

Seamless Integration of Labels into Interactive Virtual 3D Environments Using Parameterized Hulls

S. Maass¹ and J. Döllner¹

¹University of Potsdam, Hasso-Platter-Institute, Germany

Abstract

This paper presents an approach for the automated, dynamic placement of labels attached to objects of 3D scenes. These labels are seamlessly integrated into the 3D scene by textured polygons aligned to parameterized hulls, which generalize an object's geometry. This way, the labels follow the principle shape of the annotated objects, emphasize the label-object relationship, reduce ambiguities of interpretation, and achieve visual concise and aesthetic results. The algorithm first sets up candidate positions across the hull. Cascaded filtering and presorting steps reject early improper placement candidates and order the remaining ones according to their quality. Then, they are iteratively tested to find positions that allow for a visible and legible label placement. If more than one position exists, the selection is controlled by a layout strategy, for which we present efficient evaluation techniques and discuss the impact on the aesthetic appearance. As proof of concept, we have implemented a 3D viewer that annotates building models of 3D city models. Compared to view plane-based labeling approaches, our approach achieves a strong visual relation between label and annotated object, treats labels as first-class objects of virtual 3D environments, and offers a high degree of customization and stylization.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, Virtual Reality

1. Introduction

Labels are essential graphical elements that allow us to add textual or symbolic information to graphical objects. Labels can be used, for example, to identify objects, to show object-related meta information, to add values or measurements, and to provide descriptions. We refer to this kind of label application as *annotation*, and to the object to which an annotation is associated as *annotated object*.

Label placement and layout represent a computer graphics challenge because there is no predefined way of embedding text into images. In particular, the integration of text into interactive views of perspective 3D scenes is still an open field of research. For example, there are only few label placement and layout techniques that address the automatic and dynamic integration of labels for virtual 3D environments. Two major challenges exist for these techniques:

- **Placement and Layout:** Placement and layout computation needs to be fast enough such that the user can continu-

ously interact with the virtual 3D environment. In contrast to static media (e.g., cartographic maps), label layout and placement do not need to be optimal for all defined labels because most likely only a subset of them is visible and, therefore, in the focus of the user.

- **Visual Integration into the 3D Space:** Labels should be tightly integrated into the 3D space of the virtual 3D environment to provide a direct and close relation to the annotated objects. If labels are represented as superimposed elements in the view plane, label perception and cognition becomes more difficult because the 2D view plane space and the 3D scene space do not naturally connect, and auxiliary elements such as pins are required to link labels and annotated objects. In particular, if a strong sense of immersion should be guaranteed, labels should be represented as true part of the 3D scene.

The presented labeling technique addresses both issues: It determines label placement and layout in real time, and it embeds labels as true 3D elements into 3D scenes. Instead

of using a 2D overlay of the view plane to embed labels, it represents labels by textured polygons assigned to planar or curved surfaces that surround the annotated object as a kind of invisible hull, i.e., as a bounding surface that generalizes the original geometry of the annotated object.

To achieve visual pleasing results, our labeling technique intends to ensure that labels are visible, legible, and unambiguously associated with the annotated object. To reduce computational costs, the algorithm searches for the best label position by grouping, ordering, and iterating through these candidates. Furthermore, different placement styles can be used to make a decision if more than one candidate with a similar visibility exist.

2. Related Work

2.1. 2D Map Labeling

Automated label placement is a well studied for 2D cartographic maps, which demand for techniques to label point, line, and area features (e.g., cities, streets, and lakes). An optimal placement position for individual labels can cause overlay conflicts with other labels or annotated objects. Therefore, visibility has to be optimized globally, which is an NP-complete problem as shown by Marks and Schieber [MS91]. A comparative overview of point label optimization strategies can be found in [CMS95]. Another approach, which tries to resolve the conflicts by relaxing attraction and repulsion between labels and features, is presented by Ebner et al. [EKW03]. Petzold et al. [PGP03] present a data structure for detecting possible conflicts at different map scales, permitting fast 2D map labeling under interactive conditions.

2.2. Labels in Interactive 3D Environments

For labeling 3D objects, we can distinguish between techniques to label 3D illustrations and 3D virtual environments. In 3D illustrations, e.g., labeling of medical, botanical, or technical details, the object of interest is mostly centered in the view, and user interaction is dominated by rotation and zoom operations. This setting naturally provides white space around the object that can additionally be used to place labels.

