Scanning and Rendering Scene Tunnels for Virtual City Traversing

Jiang Yu Zheng
Indiana Uni. Purdue Uni. Indianapolis
723 W. Michigan St. SL280,
Indianapolis, IN46202
jzheng@cs.iupui.edu

Yu Zhou
Indiana Uni. Purdue Uni. Indianapolis
723 W. Michigan St. SL280,
Indianapolis, IN46202
yzhou@cs.iupui.edu

Min Shi
Indiana Uni. Purdue Uni. Indianapolis
723 W. Michigan St. SL280,
Indianapolis, IN46202
mshi@cs.iupui.edu

ABSTRACT

This paper proposes a visual representation named scene tunnel to capture and visualize urban scenes for Internet based virtual city traversing. We scan cityscapes by using multiple cameras on a vehicle that moves along a street, and generate a real scene archive more complete than a route panorama. The scene tunnel can cover high architectures and various object aspects, and its data size is much less than video. It is suitable for image transmission and rendering over the Internet. The scene tunnel has a uniformed resolution along the camera path and can provide continuous views for visual navigation in a virtual or real city. This paper explores the image acquisition methods from slit calibration, view scanning, to image integration. A plane of scanning is determined for flexible camera setting and image integration. The paper further addresses the city visualization on the Internet that includes view transformation, data streaming, and interactive functions. The high-resolution scenes are mapped onto a wide window dynamically. The compact and continuous scene tunnel facilitates the data streaming and allows virtual traversing to be extended to a large area.

Categories and Subject Descriptors

I.4.1 [**Image Processing and Computer Vision**]: Digitization and Image Capture – *imaging geometry, scanning, sampling.*

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *animation, virtual reality.*

General Terms

Algorithms, Theory, Performance, Design, Experimentation

Keywords

Route panorama, scene tunnel, navigation, scene representation, visualization, Internet media.

1. INTRODUCTION

Virtual reality has achieved its high performances and sophisticated tasks on stand-alone systems. However, its applications over the Internet are far from sufficient. The transmission bandwidth, variations in specs of end systems, and low performance of PC have limited the reality presentation and user interactions. As an important field, current *digital cities* on the Internet have not achieved pervasive and seamless scene access, despite a large collection of overhead imagery and digital images are available. On the other hand, although graphics models constructed from aerial images can provide continuous and arbitrary viewpoints, they miss many subtle details visible on the ground. Modeling all objects in an area with texture also requires laborious work. Graphics approaches so far can highlight a particular area of interest but are hard to cover every detail in a city. On the contrary, taking a video can obtain real scenes visible on the ground. It is efficient in data acquisition, and is easy to extend the views to a new street. The drawback is a huge data size that makes it impractical to archive an entire city.

This work aims at constructing an Internet based VR environment by enriching the contents, enlarging the scale, and increasing the continuity in space exploration. To visualize a large urban area on the Internet, we organize cityscapes visible from streets in a sequential way, in addition to isolated spots of interest associated with panoramic views. Scenes along every street in an urban area can be acquired with route panoramas [1-4,5,6], which overcome the weakness of panoramic views such as lacking resolution and visibility on distant scenes. A route panorama is obtained by scanning scenes on a moving vehicle. A virtual slit (a pixel line) is set in the image frame and the temporal pixel values on the slit are consecutively accumulated to form a continuous image belt covering complete scenes along the street. It greatly increases the visibility of the city on the Internet. Fig. 1 shows a section of route panorama on one side of a street.

In this paper, we first propose a complete form called *scene tunnel* for scene acquisition and display. It is obtained by scanning scenes with a CCD ring on a *Plane of Scanning*, which is physically realized by using multiple cameras mounted on a moving vehicle. It has a high image resolution and a more complete coverage of scene in height and aspect as compared with a route panorama. We address the calibration method to set the slits properly in order to preserve the shape in the representation for rendering. Second, we render a *panoramic traversing window* on a spherical screen as the viewer virtually moves along the street. We provide functions for the viewer to look around and actively explore the environment. Combined with global panoramic views, our developed pseudo 3D display gives a realistic maneuver of city traversing. The image mapping or projection does not fall into the conventional perspective projections. It increases the visibility of scenes and preserves the topological shape of objects in the display, and simulates the optical flow direction during the translation. The limitation is lack of depth, and thus the motion parallax of objects.

VRST'04, November 10–12, 2004, Hong Kong.

Figure 1 A section of route panorama captured along a street to show side view of the route.

The advantage of developing this representation is that it can stream data on the Internet in real time. It provides larger view but much less data than a video. The applications range from virtual driving, virtual city tour and navigation, remote route guidance, address finding, urban planning, to cityscape archiving and heritage preservation. The limitation of this method is lack of depth, and hence motion parallax is not precisely rendered in the visualization.

