Eurographics Workshop on Visual Computing for Biology and Medicine (2016)
H. Hauser, A. Hennemuth, and A. Lundervold (Editors)

Short Paper

Volume Visualization Using Principal Component Analysis

Salaheddin Alakkari and John Dingliana

Graphics Vision and Visualization Group, Trinity College Dublin, Ireland

Abstract

In this paper, we investigate the use of Principal Component Analysis (PCA) for image-based volume visualization. Firstly we compute a high-dimensional eigenspace using training images, pre-rendered using a standard ray-caster, from a spherically distributed range of camera positions. Then, our system is able to synthesize arbitrary views of the dataset with minimal computation at runtime. We propose a perceptually-adaptive technique to minimize data size and computational complexity whilst preserving perceptual quality of the visualization, in comparison to corresponding ray-cast images. Results indicate that PCA is able to sufficiently learn the full view-independent volumetric model through a finite number of training images and generalize the computed eigenspace to produce high quality images from arbitrary viewpoints, on demand. The approach has potential application in client-server volume visualization or where results of a computationally-complex 3D imaging process need to be interactively visualized on a display device of limited specification.

1. Introduction

Volumetric data comprises three-dimensional (3D) information in the form of a discretely sampled regular grid, or a 3D stack of 2D images such as obtained by MRI or CT scans. In recent years, there has been a trend and a demand to visualize such datasets interactively, so that viewers may peruse the dataset from different viewpoints or based on different viewing parameters. Graphics Processing Units (GPUs) that are becoming integral components in personal computers have made it possible to generate high-fidelity interactive 3D visualizations of such data. However, the complexity of volumetric datasets in science and medicine has continued to increase to the point that, often, the dataset cannot fit in GPU memory or the processing and bandwidth overheads are too high for many of the advanced rendering techniques, such as volume ray-casting, to be achieved in real-time. At the same time, the use of portable computing devices is becoming ubiquitous, leading to a demand for visualization techniques suitable for such platforms that are more constrained than traditional desktop graphical workstations. Amongst other things, this has motivated the development of a number of *client-server* techniques, where a limited front-end client delivers the visualization whilst the bulk of the computational load or memory usage is devolved to a remote high-performance server or, indeed, a distributed source such as the cloud.

In this paper, we investigate the feasibility of using Principal Component Analysis (PCA) to improve the efficiency of visualizing 3D volumetric data on a minimal client device. Our main contribution is a prototype approach that uses PCA to generate a high-dimensional eigenspace capturing a view-independent representation of any 3D volumetric dataset. Arbitrary views of the volume can then be reconstructed *on-demand*, at real-time rates. The efficiency of the eigenspace is improved by a perceptually adaptive cell-decomposition mechanism. Experimental results indicate that this image-based approach can generate high-quality images, whilst reducing computational complexity and data bandwidth, making it particularly suitable for client-server applications.

2. Related Work

The use of PCA for analyzing 3D objects has been well reported in the last two decades in Computer Vision and Computer Graphics. Gong et al. [GMC96] were the first to find the relationship between the distribution of samples in the eigenspace and the actual pose in an image of a human face. Knittel and Paris [KP09] employ a PCA-based technique to find initial seeds for vector quantization in image compression. Fout and Ma [FM07] presented a volume compression method based on transform coding using the Karhunen-Loève Transform (KLT), which is closely related to PCA. Nishino et al. [NSI99] proposed a method, called *Eigen-texture*, which creates a 3D image from a sample of range images using PCA. They found that partitioning samples into smaller cell-images improved the rendering of surface-based 3D data. However this has not yet been applied to volume data, which poses additional challenges as the rendered image typically exposes interior details that need to exhibit consistent occlusion and parallax effects.