In contrast, 3D virtual environments provide a more immersive experience to the user, e.g., when acting as a virtual pedestrian in a virtual 3D city model. Scene objects can take large areas of the view, what motivates our approach to embed labels as textured surfaces surrounding the annotated object to strengthen the spatial correlation between label and object. This becomes even more important in augmented reality applications.

2.2.1. Labeling 3D Illustrations

A first labeling technique for 3D illustrations is presented by Preim et al. [PRS97]. Explicit regions are defined around the

object on the view as placement containers, whereby annotations placed inside are linked with an explicit line to the corresponding feature. Ritter et al. [RSHS03] propose a technique that uses the projected shadow of the illustrated object as reference area for labels.

The idea to use potential fields to optimize label positions is adapted to 3D illustrations by Hartmann et al. [HAS04]. Thereby, forces between labels, their references, and the screen boundary are defined. Then initial positions are optimized with an iterative relaxation process. An agent based optimization approach to label 3D illustration is presented by Götzelmann et al. [GHS06a]. Furthermore, they extend labeling techniques to handle object groups [GHS06b] or to annotate animated object parts [GHS07]. Ropinski et al. [RPRH07] present a technique allowing an interactive embedding of 3D labels into medical illustrations.

2.2.2. Labeling 3D Virtual Environments

A first approach used to label 3D building models is presented by Bell et al. [BFH01]. Occupied regions on the screen are marked by rectangles using an efficient view management data structure. This structure is initialized with the screen-space bounding boxes of important scene elements. During an iterative placement it supports fast query and marking operations to test label-label and label-scene occlusion. However, labels are placed as 2D overlays and not embedded into the scene.

Another view-plane technique, optimized for the labeling of point features on terrain-based 3D virtual environments, is presented in [MD06b]. There, the view plane is divided vertically into slots, which are used to mark occupied height intervals. Then, labels are iteratively placed from front to back, whereby an additional z-buffer test is used to avoid occlusion from closer scene elements.

The approach in [MD06a] integrates camera-oriented billboards to label a campus model. As in this contribution, a hull approximating the original geometry, is used for the position selection, but in combination with a building skeleton. With a ray test between the camera view vector and each building skeleton a point on this hull is determined to fix the billboard; occlusion with other annotations or scene elements is not explicitly tested.

In [MD07], embedded labels are placed for street names of an interactive 3D city model. Discrete candidate positions are distributed along the line features and additionally used for visibility calculations, an idea we adapted to surfaces in our technique. Additionally, possible overlays between labels are pre-calculated and stored in a conflict graph that is used to speed-up the placement process.

Another example of object-embedded labels is given by Petrovic et al. [PFK07]. Implicit given texture coordinates are used in GPU shader programs to label tuboids that visualize structures in the human brain. However, instead of ex-

Explicitly selecting a visible position, labels are repeated along tuboids with a constant distance.

3. Parameterized Hulls

Hulls, representing the generalized object's geometry, are used to embed labels into the 3D scene; the labels are represented by textured polygons aligned to the hull. To be suitable for label placement, a hull typically should meet the following requirements:

- **Coverage:** The hull has to cover all areas of the annotated object that are adequate to carry a label. Thereby the shape of the hull should preserve the shape of the annotated object and small details should be smoothed. Because the glyphs of the label are positioned across the hull, no part of the hull should intersect the annotated object. The distance between hull and annotated object should be kept minimal to avoid ambiguities with other scene objects.
- **Parameterization:** The hull should be parameterized, i.e., its surface should be unambiguously defined by a two-dimensional coordinate system. The parameterization should support the definition of optimal positions, e.g., center points of continuous surface patches, and the fast search for good placement alternatives, if these positions are occluded.

For our application, we have implemented three different hull types, rectangle set hull, cylinder hull and footprint hull. These hull types focus on the typical shapes of 3D building models in 3D city models, but appear to be generally usable for objects of virtual 3D worlds.

3.1. Rectangle Set Hull

To build a generalized hull around an annotated object we can create a set of rectangles that surround the object by generating one rectangle per facade and roof. Fig. 1 shows the hull for two buildings with horizontally and vertically embedded labels. Each rectangle is parameterized by two perpendicular edges. This allows us to distribute candidate sample points with equal distances along the edges of the rectangles. To balance precision and performance, we choose a sample point distance that results in approximately one sample per letter for a fixed text size.

