View-Dependent Peel-Away Visualization for Volumetric Data

Åsmund Birkeland and Ivan Viola*
University of Bergen

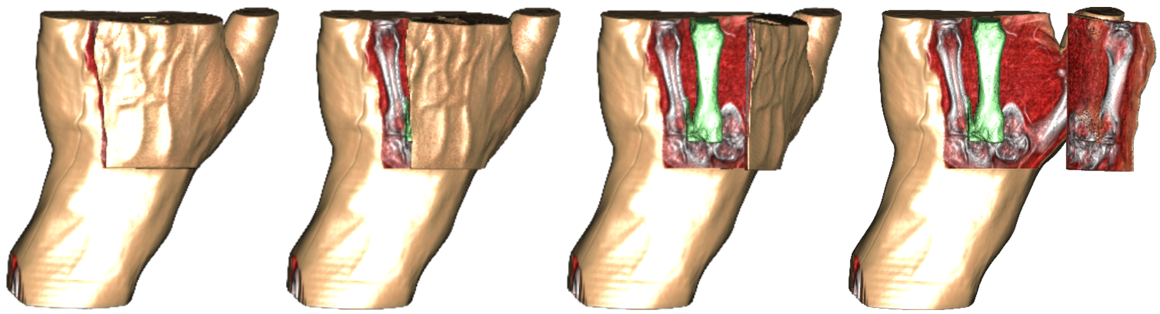


Figure 1: Automatically generated peel to reveal the meta carpal bone of the middle finger.

Abstract

In this paper a novel approach for peel-away visualizations is presented. The newly developed algorithm extends existing illustrative deformation approaches which are based on deformation templates and adds a new component of view-dependency of the peel region. The view-dependent property guarantees the viewer an unobstructed view on the inspected feature of interest. This is realized by rotating the deformation template so that the peeled-away segment always faces away from the viewer. Furthermore the new algorithm computes the underlying peel template on-the-fly, which allows animating the level of peeling. When structures of interest are tagged with segmentation masks, an automatic scaling and positioning of peel deformation templates allows guided navigation and clear view on structures in focus as well as feature-aligned peeling. The overall performance allows smooth interaction with reasonably sized datasets and peel templates as the implementation maximizes the utilization of computation power of modern GPUs.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

Keywords: illustrative visualization, peel-away, view-dependent techniques

1 Introduction

Traditional illustrators have invented a great selection of tools to better convey the information about complex three-dimensional shapes to the viewer. When displaying several structures defined over three-dimensional space it is often the case that one structure occludes another, whereas the occluded one can often be of higher interest to the viewer. This creates the challenge of showing the features of interest while providing an understandable representation of the surrounding structures. Techniques such as: peel-aways, cut-aways and exploded views, have been used by illustrators since the Renaissance to solve the problem of occluding structures.

Nowadays, three-dimensional volumetric structures are obtained digitally from various domains of science and medicine and are analyzed using computers. A challenge, with for example anatomical computer tomography scans, is to visualize the information in an easily perceptible way. Volumetric datasets are usually large and a good overview can be difficult to obtain when displaying two-dimensional slices only. A 3D representation of volumetric data can provide a better overview of complex structural arrangement as compared to 2D slice rendering. However when providing a 3D representation of an object, the features inside can get occluded by the outer structures. Therefore expressive illustration techniques are a strong inspiration in the development of interactive illustrative volume visualization approaches.

There are many existing techniques for dealing with occlusion in volume visualization of which the most standard technique is reducing the opacity of the outer structures using a transfer function. Such an approach changes the visual representation globally which leads to visual suppression of contextual information even in regions where context does not occlude the feature of interest. Here, cut-away illustration technique has been adapted for interactive visualization to tackle the problem of global context suppression. The cut-away is generated in a view-dependent way to guarantee that the feature of interest is visible from each viewpoint and only that parts of contextual structures are suppressed that occlude respective interesting feature.

The general drawback of cut-aways is that they in fact remove parts of data from the visualization. Other illustrative approaches such as exploded views or peel-aways avoid this weakness and attempt to show all data. The solution for resolving the occlusion is realized in a way so that the occluding context information is displaced from its original spatial location. These illustration techniques inspired the development of interactive visualization approaches known as illustrative deformation. They are using precomputed deformation templates that are placed into the spatial frame of reference of the volumetric dataset and displace contextual region driven by the displacement defined in the template. This approach results in a visually pleasing visualization. However, the specification of position of the template with respect to the data, its extent and orientation have to be explicitly defined by the user. Currently these approaches are lacking an automated controlling mechanism which will for each viewpoint guarantee clear view on the feature of interest without explicit involvement of the user.

