Dynamic Annotation of Interactive Environments using Object-Integrated Billboards

Stefan Maass
University of Potsdam
Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Strasse 2-3
14482 Potsdam, Germany
maass@hpi.uni-potsdam.de

Jürgen Döllner
University of Potsdam
Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Strasse 2-3
14482 Potsdam, Germany
doellner@hpi.uni-potsdam.de

ABSTRACT

We present a technique for the dynamic annotation of three-dimensional objects in interactive virtual environments. Annotations represent textual or symbolic descriptions providing explanatory or thematic information associated with scene objects. In contrast to techniques that treat annotations as two-dimensional view-plane elements, our technique models annotations as separate three-dimensional scene elements that are automatically positioned and oriented according to the shape of the referenced object. The shape of such an object is generalized by an annotation hull and skeleton used to determine an adequate position and orientation of the annotation with respect to the viewing direction. During camera movements, annotations float along the surface of the annotation hull. Additional constraints for the generalizations provide further control about geometric and dynamical properties. In a case study, we show how this technique can be applied for annotating buildings and other components of virtual 3D city models.

Keywords

Annotation, labeling, real-time rendering, visualization, user interfaces.

1. INTRODUCTION

Annotation techniques differentiate between *internal annotations*, overlaying the referenced objects, and *external annotations* that are placed nearby the objects and connected with additional elements, mostly lines and arcs. An internal annotation partially obscures the referenced object while directly establishing a mental link between both parts. External annotations are preferably used to annotate small objects as well as large numbers of objects, or to group objects spatially by a specific topic. Here, the user follows the connecting element to associate the annotation with the referenced object. Crossings or long distances between the object and the annotation, therefore, should be avoided.

There is a long tradition of annotating two-dimensional graphics, such as medical, botanical, or geological illustrations and geographical maps. Today, most techniques still lay out annotations in a two-dimensional manner even for interactive virtual 3D environments. Annotations are added after the scene elements are projected, that is, annotations are handled as 2D view-plane elements. This approach is referred to as view management and reduces the complexity for the arrangement by simplifying the annotation layout task to a two-dimensional placement problem. It has the advantage that these annotations parallel to the view-plane provide optimal readability.

However, whether an annotation technique is adequate or not depends to a large degree on the application context. If the user wants to inspect a single object from a bird's-eye view, the object can be centered in the view such that there remains enough white space around in which external labels can be placed. Elements can even be overlaid with textual descriptions if the visible amount of those parts is large enough. In contrast, if the user is immersed into a virtual environment (e.g., simulators for virtual

FULL Papers conference proceedings ISBN 80-86943-03-8
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

buildings and cities) the annotated objects surround the user. External annotations turn out to be problematic in this case because the white space for them needs to be detected, and this can only be solved at a semantic level. Even if a visible part of an object is large enough for an internal annotation, size, position, and number of these regions can completely change during the user movements, so they have to disappear or temporally overlay other objects.

In this contribution, we introduce the concept of *object-integrated* annotations, which are represented by separate 3D scene elements that keep dynamically attached to the referenced objects. They do not lose the visual link to the referenced objects and communicate their scope. We have implemented the approach in a prototypic system and applied for virtual 3D city models. In the following, we refer to *objects* of virtual environments or, if more specifically, to *buildings*.

2. RELATED WORK

For two-dimensional images, annotations can be implemented by integrating textual descriptions into the referenced image parts [Chi01a]. This concept of "dual use of image space" addresses the important problem of sharing space between image and text explanations and "achieves a smooth transition between the representation of an object as an image and its representation as a text" [Chi02a] to improve readability. For this Chigona and Strothotte develop a morphing technique that transforms a selected object into a rectangular window. In a sense, we extend this work to a "dual use of 3D space"; however, we do not distort 3D scene geometry.

In cartography, static label placement has been investigated yet for a long time, where typically text is integrated into the 2D maps; for a survey of algorithms see [Chr95a]. To achieve a high quality annotation placement, criteria such as disambiguation, selectivity, and expressivity of annotations are optimized [Har04a][Ebn03a][Edm96a].

