

Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes

Stefan Maass and Jürgen Döllner

Department of Computer Graphics Systems
University of Potsdam, Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{maass, doellner}@hpi.uni.potsdam.de

Abstract. We present a dynamic placement technique for annotations of virtual landscapes that is based on efficient view management. Annotations represent textual or symbolic descriptions and provide explanatory or thematic information associated with spatial positions. The technique handles external annotations as 2.5 dimensional objects and adjusts their positions with respect to available space in the view-plane. The approach intends to place labels without occlusions and, if this cannot be achieved, favors those annotations that are close to the observer. This technique solves the visibility problem of annotations in an approximate but user-centric way. It operates in real-time and therefore can be applied to interactive virtual landscapes. Additionally, the approach can be configured to fine tune the trade off between placement quality and processing time with a single parameter.

1 Introduction

Annotations are essential elements to enhance depictions with meta information such as explanations and thematic information. While annotation techniques are well studied and developed for traditional static two-dimensional media, e.g., in geographic maps or medical illustrations, annotation techniques for dynamic three-dimensional virtual environments still represent an important challenge for computer graphics and visualization.

The depiction of annotations in interactive virtual 3D environments shows fundamental difficulties because annotations by their nature are not inherently three-dimensional geometric objects and, therefore, cannot directly be represented as regular scene elements. Partly, these difficulties arise from the general problem of integrating text and image representations in a perceptive and cognitive efficient as well as aesthetically convincing way.

In this paper, we present a new technique for the management of annotations in virtual landscapes such as 3D maps, 3D landscape models, or 3D city models. By a virtual landscape we refer to a virtual 3D environment that contains as predominant element a terrain surface. The annotations refer to point features of these landscapes. The technique handles annotations as 2.5 dimensional objects of the scene and adjusts their positions with respect to available space in the view-plane (cp. Fig. 1). Our



Fig. 1. Sample virtual landscape with dynamically placed annotations.

technique intends to place annotations without occlusions. If this cannot be achieved, it favors those annotations that are close to the observer. The view management operates in real-time and, therefore, can be applied to interactive virtual landscapes. We used scene integrated annotations for our approach. They provide the important property of a depth cue because an annotation's size depends on its 3D position in the virtual landscape.

Applications of our approach include the annotation of spatial objects and spatially referenced data. In particular, we can label features of high interest or mark inspected positions or regions. Additionally, annotations play an important role in interactive collaborative geovirtual environments, and are required to include spatial comments and explanations.

2 Related Work

In general, we can differentiate between internal and external annotation techniques. An *internal annotation* is drawn inside the visual representation of the referenced object and partially obscures that object. Internal annotations directly establish a mental link between annotation and annotated object. They are preferably used if the depiction of their referenced object offers sufficient space to enclose the annotation and does not distort the depiction. To share the space between an image and text explanations Chigona et al. [4] extend this to the principle of “dual use of image space”, where a depiction can transform between the representation as an image and

the representation as text. A technique for dynamic placement of object-integrated internal annotations of typical objects of geovirtual environments (e.g., buildings) is presented in [12].

An external annotation is drawn outside the visual representation of the referenced object and uses a connecting element such as a line or arc to associate the annotation with the referenced object. To avoid ambiguities, crossings or long distances between the object and the annotation, therefore, should be avoided. 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.

General criteria for the quality of an annotation technique include non-overlapping placement, the support of annotations with different priorities, interactive placement, and aesthetic label layout [1], [6], [10].

2.1 Label Placement Techniques

In cartography, the static label placement for point, line, and area features represents a classical problem and has been investigated yet for a long time, where typically text is integrated into 2D maps; for a survey of algorithms see [5]. For some detailed labeling problems it was shown that finding an optimal solution is NP-hard [13].

To achieve a high quality annotation placement, criteria such as disambiguation, selectivity, and expressivity of annotations are approximated [6], [7], [9]. Some approaches optimize these criteria with force-based algorithms [6], [9]. The annotations are placed at initial positions on the view plane. Attracting and repulsive forces are defined among between them and the border of the view. A relaxation process then minimizes the overall forces over multiple iterations, so that the annotations obtain improved positions. The computational costs do not allow for real-time label placement and, hence it needs to be performed in a post-processing step.

