An Adaptive Cutaway with Volume Context Preservation

S. Grau[1] and A. Puig[2]

[1] Polytechnic University of Catalonia, Spain
[2] University of Barcelona, Spain

Abstract. Knowledge expressiveness of scientific data is one of the most important visualization goals. However, current volume visualization systems require a lot of expertise from the final user. In this paper, we present a GPU-based ray casting interactive framework that computes two initial complementary camera locations and allows to select the focus interactively, on interesting structures keeping the volume's context information with an adaptive cutaway technique. The adaptive cutaway surrounds the focused structure while preserving a depth immersive impression in the data set. Finally, we present a new brush widget to edit interactively the opening of the cutaway and to graduate the context in the final image.

1 Introduction

Knowledge expressiveness of scientific data is one of the most important visualization goals. The abstraction process that the final user should carry out in order to convey the desired information in the underlying data is normally difficult and tedious. Several methods have been published to gather visual information contained in the data. However, volume renderings often include a barrage of complex 3D structures that can overwhelm the user. Over the centuries, the traditional illustration techniques for visual abstraction enhances the most important structures into a context with different painting techniques (Figure 1A). Several approaches provide interactive focus selections and volume visualizations, such as cutaways [1] and close-ups [2]. Some methods are based on NPR techniques and ghosting shading to simulate the illustrator tools. All of these methods are included in the field of illustrative visualization [3], where the main goal is to develop applications that can integrate illustrations in the expert's ordinary data analysis in order to get more semantics from the data.

Some metaphors of interaction have been provided to help users in data navigation between focus and context (importance-driven, VolumeShop, exoVis, LiveSync++, ClearView). Specifically, ClearView proposes a simple point-and-click interface that enables the user to show particular areas of the focused object while keeping the surface context information (Figure 1B). In some applications, the context region's volume information is especially important, such as we can see in Figure 1C, where the different strata around the eye, which is the focus region, must be visualized. Strata representing layers of the context may help

G. Bebis et al. (Eds.): ISVC 2009, Part II, LNCS 5876, pp. 847–856, 2009.

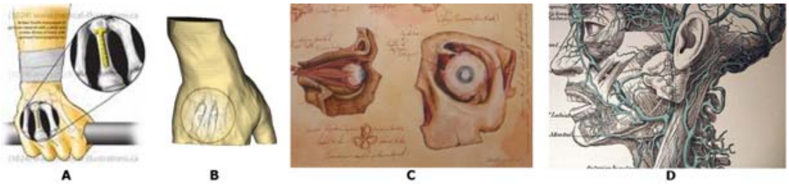

Fig. 1. Illustration examples of the underlying ideas of the adaptive cut-away visualization: (A) An illustration example, (B) visualization based on ClearView method ([4]), (C) dual-camera illustration (image from http://www.keithtuckerart.com/Illustration.html) and (D) Context preservation -ear- although it occludes the focused region -vessels- (image from the medical dictionary Allen's Anatomy)

to locate the focused structure [5]. Moreover, in some cases, the context which occludes focused structures should be preserved (see Figure 1D). For this reason, an interactive tool to preserve the context is useful to obtain the desired final image. In this paper, we propose an enhancement of the ClearView paradigm where the context's volume data is adaptively clipped around the focus region. To show the layers of the context around the focus, we use volume rendering. We use ghosting techniques to preserve the context without occlusion of the focus. Moreover, we propose a new brushing widget to edit interactively the gosthing effect and to graduate the contribution of the context in the final image.

On the other hand, a complementary parameter is the initial location of the camera, from the user can start a free exploration of the data set. To select a starting good viewpoint is sometimes a tedious task for a non-experimented users. Some traditional illustrations use two different views of the region of interest to enhance the perception and construction of the mental image (see Figure 1C). Actually, in the clinical routine, prefixed views are used, based on saggital, axial and coronary views, despite that they are not always the best. In the bibliography, we can find techniques that automatically locate the viewpoint according to the importance of the structure to be rendered. Most of them are computed in a preprocessed stage. We propose an efficient GPU-based computation of two initial correlated views that shows maximal information of the focused structure. In case of Figure 1C, the focused structure is the eye and it is showed in two dual viewpoints, one in a oblique view and one in front of the eye. Also, the user can adjust interactive any of the suggested cameras, and our system efficiently computes the new dual view.

