EUROGRAPHICS 2002 / I. Navazo Alvaro and Ph. Slusallek
(Guest Editors)

# Concise Tour to the Virtual Old Prague

Jiri Zara

Department of Computer Science & Engineering, Czech Technical University in Prague, Czech Republic

**Abstract**

*This paper describes a structure and a user interface of the Internet based 3D presentation of existing historical part of city Prague. The application offers a smooth tour to selected regions of the real city, utilizing number of standard and modern technologies including VRML, Java, and PHP. The architecture of the whole virtual model is presented together with several advanced and specific features like dynamic loading und unloading objects or newly proposed LOD for urban scenes. This running web application seems to be currently unique in terms of complexity, functionality, and scaleability for further growth. The paper introduces methodology suitable for immediate creation of arbitrary virtual city according to a real one.*

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism---virtual city

## 1. Introduction

The virtual reality technology is currently not considered as something terrific and super as it was called ten years ago. Real-time and interactive presentation of virtual environment has become a common practice in many applications. Thanks to the progress in graphics hardware and network technologies, virtual reality (VR) is available to almost all people connected to the Internet.

With respect to that fact, one can be unpleasantly surprised by the lack of large virtual models of existing real places and objects. It is especially true for the case of virtual cities on the Internet.

In this paper, we present the results of the Virtual Old Prague (VOP) project that could become one of the first serious steps on the road from real to virtual cities. Formerly student project has been successfully finished in 2001, then published on the web and made available to Internet users. The VOP project [7] is the web application that allows walking through a virtual city stored in a remote database. The data are progressively transferred from the server according to a visitor's position. A high frame rate is achieved by rendering visible city parts only.

Although primarily concentrated on functionality of the final VOP application, couple of interesting ideas and valuable solutions for other authors/creators of virtual cities are introduced here as well.

The whole project of Virtual Old Prague follows the standard practice for creation of virtual representation of real objects. The common scheme consists of the following steps: A. Capturing digital images, B. Image processing, C. Reconstruction of 3D models, D. Storing and structuring large virtual space, E. Web based presentation. Beside VR part, the project also maintains other media like hypertext and images. Due to the limited space, we describe first two steps very briefly. More attention is given to 3D issues like LOD for urban models, client/server VR data exchange, and efficiency of rendering.

The paper is arranged as follows. Section 2 describes basic approaches when presenting real cities in a virtual form. Section 3 concentrates on creation of house models and their arrangement into logical units with the aim to ensure high speed of rendering. Functions and user interface of the final application are described in Section 4. Measurements obtained from real operation with the presented application are summarized in Section 5, advantages and disadvantages of implemented methods are discussed. Section 6 concludes the paper.

## 2. Virtual Cities on Internet

The possibility to present architectural attractions and other urban objects to Internet users is a challenge for web designers all around the world. Currently almost all bigger cities have their own web presentation showing maps, photos, panoramic views (e.g. based on QuickTime VR technology [6]) or 3D buildings (described by VRML [8] or other proprietary tools [11]). When looking at VR worlds only, we can find number of attempts to visualize either historical centers or interesting parts of cities – let us name at least a few representatives like Paris [1], Glasgow [2], New Orleans [3], Sydney [4] or Toronto [5]. Most of those VR models consist of one big chunk of geometrical and texture data, at most divided into several files. The rare exception is Paris [1] consisting of several VR segments connected by hyperlinks and proximity sensors. Continuous walk-through is not possible – when a visitor leaves examined part of a city, all previous data are replaced by the model of a new part.

Well-designed structure/hierarchy of urban model is actually the key issue for efficient presentation at end-user computer, both from the view of data transfer and real-time rendering. Since the size of a city model ranges from hundreds of kilobytes up to tens of megabytes per city district, 3D data have to be arranged into small packets and sent to a user depending on his/her current position [15]. As user walks through the virtual town, new packets should be read in advance and similarly unnecessary data have to be released from memory. When the city is subdivided into small disjoints cells, a visibility preprocessing substantially helps to select the proper data to be sent to a user.

The visibility computations/culling have been studied for a long-time and the theory is well-elaborated [10]. Spatial structuring is even supported by VRML extensions, though oriented toward GIS (GeoVRML [13]) rather than urban models. Level of details (LOD) techniques that save a lot of computations during rendering are also well investigated [12]. Several speed-up techniques suitable especially for urban environments incorporate image caching [14] and dynamically generated impostors [9].

In spite of those facts, we have not found any practical implementation of web based virtual city utilizing those theoretical principles. The reason is likely in the difference between real-time data processing performed on single computer and presentation distributed over the Internet. In the first case the rendering processes have an access to all data, thus the optimization for the sake of speed and image quality can be done directly and efficiently. The second case, a web based presentation, must fight against many limitations – data are transferred from the server to a client computer by relatively small chunks; client computers have not an access to global structure of the city, thus they utilize information stored locally in already downloaded (self organized) city data; the communication between a client and a database on a server is generally very simple and state information is limited to few parameters transferred by http protocol; the server has to provide data for large number of clients concurrently, thus it cannot perform any additional demanding computations to optimize rendering etc.