Some approaches optimize these criteria with force-based algorithms [Ebn03a][Har04a]. The annotations are placed at initial positions on the view plane. Additional attracting and repulsive forces are defined among each other and the border of the view. Over multiple iterations, a relaxation process minimizes the overall forces, so that the annotations obtain improved positions. The computational costs do not allow for real-time label placement and, therefore, needs to be performed in a post-processing step.

The third dimension drastically increases the complexity of the label placement problem, both conceptually and algorithmically. 3D visualizations also go beyond classical map-based representations and demand different kinds of, and dynamic, labeling techniques. A first approach for 3D label placement is presented by Preim et al. [Pre97a]; their technique reserves part of the view plane for fixed containers for the textual descriptions linked by lines to the referenced objects. Ritter et al. [Rit03a] introduce the concept of illustrative shadows: Annotations are linked to reference points in the shadows of objects projected onto an additional shadow plane and, thereby, support an intuitive visual association. If there is enough white space between the shadows on the shadow plane and the object, the annotation placement can be reduced to a two-dimensional problem. Sonnet et al. [Son04a] investigate annotations in the context of interactive explosion diagrams intended to explore complex 3D objects.

The immersion of the user into annotated virtual environments (e.g., virtual pedestrian navigation) extends the degrees of freedom for annotation placement and partly the requirements for the annotation strategy. For example, for way finding in 3D city models annotations can occur that do not refer to visible objects. These annotations indicate nearby streets, places, or buildings, e.g., service stations, pharmacies, or hospitals. In addition, in immersive scenarios it is generally not desirable to depict all annotations; only a subset is determined that contains annotations referencing currently spatially near-by objects or objects relevant to a specific task. Kolbe investigates the annotation of buildings in prerecorded path videos [Kol04a]. To calculate the placement of annotations an external 3D city model is required. The annotations augment the geo-

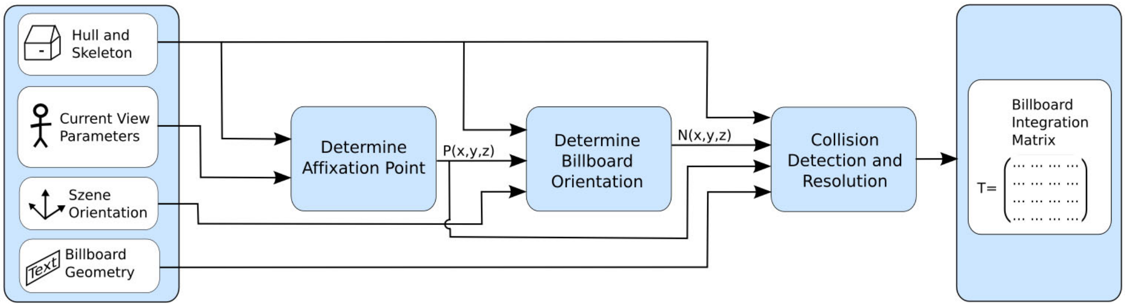

Figure 1: Overview over the annotation process.

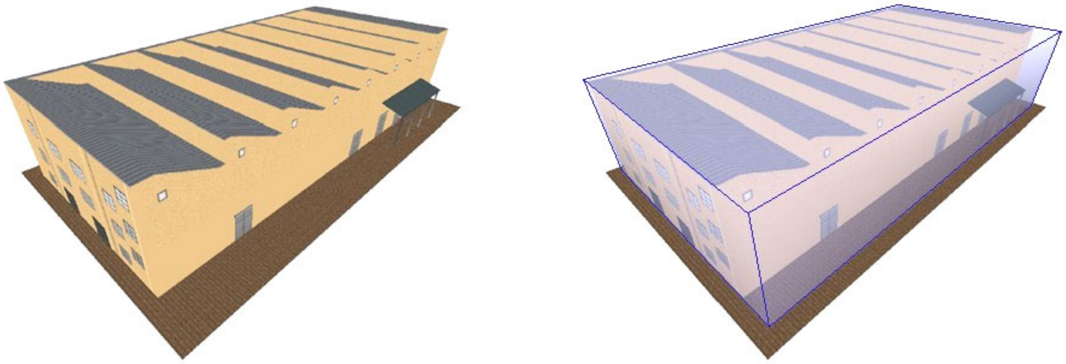

Figure 2: Factory building with a complex roof structure and a porch, with and without annotation hull.

