Salient Representation of Volume Data

Jiří Hladůvka, Andreas König, and Eduard Gröller

Institute of Computer Graphics and Algorithms
Vienna University of Technology

Abstract. We introduce a novel method for identification of objects of interest in volume data. Our approach conveys the information contained in two essentially different concepts, the object's boundaries and the narrow solid structures, in an easy and uniform way. The second order derivative operators in directions reaching minimal response are employed for this task. To show the superior performance of our method, we provide a comparison with its main competitor – surface extraction from areas of maximal gradient magnitude. We show that our approach provides the possibility to represent volume data by a subset of a nominal size.

1 Introduction

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 [8]. A different type of discontinuity – a line is also a structure of particular interest. While in a 2D image the representatives of narrow solid structures are spots and lines, in volume data this is more general. Identification of blobs, cylinders, and sheet–like structures plays a crucial role in medical visualization [16].

To represent volume data by just a small subset of important voxels is desirable for and addressed by a number of applications. Interactive volume visualization over the internet based on a client/server architecture profits from elaborated strategies for progressive data transmission. Here it is desirable that the content of a volume is visually interpretable already in the early stages of transmission to and visualization by a client. To achieve this, the server may start transmitting salient features earlier than the rest of the data. Non-distributed visualization may benefit from storing a small, representative subset of the data to disk. Such a representation can be reused later for a quick preview display using, e.g., non-photo realistic techniques [14] or algorithms yielding more realism [9, 11, 13].

For these applications, the subset of high salience has to be identified. The best established algorithms involve isosurface extraction [17], boundary identification or emphasis [7], or narrow solid structures identification or emphasis [1, 15, 16]. While the usual paradigm to identify object boundaries is to evaluate the

D. Ebert, J. M. Favre, and R. Peikert (Eds.): Data Visualization 2001, pp. 203–211,351

magnitude of the gradient, the identification of narrow solid structures requires the use of either special filters or 2nd order derivative filters.

In this work we propose a filtering technique for the identification of both boundaries and narrow structures. Our algorithm is based on identification of areas with large negative second derivatives, and handles both of the cases in a uniform way. Defining a salience function based on this quantity allows us to identify those voxels of the input volume which provide a significant content necessary for visualization.

In the following section we discuss the theoretical background for our method. In section 3 its complexity and implementation issues are discussed. Results and a comparison to the gradient method are presented in section 4.

2 The proposed method

2.1 Edge detectors and line detectors revisited

For edge detection two kinds of filters have been designed – those based on looking for maxima of the first derivative and those based on looking for zero–crossings of the second derivative.

While these concepts are intuitive for 1D signals, the situation in higher dimensions gets more complicated. We assume that the volume is given as a density function I. To use the extrema of 1st derivatives we need to know the directions in which they occur. From calculus it is known that for the first derivative this direction is the gradient vector ∇I and the magnitude of the derivative in this direction is the magnitude of the gradient: $I'_{max} = I'_{\nabla} = ||\nabla I|| = (\sum_{i=1}^{3}(\partial I/\partial x_i)^2)^{1/2}$. Looking for maxima of gradient magnitudes yields an isotropic edge detector which responds both to outer and inner side of the object equally (Fig. 1a). For the second derivative approach it is necessary to check the neighborhood of a voxel for zero–crossings, i.e., for areas where the 2nd derivative changes its sign. The 2nd derivative is usually estimated by the rotationally invariant Laplacian $\triangle I = \sum_{i=1}^{3} \partial^2 I/\partial x_i^2$. A less referred feature of the Laplacian operator is that it responds with negative values at the inner part and by positive values at the outer part of the object's edge[1] (Fig. 1b). We put this into contrast to the gradient–based edge detection, which yields an equal response on both sides of an edge and exploit this fact to represent the objects' edges only by their internal side. Such a representation requires, compared to the gradient method, a smaller amount of voxels, or in other words it provides a better distribution over the surface in early stages of the progressive transmission for a limited bandwidth.

Considering the density profile, it is evident that the concepts of 1st derivative maxima can not be directly applied for spot and line detection (or, more generally speaking, for detection of narrow areas which in 3D correspond to blobs, lines, and sheet–like structures). The response of a 1st order derivative filter to a line, for instance, results in two lines, which would require a special, nontrivial

[1] Assuming the objects are of a higher density than background and not vice versa

mechanism for detection of the area in-between (Fig. 1a). A 2nd order derivative filter, on the other hand, responds to a line by negative values at its interior (Fig. 1b).

