Internal Labels as Shape Cues for Medical Illustration

Timo Ropinski, Jörg-Stefan Praßni, Jan Roters, Klaus Hinrichs

Visualization and Computer Graphics Working Group (VisCG),
Westfälische Wilhelms-Universität Münster
Einsteinstraße 62, 48149 Münster, Germany
Email: {ropinski, j_prass01, j_rote01, khh}@math.uni-muenster.de

Abstract

In this paper we describe an interactive labeling algorithm, which allows to integrate internal 3D labels into medical visualizations generated from volumetric data sets. The proposed algorithm is motivated by internal labeling techniques found in anatomical atlases, and in contrast to existing algorithms it provides additional shape cues by fitting internal labels to the depth structure of their associated objects. In this paper we discuss guidelines for the layout of internal 3D labels and describe our labeling algorithm, which extends 2D shape fitting and introduces 3D shape fitting. Furthermore we propose related interaction concepts and provide application examples.

1 Introduction

Illustration is an important technique widely used in medical text books to communicate anatomical structures. In recent years computer-assisted interactive generation of medical illustrations from volume data sets acquired by medical scanners has advanced significantly [3, 4, 11]. Textual annotations are essential in order to assign descriptive labels to the objects of interest and thus ease identification of different parts of an illustration. Further on annotating medical data sets is an everyday task performed by radiologists during medical diagnosis. Since for intervention planing and other application scenarios it is important that visualizations can be enriched by labels in a meaningful way, several algorithms have been proposed. Most of these algorithms are inspired by existing illustrations and mimic the layout of the labels found for instance in medical text books like Gray's anatomy atlas [6]. They can be classified into three groups: internal, external and hybrid label layout algorithms. The latter incorporate both internal and external labels. While internal labels are spatially bound to the object of interest, external labels are displayed besides the object and associated with the object of interest using a connection line. Internal labels have the advantage that they allow an easy visual association with the object of interest, since they are placed on top of it and no connection lines are required. In contrast to external labels they occlude parts of the object of interest. Especially for objects with a varying depth structure, placing an internal label on top of it can make the spatial comprehension cognitively more challenging. In medical text books 3D labels are used to deal with this problem. These 3D labels do not only match the shape of the projection of the object of interest, but also match its 3D shape, i.e., the depth structure. As it can be seen in Figure 1, the shape of the objects of interest is emphasized by the distortion of the textual annotation associated with them. For instance is the font of the label *Thoracic aorta* at the ,lower middle of the image, distorted in a way that propagates that the vessel has a cylindrical shape. In this paper we propose an interactive algorithm, which allows to incorporate such 3D labels into medical illustrations generated from volumetric data sets. Compared to the usage of flat 2D labels, applying our technique allows to provide additional shape cues introduced by these internal 3D labels. Based on the proposed labeling algorithm we will also introduce interaction metaphors supporting an easy manual integration of labels into medical visualizations.

The remainder of this paper is structured as follows. The next section discusses related work about labeling, especially in the context of medical illustration. In Section 3 we will present some guidelines for the layout of internal 3D labels, which we have derived from illustrations found in medical text books. The proposed algorithm for generating

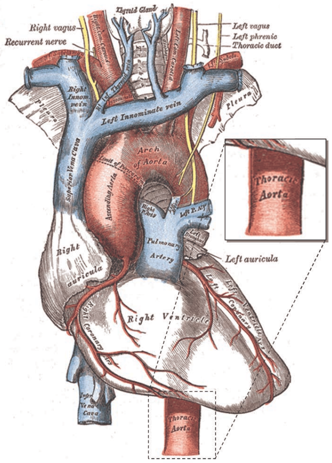


Figure 1: Usage of internal 3D labels as found in the Gray's anatomy atlas [6]. The shape of the structures is emphasized through the text distortion. For instance, the distortion of *Thoracic Aorta* shows that the vessel has a cylindrical shape.

internal 3D labels is described in Section 4, while the label rendering is described in Section 5. Section 6 briefly describes some interaction concepts that can be used to manually integrate labels and edit an existing label layout. Application examples of our technique are discussed in Section 7 before the paper concludes in Section 8.

