Exploiting Eigenvalues of the Hessian Matrix for Volume Decimation

Jiří Hladůvka, Andreas König, and Eduard Gröller

Institute of Computer Graphics and Algorihms

Vienna University of Technology

Karlsplatz 13/186, A-1040 Wien, Austria

http://www.cg.tuwien.ac.at/home/

e-mail: {hladuvka | koenig | groeller}@cg.tuwien.ac.at

ABSTRACT

In recent years the Hessian matrix and its eigenvalues became important in pattern recognition. Several algorithms based on the information they provide have been introduced. We recall the relationship between the eigenvalues of Hessian matrix and the 2nd order edge detection filter, show the usefulness of treating them separately and exploit these facts to design a combined threshold operation to generate sparse data sets.

Keywords: Volume Rendering, Sparse data, Hessian matrix, Eigenvalues, Laplacian filter.

1 INTRODUCTION

A common approach to analyze the local behaviour of a 2D/3D image I is to consider its Taylor expansion in the neighborhood of a point x_0

$$I(x_0+\Delta x) \approx I(x_0)+\Delta x^T \nabla I(x_0)+\Delta x^T H(x_0)\Delta x \tag{1}$$

where ∇I is the gradient vector and H denotes the Hessian matrix – a matrix built of the second partial derivatives of I.

$$H = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \tag{2}$$

with $I_{ab} = \frac{\partial^2 I}{\partial a \partial b}$ defines the Hessian matrix for the 3D image (volume).

1.1 The Hessian matrix and its eigenvalues

Expansion (1) plays a crucial role in filter design [Mölle97], [Theuß00] and, up to second order, locally approximates *the structure of the image* [Frang98]. Its components, both the gradient and the Hessian matrix, can also be used separately. The gradient vector is widely used as a normal to an implicitly defined isosurface. Its magnitude provides us with an edge detector and can be used as a modulation factor in 3D imaging [Levoy88]. The usefulness of the Hessian matrix is less known. Its elements approximate 2nd order derivatives, and therefore encode the shape information – both a qualitative and quan-titative description of how the normal to an isosurface changes. Haar Romeny et al. [Haar 94] exploit the transformation of an image into so–called Gauge coordinates to find a compact expression for Gaussian curvature of the intensity surfaces given by 2D images:

$$K = \frac{detH}{1 + \|\nabla I\|^2} \tag{3}$$

For 3D images, the expressions for Gaussian and mean curvatures are more complicated, nevertheless still expressed in terms of elements of the Hessian matrix and the gradient vector.

Particularly interesting are the eigenvalues and eigenvectors of the Hessian matrix. Sato [Sato98] and Frangi [Frang98] independently employ eigenvalues to design a filter for vessel enhancement in 3D medical digital images. Two years later Sato et al. [Sato00] generalize the previously introduced concept to enhance tubular, blob, and sheet–like structures. This approach, however, implements the concept of a multifeature transfer function, and leaves the user to set the parameters in 5D space (I, ∇I, and three functions of the eigenvalues) or even, including segmentation information, in 6D space.

1.2 Sparse volumes

Sparse volumes can be, when appropriately re-organized, displayed at interactive frame rates without hardware acceleration. There are different techniques both to generate reasonable sparse data sets and to display them. Saito [Saito94]

proposes a non–uniform Poisson–disk–sampling. Each voxel of the generated sparse data set is represented by a simple 3D entity like a point, line, cross or a triangle, and is passed to conventional rendering technology achieving real–time results. This approach can be considered as an interactive non–realistic previewing. The algorithm introduced by Mroz et al. [Mroz00] decimates a volume with respect to the visualization technique to be applied - MIP. In this case back projection is used. Other real–time, but still high–quality back projection techniques have been introduced in [Pfist00], and [Rusin00].

1.3 Outline of the presented work

We introduce a framework for using eigenvalues of the Hessian matrix to reduce volume data. We show how the eigenvalues are related to the Laplacian edge filter, demonstrate the suitability of thresholding them for feature detection, and propose a technique resulting in a sparse volume.

Rather then back–projecting sparse volumes to benefit from high frame rates, in this work we concentrate on a visual comparison of ray–traced original volumes vs. sparse volumes generated using our novel approach. Backprojection/splatting display with frame rate evaluation is left as a subject to future work.