As a result we get a twofold interpretation of areas where the 2nd order derivative operator responds with negative values. Firstly such areas correspond to internal parts of a boundary, secondly they identify narrow structures. To make the search for negative areas more feasible for separation by thresholding, we are interested in the directions where the 2nd order derivatives reach the minima.

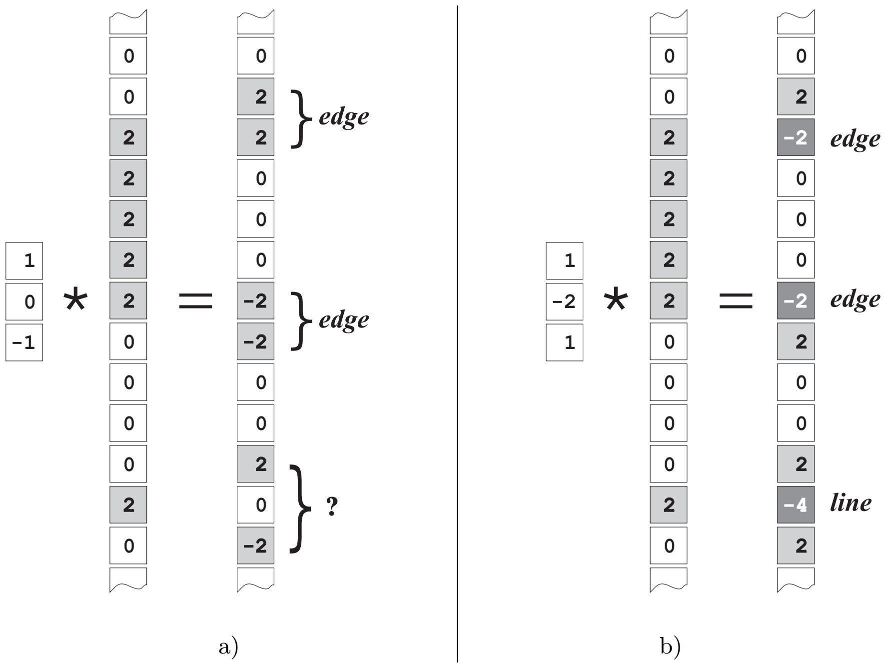

Fig. 1. Examples for 1D density profile. Responses of a 1st order derivative filter (a) and a 2nd order derivative filter (b) to an edge and to a line.

2.2 The smallest 2nd order directional derivative

Stating the Taylor expansion of a 3D density function I for the first three terms in the vicinity, spanned by vectors Δx, of a 3D point x_0

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

it is evident, that the 2nd order information is entirely expressed by the symmetric Hessian matrix:

$$H = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \qquad \text{where the terms} \quad I_{ab} = \frac{\partial^2 I}{\partial a \partial b} \tag{2}$$

denote the 2nd order mixed partial derivatives. For a fixed x_0, the term $\Delta x^T H \Delta x$ gives the second derivative of the density in the direction of Δx. The Hessian matrix, as a real–valued and symmetric matrix, has real–valued eigenvalues λ_i. From the definition of the eigenvalues, $Hv_i = \lambda_i v_i$, follows that the eigenvalues λ_i give the second derivatives in the direction of the eigenvectors v_i: $\lambda_i = v_i^T H v_i$. Since H in this context represents a quadratic form, computing the smallest eigenvalue directly yields the minimal directional derivative at the point x_0. In the following, we will denote this eigenvalue λ_3, i.e., $I''_{min} = I''_{v_3} = \lambda_3$.

2.3 Salience by smallest eigenvalue

So far we described a mechanism to finding the smallest 2nd order directional derivative at a given grid point of a volume. As λ_3 is just a special case of a 2nd order directional derivative, the interpretation from section 2.1 in the direction of the corresponding eigenvector remains the same:

1. Areas featuring very low λ_3 represent an inner part of an object's boundary. Unlike in the case of gradient magnitude where looking for the maxima yields a representation of the boundary from both the outer and the inner side, our approach restricts the representation of boundaries just to the inner side. Compared to the gradient magnitude operator, the boundary can therefore be represented by a smaller amount of voxels.
2. Areas with very low λ_3 correspond to a structure proportional to a scale given by a derivative operator. Tracking of structures in volumes (like blobs, lines, and sheets) by a first derivative operator would be hard if not impossible.
3. Having the minimal second derivative is more suitable for separation by thresholding than having a second derivative in an arbitrary direction.

