LNCS 4073

Andreas Butz
Brian Fisher
Antonio Krüger
Patrick Olivier (Eds.)

Smart Graphics

6th International Symposium, SG 2006
Vancouver, Canada, July 2006
Proceedings

Springer

Andreas Butz Brian Fisher
Antonio Krüger Patrick Olivier (Eds.)

Smart Graphics

6th International Symposium, SG 2006
Vancouver, Canada, July 23-25, 2006
Proceedings

Springer

Volume Editors

Andreas Butz
Ludwig-Maximilians-Universität München
Institut für Informatik, LFE Medieninformatik
Amalienstrasse 17, 80333 München, Germany
E-mail: butz@ifi.lmu.de

Brian Fisher
Simon Fraser University at Surrey
13450 102 Ave, Surrey BC V3T 5X3, Canada
E-mail: fisher@cs.ubc.ca

Antonio Krüger
Westfälische Wilhelms-Universität
Robert-Koch-Str. 26-28, 48149 Münster, Germany
E-mail:antonio.krueger@uni.muenster.de

Patrick Olivier
University of Newcastle Upon Tyne
Information Research Institute
Newcastle upon Tyne NE1 7Ru, UK
E-mail: p.l.olivier@ncl.ac.uk

Library of Congress Control Number: 2006928506

CR Subject Classification (1998): I.3, I.2.10, I.2, I.4

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-540-36293-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-36293-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11795018 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4073

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3981

please contact your bookseller or Springer

Vol. 4073: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 263 pages. 2006.

Vol. 4072: M. Harders, G. Székely (Eds.), Biomedical Simulation. XI, 216 pages. 2006.

Vol. 4071: H. Sundaram, M. Naphade, J.R. Smith, Y. Rui (Eds.), Image and Video Retrieval. XII, 547 pages. 2006.

Vol. 4069: F.J. Perales, R.B. Fisher (Eds.), Articulated Motion and Deformable Objects. XV, 526 pages. 2006.

Vol. 4068: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), Conceptual Structures: Inspiration and Application. XI, 455 pages. 2006. (Sublibrary LNAI).

Vol. 4067: D. Thomas (Ed.), ECOOP 2006 – Object-Oriented Programming. XIV, 527 pages. 2006.

Vol. 4066: A. Rensink, J. Warmer (Eds.), Model Driven Architecture – Foundations and Applications. XII, 392 pages. 2006.

Vol. 4065: P. Perner (Ed.), Advances in Data Mining. XI, 592 pages. 2006. (Sublibrary LNAI).

Vol. 4064: R. Büschkes, P. Laskov (Eds.), Detection of Intrusions and Malware & Vulnerability Assessment. X, 195 pages. 2006.

Vol. 4063: I. Gorton, G.T. Heineman, I. Crnkovic, H.W. Schmidt, J.A. Stafford, C.A. Szyperski, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 394 pages. 2006.

Vol. 4060: K. Futatsugi, J.-P. Jouannaud, J. Meseguer (Eds.), Algebra, Meaning and Computation. XXXVIII, 643 pages. 2006.

Vol. 4059: L. Arge, R. Freivalds (Eds.), Algorithm Theory – SWAT 2006. XII, 436 pages. 2006.

Vol. 4058: L.M. Batten, R. Safavi-Naini (Eds.), Information Security and Privacy. XII, 446 pages. 2006.

Vol. 4057: J.P. W. Pluim, B. Likar, F.A. Gerritsen (Eds.), Biomedical Image Registration. XII, 324 pages. 2006.

Vol. 4056: P. Flocchini, L. Gąsieniec (Eds.), Structural Information and Communication Complexity. X, 357 pages. 2006.

Vol. 4055: J. Lee, J. Shim, S.-g. Lee, C. Bussler, S. Shim (Eds.), Data Engineering Issues in E-Commerce and Services. IX, 290 pages. 2006.

Vol. 4054: A. Horváth, M. Telek (Eds.), Formal Methods and Stochastic Models for Performance Evaluation. VIII, 239 pages. 2006.

Vol. 4053: M. Ikeda, K.D. Ashley, T.-W. Chan (Eds.), Intelligent Tutoring Systems. XXVI, 821 pages. 2006.

Vol. 4052: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, Part II. XXIV, 603 pages. 2006.