Many remote visualisation techniques have been proposed in the scientific and medical visualization literature. The motivation for these range from facilitating collaborative multi-user systems [KBK*16], performance improvements through distributed parallel rendering [FK05], web-based visualization on browsers [PA*01],

DOI: 10.2312/vcbm.20161271

remote collaborative analysis by distant experts [SMD*13] and to achieve advanced rendering on low-spec client devices [MW08]. One strategy, in client-server volume rendering is to transmit 3D data on-demand to the client, after compression [MW08], partitioning [Bet00] or using a progressive rendering [CBPS06]. The client in these approaches is required to do further processing to render the data. A second alternative, such as in [EE99], is for a high-end server to remotely render the data and transmit only images to the client, which has a much reduced responsibility of simply displaying the pre-rendered image. This strategy, often referred to as *Thin Client*, is a popular approach for visualization on portable devices such as a mobile tablets, which may be restricted in terms of computational capacity and GPU components.

In between these ends of the spectrum, some image-based approaches pre-compute intermediate 2D images that are post-processed or composited by the client before display [QT05, Bet00, TCM10]. Image-based approaches, in general, have been of interest, for improving the efficiency of volume visualization [CS98,CKT01,MPH*05]. At the cost of some additional computational load on the client, such a solution may provide improvements such as reduced latency during interaction and it is in this category that our contributions lie.

3. PCA-based Volume Visualization

In this section, we define the basic methodology by which we exploit PCA within an image-based solution for interactive volume visualisation. Although the main motivation is to provide an alternative to a thin-client solution, the approach could also have advantages where run-time rendering complexity needs to be reduced at the cost of pre-processing computations. We first presented the core of the approach in [AD16], however, for completeness, the methodology is outlined below with a summary of initial results in Section 4. In the rest of the paper, we discuss how we extend upon the original work by including a novel adaptive decomposition technique, and a means of improving the trade-off between data reduction and image quality by employing a perceptual metric. Furthermore, we provide a more detailed analysis including the use of larger image resolutions of an order that is likely to be used in practice.

The basic approach to PCA is as follows. Given data samples $X = [x_1\, x_2 \cdots x_n] \in R^{d\times n}$, where each sample is in column vector format, the covariance matrix is defined as

$$C = \frac{1}{n-1} XX^T = \frac{1}{n-1} \sum_{i=1}^{n} x_i x_i^T. \tag{1}$$

We can find the optimal low-dimensional bases that cover most of the data variance by extracting the most significant eigenvectors of the covariance matrix C. Eigenvectors are extracted by solving the following characteristic equation

$$(C - \lambda I)\, v = 0;\, v^T v = 1, \tag{2}$$

where $v \in R^d$ is the eigenvector and λ is its corresponding eigenvalue. Eigenvalues describe the variance maintained by the corresponding eigenvectors. Hence, we are interested in the subset of eigenvectors that have the highest eigenvalues $V = [v_1\, v_2 \cdots v_p]$; $p \ll n$. Then we encode a given sample x using its p-dimensional projection values (referred to as *scores*) as follows

$$y = V^T x. \tag{3}$$

We can then reconstruct the sample as follows

$$x_{reconstructed} = V y. \tag{4}$$

One advantage of PCA is the low computational complexity when it comes to encoding and reconstructing samples.

For ray-cast volume rendering, where the final image is highly dependent on the viewing angle, we assume a set of rendered images as training samples. Each image is considered a high-dimensional vector and used as input into Equation 1. We can compute the eigenspace of the volume dataset by applying PCA to a number of training images of uniformly distributed viewing angles. By interpolating the scores of training samples in the eigenspace, we can synthesize output samples from novel viewing angles using the interpolated scores.