referenced frame sequence of the video.

The work of Bell et al. [Bel01a] develops a technique for dynamically annotating 3D buildings by labels placed on the view plane based on the upright rectangular extent of object projections. Their viewing management system resolves visibility aspects and automates the switch between internal and external annotations. In our approach, in contrast, we integrate annotations in the 3D scene geometry, i.e., labels are not constrained to be parallel to the view plane.

3. OBJECT-INTEGRATED LABELS

In our approach, annotations are represented by 3D scene elements and, therefore, form part of the 3D scene geometry. An annotation is modeled by a textured billboard. The design of the annotation, i.e., its contents, is completely defined by the texture data. This way, we obtain a high degree of freedom for the graphical design of annotations; for the most common case of textual labels, corresponding textures can be created automatically. We can also specify multiple texture variants for different levels of detail, which are selected depending on the distance of the viewer.

The process of placing and integrating of an annotation can be divided into three steps (Figure 1). First, a fixation point at the referenced object is determined for the center of the billboard. Next, the billboard is orientated so that it appears to be integrated into the object. Because this can cause collisions between the annotation and the object, in the final step we detect and, if required, resolve such collisions.

3.1 Hull and Skeleton

To position the annotation our algorithm requires two *generalized variants of the annotated object*, a hull and a skeleton.

The hull represents the annotated object by a generalized geometry having a lower polygonal resolution. To generate a hull, we can take a low level-of-detail variant of an 3D object (e.g., progressive meshes) or derive the hull by object-specific techniques. For example, the hull of a 3D building model can be derived by eliminating small geometric details of its footprint and extruding the simplified footprint to its maximum height.

The hull is required to reduce the influence of detail geometry (e.g. in the case of 3D buildings: balconies, oriels, protrusions, etc.) that would cause a disturbed dynamic annotation placement. At the same time this decreases the computational costs in the case of geometric complex models. Figure 2 shows an example where a simple box is used as an annotation hull for a factory building with a complex roof structure and a protrusion.

The skeleton represents an internal supporting structure of a 3D object. For example, the skeleton of a 3D sphere can be represented by a single 3D point whereas the skeleton of a 3D torus can be represented by a 3D circle. For our approach, we require the straight skeleton, a collection of line segments that indicate the medial axes of the annotated object [Aic95a][Fel89a].

The skeleton is used to find a fixation point for the annotation. In particular, it is useful for those objects having a hull shape that is more complex than a sphere or a cube (e.g., buildings with non-simply shaped floor plan).

3.2 Affixation of Annotations

We propose two techniques for the calculation of the affixation point of the annotation in the 3D space. In the first variant a ray is sent from the viewer to the center of the annotation hull. The first ray intersection with the hull is used as the affixation point for the annotation. For simple object types (e.g., nearly cubical buildings in our case study) this affixation strategy is efficient.

For complex 3D objects, we can extend the affixation strategy. For example, in the case of a building with a complex floor plan, we create a skeleton that represents the structure of the floor plan. Techniques to create such skeletons are known from the automatic creation of roof geometries [Aic95a][Fel98a] and can be transferred to other object types as well. To determine the affixation point first the point at the skeleton is calculated that has a minimal distance to the ray in the center of the observer's view. If this point is known, a ray will be sent from the observer's view. The destination is the calculated point on the skeleton instead of the hull's center.