2 METHOD

2.1 Relation of eigenvalues to Laplacian

The Hessian matrix H, as a real–valued and *symmetric* matrix, has real–valued eigenvalues. Finding the eigenvalues and eigenvectors of the Hessian matrix is closely related to its decomposition

$$H = PDP^{-1} \qquad (4)$$

where P is a matrix of H's eigenvectors and D is a diagonal matrix having H's eigenvalues at the diagonal. Equation (4) is a similarity transformation under which the trace of matrix, $Tr(X) = \sum_{i=1}^{n} X_{ii}$, is an invariant. Putting these facts together for a 3D image, we get

$$\begin{aligned} \lambda_1 + \lambda_2 + \lambda_3 &= Tr(D) = Tr(PDP^{-1}) = \\ &= Tr(H) = I_{xx} + I_{yy} + I_{zz} = \\ &= L \end{aligned} \qquad (5)$$

The rightmost term denotes the Laplacian filter and the equation (5) puts it into relationship with the eigenvalues of the Hessian matrix.

The Laplacian filter and its variant, the Laplacian of Gaussian (LoG), provides the image processing community with an isotropic edge detection filter [Jähne97]. The importance of edge information for machine vision is usually motivated from the observation that under rather general assumptions about the image formation process, a discontinuity in image brightness can be assumed to correspond to a discontinuity in either depth, surface orientation, reflectance or illumination [Linde96]. The importance of Laplacian and LoG for bioperception has been emphasized by the work of Marr [Marr82]

2.2 A simple way of using eigenvalues

Given the Hessian matrix eigenvalues, equation (5) shows how to combine them into the Laplacian operator. Experimenting with eigenvalue images, however, we have found that treating them separately rather than adding them provides us with information more suitable for thresholding. Fig. 1 and further examples at our project web page [Hladů00] demonstrate this property for 2D images. For the decimation of 2D images we propose a combined thresholding as follows:

$$I_{new}[p] = \begin{cases} I[p] & \text{if } \lambda_1[p] \geq T_1 \vee \lambda_2[p] \leq T_2 \\ 0 & \text{otherwise} \end{cases}$$
$$(6)$$

where $p = [x, y]$ denotes the coordinates of a pixel.

For a 3D image there are three eigenvalues, $\lambda_1 \geq \lambda_2 \geq \lambda_3$, for each voxel. Fig. 2 shows axial slices of eigenvalue volumes computed from a CT head data set. Since, in our experience, images corresponding to λ_2 always lack good contrast, we exclude them from the decimation process and specify a combined thresholding just with the two remaining eigenvalues λ_1, λ_3:

$$V_{new}[v] = \begin{cases} V[v] & \text{if } \lambda_1[v] \geq T_1 \vee \lambda_3[v] \leq T_3 \\ 0 & \text{otherwise} \end{cases}$$
$$(7)$$

where $v = [x, y, z]$ denotes the coordinates of a voxel.

Definitions (6) and (7) preserve just those pixels/voxels of the original data with salient features. This can be seen from the fact, that finding eigenvectors of the Hessian matrix is related to finding the directions with extreme values of the second derivatives, i.e., directions of extreme normal–to–isosurface change.

