Centralized and Interactive Compression of Multiview Images

Andriy Gelman[a], Pier Luigi Dragotti[a] and Vladan Velisavljević[b]

[a]Communications and Signal Processing Group, Imperial College London, UK, SW7 2AZ
[b]Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany

ABSTRACT

In this paper, we propose two multiview image compression methods. The basic concept of both schemes is the layer-based representation, in which the captured three-dimensional (3D) scene is partitioned into layers each related to a constant depth in the scene. The first algorithm is a centralized scheme where each layer is de-correlated using a separable multi-dimensional wavelet transform applied across the viewpoint and spatial dimensions. The transform is modified to efficiently deal with occlusions and disparity variations for different depths. Although the method achieves a high compression rate, the joint encoding approach requires the transmission of *all data* to the users. By contrast, in an interactive setting, the users request only a *subset* of the captured images, but in an unknown order a priori. We address this scenario in the second algorithm using Distributed Source Coding (DSC) principles which reduces the inter-view redundancy and facilitates random access at the image level. We demonstrate that the proposed centralized and interactive methods outperform H.264/MVC and JPEG 2000, respectively.

Keywords: Multiview image compression, interactive communication, wavelet transform.

1. INTRODUCTION

In recent years, image-based rendering (IBR) has been proposed as an alternative to traditional rendering techniques. The approach is based on the notion that novel viewpoints can be synthesized by interpolating existing images. The method achieves photo-realistic results with low computational complexity and therefore supports immersive viewing applications. To obtain artifact-free results, however, the scene must be sampled with a large number of cameras. These images are either transmitted or stored, which means efficient compression is an essential part of IBR systems.[1]

There are a number of trade-offs which must be considered when encoding multiview images. The main one is in terms of compression performance and random access.[1] Random access is an important feature for interactive applications where only a subset of the images is transmitted in an unknown order a priori. An example is an online streaming service where the user interactively selects images used for interpolation as she/he moves through the scene. One of the main issues, however, is that compression performance relies on removing inter-view redundancy. In most cases this is implemented using disparity compensated prediction which inevitably limits the random access capabilities.

In terms of ongoing research, the majority of compression literature has focused on achieving a high compression using techniques such as hierarchical prediction[2] or subband coding.[3,4] A number of methods have been proposed which achieve a high compression and still maintain random access. For example, in[5] the authors propose storing multiple representations of an image for a set of possible predictions to reduce the transmission rate and eliminate drift. This method, however, requires high storage requirements at the server. A different approach[6,7] has been to use Distributed Source Coding (DSC) principles to reduce the storage size and eliminate the side information uncertainty.

In this paper we aim to address the random access and compression efficiency issues. We propose two methods: the first is a centralized scheme which decomposes the data using a multi-dimensional wavelet transform. The

Further author information: (Send correspondence to Andriy Gelman)
Andriy Gelman: E-mail: andriy.gelman@imperial.ac.uk
Pier Luigi Dragotti: E-mail: p.dragotti@imperial.ac.uk, Telephone: +44 (0)20 7594 6192
Vladan Velisavljević: E-mail: vladan.velisavljevic@telekom.de, Telephone: +49 (0)30 8353 58537

approach achieves a high compression, however, does not support random access due to the transform coefficient inter-dependence. In the second algorithm, we trade-off compression efficiency to obtain random access at the image level. We implement this feature by using DSC principles that we use to remove the uncertainty in the user's viewing trajectory. Both of the proposed methods are based on the concept that a multiview image dataset can be partitioned into a set of layers each related to a constant depth in the scene as shown in Fig. 5. We call this partition the layer-based representation.

The outline of this paper is as follows. In the following section we discuss the multiview data structure and review the layer-based representation. We present our centralized and interactive methods in sections 3 and 4, respectively. The performance of both algorithms is evaluated in Section 5 and we 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 which 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*.[8] Introduced by Adelson and Bergen, this function parameterizes 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:

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

where I corresponds to the light ray intensity.

In practise, it is not feasible to store, transmit or capture the seven-dimensional function and a number of simplifications are 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 parametrization of the plenoptic function, known as the *light field*[9] defines each light ray by its intersection with a camera and a focal plane:

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

where as illustrated in Fig. 1, (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 one point of the camera plane. We illustrate a 4×4 image example of a light field in Fig. 2.

The light field can be further simplified by setting the 2D camera plane to a line. This is also known as the epipolar-plane image (EPI) volume[12]:

$$I = P_3\left(V_x, x, y\right). \tag{3}$$

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. Next, we review the EPI and light field structure and present the layer-based representation.

2.2. EPI and Light Field Structure

Multiview images capture the same scene from different locations and because of this are structured and redundant. Consider the EPI volume, where as shown in Fig. 3(a) we model each camera using the pinhole model.[13]

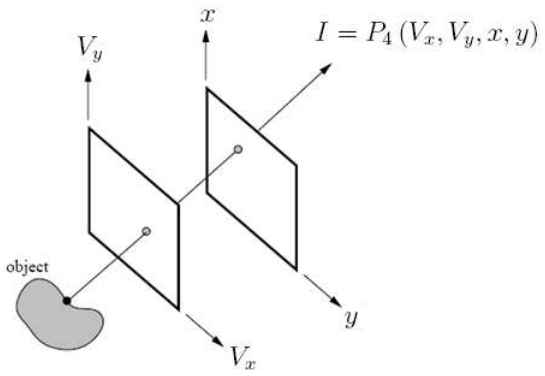


Figure 1. Light field parametrization. Each light ray is defined by its intersection with a camera (V_x, V_y) and a focal plane (x, y).[10]

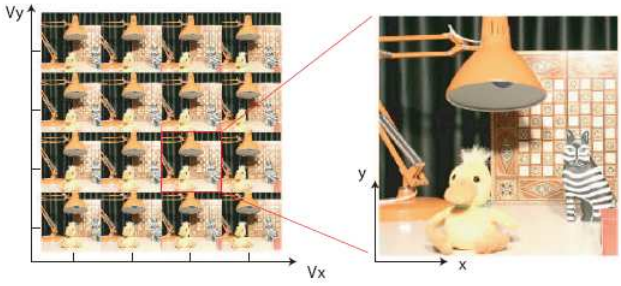


Figure 2. Captured light field.[11] Dataset can be analysed as a 2D array of images.

