Appears in the IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 27-30 June 2004.

Automatically Designed 3D Environments for Intuitive Exploration

Nelson L. Chang and Amir Said
Hewlett-Packard Laboratories, 1501 Page Mill Road MS 1203, Palo Alto, CA 94304 USA
{nelson.chang | amir.said}@hp.com

Abstract

While popular as a visual interface, current interactive 3-D environments are often times painstakingly created by hand. This paper proposes interactive 3-D environments that are automatically designed on-the-fly based on various design rules, thereby separating layout and content. The resulting environments are compelling, customizable, and intuitive for information exploration. The interface also serves as an integrated framework for different types of rich media. The complete system for designing, constructing, and rendering the environments in real time is discussed and shown to have implications in a variety of applications including e-commerce.

1. Introduction

The advantages of rich graphical user interfaces are by now well known. For today's highly complex and visual data, interfaces will increasingly use three-dimensional (3-D) graphical or virtual environments that have been shown to be, for certain applications, extremely powerful, intuitive, and easy to use. Powerful yet affordable graphics hardware is enabling more users to produce and even come to expect captivating virtual environments. While more commonly found in gaming and entertainment industries, such environments are also beginning to populate the World Wide Web for applications like multiuser chat and collaborative shopping [2][6][7][5].

Despite their maturity and available design tools, virtual environments may require the painstaking manual process of creation, achieved through a combination of programming and artistic skills, taking days or even weeks of skilled artisanship. A different approach is required to address the automatic creation of content in large scale. The Internet has shown the need to produce content by the thousands or millions. For instance, it would be impractical for web pages for e-commerce showing different products to be created individually one at a time. Hence, it is necessary to employ templates and smart rules, based on the database's metadata, to automatically generate and customize to the user's preferences the required pages.

This paper applies these ideas to the automatic creation of virtual environments with the proposed VEDA architecture. Instead of conforming to a rigid blueprint defined by constraints from the real world, one has an enormous amount of freedom when creating virtual environments as a means for communication. Theoretically

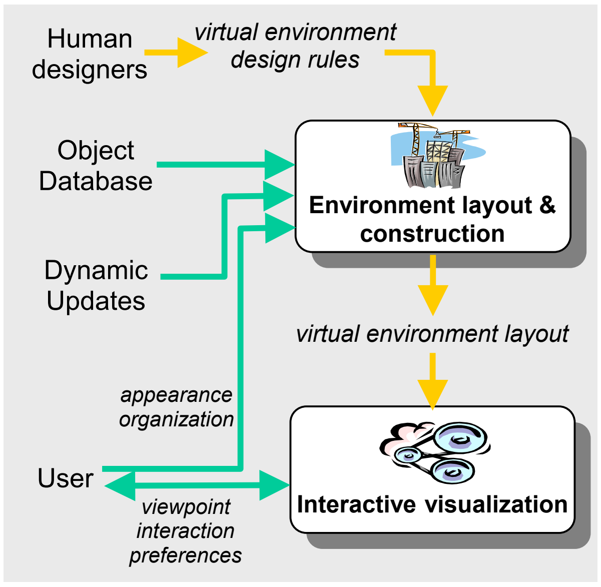

Human designers
virtual environment design rules
Object Database
Environment layout & construction
Dynamic Updates
virtual environment layout
appearance organization
User
viewpoint interaction preferences
Interactive visualization

Figure 1. The VEDA system architecture.

the proposed architecture creates not just a few environments from a slow and tedious manual process, but rather a seemingly infinite set of possible environments in real time. Moreover, the aesthetic component of the design can be integrated into the rules for environment creation, thus preserving artistic creativity.

2. Virtual environment design automation (VEDA)

Devising effective layouts automatically is both crucial and difficult. It is important for the user to be able to access massive databases without feeling overwhelmed. There needs to be some sort of organization to make the data accessible, clustering the data into meaningful groups based on metadata and organizing these groups according to some taxonomy. In addition, finding the optimal layout is usually NP-hard and needs heuristics to achieve an approximate solution in a reasonable time. Even so, such approaches may not be practical for an application that requires layout design in real time.