The only task of the user is to specify, in general, two threshold values (Fig. 2). In our experience, however, the extreme eigenvalues exhibit a symmetry $\lambda_1[v] \approx -\lambda_3[v]$ in edge areas, so the thresholds can be specified symmetrically in the initial step with the possibility of non symmetric refinement in the final step.
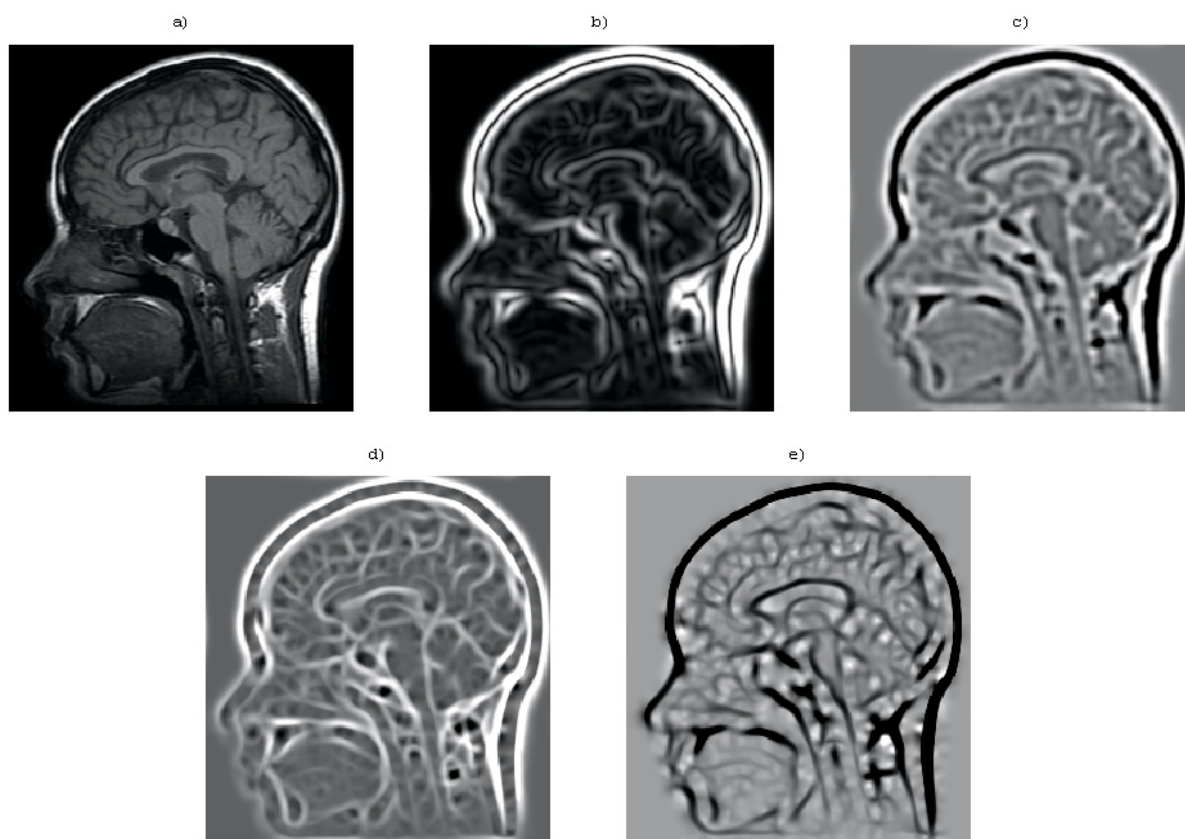
Figure 1: An example of an MRI image. a) original I, b) gradient magnitude $\|\nabla I\|$, c) Laplacian operator applied to an image, d) λ_1 image, and e) λ_2 image. The Laplacian image corresponds to the sum of eigenvalue images, which can be thresholded for salient features.
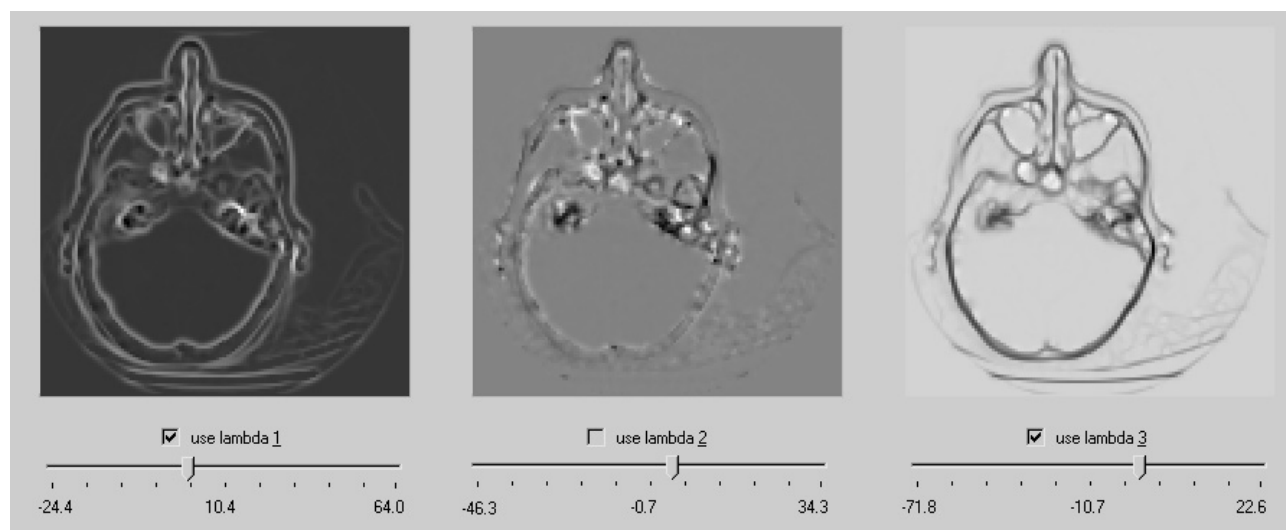

