Synthesising Light Field Volumetric Visualizations in Real-time using a Compressed Volume Representation

Seán Bruton, David Ganter and Michael Manzke
Graphics, Vision and Visualisation (GV2), Trinity College Dublin, the University of Dublin, Ireland

Keywords: Volumetric Data, Visualization, View Synthesis, Light Field, Convolutional Neural Network.

Abstract: Light field display technology will permit visualization applications to be developed with enhanced perceptual qualities that may aid data inspection pipelines. For interactive applications, this will necessitate an increase in the total pixels to be rendered at real-time rates. For visualization of volumetric data, where ray-tracing techniques dominate, this poses a significant computational challenge. To tackle this problem, we propose a deep-learning approach to synthesise viewpoint images in the light field. With the observation that image content may change only slightly between light field viewpoints, we synthesise new viewpoint images from a rendered subset of viewpoints using a neural network architecture. The novelty of this work lies in the method of permitting the network access to a compressed volume representation to generate more accurate images than achievable with rendered viewpoint images alone. By using this representation, rather than a volumetric representation, memory and computation intensive 3D convolution operations are avoided. We demonstrate the effectiveness of our technique against newly created datasets for this viewpoint synthesis problem. With this technique, it is possible to synthesise the remaining viewpoint images in a light field at real-time rates.

1 INTRODUCTION

Volumetric information is increasingly used across many domains including civil engineering, radiology and meteorology. Visualization of this volumetric information is an essential part of workflows in these areas, with users tasked with inspecting the visualization for patterns of interest. With the development of advanced display technologies, such as 3D displays and virtual reality, this visual inspection process may be performed in a more natural manner, potentially leading to easier discovery of salient patterns.

These future displays are likely to be enabled by the use of light field technology (Wetzstein et al., 2012; Lanman and Luebke, 2013). Displays employing light fields permit added perceptual quality, with parallax, accommodation and convergence cues making interactions with the displays more compelling and natural. In a medical inspection setting, it has been reported that light field displays add impressive depth perception of structures in 3D scan data, and facilitate easier discrimination by the physician (Agus et al., 2009).

A light field is a function that describes the direction and wavelength of all light rays in a scene (Levoy and Hanrahan, 1996). Imaging technologies for light fields, such as the Lytro Illum camera, simplify this five-dimensional function to four dimensions, allowing capturing of a light field by a discrete rectangular grid of camera lenses. In a virtual setting, this light field formulation can be represented by a 2D grid of aligned cameras, where the image size at each camera is given by the *spatial resolution*, and the number of cameras in the 2D grid is given by the *angular resolution*. Thus, to render such a light field within fixed computational bounds, there exists a trade-off between the spatial resolution and the angular resolution. In this work, we seek to efficiently increase the angular resolution of a light field rendering of volumetric data.

Rendering this volumetric data is itself a non-trivial problem. The volume data can be highly detailed, with multiple dimensions represented, such as temperature and pressure in the case of a meteorological simulation. This richness necessitates advanced rendering techniques to be used (Philips et al., 2018). The choice of methods, such as isosurface extraction, and parameters, such as the transfer function, are dependent on the domain and purpose of the visualization. Furthermore, due to the dense nature of the volume, as opposed to sparse mesh representations, informative visualizations often require expensive ray

Bruton S., Ganter D. and Manzke M.
Synthesising Light Field Volumetric Visualizations in Real-time using a Compressed Volume Representation.
DOI: 10.5220/0007407200960105
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 10th International Conference on Information Visualization Theory and Applications)*, pages 96-105
ISBN: 978-989-758-354-4

tracing techniques to permit observation of salient internal substructures. Thus, rendering multiple visualization frames at real-time rates is very demanding for modest hardware.

For virtual light field cameras, in which the sensors are arranged in a closely-spaced array, the rendered image may change only slightly between different sensors' viewpoints. As such, the rendering power used to produce one light field viewpoint image can be re-used to help produce another. Advances in video interpolation and viewpoint synthesis have shown that such a problem is becoming more tractable, especially with the use of convolutional neural networks that can learn how to estimate the dynamics of scene components between views. In such videos, the natural scene is often composed of opaque objects, and so unique correspondences between pixels in pairs of images can be determined. Such correspondences can be used to calculate optical flow and disparity, which are key inputs of many of these interpolation techniques. However, in the case of volumetric rendering, the use of alpha values in the transfer function violates the optical flow assumption of constant brightness of object points. Furthermore, due to the contribution of multiple object points, with differing alpha values, to a final pixel value, a disparity value between two images cannot be uniquely defined. Thus to obtain a correct pixel value for a generated image, it is necessary to understand the dynamics of multiple dense volumetric scene components.

Another difference between this image interpolation problem and that of natural scenes, is that the volume data is available to us. Using this data as an input could potentially improve the quality of the generated images. However, if a neural network approach to the image interpolation problem is used, it is not clear how best to exploit this volume data. Performing 3D convolutional operations over the typically large volumes would be highly demanding and may preclude the use of this technique as part of a real-time system. Accordingly, we propose the use of a compressed image representation of the volume, via a rank pooling of volume slices. We show that such ranked volume images can be used as part of a neural network approach to improve image synthesis results.