2 Related Work

Bell et al. [2] were the first who investigated how meaningful annotations can be integrated into virtual environments. They have already identified the main problems of interactive labeling algorithms, which are caused due to changing viewing parameters and possibly occluding objects. Hartmann et al. [7] were the first to address the occlusion problem. While Bell et al. exclusively use internal labels drawn centered on top of the objects of interest, Hartmann et al. introduce floating labels. Floating labels are external labels which change the position when the object of interest is moved or the viewing parameters have been changed.

In the following years Hartmann et al. have further improved their label placing algorithms and have combined internal and external labels [1, 13]. This combination has revealed the need for aesthetic label layout algorithms which produce a visually pleasing placement as found in manually created illustrations [8, 14].

In 2006 Götzelmann et al. [12] have presented an agent-based approach used to deform the text path of internal labels in a manner that the 2D shape of a label fits the 2D shape of the object of interest. This approach has been incorporated into our extension for 2D shape fitting, which is described in Subsection 4.1.

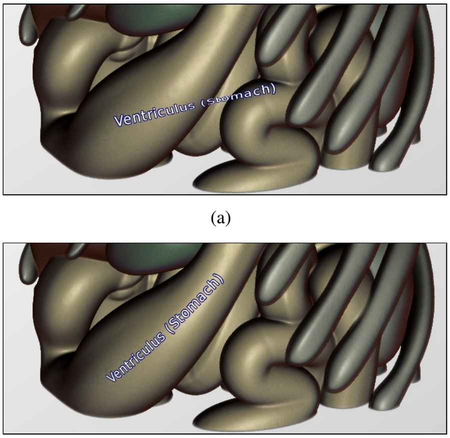
Based on the basic work done for investigating how to achieve interactive annotations, labels have been integrated into various applications, e.g., for medical illustration [4] or in general to combine text and images [5]. To our knowledge the only work towards providing 3D shape cues by means of labels is the application of labels to 3D city models done by Maass and Döllner [10]. They project viewpoint-depending labels onto the bounding boxes of building objects. When the viewpoint changes, the labels are repositioned, and the authors state that this label movement provides shape cues, i.e., the label floating over the invisible bounding box confines the borders of a building.

3 Guidelines for Internal 3D Label Layout

As mentioned above different hybrid label layouts have been described by Götzelmann et al. as well as Hartmann et al. [8, 14]. In this paper we introduce design rules for the layout of internal labels, whereas we do not exclusively focus on the positioning of internal labels but also on their 3D orientation. The presented guidelines are motivated by the layout of internal labels as found in medical illustrations.

As already proposed in [12] for internal 2D labels, internal 3D labels should also fit the 2D shape of the object of interest. A possible way to achieve this goal is to orient a label along the medial axis of its object of interest as it is defined in image space. This criterion ensures a 2D association between an object and its label solely by this 2D path. However, when extending the labels to 3D, additional layout guidelines should be taken into account in order to achieve a satisfying label placement.

First of all, when labels are distorted based on the underlying surface, it has to be ensured that the



(a)

(b)

Figure 2: Internal 3D labels crossing contours may lead to unwanted distortion and have a disturbing effect (a). By avoiding contour crossing this distortion can be prevented (b).

labels still remain readable. Therefore a tradeoff between label distortion and 3D shape fitting, i.e., alignment along the associated surface, has to be taken into account. In order to ensure readability a very bumpy surface should not be labeled by simply projecting a label onto it. Moreover the characteristic surface structures should be considered by omitting the bumpy arbitrary surface shifts. A possible realization would be using Bezier surfaces to approximate the surface of the object of interest. By using Bezier patches having a high polynomial degree, the surface is approximated very closely and surface details will be reflected. In contrast, when minimizing the polynomial degree of the approximating Bezier surface, it acts as a smoothing filter only reflecting the larger scale surface structure. In Subsection 4.2 we will describe how we will take this approximation into account.