3.2. Cylinder Hull

As next hull type, we support cylindrical hulls (Fig. 2), whose axis points into the up direction. On the cylinder's lateral surface, labels can be integrated horizontally or vertically. For top views, a horizontal label can be bended along the silhouette. To distribute candidate sample points, the cylinder is parameterized by rotation angle and height.

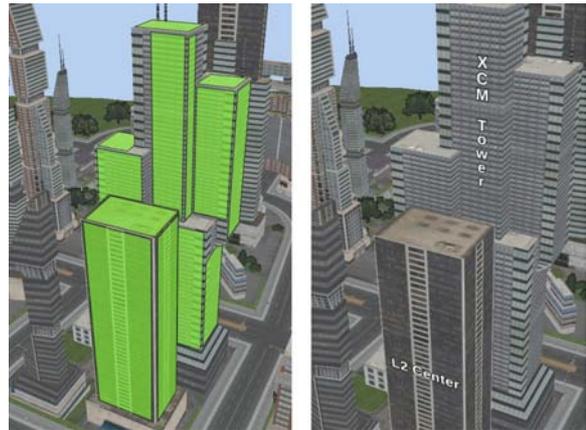


Figure 1: Visualized rectangle set hull for two building models (left) and their use for a horizontal (preferred) and a vertical label integration (right).



Figure 2: A tower building with a cylinder hull used for integrating labels on the lateral surface or around the top face (from left to right).

3.3. Footprint Hull

For annotated objects with a complex structure, a footprint hull (Fig. 3) can be used, which is constructed by extruding the (possibly simplified) object's footprint to a constant height. Neighboring segments of the footprint with a low orientation deviation are merged to continuous patches. These patches are parameterized along the footprint baseline and the extrusion height. As for the rectangle set, candidate position points are equally distributed over these patches. To achieve a smooth contour, face normals are interpolated smoothly in a small interval at inner segment borders. Additionally, one or more curves can be defined on the top surface, carrying label position candidates for a view from above.

4. Workflow

Fig. 4 shows the steps accomplished by our technique and how they fit into the classical model of the visualization pipeline. First, the algorithm filters all hulls that are too far away, not visible, or have a projected bounding box that is

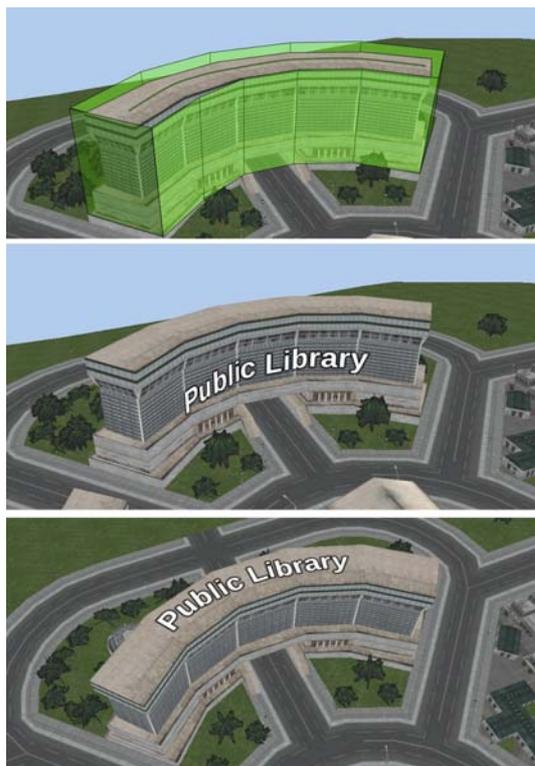


Figure 3: A footprint hull composed of an elevated ground plane and a top face skeleton curve (top), used for label integration at the facades (middle) or the top face (bottom).

too small to integrate a label. For a fast determination of the relevant subset, these tests are ordered by their computational complexity and are applied first for whole hulls and after that for the hull parts, e.g., single planes of the rectangle set hull. In addition, we test orientation and visibility of hull parts to eliminate back-faced or occluded sample point sets. For the visibility test, the depth of hull sample points is compared with the corresponding value of a linearized depth buffer, generated by a scene rendering without labels.