The contributions of this work are as follows:

- We are the first to demonstrate a neural network approach to synthesising light fields for volume visualizations.
- To exploit the availability of the volumetric data, we propose the use of a compressed representation of dense real-valued volumetric data. This avoids the prohibitive cost of 3D convolutional filters for real-time performance and is shown to improve image synthesis results.
- Our neural network approach, generating a light field of angular resolution 6×6, performs at real-time rates, significantly reducing the time taken using traditional ray-tracing techniques.
- We prepare and release two datasets for future use in tackling this problem. All experimental code and data appears in our code repository (https://github.com/leaveitout/deep_light_field_interp).

We envisage that the compressed volume representation may have uses in other volume visualization problems and intend to examine such solutions in future work.

2 RELATED WORK

Convolutional neural networks (CNNs) have undergone a significant renaissance in recent years, and since their application to image classification tasks (Krizhevsky et al., 2012) the nonlinear complexity that can be modelled by a CNN has been shown to be useful for many image generation tasks (Park et al., 2017; Liu et al., 2017; Zhou et al., 2016). Techniques for the generation of images based on an input image, often utilise flow or disparity information. In the closely-related case of light field viewpoint generation, an estimate of disparity has been integrally used to generate viewpoint images using deep learning approaches (Kalantari et al., 2016; Srinivasan et al., 2017). Optical flow has been used as part of novel view synthesis techniques (Zhou et al., 2016) and as part of video interpolation techniques to guide pixel determination from a pair of images (Liu et al., 2017). Another recent technique of video interpolation does not rely on flow information, but rather on learning global convolutional filters that operate on the pair of images to be interpolated (Niklaus et al., 2017). These approaches are developed, however, for the purposes of generating natural images and so do not make use of data being visualized, as in our case.

Convolutional architectures for the specific processing of volumetric data have been developed (Wu et al., 2015; Maturana and Scherer, 2015). These initial works made use of 3D convolutional neural networks as natural extensions of 2D convolutional networks. It was observed, however, that this leads to an increase in convolutional filter size, hence requiring more data for training and precluding deeper networks under a fixed computation budget (Qi et al., 2016). Furthermore, 3D CNNs were outperformed by 2D CNNs trained on multiple views for object recognition tasks (Su et al., 2015). Other techniques for efficient learning from volumetric data have utilised probing techniques (Li et al., 2016), tree representations (Wang et al., 2017b; Riegler et al., 2017; Klokov and Lempitsky, 2017) and point cloud representations (Qi et al., 2017). Each of these approaches assume a binary occupancy volume and have not yet been shown to work with real-valued volumes or indeed multi-dimensional volumes.

Ranking is a commonly used machine learning task encountered in information retrieval. Image features extracted from video frames (Fernando et al., 2017) have been ranked according to their temporal order. The ranking representation allows videos of variable lengths to be encoded into a fixed size descriptor that can then be used for classification tasks, such as action recognition. Recently, this technique has been adapted for use on pixel data (Bilen et al., 2016), with the ranking descriptor taking the form of a single image, a compressed representation of the dynamics of the video. This “dynamic image” representation has been shown to benefit action recognition pipelines when trained on colour, optical flow, and scene flow information (Wang et al., 2017a). Such ranking representations have yet to be applied to volumetric information as part of a visualisation pipeline. We show in this work, that these representations are effective in extracting details from volume data to improve image synthesis quality.

A number of previous works have focussed on the specific topic of rendering volumetric data for light field displays (Mora et al., 2009; Birklbauer and Bimber, 2012). One work looks at volume rendering using emissive volumetric displays and addresses the inherent challenges of using such a display for this purpose (Mora et al., 2009). Volume rendering for light fields has also been performed on other specific light field displays (Agus et al., 2009). This method iteratively refines a rendering when the viewpoint changes and is thus susceptible to slow rendering when the viewpoint changes significantly. Another work speeds up volume rendering for light fields by maintaining a render cache, based on the current viewpoint, that is filled during idle times and used to compose the final renders (Birklbauer and Bimber, 2012). Our work seeks to make light field volume rendering tractable for large display sizes, and independent of viewpoint, by using a neural network approach to increase the angular resolution.

3 METHOD

Here, we detail our approach of synthesising light field viewpoint images using a subset of rendered viewpoints and a compressed volume representation. We first describe how we formulate these compressed volume representations, before describing how it is used in our neural network architecture to synthesise the images.

