1^{\perp} = \mathcal{M}_1$ and $\mathcal{M}_2^{\perp} = \mathcal{M}_2 \cap \overline{\mathcal{M}_1^{\perp}}$.

An alternative approach consists in setting up a competition formulation between the front and background regions and using an iterative approach. The energy in this case becomes

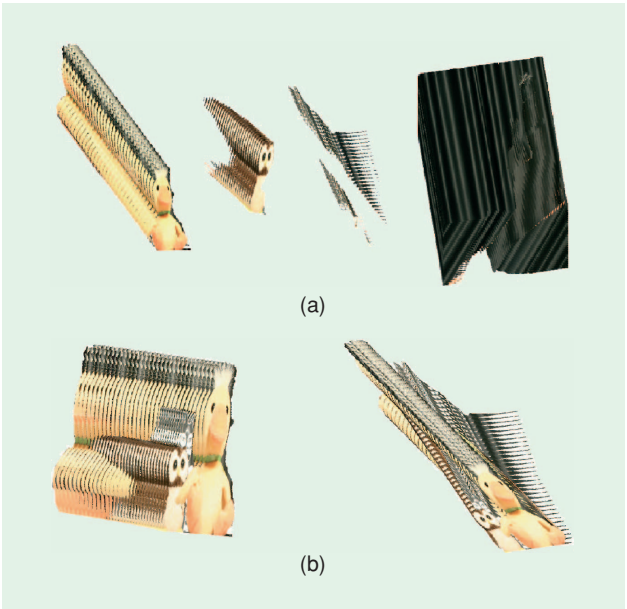
$$E_{\text{data}} = \sum_{n=1}^{N} \int_{\mathcal{M}_n^{\perp}} d_n(\vec{x}) d\vec{x},$$

which can be minimized by iteratively evolving one $\mathcal{M}_n^{\perp}$ while fixing the others. This leads to the evolution equation in (6) where all the other manifolds are gathered in $\overline{\mathcal{M}_n^{\perp}}$. The idea is that each iteration will contribute to minimize the total energy. Note that, as a result of the occlusion constraint in (7), evolving $\mathcal{M}_n^{\perp}$ will modify not only it but also all the manifolds it occludes (i.e., $\mathcal{M}_i^{\perp}$ where i goes from $n+1$ to N). Therefore these manifolds will contribute to the $dE_{\text{data}}/d\tau$ and influence the competition. That is, a manifold competes with all the regions it will occlude throughout all the views. However, a background manifold evolving does not affect the shape of the foreground ones (i.e., $\mathcal{M}_i^{\perp}$ where i goes from 1 to $n-1$) and therefore they will not contribute to $dE_{\text{data}}/d\tau$ and hence do not compete. An overview of the algorithm that shows how the two constraints are combined is presented in Table 1. In the next section, we show some experimental results with partial and total occlusions in order to demonstrate the benefits of applying this interplay between occluding and occluded regions.

IMAGE-BASED RENDERING IS ESSENTIALLY THE STUDY OF THE SAMPLING, INTERPOLATION, AND EXTRAPOLATION OF THE PLENOPTIC FUNCTION.

[TABLE 1] OVERVIEW OF THE PLENOPTIC MANIFOLD EXTRACTION ALGORITHM.

STEP 1:	INITIALIZE A SET OF PLENOPTIC MANIFOLDS
STEP 2:	ESTIMATE DEPTH PARAMETERS
STEP 3:	UPDATE OCCLUSION ORDERING AND UPDATE MANIFOLDS
STEP 4:	FOR EACH MANIFOLD FIX THE OTHER MANIFOLDS COMPUTE SPEED FUNCTION WITH COMPETITION TERMS EVOLVE BOUNDARY
STEP 5:	GO TO STEP 2 OR STOP WHEN THERE IS NO SIGNIFICANT DECREASE IN ENERGY



[FIG6] Automatically extracted 3-D plenoptic manifolds from the EPI volume in Figure 4. (a) The first row illustrates the individual volumes carved out by the different layers in the scene, and (b) the second row shows how the coherent regions fit together in the original data. Note how the foreground objects carve through the background ones.

