

Dynamic Label Placement for Forensic Case Visualization

C. Sirk^{†1} D. Kalkofen² D. Schmalstieg² A. Bornik³

¹Graz University of Technology, Graz, Austria

²Institute for Computer Graphics and Vision (ICG), Graz University of Technology, Graz, Austria

³Ludwig Boltzmann Institute – Clinical-Forensic Imaging (CFI), Graz, Austria

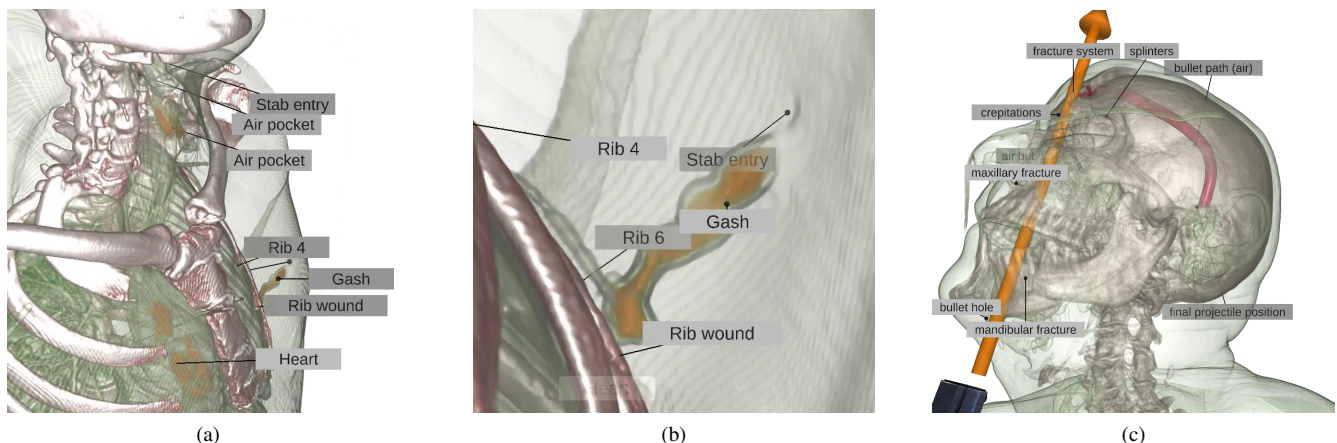


Figure 1: Label placement in forensic scenes: Annotated stabbing case visualization based on CT data. Overview frame (a) and close-up view (b) from an interactive exploration of the stab wound. (c) Case illustration showing an unusual bullet path based on CT data, and meshes for weapon, expected, and actual bullet path. Our approach is able to integrate both, polyhedral model and volume rendering.

Abstract

Forensic case analysis and in court presentation requires comprehensible illustrations and animations of findings and their relations to the course of events. Often this can only be achieved by adding textual descriptions. From a systems point of view, this requires automated label placement functionality for scenes composed of translucent polyhedral models and volumes, which we achieve through tight integration of the automated label placement algorithm and the hybrid volume/surface rendering system. Our method exploits transparency in order to place labels close to their anchors, either inside the scene, on-top or outside the occupied screen region. Inside placement makes it possible to zoom into the dataset, leads to more temporal coherency, and improves layout quality, especially for large numbers of labels. New measures for scene content importance and label occlusion prevent masking of important scene details by labels and vice versa.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Label Placement, Volume Rendering

1. Introduction

Comprehensible visualization of forensic data is a challenging task. Volumetric scans of inner injuries have to be presented to judges, juries and lawyers without the medical and anatomical background required to understand findings and their connection to the course

of events from the medical data itself. Therefore, forensic presentation aims at largely self-explanatory illustrations, depicting individual findings and explaining them and their expert interpretations using text labels. Presentation in form of videos and interactive demos further aids case understanding, but requires automated label placement.

Forensic case illustration needs combine different types of data like CT volumes and polyhedral models like 3D scans of weapons. Moreover, forensic scenes exhibit a high degree of transparency to be able to depict spatial relations between exterior and interior find-

[†] This project is financed by the KIRAS programme (no. 850183, CSIS-martScan3D) under supervision of the Austrian Research Promotion Agency (FFG).