Vol. 4051: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, Part I. XXIII, 729 pages. 2006.

Vol. 4048: L. Goble, J.-J.C.. Meyer (Eds.), Deontic Logic and Artificial Normative Systems. X, 273 pages. 2006. (Sublibrary LNAI).

Vol. 4046: S.M. Astley, M. Brady, C. Rose, R. Zwiggelaar (Eds.), Digital Mammography. XVI, 654 pages. 2006.

Vol. 4045: D. Barker-Plummer, R. Cox, N. Swoboda (Eds.), Diagrammatic Representation and Inference. XII, 301 pages. 2006. (Sublibrary LNAI).

Vol. 4044: P. Abrahamsson, M. Marchesi, G. Succi (Eds.), Extreme Programming and Agile Processes in Software Engineering. XII, 230 pages. 2006.

Vol. 4043: A.S. Atzeni, A. Lioy (Eds.), Public Key Infrastructure. XI, 261 pages. 2006.

Vol. 4042: D. Bell, J. Hong (Eds.), Flexible and Efficient Information Handling. XVI, 296 pages. 2006.

Vol. 4041: S.-W. Cheng, C.K. Poon (Eds.), Algorithmic Aspects in Information and Management. XI, 395 pages. 2006.

Vol. 4040: R. Reulke, U. Eckardt, B. Flach, U. Knauer, K. Polthier (Eds.), Combinatorial Image Analysis. XII, 482 pages. 2006.

Vol. 4039: M. Morisio (Ed.), Reuse of Off-the-Shelf Components. XIII, 444 pages. 2006.

Vol. 4038: P. Ciancarini, H. Wiklicky (Eds.), Coordination Models and Languages. VIII, 299 pages. 2006.

Vol. 4037: R. Gorrieri, H. Wehrheim (Eds.), Formal Methods for Open Object-Based Distributed Systems. XVII, 474 pages. 2006.

Vol. 4036: O. H. Ibarra, Z. Dang (Eds.), Developments in Language Theory. XII, 456 pages. 2006.

Vol. 4035: T. Nishita, Q. Peng, H.-P. Seidel (Eds.), Advances in Computer Graphics. XX, 771 pages. 2006.

Vol. 4034: J. Münch, M. Vierimaa (Eds.), Product-Focused Software Process Improvement. XVII, 474 pages. 2006.

Vol. 4033: B. Stiller, P. Reichl, B. Tuffin (Eds.), Performability Has its Price. X, 103 pages. 2006.

Vol. 4032: O. Etzion, T. Kuflik, A. Motro (Eds.), Next Generation Information Technologies and Systems. XIII, 365 pages. 2006.

Vol. 4031: M. Ali, R. Dapoigny (Eds.), Innovations in Applied Artificial Intelligence. XXIII, 1353 pages. 2006. (Sublibrary LNAI).

Vol. 4029: L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J. Zurada (Eds.), Artificial Intelligence and Soft Computing – ICAISC 2006. XXI, 1235 pages. 2006. (Sublibrary LNAI).

Vol. 4027: H.L. Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreasen, H. Christiansen (Eds.), Flexible Query Answering Systems. XVIII, 714 pages. 2006. (Sublibrary LNAI).

Vol. 4026: P.B. Gibbons, T. Abdelzaher, J. Aspnes, R. Rao (Eds.), Distributed Computing in Sensor Systems. XIV, 566 pages. 2006.

Vol. 4025: F. Eliassen, A. Montresor (Eds.), Distributed Applications and Interoperable Systems. XI, 355 pages. 2006.

Vol. 4024: S. Donatelli, P. S. Thiagarajan (Eds.), Petri Nets and Other Models of Concurrency - ICATPN 2006. XI, 441 pages. 2006.

Vol. 4021: E. André, L. Dybkjær, W. Minker, H. Neumann, M. Weber (Eds.), Perception and Interactive Technologies. XI, 217 pages. 2006. (Sublibrary LNAI).

Vol. 4020: A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi (Eds.), RoboCup 2005: Robot Soccer World Cup IX. XVII, 727 pages. 2006. (Sublibrary LNAI).

Vol. 4019: M. Johnson, V. Vene (Eds.), Algebraic Methodology and Software Technology. XI, 389 pages. 2006.

Vol. 4018: V. Wade, H. Ashman, B. Smyth (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. XVI, 474 pages. 2006.