The lack of controlling mechanism that guarantees a clear view on structures in focus for peel-away illustrative deformations is

*Email: aasmund.birkeland@gmail.com, ivan.viola@uib.no

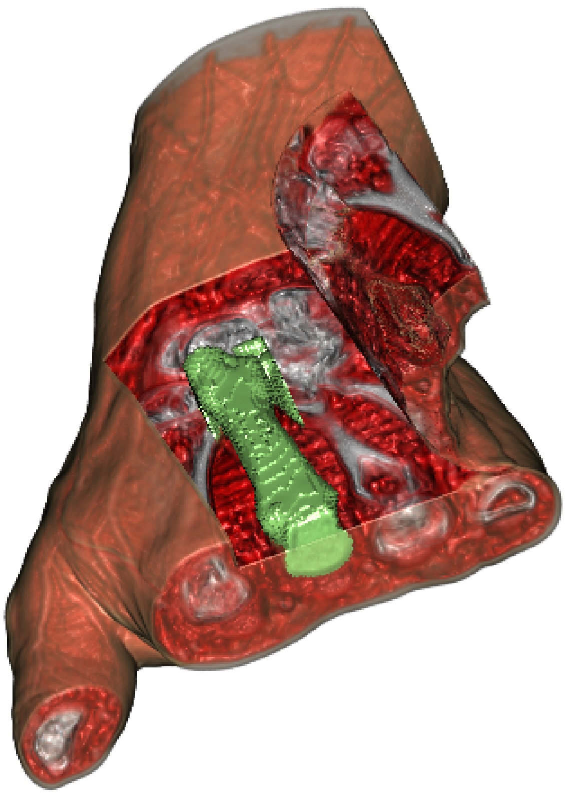


Figure 2: Peeling away the outer structures reveals the meta-carpal bone in the middle finger.

the main motivation for the research presented in this paper. We propose a solution that combines the best of both illustrative approaches, using view-dependent cut-aways and illustrative deformations. We guarantee a clear view on feature of interest and at the same time we do not remove any data from the visualization. Our view-dependent peel-aways provide visually pleasing visualizations that can be utilized for visual communication and presentation of interesting features within the volumetric data as shown in Figure 2.

2 Related Work

In recent years, traditional illustrations have inspired the field of illustrative volumetric visualization to incorporate the metaphors created by illustrators into the field of scientific visualization to resolve the problem of occlusion. In 3D rendering, insights can be obtained by adjusting the opacity of the outer structures. However, investigating the feature of interest can become difficult if a semi-transparent surface is rendered on top of the feature in focus. Therefore, a set of rules has to be considered. For example, that transparency should not be constant and should decrease towards edges of the outer structure [Diepstraten et al. 2002]. While the use of transparency can aid insight into volumetric dataset, using global parameters for adjusting transparency does not always provide satisfying results. Using geometrical structures can be used to select certain areas in the rendering scene to be represented differently or removed from the image entirely. This process is known as clipping and cut-aways which are used for exploration and visual communication respectively. Clipping is the process of removing structures, often defined by geometrical shapes, from the final image [Weiskopf et al. 2002]. In importance-driven volume rendering, cut-away is the automated process of removing structures to reveal features of interest. The idea is to visually suppress contextual structures between the viewpoint and the feature of interest [Viola et al. 2004].

Volume splitting is another technique for handling the problem of occlusion. Splitting is a technique inspired by clipping, where the volume is divided into several different parts. Each part can then be transformed and rendered separately. However, the different parts can still occlude the features of interest. Exploded views utilize the concept of splitting to transform each part so that the feature in focus becomes visible [Bruckner and Gröller 2006]. Proposed interactive visualization using exploding views metaphor, similarly to our approach, defines a scheme how individual parts have to be displaced. However when exploded views consist of many parts which are dislocated from their original position understanding the original spatial arrangement might become difficult.

Deforming a volumetric object can also be used to give focus to or to reveal interesting areas. Deformation can be divided into two separate types which deal with different problems, continuous and discontinuous deformation. Continuous deformation, also called shape deformation, is distorting the shape of an object to enhance the interesting features on the surface of the object. Ray deflectors [Kurzion and Yagel 1995] were designed to create continuous deformations. A ray deflector works as a gravity field which inflicts the rays in a ray caster. This enables an magnification of the area on which the deflector is applied. Shape deformation using magic lenses provides good view of both feature in focus and surrounding context [Wang et al. 2005], where ray bending is applied at a user selected area. The ray deflector was later extended to incorporate discontinuous deformations, such as cuts [Kurzion and Yagel 1996]. Discontinuous deformation extends its continuous counterpart by introducing cuts as means to explore features underneath the surface.

Defining discontinuous deformations can be realized in several different ways. McGuffin et al. propose using an extensive set of various types of deformation as means for browsing through volumetric data [McGuffin et al. 2003]. In the approach of McGuffin et al., points were used as primitives for rendering. This simplifies the problem of discontinuity because the empty space is not rendered and a forward displacement transformation of each sample point can be used.

