Mapping Cityscapes to Cyber Space

Jiang Yu Zheng and Min Shi
Dept. of Computer Science, Indiana University Purdue University Indianapolis
jzheng, mshi@cs.iupui.edu

Abstract

This work establishes a cyber space of an urban area for visiting on the Internet. By registering entire scenes along every street and scenes at many locations, people can visually travel from street to street, and reach their destinations in cyber city. The issues we discuss here are how to map a large scale area to image domain in a small amount of data, how to cover visual information as complete as possible in the mapping, and how to effectively display the captured scenes for various activities in cyber city. Route Panoramas captured along streets and Panoramic Views captured at widely open sites are associated to city maps to provide a navigation function. This paper focuses on the properties of our extended images — route panorama, the archiving process applied to an urban area, and a real environment we developed to transmit and display scenes on the WWW and portable devices. The created cyber spaces of real cities have broad applications such as city tour, real estate, E-commerce, heritage preservation, urban planning and construction.

1 Introduction

A cyber city could either be a computer generated 3D space containing a collection of nodes as some fiction movies display, or be a duplicate of a real geographical city with some virtual functions applied to it. The later normally is more difficult because it has to acquire data faithfully from the real world. Recently, multimedia and VR techniques have provided more and more maps, images, video clips and 3D VRML models to represent real cities. However, the visual data are still in coarse, sparse and discrete formats. It is difficult for people who have no knowledge of an area to build up an entire space in their mind from limited local scenes searched on the web. This work investigates full mapping of scenes in an urban area to a cyber space for visual navigation.

Maps (aerial images) and images (including extended format such as panoramic views) project scenes to planes orthogonally or to focal points in perspective projection. However, a map has less detailed visual information on the ground, while discrete images may not contain sufficient global scenes. The linkage of maps and images has achieved good perception style on the web. To bridge global and local information more closely, this work creates another dimension of focus to project scenes towards lines along streets. This mapping is achieved by dynamic scanning of scenes with a camera moving along streets. It generates Route Panoramas that provide continuous visual data of the streets.

Streets are important components of a city not only because they connect one place to another spatially, but also because they possess rich visual context closely related to our life style and reflects human civilization. The established street models in the cyber cities can facilitate a broad range of applications from finding address, cyber tour, E-commerce, virtual museum and heritage sites, urban planning and renewing, traffic navigation, etc.

The objectives of this work are to:

1. Design a scheme to capture scenes of interest in various image formats including route panoramas, panoramic views and around-object images. We scan all streets in an urban area to build an image based city model.
2. Introduce the route panorama and its properties for mapping scenes to a grid of city streets. We will focus on the projection, the generated 2D shape and the visibility of the route panoramas.
3. Develop display tools so that route panoramas will be transmitted and displayed on the Internet, linked from a map, and scrolled on small portable terminals. Users will be able to navigate the cyber city with several kinds of projections from scenes.

The related works so far include panoramic views that project 360 degrees scenes toward static points, and the early version of the route panorama [2-3]. Currently, there are several approaches to capture panoramic views. The first is to scan scenes through a slit line while rotating a camera [5]. The second approach is called mosaicing or photo stitching [9, 14]. The panoramic views are particularly representative at wide and open spaces.

A route panorama displays scenes along a path. It is based on our early research of generalized panoramic views (GPV) first invented for mobile robot navigation [2,3,4]. The GPV is a special case of a more general image representation called dynamic projection image [13], which is comprised of many slit views taken at different time instances when the camera moves along a path or a static camera looks at a dynamic flow. On the display aspect, Li [15] has associated the generated panoramic views with the global positioning system (GPS) for car navigation in an urban area. This topic was later explored and expanded [18,19] from slit scanning to the stripe mosaic using narrow
image patches. The mosaicing technique [7,8] requires feature correspondence between consecutive views. Although they eventually generate a nice 2D view, the expensive matching algorithm limits the extendibility to long routes.