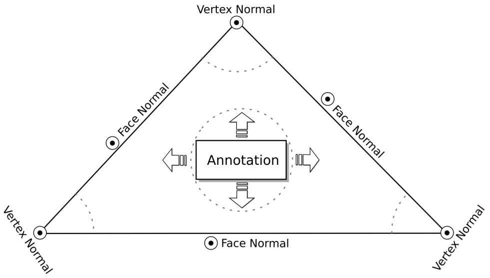

Figure 3: Normals of adjacent vertices and faces impact the resulting orientation, when the annotation affects their range of influence.

3.3 Orientation

The degrees of freedom for the annotations after the affixation permit only an orientation by a rotation around the affixation point. To find an orientation we use the vertex and face normals of the hull from the polygons that are near the polygon containing the affixation point. If the annotation can be completely integrated into this polygon of the hull without overlapping an edge, the orientation is equal to the face normal:

$$\vec{n}_o = \vec{n}_{affixation}^{\,face}$$

At hard edges the labels should smoothly disconnect from the current object face (e.g., a facade of a building), rotate around the edge, and smoothly integrate again into the adjacent polygon. For this, the normals at the vertices and adjacent faces must be taken into account as well (Figure 3).

The orientation of the label is influenced by a vertex normal if the distance d_i from the label center to the vertex is below the radius d_{max}, defined by a half of the label's diagonal length. In this case the orientation is calculated by

$$\vec{n}_o = \sum_{i=0}^{\#vertices} (d_{\max} - d_i)\cdot \vec{n}_i^{\,vertex} + d_i \cdot \vec{n}_m ,$$

where the second interpolation argument ($\vec{n}_m$) is the result of an interpolation in itself. For this the proportion of the angle between the adjacent polygon edges weights the face normals in clock and counter clockwise order around the vertex:

$$\vec{n}_m = ((1 - w_a)\cdot \vec{n}_{ccw}^{\,face} + w_a \cdot \vec{n}_{cw}^{\,face})$$

If the label is not near a vertex, but near an edge of the polygon, the orientation is determined in a slightly different way. For every edge with a distance below d_{max} we interpolate between the face normal of the adjacent polygon and the face normal of the polygon containing the affixation point as follows:

$$\vec{n} = \sum_{j=0}^{\#faces} (d_{\max} - d_j)\cdot \vec{n}_j^{\,face} + d_j \cdot \vec{n}_{affixation}^{\,face}$$

Afterwards, the annotation is oriented so that the face normal is pointing in the same direction as the calculated normal. The result of these calculations is demonstrated in Figure 4, where these orientation normals are calculated in discreet steps over the hull.

3.4 Collisions

The affixed annotation can intersect the object so that it is partially occluded after it has been oriented (Figure 6). To avoid this, an additional step must detect and resolve such collisions.

3.4.1 Image Space

A simple screen-space technique can be used to

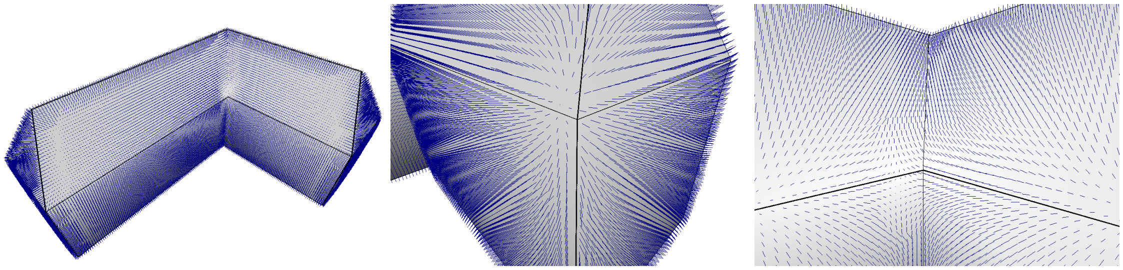

Figure 4: Normals, used to orient the annotation, calculated at discrete samples of the hull (overview, detailed view on a convex and concave edge).