By our best knowledge the Virtual Old Prague project is currently the only virtual city on the Internet where the theory has led to running application offering smooth walking through potentially unlimited three-dimensional environment [17] with acceptable quality of rendering.

### 2.1. Complexity of urban presentations

The reason why large virtual cities are still very rare on the Internet lies also in the complexity of information requested. The real-time presentation of three-dimensional data is unfortunately only one requirement and perhaps not the most important. If the virtual city serves for tourist information, cultural presentation, advertisement, and many other purposes, the target web application should match up to the following criteria:

- *Synchronized navigation*. Most people still prefer to navigate using 2D city maps. Such a map should be always a part of the application. A user's position in 3D virtual space can be synchronously shown on 2D map. On the other hand, a user's selection from a sensitive 2D map should lead to a jump into corresponding viewpoint in 3D space.
- *Description*. Additional textual, image and audio information connected to places investigated by a user should automatically enrich the experience in the virtual reality.
- *Hyperlinks*. Thanks to the web, people have become familiar with the principle click-and-go. They expect to get further information when clicking to links in hypertext; the same is true for 3D objects. All models of shops, exhibitions, offices and other important places in virtual city should be sensitive and connected to appropriate web pages or teleports in VR.
- *Guided tours*. Some people prefer passive watching a presentation of virtual city with the possibility to stop/restart the guided tour or switch to independent examination of given virtual space. This can be achieved using animated viewpoint in VR.
- *Animations*. Although animations are definitely not an ultimate request for virtual cities, they enhance the user's experience a lot. Since the virtual city is usually static, without people, cars, pigeons etc., animations freshen up that rigid urban environment. Thus fountains with splashed water, blinking colored signs, waving vendors and other attractions are highly welcome. A special way to make a virtual city alive is to design the application as multi-user. This is the case of Paris [1].
- *Search functions*. Web users frequently use search machines to get specific information. In case of virtual cities, search automaton should return not only list of information found, but offer an automatic navigation from current to target place(s) in 3D environment.
- *Help subsystem*. A complex application always requires well-prepared system of hints, advices, and examples. This is extremely important for virtual cities, because most people are still not familiar with VR and they tend to consider it as an experimental environment for young game players, not as a real tool for getting useful information and experience.

- *Flexibility*. Users should have possibility to customize user interface (switch on/off some information windows), tune the presentation with respect to the network bandwidth and a power of their computer.

The list of functions for the virtual city is unfortunately too complex to be easily and fully implemented. Moreover, every city is a subject of changes – new buildings are constructed, offices and shops are moved, hyperlinks are altered. The long-term maintenance can be as difficult as the initial creation of the system.

When the Virtual Old Prague project was started, we were acquainted with all the range of possible requirement listed above, but the administration of 3D data was our main goal. That is why the VR part of the project is quite well tuned, while other parts still not.

## 3. Modeling Virtual Old Prague

The whole process of modeling virtual buildings typically consists of several steps. The support from GIS is welcome because it helps to set main distances and altitudes properly, but namely photos of real buildings are finally the main source of realistically looking data. A number of methods for 3D reconstruction from images have been discovered in the field of image processing and hopefully some of them could be directly used for reconstruction of buildings, because known constraints could make such a process robust (horizontal walls, perpendicular window frames, etc.). Although we do believe that automatically performed 3D reconstruction is the future for modeling virtual worlds based on real scenes, we have not found any available practical tool for the VOP project and we have chosen another approach for the sake of simplicity.

When we studied various existing cities on the web, we have noticed that geometrically perfect and detailed 3D models of houses are not the main request for the nice and convincing user's experience. Indeed, textures and colors have the main importance, while geometrical details remain almost not perceived, although they are highly important for architects and historians. This observation is not new – the well-known QuickTime VR technology [6] is a proof that images can perfectly replace the 3rd dimension and classical geometry when modeling real world. Based on that fact we have decided to build the virtual city with simple geometry but special management of textures.

A house in VOP project is represented by set of facades that are visible from the streets. A simple roof modeled by few polygons can cover top of facades, but it can be omitted in many cases. People walking through the city are mostly influenced by facades and not by roofs. Since the façades carry the most visually important information, our approach could be called *façade-based virtual city*. The following subsection introduces hierarchical description of a façade.

### 3.1. Urban LOD

We have designed and implemented four-step hierarchy for the façade of a real house. Firstly, we have to prepare one image covering the whole façade. All obstacles like cars, street lamps, trees, kiosks, people, and dogs have to be removed. Instead of automatic retouching utilizing set of photos from different viewpoints, we made a good experience with common raster image editors like Adobe PhotoShop or Corel PhotoPaint, where such changes were made by hand. Similarly we removed perspective distortions, combined several images of the same facade into a big one and tuned the contrast.