In the development of cyber city, graphics generated models lack reality. Some projects have used texture mapping from images to improve the realism. However, the generation of an area requires much laborious modeling by using interactive software.

2 Mapping Scenes to Cyber Space

2.1 Visual Maps for Cityscapes

The goal of this work is to map all designated scenes of interest into a cyber space for city navigation and indexing. The criteria of cityscape mapping are

- **Complete**: mapped scenes should include entire area in a city, covering the landscapes and architectures as much as possible in all designated directions.
- **Continuous**: visual data should be seamlessly connected or easy to switch for streaming media transmission and display, which allows a viewer to travel from place to place in the city.
- **Compact**: images should have less redundant coverage of scenes so that they minimize storage and transmission bandwidth.

Users may start from a map (Fig. 1) to traverse the cyber city. As commonly displayed on the Internet, many locations have links to discrete images showing scenes there. In the case where many scenes around can be observed from one location such as a park, a square, or an outlook, a panoramic image can be taken to include scenes at all different orientations. Scenes around the focal point are then mapped onto a cylindrical image surface, a conic image surface for high architectures [13] or a spherical image surface [14].

An aerial view can be generated either from the slit view scanning during the rotation [2] or from stitching a sequence of images [9]. The panoramic view has a data size smaller than the image sequence because the redundant data in the overlapped images are dropped. Similarly, a route panorama saves much data compared with a sequence of discrete images covering the same scenes along the route.

If an architecture or object has rich visual context on each side, several discrete images may be taken at selective distances and orientations to cover all its aspects. This may be suitable for observing a monument, a sculpture, or a house. If the images are densely taken, we obtain around-object views of the object.

In this work, we add a new mapping called route panorama to the existing viewing scheme [20]. An example of route panorama is given in Fig. 2. We project scenes on one side of a street towards a smooth path along the street, which might be a curved one, on the horizontal plane. A route panorama is created by scanning route scenes continuously with a virtual slit camera, which is substantially by picking up a pixel line in the image frame.

The connection of pixel lines in the consecutive images forms a long, continuous 2D image belt containing major scenes of the street.

2.2 Viewing A Cyber City

We define visual node as a data set for a location in the cyber city. Visual nodes can have different types as site, route, and unit. A visual node consists of a panorama type image, a text file, a map location, and its relation with other nodes. For different types of visual nodes, panorama can be

![A Different types of images taken at various locations.](image1.png)

![Coverage of scenes in an urban area.](image2.png)

Fig. 1. Various mapping covering cityscapes in a map.
defined differently. An open site is suitable to be covered by a panoramic view. A narrow route is covered by route panoramas. And a unit such as a building or a monument is snapped with around-object images.

For a global area such as a district or a town containing many local locations, a visual node also includes many discrete local views that are representative for general understanding of the area. A text file is prepared for introducing general idea of the node, and the position of the area is indicated in a global map. Links to more detailed local visual nodes are also included.

We have designed a display in JAVA for displaying a single visual node (Fig. 3). A map window is prepared for indicating the location of the current visual node in a global city map. A panorama window is arranged for rotating a panoramic view, scrolling a route panorama, or listing around-object images, depending on the type of the visual node. It is also used to list representative local views for a global visual node. Another image window is set for a focused discrete image or dynamic slide show of around-object images. A text window is prepared for detailed description. Other menu windows are also presented for linking to other visual nodes, either a global level parent node, or several detailed local level nodes. The menus are categorized by space, time, and property, respectively.

We register many visual nodes in a city including their orientated discrete images, panoramic views, and route panoramas are taken. Some around-object views can also be collected from densely taken panoramic views or route panoramas nearby.

Global and local visual nodes are organized in a hierarchical structure (Fig. 4). Links from a global visual node to local visual nodes are embedded in the space menu, map, and representative local images. Users are able to jump to a local visual node through any of these links. The entire display will switch to the selected local visual node. We call this scheme in-depth access.