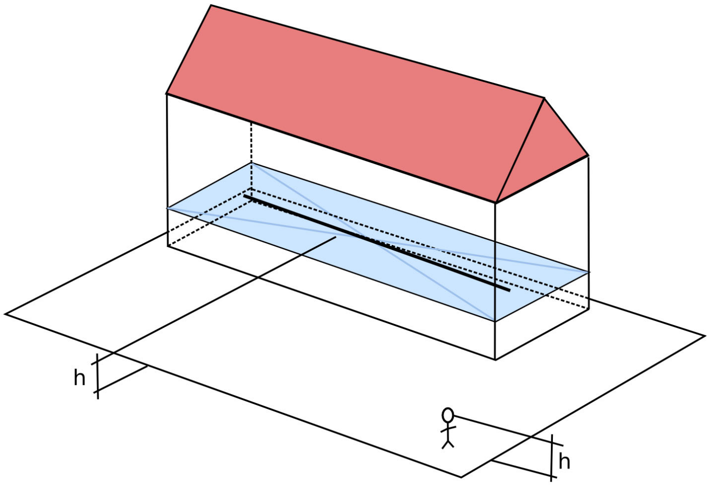

Figure 5: Affixation of the skeleton at the viewer's eye level instead in the center of the object.

avoid the occlusion of an annotation from the referenced object but requires a sorting of all scene elements (without annotations) by their distance to the observer. Starting with the furthest object all elements are drawn in this ordering [Fuc80a]. If this process comes upon an annotated scene element, the object is drawn first. After this the assigned annotation is rendered but with disabled depth-buffer test. As a consequence, it is drawn over the object, even if penetrating it.

3.4.2 Object Space

For scenes with nearby objects or higher geometric complexity an efficient and well-defined ordering is not always possible. This is the reason to develop a technique that detects and resolves the collisions in object-space.

For the annotation billboard we span a plane into the object space and check for possible intersections with the annotation hull. If these intersections are inside the borders of the billboard, it is moved using the direction of the orientation normal as a separating axis until the conflict is resolved.

For these tests only the polygons of the hull are used to reduce computational costs. If required, the intersection tests can be accelerated by making use of bounding volume hierarchies (e.g., [Got96a]) to reduce intersection tests to a few polygons only.

3.5 Annotation Selection

Using our approach in interactive applications with larger data sets raises the question of scalability. Dependent of the kind of application and the number of annotations, this question can be answered differently:

Thematic Limitation: In interactive applications the user can chose subsets of annotations that are to be displayed. These groups could be built from semantic information (e.g., landmarks, street names, building usage) in advance or created by a search request of the user (e.g., hints for way finding).

Restriction by Visibility Culling: We can reduce the number of annotations to be handled by visibility culling [Ass00a]. Only for visible objects or objects near the view frustum annotations are considered to be active. Additionally, the distance to the observer can be used to exclude annotations that are far away. During the transition between the visible and non-visible state the annotation's transparency is smoothly faded to avoid popping effects.

Delayed Annotation: We can give up or delay the interactive placement of annotations, that is, annotations stay fixed at their last positions until the interaction process comes to an idle state. Then, these annotations can directly be placed (with fade-in) at their new positions or can move along a path on the hull surface in order to allow the observer's eyes to follow them.

4. CONSTRAINTS

Constraints provide additional control how to place annotations in the 3D scene. We have worked out two constraint types so far.

4.1 Observer Height Constraint

Since the position of the skeleton inside the annota-

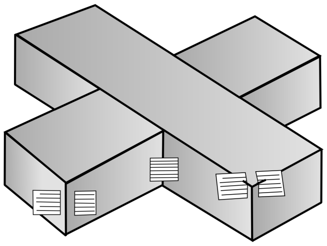

Figure 6: Different kinds of annotation-object collisions after the orientation pass of the process.

tion hull strongly affects the place of affixation, we can control the annotation placement by constraining the skeleton. Commonly, the skeleton is centered in the annotation hull to allow for an equal distribution of affixation points. Keeping the skeleton at the same height as the viewer's eye level (Figure 5) allows us to control the placement not only by the hull's shape but also by the viewer's position. As a result annotations tend to be more prominent in the user's view. For example, this type of constraint improves the placement in scenarios where the user interacts as a virtual pedestrian.