APPLICATIONS IN IMAGE-BASED RENDERING

Image-based rendering is essentially the study of the sampling, interpolation, and extrapolation of the plenoptic function. It has attracted a lot of attention recently thanks to its ability to recreate visually pleasing and realistic virtual viewpoints from a set of multiview images. This ability is without a doubt one of the most important issues when it comes to free-viewpoint visual media systems. In this section, we analyze some natural multiview data sets with the variational framework presented above and discuss some of the applications related to IBR that benefit from the extraction of plenoptic manifolds.

As with all partial differential equation based—methods, initialization is an important issue. For the results shown, the initialization of the regions is performed in an unsupervised fashion by computing local directions in the EPI images and merging regions with similar slopes (i.e., similar depths). In more complicated camera setups, however, a more sophisticated method such as a stereo disparity estimation algorithm could be used. As in [2], [4], the descriptors used minimize the variance along plenoptic trajectories. The depth model adopted is piecewise constant. Using the competition formulation and imposing the geometry and occlusion constraints, the algorithm applied to the sequence in Figure 4 automatically extracts the plenoptic manifolds depicted in Figure 6. The data consist of 32 images containing a background and three objects, two of which are partially or totally occluded in numerous views. Note that occlusions and disocclusions are correctly captured. This is most obvious in the "cat" and "owl" layers.

The global nature of the segmentation scheme also suppresses some of the discontinuities visible when individual slices of the EPI volume are analyzed, as in [4]. We refer to [13], [14] for more experimental results. The running time is approximately 1000 seconds when the classical implementation of the level set method is used, although improvements of several orders of magnitude can be expected with faster implementations [24].

Several applications are possible once the multiview data has been segmented into individual plenoptic manifolds. For instance, one of the typical issues is view interpolation. This problem was studied by Chai et al. [9], using a classical signal processing framework, and was further generalized by Zhang and Chen in [10]. Both showed that, thanks to the particular structure of the data, the band of the plenoptic function is approximately bound by the minimum and maximum depths

in the scene. This fact makes it possible to give an answer to the minimum sampling rate needed (i.e., the number of cameras) in order to have an aliasing-free rendering. However, when the scene has somewhat large depth variations, this rate becomes very large and the number of cameras required may be cumbersome.

> THE ANALYSIS OF MULTIVIEW IMAGES CALLS FOR MULTIDIMENSIONAL SIGNAL PROCESSING ALGORITHMS THAT TAKE ADVANTAGE OF THE INHERENT REGULARITY.

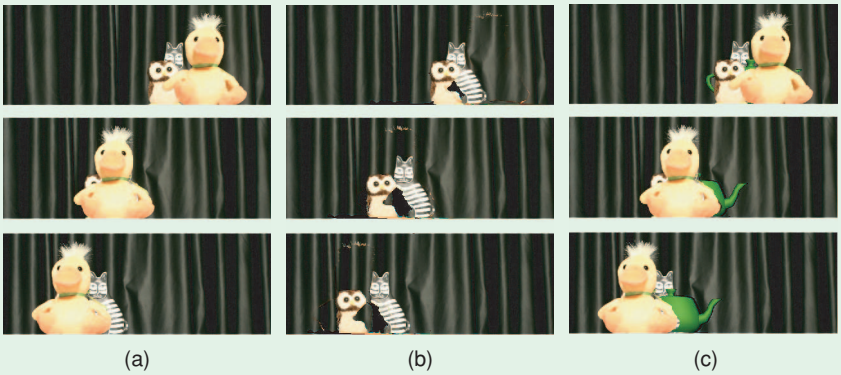
In [18], the scene is segmented into coherent regions that can be individually rendered, free of aliasing. The method uses a coherence matting approach to blend the layers and alleviate some of the errors caused by oversegmentation or undersegmentation. In a similar spirit, we use the plenoptic manifold segmentation scheme to interpolate new viewpoints. Figure 7(a) illustrates a linearly interpolated viewpoint, using a constant depth plane at the optimal depth [9]. The same rendered view in which three extracted plenoptic manifolds are individually interpolated by using their estimated depths is depicted in Figure 7(b) for comparison. The blurring is greatly reduced while the natural aspect of the images is maintained. As pointed out in [9], the aliasing is reduced because the individual depth variations in each plenoptic manifold are much smaller than in the whole scene, hence fewer cameras are needed for an aliasing free rendering.