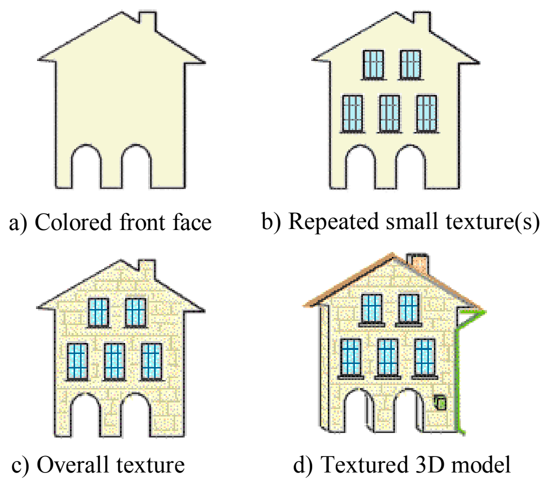


**Figure 1:** *Newly proposed urban LODs for virtual houses. The simplest representation consists of one polygon, two intermediate levels use textures and the best model is fully three-dimensional.*

Well-prepared image of a façade is the key for the successful subsequent process called *urban LOD*. Each façade is represented by four models suitable for different purposes (Figure 1). The first level is the simplest one and it consists of one planar polygon defined by the outline of the façade. An author should pick one solid color filling the polygonal shape and representing a color of the house as seen from the big distance.

The second representation suits very well for fast visualization when users do not want to wait too long for large texture data. Here the polygonal façade is enhanced by small textures representing visually most important parts of a house – typically windows, doors, and paintings on a wall. Since those textures are usually very small and even repeatedly placed on the façade, they are transferred very quickly from the server to a user's client computer, thus increasing the ratio between the visual quality and time delay when walking through a large virtual city. The visual effect based on utilizing small textures placed on a big polygonal shape filled by a solid color is surprisingly high as seen in Figure 2, where one of the houses is rendered in the 2nd urban LOD, while the rest is represented by the next level – full textured façades.

The high quality texture image mapped on the polygonal façade in the 3rd level replaces all previously used small textures. Here the proposed urban LOD methodology

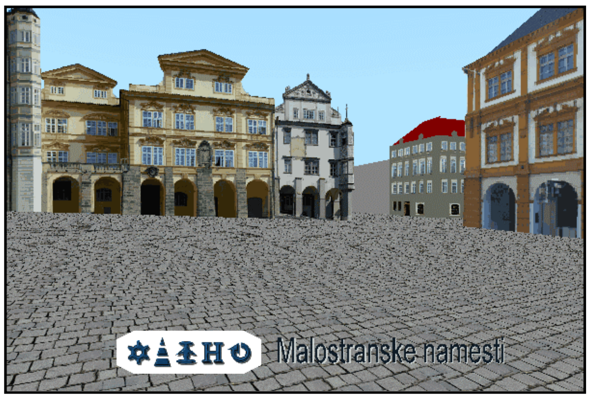
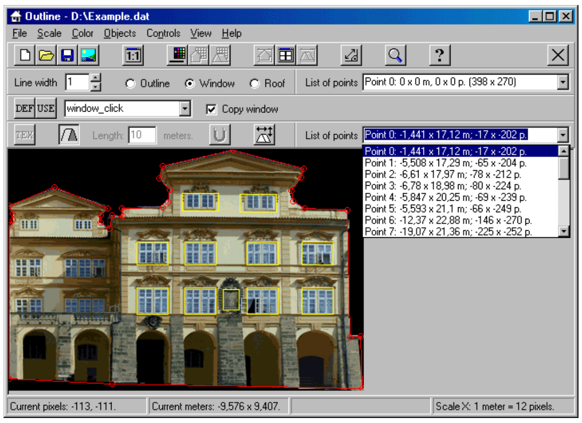


**Figure 2:** *Snapshot showing full textured houses and a house with textured windows only (upper right corner).*

shows certain redundancy, because previously loaded textures for windows and other objects are not further utilized. This is the main difference from common LOD algorithms based on the mesh simplification [12]. On the other hand, texture images used in the 2nd urban LOD are small not only in the resolution, but also in the number of colors used. While the full texture image for the 3rd urban LOD is usually true color, sixteen colors is reasonable upper limit for small textures thus their size is really tiny. Real sizes for various urban LODs are presented in Section 5.

The last, 4th urban LOD consists of 3D model with possibly complex (not only planar) geometry and textures. It is used when a visitor looks at the house from very small distance. It should be stressed that almost none of houses in the VOP project was modeled with the 4th urban LOD, because the previous levels, namely the 2nd and the 3rd one, were visually impressive enough. This has been true also for old buildings containing a lot of architectural details (Figure 2).

