This paper presents an Internet-based kiosk displaying the campus of the University of Hong Kong in a virtual reality environment. The objective is to construct an online guided-tour system for the campus and provide visitor information. The virtual campus indeed can be accessible through any Java-enabled WWW browser with VRML capability in addition to the kiosks. An automatic way-finding mechanism has also been implemented to help visitors to look up places within the campus. We will discuss the design of the environment, the construction of the 3D campus model, the techniques used to achieve reasonable navigation performance even on low-end systems, and the algorithm of the way-finding mechanism. The algorithm is based on graph theory to compute the shortest paths between any two locations inside the campus with various intelligent constraints.
purpose. With the level-of-detail node, a particular building can be specified in different forms with different complexities. The closer it is to a building, a more complex model will be employed to reveal its detail. Similar the further apart, the lesser complex model is used so as to increase the 3D rendering speed.

The first task of the project is to outline the hill-sided landscape of our campus. It is then applied with proper greenery texture (Figure 1).

The rest of the campus buildings are constructed directly with Cosmo World running on an Silicon Graphics machines. The tool allows construction and editing of geometries, defining texture mapping, importing other virtual scenes, and generating animations. We can also define object behavior inside a VRML world by programming VRML scripts.

The painting texture covering each building is derived from high-quality photos, with possible modification to take care the perspective problems. At this stage, for certain buildings such as the Main Library, it is necessary to apply the level-of-detail mechanism by creating several models with increasing complexity. After importing all the buildings into the landscape with Cosmo World, the 3D model of the campus is known to be completed as in Figure 2.
Interface Design

The user interface of the system consists of two frames, with the left showing the 3D campus site and the right depicting a corresponding 2D map and providing information for a highlighted building (Figure 3).

Information on each building can be obtained by selecting its corresponding geometry on the 2D map or the Info icon on the 3D campus model-viewing screen.

![Figure 3. The User Interface](image)

Basically, a user can navigate freely around the campus with the standard command dashboard provided by the Cosmo Player plug-in. However, in order to perform a faster navigation and enjoy the advantage of guided tour, navigation can also be achieved by a set of viewpoints which can be changed dynamically depending on the visiting scene and the user’s request. Every viewpoint has its own menu system which in general consists of the following icons:

Information  
Aeroplane  
Tour  
Lego  
Navigation

The function of each icon is explained below:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>to provide detail information on current building</td>
</tr>
<tr>
<td>Aeroplane</td>
<td>to take an aerial view of current building.</td>
</tr>
<tr>
<td>Tour</td>
<td>to subscribe the “round the campus” trip.</td>
</tr>
</tbody>
</table>
to follow this tour guide to walk around the building.

Following navigation icon to follow the link to the next waypoint and by following successive navigation icon, final destination is reached.

On each viewpoint, the user can click the menu icon to perform different actions. Detail information of the current building will be shown on the lower right frame of the browser. The Navigation icon can be used in conjunction with the way-finder method to guide the user to reach a destination.

The Way-Finder

In addition to 3D visualization, our system also includes a way-finder mechanism implemented in the Javascript language. The way-finder is an extension of a conventional shortest path algorithm to help locating any particular building by calculating the most convenient walking path. The extension considers 1) user’s preference, 2) outdoor weather environment, 3) accessibility of a path which may change due to contingency events, and 4) intermediate drop-off points.

To simplify the navigation, there are convenient waypoints defined throughout the campus. When a user clicks on the Navigation icon, his viewpoint will change gradually and he will be brought to another waypoint referenced by another Navigation icon. On successive clicking, he can travel throughout the campus at finger tips.

The important task in controlling a new waypoint by each successive navigation is performed by the extended shortest path algorithm. By first choosing the target building, the algorithm comprehensively calculates the shortest paths from all viewpoints to the destination. Therefore, even if a user changes his/her mind and requests for a detour, the final target will always be aimed at.

The first step in the algorithm is to determine all possible paths to a destination using a non-directed graph representation of the campus. Every branch of the graph represents a possible branch of a path, and every node represents a specific reachable location. It is then a standard practice in deducing most convenient paths (Figure 4).

Figure 4. The graph constructed based on the campus map
To introduce the notion of “accessibility” to a path, every branch shown on the graph is weighted so as to indicate its preference index. The lower the weight of a branch, the better it is to walk through it. The most important extension of our algorithm is that the weight is indeed a linear combination of user preference (e.g. stair, elevator, or tunnel), weather status, or path condition etc. All these factors may liable to change over time. This is particularly useful if we want to reflect the genuine situation. For example, if it is raining, an indoor path rather than the shortest one is definitely preferable. Or if a certain path inside the campus is under construction, we can even delete it temporarily from the graph altogether so that it will never be chosen until reopened.

To change the graph used by the way-finder, we make use of the Live-Connect, a programming interface that enable Javascript to Java communication within the same or different page frames within the browser. The Java applet could, in turn, communicate with the server to obtain an updated graph in real time.

Figure 5. A path is found in the campus

Conclusion

In this paper, we have discussed a virtual guided-tour system for the University of Hong Kong. It provides a 3D view of the main campus so that a visitor can develop a feeling of “being there”. Information about the buildings is also conveniently campus. Users can also raise question like “How to go there?” and let the system perform the guidance in the most convenient way.

Up to now, the major criticism in VRML development over Internet is still centered around performance issues. In order that low-end computer systems can also take the advantage, we embed various complexity levels of the same model and selectively change one level to another depending on viewing distant. We expect that high performance and yet low-cost graphics accelerated card with 3D capabilities will become available very soon to alleviate the problem.

References

David Kurmann and Maia Engeli : 1996, Modeling Virtual Space in Architecture
Adrian Dobson, Teaching Architectural Composition Through the Medium Of Virtual Reality Modeling
Aleksander Asanowicz, 1997 Computer - Tools vs. Medium