Figure 2: Part of an interface and demonstration of suitability for thresholding of the eigenvalues λ_1 and λ_3. The image corresponding to eigenvalue λ_2 lacks contrast information, and is therefore excluded from thresholding.

3 IMPLEMENTATION ISSUES

The presented approach benefits from the information provided by eigenvalue volumes. In case the eigenvalues already have been computed and stored to disk for later use for any of the applications mentioned in section 1.1, it provides the user with the possibility of an interactive preview of the sparse data set after the simple interaction discussed above.

The remainder of this section discusses implementation hints for the case when the eigenvalues are not yet computed.

3.1 Computation of the Hessian matrix

The computation of the Hessian matrix requires an approximation of the second order partial derivatives (Eq. 2). A common approach to this is to perform a convolution of the input data with the derivatives of LoG filter. Convolution, in general, is known to be a time consuming process. For the LoG convolution, however, the separability of the Gaussian kernel can be exploited reducing the time complexity, for one voxel, from $O(k^3)$ to $O(3k)$, where k is the kernel size. The other possibilities to speed up the convolution reside in using hardware specific features. For the SGI architecture Hopf and Ertl [Hopf99] speed up the 3D convolution up to 7 times. For the Intel/Windows platforms a good starting point could be the Image Processing Library [Intel00] optimized for Pentium processors.

3.2 Eigenvalues of the Hessian matrix

While solving a 2×2 matrix for eigenvalues leads to a quadratic equation, the explicit formula for the 3×3 case would be complicated due to cubic polynomials. In this case it is better to use a numeric solution. For symmetric real–valued matrices, the fast converging Jacobi's method is recommended [Press92].

4 RESULTS

Table 1 and figures 3–5 show that using definition (7) the volume can be, for visualization purposes, represented by approximately 10 % of its voxels. The thresholds T_1 and T_3 from this definition allow the user to interactively control the trade–off between quality of the display and quantity of the information stored in the decimated volume.

Equation (7) defines a variant of a 2nd order edge detector which is important for bioperception (see section 2.1). This is most noticeable by comparing the rendered images of the engine block data

sets (Fig. 3). In the decimated volume, areas corresponding to edges are emphasized and provide the observer with a better topological information of the data set.

All tested volumes have been quantized to 256 gray levels. To compare the visual appearance, the volumes have been displayed by a direct volume rendering algorithm (DVR) implemented in the VolumePro architecture [Pfist99].

5 CONCLUSION

We propose an easy–to–use framework for exploiting eigenvalues of the Hessian matrix to reduce volume data. We recall the relation of eigenvalues to the Laplacian filter, the suitability for thresholding eigenvalue images/volumes and define a combined threshold operation to generate a sparse data set. We evaluate our approach for several volume data sets of different modalities.

The advantage of sparse data sets resides in the possibility of their fast visualization due to back projection techniques mentioned in section 1.2. In this work we rather concentrate on the visual comparison between the DVR generated images of original volumes and volumes generated using our approach. A backprojection/splatting display with frame rate evaluation is left as a subject to the future work.

References

[Frang98] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496:130–137, 1998.

[Haar 94] B. M. ter Haar Romeny, L. M. J. Florack, A. H. Salden, and M. A. Viergever. Higher order differential structure of images. *Image and Vision Computing*, 12(6):317–325, 1994.