Visual depictions in 3D demand dynamic and different types of annotation techniques which are both conceptually and algorithmically more complex. Preim et al. [15] present a first approach for 3D label placement, where fixed containers are reserved on the view plane to hold textual descriptions linked by lines to the referenced objects. Ritter et al. [16] 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. Sonnet et al. [17] investigate annotations in the context of interactive explosion diagrams intended to explore complex 3D objects. Kolbe investigates the annotation of buildings in pre-recorded path videos [11], which augment the geo-referenced frame sequence of the video. To calculate the placement of annotations an additional 3D city model is required.

2.2 View Management Techniques

The term *view management* denotes techniques that handle the available space in the 2D view plane. In the context of labeling, view management is used to avoid the overlapping of labels after their projection. All listed approaches assume that the view

planes as well as all free or occupied regions on it are represented with axis parallel rectangles. Typical operations are the mark and release of regions, as well as query operations, e.g., if a given rectangle intersects some occupied region.

Bell et al. develop a view management technique for the placement of widgets on the screen and use this later on to annotate buildings with text and symbols in an interactive 3D environment [2], [3]. In this approach the unoccupied regions are efficiently represented by a set of maximum extended rectangles. Thereby all non-occupied regions are easily accessible. In contrast to our approach, it is not possible to calculate directly an alternative position in case a label intersects an occupied region; all free regions must be iterated.

The work of Müller and Schödel [14] describes a technique to label the components of column charts. Their approach uses up to four borders to manage the remaining space, each one growing from a different edge of the view plane. However, they focus on a different, two dimensional scenario with other constraints.

3 Annotation Placement Strategy

Our dynamic placement technique for annotations in virtual landscapes takes advantage of two general characteristics of perspective views of terrain-based scenes:

- Typically, users look from a position at the upper hemisphere down to a point on the terrain surface. Therefore, the anchor points of annotations near to the observer are rather located in the lower area while reference points far away tend to be located in the upper area of the view plane.
- Annotations near the observer tend to be of higher interest than annotations far away.

Taking these characteristics into account we can define an adequate placement by arranging annotations close to the observer in the lower part of the view plane and raising their height with increasing distance to the viewer. Close annotations are connected with short pole lines to their anchor points. Because we can assume that these annotations are in the current focus of the user, the placement strategy fulfills one important criterion for a good placement [6]. Placing the annotations the other way around would generate close annotations in the upper screen area whose pole lines would overlay annotations that are far away and located in the lower area.

The annotation placement strategy avoids overlapping annotations by reserving disjunctive screen space in the view-plane for each annotation. Since overlaps cannot be avoided if the number of annotations and the available screen space do not correspond, a non-overlapping annotation near the observer should be preferred to a non-overlapping annotation far away. In our implementation, we sort the annotation anchor points on the terrain ground according to their depth value, and process them in a front-to-back order. Processing the annotations in this order supports our objective because annotations close to the viewer can reserve their space by the view management at an early stage in the placement process.