2 Related Work

Comprehension of the meaningful structures in an image is a low-level cognitive process. Single visual events are processed in an intuitive way. However, multiple visual events involve a more complex cognitive process. Many works have been addressed this problem. In the following, first, we will review the use of cut-away views and focus and context techniques that render the contextual information using NPR-shading and ghosting techniques. Secondly, we review the previous approaches in optimal camera location estimation.

Cutaways: Usually, traditional artists render anatomy layered structures using clipping planes or curved clipping surfaces. Following this idea, several approaches have been proposed for polygonal rendering [6] and for volume rendering [2]. Interactive cutaways allow users to specify the cuts to be performed: by planes oriented along principal axes, user-defined interactive sculpting tools [7] and user-manipulated deformable meshes [8]. Users can explore regions with peel-away [9] and exploded views [10]. 3D automatic cuttings design the appropriate cut of a focused internal structure based on the feature specification. Viola et al. [1] use the object importance to avoid unwanted occlusions. Zhou et al. [11] use the distance to the features to emphasize different regions. Bruckner et al. [2] present cuts completely based on the shape of the important regions. Krüger et al. [4] presents in ClearView a region-focal based interaction that preserves the focused structure. Many of these approaches keep contextual information to give a better impression of the spatial location of the focused structure. Usually, the surface context information can be kept in different ways: with high transparency, low resolution, different shading styles [1]. These algorithms are designed primarily to expose perfectly layered structures in the context but they can not show the intertwined volume structures of the context often found in 3D models. In order to preserve the contextual volume information, we present an automatic feature-based cutaway approach. We propose a view-dependent cutaway opening that guarantees the visibility of the selected inner structure for any viewpoint location.

Optimal viewpoint selection: Setting the camera so that it focuses on the relevant structures of the model in volume direct rendering has been addressed by many authors [12,13,14,15,16,17], extending the ideas used in surface-based scenes. Visual information is the measure used to estimate the visibility between a viewpoint and the structure of interest of a volume data set. Then, the analysis of the best camera position is focused under heuristic functions [17,15,13] and information theory approaches (such as viewpoint entropy [12] and the mutual information entropy [14,16]). These metrics are not universal and, as [18] conclude, not one descriptor does a perfect job. However, as heuristic approaches define a weighted energy function of view descriptors, they could be easily accelerated using blending operations between view descriptors. Some approaches study correlations between cameras in varying time measuring the stability between the views based on the Jense-Shanon divergence metric [12]. Also, in path-views searching a Normalized Compression Distance is used [16]. Our goal is to obtain the most representative dual projection according to the position of the best view. Then, we restrict the search space of the dual-camera position in a set of positions related to the best one. Thus, we guarantee the distance between the correlated cameras and, moreover, we reduce our searching space.

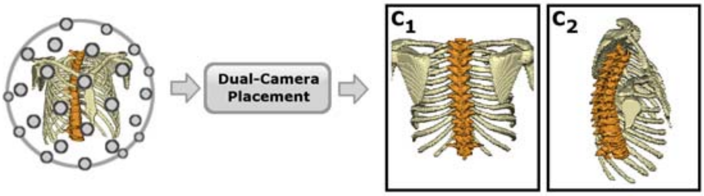

Fig. 2. The dual-camera placement method computes two camera locations from the sampled bounding sphere that show complementary information of the focused regions (in orange). The method takes into account the adaptive cut-away strategy that shows always the focused regions.

3 Overview

The main goal of our system is to provide an interactive exploration of volume data sets for inexperienced users. First of all, the focused structures should be selected with the aid of a transfer function classification. With presegmented labeled data sets, the user can directly use a value-based function widget. Once the focused regions are selected, the user can explore the data set using an automatic adaptive cutaway strategy that shows always the focused regions but also keeps the relevant contextual volume information. Moreover, if the user is not entirely satisfied with the cutaways results, a simple point-and-click editing brush tool aids him to adjust the final image, adding and removing contextual volume appearance.

Before this process, camera position should be fixed. Two different camera locations, (C_1, C_2), are computed as a preprocess of the adaptive cutaway (see Figure 2). C_1 represents the best view to visualize the focused regions according to user-defined criteria. C_2 represents the correlated dual-camera that preserves the maximum information integrated in both views. This step is called dual-camera placement and it suggests two views that can be adjusted manually by the user to obtain the final image. The system can efficiently compute the correlated dual-camera of any user-defined best view.

4 Adaptive Cutaway with Context-Volume Preservation

The automatic feature-based cutaway opening is generated in two stages. In the first stage, we setup the parameters to optimize the raycasting. In the second stage, we compute context and focus image layers using the raycast algorithm and we blend both image layers using an auxiliary distance map.