The illustrative deformation developed by Correa et al. [Correa et al. 2006; Correa et al. 2007] makes use of discontinuous displacement maps to define the deformation in the rendering process. Pre-defined templates which represent the deformation are manually positioned into the rendering scene to illustrate peels, separators and other types of deformation. In this work, a template based deformation technique is utilized for fast exploration and automatically generating peels to reveal interesting features. The problem with this technique is the necessity of involvement of the user in the template placement, orientation, and scale. This has been addressed in our newly proposed approach. This technique has been developed in the scope of the master thesis [Birkeland 2008] which served as basis for the research presented in this article.

3 View-Dependent Peel-Aways

The strongest argument for the work presented here is to eliminate the unnecessary user interaction from illustrative deformation approach. Illustrative deformation is in general a flexible concept

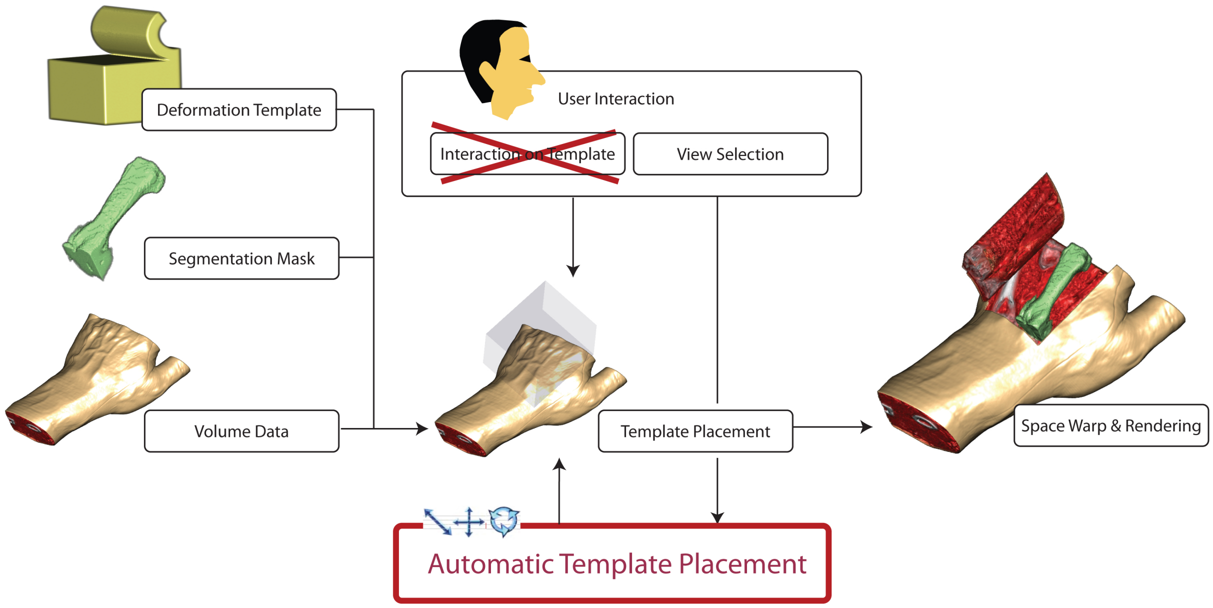


Figure 3: The peeling framework; volume data, segmentation mask and a deformation template are placed to the common reference frame. Previously the user would need to manually position and scale the deformation onto the volume. This is now replaced by an automatic template placement algorithm. After the template has been positioned, template aware ray casting is performed to render the final image.

where the deformation templates are pre-computed and can be easily integrated into traditional direct volume rendering. *Unnecessary user interaction* We consider the time-consuming specification of template position, its extent, and orientation with respect to the viewpoint. All these interactions can be automated as the viewpoint settings are known and structures of interest are well defined by segmentation masks.

Our view-dependent peel-away framework is depicted in Figure 3. Similarly to illustrative deformation, the input to our pipeline is the volumetric dataset with an associated segmentation masks which define our feature of interest. Our templates are initially defined analytically and their representation in the pipeline is a discrete displacement volume. The user defines the visual representations of the volumetric structures and the viewpoint from which she wants to analyze the three-dimensional structure of the feature of interest. In our pipeline, *unlike* the recent illustrative deformation approach, the operations related to the template placement into the frame of reference of the volume are computed instead of manual selection. For each viewpoint change the peel is animated from the *closed* peel to the *open* peel. The animation provides the user with information of the original spatial position of the peeled context structures. The level of necessary peel opening is retrieved from the specific peeling-template in order to give a clear view on the feature of interest, since each template require a different degree of peeling. The peeling direction is defined by the viewpoint so the peel does not open into the region between the viewpoint and the feature of interest to guarantee visibility of the feature in focus.

A detailed discussion on the generation of peel templates, which is the first computational step in our pipeline, is presented in the following Section 4. The discussion on individual components of the view-dependent automatic peel placement follows in Section 5.