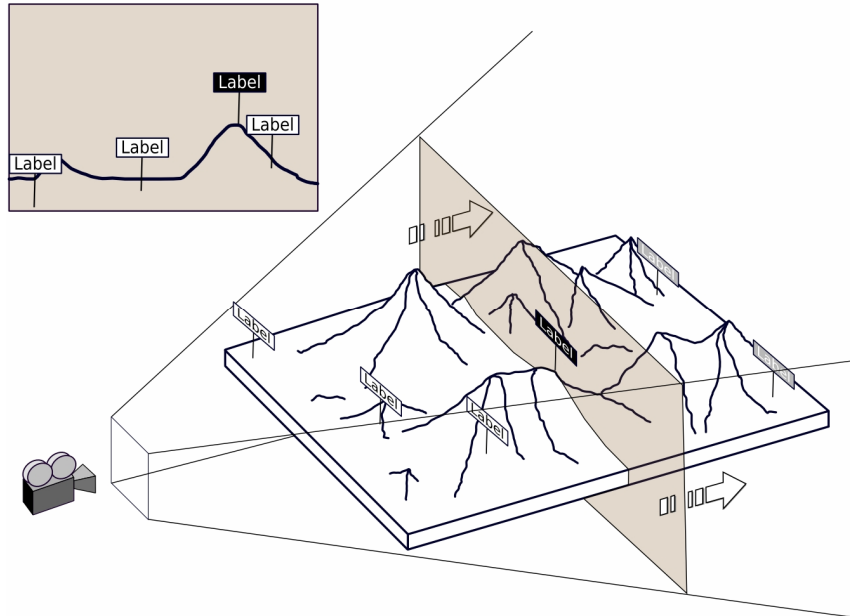


Fig. 2. Illustration of the placement strategy for virtual landscapes: Labels are processed in front-to-back order. Possible collisions are then resolved within their two-dimensional plane.

Fig. 2 illustrates the placement strategy for a sample virtual landscape. The annotations in front are already processed and placed within the view-plane. The projection of the currently processed annotation can collide with already processed annotations, so that the view management determines alternative placements. To avoid mixing of orthogonal and straight-line label layouts ([1], [10]) annotations are elevated until a free view-plane position is found. To find an alternative position the strategy does not look into directions other than the look-up direction.

To achieve high readability, the annotation placement strategy orients the annotations parallel to the view-plane. Nevertheless, they are treated as 3D objects in the virtual landscape having a depth value and being projected accordingly. The view management uses the view-plane up-right extends of the oriented label to detect overlaps with others. The height over the anchor point is the only degree of freedom for the label, which allows the algorithm to find a new and probably collision-free position.

For each frame, our annotation technique performs three main steps.

1. The annotation anchor points are culled against the view frustum to skip all labels that are not visible or too close to the view-plane. The technique uses a conservative culling because anchor points may be outside the view frustum while the annotation is partially visible. Then, the remaining anchor points are sorted by their distance to the observer. This ordering is used in the next two steps.
2. For the actual placement, the algorithm processes the anchor points for the visible annotations in front-to-back order and determines the positions of each annotation on the view plane. For the final view-plane position, the corresponding world coordinates are calculated and stored. Note that the probability to find free positions on the view plane is higher for the labels close to the observer. In addition, close labels are placed below distant labels.
3. The virtual landscape, anchor points, pole lines, and annotations are rendered. The labels are drawn in back-to-front order to guaranty that transparent parts are shown without artifacts.

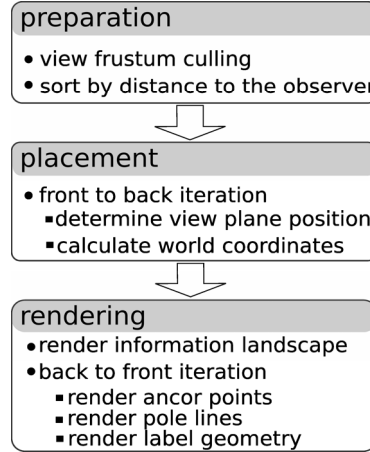


Fig. 3. Overview of the rendering

4 View Management

The view management divides the screen space horizontally into discrete slots due to the vertical search direction in our technique. As a result, query and update operations can easily be restricted to the horizontal band that an annotation occupies.

Dividing the vertical dimension in discrete steps as well would lead to a raster structure. To achieve high placement precision these cells would have to be small involving high costs for query and mark operations. Since the spatial extents of annotations on the screen are larger than the targeted precision, we use a linked list for each slot, where each element stores a non-occupied interval, starting from the lower edge of the screen.

For the view management a normalized view-plane coordinate system is used, with coordinates $x_{\min}, y_{\min} := (0,0)$ at lower left and $x_{\max}, y_{\max} := (1,1)$ at the upper right corner. Starting with a completely empty view-plane, rectangular regions (x_0, y_0, x_1, y_1) , $x_i, y_i \in [0,1]$ can be added to mark occupied areas.

To detect collisions efficiently and to perform the label placement in real-time, the view management needs efficient operations for the following tasks:

- Mark new regions as occupied
- Test if a given rectangle overlaps occupied regions
- Find the closest empty region for a given rectangle in the case of overlaps

Classical space partitioning techniques, e.g., regular grids or quad-tree structures, can be easily adapted to fulfill the first and second task. But they are less suitable for searching empty regions or to directly determine closest unoccupied region. For this reason we develop two variants of view management that consider the special properties of our setting; the *growing-border* and *interval-slot view* management.