Figure 1 illustrates the steps for reconstructing and rendering a novel view image using two alternative PCA techniques: *Standard PCA*, as previously described, and a variant referred to as *Cell-based PCA*, where the viewport is partitioned into rectangular cells and the eigenspace computed for each cell image individually. In the latter approach, we can encapsulate the whole 3D model using a small number of eigenimages of reduced size, thereby increasing the precision of each eigenspace to capture the localised variance. Computational complexity is reduced in both cases, since the final image is effectively obtained by a simple weighted sum of eigenimages, which are much fewer in number than, for instance, the average sampling rate in a ray-caster. Furthermore, it should be noted that the cell-based technique has similar computational complexity and memory footprint as the direct PCA technique as we essentially perform a larger number of much smaller iterations.

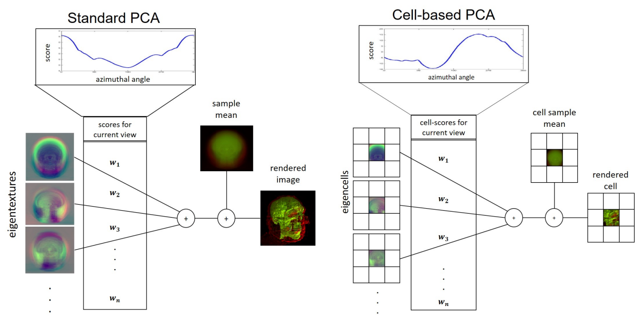

Figure 1: *Overview of our technique for image reconstruction using PCA. Left: Standard PCA; Right: Cell-based PCA*

4. Feasibility Study

As a first test, we applied the two PCA techniques mentioned above for visualizing the *VisMale Head* from the Visible Human dataset, at a resolution of 300×300 pixels. We use 1,500 training images from uniformly-spaced viewing angles (3.6° spacing for the azimuthal angle and 12° spacing for the elevation angle) to compute the eigenspace. These are generated using an implementation of a standard GPU ray-caster based on [HKRs*06] with sampling rate of 1,000 samples per ray. We then acquire test samples by applying 0.9° spacing for the azimuthal angle and 30° spacing for the elevation angle leading to a total of 2,400 unique views.

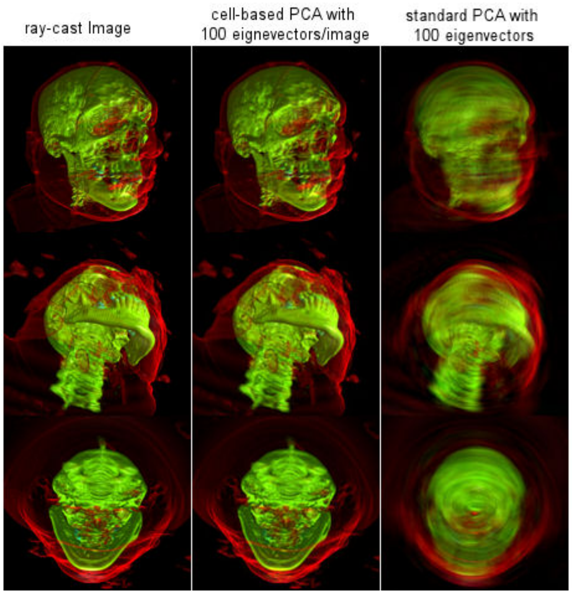

Figure 2: *Three novel views rendered using PCA.*

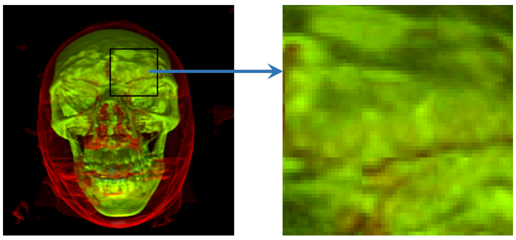

Figure 3: *Zoomed-in view of edge-discontinuities in cell-based PCA for a* 300×300 *resolution rendering*