In the environment visualization, panoramic views provide surrounding scenes at static locations [5,10]. Densely located panoramic views are still discrete and have less overlapping. Even with the current panoramic camera and software, taking panoramic views exhaustively in a space is still a heavy task. It is difficult for panoramic views to extend to a large area with a close interval. Image based rendering (IBR) generates arbitrary new views from several images, but image correspondence is required in many of such algorithms. The additional morphing or interpolation of entire images is still not in real time, even with the matched features in prior. Most of the IBRs so far are limited to local and unchanged scenes.

An alternative way to present seamless views is to perform a scanning along a path [5], which covers a long route with small amount of data. A linear CCD scanning in airborne terrestrial photogrammetry [3,7,9] captures large areas from sky. A general use of line sensors is summarized in [19]. Along this line, small image patches in the image sequence are also matched and manifold mosaic is introduced [17,18]. Because the consecutive views from a moving camera focus cannot overlap perfectly in the 2D domain, morphing or interpolation based on feature correspondence is carried out [18]. However, there are problems in occlusion and lacking features. There is no report on modeling a large area targeting at VR. The recent route panorama [1] has increased the length of scanning for access on the Internet. However, there are still some limitations such as the sideways scanning direction (missing some object aspects and thus generates unrealistic scenes in the moving direction), small camera field of view (high parts of buildings in narrow streets missed), and shape deformation due to improper slit setting during the scanning.

This paper starts from defining a general scheme for route scanning in the 3D space. It then introduces how to realize this scanning using multiple cameras. The generated shapes are analyzed briefly. Finally, an interactive and dynamic display of route in a wide window will be introduced.

2. SCANNING SCENE TUNNEL

2.1 Seeing Landscapes on Paths

In a city consisting of architecture blocks and streets, the action is traversing different routes and watching landscapes progressively. Images or panoramic views taken at discrete positions will not have an even image resolution on cityscapes stretching away. An alternative image based approach is to project scenes towards a trajectory for visualizing a transitional movement in the virtual environment. Route panorama is such an example [1]. This paper will give a more general scheme to capture complete views around the path, and increase the aspects of objects along the route. We scan scenes on a vehicle along smooth paths on a horizontal plane. A four-wheeled vehicle can realize this motion. We denote the camera path by ***S(t)*** which is related to time t by reading GPS data during the vehicle movement.

2.2 Plane of Scanning

We propose a general camera and slit setting approach that can (1) acquire general and typical scene aspects, (2) include all heights, and (3) preserve the shapes of scenes to the maximum extent. As depicted in Figure 2, a focus ***O*** moves along a smooth path on the horizontal plane. A vertical *plane of scanning* (*PoS*) originating at ***O*** scans scenes during this motion. At any instance, scenes in the *PoS* are projected towards ***O*** and are imaged on a pixel ring ***r*** centered at ***O***. If we continuously accumulate temporal pixel data on ***r*** and arrange them consecutively along the time axis, we obtain a 2D image $\boldsymbol{I(t,\theta)}$ that is the half side of a scene tunnel, which has the spanning angle $\theta\in[-\pi/2, \pi/2]$ in the PoS as its vertical axis and time t as its horizontal axis. If we replace the ring with a CCD line, it becomes a push-broom sensor with a limited spanning angle [6] or dynamic projection image [19]. The *PoS* is set briefly to emphasize cityscapes sideways. The sky and road area in the forward direction are not as important as the side scenes along a street.

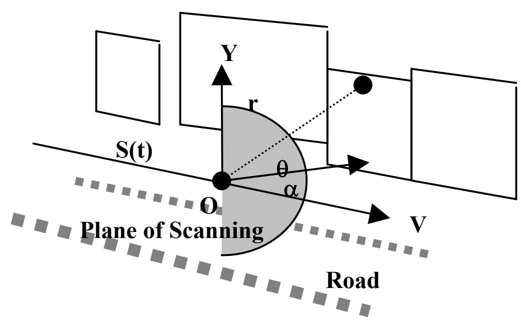


Figure 2 A plane of scanning from the camera focus that moves along a smooth path on a horizontal plane.

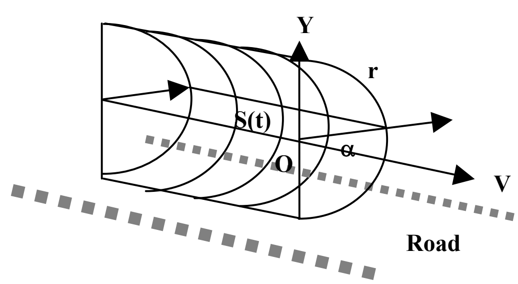


Figure 3 A half-tunnel image obtained along a linear path.