4 Peel-Template Generation

When applying a deformation or displacement to a structure, the idea is to transform the desired elements to a different position. This transformation is here denoted the forward transformation of an element. We use the GPU for rendering and we have to face the fact that the fragment shader renders each pixel separately. This means that a voxel cannot be transformed to another position within the GPU-based ray caster. The shader retrieves samples from the volume at positions along the ray. Because of this limitation the *inverse displacement* is required. The *inverse displacement* provides the shader with the information of where the original voxel is positioned.

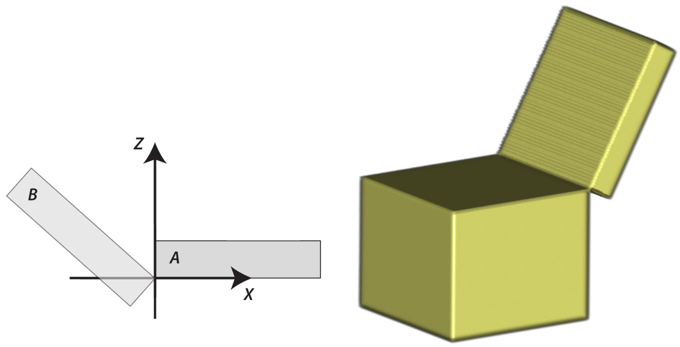
The peel-template is in essence a vector field that contains the inverse transformation of the peeled structure. To generate an inverse vector field, an invertible transformation function is required. The vector field is sampled at each point with the inverse transformation function. For defining discontinuous regions in the rendering process, a binary α-mask is generated. The displacement vectors are then stored in the RGB-channel and the α-mask is stored in the A-channel of a 3D RGBA texture.

For the rigid peel-template, the transformation function is the rotation matrix around the y-axis, as shown in Figure 4(a). The inverse matrix can be calculated in a straightforward way. The inverse transformation function for the rigid peel is calculated to be the following:

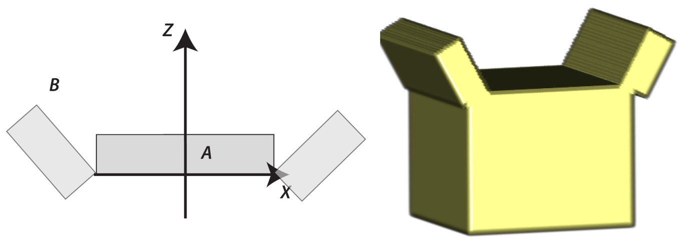
$$T_f^{-1}(x,y,z) = \frac{1}{\cos^2\phi + \sin^2\phi} \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

.

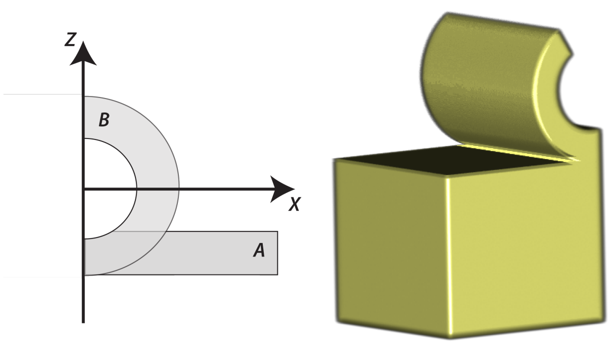
The discontinuous region is defined as the points which are above



(a) Rigid peel.



(b) Cabinet peel.



(c) Deformed peel.

Figure 4: Three types of peeling metaphors applied to a synthetic dataset.

the original normal of the plane and *below* the transformed normal. The cabinet peel-template divides the template into two parts. The transformation is then calculated in the same manner as the rigid peel where one part is the mirrored transformation of the other. In Figure 4(b) the cabinet peel is applied to a synthetic dataset.

The deformed peel is based on the peeling template from the work of McGuffin et. al. [McGuffin et al. 2003]. They use the forward transformation, which transform *(x,y,z)* to *(x',y',z')*, which is defined as

$$\begin{aligned} x' &= -z \cdot sin(\tfrac{x}{R}) \\ z' &= z \cdot cos(\tfrac{x}{R}) \\ y' &= y, \end{aligned} \qquad (2)$$

where R is the radius of the bend. To calculate the inverse displacement, the inverse of this transformation is calculated to be:

$$\begin{aligned} z &= \sqrt{x'^2 + z'^2} \\ x &= R \cdot cos^{-1}(\tfrac{x}{R}) \\ y &= y'. \end{aligned} \qquad (3)$$

For the deformation peel-template, the discontinuous region is not defined in the α-mask. The discontinuous region is generated in the rendering process as displacements outside the peel-template.