4.2 Object Height Constraint

A simple way to constrain the placement more by the shape of the hull and to limit it to only one dimension is to define a plane through an object. The position of the annotation can be restricted to the intersection line of this plane with the annotation hull. For this, the skeleton and the position of the viewer are projected down to the plane before the ray intersection between the viewer and the hull is calculated. As an example, for buildings this can be used to limit the annotation placement at the facade to the white space between the windows of two adjacent floors.

To restrict the placement area to a horizontal band of the object surface, a second plane, parallel to the first one, can be defined. If the observer is located between these planes the algorithm without any modifications is used, if not the position is projected to the nearest plane as described before.

5. RESULTS

We have tested our annotation technique with a set of individual objects and a 3D city model of our university campus (Figure 7, 8). The annotations have a defined background color and, therefore, provide a good compromise between high perception quality and seamless object-space integration. Since these annotations can occlude parts of the objects they can be rendered transparently.

In comparison to two-dimensional, view plane parallel annotations, the object-space integration of the annotations reduces the readability. This is uncritical in immersive scenarios where the user is navigating through the information space. The users realize the annotations as an additional information source and can easily obtain a position with better readability.

In our future work, we will develop additional quality criteria (e.g., degree of object-integration, quality of the integration area) and relate them to existing conditions (e.g. visibility, legibility). As a first step, a visibility determination feedback loop could be integrated into the process that optimizes the annotation position along the visible part of the hull.

A strength of the dynamic object-space annotations is that they allow for an exact identification of the annotated object parts. Unlike the additional lines of external labels or two-dimensional overlays of internal labels, the object-integrated labels communicate their areas of validity. Through this, in our case study the user was able to distinguish between the cases where the whole building, only a part, or a facade is referenced by the annotation.

Another benefit of object-integrated annotations is the high utilization of the available screen space – an aspect important for mobile navigation assistants having small displays.

While the user navigates through the 3D scene, the annotations are floating dynamically over the object

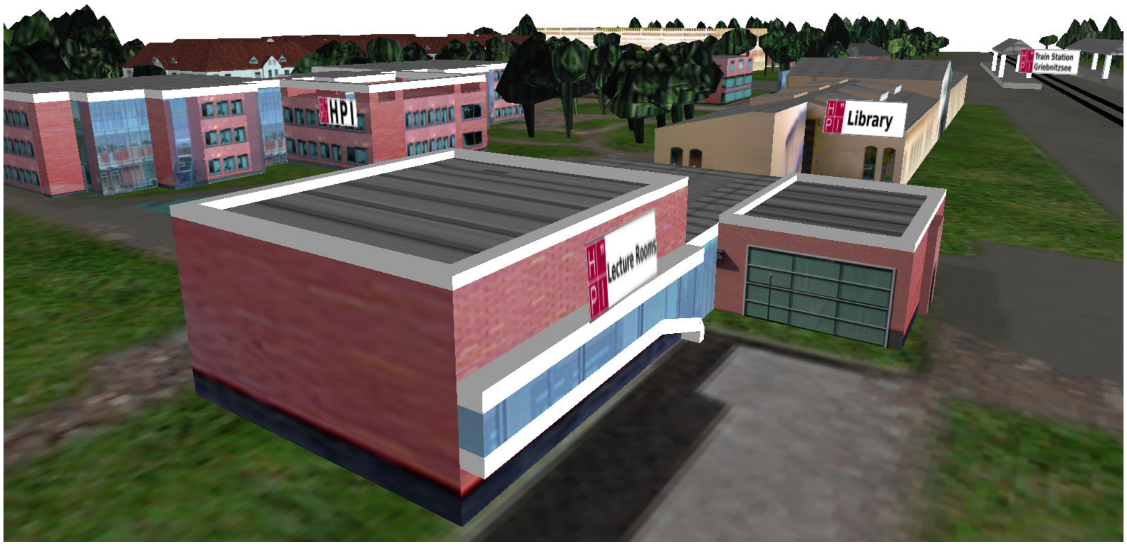

Figure 7: User study with multiple objects: annotated buildings of a campus model.