Besides bumpy surfaces, an internal 3D label may also be distorted when crossing a contour (see Figure 2). Therefore internal 3D labels should be positioned in a way that they do not cross contours of the objects of interest.

Furthermore the perspective distortion of an internal label should be minimized. Therefore the layout should try to maximize the number of labels which are aligned almost perpendicular to the viewing direction in order to improve readability.

In the following section we will explain how to follow these guidelines, by among others exploiting object normals as well as contour detection.

4 Generating 3D Labels

By considering the guidelines proposed in the previous section, we have developed a labeling algorithm for automatically enriching a medical volume visualization with internal 3D labels. The proposed algorithm combines a 2D and a 3D shape fitting approach. With the 2D shape fitting we ensure that the path of the label matches the shape of the projection of the current object of interest given in image space. After this path has been determined we analyze the depth structure at the appropriate positions and generate Bezier patches to fit the 3D shape. Afterwards the label itself is rendered by texturing the generated patches. In the next two subsections we describe the necessary steps for performing the 2D and the 3D shape fitting. The workflow of the algorithm is illustrated in Figure 3.

4.1 2D Shape Fitting

To perform the 2D shape fitting, we use a modified version of the agent-based approach proposed in [12]. After the rendering parameters (RP), e.g., the transfer function, have been specified, we render a segmented volumetric data set into an *ID map* as well as a *depth map*. In the ID map each pixel has a unique color which has been associated with the segment it belongs to. Thus the region covered by each segment in image space can be determined, and its medial axis can be calculated. To comply with the guidelines of the previous section and to ensure that the internal labels do not cross any contour edges, we apply an image-based contour detection before calculating the medial axis. This contour detection is performed by applying a 3×3 filter kernel to the depth map in order to identify discontinuities of the depth values. Pixels having a high depth discontinuity belong to contour edges and are drawn as black pixels onto the ID map, resulting in the *edged ID map*. Based on the edged ID map we can perform the medial axis calculation, which is done similar to the technique described in [1]. Since we have initially super-imposed the contour edges, the resulting *distance map* contains no medial axis crossing any contour edges. However, using our

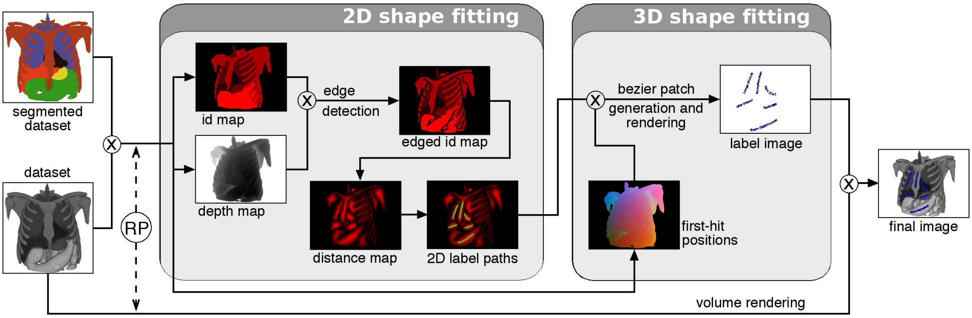


Figure 3: **Workflow** of our algorithm, with all steps used to render internal 3D labels. After the rendering parameters (RP) have been set, a 2D shape fitting is performed to find a label path matching the shape of the object of interest, before 3D shape fitting is used to match its depth structure.

