Pervasive Views: Area Exploration and Guidance Using Extended Image Media

Jiang Yu Zheng Xiaolong Wang
Department of Computer Science, Indiana University Purdue University Indianapolis
723 W. Michigan St., SL280, Indianapolis, Indiana, IN46202, USA
1-317-278-2365
jzheng, xwang@cs.iupui.edu

ABSTRACT
This work achieves full registration of scenes in a large area and creates visual indexes for visualization in a digital city. We explore effective mapping, indexing, and display of scenes so that an area becomes "visible". Users can virtual navigate city on the Internet and achieve real guidance with a PDA. Extended images such as route panoramas, scene tunnels, panoramic views and spherical views are acquired in an urban area and associated with geospatial locations. A 3D LIDAR elevation map is used to generate a scanning plan based on visibility, image properties, and importance of scenes. Scanning scenes along streets and at spots of interest allows for compact and complete visual data collection. To access city information, visual indexes from scenes to spaces are created pervasively for flexible space exploration and transition. To visualize a space seamlessly in a large view frame and synchronize scenes with the virtual movement in the map, we stream image data on the Internet. An engine is developed for continuous space traversing, accessing spatial information, and transiting between spaces through visual links. A real urban area has been modeled to verify the effectiveness of such a system.

Categories and Subject Descriptors
I.4.1 [**Image Processing and Computer Vision**]: Digitization and Image Capture – *scanning, imaging geometry, sampling*. H.2.4 [**Database Management**]: Systems – *multimedia database*. H.4.3 [**Information Systems Applications**]: Communications Applications – *Information browsers.*

General Terms
Algorithms, Measurement, Design, Experimentation, Theory

Keywords
Route panorama, scene tunnel, image media, visual indexing, navigation, pervasive computing, image streaming, GIS.

1. INTRODUCTION
Recent digital cities have combined heterogeneous data including text, map, image, video and model for business and social activities such as urban planning, virtual tourism, location finding, e-commerce, real estate, geo-referencing, heritage archiving, museum, learning, training, and crisis management. Landscapes recorded in multimedia will convey rich contexts for city traversing. However, navigating on the web seamlessly is still difficult because the visual data collected have not been full archives of physical spaces. Capturing scenes pervasively requires tremendous work and huge storage space. Current media only highlights partial spaces for virtual sightseeing, which is inadequate for location finding and guidance. Pervasiveness provides value to the data set for city exploration and assessment.

To present spaces, maps/satellite images are mostly used, from which snapshots and panoramic/spherical images have been indexed to wide and open locations [1][8][13]. Efforts have been made to register scenes along streets with a 2D image series [3]. To add the continuity, video clips have been captured [2][14][19], from which walkthrough views can be generated [9][10]. Video recording and view extraction have not been extended to large areas because of the large data size. In urban environments visible from the ground, mosaicing translational images has difficulties in stitching scenes with inconsistent disparities from depth variations [6][11][20]. Therefore, the slit scanning approach using a single pixel line was proposed to avoid inter-frame matching [4][5][24].

On the other hand, 3D urban models from LIDAR data have been textured with ground based images [21]. A photorealistic model requires taking images exhaustively, which is laborious manual operation. So far, high-resolution models are usually local, while global models are very coarse. A large data set for the 3D model slows down the data transmission on the Internet.

The goals of this work are to acquire real scenes pervasively in urban areas using extended image media such as route panoramas [4], scene tunnels [5], panoramic/spherical views [8][13] and digital images and realize web-based geospatial information visualization (Figure 1). Functions such as presenting views at dense locations, navigating routes with continuous views, and providing static/dynamic information associated to scenes can be realized on the Internet for remote access, and on PDAs for on-site area guidance. Although the extended images have 2D distortion for their specific projections, they are

- Compact: include less redundant scenes than video.
- Complete: cover every street and many locations.
- Continuous: long route scenes suitable for navigation
- Comprehensive: indexes from/to maps and spaces provide flexible space transition and traversing.

The contributions of this paper are as follows.