For Standard PCA, the eigenspace is computed for the full-size training images. In the case of Cell-based PCA, we partition each image into a number of equally-sized cell images (20×20 pixels). Then, we compute the eigenspace of each cell image individually. We use 100 eigenvectors to represent the eigenspace and, for each unique view, we synthesize the corresponding scores (projection values into the first 100 significant eigenvectors). Figure 2 compares the reconstructed novel view images for both Standard PCA and Cell-based PCA with a ray-cast rendering from the corresponding view. Clearly, the cell-based approach produces much better quality results compared to the somewhat blurry images resulting from the standard technique with the same distribution of training samples, consistent with what was reported in the previous literature [NSI99]. However the cell-based PCA results in subtle discontinuity artefacts at the cell boundaries in the reconstructed images (see Figure 3). In terms of complexity, the PCA based methods, in the test scenarios, require only 100 scalar-vector multiplications at run-time, which is computationally much cheaper compared to the operation required in the equivalent ray-cast rendering.

5. Adaptive-cell PCA

Once we established that Cell-based PCA was capable of achieving reliable view reconstructions of volume data, we were interested in whether the approach scales well to larger image resolutions as might be used in practical applications; and whether adaptively varying the eigenvectors per cell could lead to a more optimal tradeoff between performance and quality.

For this study, we apply PCA for visualizing the VisMale *Head* and Tooth datasets. The latter, obtained from the Volume Library of Stefan Roettger, is chosen in order to determine how the approach scales to a dataset of different complexity, in this case lower voxel resolution and a structurally simpler object. The training samples are acquired using a similar method as before, but at a resolution of 1080×1080 pixels and a cell-size of 30×30. Unlike in the previous section, where all cells had the same number of eigenvectors, we now adapt this number as required for each cell, and refer to this approach as *Adaptive-cell PCA*. The number required is determined based on total variability encoded by the first n eigenvectors. This can be expressed as follows

$$\Theta = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{N} \lambda_j} > T,$$

where λ is the eigenvalue, n is the number of first significant eigenvectors, N is the total number of eigenvectors (in our case $30 \times 30 = 900$) and T is a threshold value, which affects the trade-off between high variability and low mean number of eigenvectors per cell.

In order to choose an appropriate threshold, the computed eigenspaces are tested by reconstructing 100 random views and comparing these to the original sample images using a perceptual similarity metric, HDR-VDP-2 [MKRH11]. Using this metric we compute quality values (Q-values) which should be above 50% in order to ensure an acceptable resemblance between a reference image and a test image. The choice of threshold is made manually, so that it will result in a Q-value conservatively above 50% whilst minimizing the number of eigenvalues.

Table 1 shows the threshold and mean number of eigenvectors across all cells for each dataset. Note that the Tooth dataset has higher threshold value and lower number of eigenvectors, due to its lower complexity and detail compared to the Head dataset. Figure 4 shows two novel views from each dataset reconstructed using Adaptive-cell PCA. We can see that the use of the perceptual metric to define the threshold, T, ensures that artefacts, including those on cell-boundaries, are visually reduced at normal viewing distances.

Next, we consider how our approach would compare, in terms of bandwidth, to the alternative of a thin-client solution that performs all rendering on the server, transmitting only images that are simply displayed on the client. Once initial eigenimages are downloaded after precomputation, at run-time our approach needs to simply transmit scores that determine how the eigenimages are combined. For the VisMale Head dataset this entails approximately

Table 1: *Average number of eigenvectors per cell and threshold value for each dataset, which is conservatively chosen to ensure a high perceptual similarity based on the HDR-VDP-2 metric.*

Dataset	VisMale head	Tooth
volume dimensions	$128 \times 256 \times 256$	$140 \times 120 \times 161$
mean number of dimensions per cell	46.27	26.4
T	98%	99.7%
Mean Q-value of PCA	60%	62.2%
Mean Q-value of equivalent JPEG	66.8%	60.1%