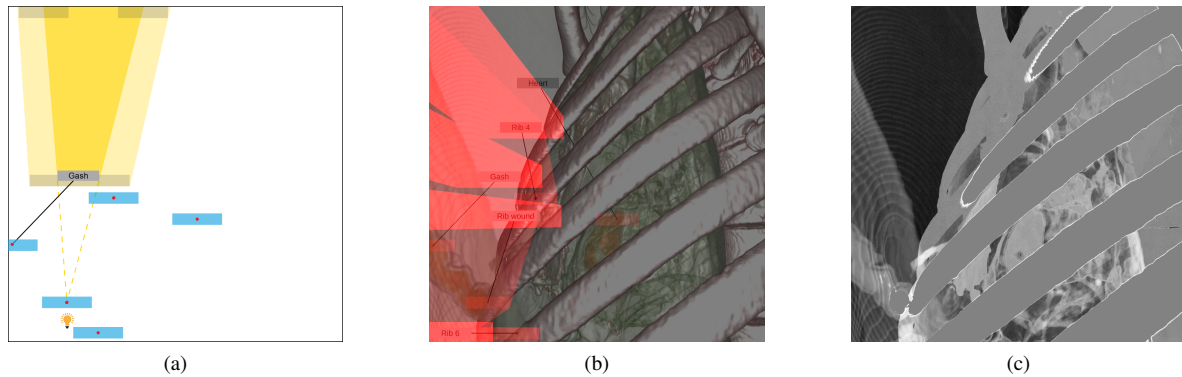


Figure 2: Illustration of constraints creation: (a) depicts shadow polygon creation for terms 2 to 4 in Equation 1. Constraints for a new label “Rib wound” with anchor indicated by a bulb are computed using shadow polygons. The yellow area spanned by the new anchor and label “Gash” is one of them. The light region denotes the dilation with the new label box needed for efficient look-up. (b) shows a complete set of such constraints regions as an overlay on the scene. Label box shadow constraints are displayed in bright red and the connector shadows in red. (c) is the integral cost image (term 5) based on saliency and occlusion for the second layer. Dark regions indicate low placement costs.

ings. Existing label placement approaches are designed with largely opaque scenes with well-defined surfaces and a viewer positions outside the data in mind. Such conditions are perfectly suited for label placement on-top of other scene content and in empty screen regions, while forensic case exploration demands spatially and temporally consistent label placement for arbitrary viewpoints inside and outside transparent translucent scenes. Furthermore automated label placement should utilize free space inside and behind semi-transparent content.

Automatic label placement has received much attention in research for decades (see *Oeltze-Jafra et al.* [OJP14] for a recent overview of techniques used in medical visualization). One popular strategy for automated label placement is to encode layout criteria as forces in an iterative force framework with the goal of convergence to a good layout [HAS04, SSB06]. Such methods are sensitive to local minima and initial conditions. Thus, they are mainly targeting still-image illustrations with labels placed in empty space around a central object of interest.

Stein and Decorèt [SD08] use an energy based formulation for outside label placement. They propose a greedy algorithm for an approximate solution of the resulting optimization problem based on a heuristic placing labels with anchors far from empty screen regions first. The algorithm yields good placement results for distant views and a small number of labels. On the downside, there are bad temporal coherency and no support for transparent scenes, and on-top/inside label placement. Furthermore, insertion order computation is expensive.

Temporally coherent placement behavior without jumping or jittering has been addressed by *Bell et al.* [BFH01] using a discrete optimization approach. Force-based approaches [PHTP*10, VTW12, TKGS14] can be extended with temporal coherence enforcement.

In our work, we use a saliency map in analogy to *Grasset et al.* [GLK*12] as an importance measure with the goal to avoid label placement on-top of important screen content.

Our work addresses limitations of the aforementioned methods

with three contributions: First, we introduce automatic label placement as part of a volume rendering system, which supports transparent scenes and automatically avoids occlusion between important scene details and labels. Second, our method utilizes space inside/behind translucent scene content, which aids temporally coherent label placement and increases the number of supported labels. Third, we introduce a new importance measure to prevent occlusion of important scene content by labels and vice versa. All together, the resulting method is to the best of our knowledge the first dynamic label placement approach for supporting arbitrary viewpoints in transparent scenes, and featuring temporally coherent outside, on-top, and inside label placement.