In order to preserve the shapes of scenes in the accumulated images, we define the *PoS* to be vertical in the 3D space. This will guarantee that the vertical lines in the 3D space such as architecture rims and poles on the roadside are scanned instantaneously and their images in the scene tunnel are still vertical. Figure 3 shows a half tunnel. With the vertical *PoS*, we can further choose its orientation with respect to the moving direction. Denoting the path tangent by ***V*** (velocity of the vehicle), the angle α ($\alpha \neq 0$) between PoS and ***V*** is fixed during the scene tunnel acquisition. This angle determines the aspects of objects in the scene tunnel. We obtain three typical viewing directions: fore-side view, side view, and rear-side view of a route by setting vertical *PoS* with $\alpha<\pi/2$, $\alpha=\pi/2$, and $\alpha>\pi/2$ on each side of the route respectively (Fig. 4). The fore-side and the rear-side views of the route are particularly important in displaying forward traversing and are exchangeable when traveling inversely. The side view of the route is suitable for observing building front. Figure 5 shows three aspect views of the same house.

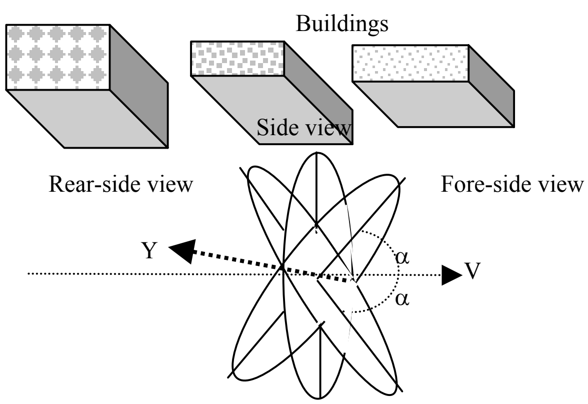


Figure 4 Three rings located around the focus to form three PoSes for route scene scanning (view from top).

Figure 5 Different aspect views of a house scanned with three PoS. Only middle part of θ is included.

2.3 Shapes in the Opened Scene Tunnel

A vertical *PoS* preserves good shape properties in the generated scene tunnel. First, the 3D vertical lines are also vertical in the scene tunnel regardless the curvature of the path. A vertical *PoS* is parallel to all vertical lines. This is invariant to the camera translation and rotation, and thus invariant to any linear or curved vehicle trajectory on the horizontal plane. Second, any line or curve parallel to the path will leave its projection in the scene tunnel as a horizontal line. Many horizontal lines along the street are applied. The shapes of front faces of architectures are preserved in the scene tunnel. We examine the shapes of the rest of the lines in the scene tunnel for a linear path.

Lines orthogonal to the path such as the ones on the side surfaces of buildings, roads, and pedestrian crossings, as Fig. 6 depicted, are scanned instantaneously by the middle *PoS* ($\alpha=\pi/2$). The resulting projections are vertical in the scene tunnel. In other two scene tunnels obtained from fore-side, and rear-side *PoS*, such a line becomes a curve (Fig. 6b). Locating a coordinate system $\boldsymbol{O}_0$-***XYZ*** at an arbitrary position on the path of ***O***, with the ***X*** axis parallel to the path and the ***Y*** axis being vertical, a point ***P***(X,Y,Z) on the line is scanned by the *PoS* at time *t*. The location of *PoS* is then at ***O***(*vt*, 0, 0), where *v* is the normalized vehicle speed. The line intersects with the vertical plane on the path (i.e., XY plane) at ***B***(X,Y,0). The projection of the line on the scene tunnel ***I****(t,θ)* is then

$$\theta = \tan^{-1}(Y/D) = \tan^{-1}(Y\cos\alpha / (X - tv)) \qquad (1)$$

D is the horizontal distance from ***O*** to the line within the PoS.

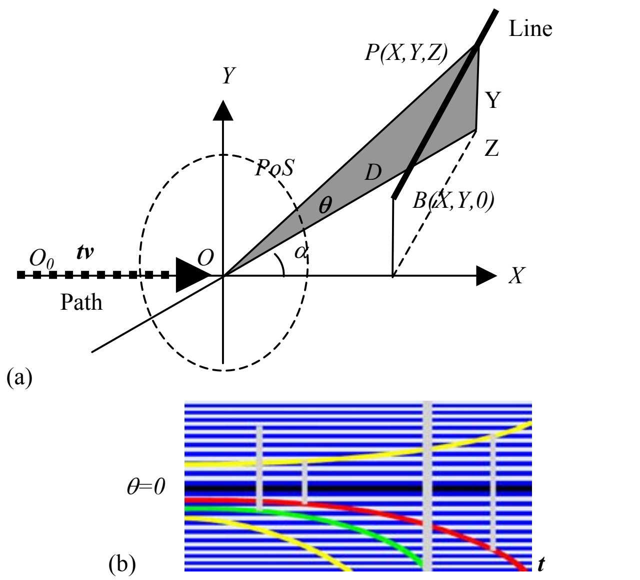


Figure 6 Lines projected onto a scene tunnel. (a) Scanning a line orthogonal to the path with a PoS. (b) Basic types of lines in the space display their shapes in the scene tunnel.