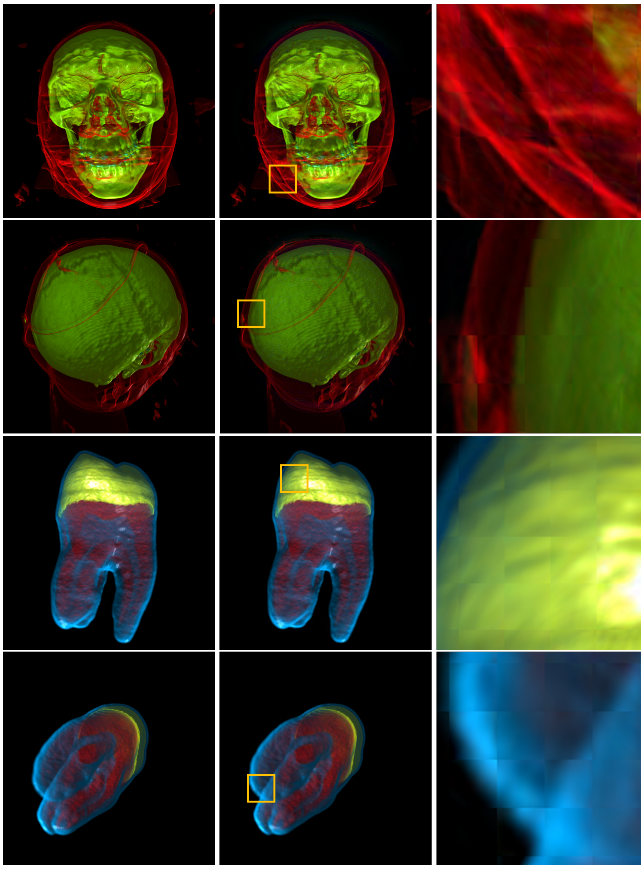

Figure 4: *Novel views of the Head and Tooth dataset visualized at 1080p resolution using the adaptive cell PCA technique. For each row, a ray-cast rendering (left) is compared to a reconstructed view (middle). The two images are perceptually indistinguishable at normal viewing size but cell-boundary artefacts are visible in some areas when closely zoomed in (right).*

$46.27 \times 36 \times 36 =$ 59kByte of information, resulting in an image with an average Q-value of approximately 60% based on the HDR-VDP-2 metric. A JPEG image of the same resolution, with a compression factor set to achieve a similar file size, was found to have a slightly higher average Q-value of 66.8%. For the Tooth dataset, PCA did a bit better on average with a Q-value of 62.2%, compared to 60.1% for the equivalent-sized JPEG. Due to the relative simplicity of the Tooth, PCA was able to accurately encode the information with far fewer eigenimages, resulting in a bandwidth equivalent to a 36kByte JPEG. It should be noted that the transmitted scores themselves could be losslessly compressed to reduce bandwidth further.

In order to visually compare with typical volume rendering systems, we used a real-time volume ray-caster to generate the training images. However, it should be noted that an advantage of our approach is that the run-time cost of generating images is independent of the computational complexity of the rendering process or the dataset. For instance, Figure 5 shows a proof-of-concept reconstruction, using adaptive-cell PCA, of a chest dataset rendered at 1080×1080 resolution using the Exposure Renderer [KPB13], which achieves interactive progressive rendering by exploiting high-end GPUs. Our approach can be used to efficiently recreate such detailed images in high detail at real-time, even on a display device without a powerful GPU. The eigenspace computation for the results reported in the paper is considerably expensive, of the order of several hours for a typical dataset running on a rudimentary implementation on Matlab [Mat15]. There are a number of alternate approaches in the literature, such as [SP07], which could be used in future work to speed up the calculation of the eigenspace. However, we have not yet conducted a systematic study of such alternatives as this phase occurs as a once-off pre-process that we assume will be conducted offline.