To address these issues, this paper proposes the virtual environment design automation (or VEDA) architecture diagrammed in Figure 1. At its core, VEDA starts with an

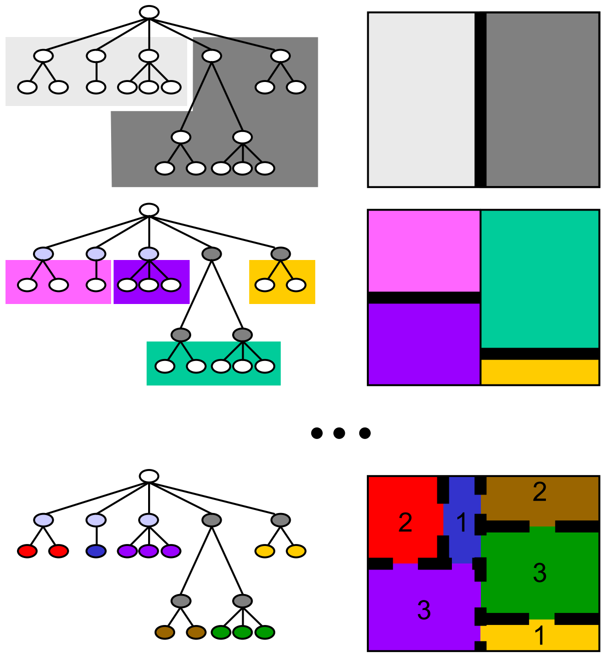


Figure 2. Evolution of recursive layout algorithm to derive rooms given layout tree.

object database and a set of design rules to automatically construct a visually rich 3-D environment.

In general, the database consists of a collection of objects, where each object has an appropriate visual representation (referred to as an image object) and a number of attributes in the form of associated metatdata. The attributes define various characteristics of a given object where not all attributes have to be specified for every object. Furthermore, each object may have additional attributes that link to one or more related rich media (e.g. text, URLs, audio, images, videos, "rotatable" 3-D objects, and 3-D models).

The layouts produced by VEDA are governed by three different categories of design rules:

- *Environment design rules* for the physical structure
- *Appearance design rules* for style and architecture
- *Database-specific layout rules* as functions based on object attributes and metadata.

Ultimately, these rules come from professional designers, such as artists and interior designers, who specify characteristics of the environment to make it aesthetically pleasing and appeal to human perception cues. Currently, VEDA uses XML-based rules that specify an environment comprised of adjacent rectangular partitioned rooms such that nearby rooms are somewhat related in organization. Each room is constrained to be spacious enough to accommodate all associated image objects and reduce the occurrence of "skinny" rooms.

Once the environment has been designed and constructed, VEDA renders it for the user to interact with it and effortlessly explore the object database. The rendering algorithm addresses many issues including

- Rendering at interactive rates
- Handling predominantly textured environments (i.e. image objects, texture maps) with limited geometry
- Overcoming limited texture memory in hardware
- Adding effects for improved realism
- Providing intuitive visual cues for exploration

Typical rendering algorithms for games and VRML-based environments may not be applicable since they assume distinct and geometry-centric worlds with highly reusable textures.

3. Details of VEDA

The entire VEDA system architecture consists of two primary components briefly summarized in the upcoming subsections and described in more detail in [1]. The first component generates a description of the environment layout given various design rules and object information. The second component uses this description to provide a rich interactive visualization of the environment for the user.

3.1. Environment layout and construction

The first aspect of automatic layout carves the environment into partitioned areas (i.e. rooms) and assigns the appropriate subset of the database to each area. A layout tree is first constructed to encapsulate the chosen design rules applied to the object database. In particular, the leaf nodes of the layout tree point directly to the appropriate objects in the database. VEDA then partitions the environment into meaningful and spatially related rooms using a top-down algorithm similar to Johnson and Shneiderman's tree-map algorithm [3]. As shown in Figure 2, the proposed algorithm starts with a given rectangular floorplan to specify the overall dimensions of the environment. For each level of the layout tree, the algorithm divides the children nodes of the current parent node into approximately equal subtrees based on total number of leaf nodes. Finally, VEDA performs recursive binary space partitioning on the environment with guillotine cuts that minimize the appearance of "skinny" rooms. At this point, the approach produces an environment subdivided into adjacent rectangular rooms, each with a subcollection of objects associated with it.