Due to these reasons, areas featuring low negative eigenvalues λ_3 yield a better representation of a volume then those with high positive values of the gradient magnitude. In order to provide a comparative study between these two approaches, we define the two following salience functions S_Γ, S_Λ of a voxel v and the two corresponding p%-subsets of an input volume V they determine:

$$\begin{aligned} S_\Gamma[v] &= ||\nabla||\,[v] & \qquad \Gamma[p] &= \{p\,\%\text{ of } V \text{ with the highest } S_\Gamma\} \\ S_\Lambda[v] &= \;-\lambda_3[v] & \qquad \Lambda[p] &= \{p\,\%\text{ of } V \text{ with the highest } S_\Lambda\} \end{aligned} \tag{3}$$

For a given percentage p, functions S_Γ, S_Λ determine the $p\,\%$ of 'top salient' voxels which will represent the volume. For a progressive transmission of data through a network, these functions determine the priority of transmission: the voxels with higher salience will be transmitted earlier.

Obviously, there are also other candidates which might succeed well in the task of volume representation by a fraction of the data. In the following we give an overview of possible competitors and argue why we do not compare them to our method:

Isosurface methods require a user input to specify the density which determines an isosurface. The result is dependent on and yields only structures defined by this choice. Our method processes data automatically and delivers surfaces of more than just one isolevel.

The "distance to closest boundary" method, as introduced by Kindlmann and Durkin [6] yields an opacity transfer function for boundary emphasis. The algorithm performs a statistical analysis of the zero, 1st, and 2nd order derivatives in the direction of the gradient, providing information on which densities contribute most to boundaries. Defining a salience function based on density is essentially inconsistent with our approach, which is position based, and a comparison would be hardly possible.

Density distribution analysis based on *all* eigenvalues of the Hessian as proposed by Frangi et al. [1] or Sato et al. [15, 16] restricts the search space just to structures of a particular shape and a certain scale, and excludes boundaries of objects. In contrast, our approach handles both boundaries and structures in a uniform way.

In our previous work [2] we suggested taking into account also areas featuring high magnitudes of the largest eigenvalue λ_1. The maxima of this eigenvalue correspond to outer parts of objects' boundaries and do not contribute to the output significantly.

3 Implementation and complexity

3.1 Hessian matrix versus gradient vector

Computation of both the gradient vector and the Hessian matrix at grid points involves an approximation of the first and the second partial derivatives, respectively. For this task, convolution of the data with kernels designed for a particular derivative in a specific direction is usually employed.

For the first derivatives, kernels of size up to three are usually found in the textbooks: Roberts, Prewitt and Sobel filters are feasible for fast computation.

The Hessian matrix requires an estimation of 2nd order derivatives which is, especially for small kernels, much more sensitive to noise. The usual practice is to pre-smooth the input data with a Gaussian filter. Due to the associativity of convolution, the smoothing step and the derivation can be combined, resulting in a convolution of the data with a derivative of the Gaussian filter of a bigger size. The second reason to use the derivatives of the Gaussian filter is that we want to detect features represented at a certain *scale*. The Gaussian filter is the *only* filter which meets both the *minimum-maximum principle* and *scale invariance* necessary for such a representation. For more details on scale spaces we refer the reader for instance to Jähne [5].

To remain consistent for comparison of both the quality of results and the computational costs we used filters of the same size both for 1st and 2nd derivatives. Using the Gaussian filter requires that its size k is proportional to the standard deviation, so the kernels usually involved are 5, 7, or 9 voxels wide. Convolution with moderately–sized kernels is usually a computationally expensive process. To speed it up, hardware features can be used for specific platforms [3, 4]. For a software implementation, the separability of the Gaussian derivative kernels can be exploited:

$$\left(\overbrace{\frac{\partial^o}{\partial x^a \partial y^b \partial z^c} G_\sigma(x,y,z)}^{k\times k\times k}\right) * I = \overbrace{\frac{d^a}{dx^a} G_\sigma(x)}^{k\times 1\times 1} * \left(\overbrace{\frac{d^b}{dy^b} G_\sigma(y)}^{1\times k\times 1} * \left(\overbrace{\frac{d^c}{dz^c} G_\sigma(z)}^{1\times 1\times k} * I\right)\right) \tag{4}$$

where nonnegative integers $a + b + c = o \in \{1, 2\}$ determine the order of derivation, and σ is the standard deviation of the Gaussian filter $G_\sigma(x) = exp(-\frac{x^2}{2\sigma^2})/\sqrt{2\pi}\sigma$. The decomposition according equation (4) reduces the overhead, for a partial derivative at a grid point, from convolution with a 3D kernel (complexity $O(k^3)$) to three convolutions with a 1D kernel (complexity $O(3k)$).