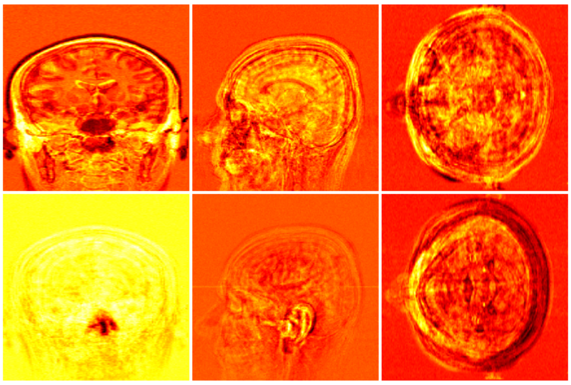

Figure 1: Volume ranked images calculated across entire coordinate axes for a volume dataset of a magnetic resonance imaging scan of a person’s head. The entropy-order images of the top row appear to capture greater detail than appears in the bottom row of axis-order images. In particular, we observe that the axis-order representation devotes more features to discriminating elements such as the nose and ear, and less to internal structures. Here, the images are encoded in a single image, however, it is possible to use more than one image in our final encoding, where we will test whether entropy or order is more effective as part of our synthesis method.

3.1 Volumetric Representation

The rank pooling method of generating a compressed representation of a video sequence is applied to a dense volume to produce the volumetric equivalent of a dynamic image (Bilen et al., 2016). Describing the volume, V, as a sequence of ordered slices $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$, we let $\psi(\mathbf{v}_t) \in \mathbb{R}^d$ be a representation of an arbitrary slice. In our case, we use the slice information itself, and so $\psi(\mathbf{v}_t) = \mathbf{v}_t$. Pairwise linear ranking is characterised by parameters $\mathbf{u} \in \mathbb{R}^d$, such that a ranking function $S(t|\mathbf{u}) = \mathbf{u}^T \cdot \mathbf{v}_t$ produces a score for the slice at t. As such, $\mathbf{u}$ as a matrix that when multiplied by a slice $\mathbf{v_t}$, outputs a score indicating the rank of this slice.

The parameters $\mathbf{u}$ are optimised such that the given ordering of the slices is represented in the scores, i.e. $q > p \implies S(q|\mathbf{u}) > S(p|\mathbf{u})$. Such an optimisation can be formulated as a convex problem and solved using Rank Support Vector Machines (Smola and Schlkopf, 2004):

$$\mathbf{u}^* = \rho(\mathbf{v}_1, \ldots, \mathbf{v}_n; \psi) = \arg\min_{\mathbf{u}} E(\mathbf{u}) \qquad (1)$$

$$E(\mathbf{u}) = \frac{2}{n(n-1)} \times \sum_{q>p} \max\{0, 1 - S(q|\mathbf{u}) + S(p|\mathbf{u})\} + \frac{\lambda}{2}\|\mathbf{u}\|^2 \tag{2}$$

In Equation 1, we see that the optimal representation, $\mathbf{u}^*$, is a function of the slices, $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and the representation type ψ. This can be framed as an energy minimisation of the energy term $E(\mathbf{u})$. The first term in Equation 2 is the hinge-loss soft-counting of the number of incorrectly ranked pairs, and the second term is the quadratic regularization term of support vector machines, where the regularization is controlled by hyperparameter λ. The hinge-loss soft-counter term ensures that there is at least unit score difference between correctly ranked pairs, i.e. $S(q|\mathbf{u}) > S(p|\mathbf{u}) + 1$.

The optimised ranking representation $\mathbf{u}^*$ thus encodes information regarding the ordering of the volume slices and can be seen as the output of a function $\rho(\mathbf{v}_1, \ldots, \mathbf{v}_n; \psi)$ that maps an arbitrarily sized volume to a smaller size representation. Furthermore, this representation is 2D and so 2D CNNs can be used to learn from the volume data, instead of expensive 3D CNNs if we were to learn from the volume directly.

In our volumetric representation, we seek complex features present in the volume to persist in the compressed representation. Thus, in addition to calculating volumetric representations based on a ranking of the order of slices along an axis, we propose an alternative ranking method. This ranking is performed according to the Shannon entropy of the slice, $H = -\sum_i p_i \log p_i$, where p_i is the normalised probability of a specific cell value in a slice. Entropy provides a heuristic of the complexity of the slice and thus the ranking learns to discriminate this aspect, rather than the arbitrarily positive or negative axis order. In Figure 1, we show example images for the two ranking formulations. We shall refer to the compressed volume representations for axis-order ranking and for entropy ranking as Order-Ranked Volume Images (ORVI) and Entropy-Ranked Volume Images (ERVI), respectively.

3.2 Viewpoint Synthesis

In our problem formulation, we assume that a subset of the viewpoint images of the light field are rendered using traditional ray-tracing approaches. We select as this subset the four corner images of the light field array as per Figure 2. We wish to efficiently synthesise all the remaining viewpoint images. In recent work (Kalantari et al., 2016), a convolutional autoencoder approach was shown capable of performing such image synthesis problems for a single viewpoint image in light field photographs. We thus investigate such solutions for our problem of synthesising an entire light field.

By estimating scene geometry, it is possible to generate an entire light field for natural images from a single input image (Srinivasan et al., 2017). However, as a mapping between pixel values in a pair of light field volume renderings with varying alpha values is not injective (one-to-one), disparity or flow-based approaches of image interpolation are not ideal. Hence, our network cannot rely on the presence or calculation of these features.