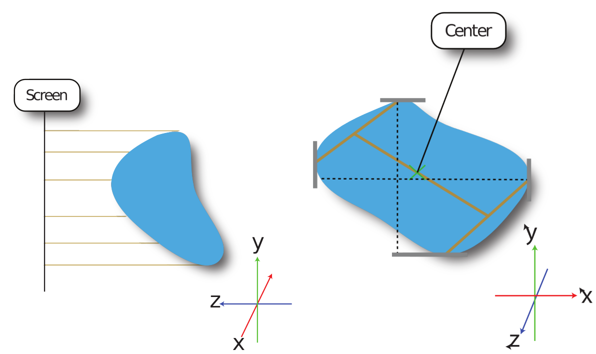
Because of the low resolution of the template, a template can be generated on-the-fly. This means that the peel can be animated. Animating the peel provides a better correlation between the peeled context and its original position, as shown in Figure 1.

After the peel-template has been generated, the next step is to place and scale the template to produce the desired peel. In the next section, the procedure for automatic positioning and scaling for according to user selected features will be presented.

5 Automatic Peel Placement

To reveal features of interest, the user does not need to control the position, scale or the bend of the peel. The user selects a feature of interest and the peel is automatically positioned and scaled according to the position and size of the feature. To ensure that the entire feature is revealed, the template must encapsulate the entire extent of the feature. To calculate the extent of the feature selected from the current view point, the feature is first projected onto the screen using a first-hit ray caster. The ray caster stores the first-hit coordinates in the RGB-channels of the output texture.

The size of the feature in focus is then determined by considering the first-hit coordinates at the extreme x and y positions of the projection. To estimate the logical center of the feature, an interpolation between the four first-hit coordinates at the extreme positions is interpolated. The process is illustrated in in Figure 5.

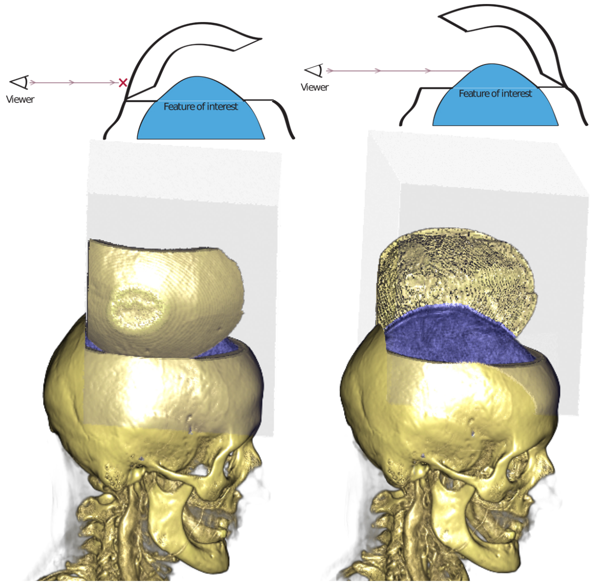


(a) (b)

Figure 5: Feature is projected onto the screen (a). Then the four extreme positions are estimated to calculate an approximate center (b) and extent of the feature.

Peeled structures can occlude the feature of interest if the peel is oriented towards the viewer. To ensure a clear view of the peeled structures and the structures underneath, the peel is oriented in a way to always bend away from the viewer, as shown in Figure 6.

The peel can still occlude other contextual areas. To provide the most information about the original volume, the peel could be additionally oriented in such a way that it covers the least amount of the original volume. The least occluding direction can be considered the direction which is has shortest distance from the center of the peel towards the border of the projection of the volume. To calculate the distance to the border, a distance transform of the projected volume would be generated. The direction would then be calculated as the gradient of the distance transform. This is not yet implemented into our proof-of-concept prototype.



(a) (b)

Figure 6: When the scene is rotated, the peel occludes the underlying structures (a). View-dependent re-orientation of the peel ensures the maximum visibility of the underlying structures while minimizing the transformation of the peel (b).

6 Rendering

The final step for visualizing view-dependent peel-aways is rendering. The rendering is carried out by using front-to-back ray caster with an extended sampling function. Regular sampling retrieves values along the ray from the volume at the current position. For template-based peel-aways, the sampling function must consider the displacement of the peel. The sampling function first looks up the value of the α-mask for the current position in the peel-template. The value encodes whether the current position has a valid inverse displacement. If the displacement is valid, the sample is retrieved from the original position. If the α-value encodes a discontinuous region, empty space is rendered.

The surface of the peel has two types of alignments. It can be aligned with the axis of the peel-template, or it can be aligned with a pre-defined feature. Axis aligned peels enable the user to cut through objects and examine the inner structures, while feature aligned peels provide a better overview of the structure in focus.

In volumetric visualization gradients are used for approximating normals for shading calculations. Gradients are calculated on-the-fly within the ray-caster using the central differences method. Previously this was considered to be too time-consuming for illustrative deformation. However, due to the increase in performance of dedicated graphics hardware, the gradients can be calculated on-the-fly while still achieving interactive framerates.