Assuming there are no occlusions, it can be shown that a light ray originating from a point in space (X, Y, Z) intersects each camera focal plane with coordinates

$$x = \frac{fX}{Z} - \frac{fV_x}{Z}, \tag{4}$$

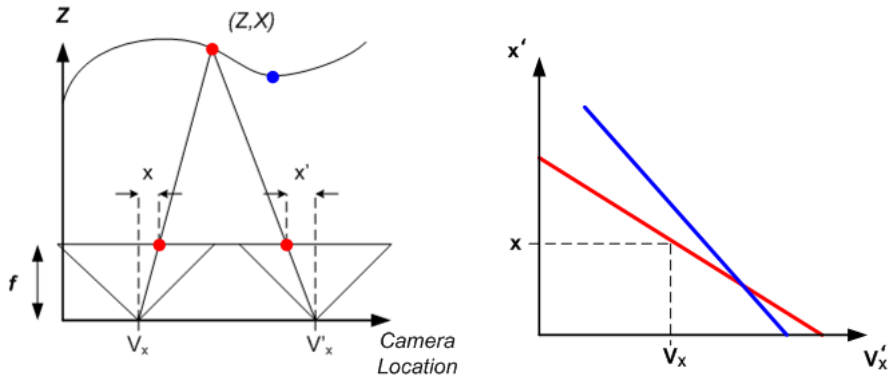
$$y = \frac{fY}{Z}, \tag{5}$$

where f is the focal length. Observe that the spatial coordinate x is linear with respect to the camera position and that the gradient, which is also called the disparity $\Delta p = \frac{f}{Z}$, is inversely proportional to the depth. We define the set of coordinates in (4) and (5) for each point in space (X, Y, Z) as an EPI line. Furthermore, assuming that the scene is Lambertian*, and that the light ray intensity does not change along its path, the intensity along an EPI line is constant. This property is illustrated in Fig. 3, where we show that each point in space is mapped to a line in the EPI volume. In addition to the constant intensity along the EPI lines, observe that the occlusion ordering can also be predicted. Recall that the disparity Δp of an EPI line is inversely proportional to the depth. Therefore, as shown in Fig. 3(b), when two lines intersect, the line corresponding to the larger disparity will occlude the other.

This concept can also be extended to the light field where, instead of a line, a point (X, Y, Z) maps onto a 2D plane as

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

*In a Lambertian scene the light ray intensity is constant when an object is observed from a different angle.



(a) (b)

Figure 3. (a) The camera moves along a straight line and is perpendicular to the baseline. (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 $x' = x - \frac{f\left(V_x' - V_x\right)}{Z}$ and $y' = y$.

2.3. Layer-based representation

The layer-based representation is an extension of the EPI line concept. The representation partitions the multi-view data into homogenous regions, where each layer is a collection of EPI lines related to a constant depth in the scene. An example of the representation is shown in Fig. 5.

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

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

where s parameterizes the contour $\gamma_k(s)$. This concept is further illustrated in Fig. 6(a) which shows an unoccluded layer from the Animal Farm dataset.[11] Observe that the complete segmentation can be defined by the red boundary $\gamma_k(s)$ on the first image viewpoint projected to the remaining frames. In terms of compression this concept is important since it means that the data segmentation can be inferred at the decoder by transmitting the contour $\gamma_k(s)$ and the layer's disparity Δp_k.

Note that the above approach does not take into account occlusions. Using the same principles as in the case of an EPI line, a layer will be occluded when it intersects with other layers which are related to a smaller depth. We illustrate this in Fig. 4, which shows that when two layers intersect we obtain their occluded representations $\mathcal{H}_{k-1}^{\perp}$ and $\mathcal{H}_k^{\perp}$. 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 addition, we show an example of an occluded layer from a real dataset in Fig. 6(b). We can use this notion to reconstruct the complete segmentation of each layer using the set of layer contours $\{\gamma_1(s), \ldots, \gamma_N(s)\}$ and the corresponding disparities $\{\Delta p_1, \ldots, \Delta p_N\}$.

3. CENTRALIZED MULTIVIEW IMAGE COMPRESSION

In this section we present our centralized compression method. The joint encoding means that we can efficiently reduce the data redundancy in the layer-based representation shown in Fig. 5 and thus achieve a high compression. The method, however, requires that all of the data is transmitted and decoded. Therefore, a drawback of the scheme is that it does not support interactive communication.

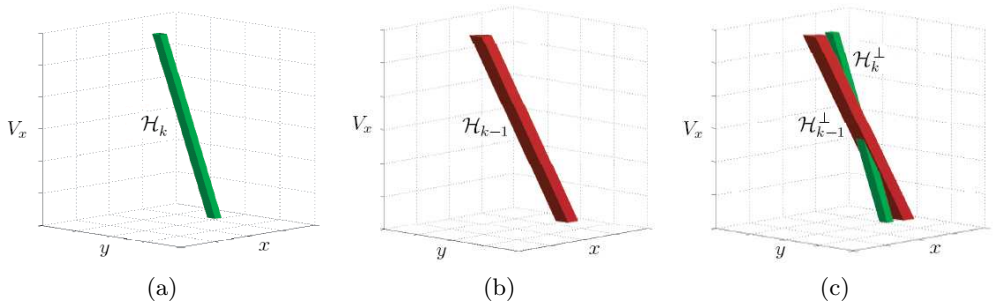


Figure 4. Comparison between two volumes $\mathcal{H}_{k-1}$, $\mathcal{H}_k$ and their intersection. The volumes are ordered in terms of increasing depth (i.e. $\mathcal{H}_{k-1}$ corresponds to a smaller depth than $\mathcal{H}_k$). (a) A set of EPI lines related to a constant disparity Δp_k. The collection of EPI lines carve out a volume $\mathcal{H}_k$. Observe that the complete segmentation of the volume can be defined by a boundary on one viewpoint projected to the remaining frames. (b) Volume $\mathcal{H}_{k-1}$ modeled by a constant disparity Δp_{k-1}. (c) When the two volumes intersect, layer $\mathcal{H}_{k-1}$ will occlude $\mathcal{H}_k$ as it is modeled by a smaller depth. We define the visible volumes with $\mathcal{H}_{k-1}^{\perp}$ and $\mathcal{H}_k^{\perp}$.