Instead, we use a more direct approach of interpolating between the rendered images, using a convolutional autoencoder architecture. This architecture is composed of successive dimension reducing encoder blocks and dimension increasing decoder blocks. For the design of these blocks, we follow the proven order of operations for the similar problem of video frame interpolation (Niklaus et al., 2017). In an Encoder block, the order of operations is a convolution with 3×3 filter, Rectified Linear Unit (ReLU) activation ($f(x) = \max(0, x)$), followed by another convolution, ReLU activation, and a final average pooling with 2×2 window and stride of 2. The second convolutional operation maintains an equal number of output channels as the previous convolutional operation. A Decoder block differs in that instead of an average pooling operation, a bilinear upsampling filter is applied to double the output dimensions, followed by convolution and ReLU operations. The convolution operations again maintain the number of output channels. Bilinear sampling has the advantage of being differentiable, thus permitting back propagation through the entire network.

As a baseline, we compose a network architecture that takes as input solely the rendered visualisation images. We refer to this approach as Light Field Synthesis Network - Direct (**LFSN-Direct**). The number of input and output features of the Encoder and Decoder blocks in this architecture, which determine the number of convolutional filters according the above block definitions, are shown in Table 1.

In the architecture, residual connections (He et al., 2016) are used between Encoder and Decoder blocks to encourage less blurry results. Residual connections formulate the intervening operations between a connection as a function, $\mathcal{F}$, that is tasked with learning the change (or residual) required to the input, $\mathbf{x}$, to improve the output $\mathbf{y}$, i.e.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}, \tag{3}$$

rather than being tasked with learning the entire out-

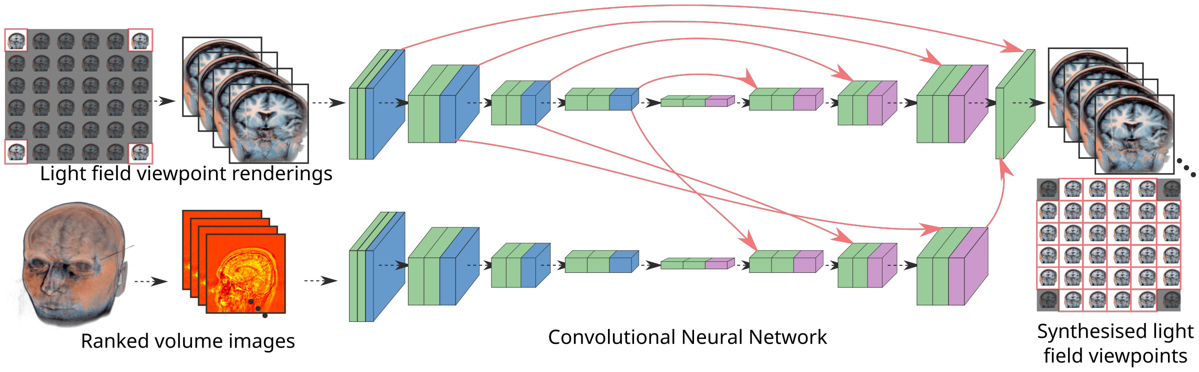
Light field viewpoint renderings
Ranked volume images
Convolutional Neural Network
Synthesised light field viewpoints

Figure 2: An illustration of the convolutional autoencoder architecture used to synthesise light field viewpoint images. One network operates on rendered views, while the other operates on ranked volume images. In the network, green represents a convolutional layer, blue an average pooling layer, purple a bilinear upsampling layer and red arrows represent skip connections.

put. Connections are made between the output of the final convolution of the Encoder block and the final output of the Decoder block, for corresponding block numbers in Table 1.

Table 1: The network structure for the Light Field Synthesis Network. The number of input features of the network corresponds to four RGBA images concatenated along the channel dimension, representing the rendered images. The number of output features of the network corresponds to 32 RGBA images concatenated along the channel dimension, representing the 32 remaining images of the 6×6 light field.

Block	# input features	# output features
Encoder 1	16	32
Encoder 2	32	64
Encoder 3	64	128
Encoder 4	128	256
Decoder 4	256	256
Decoder 3	256	128
Decoder 2	128	128
Decoder 1	128	128

Residual connections are made using an addition operation, and thus it is required that both operands are of the same dimension. As can be seen in Table 1, the number of output features do not match between all corresponding Encoder and Decoder blocks for residual connections. The Decoder blocks have a larger number of output features to facilitate the generation of the larger number of output images. To overcome this, we repeat the outputs of the corresponding convolution operation in the Encoder block along the channel dimension before addition to the output of the Decoder block. For example, the residual connection between Encoder 1 and Decoder 1 requires that the output of the convolution operation is repeated four times. Using the same notation as above, this residual connection can be expressed as

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + [\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x}]. \tag{4}$$

This concatenation of the input may have potential benefits as it would encourage Encoder 1 to learn features that output an **x** that is more generally applicable across the light field due to the joint effect of **x** upon multiple output images.