The urban LOD approach is general enough to represent very large buildings where the width is extremely big. Such long façade can be maintained as several individual facades placed in a row.



```
Line width 1    ○ Outline  ◉ Window  ○ Roof    List of points  Point 0: 0 x 0 m, 0 x 0 p. (398 x 270)
window_click    ☑ Copy window
Length: 10 meters.    List of points  Point 0: -1,441 x 17,12 m; -17 x -202 p.

Point 0: -1,441 x 17,12 m; -17 x -202 p.
Point 1: -5,508 x 17,29 m; -65 x -204 p.
Point 2: -6,61 x 17,97 m; -78 x -212 p.
Point 3: -6,78 x 18,98 m; -80 x -224 p.
Point 4: -5,847 x 20,25 m; -69 x -239 p.
Point 5: -5,593 x 21,1 m; -66 x -249 p.
Point 6: -12,37 x 22,88 m; -146 x -270 p.
Point 7: -19,07 x 21,36 m; -225 x -252 p.

Current pixels: -113, -111.   Current meters: -9,576 x 9,407.   Scale X: 1 meter = 12 pixels.
```

**Figure 3:** *Snapshot of Outline tool where polygonal facade, small textures, and full texture are prepared.*

We have developed a special software tool *Outline* for modeling first three urban levels of details (Figure 3). The input required is a cleaned-up ortho photo image of one façade, an altitude of the ground and a height and a width of the real façade. The program allows the author to define a polygon representing the façade, to specify visually important objects like windows and define their repetitive arrangement. Such data are finally exported in a form of VRML file with LOD and IndexedFaceSet nodes plus associated PNG texture files. The Outline program is currently freely available.

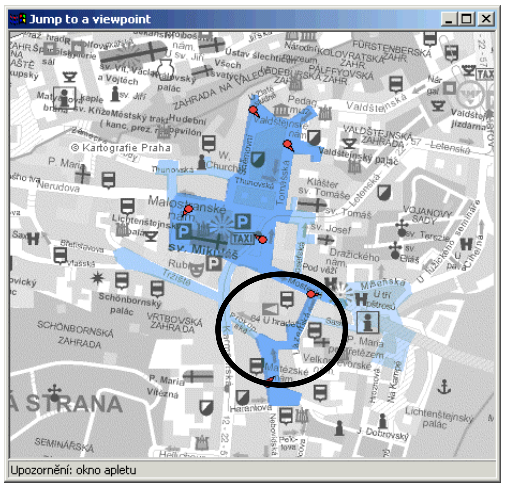
## 3.2. Data arrangement

Hierarchical representation of facades described in previous section is definitely not sufficient enough for fast rendering. Since virtual cities consist of hundreds of streets and thousands of houses, the whole city have to be subdivided into small areas with the aim to render only visible areas from the given viewpoint. The visibility preprocessing is highly useful method and works very well when users walk on the streets only and cannot see more than a few buildings in a close neighborhood. A simple 2D area visibility can be then used to preprocess city data. The obvious drawback is the limitation to planar city layout – it should be forbidden for users to visit hills or towers from which the large city area is visible. Big panoramic or aerial views slow down the rendering (because of big amount of presented data) and decrease the quality of rendered image in the case that houses are simplified to non-textured walls, even without roofs. One possible solution is using pre-computed impostors for such specific “hill” locations.

The geometry of Virtual Old Prague is structured into disjoint but connected cells (or sectors) as seen in Figure 4. Each cell has a polygonal shape - its borders are either house facades or portals to other cells. Standalone virtual objects like statues or fountains can be placed inside a cell as well. One street is usually covered by a row of connected cells, also bigger squares should be partitioned into several cells as practical experiments showed us. A database stores the following information for each cell:

- *Geometrical description* for each façade (altitude and orientation, link to external VRML file with urban LOD definition, etc.) and other virtual, possibly interactive objects.
- *Sensitive portals* to neighboring cells. A special case is a “blind portal” representing a border between modeled and non-modeled part of a virtual city. Users are not allowed to cross the blind portal, although they can see a photo of real open area mapped on it.
- *Visibility map* – a list of cells those are potentially visible from given cell.

When rendering the virtual city, a client browser dynamically loads and releases cells visited by a user. A list of cells already loaded into a memory is updated each time a user crosses an invisible sensitive portal between two cells. A client browser requests new data from the server and releases cells that are far/invisible from the current cell.



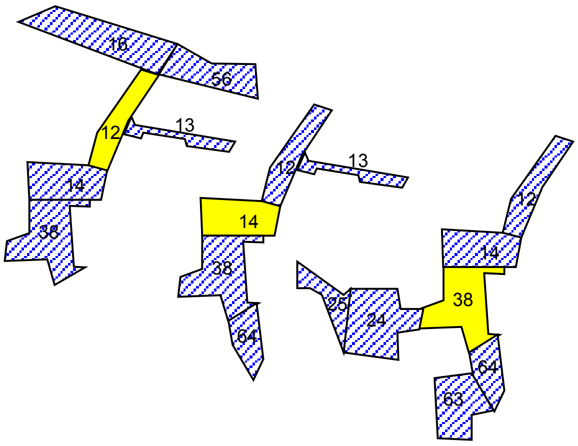


**Figure 4:** *The upper map shows modeled area (dark color). Images below are three snap-shots taken from visualization window showing a process of loading/releasing cells when a user walks through cells 12, 14, and 38 (in the southern region marked by ellipse in the upper map).*