To obtain complete visual archives of scenes with fewer images, 3D LIDAR data are used to plan the view acquisition based on the significance of viewpoints. The scene distribution and visibility, as well as the image properties are taken into the consideration. The viewpoints and routes with high significances are selected for scanning in order and a large area is scanned.

To ensure an effective access of an area, we design a scheme to enable *view-to-space* indexing. When a viewer virtually moves into the scope of a scene, its information is superimposed onto the scene, and the related space is further accessible through interaction. The combination of view, text and map information enhances the space perception. Inversely, the pervasive views create a new paradigm for space indexing and understanding.

To realize efficient virtual travel, we transmit and display scenes continuously over the Internet to respond to the viewer's virtual movement. Various functions are prepared to facilitate transitions from space to space. The data can further be downloaded to a PDA or wireless device for interactive guidance as visitors walk through a real space.

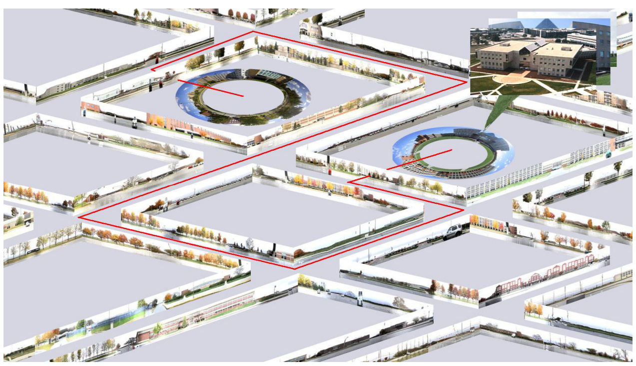


Figure 1 A visualization model of urban area by using extended images. Real scenes are projected towards a grid of paths, spots and images and are archived in database.

The following section covers the view planning, and section 3 introduces the area scanning. Section 4 discusses visual indexing, and section 5 focuses on view streaming for exploration and guidance. Section 6 provides results on experiments.

2. PLANNING VIEW ACQUISITION

2.1 Significance of View Points

Before starting image capturing and scanning, we plan the locations and routes by estimating the *view significance* based on a LIDAR elevation map. A location with a large amount of horizon is not considered as significant as a location full of scenes delivering more urban information. A view covering a large space from an overlook is more significant than a view at a narrow valley of buildings in telling global locations. For those building surfaces rarely visible in the environment, the planning may exclude them in the acquisition. This is different from a graphics model in which a surface missing texture will affect the reality. We define the *view significance* at a viewpoint to evaluate its sight coverage, i.e., how many, how large, and how far the 3D scenes are visible from that point. This estimation is evaluated over all positions on the ground and the panoramic views will be placed at the positions with high output.

The view evaluation at a spot is determined from scenes, rather than image sizes. Figure 2 shows the idea of computing the view significance from a view shed. Denote a visitor reachable position in the space as $P(X,Y,Z)$. The ray from it is $\boldsymbol{n}(\phi,\varphi)$, where $\phi\in[0,2\pi]$ is the orientation and $\varphi\in[-\pi/2,\pi/2]$ is the azimuth angle, respectively. If the ray hits an object surface at distance $D(\varphi,\phi)$, a sign function, $\lambda(\phi,\varphi)$, takes value 1, and otherwise 0. We define significance $\Sigma(P)$ at the viewpoint according to the total areas of visible surfaces, and calculate its value from LIDAR data as

$$\Sigma(P) = \iint_{\phi,\varphi} w\lambda(\phi,\varphi)\frac{D(\phi,\varphi)}{D(\phi,\varphi)+D_0}d\phi d\varphi \qquad (1)$$

where w is the weight of importance assigned to a surface in a scene (building, tree). D_0 is a large constant and the denominator function describes the image quality degrading on distance scenes due to intervening atmosphere. It avoids a close-to-infinite scene to be integrated largely. Generally, weight w can take a uniformed value unless some important façade and symbolic features need emphasis. The weights are assigned building-wise in the LIDAR data. The significance value is high at wide sites surrounded with large and high buildings. Figure 3a shows the significance map of all the viewpoints in a large area.