Raycasting Parameters Setup

The first step finds out for each sampled ray $p_{i,j}$ two hit points against the volume data set in FTB order: the first intersection with the context, $I_{context}(p_{i,j})$, and the first intersection with the focus, $I_{focus}(p_{i,j})$ (see Figure 3). Rays with no intersection are skipped for the next steps.

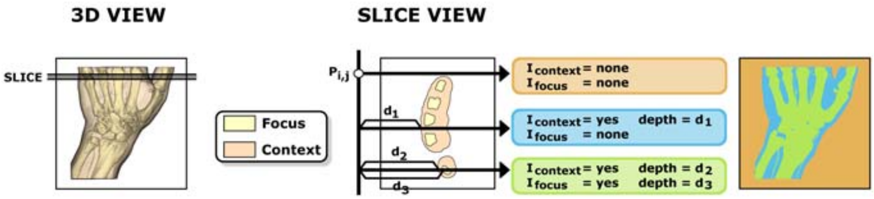

Fig. 3. Ray focus and context intersection's computation. The slice view shows the three possible ray intersection's cases: no focus and context intersection (orange), only context intersection (blue) and focus and context intersections (green).

When we want to show always the focus region, the sampled rays may begin at the starting point $I_{focus}(p_{i,j})$ if exists and at $I_{context}(p_{i,j})$, otherwise. However, the depth perception between focus and context is not clear (see Figure 4A).

A more deeper immersive impression could be obtained when different strata of the context are shown. Thus, our proposal consists of gradually opening the clipped area between the selected feature and the context (see Figure 4B) in object space as well as in image space. We adaptively open the cutaway in function of the depth of the sample and the Euclidean distance of the pixel $p_{i,j}$ ($DM(p_{i,j})$) to the nearest pixel that belongs the focused region projection (Figure 3 right shows the focused projection in green). When the pixel belongs to the projection, $DM(p_{i,j})$ is zero.

The first cast of the ray for the pixel $p_{i,j}$ is done between the $I_{context}(p_{i,j})$ and the $I_{focus}(p_{i,j})$. In those pixels that do not have $I_{focus}(p_{i,j})$, it is estimated using a pyramidal method for interpolation of scattered pixel data [19]. We extrapolate I_{focus} by averaging only using the known ones in an analysis process and by filling unknown ones in a synthesis process. At the end of this process, we know all the $I_{focus}(p_{i,j})$ and $I_{context}(p_{i,j})$ and they are stored in a 2D texture.

We apply a progressive cutaway for those rays whose distance map values are less or equal than a user-defined opening width (w_{open}). We can identify these pixels $p_{i,j}$ as those that fulfill that $\|DM(p_{i,j})\| \leq w_{open}$. For these rays, the starting point of the ray is interpolated between the $I_{focus}(p_{i,j})$ and the $I_{context}(p_{i,j})$ weighted linearly by $\|DM(p_{i,j})\|$.

Raycasting

A common volume raycasting with early-termination is used to create the image of Figure 4 A or B. The cast of the ray begins in the sample computed between $I_{focus}(p_{i,j})$ and $I_{context}(p_{i,j})$. This image is stored in a texture called *focus image layer* (L_{focus}).

Moreover, to give a better impression of the spatial location of the focused structure in the context area, we employ context preserving strategies to keep context edges on the focus projection. It is done computing the curvature, illumination and value at $I_{context}(p_{i,j})$ sample point for all the pixels inside the focus projection. These values are stored in the *context image layer* ($L_{context}$).

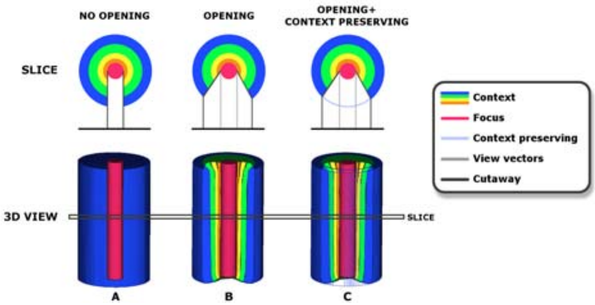

Fig. 4. Cutaway example using a phantom data set. It shows the adaptive cutaway (A), the opening cutaway (B), and the context preservation (C).