4.1 Growing-Border View Management

For the growing-border view management each slot contains only one y -interval during the whole placement process. It is initialized with $(y_{\min} = 0, y_{\max} = 1)$ to mark the whole slot as empty. If an area (x_0, y_0, x_1, y_1) should be marked as occupied, the algorithm iterates over all slots that are overlaid by the interval $[x_0, x_1]$ and replaces the value y_{\min} in every slot with $\max(y_1, y_{\min})$.

To query for a collision, all these slots are iterated in the same way. If one of them contains a y_{\min} -value greater than the query rectangle's y_0 , the iteration can stop because a collision is found.

Furthermore, the query algorithm can directly determine at which height the rectangle must be placed to avoid the collision. For this, the iteration continues until all slots are taken into account. The maximum y_{\min} value found in the slots during this iteration is the new height for the placement. To avoid placement outside of the view-plane a horizon defines a maximum y value. With this, the maximum y value that a label can reach is $y - \text{labelheight}$. Fig. 4 illustrates the placement of four labels A-D with the growing-border technique.

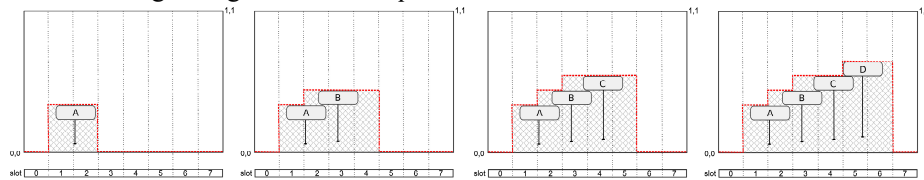


Fig. 4. Placing the labels A-D by the growing-border view management. The occupied view-plane region is hatched.

4.2 Interval-Slots View Management

With the growing-border technique annotations are simply stacked, so that typically large parts of screen space are unnecessarily marked as occupied. For this reason we extend the view management in such a way that we store for every slot an ascending sorted list of empty intervals (y_{\min}, y_{\max}) , $y_{\min} < y_{\max}$. Initially, each slot contains one interval $(y_{\min} = 0, y_{\max} = 1)$, to mark the whole slot as empty.

To test a new rectangle for a collision with, the algorithm needs to iterate over the involved slots as before. Additionally, for each slot list we search for empty intervals within the region $[y_0, y_1]$. If there is one slot that shows a collision with the rectangle, the query can stop with a negative result.

With a slight modification the algorithm can be extended to return the next empty position for a query rectangle. As long as there is no collision during the iteration, we keep track of the empty region above the current label position. If one slot raises a collision, we check if the label fits into the current empty region after we have lifted it above the obstacle. In this case the iteration can continue. Otherwise, it is restarted after the rectangle has been lifted above the whole empty region.

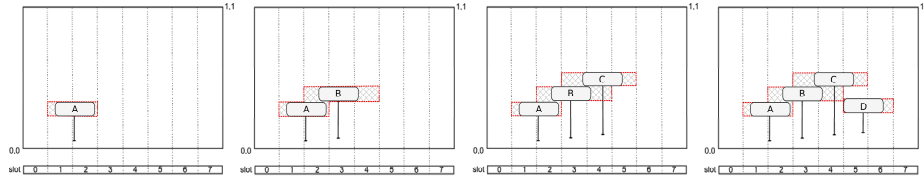


Fig. 5. Placing the labels A-D by the interval-slot view management.

5 From Planar Maps to 3D Landscapes

The algorithm as described so far works perfectly for (near) planar maps shown in perspective views. For 3D landscapes, which show significant non-planar morphology, the 3D geometry has to be considered to avoid the intersection of labels. To accomplish this, we extend the test of our placement strategy.

For the calculated view-plane position of an annotation we additionally verify for the screen area the label overlays that no part of the 3D landscape is closer to the observer than the label. If this test fails the label is raised until this conflict is resolved as well.