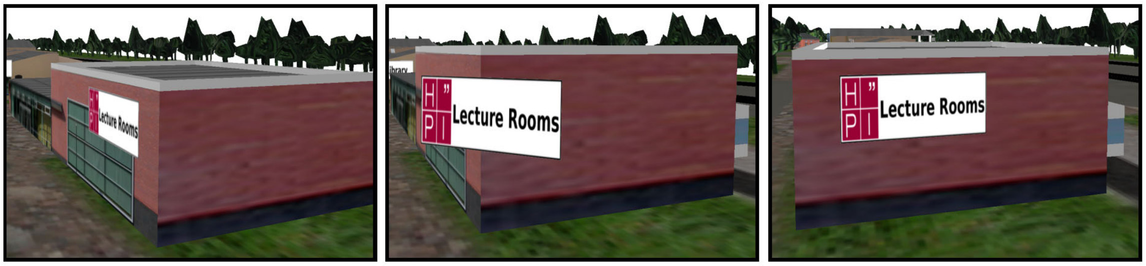

Figure 8: Sequence, showing an annotation detaching from one facade, rotate around the edge an integrate into the adjacent facade.

surfaces. Using a skeleton with more than one segment can sometimes cause some unsteady jumps, if the part changes that contain the point nearest to the view ray. An interpolation over the hull surface could solve this problem but requires additional calculations to plan the path between the last and the new position.

Because the geometry of the annotation hull controls the affixation and orientation of the annotation, the annotation creator gets degrees of freedom for designing this hull. For example, a more rounded hull could cover buildings with a lot of hard edges near by each other but if the difference between the hull and the object gets over a considerable threshold, the quality of the correlation will suffer.

Object-integrated annotations are suited for scenarios where the user is in the role of a virtual pedestrian. The annotations are perceived as commonly known plates, with the only difference that they are moving. The benefits decrease more and more if the user comes to a bird's eye perspective. In our test environment, especially buildings with flat roofs turned out to be problematic because the annoations get integrated rapidly into the roofs and, therefore, cannot be seen anymore as outstanding labels. Partially, this can be corrected with an adjustment at the hull. A more general solution should look for a transformation to another annotation technique, if the user is leaving the immersive perspective.

6. CONCLUSIONS

The concept of object-integrated annotations provides a solution to enrich 3D objects with thematic, textual or symbolic information within an immerse view to a virtual environment. Compared to view-plane parallel annotations, object-integrated annotations communicate more precisely their areas of validity in interactive applications. Additionally, they reduce the irritating situation that user interaction in a 3D environment results in 3D visual feedbacks from the scene together with 2D repositioning responses from the annotations.

Our approach allows for an easy integration of annotations into 3D models with a time-coherent movement during user interaction. The interactive behavior can be simply controlled by the explicit design of the hull, the skeleton, and additional constraints.

In our future work, we want to complete the implementation by an automatic generalization technique that extracts a first variant of the hull and the skeleton from the original models. Furthermore, level of detail techniques can be integrated at different points of the approach to improve the display of the annotations.

For the extension to non-immersive environments, more attention must be paid to the occlusion of annotations among each other and by other objects. Moreover, our concept can be complemented by a non object-integrated annotation technique. The transition between different annotation techniques is another research topic we plan to study.

Concerning form and placement, the buildings in our case study show that objects, where planar areas are predominating, are especially suited for embedding billboards. Here, the annotations are directly mounted at the facades, resulting in a good correlation between annotation and object.