Second, for each object in the remaining set, the mapping step selects an embedding area and a particular label position on it. This includes the evaluation of the orientation, embedding area size, and label visibility for the candidates.

Third, scene graph representations for the calculated label placements are constructed and the added elements are rendered to combine scene elements and labels into one depiction.

5. Calculating Label Positions

For a better understanding this part is discussed in detail for the rectangle set hull first. Extensions specific to the cylindrical and footprint hull are described afterwards.

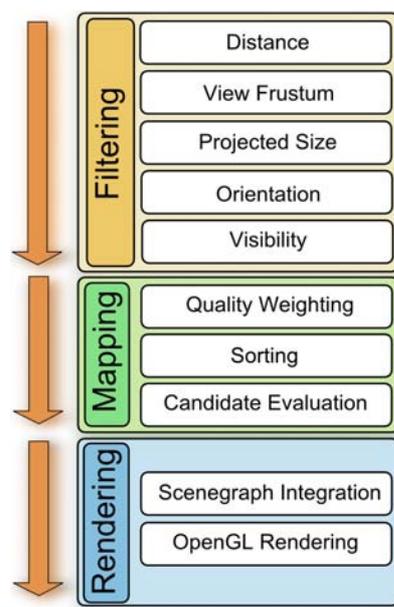


Figure 4: Overview of the single steps of our automated labeling technique.

All elements of a rectangle set hull are handled in the same way by our technique, except that text on roof rectangles is never integrated vertically. The sample points, equally distributed across the hull, are used for two purposes: for visibility approximation and as label position candidates. To determine the candidate position that gives best quality, testing all positions at all rectangles that were not rejected by the filtering would be too complex to be applied in real time. Therefore, all remaining rectangles are sorted and tested regarding to their quality potential. Moreover, the sample points for a chosen rectangle are evaluated in an order related to the placement style, enabling us to test candidate positions close to the optimum first.

5.1. Ordering Position Candidate Groups by their Quality Potential

Looking at perspective depictions containing hand-placed labels, it can be observed mostly large surface parts and areas that are oriented towards the viewer are used as labeling area. In the same way, the potential of each rectangle to carry the best candidate point that allows for a visible, readable, and aesthetic text integration, depends on its orientation and relative screen size. Instead of a weighted linear combination, our technique uses a discrete scoring system for orientation and size values that includes four quality categories (perfect, good, acceptable, poor). This turned out to be more intuitive to configure and considers that the combination of both attributes is the most important fact.

The first score assigns to all remaining rectangles i an orientation score ($score_O$), which is calculated from the angle α_i between normal and direction from the center of the rectangle to the camera position. The discrete orientation score is defined by

$$score_O(i) := \begin{cases} 3 & \text{for } 0^\circ \leq \alpha_i \leq 15^\circ, \\ 2 & \text{for } 15^\circ < \alpha_i \leq 30^\circ, \\ 1 & \text{for } 30^\circ < \alpha_i \leq 50^\circ, \\ 0 & \text{else} \end{cases}$$

If a hull contains rectangles with large difference in size, they should be divided into similar sized tiles for a more balanced version of this measure. In this case $score_O$ is calculated using the minimum angle of all tiles.

The second score examines the rectangle sizes on the view port. Large surface parts should be favored for the label integration over small ones, because this strengthens the communicated relation between label and annotated object and increases the probability to find a position providing a fully visible label. For the calculation of the projected area, the rectangle vertices are projected onto the 2D view port. To determine the *visible* projected area, this value is multiplied by the relative visibility of the rectangle, which can easily be approximated using the ratio of visible to all candidate sample points. As done for the orientation, we define the score to separate candidate sets with a perfect, good, acceptable, and poor ratio, compared to the maximum visible projected area A^{max} .

$$score_A(i) := \begin{cases} 3 & \text{for } A_i \geq 0.9 \cdot A^{max}, \\ 2 & \text{for } 0.9 \cdot A^{max} > A_i \geq 0.75 \cdot A^{max}, \\ 1 & \text{for } 0.75 \cdot A^{max} > A_i \geq 0.5 \cdot A^{max}, \\ 0 & \text{else} \end{cases}$$

The remaining rectangles are sorted by the sum of both scores. Elements with the same sum form a group, in which they are additionally ordered by their orientation angle α_i . During the determination of the best label position the rectangles in this structure are evaluated group by group. For all rectangles in the group with the highest sum, embedding a horizontal text is tested first. If this is not possible, e.g., because the rectangle is partially occluded or too small, the same group will be iteratively tested for a vertical text integration. If even this fails, these tests are repeated with the next group of equally scored rectangles.