With object clusters formed, the algorithm then distributes the image objects assigned to every room so as to maximize their dimensions and balance their distribution within the room while preserving their aspect ratios. For every room, the image objects are split into successive subsequences by minimizing the difference of cumulative sum of aspect ratios and cumulative sum of wall widths. The algorithm also automatically selects the appropriate number of rows of image objects to maximize the overall size of the images constrained within the wall's dimensions. In the end, the image objects are placed at specific locations on the walls with the chosen rendering style.

The proposed approach has numerous benefits. The top-down layout algorithm determines the number and dimensions of the walls defining every room, and it is computationally linear in the number of nodes in the layout tree. It ensures that the floor area of each room is proportional to the number of objects corresponding to that room. Hence, at a glance, one can compare the number of objects across rooms. The resulting layouts enforce spatial proximity of rooms corresponding to nodes at the same level in the layout tree. This behavior follows the Gestalt proximity principle that items that are spatially near one another are likely be related and vice versa [4]. In addition, this approach is flexible enough to enable multiple rules to be applied to different portions of the layout tree, thereby providing customized organization of the data.

3.2. Interactive visualization

VEDA performs many speed enhancements to ensure real-time rates during rendering. It performs portal culling (i.e. recursive visibility check of adjacent rooms through the portals) to substantially reduce the number of graphical objects in the hierarchically ordered list to the candidates that are actually visible to the user. With limited texture memory, image objects are automatically cached and scaled to the appropriate resolutions based on the user's viewpoint to maintain interactive rates. Additional rendering elements, including efficient lighting and shadow effects, further enhance the user's overall experience.

In addition to rendering issues, it is important to make the 3-D environment user-friendly and provide a lot of useful cues. Numerous user interface elements are added to improve usability. For example, the system eliminates rolling motion and ties tilting motion with changes in height to lessen the confusion with navigating in free space. Automatically rotating 3-D signage is provided throughout the environment to emulate signs available at many department stores and to improve visibility. The ceilings of each room are rendered in a translucent mutable color to enable visibility into every room and to give cues about how the rooms are implicitly grouped.

There are also numerous modes of interaction throughout VEDA. A brief description of any image object is instantly obtained simply by hovering the mouse cursor over the desired object. The user can click on any image object of interest and be automatically moved toward that object using Dijkstras shortest path algorithm and linear interpolation of viewpoints. Alternatively, the user can click and activate the associated rich media links to an image object, thereby starting the integrated playback and simulating interaction. For example, the user can launch a corresponding web page, start playing a related audio or video clip, view a "rotatable" 3-D object as a digital hologram embedded in the environment, toggle the visibility of 3-D models, and so forth. The user can also immediately redesign the environment based on a different set of rules or style.

4. Experimental results

Instead of using geometry-centric VRML or Web3D, the current system consists of customized layout and visualization software written in C++ for the Windows platform using OpenGL and DirectX 8.1. VEDA needs only a moderately fast computer with a reasonably powerful graphics card to generate compelling layouts in only a few seconds. The results shown in this section come from a 800 MHz Pentium III with an GeForce3 graphics card with 64 MB of texture memory.

The object databases with associated metadata are described in XML with 2-D image objects; VEDA could just as easily work with typical relational databases or 3-D object models. In addition to the three example databases presented in this section, VEDA has been tested with many different databases, including a 4000+ high-resolution image database and even a collection of music tracks for audio browsing. All of the databases run at interactive rates, giving the user a responsive and visually rich interface.