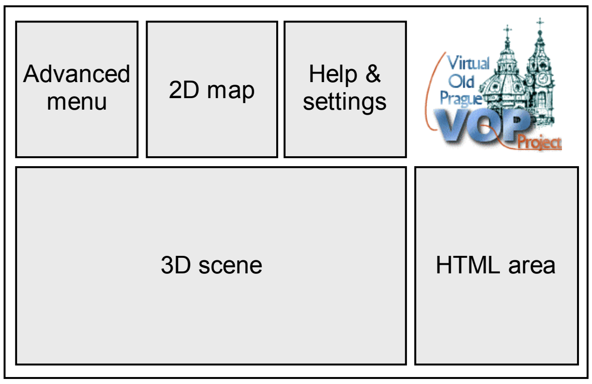


**Figure 5:** *The VOP client window contains several areas with various media and information. Many of them are synchronized with user's walk through the town.*

Note that this approach does not burden a server, because the responsibility for data management lies on the client side. This is extremely useful for web-based applications where plenty of users can concurrently examine the virtual city. The remote interactive walkthrough controlled by a server (as suggested e.g. by Teler and Lischinski [15]) can perfectly optimize a data transfer and a visual quality of virtual scenes, but in principle reduces the number of concurrent users.

## 4. Functions of Virtual Old Prague

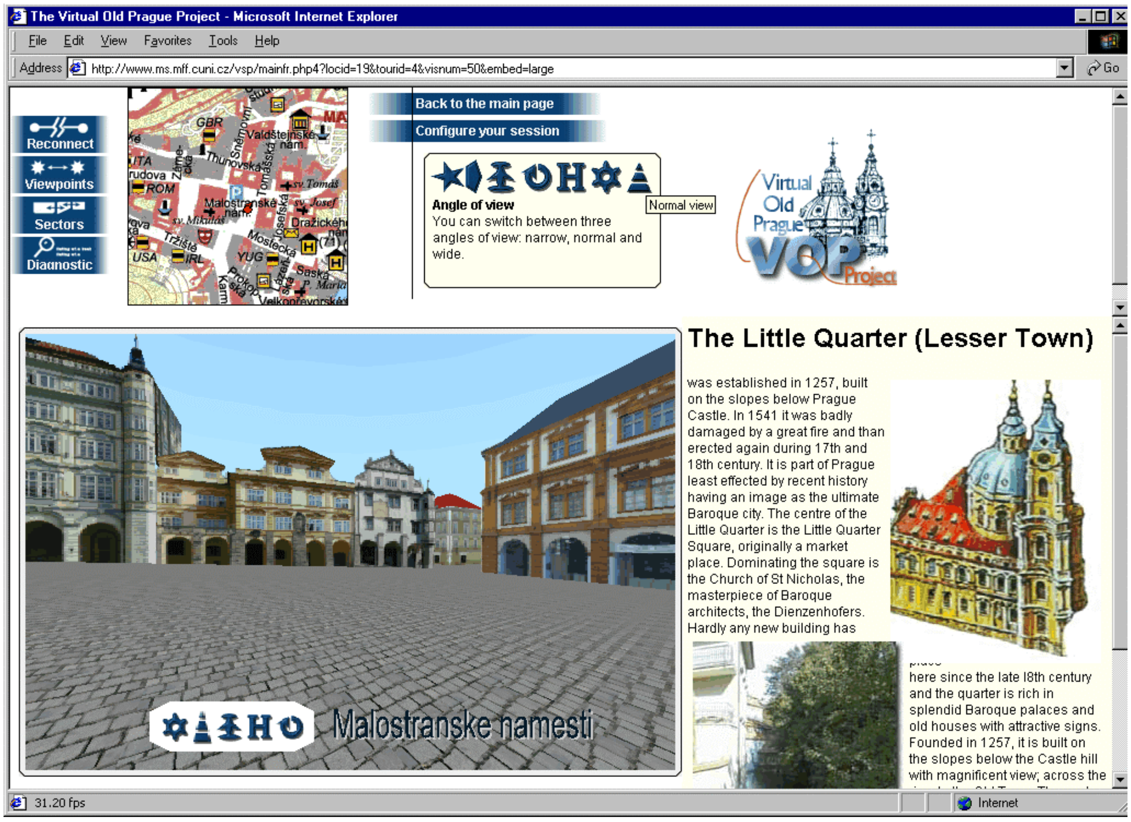
The Virtual Old Prague project consists of a server and a client for web access to the virtual city. The server maintains the data stored in mySQL database and standalone files. The client is composed from HTML frames, VRML browser, Java applet and programs in JavaScript. PHP scripts control a communication between a client and the server.