[Hladů00] Jiří Hladůvka and Eduard Gröller. (a web page related to this paper). http://www.cg.tuwien.ac.at/research/vis/vismed/evDecimation, October 2000.

[Hopf99] Matthias Hopf and Thomas Ertl. Accelerating 3D convolution using graphics hardware.

Original Volume			Decimated Volume				
Data set	Dimensions	kB	T_1	T_3	kB	%	Fig.
Lobster	$120 \times 120 \times 34$	478	28.872	-35.992	42	8.76	–
Engine Block	$256 \times 256 \times 110$	7 040	21.953	-23.381	658	9.36	3
CT Head	$128 \times 128 \times 113$	1 808	16.981	-19.182	183	10.12	4
MRI Head	$256 \times 256 \times 109$	6 976	7.822	-6.364	871	12.48	5

Table 1: Results for some typical volume data sets. Columns from left to right: name of the data set, its dimensions, number of voxels in kB, threshold values used for decimation, number of nonzero voxels in kB in the sparse data set, relative size to original volume, and a reference to the color plate.

In *Proceedings of the 1999 IEEE Conference on Visualization (VIS-99)*, pages 471–474, 1999.

[Intel00] Intel Corporation. IPL–Intel Image Processing Library, v2.5, 2000.

[Jähne97] Bernd Jähne. *Digital Image Processing*, chapter 10.10 Laplace–Based Edge Detection, pages 336–339. Springer–Verlag Berlin Heidelberg, 4 edition, 1997.

[Levoy88] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[Linde96] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 465–470, 1996.

[Marr82] David Marr. *Vision*. Freeman Publishers, 1982.

[Mölle97] T. Möller, R. Machiraju, K. Müller, and R. Yagel. Evaluation and Design of Filters Using a Taylor Series Expansion. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):184–199, 1997.

[Mroz00] Lukas Mroz, Helwig Hauser, and Eduard Gröller. Interactive high-quality maximum intensity projection. *Computer Graphics Forum*, 19(3):341–350, 2000.

[Pfist99] Hanspeter Pfister, Jan Hardenbergh, Jim Knittel, Hugh Lauer, and Larry Seiler. The VolumePro real-time ray-casting system. In *Proceedings of ACM SIGGRAPH*, pages 251–260, 1999.

[Pfist00] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *SIGGRAPH 2000, Computer Graphics Proceedings*, pages 335–342, 2000.

[Press92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*, chapter 11, pages 456–469. Cambridge University Press, 2 edition, 1992.

[Rusin00] Szymon Rusinkiewicz and Marc Levoy. QSplat: A multiresolution point rendering system for large meshes. In *SIGGRAPH 2000, Computer Graphics Proceedings*, pages 343–352, 2000.

[Saito94] Takafumi Saito. Real-time previewing for volume visualiztion. In *Symposium on Volume Visualization*, pages 79–106, 1994.

[Sato98] Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Hideki Atsumi, Shigeyuki Yoshida, Thomas Koller, Guido Gerig, and Ron Kikinis. 3D multi–scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2):143–168, 1998.

[Sato00] Yoshinobu Sato, Carl-Fredrik Westin, Abhir Bhalerao, Shin Nakajima, Nobuyuki Shiraga, Shinichi Tamura, and Ron Kikinis. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.