At the same level of details, geographically neighboring locations are also linked to each other in the map, route panoramas or panoramic views. Users can wander around connecting visual nodes by clicking a map position or by specifying a travel direction in the displayed route.
panorama, panoramic view, or oriented discrete images. We call this scheme *in-breadth access*.

In the panorama window, continuous scenes are displayed and scrolled with mouse input. At a street crossing, viewers are able to switch to another street. Between close visual nodes of different types such as streets, locations, and units, free switching between them can be realized by clicking links embedded in the various images.

As a streaming media, the route panorama can be transmitted via the Internet in real time enabling viewers to easily scroll back and forth along a route. Various rendering approaches for route panoramas described in later sections can realize virtual navigation in a city on wired/wireless Internet or even on portable information terminals.

### 3 Acquiring Route Panoramas

#### 3.1 Scanning Scenes from A Vehicle

We discuss a general projection of route panoramas for flexible camera settings. Through our investigation, we find that capturing which type of surfaces (more front surfaces or side surfaces) of architectures along a route is mainly determined by setting a proper slit. The pose or direction of the camera mainly determines the vertical field of view of the route panorama for a certain height of scenes and can be set freely.

Through a slit, the plane of sight scans the scenes during the camera motion (Fig. 5). We name it *plane of scanning* (POS). On the POS, scenes are projected towards the camera focus through a lens, which is a *thin perspective projection*. The angle between the POS and the motion vector $V$ of the camera (tangent of the path) is denoted by $\alpha$ ($\alpha \neq 0$) and is fixed after the camera is mounted.

![Fig. 5 A POS and a slit in acquiring route panorama.](image)

There are many vertical lines on architectures, and the camera moves along smooth paths on a horizontal plane. If we select the POS to be vertical in the 3D space to scan the route panorama, we will obtain many good properties either for linear or curved camera path. These properties will be discussed in the next section.

By setting angle $\alpha$ of the POS from the motion vector, we can obtain different aspect views of architectures or scenes. The more the angle deviates from sideways, i.e., close to $V$ direction with small $\alpha$, the larger the side surfaces of architectures are exposed in the route panorama. We obtain a relative *forward view* or *backward view* than the *side view* of the route ($\alpha=\pi/2$) that contains mainly front surfaces of architectures).

#### 3.2 Locating Slit in the Image Frame

After the direction of POS is determined, the vehicle is able to move out with an approximate camera setting, which is much flexible in real situation. Under the condition of a vertical POS, we can adjust the vertical field of view of a route panorama by setting the camera azimuth angle. By fix the camera axis upward, we can capture high-rise buildings.

Now, we will locate the virtual slit (pixel line) in the image frame after the camera is oriented. Locating a slit exactly at the projection of POS will produce good shapes of objects in the route panoramas. According to the constraint of vertical POS, 3D vertical lines are instantaneous scanned. This is invariant effect to the camera translation and rotation, and therefore invariant to the camera motion along a smooth path on the horizontal plane. The vertical lines in the 3D space then are guaranteed to be vertical in the route panorama. At any instance during the camera movement, the projections of the 3D vertical lines in the image frame have a vanishing point if they are extended, according to the principle in computer vision. If the camera axis is horizontal, the vanishing point is at infinity. If we name the vertical axis through the camera focus by the *position axis* of camera, the vanishing point is the penetrating point of the position axis through the image plane. It is not difficult to prove that the slit, which is the intersection of image plane and the vertical POS, should also go through the vanishing point.

![Fig. 6 Route panoramas obtained by using an upward camera in a sideways direction. SIDE VIEW: the camera](image)
axis directs upward higher than the horizon. IMAGE: vertical lines pass a vanishing point if they are extended. Horizon is lowered because of the upward camera direction. Slits are set passing through the vanishing point.