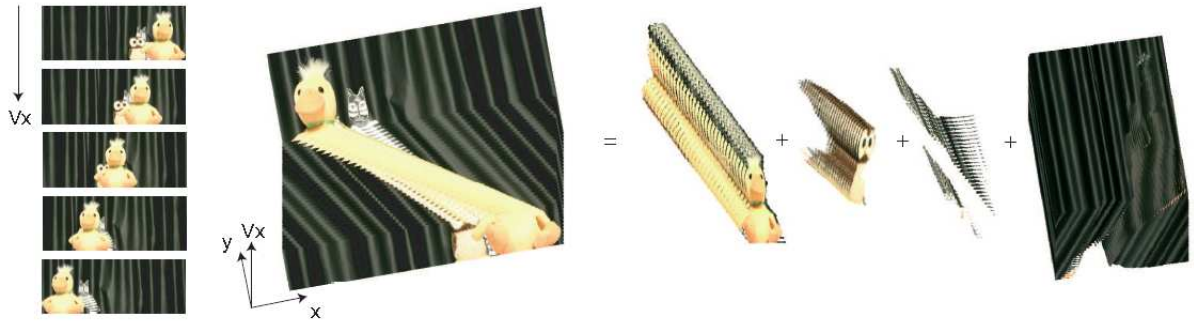


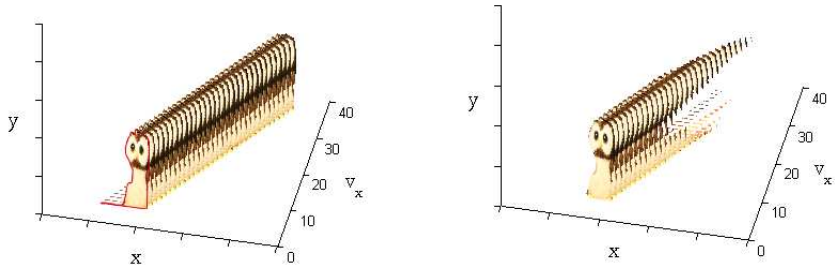
Figure 5. Animal Farm layer-based representation.[11] The dataset can be divided into a set of volumes where each one is related to a constant depth in the scene. Observe that the layer contours at each viewpoint remain constant, unless there is an intersection with another layer which is modeled by a smaller depth.

The high-level overview of the method is shown in Fig. 7. In the first stage we extract the layer-based representation reviewed in Section 2.3. The segmentation method is beyond the scope of this paper and we refer the reader to[11] for an in depth explanation of the approach. Each extracted layer is defined by a segmentation on the first image viewpoint and the corresponding disparity. We losslessly encode this information and transmit it to the decoder. We then reduce the redundancy of each layer using a 4D Discrete Wavelet Transform (DWT) applied in a separable fashion, first across the viewpoint dimensions, followed by a 2D DWT applied to the image dimensions. Recall that each layer is related to a constant depth in the scene. As a consequence, we disparity compensate the inter-view transform to achieve a more compact representation. The obtained transform coefficients are finally quantized, entropy coded and transmitted to the decoder. Next we present each of the stages in more detail.

3.1. 2D Inter-view 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 followed by the column images as illustrated in Fig. 8. This 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 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 compact representation. It is applied by modifying the standard lifting equations[14]



(a) (b)

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 $\mathcal{H}_k^{\perp}$ can be inferred by removing the regions which intersect with other layers related to a smaller depth.