More generally, we consider an arbitrary unit vector (a, b, c) with a change in depth form the path (i.e., $c\neq0$). A line along this vector has an intersection ***B***(X_0,Y_0,0) with the vertical plane on the path (XY plane). The line is scanned at time *t* by a *PoS* with orientation α. We have the relation

$$[(X_0 - tv,\ Y_0,\ 0) - \frac{Y}{\tan\theta}(\cos\alpha,\ \tan\theta,\ \sin\alpha)] \times (a,\ b,\ c) = 0 \qquad (2)$$

It can be further written as a curve

$$\theta = \tan^{-1}\frac{(a\sin\alpha - c\cos\alpha)Y_0 - (vt - X_0)b\sin\alpha}{(vt - X_0)c} \qquad (3)$$

which also includes the specific case of (1), if a=0 and b=0.

3. SCENE CAPTURING WITH CAMERAS

3.1 Flexible Camera Setting

We use cameras to capture scene tunnels along streets. The camera can be set flexibly after a *PoS* is fixed. The intersecting line of the image frame with any *PoS* forms a pixel line as Figure 7a shows. To avoid heavy camera calibration, we set cameras briefly towards desired directions and leave the geometry calculation to the slit calibration. The requirement of the camera setting is to intersect one or more *PoS* for single or multiple slits in the image frame.

In order to obtain an image ring on each *PoS* as complete as possible, we stack multiple cameras facing up-and-down for different heights. Figure 7b shows a *PoS* intersecting two camera frames with different tilts. Some overlap between up-and-down frames must be kept. If one camera is adequate to cover entire height along a route (when buildings are low and the street is wide), the data from the upper camera are ignored to save transmission and rendering speed in the VR display.

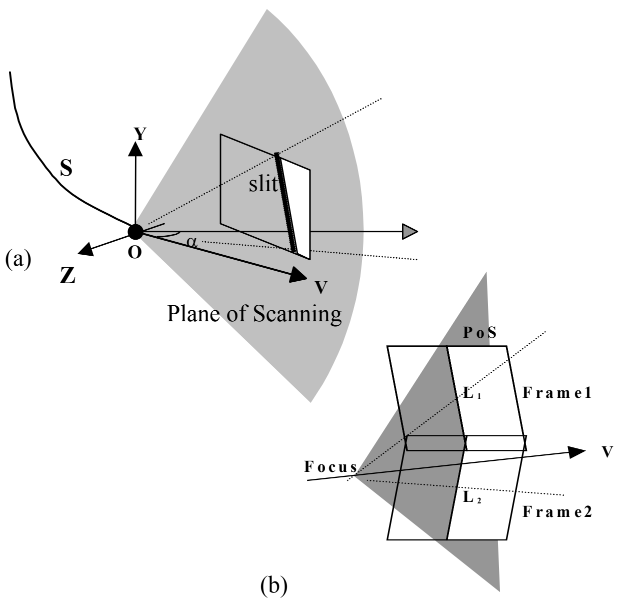


Figure 7 Virtual slits in the image frames for scanning scenes.

3.2 Locating Slits for Quality Images

Locating a slit precisely is important to preserve the shape in the scene tunnel. Because none of the camera axes (optical axis and frame axes) can be guaranteed to be horizontal, we locate the slit accurately in the image frame to ensure the vertical *PoS* in the 3D space. Before heading to the street, we stop the vehicle at a horizontal ground and fixed the camera. A vertical sampling line on a building is selected. It is captured in multiple camera frames for memorizing its positions as slits. By setting the sampling line at different α angles, we can locate slits for different *PoS*. In general, if a camera has an up-facing tilt, the slit for the fore-side view or rear-side view leans in the image frame (Fig. 8).

If the slit does not yield a vertical *PoS*, 3D vertical lines will be slanted in the generated scene tunnel. An even worse result is that the leaning angles of vertical features depend on their depths, which are impossible to recover with any simple 2D transformation without depth information. This is undesirable in the scene visualization during the city visualization.

Figure 8 Vertical lines projected onto an image. Their projections are slanted and extended to a common vanishing point in the image plane.

In case the detection of the sampling line is inaccurate, we use a vanishing point based calibration method to improve the slit accuracy. According to the computer vision principle, all vertical lines have their image projections pass through a vanishing point. In the 3D space depicted in Fig. 9, the vanishing point is the penetrating point of the *Y*-axis through the image plane. We therefore determine a rule to set the slit in the image frame.

Slit setting: The slit, which is the intersection of the vertical *PoS* and the image plane, must pass the vanishing point of the projections of the 3D vertical lines in the image plane.

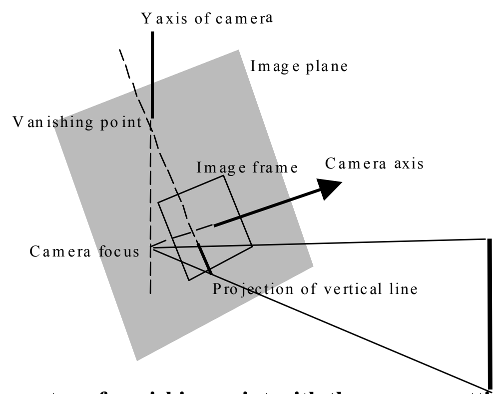


Fig. 9 Geometry of vanishing point with the camera setting.