Cylinder and Footprint Hull Extensions

For the cylinder hull only two candidate groups exist. The first one is formed by all sample points on the lateral surface oriented towards the viewer. The second one is the upper row of sample points around the cylinder, which is used to integrate labels if viewed from the top. Which one should be investigated first, is decided by the discrete orientation scores of the top and lateral faces. The same schema is used

for the footprint hull to select the lateral surface or the top path as first candidate group.

The side surface of the footprint hull is divided into continuous segments with a low deviation of the surface normal. As for the rectangles, a back-facing test is done for each part of these segments. This step possibly reduces the segment width or splits the segment into several parts. Each continuous segment part is then treated in the same way as the rectangles before, whereby the average normal is used for orientation related measurements.

5.2. Candidate Selection

To enable a fast placement of visible, legible, unambiguous, and visually pleasing labels, it should be avoided to calculate all quality measurements for all possible candidates. As first step, hull elements were ordered by their quality potential. Now, the elements in a group with the same score sum are iterated and the candidates are tested for occlusions until a placement position is found.

To speed up the calculation of the degree to which a label is occluded if it would be positioned at a particular sample point, a direction-separated distant field is constructed. Thereby, for each sample point the number of continuously visible neighbor points is stored for the left, right, up, and down direction (Fig. 5). If the label is centered on the sample point, the label extents define the minimum number of free samples needed in each direction for a full visible embedding.

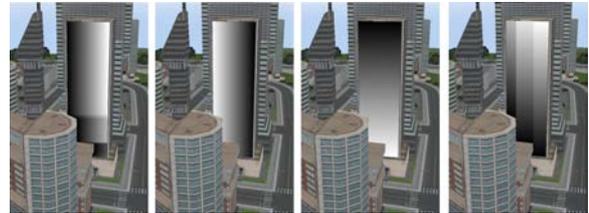


Figure 5: Illustration of the direction-separated distance field. The intensity denotes the number of continuously visible samples left, right, above, and below the current point.

If more than one sample point allows for embedding with a similar visibility, the selection can be controlled by different styles. We adapt the styles proposed in [MD07] for the labeling of line features to areas.

The first style favors positions *close to the center of the hull*. Instead of testing the label visibility for all samples, calculating the distance to the center, and choosing the best candidate afterwards, we decide to order the iteration regarding to the style. Starting at the center point, all samples are visited with a spiral pattern (Fig. 6 left). This permits fast iteration exit if a position with a fully visible embedding is found.

The second style tries to integrate labels *as close as possible to the observer*. The closest vertex or edge of the current hull is determined in a first step. If the observer is close to a vertex, the iteration is done using the spiral pattern, centered on the vertex. Otherwise, the oscillation pattern, illustrated in Fig. 6 (right), is used. This pattern starts at the midpoint sample near the found edge (in the example at the left border) and continues first alternating around this midpoint and then with an increasing distance parallel to the edge.

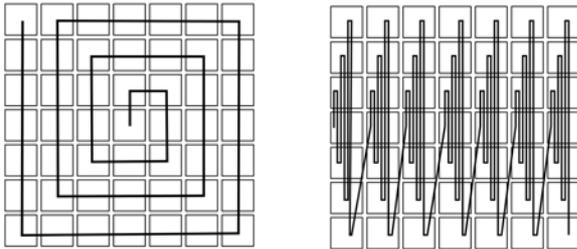


Figure 6: *Spiral (left) and oscillation search pattern (right).*

The third style *centers the label in the largest visible part* of the hull. Because an exact search for the largest visible part is computational expensive, we developed a heuristic that uses the direction-separated distance field and processes all sample points only once. During this iteration, for each sample point five values are calculated and used for a comparison: the absolute difference between the left and the right ($diff_s$) and between the up and down value ($diff_t$), the sum of the left and right (sum_s) and the up and down value (sum_t), and the sum of the differences ($diff_{st} := diff_s + diff_t$). The values for the current sample point are compared against the values of the best position found before in the following order: $diff_{st}$, $diff_s$, $diff_t$, sum_s , and sum_t . For the first three terms a lower value lets the current sample point take over the position of the best candidate, for the last two a higher value. Only if the current and the best value are equal, the next term is evaluated. Otherwise, the decision is definite. To smooth out jumping artifacts, caused by the differences at odd and even positions, we added an epsilon of one sample point to these tests.