[Theuß00] Thomas Theußl, Helwig Hauser, and Eduard Gröller. Mastering windows: Improving reconstruction. In *Proceedings of IEEE Volume Visualization*, pages 101–108, 2000.
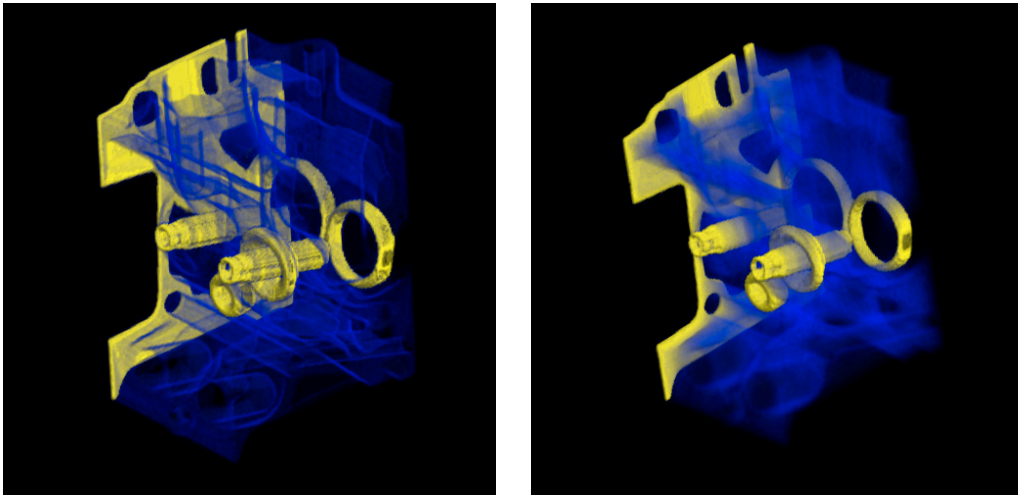
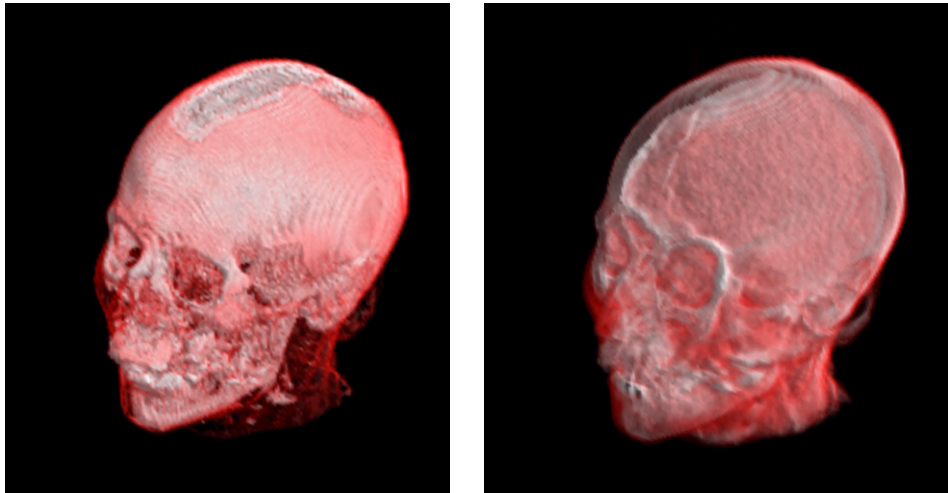Figure 3: Engine Block. DVR of decimated (left) and original (right) data sets

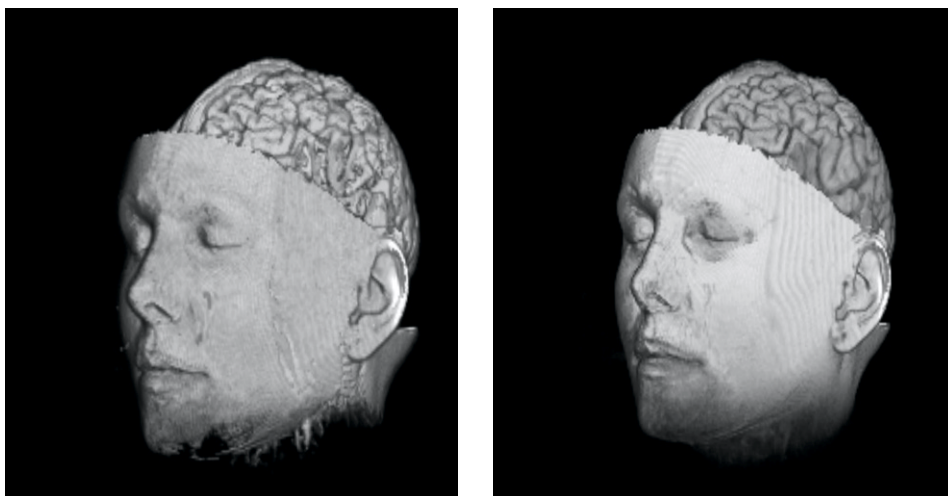Figure 4: CT Head. DVR of decimated (left) and original (right) data sets

Figure 5: MRI Head. DVR of decimated (left) and original (right) data sets