7 Results

In this work, several types of peel-templates have been implemented: rigid peel, cabinet peel and deformed peel. The rigid peel has the advantage of preserving the spatial relationship for the structures within the peel. Stair-case artifacts appear on the cut-surface of the peeled structure. These artifacts appear when low resolution of the peel-template is used compared to the volume resolution and because the discontinuous region is defined by a binary α-mask. Correa et al. suggested a solution to avoid this aliasing, by smoothing the α-mask. However, smoothing the mask has been considered too time consuming and reduces performance. In Figure 7, the rigid peel-template is applied to the hand dataset. When the view point changes, the peel-template is scaled to ensure the visibility of the feature of interest.

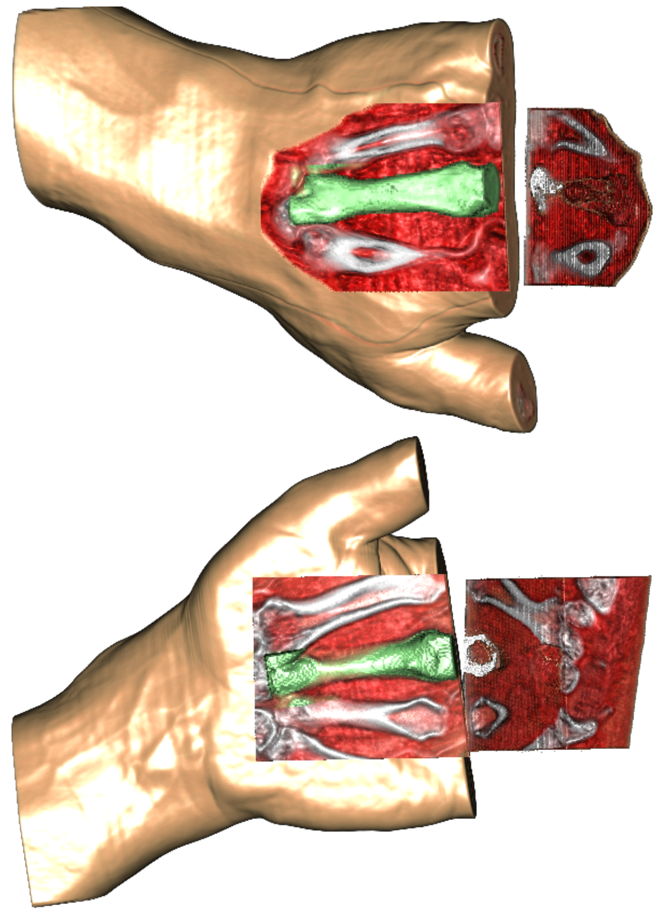


Figure 7: Rigid peel reveals the bone previously occluded. Images show peels from two different view points.

The cabinet peel is an extension of the rigid peel, where the left half of the transformed region is peeled to the left and the right half are peeled to the right. The cabinet peel-template contains the advantage of not distorting the spatial relationship between the structures within the peel. In Figure 8,

The deformation peel-template does not require an α-mask to define the discontinuous region. To prevent the same sample appearing at multiple positions, the inverse displacement outside the template is not considered as a valid displacement. Any point in the cut region would have a displacement outside the peel template. This is only possible for transformation which do not move elements below its original position according to the z-axis. Since the discontinuous region is not generated by a binary α-mask, the surface of the peel becomes much smoother than with the rigid or cabinet peel-template. This can be seen in Figure 9 and Figure 10.