distance map we can apply the algorithm as proposed in [12] to determine the 2D label paths. The goal of this technique is to find an approximation of the medial axis implicitly defined by the distance map. The used approach first determines for each segment the pixel with the maximum distance to the segment's border. In the next step, in contrast to the previous approach we do not consider all directions on a circle around the current position, but look only at those directions with maximal distance to the border to find the 2D path. This can be done by using our intermediate maps, which are generated when calculating the final distance maps. Since in the intermediate maps we have the distances separated for the three main directions, i.e., horizontal, vertical and diagonal. Thus by exploiting the current position and the determined direction, an arc with a certain angle and a given radius defines the pixels that are checked, in order to determine the two pixels among them having the maximum distance to the segments border. This process is repeated iteratively until the desired number of control points is found or the border of the segment is reached. In addition to the technique described in [12], we also consider the surface normals in order to ensure that labels do not appear on surfaces being almost parallel to the viewing direction, since their perspective distortion would be too high. Finally, the 2D label path is calculated by using a 2D fitting curve defined by the estimated control points.

With the thus computed 2D label paths we enter the next stage of our algorithm to perform the 3D shape fitting.

4.2 3D Shape Fitting

Now that the 2D label paths are available we take into account the underlying depth structure in order to generate Bezier patches approximating a segment's surface. To determine the control points for the used Bezier patches, we consider the volume coordinates for each pixel. When using GPU-based ray-casting [9] as in our system, the volume coordinates for a surface can be easily obtained by writing the first-hit positions into a 2D texture. Thus we can fetch for each point lying on the 2D path its corresponding volume coordinate and get the resulting 3D path. This 3D path is approximated by a 3D fitting curve defined by the control points later referred to as s_i and can be thought of as an approximation of the medial axis of the Bezier surface used to project the labels onto. Based on this medial axis we construct the Bezier patches as illustrated in Figure 4. As it is shown the s_i serve as control points for the Bezier patches. Additional control points are generated by considering the main normalized path direction of the current segment l as well as the normalized gradient g at each s_i. The distance of the newly computed control points to the s_i is given by the volume space offset h. Thus we can construct the outermost control points for the Bezier patches by calculating $m = h \cdot (l \times g)$ as well as $-m = -h \cdot (l \times g)$ and projecting these points back on the object's surface as shown in Figure 4. Since these are the outermost control points, h directly determines the label height. Depending on the degree of the Bezier patches we may add additional control points along $l \times g$ or between two successive

s_i. While a higher horizontal degree, i.e., along the main path direction, may be reasonable a high vertical degree, i.e., along $l \times g$, is usually not necessary. Instead a higher vertical degree may result in decreased readability through high perspective distortion.

5 Label Rendering

Based on the Bezier patch control points derived as explained in the previous section, the Bezier patches are generated to serve as a surface for the labels. The actual label rendering is done by exploiting 2D texturing functionality. Therefore we simply use the Bezier patch parameters s and t as texture coordinates in order to apply the labels, which we have rendered into a bitmap initially. During this bitmap generation we render the text with twice the desired font size. This ensures that we get smoother looking labels, since we can exploit the filtering capabilities of the texturing hardware during the minification process.

To increase contrast and further enhance readability we additionally render a halo around each letter. This is also done in the preprocessing rendering pass where the label textures are generated. Each label is rendered four times using the halo color before rendering it once in the current font color. During each halo rendering pass the label is shifted by one pixel in the four main directions. This ensures that the text rendered on top of this halo is surrounded by the halo color in all directions. The effect of the halo color becomes clearly visible in the Figures shown in Section 7.

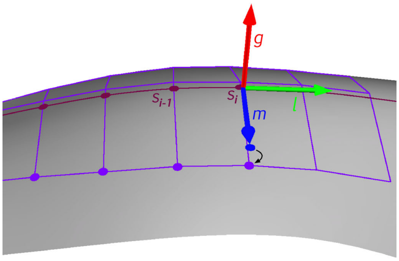
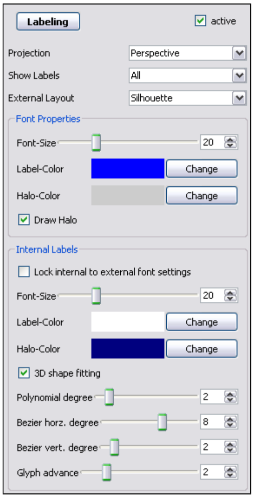


Figure 4: The control points for the Bezier patches are generated by adding points lying on the surface vertically perpendicular to the respective gradient.