For the depth value test we render the 3D landscape in a separate depth buffer with a low resolution before the placement starts. Because of the planar shape of the label and its orientation to the observer, the depth value of the label is constant and needs to be calculated only for one point. To speed-up this test a hierarchical z-pyramid can be used [8]. For 3D landscapes with predominant convex features the evaluation of the depth values can be restricted to sample positions. We found that a sampling at the label corners and at the midpoints of the horizontal label borders suffice in the most cases and introduce artifacts only in rare situations.

As an alternative, the depth criteria can be validated with explicit ray tests as well. This method is useful if the scene has a low depth complexity, contains only a few visible labels, or the precision of the discrete depth buffer does not suffice.

6 Extensions

Visibility of Scene Elements. The view management can be used to manage the visibility of other scene elements as well. For example, in some scenarios users report the need of a clear indication where anchor points for the label are exactly located. Here, additional spheres or boxes are used to mark these points. To guaranty the visibility of the nearest, their bounding boxes are projected onto the view plane and

inserted into the view management before the label placement starts. Another example is the initial masking of a screen space application logo to avoid label placements in this region.

Priorities. In some applications, specialized labels exist that must be preferred in terms of their visibility. For this the placement process can be extended to support labels with different priorities. It iterates first over all priorities and in an inner loop over the labels with the current priority from a front-to-back order. For the rendering step this attribute can be ignored so the labels are drawn back-to-front without the consideration of priorities.

Dynamic Aspects. During navigation through the 3D landscape, the label positions can change at a high frequency and possibly spread over the whole view-plane space. This causes unsteady jumps in the visualization, irritating the user and makes it nearly impossible to trace labels that were currently in focus. To ease this situation we add a hysteresis characteristic to the placement. This way the labels keep their current positions during the 3D interaction and perform a continuous movement to their final destination once the user pauses or finishes the interaction.

7 Results

We have developed a prototypic implementation of our dynamic annotation placement approach. Fig. 6 shows the different placement strategies on a flat map. In the example, 100 labels with the same text are placed randomly with a constant height (Fig. 6a), with the growing border (Fig. 6b), and with the interval-slots view management (Fig. 6c). In contrast to the constant height placement, the view management strategies try to avoid overlapping labels and the occlusion of labels from nearer scene objects (Fig. 1, Fig. 7). These visibility criteria, most likely achieved for labels near the observer, result in an improvement of legibility.

We expected that the interval-slot view management results in a too compact label arrangement but due to the perspective scaling closely placed labels can be distinguished from each other. It seems that the greater pole lengths generated by the growing-border approach are more visual disturbing than the compact label layout of the interval-slot approach. This needs to be evaluated in a user study.

Currently all labels are placed centered over their pole line. This provides the user with a clear mapping between the text, the line and the anchor point. Introducing an additional degree of freedom at this point could be used to avoid a clustering of labels in a too compact layout and to support explicit aesthetic parameters in the placement process.