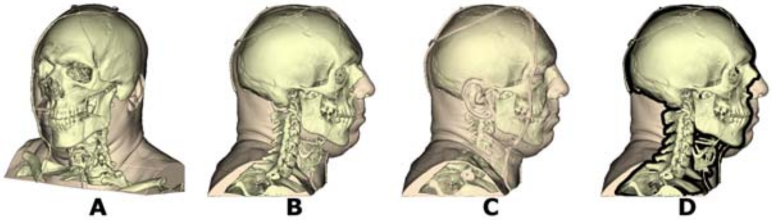

Fig. 5. Cutaway openings in two viewpoints positions without (A and B) and with (C) context preservation or using the border effect (D)

A simple blending process combines the context and focus image layers properly obtaining the final image. Figure 5C shows the enhancement of the context preserving idea.

An additional border effect can be applied in function of the computed distance $DM(p_{i,j})$ (see Figure 5D). The pixels on a surrounding area of the border projection's boundary can be easily detected by their values and rendered as a silhouette.

In summary, with a user-defined opening and border widths (w_{open} and w_{border}), during the composition stage, the final color $Col_{i,j}$ at each pixel $p_{i,j}$ is computed as:

$$Col_{i,j} = \begin{cases} blend(L_{focus}(p_{i,j}), L_{context}(p_{i,j})) & \text{if } 0 \leq \|DM(p_{i,j})\| \leq w_{open} \\ border\_color & \text{if } w_{open} < \|DM(p_{i,j})\| \leq w_{open} + w_{border} \\ L_{focus}(p_{i,j}) & \text{otherwise} \end{cases}$$

where, Col is the final pixel color and *blend* is a blending function.

Interactive Brush Widget

In order to allow users modify interactively the starting cast of the ray between $I_{focus}(p_{i,j})$ and $I_{context}(p_{i,j})$, we define a 2.5D brushing tool. Upon the cutaway is visualized, the user can paint over the final image using a circular depth-brush that changes the ray's starting point for all the pixels inside the circle. The depth of the starting point is increased/decreased according to the depth of the current brushing tool. Thus, only the affected pixels of the focus layer, L_{focus}, are re-sampled.

5 Dual-Camera Placement

The camera's searching space's continuity and smoothness is essential to guarantee the stability of the view quality. Our solution space S is the bounding sphere of the complete model that contains the set of all possible camera locations. The space of N camera locations is defined as a discrete and finite set of samples over S that are iso-distributed along the surface of the sphere (we use the HealPix package [20]). We assume that the up vector can be arbitrarily chosen at each location due to the camera rolls not having an effect on the visibility.

We propose a heuristic method with the following image-based descriptors: visibility, coverage and goodness of the location of the focus on the viewport. The visibility descriptor (vd_1) evaluates how far is the focused regions to the camera location using the average depth, and vd_1 is bigger for higher average depth. The quality of the view can be enhanced with the coverage of the focused regions (vd_2) that arises with the size of the focus in the final projection. Finally, the goodness of the location (vd_3) measures the centroid of the projected area. The focus should be centered in the viewpoint area. Only the focused regions are taken into account in the camera quality estimation. For this reason, the method computes first, for each camera location, an image that stores for each pixel the first intersection with the focused regions and the corresponding depth, when this intersection exists. This image is next used to compute the image-based descriptors. All these view descriptors are computed on the GPU. The number of pixels, the bounding box and the average depths of all pixels are computed following a GPU-based hierarchical strategy [19]. The final view quality estimation of a camera location c is computed by the user-defined weights of the different viewing descriptors, vd_i.

To find the best camera placement (C_1), we search the maximum view quality estimation using a GPU-based hierarchical method. Its efficiency depends directly on the size of the texture that stores the N viewpoint values. For small sizes, we search the maximum value on the CPU directly.

Illustration techniques that enhance the perception of the model with dual-camera inspire us to define two correlated views (see Figure 1C). Classical illustrations shown quasi-orthogonal projections of a region. In order to find out the second camera, C_2, we simply reduce the searching samples of the solution space to orthogonal regions to the best camera location, C_1. We have tested orthogonal regions as streams and semi-spheres (see Figure 6).