For horizontal text at curved surfaces, we experimented with a fourth style, which favors embeddings centered on the column of sample points whose normal is pointing closest to the opposite viewing direction. This could increase the readability because the orientation of the single letters varies around the optimum. For this style a modified version of the oscillation pattern is used that starts at the center point of the column with the best normal, tests all candidates in this column with an increasing distance to this center first, and continues alternating with the parallel column left and right.

6. Label Size and Level-of-Detail Concepts

The label extents can be defined with a fixed size in the 3D scene or continuously adjusted so that they appear with a constant size on the 2D viewport. In the case of a fixed 3D scene size, the labels scale uniformly with the related object and the whole environment during the zooming operations of the user. This behavior strengthens the label-object relationship and results in a perfect integration of the labels into the virtual 3D environment. The second variant, keeping the labels at a constant view plane size, has the advantages that the readability is increased for distant labels and the occlusion of annotated objects by the associated labels is minimized for objects close the observer. However, this impairs the visual impression of the label integration during scene navigation and adds the effects that labels can reach extents that are too large for embedding or too small to communicate their object relation ship to the user (e.g., if bent around a cylinder).

One solution to deal with these effects is to combine the placement technique with a level-of-detail (LOD) concept. First, different representations for one label can be used, for example a complete name, an abbreviation, and a small symbol. This enables the technique to choose one item out of this set, depending on the available embedding area. Second, the annotated objects can be used to form a generalization hierarchy. For example, with a decreasing distance of a user to the scene first the district, then the building itself, and after that the elements of the facade can be labeled. For this, separate hulls need to be constructed for each generalization level. This could be supported by automated generalization techniques, e.g., as described in [GD07], where 3D building groups are merged to a simplified geometry based on the underlying infrastructure network.

7. Dynamic Aspects

Placing Labels for each frame can cause the effect that labels bounce between distant positions, because small changes of the user's view can result in large changes of the visibility of the annotated object. To ease this situation, the placement can be decoupled from the interaction. Thereby, labels keep their current positions during the interaction with the 3D scene and smoothly blend in at their new positions once the user finished navigating.

8. Results

As proof of concept, we applied our labeling technique to an interactive virtual 3D city model (Fig. 7). Hulls for 38 buildings have been manually constructed. Our test system was equipped with a Core2Duo processor (2.4 GHz), 2 GB of system memory, and a GeForce 8600 GT M graphics board with 512 MB video memory. The performance of our labeling technique is influenced by a number of parameters, such as the number of currently visible hulls, the number of their

elements, or the used sample point density, what makes it difficult to give exact performance measurements. However, for our city model the frame rate does not fall below 30 fps, if labels are positioned for each frame. Despite the fact, that our implementation is straight forward and leaves enough room for improvements (e.g., use of multi core functionality or an optimized scene graph) this would be fast enough if the label placement is decoupled from scene interaction.

Regarding to the visual appearance, the best results were achieved using the style that places position labels centered in the largest visible area. Only if horizontal text is placed onto a lateral cylinder surface, centering the text along the optimal normal column gives constantly a more appealing and readable result. The style that places the labels as close as possible to observer can slightly improve the readability at flat surfaces because of the higher label extent on the screen. However, aesthetically there is no advantage over labels centered in the largest areas, but this style can seriously decrease the readability at curved surfaces. At least, the style placing labels as close as possible to the facade center point minimizes the positional changes during the interaction, resulting in an steadied presentation.

9. Conclusions and Outlook

The presented labeling technique for virtual 3D environments is based on an object-space data structure, which supports placement evaluation and determination. Parameterized hulls allow us to define locations for high-quality placement positions (e.g., in the center of a plane, cylinder, or footprint patch) and to find alternatives if these locations are occluded by objects closer to the observer. Even if an annotated object is only partially visible, the integrated labels follow the original shape of the object. This effect strengthens the relation between label and annotated object and cannot be achieved in the same quality with labeling techniques based only on screen-space information such as ID and normal buffers.