All 3D data are converted from the language independent description (stored in the database) into VRML language and sent to a VRML browser. We have extensively tested VRML browsers in terms of stability, conformity and performance [18]. Currently the best results are achieved by the Cortona VRML browser from Parallel Graphics running as a plug-in in MS Internet Explorer. Other combinations are also possible (e.g. CosmoPlayer within Netscape), but the stability between the VRML browser interface (EAI) and other parts of the application is not satisfactory. Due to that fact, the VOP project runs with full functionality in MS Windows environment only. We still wait for any full VRML compliant browser available at UNIX/Linux platform.

The scheme of user interface is shown in Figure 5, a snapshot from the real application is in Figure 6. The most informative area is the VRML browser window. A user navigates using a mouse or a keyboard. His/her current position is continuously visualized in 2D map using an icon showing a position and orientation. As user walks through the city, both windows (2D map, 3D view) are updated simultaneously. Moreover, additional information in form of HTML page (text and images) is synchronously presented in a frame on the right side.

The VRML window contains a set of symbols always placed at the bottom (head-up display, see also Figure 2). Those controls allow to switch viewer's field of view into the one of three predefined values, to set the user's height from standard 1.6 meters to 3 meters, to invoke menu with various settings, to re-connect to the server, to call help function, etc. The explanation for those controls is placed at the top of the application window in help area.

The VOP project offers three kinds of navigations. The simplest one is a guided-tour. A user selects predefined tour and the application runs without any further user navigation input (3D navigation input is redirected and is taken from the file instead of a mouse/keyboard). It reminds of replaying movie but all data are loaded and rendered in a real-time. A user can interrupt the guided tour anytime.


The Virtual Old Prague Project - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ms.mff.cuni.cz/vsp/mainfr.php4?locid=19&tourid=4&visnum=50&embed=large
Go
Reconnect
Viewpoints
Sectors
Diagnostic
Back to the main page
Configure your session
Normal view
Virtual Old Prague VOP Project
Malostranske namesti
31.20 fps
Internet

**Angle of view**
You can switch between three angles of view: narrow, normal and wide.

**The Little Quarter (Lesser Town)**

was established in 1257, built on the slopes below Prague Castle. In 1541 it was badly damaged by a great fire and than erected again during 17th and 18th century. It is part of Prague least effected by recent history having an image as the ultimate Baroque city. The centre of the Little Quarter is the Little Quarter Square, originally a market place. Dominating the square is the Church of St Nicholas, the masterpiece of Baroque architects, the Dienzenhofers. Hardly any new building has

here since the late l8th century and the quarter is rich in splendid Baroque palaces and old houses with attractive signs. Founded in 1257, it is built on the slopes below the Castle hill with magnificent view; across the

**Figure 6:** *A snapshot of the Virtual Old Prague application.*

The second kind of navigation is based on selecting one of predefined viewpoints. Such viewpoints are shown on sensitive 2D map that can be displayed when clicking on left textual menu. The last navigation mode is interactive walking mode suitable for users experienced in navigation in 3D virtual space.

Advanced menu contains a lot of settings that help to tune the application performance and to adapt it to user's needs. The VRML window can be enlarged to several predefined resolutions and some frames can be hidden. It is also possible to customize the urban LOD presentation. If the Internet connection is slow, users can forbid the 3rd and 4th level. When the bandwidth is good, it is useful to suppress the 2nd level and to use full textures only.

### 4.1. Scientific visualization functionality

The motivation for the VOP project was not only to design and implement a scaleable model for virtual cities, but also to experiment with various data structures and algorithms. Thanks to that background, the final presentation contains several functions showing the process of maintaining 3D data. Such functions are a kind of scientific visualization and serve for testing purposes as well.

The simplest data visualization is contained directly in VRML browser window. Whenever a client requests the server for a new cell, a "blue stone" symbol is added to the top of VRML scene. The number of stones approximately shows the amount of data to be transferred. This helps users to understand why some streets are built house by house and how data are read by pieces. Blue stones disappear immediately when corresponding data are downloaded.

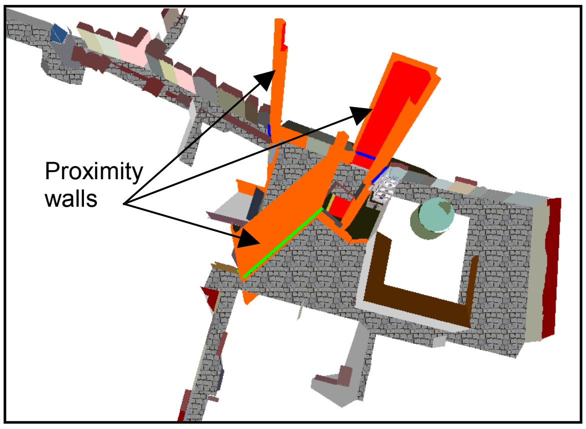


**Figure 7:** *"Expert mode" window allows simulating user's walk by clicking on artificial "proximity walls" representing sensitive portals between cells. A bird-view gives a good overview of loaded/released cells. The image shows a bigger square with a church in the middle. Currently visited cell in the upper left corner of the square has a triangular shape and contains three portals.*

When users are more interested in the internal structure of the virtual city, they can open (from the left menu) a window with a Java applet, where 2D layout of cells is synchronously drawn as users walk through the virtual space (see also Figure 4). The whole application then reminds of complex dynamic environment consisting of 3D scene, 2D city map, 2D cell layout, and HTML tourist information, all driven by user's navigation.