6 Simulations

We have evaluated the performance of the proposed methods using a Pentium Dual Core 3.2 GHz with 3GB memory equipped with an NVidia GeForce 8800

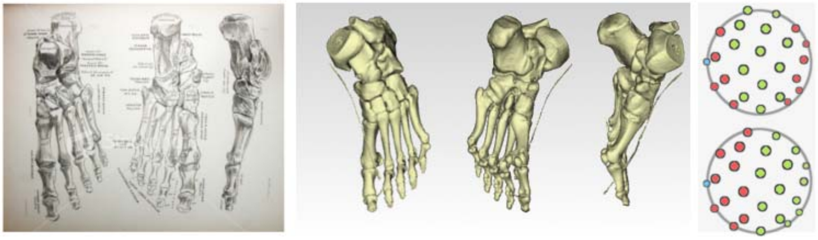

Fig. 6. Dual-camera illustrations: (left) the reference illustration (image from the medical dictionary Allen's Anatomy); (center) our results: best camera, semi-sphere dual camera and stream dual camera, respectively; (right) the orthogonal regions (green) of the best camera location (blue): (top) streams and (bottom) semi-spheres

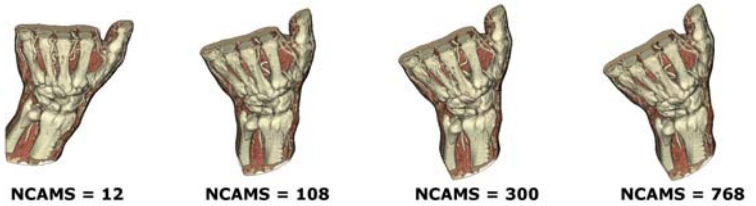

Fig. 7. Stability of camera's computation varying the number of sampled cameras with the hand data set

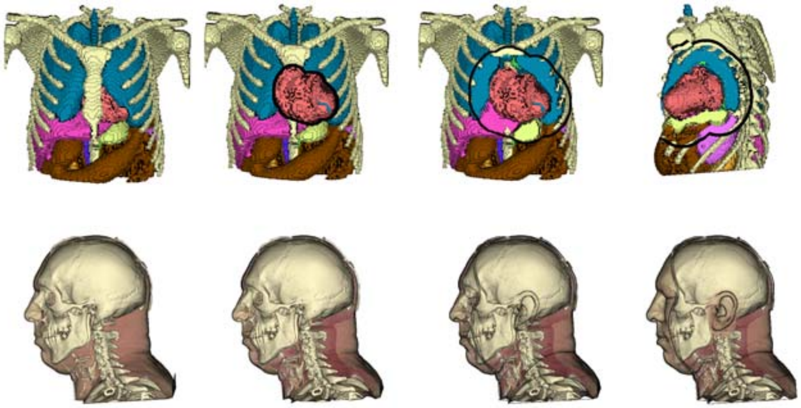

Fig. 8. Different cutaways of thorax (top) and VMHead (bottom) data sets. The first row shows visualizations with a classical raycasting (left), and with different w_{open} and w_{border} values. Second row shows from left-to-right, $w_{open} = 0$, $w_{open} = 24$ without context preservation, $w_{open} = 24$ with context preservation. The last is edited with the brush tool to enhance context regions as ear and nose.

GTX GPU with 768 MB of memory. The viewport size is 700 × 650. To test the methods, different synthetic and real data sets have been used. In all the tests, spheres of 12, 108, 300, 588 viewpoints has been used and two different orthogonal regions, stream band and semi-sphere has been fixed by the user. We have used several data sets with different sizes. For each data set, different features can be selected. The *thorax* data set represents a segmented phantom human body. *VMHead* is a CT human brain obtained from the Visible Human Project. *Foot* and *Hand* are non-segmented CT scan of a human foot and a human hand respectively.

First of all, we have evaluated the minimum number of cameras to be sampled in order to converge to an stable solution. We have tested the different data sets increasing the number of samples in the sphere, N, from 12 to 1200 (see Figure 7). In general, between 108 and 184 samples, the computed camera's positions became stable. In these cases, the initial step of the camera location is averaging between 0.75 s. and 1 second by measuring with different data sets and selecting various structures. Changes on camera descriptors weights and searches of the dual-cameras do not reduce the frame rates significantly. In the tested data sets, the stability of the dual-camera location depends directly on the variation of the number of samples on the bounding sphere as well as the used searching space.

Figure 8 shows different cutaway widths, with context preserving, border highlights and editing brush tools of the different data sets. We obtain interactive rates for all tested visualizations, between 50FPS and 60FPS.

7 Conclusions and Future Work

In this paper, we have presented a system for adaptive context volume visualization using automatic cutaways and dual-camera placements, based on GPU strategies. Our system provides new insights on the interaction with a focused structure to be analyzed in a volume context environment. It helps users to better understand the relationships between the focused structure and the context. Our approach enhances and extends the ClearView metaphor providing volumetric information in a cutaway context opening. The proposed adaptive cutaway exhibits the volume information contained in the context data, as stratified information surrounding the focused region. Also, we have proposed a new brushing widget to graduate interactively the contribution of the context in the final image. In addition, an efficient GPU algorithm is implemented to find the best two correlated viewpoints in the focused structure's bounding sphere.