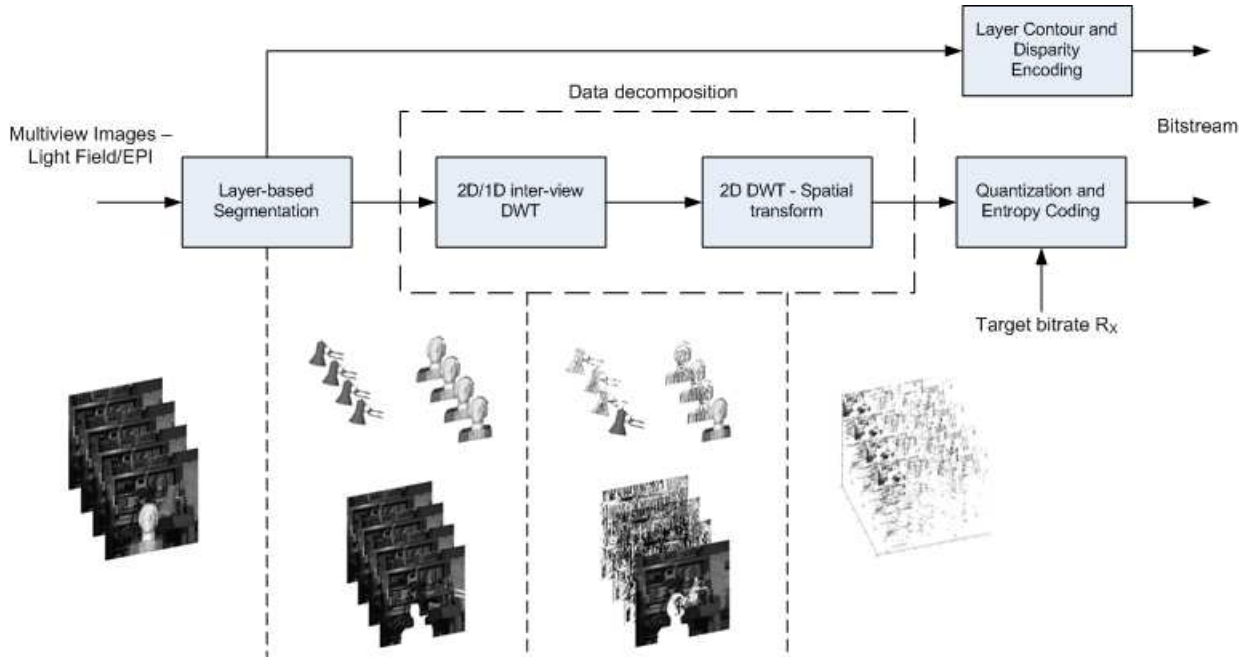
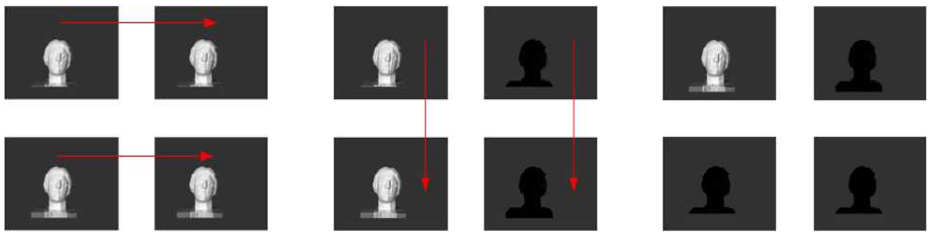


Figure 7. Centralized compression method block diagram. Additionally, we illustrate the obtained transform coefficients at each stage of the method.



(a) (b) (c)

Figure 8. Inter-view 2D DWT implemented in a separable approach by filtering the image rows followed by the image columns 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.

and including a warping operator $\mathcal{W}$ as follows:

$$\begin{aligned}\mathscr{L}_o[n] &= \frac{P_o[n] - \mathcal{W}\{P_e[n]\}}{2},\\ \mathscr{L}_e[n] &= P_e[n] + \mathcal{W}\{\mathscr{L}_o[n]\},\end{aligned} \tag{8}$$

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 (8), $\mathscr{L}_e[n]$ 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.[15]

In both the lifting and update steps in (8), the warping operator $\mathcal{W}$ is chosen to maximize 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 modeled by a constant disparity, we define the warping operation from viewpoint n_1 to n_2 along the V_x dimension as:

$$\mathcal{W}_{n_1\to n_2}\{P[n_1]\}(x,y) = P[n_1](x + \Delta p(n_2 - n_1), y), \tag{9}$$

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 compression efficiency. To prevent this, we use the concept proposed in[16] to create a shape-adaptive transform in the view domain. The transform in (8) is modified whenever a pixel at an even or odd location is occluded such that

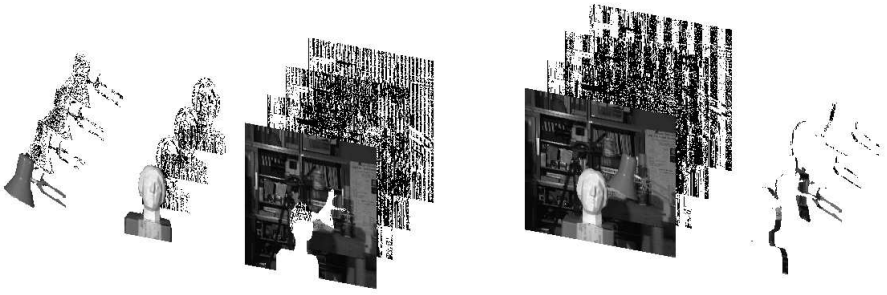
$$\mathscr{L}_e[n] = \begin{cases} P_e[n], & \text{occlusion at } 2n+1\\ \widehat{\mathcal{W}}\{P_o[n]\}, & \text{occlusion at } 2n\end{cases}, \tag{10}$$

and the high-pass coefficient in $\mathcal{L}_o[n]$ is set to zero. In (10), the warping operator $\widehat{\mathcal{W}}$ is set to an integer pixel precision to ensure invertibility and this is done by *ceiling* the disparity in (9).

3.2. 2D Spatial DWT

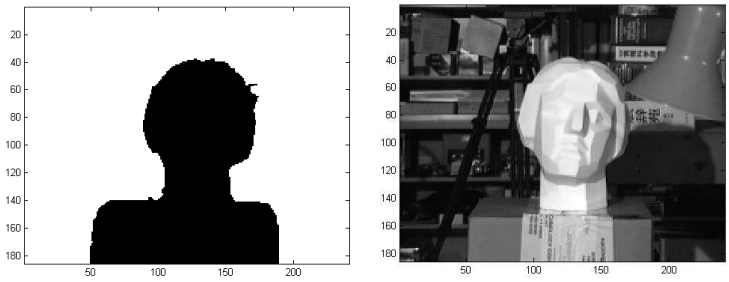
To improve the efficiency of the spatial transform, the subbands from each layer are grouped together into a single image and are further jointly processed. A comparison between the original and recombined layers is illustrated in Fig. 9. Note that due to occlusions and the way in which the inter-view 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[16] 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[17]), which, together with the symmetric signal extensions, leads to critically sampled transform subbands.



(a) (b)

Figure 9. (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) (b)

Figure 10. (a) Layer contour from the Tsukuba dataset.[18] (b) Corresponding Tsukuba texture.