The estimation of the vanishing point is done by picking up slanting edges extracted from multiple images and the least squared error method finds the optimal point of their crossings. Through the vanishing point, a slit can be located precisely in the image according to preferred α (the direction of *PoS*). The obtained slit works for entire paths including curved ones for obtaining scene tunnels.

3.3 Multi-Slits Setting And Image Integration

If the street is narrow, or high rises are on the roadside, multiple cameras are used for the scene tunnel. The camera focal points are set as close as possible and their frames share a common part on the *PoS*. A slit is located in each frame on the projection of a common vertical line; a *PoS* then passes all the slits in different frames. As the vehicle moves out, the coordinated slits scan scenes simultaneously at different heights. The resulting up-and-down image belts undergo a transformation that maps the slit pixels onto the ring ***r*** by

$$\theta = \tan^{-1}(y / f) \tag{4}$$

where y is the coordinates of the slit pixel and f is the focal length of the slit. The slit focal length is further calculated from the focal length of the camera and the image distance of the slit from the image center. The transformed image belts are consistent along the time axis (horizontal length) and the overlapped portions also fit well. We thus compose the scene tunnel along the path by stitching them together.

Figure 10 gives an example in which two image belts form the scene tunnel. The heights of the slits are 640 pixels and the overlapped part is 200 pixels in height. The combined one may have resolution as high as 1000 pixels. The height of horizon (θ=0) in the scene tunnel is calculated either from the vanishing point in the sampling images or from the height of asymptotic line of those curves that stretch in depth from the camera path (Fig. 6).

Due to the auto-exposure of the up-and-down cameras, the overlapped parts in both images may have different color distributions. Particularly, the high building in the upper image belt may be improperly exposed because of the direct lighting. For architectures in the up-and-down image belts, we calculate the average color components (R_{ave}, G_{ave}, B_{ave}) in the overlapped regions. Then, the color values in the upper image belt are scaled so that the averages of up-and-down image belts become consistent in the overlapped part. We therefore achieve a smooth integration of up-and-down images.

When the vehicle moves on uneven roads, the bumping camera may disturb the angle θ, which may cause zigzags on the horizontal structures in the scene tunnel. We employ an algorithm to rectify the zigzags and reduce the waves in order to improve the image quality of the scene tunnel [13].

A slit or pixel line physically employs a narrow perspective projection [1]. The slit-covered width on a distance object is wider than the width on a close object. However, the length of the object along the t axis is unrelated to the depth, because the parallel PoS in the scanning. The distant scene projected to the pixel line is then the Gaussian average of the surface color in a wide range. This causes a blurring effect along the t axis, which is named as stationary blur. A 3D point at distance tends to appear in multiple slits in the scene tunnel, which is similar as the motion blur on moving objects in a perspective image.

4. VISUALIZATION OF ROUTE SCENES

4.1 Sequential Data Transmission/Loading

Similar as the route panorama, one of the fundamental merits of the scene tunnel is its compactness, because the pixel ring does not scan the same scenes twice. This eliminates the redundant coverage of scenes and allows the cityscape archiving for a long distance. In addition, JPEG or GIF compression of the scene tunnels further drop the data to half or 1/3. Eventually, the scene tunnel results in the data much less than any compressed video. We can drive the vehicle along every street in an urban area to register complete route scenes.

This work achieves the transmission of the scene tunnel on the Internet. Because the scene tunnel is a sequential and continuous image representation, only the data near the viewer's position are necessary to be transmitted; it handles a constant amount of data at any instance. Such a VR environment requires less computation and memory resources than a VR environment based on graphics model. With the viewer's virtual position on the route, we display the current section of scene tunnel and buffer the following section for seamless city traversing. Denoting the scene tunnel of *i*th route by ST_i ($\boldsymbol{S}$), $ST_i(\boldsymbol{S}\pm\Delta)$ is displayed where Δ is a visible distance along the route. This progressive data transmission makes the large city traversing possible on the Internet. Associated to the scene tunnel, text descriptions TR_i ($\boldsymbol{s}_j$, $\boldsymbol{e}_j$), j=1,2,…, such as addresses and architecture names are prepared and pop up when the text display mode is on and the passed distance $\boldsymbol{S}$ reaches object *j*, i.e., $s_j < \boldsymbol{S} < e_j$.

4.2 Rendering Scene Tunnels

In order to provide views for traveling along a street, we render the scene tunnel dynamically on a *spherical traversing window*. It is a pseudo 3D display — warping 2D image from scene tunnel to the display window. Although it lacks object depth or motion parallax, we can still simulate the traversing along a route at an acceptable level. We rank the importance of visual information from visibility, topological relation of shapes, flow direction and then depth and motion parallax. The display provides most of the information except the true geometric depth.

Figure 10 Integrating image belts to form a half-side tunnel. (a) Upper image, (b) Lower image, (c) Merged image.