The extraction of the plenoptic manifolds may also provide a first step in scene understanding. The fact that occlusions and object boundaries are known, for instance, may be utilized in object recognition algorithms. This understanding also allows to manipulate the multiview data in a coherent fashion. New scenes can be created by combining the plenoptic manifolds in different ways. For example, occluded regions may be extrapolated by using the available plenoptic trajectories and their intensities. New images are generated where background objects are disoccluded. Other plenoptic functions may be constructed by inserting the plenoptic manifolds of external objects (captured by a camera array or synthetically created) into the scene. Figure 8 illustrates some of these manipulations. Despite the simplified depth model used, the objects still show their original shapes in the rendered images (see the duck's beak, for instance). This is because the whole plenoptic manifolds are recombined instead of using a layer representation (i.e., alpha map, texture, and plane or motion parameters).

CONCLUSIONS

In this article, we looked into the coherence of multiview images from the plenoptic function point of view, emphasizing that looking at the problem from this angle provides a nice framework for studying the data in a global manner and imposing a coherent segmentation. Using this representation, we looked into the nature of the function, such as the structure, and suggested that in extension to the object tunnels in videos and EPI-tubes in multibaseline stereo data, objects carve multidimensional hypervolumes in the plenoptic function that we called plenoptic manifolds. Just as in the three-dimensional cases, the manifolds contain highly regular information, since they are constructed with images of the same objects. There is therefore clearly potential for robust analysis and efficient representation.

[FIG7] **Image-based rendering by plenoptic manifold interpolation. (a) The linearly interpolated viewpoint using a single plane at the optimal constant depth. Note that the image is blurred since there are not enough viewpoints for an aliasing free rendering. (b) The rendered image obtained by interpolating the data in the extracted plenoptic manifolds on the basis of their individual estimated depths.**



[FIG8] **"To duck or not to duck, or maybe to teapot?": (a) illustrates some of the original image taken from a multiview image sequence, (b) shows the same images with the manifold carved out by the duck removed and the background manifolds extrapolated. Note that there are some incomplete regions, since they are never visible in the entire stack of images, and (c) illustrates the insertion of a synthetic manifold generated by a teapot.**

We then looked into extracting these manifolds in simple camera setups, using a variational framework that is flexible in terms of the number of dimensions (depending on the camera setup), the depth estimation, and the descriptors used. This flexibility is important for several reasons. First, the same framework can be used for different camera setups. Second, some applications in image-based rendering do not always necessitate an accurate depth reconstruction. Third, possible extensions to take into account large textureless regions and specular effects, for instance, may be incorporated into the descriptors. Future work may explore more complicated and unstructured camera setups as well as dynamic scenes and nonrigid objects. This would eventually lead to the extraction of seven dimensional manifolds.

ACKNOWLEDGMENTS

The authors wish to thank the Audiovisual Communications Laboratory (LCAV) at the Swiss Federal Institute of Technology (EPFL) for providing the equipment to capture the data sets shown in this article. They also thank Yizhou Wang for helping to prepare the images in Figure 3. Jesse Berent acknowledges the Engineering and Physical Sciences Research Council (EPSRC) for the doctoral training grant. The work presented is also funded in part by the Royal Society.

AUTHORS

Jesse Berent (jesse.berent04@imperial.ac.uk) received his master's degree in microengineering from the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland in 2004. He is currently with the Communications and Signal Processing Group at Imperial College London where he is working on his Ph.D. thesis. In 2006, he was a visiting research student at the Audiovisual Communications Laboratory at EPFL, Switzerland. His research interests include multiview imaging, biomedical image processing and sampling theory. He is a Student Member of the IEEE.

Pier Luigi Dragotti (p.dragotti@imperial.ac.uk) is a senior lecturer in the Electrical and Electronic Engineering Department at Imperial College, London. He received the Laurea in electrical engineering from the University Federico II, Naples, in 1997, the master's degree in communications systems from EPFL, Lausanne, in 1998, and the Ph.D. degree from EPFL in 2002. He was a visiting student at Stanford University and a summer researcher in the Mathematics of Communications Department at Bell Labs, Lucent Technologies. His research interests include wavelet theory, image and video processing and compression, joint source-channel coding, and signal processing for sensor networks. He is a Member of the IEEE.

REFERENCES

[1] E.H. Adelson and J.R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. Cambridge, MA: MIT Press, 1991, pp. 3–20.