The algorithm was implemented as part of the forensic case analysis tool by *Urschler et al.* [UBS*12].

2. Method

An exhaustive search for suitable label positions in 3D space is infeasible. We attack this problem by reducing search space to a discrete set of planes defined by their respective anchors, where the labels are placed. This simplification effectively reduces search space to two dimensions while still preserving label depth ordering. For better scalability with large numbers of labels we limit the number planes by partitioning z-space into layers. Each label is assigned to the layer closest to its anchor and placed on its supporting plane [TKGS14]. Visually correct integration of labels with volume rendering is achieved by interrupting ray casting at each plane. The intermediate volumetric rendering results are further used as guidance images, to place the labels in the layer associated plane based on an optimization approach.

Layer selection In order to minimize depth differences between the anchors and labels we use a clustering approach in post-perspective space, leading to more layers and higher depth precision close to the viewer. Clustering is done using *k-means++* [AV07]. Unlike the original *k-means* algorithm [Mac67, Llo82], *k-means++* ensures that all clusters are filled with elements through optimized seeding. Layer planes are positioned at the me-

dian anchor depth, to prevent outliers from influencing the layer plane position.

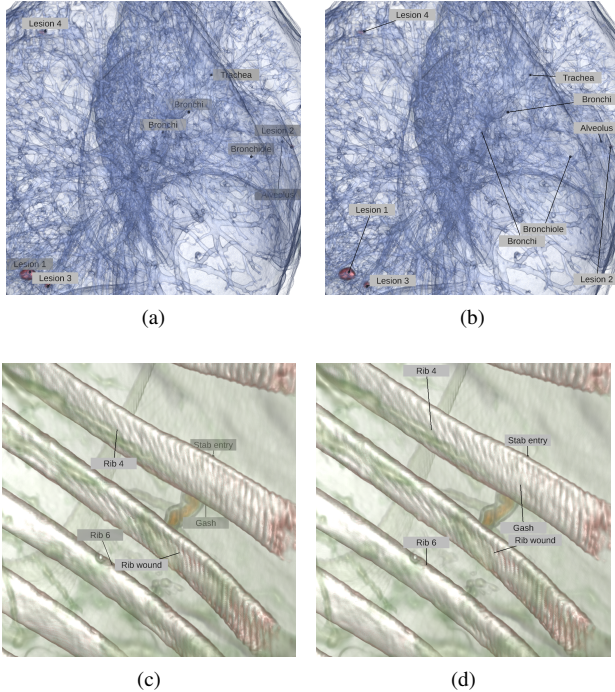


Figure 3: Comparison. Our method (a), (c) places labels closer to the anchor than in the method of Stein & Décoret [SD08] (b), (d). Some labels further back are partially occluded by details in the foreground, but still remain legible. This also adds depth cues.

Energy optimization We use an optimization approach inspired by Stein and Décoret [SD08]. In contrast, we combine the layout criteria in a single objective function rather than enforcing a separate set of hard constraints. This leads to better scaling with the number of labels and temporal coherency at the price of increasingly suboptimal solutions.

The energy is as follows:

$$E_i(x_i, y_i) = w_1 |\mathbf{a}_i - \mathbf{l}_i| + w_2 f_{lab}(x_i, y_i) + w_3 f_{conn}(x_i, y_i) + w_4 f_{anch}(x_i, y_i) + w_5 \iint_{B(\mathbf{l}_i, \mathbf{s}_i)} g(u, v) du, dv \quad (1)$$

The first term in (Equation 1) penalizes the distance between the anchor position \mathbf{a}_i and the currently evaluated position candidate $\mathbf{l}_i = [x \ y]^T$ for label i .

Term 2 avoids label placement leading to the occlusion of previously placed labels. Terms 3 and 4 do the same for connector lines, and anchors. In practice we compute cost look-up tables for them by rendering shadow polygons (see Figure 2a and 2b) spanned by existing labels/connector lines as proposed in [SD08] to an accumulating cost buffer.