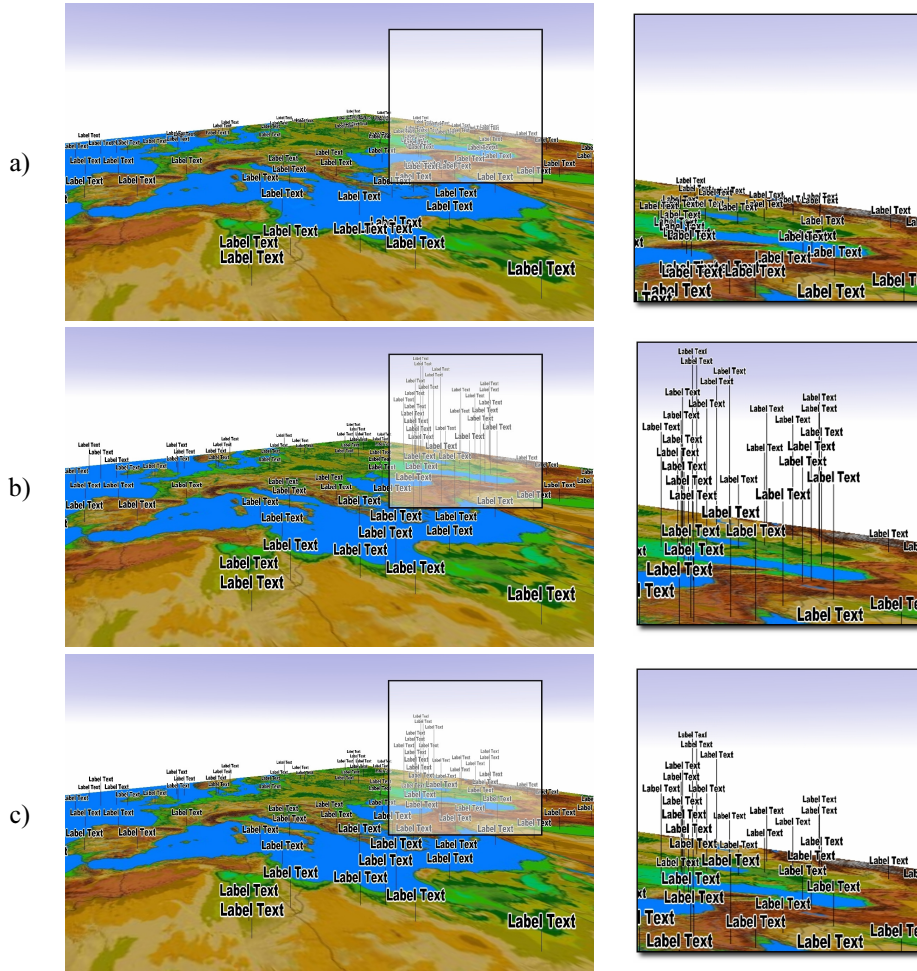


Fig. 6. Label placement with a) constant height, b) growing border, and c) interval slot view management.

The number of interval slots controls the precision of the placement. If the view plane is divided in only a few slots, the region marked as occupied for each label expand along the horizontal axis of the screen, so that the whole labeling extents into the height. An upper bound for the number of interval slots is the number of pixels used in this dimension. Then neighboring labels can be placed with maximum accuracy using all available screen space with the downside of higher costs for update and query operations. Adjusting the number of interval slots allows the user to balance between placement precision and processing time.

Fig. 7 shows label placements with the interval-slot view management at a 3D landscape.

8 Conclusions and Future Work

This paper presents a point-feature dynamic annotation placement strategy for virtual landscapes. Our concept uses the properties of virtual landscape views in combination with an efficient view management. Annotations are not only two-dimensional screen objects but represent three-dimensional scene objects so that they appear as integrated elements of the 3D scene and provide inherent depth cues.

The implementation operates in real-time and, therefore, can be applied in interactive applications. It can also be easily extended to force the visibility of other scene elements, supports labels with different priority, and provides smooth transitions between changing annotation positions.

Future work will focus on interactive exploration techniques. We also plan to investigate how to add interactive behavior to annotations. For a top view on the 3D landscape established cartographic label placement techniques could be used to find placements that are more common for traditional map views.

Acknowledgements

We would like to thank the Neanderthal Museum in Mettmann, Germany, especially Jan Kegler, for providing us with the cartographic material.