We scroll the scene tunnel around a spherical retina in different speeds to show the virtual move along a street. Scene tunnel on the cylinder along the camera path is not suitable for direct display, because the vertical lines on its surface will be unfavorably bended. In our display, we first project the scene tunnel onto two vertical walls. The vertical lines in the 3D space are now standing vertical on them. The mapping from the sampling ring towards the virtual wall is

$$Y=W_{\alpha} \tan \theta \qquad (5)$$

where W_α is the horizontal distance from the camera center to the wall in the *PoS*. In the second phase, the vertical walls are mapped onto a spherical retina (Fig. 11). The resulting view can be projected to a dish type spherical screen, or displayed as an opened window on flat screens.

In more details, the scene tunnel within X$\in$[S-Δ, S+Δ] is mapped towards the *spherical traversing window* at viewer's position S, where Δ is a defined visible distance. Assume a point $\boldsymbol{I}(t,\theta)$ in the scene tunnel is mapped onto $\Omega(\phi,\beta)$ on the spherical screen, where $\phi \in$[-180°,180°] and $\beta \in$[-90°, 90°]. The coordinates of Ω at S are calculated by

$$\beta = \tan^{-1}\frac{\tan\theta \sin\phi}{\sin\alpha} \tag{6}$$

$$\tan\phi = \frac{W}{X - S} = \frac{W}{vt - S} \tag{7}$$

where W is the orthogonal distance from the sphere center to the virtual wall (Fig. 12).

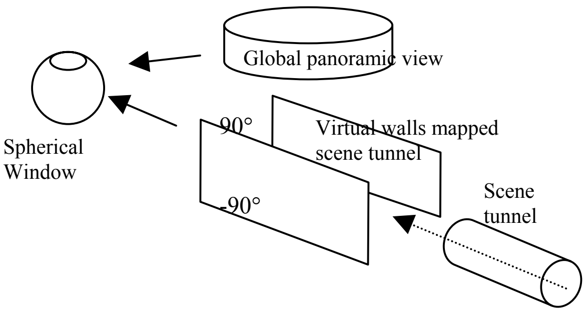


Figure 11 Image warping from the scene tunnel to a spherical traversing window

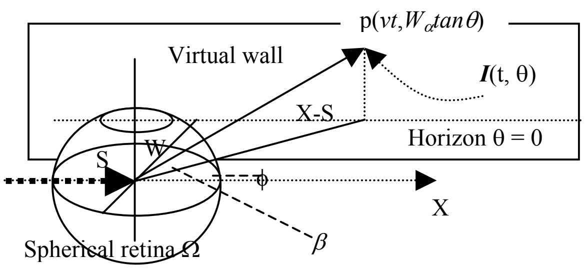
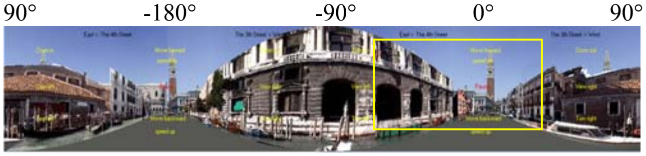


Figure 12 Mapping from scene tunnel to a spherical retina to simulate virtual translation in the VR environment.

Figure 13 Display of a panoramic traversing window.

90° -180° -90° 0° 90°



Moving forward S(Δ=0) Δ=∞

Figure 14 Entire spherical display (360 degree) combined from global panoramic view and two-sides of the scene tunnel.

Any line parallel to the camera path in the 3D space (horizontal in the scene tunnel) appears as a curve, and vertical lines in the 3D space appear to be vertical, in the opened image of the spherical retina. Viewers can look around in front of a large window as Figure 13 shows, or watch a view-port cut from the opened spherical image if the viewer's head orientation is sensed.

In the rendering process, we prepare an intermediate layer to combine several image sources before posting them to the window. In the scene tunnel, we first make the sky area transparent. The rendering process is then as follows.

1. A global panoramic view (described in the next subsection) is projected to the intermediate layer, on which the ground area below the horizon (β=0) is further painted.
2. Over that, the buffered sections of the scene tunnel are rendered according to Eq. 6-7 around viewer's position S. The advancing step of scene tunnel is changeable in response to viewer's virtual speed. This dynamic rendering during the translation generates optical flow in the intermediate layer expending from (0°,0°) and merging at (180°,0°) on the horizon (β=0).
3. A view port cut from the intermediate layer is pasted onto the screen to show the traversing. It can be shifted and scaled according to the viewing direction.

The translation can be kept at a reasonable speed if high performance PCs and transmission bandwidth are available. We can realize translation from walking speed, to running and even driving speed according to the size of view port. The viewer's translation and rotation are visualized simultaneously. Depending on the screen, the image size can be as large as 2000×1000 pixels approximately. Figure 15 shows a section of the route displayed on the entire screen.

Fig. 15 Forward motion with fore-side views of architectures.