Vol. 4016: J.X. Yu, M. Kitsuregawa, H.V. Leong (Eds.), Advances in Web-Age Information Management. XVII, 606 pages. 2006.

Vol. 4014: T. Uustalu (Ed.), Mathematics of Program Construction. X, 455 pages. 2006.

Vol. 4013: L. Lamontagne, M. Marchand (Eds.), Advances in Artificial Intelligence. XIII, 564 pages. 2006. (Sublibrary LNAI).

Vol. 4012: T. Washio, A. Sakurai, K. Nakajima, H. Takeda, S. Tojo, M. Yokoo (Eds.), New Frontiers in Artificial Intelligence. XIII, 484 pages. 2006. (Sublibrary LNAI).

Vol. 4011: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications. XIX, 726 pages. 2006.

Vol. 4010: S. Dunne, B. Stoddart (Eds.), Unifying Theories of Programming. VIII, 257 pages. 2006.

Vol. 4009: M. Lewenstein, G. Valiente (Eds.), Combinatorial Pattern Matching. XII, 414 pages. 2006.

Vol. 4008: J.C. Augusto, C.D. Nugent (Eds.), Designing Smart Homes. XI, 183 pages. 2006. (Sublibrary LNAI).

Vol. 4007: C. Àlvarez, M. Serna (Eds.), Experimental Algorithms. XI, 329 pages. 2006.

Vol. 4006: L.M. Pinho, M. González Harbour (Eds.), Reliable Software Technologies – Ada-Europe 2006. XII, 241 pages. 2006.

Vol. 4005: G. Lugosi, H.U. Simon (Eds.), Learning Theory. XI, 656 pages. 2006. (Sublibrary LNAI).

Vol. 4004: S. Vaudenay (Ed.), Advances in Cryptology - EUROCRYPT 2006. XIV, 613 pages. 2006.

Vol. 4003: Y. Koucheryavy, J. Harju, V.B. Iversen (Eds.), Next Generation Teletraffic and Wired/Wireless Advanced Networking. XVI, 582 pages. 2006.

Vol. 4001: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering. XVI, 560 pages. 2006.

Vol. 3999: C. Kop, G. Fliedl, H.C. Mayr, E. Métais (Eds.), Natural Language Processing and Information Systems. XIII, 227 pages. 2006.

Vol. 3998: T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.), Algorithms and Complexity. XII, 394 pages. 2006.

Vol. 3997: W. Grieskamp, C. Weise (Eds.), Formal Approaches to Software Testing. XII, 219 pages. 2006.

Vol. 3996: A. Keller, J.-P. Martin-Flatin (Eds.), Self-Managed Networks, Systems, and Services. X, 185 pages. 2006.

Vol. 3995: G. Müller (Ed.), Emerging Trends in Information and Communication Security. XX, 524 pages. 2006.

Vol. 3994: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra, Computational Science – ICCS 2006, Part IV. XXXV, 1096 pages. 2006.

Vol. 3993: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra, Computational Science – ICCS 2006, Part III. XXXVI, 1136 pages. 2006.

Vol. 3992: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra, Computational Science – ICCS 2006, Part II. XXXV, 1122 pages. 2006.

Vol. 3991: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra, Computational Science – ICCS 2006, Part I. LXXXI, 1096 pages. 2006.

Vol. 3990: J. C. Beck, B.M. Smith (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 301 pages. 2006.

Vol. 3989: J. Zhou, M. Yung, F. Bao, Applied Cryptography and Network Security. XIV, 488 pages. 2006.

Vol. 3988: A. Beckmann, U. Berger, B. Löwe, J.V. Tucker (Eds.), Logical Approaches to Computational Barriers. XV, 608 pages. 2006.

Vol. 3987: M. Hazas, J. Krumm, T. Strang (Eds.), Location- and Context-Awareness. X, 289 pages. 2006.

Vol. 3986: K. Stølen, W.H. Winsborough, F. Martinelli, F. Massacci (Eds.), Trust Management. XIV, 474 pages. 2006.

Vol. 3984: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part V. XXV, 1045 pages. 2006.

Vol. 3983: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part IV. XXVI, 1191 pages. 2006.

Vol. 3982: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part III. XXV, 1243 pages. 2006.