To provide a better view of the peeled structure, the visual repre-

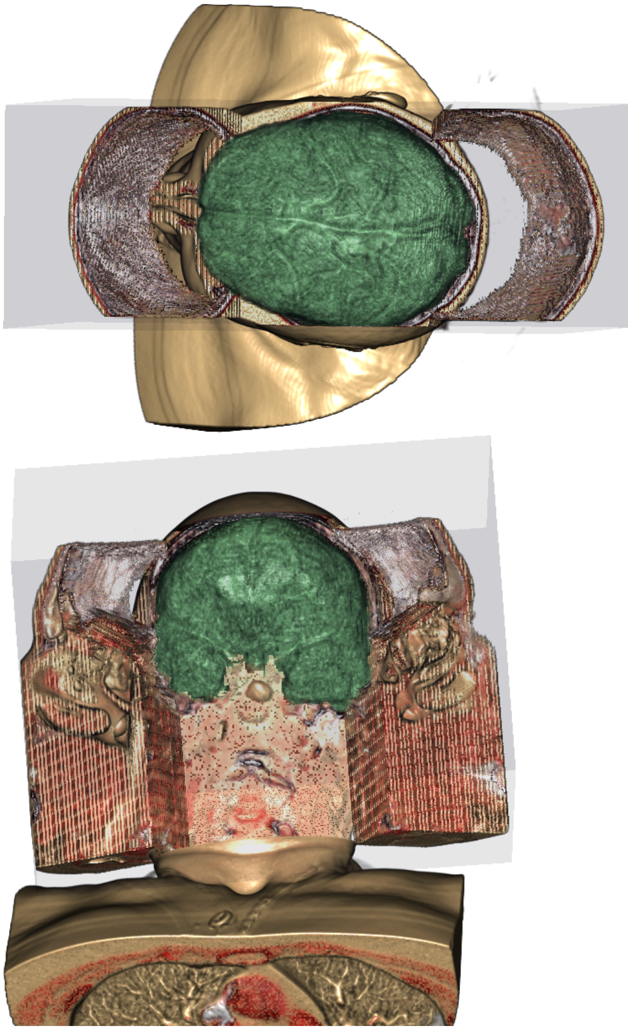


Figure 8: The cabinet peel-template provides a clear view of the brain.

sentation of the peeled context can be changed using an alternative transfer function. This helps separating the peeled structure from the original volume. In Figure 11, the representation of the peel is changed to a different color encoding.

Performance is an topic for consideration when creating interactive visualization systems. Table 1 shows interactive framerates are achieved with a reasonable sized peel-template. The benchmark was performed on Dell Precision workstation T7400 with an NVIDIA GeForce GTX280.

8 Conclusions

A technique for generating illustrative view-dependent peel-aways in volumetric visualization has been presented. The major focus has been to create peels which consider the position of the viewer. Locating and investigating structures within the human body can be difficult using basic rendering methods. In the field of medical education, illustrative renderings may be helpful to understand the human anatomy. Automatically generating peels to reveal features of interest can provide a fast, simple and understandable view of

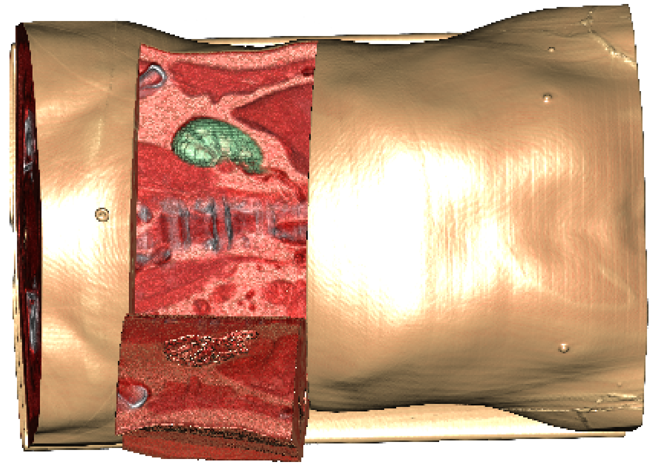


Figure 9: Revealing right kidney in the human torso dataset using the deformation peel-template. The size of the peel are adjusted by the depth of the kidney within the volume according to the current view point.

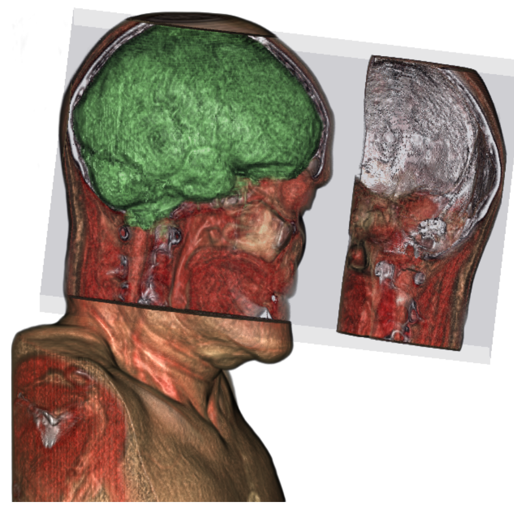


Figure 10: The brain in the head dataset made visible by applying the deformation peel-template.

the selected anatomical structure.

The displacement method in the presented system utilizes discontinuous displacement maps to generate peels within the rendering process. Since this work did not aim at simulating physical interaction with the rendered objects, the pre-defined peel-template suits the purpose of positioning and orienting peels according to the view-point. The template-based peeling has been proved to be capable of visualizing peels in direct volume rendering at interactive framerates as long as the resolution and scale of the peel template remains smaller than approximatly half the resolution and size of the test volumes.