The fifth term is crucial for label placement on top and inside scene content. It defines the costs for scene occlusion by labels and

occlusion of labels as an integral of the function $g(u, v)$ over the label box $B(\mathbf{l}_i, \mathbf{s}_i)$, defined by the label position \mathbf{l} and the label's size \mathbf{s} . The integral costs combine occlusion and saliency as follows:

$$g(x, y) = \sum_{i=0}^l o(i, x, y) + \left(1 - \sum_{i=0}^l o(i, x, y)\right) \cdot [\beta \cdot s(x, y) + (1 - \beta) \cdot \sum_{i=l+1}^{n-1} o(i, x, y)] \quad (2)$$

where l is the current layer index, n is the number of layers, $o(i, x, y)$ is the occlusion by layer i defined by the accumulated alpha value, and $s(x, y)$ is a saliency measure indicating regions with important content, where labels should not be placed. We use the gradient magnitude of the rendered color buffer without labels as a fast saliency estimation. The integral costs combine the occlusion up to, the saliency, and the occlusion contribution behind the current layer in a single measure. Figure 2c shows an example.

Terms 2 to 4 are computed by rendering the shadow polygons into an image-space buffers. Efficient dilation using a rectangle [Wei06, FH00] is implemented using an optimized geometry shader. It is inspired by a conservative rasterization algorithm for vertex shaders [HAMO05]. Term 5 involving saliency and occlusion is pre-computed in CUDA to a summed area table for fast evaluation using the algorithm by Nehab *et al* [NMLH11]. The energy minimization is also performed on the GPU.

Placement algorithm The placement algorithm is a *greedy* approach in analogy to [SD08] with the heuristics to process labels in front-to-back order. Each label's optimal position in its assigned layer plane is computed by minimizing (Equation 1). Placement results are used to recompute look-up tables for terms 2 to 4 for the next label to be placed, term 5 is updated before next layer.

For temporal coherency, we use forces inspired by Hartmann *et al* [HAS04]. We project the labels from their previous 3D positions to the current frame and adjust their positions using forces dragging the label towards their new ideal position determined through optimization with a magnitude inversely proportional to the remaining target distance. This avoids label hopping on significant layout changes, due to, e.g. fast camera motion.

Volume rendering Our rendering system must handle complex scenes including volume datasets, surface representations like 3D models and label polygons. To do so, we integrate multi-volume rendering based on ray-casting, rasterization, and an order-independent transparency algorithm in a single framework inspired by Kainz *et al*. [KGB*09]. This involves depth sorting of surface models and dataset boundaries during rasterization followed ray casting of ray segments within volumes, and combining the resulting color and transparency values. We avoid an explicit sorting stage using an HA-buffer approach [LHL14]. For visually correct label integration we also interrupt rendering at layer planes.

3. Results

Example results for real-world forensic scenes are given in Figure 1. For comparison we implemented the algorithm by Stein and Décoret including including their energy formulation using hard constraints inside our rendering framework. We used our saliency measure to guide on-top placement. Figure 3 depicts the differences between the two approaches for still images. Our approach is able

to place labels behind semi-transparent content, which enhances depth perception.

Figure 4 shows a stress test using 100 labels. Our layered placement algorithm computes a satisfactory label layout including all labels. Even though some labels are partially occluded by other labels, most of these labels are still readable. Moreover, labels can generally be placed closer to their anchors. We also found that our approach shows a higher degree of temporal coherence due to re-projection/relaxation of optimization results from previous frames (see video). We empirically found that a high weight penalizing a large distance to the anchor causes smoother label position changes and aids the user in navigating complex scenes, since the labels stay closer to the anchor.

Besides better placement performance, our layered labeling approach is faster to compute. Computational cost mainly depend on the number of labels. Table 1 contains performance measurements. Odd rows contain our results, even rows the timings of [SD08]. The “placement” part primarily entails computation of the cost term images, which needs to be done for each label, and minimization. “Update” refers to the force/relaxation based computation of the actual label positions, as well as updating the label affiliation. “Clustering” is the time to recalculate the clustering.

All performance measurements were conducted on a PC with an Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz with 4 cores, 16GB RAM and a NVidia GeForce GTX 1070 GPU with 8GB of RAM.