3.3. Layer contour and disparity encoding

The contour encoding block transmits the segmentation needed to correctly decode each layer. Recall, from the properties of multiview data outlined in Section 2.3, that the segmentation of each layer can be defined by a contour on one of the image viewpoints and the layer's disparity. The input to the layer contour encoding algorithm is therefore an array of binary images (one for each layer). A typical binary layer is illustrated in Fig. 10(a).

We losslessly encode the layer contours and transmit the data to the decoder. The problem of lossless contour encoding has been extensively studied with the majority of the work based on Freeman chain codes.[19] Here, we use the algorithm proposed in,[20] where the boundary is differentially encoded using Huffman entropy coding. It is also possible to encode the layer contours in a lossy modality. This scheme is important at low bit rates where the majority of the bit budget should be allocated to encoding the texture (rather than the layer contours). We refer the reader to[4] where this issue has been addressed.

The disparity of each layer is also losslessly encoded and transmitted. The rate required to encode these parameters is negligible in comparison to the encoding rate of the texture and the layer contours.

3.4. Quantization and Entropy Coding

In this stage the transform coefficients are quantized and entropy coded. Our codec uses context adaptive arithmetic coding to attain bit rates close to the entropy of the source.

For a given bit budget constraint $\mathbf{R_x}$, the optimal bit allocation among the transform coefficients is achieved using a method similar to EBCOT.[21] Initially, the coefficients are partitioned into blocks and losslessly encoded

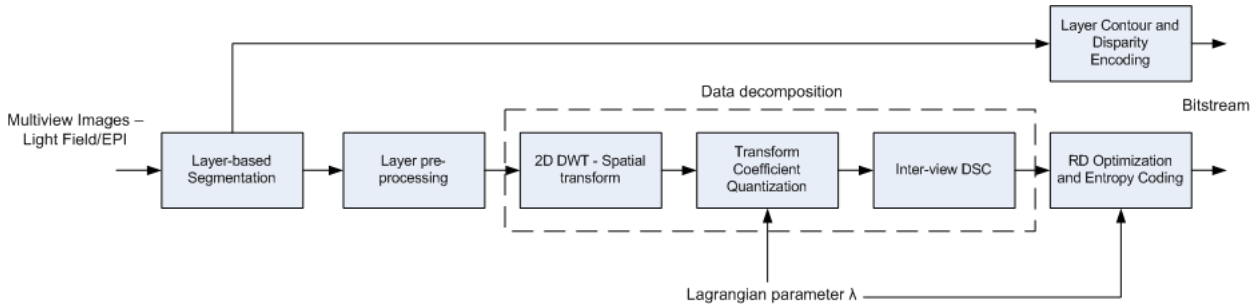


Figure 11. Interactive compression method block diagram.

by bit-plane to obtain the operational RD curves. Then, given a Lagrangian multiplier λ_x, a bit allocation R^* for each block is chosen such that the cost function J is minimized, where:

$$J = D\left(R_i\right) + \lambda_x R_i, \tag{11}$$

and $(R_i, D\left(R_i\right))$ is the rate-distortion pair associated to each operational point. To meet the allowed bit budget $\sum_l R_l^* = \mathbf{R_x}$, a bisection search[22] for the correct multiplier λ_x is applied. We note that since the operational RD curves are known, this process is not computationally expensive. Once the optimal values R_l^* are evaluated, the encoded bit streams are truncated and transmitted.

4. INTERACTIVE MULTIVIEW IMAGE COMPRESSION

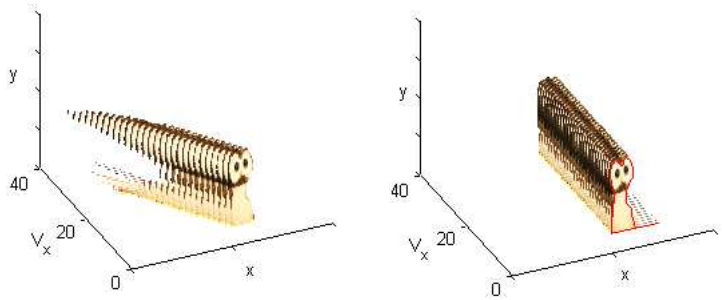
Here we present our interactive coding method which supports random access at the cost of reduced compression efficiency. One of the main issues in IBR streaming applications is that the viewing trajectory of the user is unknown. Therefore, even though there exists correlation between the requested images it is not possible to design a disparity compensated prediction structure without significantly increasing the storage requirements at the server. In the proposed method, we deal with this problem by substituting the inter-view DWT transform with DSC principles. Recall that DSC is closely related to channel coding and can be used to correctly reconstruct a data sample given its noise corrupted version. We use the same ideas to reconstruct the transmitted images given any reference side information image available in the cache of the user. The side information is treated as a noisy version of the transmitted signal. The approach also facilitates random access since each image is encoded independently.

The overview of the proposed interactive method is shown in Fig. 11. In the first stage we obtain the layer-based representation which is followed by a pre-processing stage applied to each layer. Next, we reduce the intra-view redundancy using a shape-adaptive 2D DWT applied to each image in the obtained layers. This is followed by DSC applied to the quantized subband coefficients along the inter-view domain. We then optimize the coefficients in the rate-distortion sense and entropy code the resulting data. In addition, we also losslessly encode and transmit the layer contours and the disparity of each layer.

In the following section we describe in more detail each of the encoding steps that are different from the centralized method.

4.1. Layer pre-processing

The input to this stage are the extracted layers shown in Fig. 5. The aim is to increase the inter-view correlation such that the compression efficiency of the DSC stage is more efficient. We implement the pre-processing stage by disparity compensating the images of each layer onto a common viewpoint using (6). Additionally, any occluded regions are interpolated using the mean along the EPI lines. The interpolation implies that there will be overlapping layer regions during the decoding process. However, as discussed in Section 2.3, we can infer the correct occlusion ordering using the associated depth/disparity of each layer. We show an example of the pre-processing in Fig. 12.