The most advanced visualization tool is offered in so-called *expert mode*. This special function changes the standard content of a VRML browser window, allows using arbitrary navigation mode (fly, examine) and visualizes invisible sensitive portals between cells (Figure 7). This function has been extremely useful when debugging the whole project and is still useful for teaching purposes when explaining the interactive processes inside the system.

In addition to this visualization tool, a textual window with the full journal of communication messages between a client and the server can be activated from the advanced menu.

## 5. Measurement

The VOP application has been tested both from the perspective of data transfer and rendering speed. The current model of the Prague covers more than 150 houses, arranged into 15 streets and 3 squares. Forty additional standalone objects (street lamps, trees, tram stops, etc.) variegate the original façade-based model. All data are subdivided into 50 cells, i.e. each street consists of three or four cells on average. Although the real historical city of Prague is much bigger than the modeled virtual space, the principles of urban LOD and dynamic management of cells have been proven to be scaleable and universal enough for theoretically unlimited virtual city.

Texture images comprise the biggest part of data. Figure 8 shows that the size of all big textures mapped on facades is about 3 Mbytes, while the overall size of small textures used for the 2nd urban LOD is three times smaller. We have made an interesting observation on the ratio between two successive urban levels of details – the overall size in the database for one LOD has been approximately

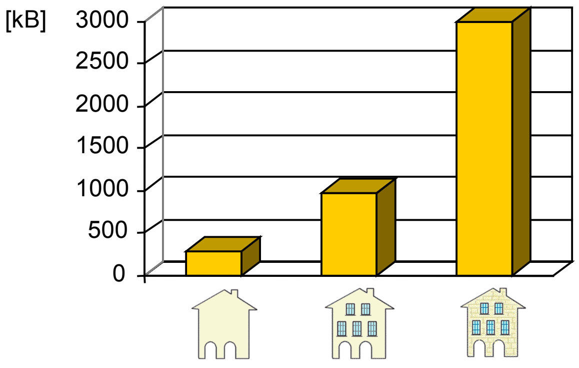


**Figure 8:** *A total amount of data of Virtual Old Prague arranged accordingly to urban levels of details.*

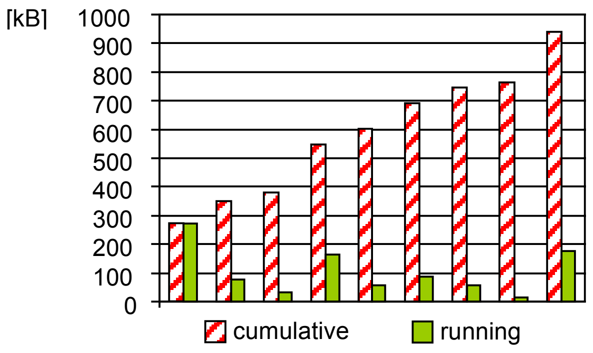


**Figure 9:** *Cumulative graph of data transferred from the server to a client browser when user visited four streets and two squares.*

one third of the size of the next level (actually 300 kB / 970 kB / 3 MB).

The process of cumulative downloading is depicted in Figure 9. The graph shows data transferred from the server as they were requested during one typical walk through the virtual city. The left column in each pair represents overall size of data received from the beginning of the session, while the right column corresponds with the data requested by the client when a new cell was entered. The session took about 2 minutes. During that time, 950 kB was transmitted by small pieces including all three urban LODs for each virtual house. Such data transfer can be achieved even with very slow Internet connection. The biggest immediate amount of data was downloaded at the beginning only, while the rest was transferred step by step.

Dynamic loading and releasing urban cells helps to achieve almost constant amount of data to be rendered by VRML browser. Thus the resulting rendering speed is stable, too. Its value depends on the overall graphics performance of user's computer, but the generally requested value of 25 frames per second for real-time visualization can be achieved without problems. We have tested the VOP project on 900 MHz PC equipped with GeForce 2 graphics accelerator and the speed has been bigger than 30 fps (depending on the size of the VRML browser window).

We should highlight the fact that the urban LOD influences the rendering speed in a slightly different way than it is expected when using mesh simplification LOD algorithms. While the 1st urban LOD (non-textured polygon) is rendered very fast, and the 4th urban LOD (full 3D textured model) has the highest computational demands, two remaining urban LODs are rendered with almost the same speed. The reason is as follows – although the 2nd urban LOD contains tiny textures, they are mapped to a number of rectangles (windows, doors). Spatial transformations and texture mapping performed on those rectangles takes a time that is comparable with the rendering of the 3rd urban LOD represented by one large texture image mapped on one polygon only. The difference between the 2nd and the 3rd urban LOD lies in saving of time during the data transfer and not during the rendering.