☑ active

Projection: Perspective
Show Labels: All
External Layout: Silhouette

Font Properties
Font-Size: 20
Label-Color
Halo-Color
☑ Draw Halo

Internal Labels
☐ Lock internal to external font settings
Font-Size: 20
Label-Color
Halo-Color
☑ 3D shape fitting
Polynomial degree: 2
Bezier horz. degree: 8
Bezier vert. degree: 2
Glyph advance: 2

Figure 5: The graphical user interface used for interactive label manipulation. The user can change type, text and position of each label as well as change the rendering parameters.

During rendering we possibly might run into occlusion issues. Since the control points of the used Bezier patch lie on the object's surface and the patch lies in the convex hull of its control points, it may possibly intersect the surface. In these intersection regions the text would be occluded by the objects of interest. Therefore we ensure that the depth test is disabled when rendering the labels.

6 Interactive Label Manipulation

Although the automatically generated label layout is sufficient for most illustration cases, sometimes it is desirable to apply modifications manually or even integrate labels manually as done by radiologists during medical diagnosis. Therefore we have integrated simple though effective interaction concepts into our user interface which allow the user to perform the label positioning manually (see Figure 6) and to modify the most important label properties (see Figure 5).

The automatic layout proposed by our system is a hybrid layout, where internal 3D labels are used for those objects of interest which provide sufficient screen space. All other objects are annotated by using external labels. The user can reposition internal as well as external labels via drag-and-

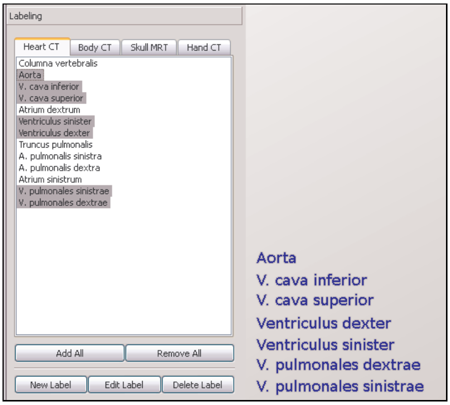


Figure 6: When dealing with non-segmented data sets, the user may select a series and add labels from that series manually to the image. Furthermore presets for frequent series can be modified by adding or removing labels.

drop, whereas the 3D shape of the internal labels is adapted to their current position automatically. In cases where the user drags an internal label outside the screen space occupied by its associated object of interest, the label turns into an external label, i.e., a connection line is added. By dragging the label back to *its* object of interest it can be reverted into an internal 3D label. Thus the user can change type and position of all labels.

The automatic layout algorithm works only in those cases where a segmentation for the current data set is available. Otherwise it is not possible to generate an ID map, which is required to calculate the label positions (see Subsection 4.1). However, often in medical diagnosis when annotation is required, no segmentation is present. For these everyday tasks we have integrated a simple though effective manual annotation technique. As it is shown in Figure 6 the user interface provides predefined label sets for different cases of diagnosis, e.g., heart CT or head MRI. These sets contain the labels which the physician expects to use during a diagnosis based on the given data set. While it is possible to add and remove label sets, they can also be modified by adding, deleting or editing labels in order to serve the needs of the physician. Once the required label sets are defined, a physician may add the labels contained in a set to the current image. This can be either done by pressing the *Add All* button or by applying a possibly multiple selection in the label set. Once the labels to be used have been specified, they appear in the lower right corner of the 3D canvas and can be dragged to the desired position. As described in Subsection 4.2, the labels also match the surface structure of the underlying objects. After the labels have been positioned, their coordinates may be saved in order to reproduce the renderings. Furthermore rendering parameters, e.g., the transfer function or the thresholding, can be changed during this task in order to be able to associate labels with internal structures.