We introduce a parallel network branch to learn from ranked volume image representations, as shown in Figure 2. This network branch has identical Encoder and Decoder blocks to the LFSN-Direct network, apart from the Encoder 1 block which may have a different number of input features. This is dependent upon the number of ranked volume images used to encode a volume. We refer to this network architecture as **LFSN-ORVI** or **LFSN-ERVI**, depending on whether ORVI or ERVI images are used, respectively. In addition to those described previously, residual connections are used in this architecture to connect the branch processing the rendered images (render branch) to the branch processing the ranked volume images (volume branch). These connections are made between corresponding Encoder blocks in the render branch and Decoder blocks in the volume branch. The final output of the volume branch is added to the output of the render branch. This encourages the volume branch to learn to specifically generate the additional details that cannot be generated from the rendered images alone.

4 IMPLEMENTATION

To demonstrate applicability to a specific use case, we restrict our datasets to two visualizations of a magnetic resonance imaging volume of a human, one of

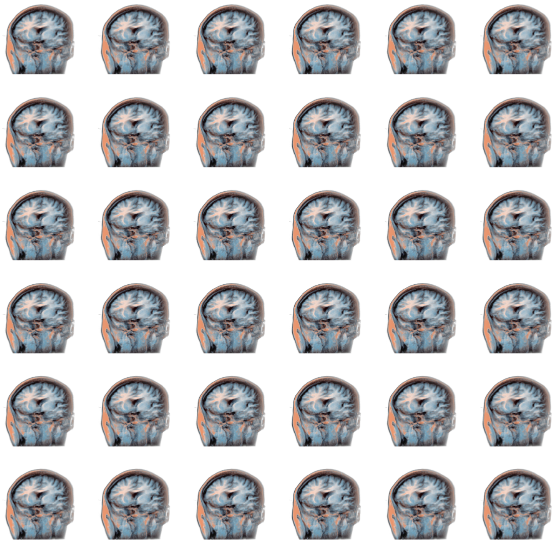

Figure 3: An example light field rendering for the MRI-Head dataset.

the torso and one of the head referred to as MRI-Head and MRI-Torso, respectively. The Interactive Visualization Workshop (Sunden et al., 2015) was used to create each dataset. Virtual cameras were arranged in a planar 6 × 6 array, with optical axes parallel and a fixed spacing between adjacent cameras. 2,000 sets of light field visualizations were collected for each, with spatial resolution of 256 × 256. To expose the internal structures of the volume, and increase the number of alpha-blended image components, the MRI-Head volume was clipped along 400 different planes and 5 light field images were captured at randomized camera locations for each different clipping. The clipping planes were selected as plausible planes for inspection purposes.

The transfer function was kept fixed throughout collection of each dataset. For the MRI-Torso dataset, we include global illumination effects and an extracted isosurface to make the synthesis problem more challenging.

For each dataset, we need to select an axis along which to calculate the ranked volume images. We select the side profile axis for the MRI-Head dataset and the front profile axis for the MRI-Torso dataset. The ERVI images were calculated using the entropy values of volume slices along the chosen axis. Ranked volume images were calculated for sets of adjacent sub-volumes of sizes $16 \times 256 \times 256$ and concatenated to form the compressed representation. This represents a compression rate of 16 times compared to the full volumetric data. Here, a trade-off is made between compressing the volumetric information and minimising the training and inference cost. This cost increases due to the first convolutional layer in the Encoder 1 block in the volume branch, which is proportional to the number of input ranked volume images. Ranked volume image calculated across the entire volume can be seen in Figure 1.

To train the network, we use images from four corner cameras of the light field array as inputs. The network is trained to output the remaining 32 images of the light field of angular resolution of 6 × 6. The dataset was split into 1,600 sets of training images, with the final 400 reserved for testing. The network is trained for a total of 250 epochs based on observations of overfitting if trained further.

To train the network, the Adamax learning rate scheduler was used with the recommended parameters (Kingma and Ba, 2015). The data was transformed to lie in the $[0,1]$ range prior to use during training. Batch normalisation was found to adversely affect results and so is not used. The loss function used was

$$\mathcal{L}_1 = \|I - I_{gt}\|_1 \tag{5}$$

where I and I_{gt} are the synthesised and ground truth light field images respectively.

The Pytorch framework was used for training all the neural network models, and the networks were trained on a Nvidia Titan V card. To improve the speed of training and inference, and to reduce memory requirements, we use half-precision floating point representations for our entire network.

5 EVALUATION

5.1 Qualitative

We inspect the outputs of the three tested approaches to compare their performances qualitatively. We note, expectedly, that internal light field viewpoint images perform worse, with images appearing more blurred as the distance from the input rendered images increases.

In the case of the MRI-Head model, all models perform well and it is difficult to distinguish any rendering artefacts. On inspection of Mean Average Error images, however, we do note that the edges of surfaces are synthesised more accurately with the LFSN-ORVI and LFSN-ERVI models.