Preface

The International Symposium on Smart Graphics 2006 was held during July 23–25, 2006, at the University of British Columbia in Vancouver, Canada. It was the seventh event in a series which originally started in 2000 as an AAAI Spring Symposium.

In response to the overwhelming success of the 2000 symposium, its organizers decided to turn it into a self-contained event. With the support of IBM, the first two International Symposia on Smart Graphics were held at the T.J. Watson Research Center in Hawthorne, New York, in 2001 and 2002. The 2003 symposium moved to the European Media Lab in Heidelberg. Since then the conference has alternated between North America and Europe. It was held at Banff Alberta Canada in 2004 and at the cloister Frauenwörth on the island of Frauenchiemsee in Germany in 2005.

The core idea behind these symposia is to bring together researchers and practitioners from the field of computer graphics, artificial intelligence, cognitive science, graphic design and the fine arts. Each of these disciplines contributes to what we mean by the term "Smart Graphics": the intelligent process of creating effective, expressive and esthetic graphical presentation. While artists and designers have been creating communicative graphics for centuries, artificial intelligence focuses on automating this process by means of the computer. While computer graphics provides the tools for creating graphical presentations in the first place, the cognitive sciences contribute the rules and models of perception necessary for the design of effective graphics. The exchange of ideas between these four disciplines has led to many exciting and fruitful discussions and the Smart Graphics symposia draw their liveliness from a spirit of open minds and the willingness to learn from and share with other disciplines

Many Smart Graphics symposia emphasize a particular aspect of the field in the call for papers. In a wrap-up session in 2005, workshop participants identified three key challenges for Smart Graphics that formed the basis for the 2006 workshop: (a) to understand human reasoning with visual representations, (b) in human decision support, to reconcile the complexity of problems that must be solved with the simplicity of representation and interaction that is desired by users, and (c) to build systems that can reason about and change their own graphical representations to meet the needs and abilities of their users and the nature of the information they present.

Accordingly this year's SG emphasized the "smart" in Smart Graphics. This includes human individual, group, and distributed cognition as well as artificial intelligence applied to the design and testing of graphically rich systems: smart design, smart systems, and systems for smart users. In order to facilitate interaction with the AI and Cogsci communities, we co-located SG with the

28th Annual Meeting of the Cognitive Science Society and the IEEE World Congress on Computational Intelligence.

We would like to thank all authors for the effort that went into their submissions, the Program Committee for their work in selecting and ordering contributions for the final program, and of course the participants who made Smart Graphics 2006 such a success.

Juli 2006

Andreas Butz
Brian Fisher
Antonio Krüger
Patrick Olivier

Organization

Organization Committee

Andreas Butz (University of Munich, Germany)
Brian Fisher (University of British Columbia, Canada)
Antonio Krüger (University of Münster, Germany)
Patrick Olivier (University of Newcastle Upon Tyne, UK)

Program Committee

Elisabeth André (University of Augsburg)
William Bares (Millsaps College)
Marc Cavazza (Teesside University)
Marc Christie (University of Nantes)
Sarah Diamond (Banff Centre)
Steven Feiner (Columbia University)
Sidney Fels (University of British Columbia)
Knut Hartmann (University of Magdeburg)
Rainer Malaka (University of Bremen)
Shigeru Owada (Sony Computer Science Laboratory, Tokyo)
W. Bradford Paley (Digital Image Design)
Bernhard Preim (University of Magdeburg)
Thomas Rist (University of Applied Sciences, Augsburg)
Stefan Schlechtweg (University of Magdeburg)
Lucia Terrenghi (University of Munich)
Sha Xin Wei (Concordia University)
Kay C. Wiese (Simon Fraser University)
Massimo Zancanaro (ITC-irst Trento)
Michelle Zhou (IBM T.J. Watson Research Center)

Supporting Institutions

The Smart Graphics Symposium 2005 was held in cooperation with Eurographics, AAAI and ACM Siggraph and the University of British Columbia.

Table of Contents

Intelligent Text Processing

Perceptive Systems

Smart Visualization

Visual Features, Sketching and Graphical Abstraction

Intelligent Image and Film Composing

Smart Interaction

Short Papers

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 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

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 1 – 12, 2006.