A direct application of equation (4) would require 18 1D convolutions for Hessian elements as compared to 9 1D convolutions for the gradient vector. Further speed-up can be achieved by appropriate reorganization and caching. Three 1D convolutions can be saved for the computation of the Hessian matrix, (e.g., $G_\sigma(x)*I$ can be reused three times and $G'_\sigma(x)*I$ twice) and one convolution can be saved for the gradient (e.g., $G_\sigma(x) * I$ can be reused twice). This reduces the number of required 1D convolutions to 15 for the Hessian and 8 for the gradient.

3.2 Eigenvalues of the Hessian versus magnitude of the gradient

While computing the gradient magnitude by the Euclidean norm requires three multiplications, two additions and one square root, the computation of eigenvalues of the Hessian matrix is more complex. The explicit formula would require solving cubic polynomials. In our implementation we used a numerical solution – the fast converging Jacobi's method as recommended by Press et al. [12] for real-valued, symmetric matrices.

Table 1 summarizes the overall costs concluding that the computation of eigenvalues is, as compared to the computation of the gradient magnitude, in average 2.7 times more expensive.

3.3 Construction of representative subsets

To build the subsets $\Gamma(p)$ and $\Lambda(p)$, we firstly construct cumulative histograms of quantities $||\nabla||$ and $-\lambda_3$, respectively, in one pass through the volume in linear time. The required percentage p controls the number of voxels to be included into the respective subset. The search for adjacent histogram bins straddling this number is logarithmic. The indices of bins correspond to a threshold which is used as a decision function.

Input Volume		$T_{\|\|\nabla\|\|}$		$T_{(\lambda_1,\lambda_2,\lambda_3)}$		factor
Data set	Dimensions	8 $*$	+norm	15 $*$	+Jacobi	
Lobster	120 × 120 × 34	3.37	3.43	6.30	8.69	2.53
Vertebra 1	128 × 128 × 74	8.52	8.69	15.91	24.58	2.83
CT Head	128 × 128 × 113	12.92	13.17	24.20	36.87	2.80
MRI Head	256 × 256 × 109	50.64	51.60	93.36	141.98	2.75
Engine Block	256 × 256 × 110	50.36	51.29	94.05	139.49	2.72
Tooth	256 × 256 × 161	73.49	74.71	137.49	184.35	2.47
Vertebra 2	256 × 256 × 241	109.89	111.96	206.58	275.45	2.46

Table 1. Time in seconds for computing the magnitude of the gradient and the eigenvalues of the Hessian as measured on a Pentium III, 800 MHz. 1D cyclic convolutions (eq. 4) with kernel of size $k = 7$ have been used. The meaning of columns from left to right: name of the data set and its dimensions; time for the computation of all partial derivatives for the gradient vector, and after Euclidean norm; time for the computation of all partial derivatives for the Hessian, and after eigenvalues search; the ratio of overall times for eigenvalues and gradient magnitude.

4 Results

To compare the quality of a volume representation by subsets Γ and Λ from equation (3), we generated sparse volumes where the density of voxels not presented in either of the subsets have been set to zero. Such volumes have been rendered by direct volume rendering provided by the VolumePro architecture [10]. In the following we refer to Figure 2 and to the project's web site[2].

Lobster: We compared 2, 4, 6, 8, and 10% representations of this data set. While the legs of the lobster are, due to the line filter, visible and good recognizable already in $\Lambda(2\%)$, in $\Gamma(6\%)$ they just start to appear. Representation by $\Gamma(2\%)$ is insufficient. The differences between Λ and Γ vanish with increasing percentage. Nevertheless, they are still noticeable between $\Lambda(10\%)$ and $\Gamma(10\%)$.

Vertebra 1: Neither $\Gamma(2\%)$ nor $\Lambda(2\%)$ provide a good representation, though there is much more content visible in $\Lambda(2\%)$. $\Gamma(4\%)$ features broken contours and is approximately on a level of $\Lambda(2\%)$. $\Lambda(4\%)$ and $\Gamma(6\%)$ represent approximately the same level, but $\Lambda(4\%)$ provides more details and more closed contours. $\Lambda(6\%)$ is already close to a good representation of the original data set.