In the case of the MRI-Torso model, the LFSN-ORVI and LFSN-ERVI models perform markedly better, as can be seen in Figure 4. The finer details appear sharper overall than the LFSN-Direct approach in these images, however a slight blur effect is noticeable across all three model outputs. This may be due

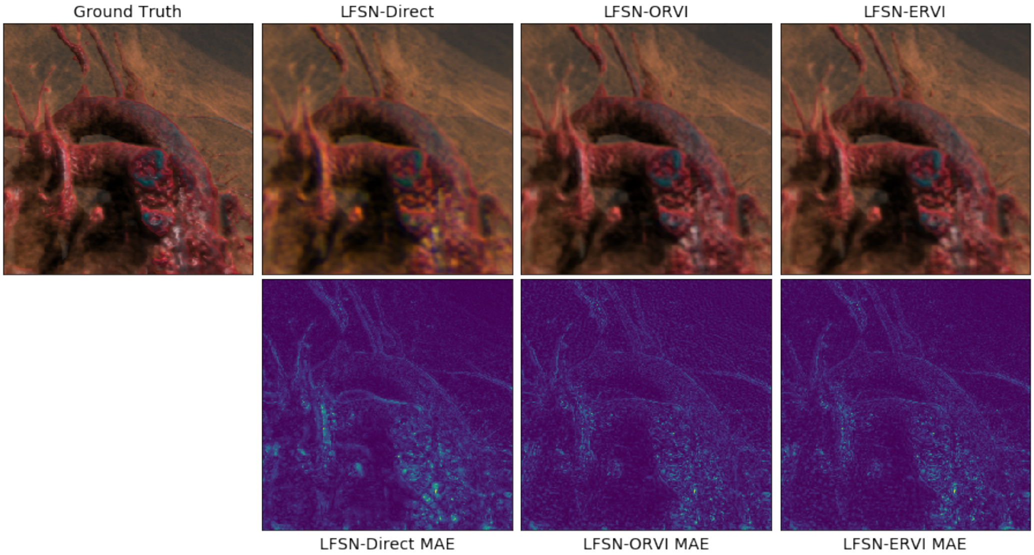

Figure 4: Result images of the viewpoint synthesis methods for the MRI-Torso dataset, generating the light field viewpoint at (3, 3) in the angular dimension. Synthesised images appear slightly blurry compared to the ground truth rendering, specifically the LFSN-Direct result. Inspecting the Mean Average Error (MAE), we see that overall the LFSN-ORVI and LFSN-ERVI produce sharper images, with their errors more contained within specific difficult regions, such as specular areas.

to the use of the L_1 loss for training, as has been observed in other works (Niklaus et al., 2017). Further example output images are shown in Figure 5.

5.2 Quantitative

To evaluate our technique quantitatively, we calculate Structural Similarity (SSIM), Peak Signal to Noise (PSNR) and Mean Squared Error (MSE) metrics for the synthesised test set images against the ground truth.

Table 2: Metrics calculated for the test set of images on MRI-Head dataset.

Network type	SSIM	PSNR	MSE
LFSN-Direct	0.9795	35.98	16.89
LFSN-ORVI	**0.9906**	**40.22**	**6.33**
LFSN-ERVI	0.9886	39.38	7.67

Our results for the MRI-Head dataset are presented in Table 2. It was found that the additional information available via the ranked volume images increased performance, however only marginally. In this case, The axis ordered representation outperforms entropy, which may indicate that it encodes the volume data better when split into smaller sub-volumes.

Table 3: Metrics calculated for the test set of images on MRI-Torso dataset.

Network type	SSIM	PSNR	MSE
LFSN-Direct	0.9182	34.78	27.86
LFSN-ORVI	0.9408	36.87	17.72
LFSN-ERVI	**0.9417**	**36.90**	**17.35**

Our results for the MRI-Torso data are presented in Table 3. This dataset is shown to be more challenging for the LFSN-Direct to synthesise. The presence of isosurface meshes and global illumination are likely responsible for the lower metrics overall. We note that the two ranked volume models outperform the baseline approach to a greater margin in this case. This indicates that the volume branch of the network is effective at learning to generate missing details from the ranked volume representation. We note that in this case, the LFSN-ERVI outperforms LFSN-ORVI. The difference is small, however, and as such further investigation on optimal methods of ordering the volume slices may be warranted.

5.3 Performance Speed

A key aspect to consider is the speed at which images can be generated. For a neural network approach to light field synthesis for volume visualisation to be feasible, it needs to significantly improve upon the time taken to render the light field using a traditional approach. To measure this, we recorded the time to render all light field images in the MRI-Torso dataset. We use the same computer, under similar load, with Nvidia Titan V graphics card used for both the light field synthesis and rendering timings. The approach that we compare against (called Baseline in Table 4)