(a) (b)

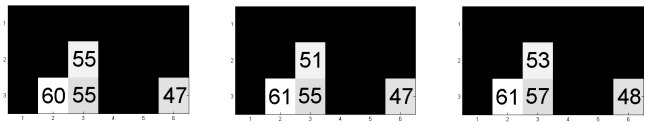
Figure 12. (a) Due to occlusions, the extracted layers may contain discontinuities along the EPI lines. (b) Pre-processing is implemented by disparity compensating each image onto a common viewpoint and interpolating the occluded pixels using the mean along the EPI lines. The layer contour outlined with the red curve is efficiently encoded using a modified version of the Freeman algorithm[20] and transmitted.

4.2. Transform Coefficient Quantization

Here, we quantize the transform coefficients following the intra shape-adaptive 2D DWT applied to each of the layer images. The quantization step-size is chosen using a Lagrangian optimization method similar to the centralized approach in Section 3.4.

4.3. Inter-view Distributed Source Coding

Consider the quantized low-pass transform coefficients from three images of one layer illustrated in Fig. 13. Observe that the subbands are correlated across the views. In this stage we exploit the inter-view redundancy using DSC principles.



(a) (b) (c)

Figure 13. Quantized low-pass subbands from three images of one layer. Observe that the blocks are correlated across the views.

During decoding, the remote user may contain any image as side information. Using DSC principles we can remove the side information uncertainty, while still reducing the number of bits which must be transmitted. Consider the following model:

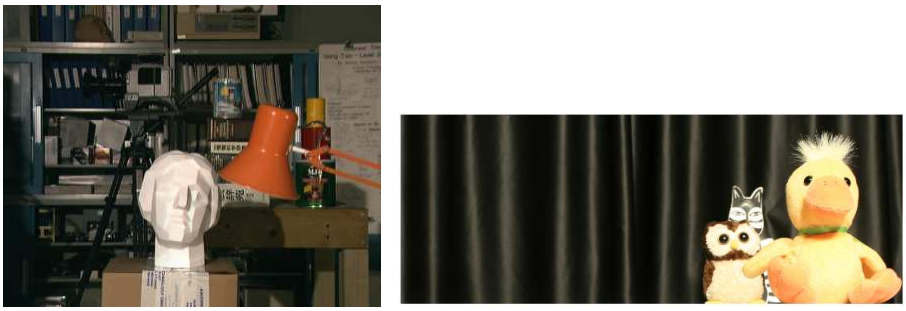
$$y = x + n, \tag{12}$$

where y is the transform coefficient requested by the user, x is the side information available in the cache and n is the residual signal. Recall that y can be correctly reconstructed by transmitting a minimum of $\lceil \log_2 (2n+1) \rceil$ least significant bits (LSB) from y. To encode a sequence of blocks shown in Fig. 13, we take the worst case scenario, where any image can be used as side information. For example, the transform sequence $\{55, 51, 53\}$ requires $\lceil \log_2 9 \rceil = 4$ LSB to correctly reconstruct the data.

This method is then applied to each set of transform coefficients across the views to determine the number of LSB that must be transmitted in order to correctly reconstruct the data given any image as side information.

4.4. RD Optimization and Entropy Coding

Observe that the outlined DSC approach described in the previous section is inefficient when the transform coefficients across the views are the same except for one frame. For example to encode $\{55, 55, 57\}$ we have to



(a) Tsukuba light field $[272 \times 368 \times 4 \times 4]$. (b) Animal Farm EPI volume $[232 \times 624 \times 16]$.

Figure 14. Multiview image analysis datasets.

transmit 3 LSB from each coefficient. Intuitively, a better solution would be to set 57 to 55, so that zero LSB are encoded. We solve this problem in an RD sense as follows: for each set of transform coefficients, we find the value which corresponds to the largest error and set it to the median of the set. Then, we evaluate the change in distortion ΔD and estimate change in rate ΔR. Using a greedy approach, if

$$\Delta D + \lambda \Delta R < 0 \tag{13}$$

we make the substitution and iterate the process until (13) is positive. The trade-off between rate and distortion is set using λ, which is chosen when evaluating the quantization step-size.

The server subsequently encodes the data using a bit-plane context adaptive arithmetic coder to attain rates close to the entropy of the source. The number of retained LSB is also encoded and transmitted with the data. This information is stored by the user for future reference. Note that the number of retained LSB provides a bit-plane significance map, which is further exploited by the entropy coder to reduce the encoding rate.

5. EVALUATION

We evaluate the performance of the proposed centralized and interactive methods using the Tsukuba light field $[272 \times 368 \times 4 \times 4]$[18] and Animal Farm EPI $[232 \times 624 \times 16]$[11] datasets, shown in Fig. 14. Without a loss of generality we only encode the monochromatic component of the data. In the following, we present the compression results for both schemes.

5.1. Centralized method

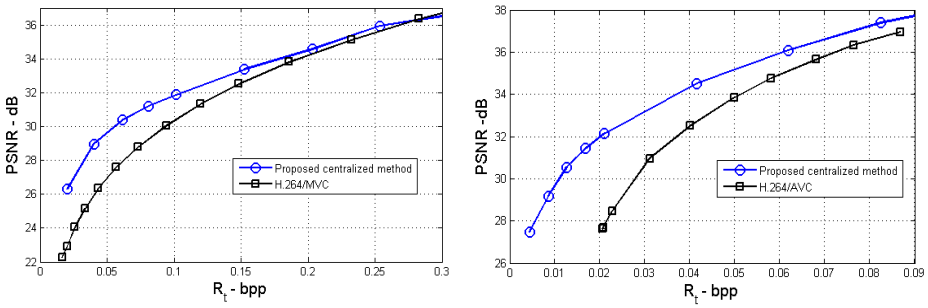
The proposed centralized method is compared to the state-of-the-art H.264/AVC[23] video coding algorithm. To encode the data, the multiview images are treated as a set of video frames along the temporal dimension and they are compressed using the High Profile, Level 2.1. In addition we use the multiview extension (MVC)[24] of the method to encode the Tsukuba light field. Recall that the light field has an additional viewing dimension which is exploited by the MVC method by applying prediction along both viewing dimensions.