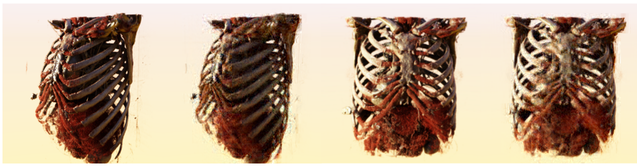

Figure 5: *Sample views of PCA reconstruction of photo-realistic volume renderings. Left image in each pair is rendered by Exposure Render [KPB13], right is an Adacptive-cell PCA reconstruction.*

6. Conclusions and Future Work

In this paper, we presented a preliminary investigation into the use of PCA for volume visualization. The main contribution is an adaptive, cell-based decomposition technique that is able to reconstruct, in real-time, any volumetric model through a finite number of training images and generalize the eigenspace to produce high quality novel view images. We used the HDR-VDP-2 metric in the process of determining the optimal parameters to ensure minimal perceptual error, trading off compression and quality. A limited comparison to the equivalent of a thin-client, receiving JPEG compressed images from a server, indicated that the approach can achieve reasonable bandwidth reduction. It should be noted that the threshold for compression was chosen conservatively to guarantee high-quality images. In theory, an automated iterative process could potentially optimize the trade-off even further. Although the approach requires further computation to reconstruct the image, its run-time performance and capabilities are essentially independent of the complexity of the rendering process or of the volume data resolution. Thus, it could even be used in a minimally-spec standalone system to allow interactive rendering of high-resolution volumetric data, or data that has been visualised using complex rendering techniques not normally possible in real-time.

One clear limitation when using PCA for rendering is that a change in transfer function (material colors and opacities) currently requires a change in the whole eigenspace. The process of computing the eigenspace is quite computationally expensive but, where a suitable transfer function can be assumed, this is done in a preprocessing step for the dataset. One potential solution to this, which we would like to investigate in the future, might be to combine eigenspaces of different materials (e.g. flesh, bone, etc) using composition techniques such as image level intermixing [SS11]. Previous authors have demonstrated that transfer function changes can be supported in a view-dependent image-based technique by multilayering [TCM10], and it would be interesting to see if similar strategies could be used to extend our approach.

In future work, we also plan to conduct perceptual user studies to evaluate our results which are currently based purely on the HDR-VDP-2 perceptual metric. Based on these results, we plan to investigate strategies to further reduce the artefacts appearing in the cell boundary regions. More exhaustive practical tests with varying cell sizes, more complex datasets and more advanced rendering techniques to study trade-off between compression and quality, will help us determine the optimal conditions for the approach.

Despite its limitations, PCA appears to be an interesting and viable alternative technique for image-based volume visualization. The benefits in terms of computational complexity and compression of information may lead to potential advantages in applications such as client-server visualization systems.

Acknowledgement

This research has been conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number 13/IA/1895. The VisMale Head dataset is courtesy of the Visible Human Project at the U.S. National Library of Medicine. The Tooth was obtained from the Volume Library of Stefan Roettger. The Chest was obtained from the DICOM sample datasets provided by OsiriX.

References

[AD16] ALAKKARI S., DINGLIANA J.: Volume Rendering Using Principal Component Analysis. In *EuroVis 2016 - Posters* (2016), Isenberg T., Sadlo F., (Eds.), The Eurographics Association. 2

[Bet00] BETHEL W.: Visualization dot com. *IEEE Computer Graphics and Applications 20*, 3 (2000), 17–20. 2

[CBPS06] CALLAHAN S. P., BAVOIL L., PASCUCCI V., SILVA C. T.: Progressive volume rendering of large unstructured grids. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1307–1314. 2

[CKT01] CHEN B., KAUFMAN A., TANG Q.: *Image-Based Rendering of Surfaces from Volume Data*. Springer Vienna, Vienna, 2001, pp. 279–295. 2

[CS98] CHOI J.-J., SHIN Y. G.: Efficient image-based rendering of volume data. In *Computer Graphics and Applications, 1998. Pacific Graphics '98. Sixth Pacific Conference on* (Oct 1998), pp. 70–78, 226. 2