Further on the user may alter the most important label properties. As it can be seen in Figure 5 the user may select which subset of labels is to be shown: all, internal only or external only. Additionally she may select the type of external label layout - currently silhouette layout as well as side layout is supported. Silhouette layout aligns the external labels along the objects silhouette, side layout renders the labels on the left and right side of the image. To adjust the look of the individual labels, font size, label color and halo color can be adapted. Furthermore the halo can be switched on and off. Changing the font as well as halo color is especially necessary for internal labels in order to increase the contrast between the label and its background color given by its object of interest. Therefore these options can be changed separately for internal labels. To adjust the shape of the internal labels, 3D shape fitting can be switched on and off and can be modified by adapting the degrees of the used fitting curves.

The text of a label can also be changed easily by performing a double click on top of it. All changes made are saved within an XML document which is associated with the segmented data set.

7 Application Examples

With the techniques proposed in the previous sections we are now able to integrate internal 3D labels into existing medical applications. As noted above, this approach is motivated by existing medical illustrations where text distortion provides an additional shape cue. In Figure 7 a comparison of 2D and 3D internal labels is shown. In both cases the text path is determined by exploiting the shape of the object of interest given in image space. In contrast to the 2D labels the 3D labels allow a better shape perception. This becomes clear when inspecting the region

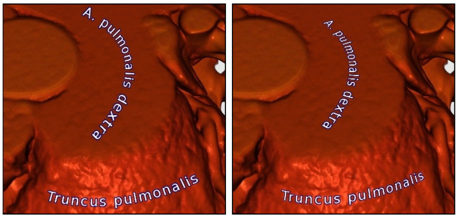


Figure 7: Comparison of internal 2D labels (left) and the introduced 3D labels (right). Although shading provides shape cues in both cases, the *floating* labels in the left column make estimation of distances cognitively more challenging.

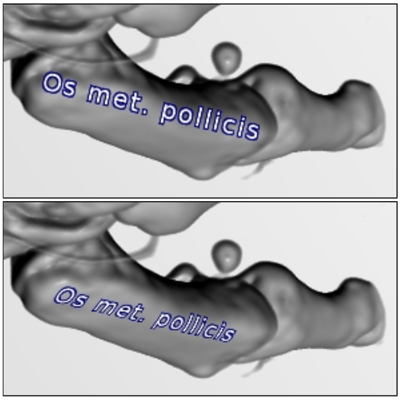


Figure 8: Comparison of internal 2D labels (top) and the introduced 3D labels (bottom). The internal 3D labels smoothly approximate the surface structure.

labeled as *A. pulmonalis dextra*. In the left image it is rather difficult to estimate the orientation of the cutting plane, the label is associated with. When using internal 3D labels the text distortion gives an additional cue for comprehending this orientation. Figure 8 shows a similar effect when visualizing curved surfaces. With 3D labels the curvature of the surface comes out more clearly, since the text is distorted accordingly. For instance when inspecting the distortion of the *ll* in *pollicis*. All these effects can be enhanced by using an appropriate label placement with the interaction techniques discussed in the previous section.

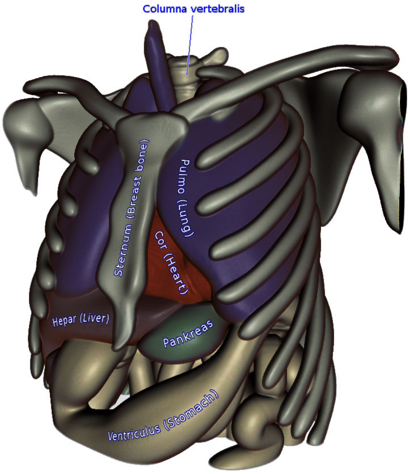


Figure 9: Application of our internal labeling technique to the NCAT phantom data set.