In order to preserve shapes in route panoramas, we design an algorithm to calculate vanishing point and then locate the slit passing through it. After the camera is fixed on a vehicle and the video sequence is taken along a route, we select several images arbitrary from the recorded sequence. We use edge detection to extract the projection of 3D vertical lines in the images. Then a least squared error method is used to find the position of the vanishing point where all extracted lines cross each other. Passing the estimated vanishing point, we locate the slit in the image frame and this fixed slit is used to scan the entire video sequence. Fig. 6 shows an example of locating three slits (corresponding three POS's) in the image frame to obtain forward, side, and backward route panoramas; all contain front surfaces, and forward and backward route panoramas contain side surfaces as well.

### 3.3 Shapes in Projected Route Panoramas

This section examines basic shapes of objects in the route panorama for display and street model recovery. The camera path can be described by $S(t)$, where $t$ is the time in the route panorama. We can divide a path roughly as linear, concave or convex segments depending on the sign of curvature. The vehicle speed is kept constant by a cruising system and the path is recorded by using GPS.

Under the defined projection, scenes along a street are mapped onto an image belt or surface that is swept out with a pixel line $l$ along the path. The pixel line has a fixed relation with the path. The horizontal line $h$ connecting $l$ from the path has a constant angle $\alpha$ with respect to the tangent direction of the path. The slit line has to be in the POS. The angle between $l$ and $h$ determines the vertical field of view of route panorama in order to cover possible high skyline.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Linear path</th>
<th>Curved path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neq \pi/2$</td>
<td>Orthogonal-perspective projection</td>
<td>Bended-orthogonal-perspective projection</td>
</tr>
<tr>
<td>$\neq \pi/2$</td>
<td>Parallel-perspective projection</td>
<td>Bended-parallel-perspective projection</td>
</tr>
</tbody>
</table>

Table 1 Projections of the route panorama according to the direction of POS.

Because the route panorama is generated by perspective projection along the slit direction and locally parallel projection towards a smooth path, next table summarize the projections of the route panoramas for various $\alpha$.

Normally, the path of the camera is restricted within the street. Architectures are also constructed in parallel to the street. We define the camera coordinate system $O$-$XYZ$. Any other lines can be represented as a linear combination of these linear vectors. If we denote these lines by $A$, $B$, and $C$, their projections in the route panorama from a linear path can be summarized in the following (Fig. 8).

(i) A vertical line in the 3D space is scanned instantaneously and leaves its projection in the route panorama as a vertical line

(ii) A line parallel to the motion vector of the camera is projected in horizontal in the route panorama.

(iii) Any other lines unable to be described by the above lines are projected as hyperbolic curves in the route panorama.

![Fig. 7 Projection of scenes onto route panoramas.](image-url)

Under the defined projection, scenes along a street are mapped onto an image belt or surface that is swept out with a pixel line $l$ along the path. The pixel line has a fixed relation with the path. The horizontal line $h$ connecting $l$ from the path has a constant angle $\alpha$ with respect to the tangent direction of the path. The slit line has to be in the POS. The angle between $l$ and $h$ determines the vertical field of view of route panorama in order to cover possible high skyline.

![Fig. 8 Typical lines and planes in the scenes along a street.](image-url)

It can be proved that the front surfaces of objects comprised of lines $A$ and $B$ retain their shapes, since the vertical rims are still vertical and lines parallel to the road are still horizontal in the route panorama. The distortion on the aspect ratio is from the camera moving speed and the depth of the surface. The scale of $A$ type lines is
proportional to their real length in the 3D space. The vertical scaling on a \( B \) type lines is from the perspective projection along the slit direction. The length of a \( B \) line is inversely proportional to its depth. These result in some good property in archiving complete street scenes. Unlike in a perspective view, a small tree will never occlude an entire building in the route panorama, since the horizontal scale of an object is proportional to its real width along the road.

The difference of the route panorama from perspective projection is a “curving” effect on lines in \( C \) category, which stretch in depth from the camera path. We can observe this effect in Fig. 8, and further prove that \( C \) type lines become hyperbolic curves. In the route panorama, the length of such a curve along the horizontal axis (the \( t \) axis) is proportional to its length in the 3D space (the details are omitted here).