[EE99] ENGEL K., ERTL T.: Texture-based volume visualization for multiple users on the world wide web. In *Virtual Environments*. 1999, pp. 115–124. 2

[FK05] FRANK S., KAUFMAN A.: Distributed volume rendering on a visualization cluster. In *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)* (Dec 2005). 1

[FM07] FOUT N., MA K. L.: Transform coding for hardware-accelerated volume rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1600–1607. 1

[GMC96] GONG S., MCKENNA S., COLLINS J. J.: An investigation into face pose distributions. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on* (1996), IEEE, pp. 265–270. 1

[HKRs*06] HADWIGER M., KNISS J. M., REZK-SALAMA C., WEISKOPF D., ENGEL K.: *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006. 3

[KBK*16] KOHLMANN P., BOSKAMP T., KÖHN A., RIEDER C., SCHENK A., LINK F., SIEMS U., BARANN M., KUHNIGK J.-M., DEMEDTS D., HAHN H. K.: *Remote Visualization Techniques for Medical Imaging Research and Image-Guided Procedures*. Springer International Publishing, Cham, 2016, pp. 133–154. 1

[KP09] KNITTEL G., PARYS R.: PCA-based seeding for improved vector quantization. In *Proceedings of the First International Conference on Computer Imaging Theory and Applications (VISIGRAPP 2009)* (2009), pp. 96–99. 1

[KPB13] KROES T., POST F. H., BOTHA C. P.: Exposure Render: An interactive photo-realistic volume rendering framework. *PLoS ONE 8* (2013). 4

[Mat15] THE MATHWORKS, INC.: *MATLAB version 8.5.0.197613 (R2015a)*. Natick, Massachusetts, 2015. 4

[MKRH11] MANTIUK R., KIM K. J., REMPEL A. G., HEIDRICH W.: Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 40. 3

[MPH*05] MEYER M., PFISTER H., HANSEN C., JOHNSON C., MEYER M., PFISTER H., HANSEN C., JOHNSON C.: *Image-Based Volume Rendering with Opacity Light Fields*. Tech. rep., University of Utah, 2005. 2

[MW08] MOSER M., WEISKOPF D.: Interactive volume rendering on mobile devices. In *Vision, Modeling, and Visualization VMV* (2008), vol. 8, pp. 217–226. 2

[NSI99] NISHINO K., SATO Y., IKEUCHI K.: Eigen-texture method: Appearance compression based on 3D model. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* (1999), vol. 1, IEEE. 1, 3

[PA*01] POLIAKOV A. V., , ALBRIGHT E., CORINA D., OJEMANN G., MARTIN R., BRINKLEY J.: Server-based approach to web visualization of integrated 3D medical image data. In *Proceedings of the AMIA Symposium* (2001), pp. 533–537. 1

[QT05] QI X., TYLER J. M.: A progressive transmission capable diagnostically lossless compression scheme for 3D medical image sets. *Information Sciences 175*, 3 (2005), 217–243. 2

[SMD*13] SANTHANAM A., MIN Y., DOU T., KUPELIAN P., D.A. LOW D. A.: A client-server framework for 3D remote visualization of radiotherapy treatment space. *Frontiers in Oncology 3* (2013). 2

[SP07] SHARMA A., PALIWAL K. K.: Fast principal component analysis using fixed-point algorithm. *Pattern Recogn. Lett. 28*, 10 (July 2007), 1151–1155. 4

[SS11] SCHUBERT N., SCHOLL I.: Comparing GPU-based multi-volume ray casting techniques. *Computer Science - Research and Development 26*, 1 (2011), 39–50. 5

[TCM10] TIKHONOVA A., CORREA C. D., MA K.-L.: Explorable images for visualizing volume data. In *IEEE Pacific Visualization Symposium* (2010), pp. 177–184. 2, 5