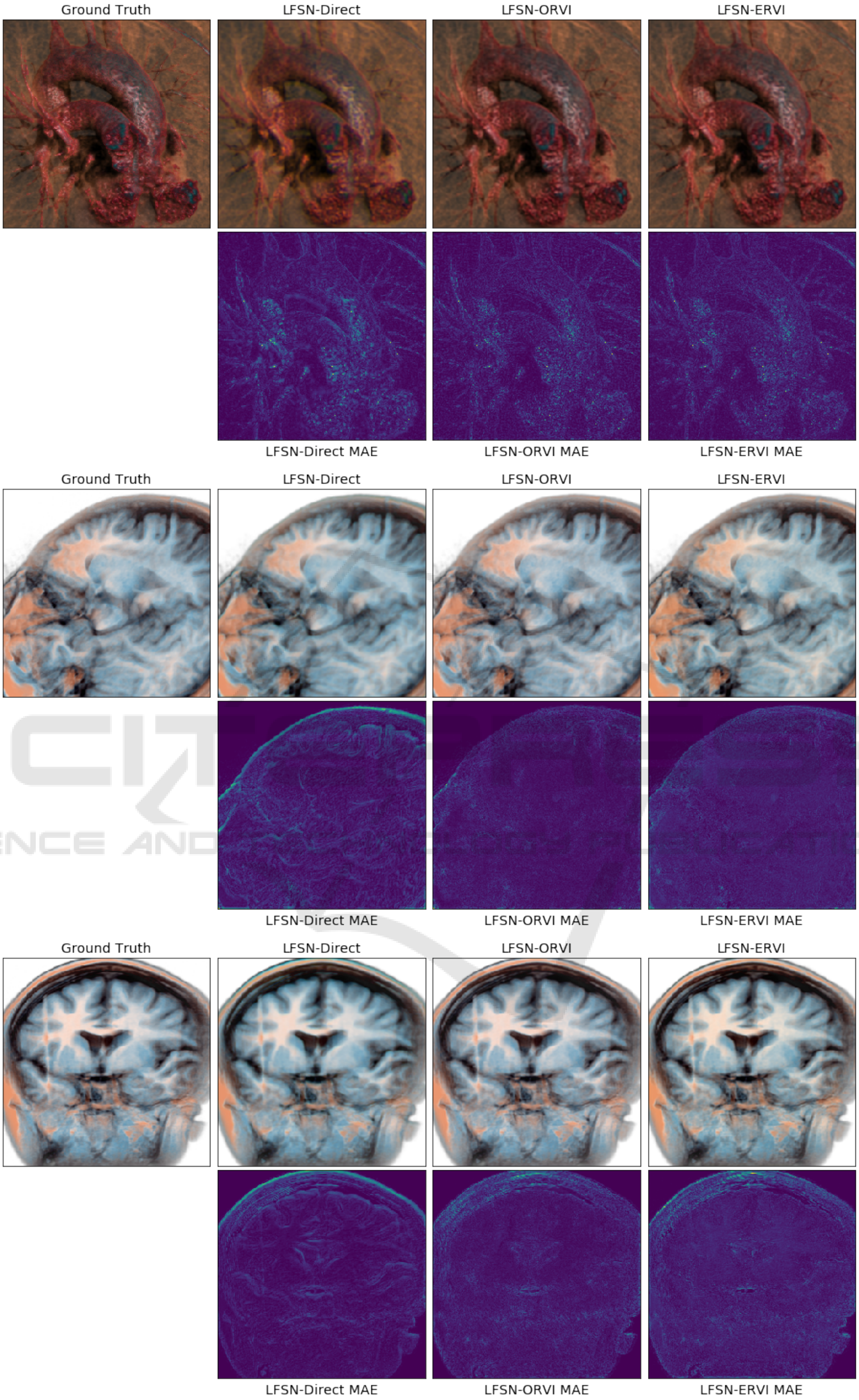

Figure 5: Result images of the viewpoint synthesis methods for the MRI-Torso dataset (top one) and MRI-Head dataset (bottom two), generating the light field viewpoint at (3, 3) in the angular dimension. It is difficult to determine artefacts for the bottom two, however, the Mean Average Error images indicate better performance by LFSN-ORVI and LFSN-ERVI with regards to edges present in the data.

uses ray-tracing as part of the volume rendering, as implemented in the Interactive Visualization Workshop.

Table 4: The timing of the different approaches of outputting an entire light field for the MRI-Torso dataset. The timing is calculated as the average over the entire test set of 400 light fields. The errors are reported as the standard deviation over the test set. Note that we include the time to render the four corner images as part of all LFSN variants. This amounts to a time of 0.08 seconds on average.

Network type	Time (s)
Baseline	0.716 ± 0.223
LFSN-Direct	0.092 ± 0.026
LFSN-ORVI	0.097 ± 0.026
LFSN-ERVI	0.097 ± 0.026

The results in Table 4 show that we substantially reduce the time taken to generate a light field. The LFSN-ORVI and LFSN-ERVI increase frames rates by a multiplier of seven, bringing frame rates to greater than 10 frames per second. Each of the neural network synthesis methods synthesise images in fixed timings due to the network acting as a fixed operation once trained. This property will also be maintained across dataset complexity and size, as along as size of the input images to the network are maintained. In the above results, we note that the majority of time taken for the LFSN variants is spent rendering the four corner images. Thus if further speed improvements are required, we should investigate methods of removing the requirement of these rendered corner images.