Another characteristic to address is the convergence of parallel lines in \( C \) category. Under perspective projection, parallel lines with depth changes in the 3D space are not projected in parallel in the image frame. Their extensions in the image plane cross at a vanishing point. In a route panorama obtained from a linear camera path, however, parallel lines stretching in depth are projected to hyperbolic curves that have a common asymptotic line. Particularly, if the parallel lines are horizontal in the 3D space, their asymptotic line in the route panorama is the projection of horizon.

4 Rendering Route Panoramas

4.1 Panoramic Traversing Window

We will display the acquired scenes for users to traverse the cyber city back and forth. This section discusses how to transmit a long route panorama on the Internet and to seamlessly scroll street scenes.

When a route panorama extends to several miles or tens of miles, it is unwise to download the entire route panorama images and then display them on a web browser. Also, the portability of the route panorama on PDA and wireless phone requires an efficient rendering algorithm. We are developing a streaming data transmission function that displays route panoramas during data downloading. This gives users quick response for free maneuvering of a route.

Besides the simple display of route panoramas as route profile (Fig. 2) and a route image scroll that simulates the side window of a sightseeing bus, we develop a panoramic traversing display of the street for virtual moving along a route on the WWW. A normal perspective view has a too narrow field of view. For a navigation task in which a user has to look around, switching between surrounding scenes dynamically may not retain a fast rendering speed and affect the perception of spatial relation of scenes. We will display a wide field of view in the panorama window (Fig. 3).

For a traditional panoramic view, i.e., surrounding scenes are projected on to a cylindrical retina, horizontal structure lines in the 3D space appear as sinusoidal curves in its opened form. Vertical lines in the 3D space stay as vertical lines. We design a panoramic traversing display to show approximately half circle (180 degree) of scenes in a direction selected freely by a viewer. We can imagine that the scenes in the route panoramas are first mapped onto walls on two sides of streets in the city (Fig. 9). If the path turns along a curved road, the walls are curved accordingly based on the city map. The two walls are then projected onto the cylindrical image surface (Fig. 10) and we display the opened 2D form of it (Fig. 11).

![Fig. 9 Every street is virtually bounded by two walls mapped with route panoramas.](image)

![Fig. 10 Panoramic traversing window displays route scenes.](image)

The panoramic traversing window can provide following kinds of motion.

*Rotation*: Viewers can rotate the half visible period by mouse to view two sides of the street stretching forward, or rotate towards sideways to see shops and buildings on one side of the street. We can have a smooth rotation and switch between four modes (forward, left side, right side, backward).

*Translation*: Viewer can translate along the path, while scenes move from one end of the street to the other in the field of view specified by viewer. Sideways translation towards side buildings can also be realized by scaling the route panorama locally.

*Field of view*: we can control the horizontal field of view from half circle to a larger area. This will provide chance to select between high rendering speed and wide field of view. However, the user’s viewing direction is always kept at the center of the panorama window.
Rendering rotation in the panoramic traversing display is simply a horizontal shift of the current displayed image, attaching appearing part and discarding disappearing part. This generates a horizontal optical flow equivalent to a rotation of the cylindrical panoramic view.

The translation is rendered at each instance by first moving the viewer’s position along the street, and then projecting route panoramas on the walls towards the panoramic traversing display. This will generate optical flow along sinusoid curves expending from a vanishing point at one end of the street, which has the same direction as the physical translation appearing in the cylindrical panoramic view. However, the displacement of each point on the sinusoid curve is not true because scenes are mapped onto the walls and the real depth of scenes is unknown.

The front part in Fig. 11 has no visual information because the route panorama primarily side views of the street with the scanning function. We can consider it as the floor or ceiling areas of a virtual traversing vehicle and display some control buttons and information boards on them, or paste there a static sky image which is translation invariant and only shifts with scenes when the viewer selects to rotate the viewing direction in the street traversing.