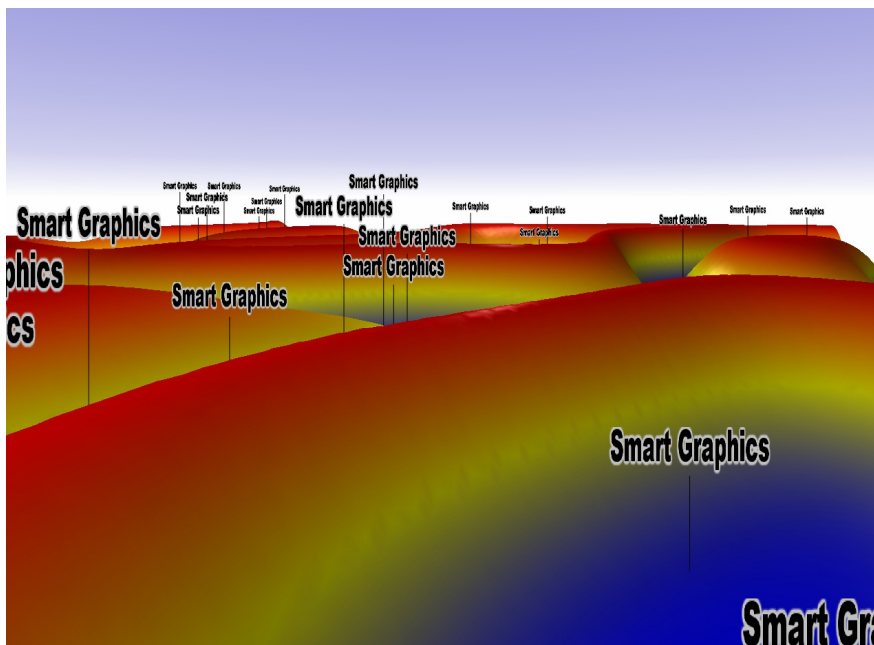


Fig. 7. Label placement on a 3D landscape represented by a height field.

References

1. Ali, K., Hartmann, K., and Strothotte, T. Label Layout for Interactive 3D Illustrations. *Journal of the WSCG*, 13 (1):(2005), 1–8.
2. Bell, B. and Feiner, S. Dynamic Space Management for User Interfaces. In *Proceedings of the 13th ACM Symposium on User Interface Software and Technology (UIST)*, volume 2 of CHI Letters. San Diego, CA, 2000. pp. 239–248.
3. Bell, B., Feiner, S., and Höllerer, T. View Management for Virtual and Augmented Reality. In *Proceedings of the 14th ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 2001. pp. 101–110.
4. 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.
5. Christensen, J., Marks, J., and Shieber, S. An Empirical Study of Algorithms for Point-Feature Label Placement. *ACM Transactions on Graphics*, 14 (3):(1995), 203–232.
6. Ebner, D., Klau, W.K., Weiskircher, R. Force-Based Label Number Maximization, Technical Report TR-186-1-03-02, Technical University Vienna, June 2003, available at <http://www.apm.tuwien.ac.at/publications/bib/pdf/ebner-03.pdf>.
7. 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/>.
8. Greene, N., Kass, M., and Miller, G. Hierarchical Z-Buffer Visibility. *Computer Graphics*, 27 (Annual Conference Series):(1993), 231–238.
9. Hartmann, K., Ali, K., and Strothotte, T. Floating Labels: Applying Dynamic Potential Fields for Label Layout. In *Smart Graphics: 4th International Symposium (SG 2004)*, volume 3031. Springer-Verlag, 2004. pp. 101–113.
10. Hartmann, K., Götzelman, T., Ali, K., and Strothotte, T. Metrics for Functional and Aesthetic Label Layouts. In A. Butz, B. Fisher, A. Krüger, and P. Olivier, editors, *Smart Graphics: 5th International Symposium, SG*, volume 3638. Springer, Frauenwoerth Cloister, Germany, 2005. pp. 115–126.
11. Kolbe, T. H. Augmented Videos and Panoramas for Pedestrian Navigation. In G. Gartner, editor, *Proceedings of the 2nd Symposium on Location Based Services and TeleCartography*, 2004.
12. Maass, S. and Döllner, J. Dynamic Annotation of Interactive Environments using Object-Integrated Billboards. In J. Jorge and V. Skala, editors, *14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'2006*. Plzen, Czech Republic, 2006. pp. 327–334.
13. Marks, J. and Shieber, S. The Computational Complexity of Cartographic Label Placement. Technical Report TR-05-91, Harvard University, 1991.
14. Müller, S. and Schödl, A. A Smart Algorithm for Column Chart Labeling. In *Smart Graphics: 5th International Symposium (SG 2005)*, 2005. pp. 127–137.
15. Preim, B., Raab, A., Strothotte, T. Coherent Zooming of Illustrations with 3D-Graphics and Textual Labels. *Proceedings of Graphics Interface*, 105-113, 1997.
16. Ritter, F., Sonnet, H., Hartmann, K., and Strothotte, T. Illustrative Shadows: Integrating 3D and 2D Information Displays. In *IUI'03: Proceedings of the 8th International Conference on Intelligent User Interfaces*. ACM Press, 2003. pp. 166–173.
17. Sonnet, H., Carpendale, S., and Strothotte, T. Integrating Expanding Annotations with a 3D Explosion Probe. In *AVI '04: Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM Press, 2004. pp. 63–70.