4.3 Combining with Global Panoramas

Global panoramas are captured at high positions or open sites from where distant skylines can be seen. The global panoramic view is projected to the intermediate layer according to the route direction, in order to provide the viewer with orientation information. The horizon projected in the global panoramic view is easily identified and is aligned with ϕ=0 in the intermediate layer. The point to take a global panorama is not to include close buildings that may destroy the scale of scenes in the intermediate layer. If there is no distinct landmark visible at any location along a route, a sky image is used, and buildings on the route sides will occupy most areas in the traversing window. Because the sky area in the scene tunnel has been made transparent, users can see background scenes over low building tops in the composite spherical traversing window. Depending on the user's viewing direction, a proper set of scene tunnel is selected from fore-side view, side view, or rear-side view for display.

5. TRAVERSING CITY IN VR SPACE

5.1 Virtual Movement along Routes

The interactions in this VR environment include changes in translation direction, traveling speed, viewing direction, zooming, turning at a street intersection, and stop. The view port is divided into nine touchable areas for user's motion control if the scene tunnel is displayed in a normal flat screen. These control regions are forward/backward translation, viewing left/right, zooming in/out, and turning left/right at next street intersection (Fig. 18). User can also use keys or other devices to control his/her actions. The visualization can be further connected to treadmill and cycle equipments for entertainment; to vehicle simulators for drive training; or even to a GPS connected display or glasses to show daylight scenes and addresses in the evening driving.

To increase the rendering speed, block-wised image copy and scaling are implemented in the transformation from the scene tunnel to the traversing window using Eq. 6, 7. At a static location, viewing direction changes can be rendered fast because only view port copy is involved in the rendering If the scenes are mainly centered at the horizon, rendering the lower image belts at will be sufficient. The data to transmit and render become much less than the entire tunnel.

5.2 System and Experiments

Our cameras have wide-angle lenses (Sony 1.3×FOV). Because of the distortion of the shape taken by these lenses, vertical lines are bended in the images. Our slits are therefore located on the bended pixel lines for sampling at different heights. The ground area under or close to the vehicle is omitted. The selection of α orientation determines the aspect of objects to be scanned. A properly selected α gives a good ratio between lengths of front and side surfaces in the displayed traversing window.

Experiments on the scene tunnel acquisition have been carried out at several districts. The vehicle speed is kept approximately constant. The slit lines in different camera frames have the same sampling rate. After scenes are captured, a shaking removing algorithm removes jitter components in the scene tunnel due to the vehicle shaking (i.e., the shaking in the camera pitch). These image belts are then transformed and merged according to their overlapped area.

We visualize the scene tunnel by using JAVA in the Internet browser. The rendering speed of the spherical traversing window varies from the image size, network bandwidth, file loading speed, and machine rendering speed. For a small view port, the rendered virtual translation can reach a train speed. If the display is a wide screen with narrow field of view in height, we can even approximate the spherical traversing window with a cylindrical window (with upper and lower cutoffs), which has an even faster rendering speed. Figure 16 shows the quality of an opened scene tunnel (entire vertical scope) and Figure 17 shows screen shots of route traversing. Further zooming in is possible to focus on a particular part of the scenes. The immersive display is possible with the sufficiently high resolution.

6. CONCLUSION

This work creates a visual representation named scene tunnel that archives cityscapes along routes in an urban area for area visualization on Internet-based interactive system. The compact data size, continuous and sequential data formats allow the VR traversing to be realized with streaming data transmission. The completeness of the scene coverage in aspect, height, length, and the high image resolution provide a virtual environment extendable to large areas and long routes. We have developed display methods of the scene tunnel for viewer's interaction when he/she is virtually moving along a street on the Internet or immersive display. The representation preserves shapes of object at an acceptable level for navigation even without 3D information. The proposed streaming data transmission and rendering have great potential as the recent machine speed and network capability increase dramatically. It will have applications in driving, navigation, training, virtual tour, on-line business such as shopping and real estates, etc.

7. REFERENCES

[1] J. Y. Zheng, Digital Route Panorama, IEEE Multimedia, Vol. 10, No. 3. pp. 57-68, 2003.

[2] S. Ozawa, M. Notomi, H. Zen: A wide scope modeling to reconstruct urban scene, Proc. ISPRS Commission V, Int. Symposium on Real-Time Imaging and Dynamic Analysis, pp.370-376, 1998.

[3] H. Zhao, R. Shibasaki, A Vehicle-borne urban 3D acquisition system using single-row laser range scanners, IEEE Trans. on SMC Part B: Cybernetics, Special Issue on 3-D Image Analysis and Modeling, Reference, SMC-3DIAM-37.

[4] J. Y. Zheng, S. Tsuji, Panoramic Representation for route recognition by a mobile robot, Int. Journal Computer Vision, (1992), Vol.9, No.1, pp.55-76.

[5] J. Y. Zheng, S. Tsuji, Generating dynamic projection images for scene representation and recognition, Computer Vision and Image Understanding, Academic Press, Vol. 72, No. 3, Dec., pp. 237-256, 1998.