We show a quantitative comparison of the method in Fig. 15. Observe that the proposed centralized scheme achieves a significantly better compression performance across the complete range of bit rates with PSNR gains of up to 4dB when encoding the Animal Farm EPI volume. A subjective comparison in Fig. 16 also shows that our approach attains more visually pleasing results than the block-based method.

5.2. Interactive method

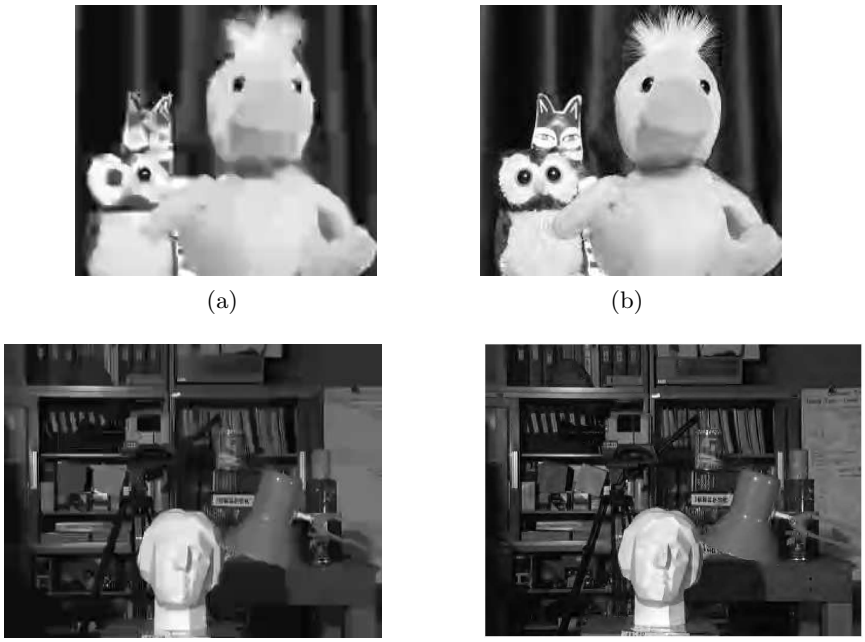
We initialize the proposed interactive method by independently encoding the first image of each layer. Then, the remaining images are randomly chosen and transmitted using the DSC strategy. We compare the method to JPEG 2000[†] which has the same random access capabilities as the proposed approach.

[†]We modify JPEG 2000 to have the same entropy coding as our approach.



(a) Tsukuba light field. (b) Animal Farm EPI volume.

Figure 15. Quantitative comparison of the proposed centralized method with H.264/AVC and MVC. (a) Tsukuba light field dataset. (b) Animal Farm EPI volume.



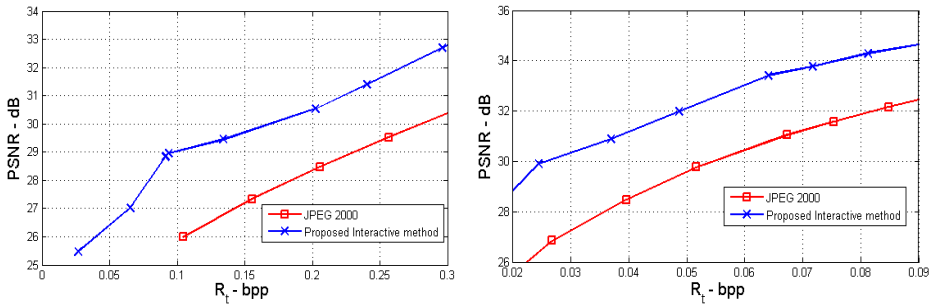
(a) (b)

(c) (d)

Figure 16. Qualitative comparison of the proposed centralized method with H.264/AVC and MVC. (a) Animal Farm encoded using H.264/AVC at 0.024bpp (PSNR 28.93dB). (b) Animal Farm encoded using the proposed centralized method at 0.021bpp (PSNR 32.14dB). (c) Tsukuba light field encoded using H.264/MVC at 0.056bpp (PSNR 27.53dB). (d) Tsukuba light field encoded using proposed centralized method at 0.052bpp (PSNR 29.77dB).

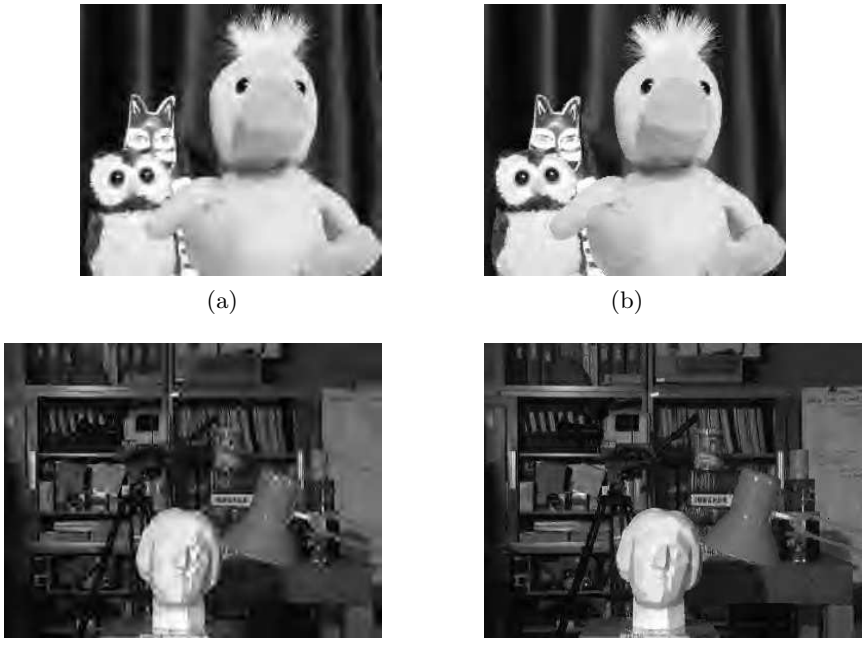
We illustrate a quantitative comparison of the proposed method with JPEG 2000 in Fig. 17. Observe that the proposed method achieves PSNR gains of up to 3dB. This is due to the fact that our approach reduces the inter-view redundancy whereas JPEG 2000 encodes each image independently. The improvement can also be seen in terms of subjective performance in Fig. 18.