## 6. Conclusion

Functionality, user interface, internal structures and measurements performed on the Virtual Old Prague project have been presented in this paper. Although the extent of the modeled city is not very big, we consider the application as a good prototype for building arbitrary virtual cities on the Internet.

The implementation has utilized number of standards therefore the development was fast and without any fundamental troubles. Thanks to working with various data, we have learned that the most critical processing deals with VRML browsers. We had no access to internal data management of VRML browser and we could not influence their performance as much as we would like. Some VRML browsers downloaded all urban LODs for given house before they showed the first level, thus the time saving specific to data transfer was none. We would also appreciate more control on data in client's cache that would bring better efficiency when a user returns back to already visited virtual places. Finally, a very nice feature would be a continuous dynamic blending between successive urban levels of details performed directly by VRML browser. A smooth transition from flat textured façade to its 3D geometrical representation in the 4th urban LOD can be achieved by simple geometry morphing as suggested by Willmott at al. [16]

Besides possible improvements of VRML functionality, the VOP application itself has a space for further improvement. The current version does not use any algorithm for visibility preprocessing among cells - the administrator of the system evaluates the visibility information himself from the planar city map and writes it directly to the central database. This has been possible for small city area with several tens of cells, but definitely not for the large virtual town. Fortunately, the layout of all cells is stored in the database in a general form therefore the visibility preprocessing is just a question of selecting the proper algorithm. We are in a process of searching for the simple but efficient algorithm not only for the visibility preprocessing, but also for automatic creation of city cells.

The important step for the following version of the VOP application should be also an implementation of the search engine followed by automatic navigation from current user's position to selected destination(s). Here we will again take advantage of the layout of cells stored in the database, since it contains all information required for finding the path between two given places, i.e. it comprises a topology graph.

## Acknowledgements

This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic under research program No. Y04/98: 212300014 (Research in the area of information technologies and communications).

## References

1. *Virtual Paris* (based on blaxxun technology) http://www.2nd-world.fr/
2. *Virtual Glasgow* (VRML model) http://iris.abacus.strath.ac.uk/glasgow/
3. *Virtual New Orleans* (VRML model) http://www.planet9.com/earth/neworleans/
4. *Virtual Sydney* (VRML model) http://www.planet9.com/earth/sydney/
5. *Virtual Toronto* (VRML model) http://www.intoronto.com/
6. *QuickTime VR* (Apple web site) http://www.apple.com/quicktime/qtvr/
7. *Virtual Old Prague Project* home page. http://www.ms.mff.cuni.cz/vsp/
8. The Virtual Reality Modeling Language. *International Standard* ISO/IEC 14772-1:1997, http://www.web3d.org/technicalinfo/specifications/vrml97/
9. X. Decoret, G. Schaufler, F. Sillion, and J. Dorsey. Multi-layered impostors for accelerated rendering. In *Computer Graphics Forum 18/3 (Proceedings of Eurographics '99)*. pp. 61-73.
10. F. Durand. A multidisciplinary survey of visibility. *ACM SIGGRAPH Course notes*. Visibility, Problems, Techniques, and Applications, July 2000.
11. A. Heinonen, S. Pulkkinen, I. Rakkolainen. An Information Database for VRML Cities. In *Proceedings of IV 2000 - Information Visualization*, London, pp. 469-473, 2000.
12. H. Hoppe. Progressive meshes. In *Computer Graphics Proceedings*, Annual Conference Series, pp. 99-108, 1996.
13. M. Reddy, L. Iverson, and Y. G. Leclerc. Under the Hood of GeoVRML 1.0. In *Proceedings of The Fifth Web3D/VRML Symposium*. Monterey, California. February 21-24, 2000.
14. J. Shade, D. Lischinski, D. H. Salesin, T. DeRose, and J. Snyder. Hierarchical Image Caching for Accelerated Walkthroughs of Complex Environments. In *Proceedings of ACM SIGGRAPH 96*, pp. 75-82, 1996.
15. E. Teler and D. Lischinski. Streaming of Complex 3D Scenes for Remote Walkthroughs. In *EUROGRAPHICS '2001 Annual Conference*, 2001.
16. J. Willmott, L. I. Wright, D. B. Arnold, and A. M. Day. Rendering of Large and Complex Urban environment for Real Time Heritage Reconstructions. In *Proceedings of VAST 2001 (Virtual Reality, Archaeology, and Cultural Heritage)*, Athens, Greece.
17. J. Zara, P. Chromy, J. Cizek, K. Ghais, M. Holub, S. Mikes, and J. Rajnoch. A Scaleable Approach to Visualization of Large Virtual Cities. In *Proceedings of the Fifth International Conference on Information Visualisation*. London, pp. 639-644, 2001.
18. J. Zara and J. Krivanek. Graphics Performance Benchmarking Based on VRML Browsers. In *VRIC 2001 Proceedings*. Laval: ISTIA Innovation, pp. 111-120, 2001.