Compared to screen-space labeling techniques, hulls have to be generated for the annotated objects. To build the hulls automatically, existing generalization [Kad05], [GD07] or shape fitting algorithms can be applied. Skeleton techniques, as used for automated roof generation, can be adapted to construct the curves for top faces of footprint buildings. Additional hull types can extend the technique, e.g., spheres, ellipsoids, or cones for specific application domains, e.g., molecule visualization.

Our current implementation constructs the data structures for all annotated objects (e.g., sampling points for hulls) in a preprocessing step at startup. Depending on the scene complexity, this can be memory consuming. For the future, we plan to store only the analytic description of the hulls and to generate all other information on the fly, with adapted precision and only for the current visible subset of annotated objects.

The intervals for the discrete orientation and area scores were empirically determined and work well for our application. However, these values should be evaluated with a higher number of users, models, and in other application domains.

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) as part of the InnoProfile research group '3D Geoinformation' (www.3dgi.de).

References

- [BFH01] BELL B., FEINER S., HÖLLERER T.: View management for virtual and augmented reality. In *Proc. of the 14th ACM Symposium on User Interface Software and Technology (UIST)* (Orlando, Florida, 2001), ACM Press, pp. 101–110.
- [CMS95] CHRISTENSEN J., MARKS J., SHIEBER S.: An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics* 14, 3 (1995), 203–232.
- [EKW03] EBNER D., KLAU G. W., WEISKIRCHER R.: *Force-Based Label Number Maximization*. Tech. Rep. TR-186-1-03-02, Institute of Computer Graphics and Algorithms, Vienna University of Technology, June 2003.
- [GD07] GLANDER T., DÖLLNER J.: Cell-based generalization of 3d building groups with outlier management. In *ACM 15th International Symposium on Advances in Geographic Information Systems (ACM GIS)* (November 2007), ACM Press, pp. 1–4.
- [GHS06a] GÖTZELMANN T., HARTMANN K., STROTHOTTE T.: Agent-based annotation of interactive 3d visualizations. In *Proc. of the 6th Int. Symposium on Smart Graphics* (Vancouver, Canada, July 2006), Butz A., Fischer B., Krüger A., Oliver P., (Eds.), Lecture Notes in Computer Science 4073, Springer Verlag, pp. 24–35.
- [GHS06b] GÖTZELMANN T., HARTMANN K., STROTHOTTE T.: Contextual grouping of labels. In *Proc. of Simulation und Visualization* (Magdeburg, Germany, Mar. 2006), Schulze T., Horton G., Preim B., Schlechtweg S., (Eds.), SCS Publishing House e.V., pp. 245–258.
- [GHS07] GÖTZELMANN T., HARTMANN K., STROTHOTTE T.: Annotation of animated 3d objects. In *Proc. of Simulation und Visualization* (Magdeburg, Germany, Mar. 2007), Schulze T., Preim B., Schumann H., (Eds.), SCS Publishing House e.V., pp. 209–222.
- [HAS04] HARTMANN K., ALI K., STROTHOTTE T.: Floating labels: Applying dynamic potential fields for label layout. In *4th International Symposium on Smart Graphics* (May 2004), Butz A., Krüger A., Olivier P.,