Vertebra 2: Subset $\Gamma(2\%)$ features only high density screws. While the ribs only begin to appear in $\Gamma(4\%)$, their are better visible already in $\Lambda(2\%)$ due to a more even distribution of boundary voxels. $\Gamma(6\%)$ yields even less information than $\Lambda(4\%)$. $\Lambda(6\%)$ is a good approximation of the original data set.

[2] accessible via http://www.cg.tuwien.ac.at/research/vis/vismed/

Tooth: There are two significant features identified in this data set: a tooth inside a surrounding cylinder. The ratio between the number of voxels belonging to either of the features is quite big, and the subsets Γ and Λ contribute to a large extent to the wall of the cylinder. For this reasons we have noticed an obvious difference in the appearance of the tooth just by a very small percentages. $\Lambda(0.65\%)$ yields more content than the corresponding $\Gamma(0.65\%)$. Rendering this data set we also have noticed a suppression of the partial volume effect in Λ data sets. We explain this suppression as a consequence of representing object boundaries by their inner parts.

5 Conclusion

We introduced a novel approach for the representation of volume data sets by a subset which contains the salient features. Our attempt is to convey information contained in two essentially different modalities, the object's boundaries and the narrow structures in an easy and uniform way. For this task we employ second order derivative operators in the directions reaching minimal response.

Compared to the methods based on gradient maxima, our method represents objects only by the internal side of their boundaries, reducing thus the amount of necessary voxels.

Looking for the minima of second order derivative yields also a structure detector proportional to the scale of the derivation operator. In contrast to the concepts of Frangi [1] and Sato [15, 16] we enforce no shape restrictions, making no distinction among blob, tubular, and sheet–like structures.

The drawback of our method is a higher computational cost. Computation of the Hessian's eigenvalues is approximately 2.7 times more expensive than the computation of the gradient magnitude.

We evaluated our method and compared it to the gradient method for several data sets. Due to the results we conclude that our method performs better than methods based on gradient magnitude. For the same level of quality of visualization it allows to represent a data set by a reasonably smaller subset. The possible applications of such an advantageous representation are, e.g., progressive transmission over the internet and the generation of preview data sets.

Acknowledgements

The work presented in this publication has been funded by the $\mathrm{V^{is}M^{ed}}$ project, `http://www.vismed.at`. $\mathrm{V^{is}M^{ed}}$ is supported by *Tiani Medgraph*, Vienna, `http://www.tiani.com`, and the *Forschungsförderungsfonds für die gewerbliche Wirtschaft*, Austria, `http://www.fff.co.at`. The Vertebra1 and Vertebra2 data sets are courtesy of *Tiani Medgraph*, Vienna.

References

1. 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.
2. J. Hladůvka, A. König, and E. Gröller. Exploiting eigenvalues of the Hessian matrix for volume decimation. In *Proceedings of 9th International Conference in Central Europe on Computer Graphics, Visualization, and Computer Vision (WSCG 2001)*, pages 124–129, 2001.
3. M. Hopf and T. Ertl. Accelerating 3D convolution using graphics hardware. In *Proceedings of the 1999 IEEE Conference on Visualization (VIS-99)*, pages 471–474, 1999.
4. Intel Corporation. IPL–Intel Image Processing Library, v2.5, 2000.
5. B. Jähne. *Digital Image Processing*, chapter 5: Multiscale Representation, pages 121–138. Springer–Verlag Berlin Heidelberg, 4th edition, 1997.
6. G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of IEEE Volume Visualization*, pages 79–86, 1998.
7. M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.
8. T. Lindeberg. Edge detection and ridge detection with automatic scale selection. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 465–470, 1996.
9. L. Mroz, H. Hauser, and E. Gröller. Interactive high-quality maximum intensity projection. *Computer Graphics Forum*, 19(3):341–350, 2000.
10. H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The VolumePro real-time ray-casting system. In *Proceedings of ACM SIGGRAPH*, pages 251–260, 1999.
11. H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *SIGGRAPH 2000, Computer Graphics Proceedings*, pages 335–342, 2000.
12. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*, chapter 11: Eigensystems, pages 456–469. Cambridge University Press, 2 edition, 1992.
13. S. Rusinkiewicz and M. Levoy. QSplat: A multiresolution point rendering system for large meshes. In *SIGGRAPH 2000, Computer Graphics Proceedings*, pages 343–352, 2000.
14. T. Saito. Real-time previewing for volume visualization. In *Symposium on Volume Visualization*, pages 79–106, 1994.
15. Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis. 3D multi–scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2):143–168, 1998.
16. Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.
17. P. M. Sutton, C. D. Hansen, and H.-W. S. D. Schikore. A case study of isosurface extraction algorithm performance. In *Data Visualization 2000*, 2000.