On the other hand, observe that in order to facilitate random access, the interactive method trades-off compression performance. In comparison to the centralized algorithm, the PSNR reduction is 3.5dB (at 0.3bpp) and 3dB (at 0.09bpp) when encoding the Tsukuba light field and Animal Farm EPI, respectively.



(a) Tsukuba light field. (b) Animal Farm EPI volume.

Figure 17. Quantitative comparison of the proposed interactive method and JPEG 2000. (a) Tsukuba light field. (b) Animal Farm EPI volume.



(a) (b)

(c) (d)

Figure 18. Qualitative comparison of the proposed interactive algorithm with JPEG 2000. (a) Animal Farm encoded using JPEG 2000 at 0.07bpp (PSNR 31.33dB). (b) Animal Farm encoded using the proposed interactive method at 0.07bpp (PSNR 33.76dB). (c) Tsukuba light field encoded using JPEG 2000 at 0.13bpp (PSNR 26.83dB). (d) Tsukuba light field encoded using the proposed interactive method at 0.134bpp (PSNR 29.44dB).

6. CONCLUSION

In this paper we have presented two compression methods for multiview images. The fundamental component of both algorithms is the layer-based representation which partitions the multiview data into layers, each related to a constant depth in the scene. The first method is a centralized scheme which jointly encodes the images using a multi-dimensional DWT. The transform is implemented in a separable fashion, first by applying a 2D DWT across the viewpoint and then the image dimensions. We modify the viewpoint DWT to efficiently deal with occlusions and depth variations. Although the method attains high compression performance, it requires that all data is transmitted to the user. By contrast, in an interactive communication scenario only a subset of the images is requested by a remote user. We deal with this issue in the second algorithm by using DSC principles to reduce the inter-view redundancy. The approach removes the side-information uncertainty and facilitates random access at the image level. We have shown that the proposed centralized and interactive schemes outperform H.264/MVC and JPEG 2000, respectively.

REFERENCES

1. C. Zhang and T. Chen, "A survey on image-based rendering–representation, sampling and compression," *Signal Processing: Image Communication* **19**(1), pp. 1–28, 2004.
2. M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Transactions on Circuits and Systems for Video Technology* **10**, pp. 338–343, Apr. 2000.
3. B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," *IEEE Transactions on Image Processing* **15**, pp. 793–806, Apr. 2006.
4. A. Gelman, P. L. Dragotti, and V. Velisavljevic, "Multiview image coding using depth layers and an optimized bit allocation," *submitted to IEEE Transactions on Image Processing* .
5. P. Ramanathan and B. Girod, "Random access for compressed light fields using multiple representations," in *Proceedings of IEEE 6th Workshop on Multimedia Signal Processing (MMSP)*, pp. 383–386, Sept. 2004.
6. N.-M. Cheung, A. Ortega, and G. Cheung, "Distributed source coding techniques for interactive multiview video streaming," in *Proceedings of the 27th conference on Picture Coding Symposium (PCS)*, pp. 269–272, IEEE Press, (Piscataway, NJ, USA), 2009.
7. A. Aaron, P. Ramanathan, and B. Girod, "Wyner-ziv coding of light fields for random access," in *Proceedings of IEEE 6th Workshop on Multimedia Signal Processing (MMSP)*, pp. 323–326, Sept. 2004.
8. E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, pp. 3–20, MIT Press, 1991.
9. M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of Computer Graphics (SIGGRAPH)*, pp. 31–42, Aug. 1996.
10. H. Y. Shum and S. B. Kang, "A review of image-based rendering techniques," *IEEE SPIE Visual Communications and Image Processing VCIP* **213**, pp. 1–12, 2000.
11. J. Berent and P. L. Dragotti, "Plenoptic manifolds: Exploiting structure and coherence in multiview images," *IEEE Signal Processing Magazine* **24**, pp. 34–44, Nov. 2007.
12. R. Bolles, H. Baker, and D. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision* **1**, pp. 7–55, March 1987.
13. R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second ed., 2004.
14. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications* **4**(3), pp. 247–269, 1998.
15. A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Transactions on Image Processing* **12**, pp. 1530–1542, Dec. 2003.
16. S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology* **10**, pp. 725–743, Aug. 2000.
17. M. Unser and T. Blu, "Mathematical properties of the JPEG2000 wavelet filters," *IEEE Transactions on Image Processing* **12**, pp. 1080 – 1090, Sept. 2003.
18. D. Scharstein and R. Szeliski, "Middlebury data sets," vision.middlebury.edu/stereo/.

19. H. Freeman, "On the encoding of arbitrary geometric configurations," *IEEE Transactions on Electronic Computers* **EC-10**, pp. 260–268, Jun. 1961.
20. Y. Liu and B. Zalik, "An efficient chain code with huffman coding," in *Pattern Recognition*, **38**, pp. 553–557, Apr. 2005.
21. D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing* **9**, pp. 1158–1170, Jul. 2000.
22. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **36**, pp. 1445–1453, Sep. 1988.
23. "H.264/AVC Video Coding Algorithm." http://x264.nl/.
24. "H.264/MVC Multiview Video Coding Algorithm." CVS - garcon.ient.rwth-aachen.de.