ACKNOWLEDGMENTS

This research is partially supported by the EU research and technology development project TNT (www.the-neanderthal-tools.org/).

We thank our anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[Aic95a] Aichholzer, O., Aurenhammer, F., Alberts, D., Gartner B. A Novel Type of Skeleton for Polygons. Journal of Universal Computer Science, 1(12): 752-761, 1995.

[Ass00a] Assarsson, U., Möller, T. Optimized View Frustum Culling Algorithms for Bounding Boxes. Journal of Graphic Tools, 5:1, Jan., 9-22, 2000.

[Bel01a] Bell, B., Feiner, St., Höllerer, T. View Management for Virtual and Augmented Reality. Proceedings of the 14th Anual ACM Symposium in User Interface Software and Technology, 101-110, 2001.

[Chi01a] Chigona, W., Schlechtweg, S., Thomas, S. Dual Use of Image Space: The Challenges of Explaining Visualizations from Within. In Schulze, T., Schlechtweg, S., Hinz, V. (eds). Simulation und Visualisierung 2001, SCS-Society for Computer Simulation Int., Delft, Belgium, 175-185, 2001.

[Chi02a] Chigona, W., Strothotte, T. Distortion for Readability of Contextualized Text Explanations. Proceedings of IEEE International Information Visualization 2002, London, 289-294, 2002.

[Chr95a] Christensen, J., Marks, J., Shieber, S. An Empirical Study of Algorithms for Point-Feature Label Placement. ACM Trans. Graph. 14, 3 (Jul. 1995), 203-232, 1995.

[Ebn03a] Ebner, D., Klau, W.K., Weiskircher, R. Force-Based Label Number Maximization, Technical Report TR-186-1-03-02, Technische Universiät Wien, June 2003, available at http://www.apm.tuwien.ac.at/publications/bib/pdf/ebner-03.pdf

[Edm96a] Edmondson, S., Christensen, J., Marks, J., Shieber, S. M. A general cartographic labeling algorithm. TR1996-004, Cartographica 33, 4, 13-23, 1996, available at http://www.merl.com/publications/TR1996-004/

[Fel98a] Felkel, P., Obdrmalek, S. Straight Skeleton Implementation. Proceedings of the 14th Spring Conference on Computer Graphics, Budmerice, Slovakia, 210-218, 1998.

[Fuc80a] Fuchs, H., Kedem, Z.M., and Naylor, B.F. On Visible Surface Generation by A Priori Tree Structures. Computer Graphics (Proceedings of ACM SIGGRAPH 80), 14(3), 124-133, 1980.

[Got96a] Gottschalk, S., Lin, M.C., Manocha, D. OBBTree: A Hierarchical Structure for Rapid Interference Detection, Computer Graphics (Proceedings of ACM SIGGRAPH 96), 171-180, 1996.

[Har04a] Hartmann, K., Ali, K., Strothotte, T. Floating Labels: Applying Dynamic Potential Fields for Label Layout, Lecture Notes in Computer Science, Volume 3031, Jan, 101-113, 2004.

[Kol04a] Kolbe, T. H. Augmented videos and panoramas for pedestrian navigation. Proceedings of the 2nd Symposium on Location Based Services and TeleCartography, 2004.

[Pre97a] Preim, B., Raab, A., Strothotte, T. Coherent Zooming of Illustrations with 3D-Graphics and Textual Labels. Proceedings. of Graphics Interface, 105-113, 1997.

[Rit03a] Ritter, F., Sonnet, H., Hartmann, K., Strothotte, T. Illustrative Shadows: Integrating 3D and 2D Information Displays. Proceedings of the 8th International Conference on Intelligent User Interfaces 2003, Miami, Florida, 166–173, 2003.

[Son04a] Sonnet, H., Carpendale, S., Strothotte, T. Integrating Expanding Annotations with a 3D Explosion Probe. Proceedings of the Working Conference on Advanced Visual Interfaces, Gallipoli, Italy, 2004.