While the results are much more impressive when directly interacting with the system, the screenshots in Figure 3 give an adequate representation. Figures 3(a) and (b) show screenshots for a database consisting of 250 movies and their associated metadata organized automatically by genre and director, respectively. Many of the user interface elements described above are clearly apparent, including naturally partitioned areas, compelling textures and lighting effects, optional metadata and map dialog windows, and automatically rotating signage. Rich media, such as the 3-D model chair and the movie trailer playing in the background, can be activated and integrated directly into the environment, creating a consistent and immersive atmosphere.

Figure 3(c) shows a database consisting of a subset of HP products laid out according to the product taxonomy. The layout algorithm naturally places rooms of similar content near one another to form pseudo-"departments." It presents the view as the user "flies" above the environment.

Figure 3(d) is a collection of digital photos arranged by subject distance from the camera. Using the EXIF metadata directly, the up-close macro shots and distant landscape shots are automatically separated and clustered appropriately without any additional work. The figure also highlights the flexibility of VEDA to allow instant changes in appearance.

5. Conclusions

VEDA is a powerful 3-D interface for information exploration and interaction. Unlike traditional 3-D interfaces, the layouts are automatically organized and designed in real time based on various prespecified rules and user preferences. It provides a pleasant, intuitive, responsive, and customizable environment that gives users a lot of freedom without feeling overwhelmed. Furthermore, VEDA employs efficient rendering algorithms for real-time interaction. It also forms a natural framework to seamlessly integrate rich media together into a single coherent environment. For e-commerce applications, this improved interface marries the benefits of traditional brick-and-mortar stores with the convenience of their online counterparts. Moreover, such an interface has implications beyond e-commerce, in areas such as education, collaboration, and personal multimedia management.

The presented work serves as an important first step toward the vision of automatic creation of virtual environments and there are many exciting directions one can take. A more thorough study of design rules will lead to more aesthetically pleasing virtual environments. It is also important to consider semi-automated tools to simplify layout and environment design. Scaling the system to accommodate very large databases requires further examination. Efficient representation and distribution of the environments will allow for multi-user collaboration and seamless rich media communication.

6. References

[1] N. L. Chang and A. Said, "Automatically Designed 3-D Environments for Intuitive Browsing and Discovery," *HP Labs Technical Report HPL-2003-92*, 28 April 2003.

[2] R. Hawkes and M. J. Wray, "LivingSpace: A Living Worlds Implementation using an Event-based Architecture," *HP Labs Tech. Rep.*, HPL-98-181, 1998.