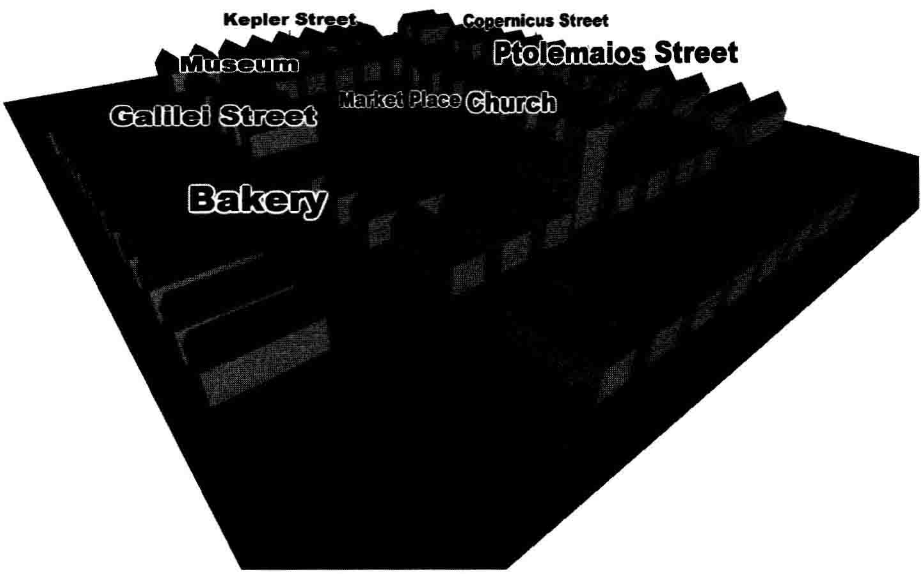

Fig. 1. Sample virtual landscape with dynamically placed annotations

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 import 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 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.

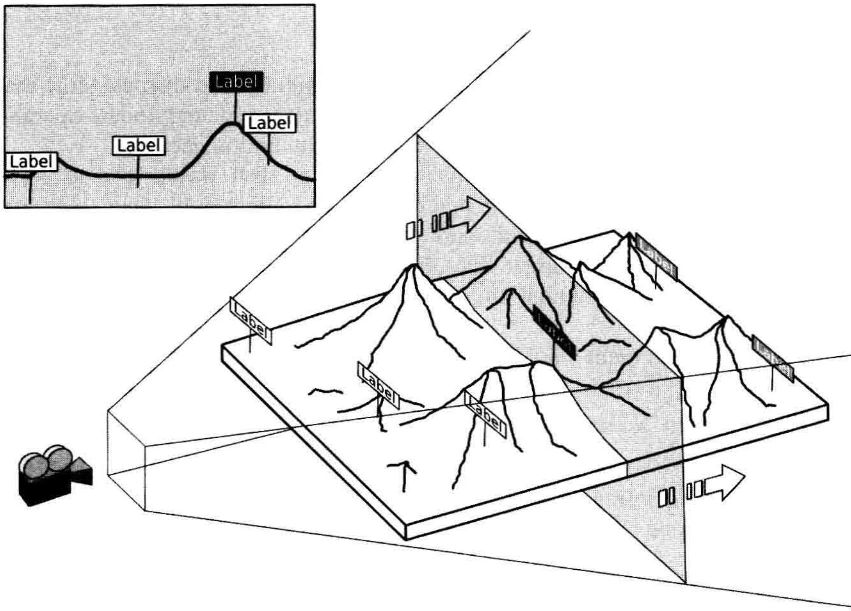

Fig. 2. Illustration of the placement strategy for virtual landscapes: Labels are processed in front-to-back order. Possible collisions are than resolved within their two-dimensional plane.

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.

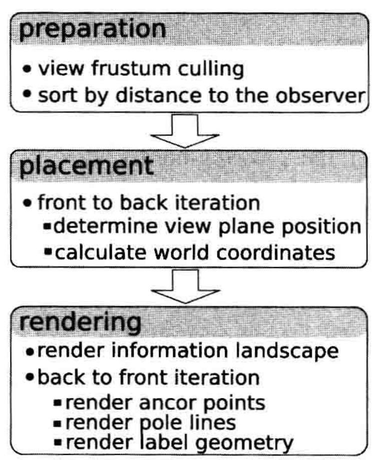

Fig. 3. Overview of the rendering process

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.