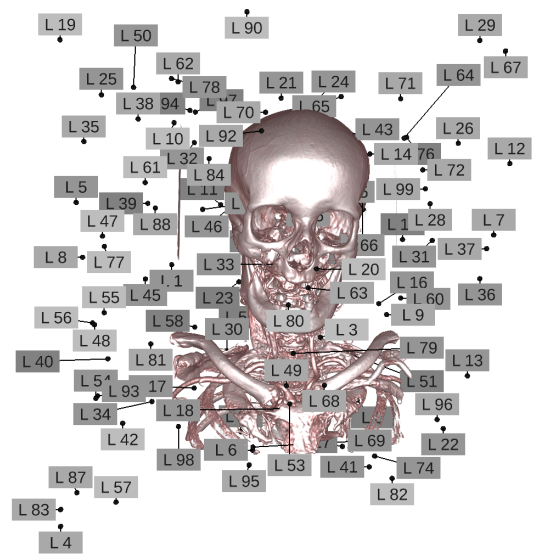
4. Conclusion and future work

We propose a novel labeling placement algorithm for volumetric scenes with transparency and high depth complexity, which supports label placement outside, on-top of and inside the scenes. By limiting label placement to discrete planes layers and using depth as natural insertion ordering, we are able to perform true 3D placement, which provides additional consistent depth cues, while keeping computational costs low.

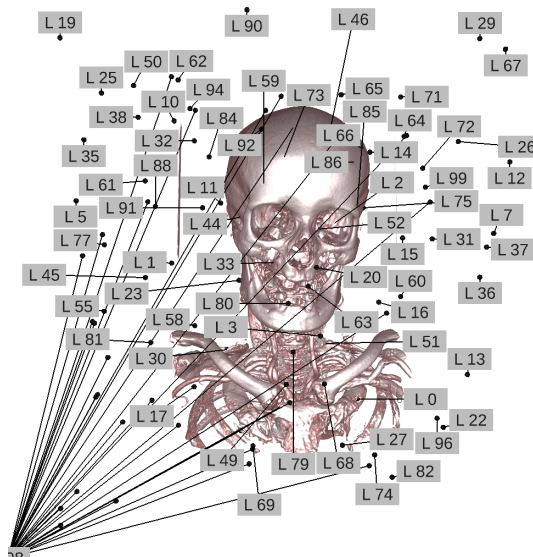
To further communicate important findings to forensic experts, we plan to add support for image labels. Improved Label movement behaviour could probably be achieved using probabilistic filters. Easing curves could be applied for a more natural feel, and also to increase the maximum label movement speed. Finally we would like to conduct user studies to validate our approach and to receive feedback for parameter tuning towards increased user satisfaction.

Table 1: Performance measurements of some test scenes. Odd rows are taken from the proposed algorithm with four layers. Even rows are derived from our Stein & Décoret implementation.

Scene	Labels	Placement	Update	Clustering
		ms	ms	ms
Knife	9	3.84	2.54	0.05
Knife [SD08]	9	11.46	2.42	0.04
Shot	9	4.87	2.28	0.06
Shot [SD08]	9	11.73	2.18	0.04
Sponza	10	2.69	2.29	0.07
Sponza [SD08]	10	8.55	2.31	0.07
Dice	22	10.75	2.54	0.10
Dice [SD08]	22	23.92	2.54	0.08
Test-Fig.5	100	70.86	7.96	0.38
Test-Fig.5 [SD08]	100	179.37	8.62	0.33



(a)



(b)

Figure 4: Stress test using 100 labels placed randomly. (a) shows our result. All labels can be successfully placed, most of them close to their anchors. Partial occlusion of label, connectors and scene content occurs, but most labels remain legible. (b) The algorithm by Stein & Décoret fails to place around 30 percent of the labels (default position: bottom left corner). Apart from that several labels are unfavorably placed far away from their anchors.