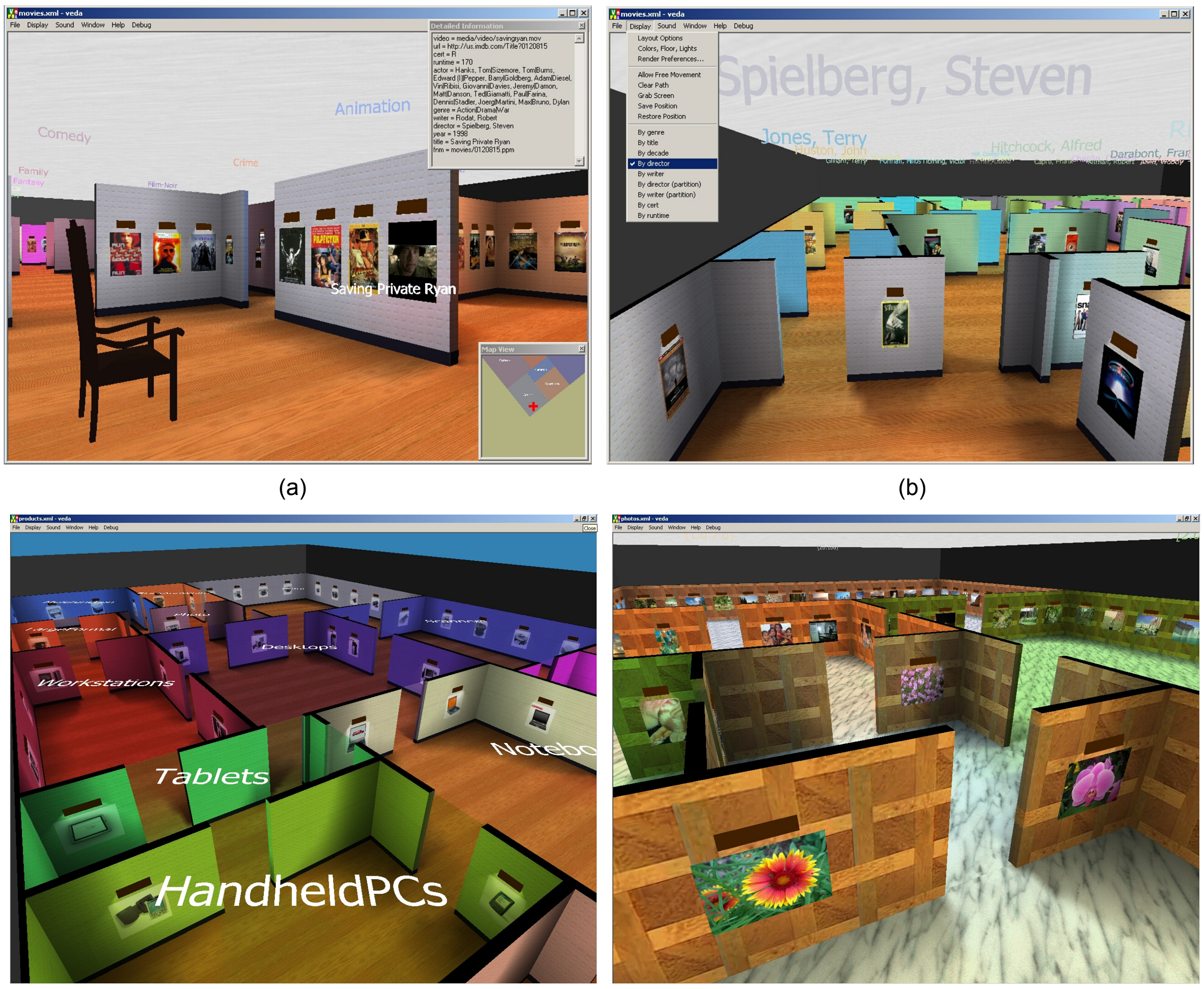
[3] B. Johnson and B. Shneiderman, "Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," *Proc. IEEE Vis, Conf.'91*, pp. 284--291.

[4] K. Mullet and D. Sano, *Designing Visual Interfaces: Communication Oriented Techniques*, Englewood Cliffs, NJ, Prentice Hall, 1995.

[5] P. G. Selfridge and T. Kirk, "Cospace: Combining Web Browsing and Dynamically-Generated, 3D, Multiuser Enviroments," *Intelligence*, 10(1), pp. 24—32, 1999.

[6] "Virtual Worlds for E-Commerce," *Blaxxun Interactive White Paper (www.blaxxun.com)*, 2001.

[7] "Welcome to the Home of the 3D Internet, Virtual Reality, and Community Chat," *Active Worlds web site (www.activeworlds.com)*, November 2002.



Detailed Information

video = media/video/savingryan.mov
url = http://us.imdb.com/Title?0120815
cert = R
runtime = 170
actor = Hanks, Tom|Sizemore, Tom|Burns, Edward (I)|Pepper, Barry|Goldberg, Adam|Diesel, Vin|Ribisi, Giovanni|Davies, Jeremy|Damon, Matt|Danson, Ted|Giamatti, Paul|Farina, Dennis|Stadler, Joerg|Martini, Max|Bruno, Dylan
genre = Action|Drama|War
writer = Rodat, Robert
director = Spielberg, Steven
year = 1998
title = Saving Private Ryan
fnm = movies/0120815.ppm

(a) (b)

(c) (d)

Figure 3. Environments automatically designed by VEDA: movie collection organized by (a) genre and (b) director; (c) HP products organized by product taxonomy; (d) digital photos organized by subject distance.