In Figure 9 the application of our technique to the NCAT phantom data set is shown. Since the *Columna vertebralis* does not provide enough screen space for positioning an internal label in the current view, an external label has been used automatically. Again the effects of surface structure emphasize becomes visible. One may argue that undistorted 2D labels would result in better readability. However, it must be considered that there is a tradeoff between the readability of the text and the perception, or readability, of the whole illustration. Obviously the best label readability would be achieved, when not even using 2D shape fitting and just integrating horizontal labels. When looking at medical text books, it can be seen that the reduced readability is accepted and 2D as well as 3D shape fitting is used. This approach has been evolved over the years and ensures besides improved perception of the illustration the characteristic style of medical illustrations. In addition to the usage of text distortion in classical medical illustrations, our approach also allows the illustrator to easily tune the parameters in order to get the most convincing representation. For instance, the degree of the fitting curves can be changed manually in order to emphasize or deemphasize certain spatial structures.

Other examples of hybrid label layouts are shown in Figure 10 and Figure 11. In contrast to the other figures, where we exploit Phong shading, a quantized toon shader has been used in Figure 10. To further enhance the illustrative effect the silhouette has been highlighted. In addition to the improved

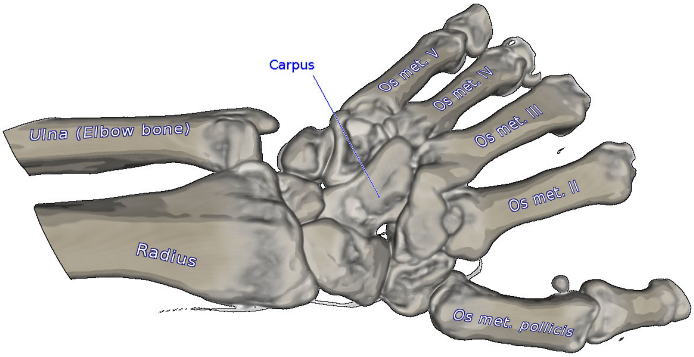


Figure 10: An automatically annotated illustration of a CT scan of a human hand. The internal labels are positioned appropriately and fit smoothly to the surface of the metacarpus bones.

shape perception through internal 3D labels, Figure 11 shows also that the association between a label and its object of interest is cognitively less demanding when using internal labels in comparison to external labels.

We have shown the images of Figure 9-11 as well as the corresponding versions using internal 2D labels to our medical partners. We got very positive comments and the physicians especially liked the internal 3D label approach to be used in anatomical illustrations containing parts of the skeleton as in Figure 10.

8 Conclusion

In this paper we have proposed an interactive algorithm for the generation of internal 3D labels. The technique has been motivated by labeling techniques found in medical illustrations and is also applicable during the interactive annotation process as used in medical diagnosis. In contrast to commonly used internal 2D labels, 3D labels have the benefit that they may provide additional shape cues. Since the distortion of the text being visualized on top of the object of interest is influenced directly by its 3D shape, spatial comprehension is expected to be improved. We have presented guidelines for the layout of internal 3D labels, have described our interactive rendering technique and presented some application examples.

Since text distortion may also influence readability, a detailed evaluation is necessary in order to find an optimal tradeoff between readability and spatial comprehension. However, it should be noted again that optimal readability can only be achieved when using horizontal labels, without applying any shape fitting. Since this would make it more difficult to perceive the association between the objects and the labels as well as the object's shape, we believe that shape fitting is a good way to enrich visualizations having internal labels. Although we already got positive comments about the interactive annotation mechanism from radiologists, we plan to evaluate this technique in order to further improve its usability for medical diagnosis.

While in this paper we focussed on volumetric visualization, it should be stated that the proposed technique can also be applied to polygonal rendering. Moreover it can be used in any visualization setup, as long as a depth image and eventually a segmentation is available.

Acknowledgements

This work was partly supported by grants from the Deutsche Forschungsgemeinschaft (DFG), SFB 656 MoBil Münster, Germany (project Z1).