6 CONCLUSIONS AND FUTURE WORK

In this work, we proposed the use of convolutional neural networks to synthesise views of dense volumetric data for applications of light field rendering. We propose the use of a ranked volume representation to allow a network access to the available volumetric data to improve synthesis results. We have demonstrated that this approach improves synthesis results, moreso for a more challenging dataset that we collected. A key advantage of this approach is the speed with which a light field can be synthesised, making real-time light field generation achievable. As part of this work we also release the code and data to encourage other researchers to tackle this and other related problems in volume visualization.

We demonstrated our work for visualisation tasks involving medical imaging data. In this domain, there may be concerns with using such a predictive approach for visual inspection of data. For example, the network may synthesise images that are missing identifying properties of a tumour that is present in the data. This represents a key challenge for the type of approach introduced in this work. Future work could address this challenge by improving performance to within an acceptable limit on a range of datasets, or by characterising uncertainty in the synthesised images.

In future work, we wish to explore other approaches of synthesis, perhaps by generating coarse synthesis estimates via geometrical heuristics, prior to refinement with a neural network. The proposed pipeline for visualization is not restricted to light field applications. In future work, we hope to demonstrate the applicability of the technique to frame extrapolation, potentially leading to increased visualization frame rates. Further avenues of investigation include extending the technique to work for time-varying volumetric data.

ACKNOWLEDGEMENTS

The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. This research has been conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number 13/IA/1895.

REFERENCES

Agus, M., Bettio, F., Giachetti, A., Gobbetti, E., Iglesias Guitin, J. A., Marton, F., Nilsson, J., and Pintore, G. (2009). An interactive 3d medical visualization system based on a light field display. *The Visual Computer*, 25(9):883–893.

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., and Gould, S. (2016). Dynamic image networks for action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3034–3042.

Birklbauer, C. and Bimber, O. (2012). Light-field supported fast volume rendering. In *ACM SIGGRAPH 2012 Posters on - SIGGRAPH '12*, page 1, Los Angeles, California. ACM Press.

Fernando, B., Gavves, E., M, J. O., Ghodrati, A., and Tuytelaars, T. (2017). Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):773–787.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Kalantari, N. K., Wang, T.-C., and Ramamoorthi, R. (2016). Learning-based view synthesis for light field cameras. *ACM Trans. Graph.*, 35(6):193:1–193:10.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego.

Klokov, R. and Lempitsky, V. (2017). Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 863–872. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Lanman, D. and Luebke, D. (2013). Near-eye light field displays. In *ACM SIGGRAPH 2013 Emerging Technologies*, SIGGRAPH '13, pages 11:1–11:1, New York, NY, USA. ACM.

Levoy, M. and Hanrahan, P. (1996). Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA. ACM.

Li, Y., Pirk, S., Su, H., Qi, C. R., and Guibas, L. J. (2016). Fpnn: Field probing neural networks for 3d data. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 307–315, USA. Curran Associates Inc.

Liu, Z., Yeh, R. A., Tang, X., Liu, Y., and Agarwala, A. (2017). Video frame synthesis using deep voxel flow. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4473–4481.

Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928.

Mora, B., Maciejewski, R., Chen, M., and Ebert, D. S. (2009). Visualization and computer graphics on isotropically emissive volumetric displays. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):221–234.

Niklaus, S., Mai, L., and Liu, F. (2017). Video frame interpolation via adaptive separable convolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 261–270.

Park, E., Yang, J., Yumer, E., Ceylan, D., and Berg, A. C. (2017). Transformation-grounded image generation network for novel 3d view synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 702–711.

Philips, S., Hlawitschka, M., and Scheuermann, G. (2018). Slice-based visualization of brain fiber bundles - a lic-based approach. pages 281–288.

Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., and Guibas, L. J. (2016). Volumetric and multi-view cnns for object classification on 3d data. *arXiv:1604.03265 [cs]*.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv:1706.02413 [cs]*.

Riegler, G., Ulusoy, A. O., and Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Smola, A. J. and Schlkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

Srinivasan, P. P., Wang, T., Sreelal, A., Ramamoorthi, R., and Ng, R. (2017). Learning to synthesize a 4d rgbd light field from a single image. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2262–2270.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 945–953, Washington, DC, USA. IEEE Computer Society.

Sunden, E., Steneteg, P., Kottravel, S., Jonsson, D., Englund, R., Falk, M., and Ropinski, T. (2015). Inviwo - an extensible, multi-purpose visualization framework. In *2015 IEEE Scientific Visualization Conference (SciVis)*, pages 163–164.

Wang, P., Li, W., Gao, Z., Zhang, Y., Tang, C., and Ogunbona, P. (2017a). Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 416–425.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. (2017b). O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 36(4):72:1–72:11.

Wetzstein, G., Lanman, D., Hirsch, M., and Raskar, R. (2012). Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Trans. Graph.*, 31(4):80:1–80:11.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920.

Zhou, T., Tulsiani, S., Sun, W., Malik, J., and Efros, A. A. (2016). View synthesis by appearance flow. In *Computer Vision - ECCV 2016*, Lecture Notes in Computer Science, pages 286–301. Springer, Cham.