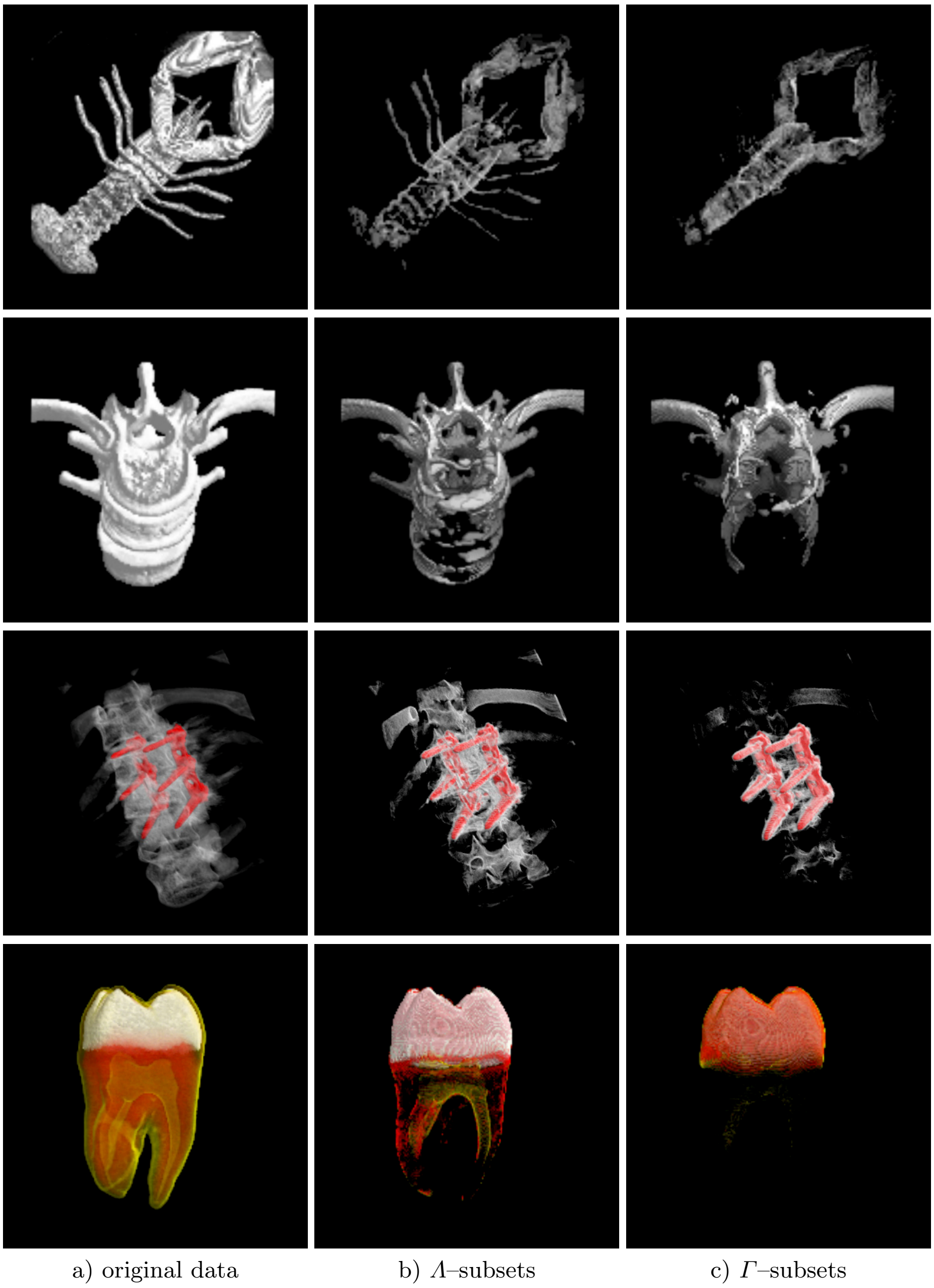

Fig. 2. Direct volume rendering of Lobster, Vertebra1, Vertebra2, and Tooth data sets (a) and their representations due to salience provided by our method (b) and by detection due to gradient magnitude (c). The Lobster subsets consist of 2.01% voxels of the original data set, Vertebra1 and Vertebra2 of 4.03% voxels, and the Tooth subsets of 0.67% voxels.