Starting from this paper, we will continue our work in different directions. In this initial work, we concentrated our efforts on a particular sampling level, instead HealPix offers the possibility to refine the sampling space, in a hierarchical model. In the future, we will explore the capability of integrating the refinement process in our GPU-based system. On the other hand, we will attempt to obtain the best dual-camera view searching for the two views simultaneously.

Acknowledgements. This work has been partially funded by the project CICYT TIN2008-02903, by the research centers CREB of the UPC and the IBEC and under the grant SGR-2009-362 of the Generalitat de Catalunya.

References

1. Viola, I., Kanitsar, A., Gröller, E.: Importance-driven feature enhancement in volume visualization. IEEE Trans. on Visualization and CG 11, 408–418 (2005)
2. Bruckner, S., Gröller, E.: Volumeshop: An interactive system for direct volume illustration. In: IEEE Visualization 2005, pp. 671–678 (2005)
3. Rautek, P., Bruckner, S., Gröller, M.E., Viola, I.: Illustrative visualization: New technology or useless tautology? SIGGRAPH Comput. Graph. 42 (2008)
4. Krüger, J., Schneider, J., Westermann, R.: Clearview: An interactive context preserving hotspot visualization technique. In: IEEE Trans. on Visualization and Computer Graphics (Proc. Visualization / Information Visualization 2006), vol. 12 (2006)
5. Patel, D., Giertsen, C., Thurmond, J., Gröller, M.: Illustrative rendering of seismic data. In: Lensch, H., Bodo Rosenhahn, H.S. (eds.) Proceeding of Vision Modeling and Visualization, pp. 13–22 (2007)
6. Burns, M., Finkelstein, A.: Adaptive cutaways for comprehensible rendering of polygonal scenes. ACM Trans. Graph. 27, 1–7 (2008)
7. Wang, S., Kaufman, A.: Volume sculpting. In: ACM Symp. Interactive 3D Graphics, pp. 151–156 (1995)
8. Konrad-Verse, O., Preim, B., Littmann, A.: Virtual resection with a deformable cutting plane. In: Proc. of Simulation und Visualisierung, pp. 203–214 (2004)
9. Correa, C., Silver, D., Chen, M.: Feature aligned volume manipulation for illustration and visualization. IEEE Transactions on Visualization and Computer Graphics 12, 1069–1076 (2006)
10. Bruckner, S., Gröller, M.E.: Exploded views for volume data. IEEE Trans. on Visualization and Computer Graphics 12, 1077–1084 (2006)
11. Zhou, J., Hinz, M., Tonnies, K.: Focal region-guided feature-based volume rendering. In: Symp. on 3D Data Processing Visualization and Transmission, p. 87 (2002)
12. Bordoloi, U., Shen, H.: View selection for volume rendering. IEEE Visualization, 62 (2005)
13. Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T.: A feature-driven approach to locating optimal viewpoints for volume visualization. vis. 0, 63 (2005)
14. Viola, I., Feixas, M., Sbert, M., Gröller, E.: Importance-driven focus of attention. IEEE Trans. on Visualization and Computer Graphics 12, 933–940 (2006)
15. Mühler, K., Neugebauer, M., Tietjen, C., Preim, B.: Viewpoint selection for intervention planning. In: IEEE Symp. on Visualization 2007, pp. 267–274 (2007)
16. Vázquez, P.P., I Navazo, E.M.: Representative views and paths for volume models. In: Smart Graphics, pp. 106–117 (2008)
17. Kohlmann, P., Bruckner, S., Kanitsar, A., Gröller, E.: The livesync++: Enhancements of an interaction metaphor. Graphics Interface (08), 81–88
18. Polonsky, O., Patane, G., Biasotti, S., Gotsman, C., Spagnuolo, M.: What's in an image? The Visual Computer 21, 840–847 (2005)
19. Strengert, M., Kraus, M., Ertl, T.: Pyramid methods in gpu-based image processing. In: VMV 2006, pp. 169–176 (2006)
20. Gorski, K., Hivon, E., Banday, A., Wandelt, B., Hansen, F., Reinecke, M., Bartelman, M.: Healpix – a framework for high resolution discretization, and fast analysis of data distributed on the sphere. The Astrophysical Journal 622, 759 (2005)