References

[1] Kamran Ali, Knut Hartmann, and Thomas Strothotte. Label Layout for Interactive 3D Illustrations. In *Proceedings of the 13th In-*

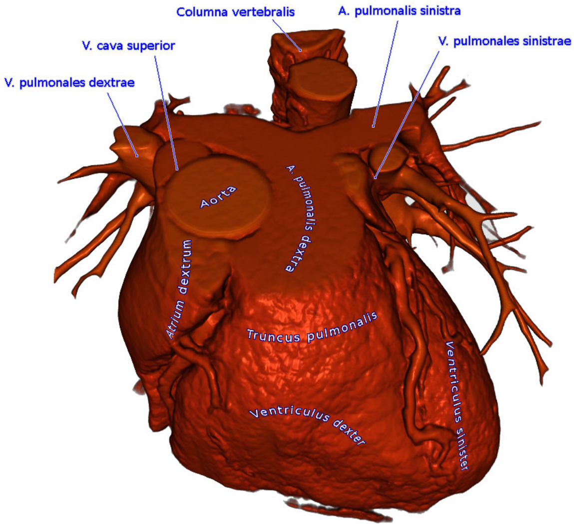


Figure 11: Manually annotated visualization of a CT scan of the human heart. A hybrid label layout is used, while the internal labels are represented as 3D labels providing additional shape cues.

ternational Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG05), pages 1–8. UNION Agency - Science Press, 2005.

[2] Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 101–110, New York, NY, USA, 2001. ACM Press.

[3] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and Meister Eduard Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, 11 2006.

[4] Stefan Bruckner and Meister Eduard Gröller. Volumeshop: An interactive system for direct volume illustration. In *Proceedings of IEEE Visualization 2005*, pages 671–678, 2005.

[5] T. Götzelmann, M. Götze, K. Ali, K. Hartmann, and Th. Strothotte. Annotating Images through Adaptation: An Integrated Text Authoring and Illustration Framework. *Journal of the WSCG*, 15(1), 2007. (in print).

[6] Gray. *Gray's anatomy atlas*.

[7] K. Hartmann, K. Ali, and Th. Strothotte. Floating labels: Applying dynamic potential fields for label layout. In A. Krüger P. Olivier A. Butz, B. Fisher, editor, *4th International Symposium on Smart Graphics 2004, LNCS 4073*. Springer Verlag, 2004.

[8] K. Ali K. Hartmann, T. Götzelmann and Th. Strothotte. Metrics for functional and aesthetic label layouts. In *5th International Symposium on Smart Graphics, LNCS 3638*, pages 115–126. Springer Verlag, 2005.

[9] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings IEEE Visualization 2003*, 2003.

[10] Stefan Maass and Jürgen Döllner. Dynamic Annotation of Interactive Environments using Object-Integrated Billboards. In *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG06)*, pages 327–334. UNION Agency - Science Press, 2006.

[11] Felix Ritter, Henry Sonnet, Knut Hartmann, and Thomas Strothotte. Illustrative shadows: integrating 3d and 2d information displays. In *IUI '03: Proceedings of the 8th inter-*

national conference on Intelligent user interfaces, pages 166–173. ACM Press, 2003.

[12] K. Hartmann T. Götzelmann and Th. Strothotte. Agent-Based Annotation of Interactive 3D Visualizations. In A. Krüger P. Olivier A. Butz, B. Fisher, editor, *6th International Symposium on Smart Graphics 2006, LNCS 4073*, pages 24–35. Springer Verlag, 2006.

[13] K. Hartmann T. Götzelmann, K. Ali and Th. Strothotte. Adaptive labeling for illustrations. In C. Gotsman E. Wu, D. Manocha, editor, *Proceedings of Pacific Graphics*, pages 64–66, 196. The 13th Pacific Conference of Computer Graphics and Applications, 2005.

[14] K. Hartmann T. Götzelmann, K. Ali and Th. Strothotte. Form follows function: Aesthetic interactive labels. In B. Gooch W. Purgathofer L. Neumann, M. Sbert, editor, *EuroGraphics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 193–200. Eurographics Association, 2005.