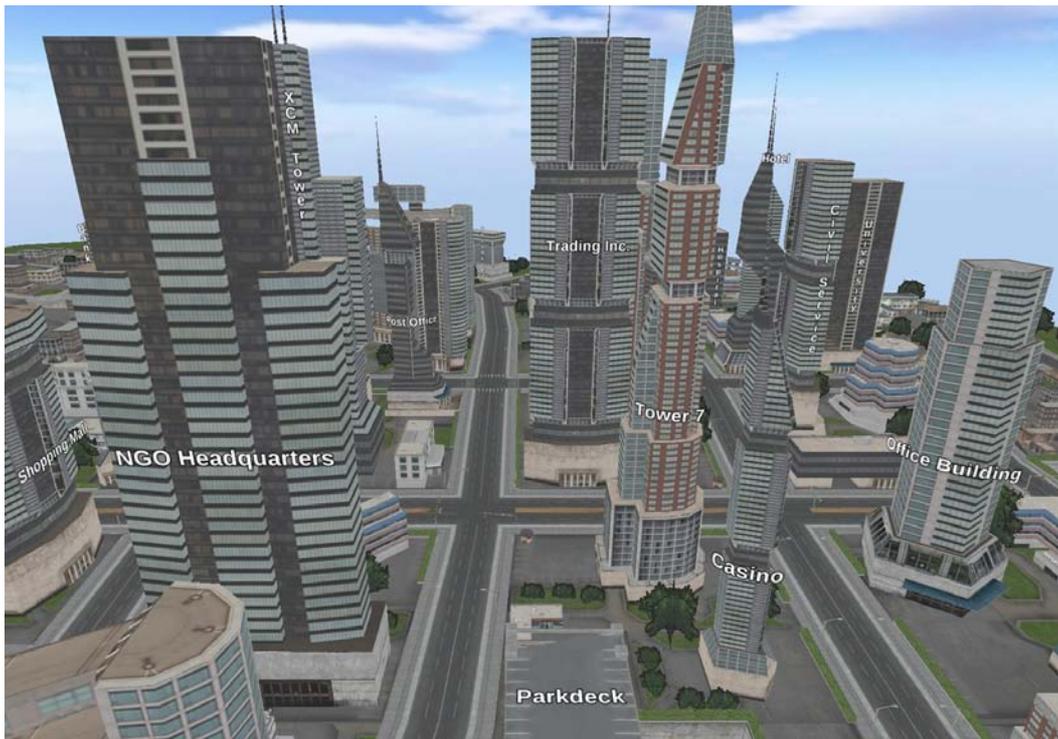


Figure 7: Our labeling technique applied to an interactive virtual 3D city model.

- (Eds.), vol. 3031 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 101–113.
- [Kad05] KADA M.: 3d building generalisation. In *22nd International Cartographic Conference* (La Coruña, Spain, 2005).
- [MD06a] MAASS S., DÖLLNER J.: Dynamic annotation of interactive environments using object-integrated billboards. In *Proc. of the 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'2006)* (Plzen, Czech Republic, Jan/Feb 2006), Jorge J., Skala V., (Eds.), pp. 327–334.
- [MD06b] MAASS S., DÖLLNER J.: Efficient view management for dynamic annotation placement in virtual landscapes. In *6th Int. Symposium on Smart Graphics* (Vancouver, Canada, July 2006), Butz A., Fischer B., Krüger A., Oliver P., (Eds.), *Lecture Notes in Computer Science* 4073, Springer Verlag, pp. 1–12.
- [MD07] MAASS S., DÖLLNER J.: Embedded labels for line features in interactive 3d virtual environments. In *Proc. of the 5th International Conference on Computer Graphics, Virtual Reality, Visualization and Interaction in Africa (ACM AFRIGRAPH 2007)* (Grahamstown, South Africa, October 2007), ACM, pp. 53–59.
- [MS91] MARKS J., SHIEBER S.: *The Computational Complexity of Cartographic Label Placement*. Tech. Rep. TR-05-91, Harvard University, March 1991.
- [PFK07] PETROVIC V., FALLON J., KUESTER F.: Visualizing whole-brain dti tractography with gpu-based tuboids and lod management. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1488–1495.
- [PGP03] PETZOLD I., GRÖGER G., PLÜMER L.: Fast screen map labeling - data-structures and algorithms. In *Proc. 23rd International Cartographic Conference (ICC'03)* (Durban, South Africa, 2003), pp. 288–298.
- [PRS97] PREIM B., RAAB A., STROTHOTTE T.: Coherent zooming of illustrations with 3d-graphics and text. In *Proc. of the conference on Graphics interface '97* (Toronto, Ont., Canada, 1997), Canadian Information Processing Society, pp. 105–113.
- [RPRH07] ROPINSKI T., PRASSNI J.-S., ROTERS J., HINRICHS K. H.: Internal labels as shape cues for medical illustration. In *Proc. of the 12th International Fall Workshop on Vision, Modeling, and Visualization (VMV07)* (nov 2007), pp. 203–212.
- [RSHS03] RITTER F., SONNET H., HARTMANN K., STROTHOTTE T.: Illustrative shadows: Integrating 3D and 2D information displays. In *Proc. of the 8th international conference on Intelligent user interfaces* (2003), ACM Press, pp. 166–173.