[6] R.Gupta, R. Hartley, Linear pushbroom cameras, IEEE PAMI, Vol.19, No.9, pp. 963-975, 1997.

[7] S. E. Chen, L. Williams, Quicktime VR: An image-based approach to virtual environment navigation, SIGGRAPH95, pp. 29-38, 1995.

[8] H. Ishiguro, M. Yamamoto, S. Tsuji, "Omnidirectional stereo", IEEE PAMI, Vol. 14, No. 2, pp. 257-262, 1992.

[9] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, D. H. Salesin, Multiperspective Panoramas for Cel Animation, SIGGRAPH 97, pp. 243-250, 1997.

[10] T. Endo, A. Katayama, H. Tamura, Image based walk through system for large-scale scenes. VSMM98, Vol. 1, 269-274, 1998.

[11] S. Peleg, B. Rousso, A. Rav-Acha, A. Zomet, "Mosaicing on adaptive manifolds", IEEE PAMI, Vol. 22, No. 10, pp. 1144-1154, Oct. 2000.

[12] Z. G. Zhu, E. Riseman, A. Hanson, Parallel-perspective Stereo Mosaics, ICCV 2001.

[13] J. Y. Zheng, Stabilizing Route Panorama, Int. Conf. Pattern Recognition, Cambridge, UK, 2004.

Fig. 16 Section of scene tunnels (side view) with high buildings. Upper and lower image belts are merged.

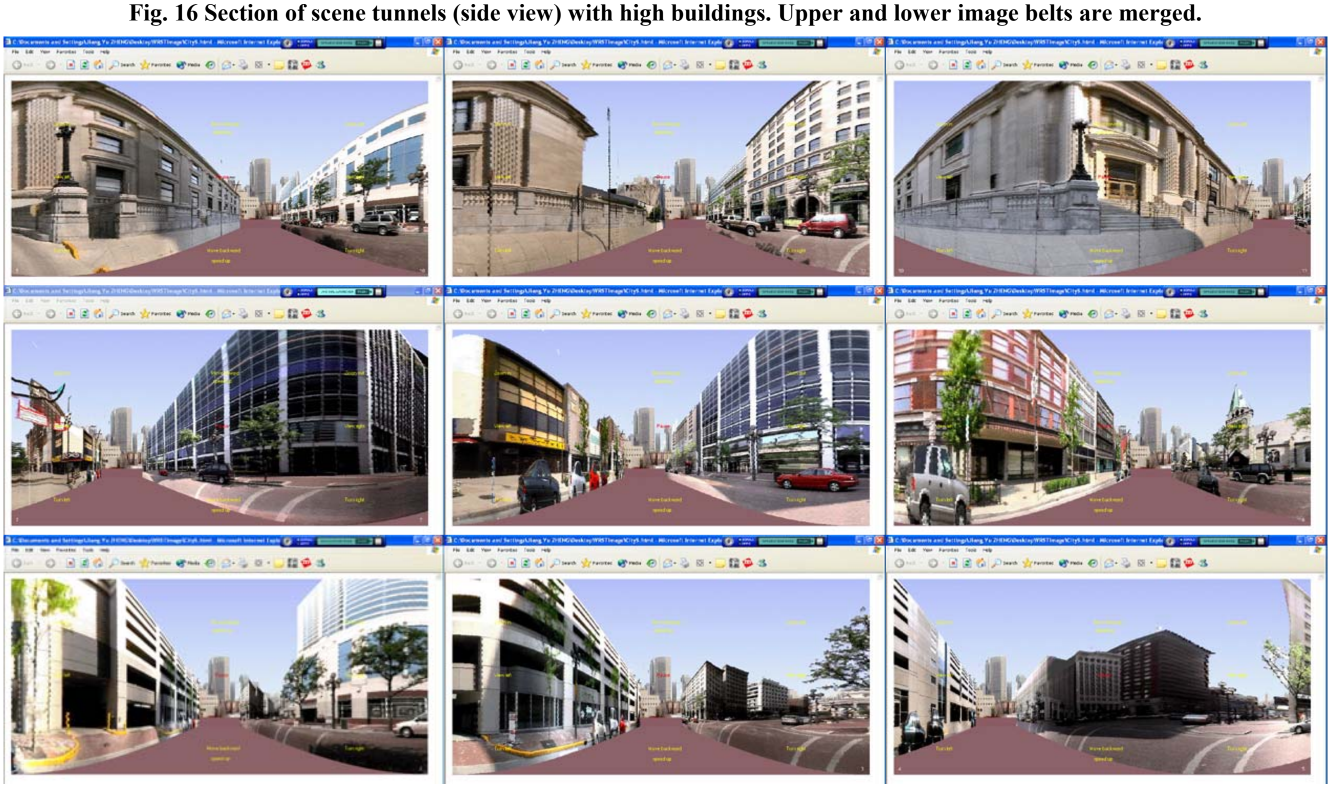


Fig. 17 Screen shots of city traversing rendered with scene tunnels and global panoramas.