[2] M. Ristivojevic and J. Konrad, "Space-time image sequence analysis: Object tunnels and occlusion volumes," *IEEE Trans. Image Processing*, vol. 15, pp. 364–376, Feb. 2006.

[3] R. Bolles, H.H. Baker, and D. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 7–55, 1987.

[4] A. Criminisi, S.B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan, "Extracting layers and analyzing their specular properties using epipolar-plane-image analysis," *Comp. Vis. Image Understanding*, vol. 97, no. 1, pp. 51–85, Jan. 2005.

[5] J.Y.A. Wang and E.H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Processing, Special Issue on Image Sequence Compression*, vol. 3, pp. 625–638, Sept. 1994.

[6] J. Shade, S. Gortler, L.W. He, and R. Szeliski, "Layered depth images," in *Proc. Comput. Graphics* (SIGGRAPH '98), 1998, pp. 231–242.

[7] Z.F. Gan, S.C. Chan, K.T. Ng, and H.Y. Shum, "An object-based approach to plenoptic videos," in *IEEE Int. Symp. Circuits and Syst.*, 2005, vol. 4, pp. 3435–3438.

[8] C.L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE Trans. Image Processing*, vol. 15, no. 4, pp. 793–806, Apr. 2006.

[9] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong, "Plenoptic sampling," in *Proc. Comput. Graphics (SIGGRAPH '00)*, 2000, pp. 307–318.

[10] C. Zhang and T. Chen, "Spectral analysis for sampling image-based rendering data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 1083–1050, Nov. 2003.

[11] J. Konrad, "Videopsy: Dissecting visual data in space-time," *IEEE Commun. Mag.*, vol. 45, no. 1, pp. 34–42, 2007.

12] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Comput. Graphics (SIGGRAPH '96)*, 1996, pp. 31–42.

[13] J. Berent and P.L. Dragotti, "Segmentation of epipolar plane image volumes with occlusion and dissocclusion competition," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, Oct. 2006, pp. 182–185.

[14] "Unsupervised extraction of coherent regions for image based rendering," in *Proc. British Machine Vision Conf.*, 2007.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge Univ. Press, 2004.

[16] I. Feldmann, P. Eisert, and P. Kauff, "Extension of epipolar image analysis to circular camera movements," in *Proc. IEEE Int. Conf. Image Processing*, 2003, pp. 697–700.

[17] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, "The lumigraph," in *Proc. Comput. Graphics (SIGGRAPH '96)*, 1996, pp. 43–54.

[18] H.Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C.K. Tang, "Pop-Up light field: An interactive image-based modeling and rendering system," *ACM Trans. Graph.*, vol. 23, no. 2, pp. 143–162, Apr. 2004.

[19] H.-Y. Schum and L.-W. He, "Rendering with concentric mosaics," in *Proc. Comput. Graphics (SIGGRAPH '99)*, 1999, pp. 299–306.

[20] L. McMillan and G. Bishop, "Plenoptic modeling: an image-based rendering system," in *Proc. Comput. Graphics* (SIGGRAPH '95), 1995, pp. 39–46.

[21] Q. Ke and T. Kanade, "A subspace approach to layer extraction," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, vol. 1, 2001, pp. 255–262.

[22] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001 [26] pp. 419–430.

[23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.

[24] J. Sethian, *Level Set Methods*. Cambridge, UK: Cambridge Univ. Press, 1996.

[25] B. Goldluecke, I. Ihrke, C. Linz, and M. Magnor, "Weighted minimal hypersurface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 7, pp. 1194–1208, July 2007.

[26] J. Solem and N. Overgaard, "A geometric formulation of gradient descent for variational problems with moving surfaces," in *Proc. Int. Conf. Scale Space and PDE Methods Comput. Vision*, 2005, pp. 419–430.

[27] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, "Minimal surfaces based object segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 394–398, Apr. 1997.

[28] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 1, no. 22, pp. 61–79, 1997.

[29] S. Jehan-Besson, M. Barlaud, and G. Aubert, "DREAMS: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation," *Int. J. Comput. Vis.*, vol. 53, no. 1, pp. 45–70, 2003.

[30] M. Lin and C. Tomasi, "Surfaces with occlusions from layered stereo," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2003, pp. 710–717. SP