In a GPU-based template-aware ray-caster, the inverse transformation is required for sampling from the original position in the volume. Because of the simple transformation calculation analytically, and the low resolution of the peel-template and the corresponding α-mask, the peel-template can be calculated on-the-fly. Interactively generating the peel-template enables animation of the peel. Animation helps understanding the original position of the trans-

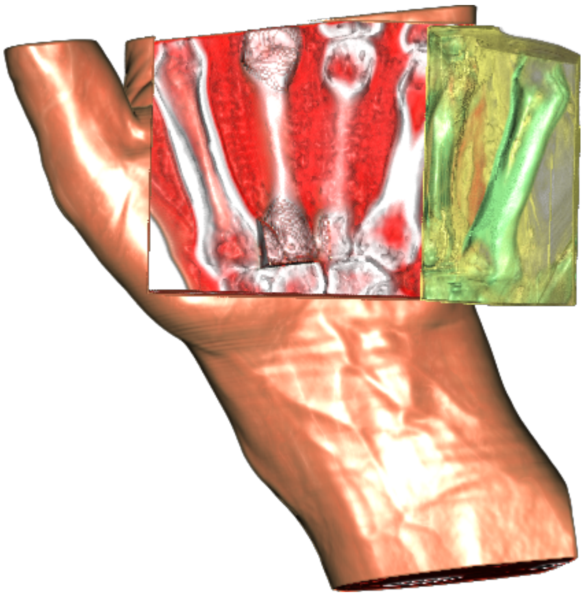


Figure 11: Applying a different transfer function to the peeled context region.

Template resolution	Dataset	FPS
$64 \times 64 \times 64$	Hand $(244 \times 124 \times 257)$	8 FPS
$256 \times 256 \times 256$	Hand	1-2 FPS
$64 \times 64 \times 64$	Head $(256 \times 256 \times 230)$	8 FPS
$256 \times 256 \times 256$	Head	1-2 FPS
$64 \times 64 \times 64$	Torso $(256 \times 256 \times 556)$	5 FPS
$256 \times 256 \times 256$	Torso	1-2 FPS

Table 1: Benchmark for the different datasets using different template resolutions.

formed structures.

Due to the discrete binary α-mask for defining discontinuous regions, and the low resolution of the peel-template, stair-case artifacts appear at the surface of the peeled structure. Stair-case artifacts distort the view of the peeled structure, however it clearly conveys that the peel was artificially transformed. Therefore we can conclude that although not visually pleasing, this artifact helps communicate which contextual region has been displaced.

A smooth deformed peel creates a more pleasing visual result compared to aliased rigid transformation, but the deformed peel suffers from a distortion of the spatial relationship within the peeled structure. Smoothening the surface of both the rigid peel template and cabinet peel template can produce results which holds the advantage from both the rigidness and smoothness from the deformed peel.

9 Future Work

The techniques presented in this paper provide a basis for exploration and visual communication using illustration techniques defined as templates. The prototype implemented for this work does not support more than one peel at a time. A future endeavour would be to extend the system to handle multiple displacement templates.

For animation the size of the template needs to be small enough to be able to generate the displacement vectors interactively. Todays hardware usually contain more than one CPU core, and generating templates is a paralellizable process since the displacement vectors does not influence each other. This means that a multi threaded process of generating templates can be implemented for larger, more sophisticated complex templates.

Acknowledgements

I would like to thank Stephan Bruckner for providing support on the VolumeShop source code.

References

BIRKELAND, Å., 2008. View-dependent peel-away visualization for volumetric data.

BRUCKNER, S., AND GRÖLLER, M. E. 2006. Exploded views for volume data. *IEEE Transaction on Visualization and Computer Graphics 12*, 5, 1077–1084.

CORREA, C. D., SILVER, D., AND CHEN, M. 2006. Discontinuous displacement mapping for volume graphics. In *Proceedings of Eurographics / IEEE VGTC Workshop on Volume Graphics*, 9–16.

CORREA, C., D.SILVER, AND CHEN, M. 2007. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics 13*, 6, 1320–1327.

DIEPSTRATEN, J., WEISKOPF, D., AND ERTL, T. 2002. Transparency in interactive technical illustrations. *Computer Graphics Forum 21*, 3, 317–326.

KURZION, Y., AND YAGEL, R. 1995. Space deformation using ray deflectors. In *Eurographics Rendering Workshop 1995*.

KURZION, Y., AND YAGEL, R. 1996. Continuous and discontinuous deformation using ray deflectors. In *Proceedings of GRAPHICON'96*, 102–110.

MCGUFFIN, M., TANCAU, L., AND BALAKRISHNAN, R. 2003. Using deformations for browsing volumetric data. In *Proceedings of IEEE Visualization*, 401–408.

VIOLA, I., KANITSAR, A., AND GRÖLLER, M. E. 2004. Importance-driven volume rendering. In *Proceedings of IEEE Visualization*, 139–146.

WANG, L., ZHAO, Y., MUELLER, K., AND KAUFMAN, A. 2005. The magic volume lens: an interactive focus+context technique for volume rendering. In *Proceedings of IEEE Visualization*, 367–374.

WEISKOPF, D., ENGEL, K., AND ERTL, T. 2002. Volume clipping via per-fragment operations in texture-based volume visualization. In *Proceedings of IEEE Visualization*.