### 4.2 Mapping from Route Panorama to Panoramic Traversing Window

Now, let us calculate the mapping from the route panorama to the traversing window. Assuming the distance between the two street walls is $W$ and the horizon is at height $H$ in the route panorama. The camera path is characterized by camera traveled distance $S$. The route panorama is mapped towards the current position $S_0$. A point $p(t,y)$ on the route panorama is mapped to $(\phi, \mu)$ in the panoramic traversing window, where $\phi \in [-90,90]$ is the angle from the moving direction. The coordinates $(\phi,\mu)$ can be calculated by

$$\phi = \pm \tan^{-1} \frac{W/2}{S - S_0}$$

$$\mu = \frac{Y \sin \phi}{W/2} + H = y \sin \phi + H$$

where $S \in [S_0, \infty]$. According to these equations, rendering a vertical pixel line in the window is done by scaling and pasting its corresponding line in the route panorama, which achieves a fast speed in displaying a continuous forward motion. Although the traversing window is not a true 3D display, major portions in the panoramic traversing window have the similar optical flow as that from real 3D scenes. We call it pseudo 3D display.

We capture three route panoramas on each side of the route, forward ($\alpha<\pi/2$), backward ($\alpha>\pi/2$) and side route panoramas ($\alpha=\pi/2$) as depicted in Fig. 6. For switching between forward, side and backward traversing, we select a proper set of route panoramas to display so that users can observe side surfaces of architectures along the route. Although this approach will increase the storage of route panorama, only one set of route panoramas is transmitted at any time over the Internet and this will not affect the display speed.

### 5 Experiments and applications

We have succeeded in capturing route panoramas along various streets and in many areas, using vehicles, trains, and ships, and they provide continuous and complete scenes in a compact data format. The obtained results are very important for the archiving of street scenes in the real space.

We are now working on entire campus of our university (Fig. 1(b)) and creating a database of route panoramas (130MB) extracted from a totally 110 minutes video. The vehicle speed is kept at 20mph in the scene scanning. According to our calculation, the route panorama increases 6MB per mile approximately. The web environment is designed for indexing and virtual navigation in the area.

During the route scanning, the vehicle may shake on an uneven road. The camera may have vertical translation due to bumping and have rotation in pitch due to left-and-right swing. If the architectures and objects are far away from the camera path, the translation component does not affect the quality of the route panorama critically. The rotation in the camera pitch may cause zigzags on structure lines of objects in the obtained route panorama. We have developed an algorithm to reduce such zigzag components along the horizontal structure lines on the architectures.

A broad range of applications can be considered by using our mapped cityscapes in the cyber space. The scenes linked from map can be used in real estate to find a house or environment in the city. By providing route panoramas, a residential division could be visualized more densely than a series of discrete images. Our modeling and rendering techniques can be used in some historical cities or towns to archive all the scenes there faithfully and completely for heritage research and exhibition. If we extend the area to an entire city, an address searching accompanied with visual information can be realized on the net; a visitor will not only find a map reaching the address, but also be able to follow the visible scenes towards the address. This will make many business and culture activities possible to be done in cyber space.

### 6 Conclusion

This paper discusses general mapping techniques to project cityscapes to a cyber space for virtual navigation and indexing. We use various mapping, particularly a new image representation — route panorama in representing the cyber city. We introduce the projection of route panorama, the acquisition approach, the generated shape in the route panorama, and the approach to render the route panoramas on WWW. A route panorama registers complete route scenes in a seamless format with a small amount of data, which is very useful for indexing and navigation of an entire city in cyber space. We have transmitted and rendered route panoramas in real time and achieved a virtual tour in an urban area. The route panoramas can even be displayed on portable devices for various digital city applications.
Acknowledgement

The authors would like to thank SBC Ameritech and Purdue Summer Research Grant for the support of this project.

References