References

- [AV07] ARTHUR D., VASSILVITSKII S.: k-means++: The Advantages of Careful Seeding. In *Proc. 18th ACM-SIAM Symp. Discret. algorithms* (2007), vol. 8, pp. 1027–1035. [2](#)
- [BFH01] BELL B., FEINER S., HÖLLERER T.: View management for virtual and augmented reality. *Proc. 14th Annu. ACM Symp. User interface Softw. Technol. - UIST '01 2001* (2001), 101. [2](#)
- [FH00] FLATO E., HALPERIN D.: Robust and Efficient Construction of Planar Minkowski Sums. In *Abstr. 16th Eur. Work. Comput. Geom.* (2000), pp. 85–88. [3](#)
- [GLK*12] GRASSET R., LANGLOTZ T., KALKOFEN D., TATZGERN M., SCHMALSTIEG D.: Image-driven view management for augmented reality browsers. In *IEEE ISMAR* (2012), pp. 177–186. [2](#)
- [HAM05] HASSELGREN J., AKENINE-MÖLLER T., OHLSSON L.: Conservative rasterization. *GPU Gems 2* (2005), 677–690. [3](#)
- [HAS04] HARTMANN K., ALI K., STROTHOTTE T.: Floating labels: Applying dynamic potential fields for label layout. *Smart Graph.* (2004), 101–113. [2, 3](#)
- [KGB*09] KAINZ B., GRABNER M., BORNIK A., HAUSWIESNER S., MUEHL J., SCHMALSTIEG D.: Ray casting of multiple volumetric datasets with polyhedral boundaries on manycore GPUs. *ACM Trans. Graph.* 28, 5 (2009), 1. [3](#)
- [LHL14] LEFEBVRE S., HORNUS S., LASRAM A.: Per-Pixel Lists for Single Pass A-Buffer. *GPU Pro 5 Adv. Render. Tech.* (2014). [3](#)
- [Llo82] LLOYD S. P.: Least Squares Quantization in PCM. *IEEE Trans. Inf. Theory* 28, 2 (1982), 129–137. [2](#)
- [Mac67] MACQUEEN J. B.: Kmeans Some Methods for classification and Analysis of Multivariate Observations. *5th Berkeley Symp. Math. Stat. Probab.* 1967 1, 233 (1967), 281–297. [2](#)
- [NMLH11] NEHAB D., MAXIMO A., LIMA R. S., HOPPE H.: GPU-efficient recursive filtering and summed-area tables. *ACM Trans. Graph.* 30, 6 (2011), 1. [3](#)
- [OJP14] OELTZE-JAFRA S., PREIM B.: Survey of Labeling Techniques in Medical Visualizations. In *Proc. 4th EG Work. Vis. Comput. Biol. Med.* (Vienna, Austria, 2014), Eurographics Association, pp. 199—208. [2](#)
- [PHTP*10] PICK S., HENTSCHEL B., TEDJO-PALCZYNSKI I., WOLTER M., KUHLER T.: Automated Positioning of Annotations in Immersive Virtual Environments. *Proc. EGVE-JVRC* (2010), 1–8. [2](#)
- [SD08] STEIN T., DÉCORET X.: Dynamic Label Placement for Improved Interactive Exploration. In *Proc. Sixth Int. Symp. Non-Photorealistic Animat. Render. (NPAR 2008, June 9–11, 2008, Annecy, Fr.* (2008), pp. 15–21. [2, 3, 4](#)
- [SSB06] STADLER G., STEINER T., BEIGLBOCK J.: A Practical Map Labeling Algorithm Utilizing Morphological Image Processing and Force-directed Methods, 2006. [2](#)
- [TKGS14] TATZGERN M., KALKOFEN D., GRASSET R., SCHMALSTIEG D.: Hedgehog labeling: View management techniques for external labels in 3D space. *Proc. - IEEE Virtual Real.* (2014), 27–32. [2](#)
- [UBS*12] URSCHLER M., BORNIK A., SCHEURER E., YEN K., BISCHOF H., SCHMALSTIEG D.: Forensic-case analysis: from 3D imaging to interactive visualization. *IEEE CG&A* 32, 4 (2012), 79–87. [2](#)
- [VTW12] VAARANIEMI M., TREIB M., WESTERMANN R.: Temporally coherent real-time labeling of dynamic scenes. In *Proc. 3rd Int. Conf. Comput. Geospatial Res. Appl. - COM.Geo '12* (2012), p. 1. [2](#)
- [Wei06] WEIN R.: Exact and efficient construction of planar Minkowski sums using the convolution method. *Algorithms-ESA 2006 006413* (2006), 1–12. [3](#)