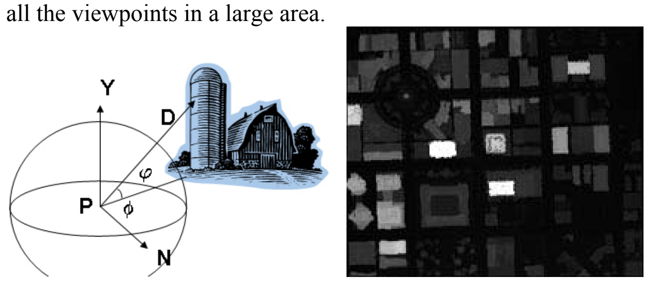


Figure 2 Computing view significance from the view shed at a point using LIDAR data. The intensity in the LIDAR elevation map represents the building height.

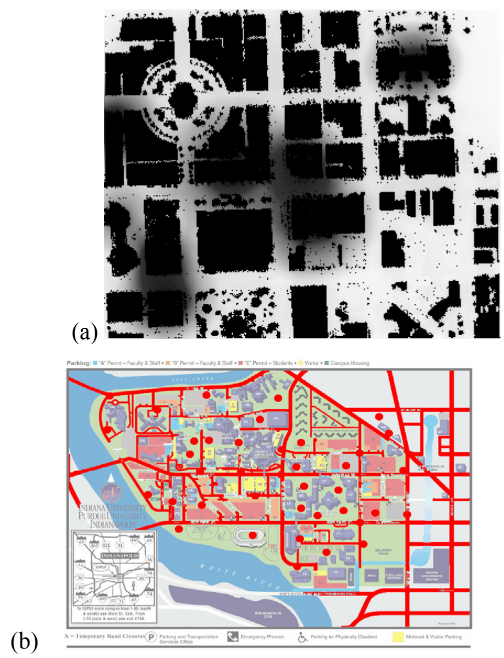


Figure 3. The view significance evaluation at all positions in an area. (a) Significances of viewpoints at height of 2m from the ground in the urban area shown in Figure 2. Buildings and small trees are assigned with zero value. (b) Planned routes and spots to take views in a campus.

One can notice that squares and parks have higher significance measure than narrow streets. The positions close to

(a) Depth map of buildings from a camera path is predicted from LIDAR data (intensity is inversely proportional to the depth).

(b) Blocks of scene tunnel taken along the real street. Parallel-central projection is employed.

Figure 4 Scanning scene tunnel according to predicted depth from LIDAR data. Vertical axis indicates azimuth angle of line of sight and the horizontal axis indicates the route distance.

high rises have low significance value. The view significance measure is different from the visibility calculation in computer vision and graphics. It is an implicit value continuously distributed on the ground, rather than a binary value determined from object aspects near occluding edges. We can add a weight to an important surface to recalculate the view significance map. The calculation can be implemented from coarse grids to fine positions in order to save the computation in evaluating a large space.

2.2 Significances of Routes

Although the vehicle has less flexibility within a lane or road, we still can obtain the significance of a route for scanning scene tunnels. Denote a ray **n**(s,ϕ) in the *Plane of Scanning* at position *s* [5], it hits a point in the space at distance $D(s,\varphi)$. The view significance measure at position *s* is

$$\Sigma(P(s)) = \int_{\varphi} w\lambda(s,\varphi)\frac{D(s,\varphi)}{D(s,\varphi)+D_0}d\varphi \qquad (2)$$

where $\varphi \in [-\pi/2, \pi/2]$ is the azimuth angle of the ray. The significance of an entire street is evaluated as

$$\Sigma(S) = \frac{\int_s \Sigma(P(s))ds}{S} \qquad (3)$$

where S is the length of the street and $s \in [0, S]$. Figure 4 shows the depth map of scenes along a section of street calculated from LIDAR data. Very open routes with many highways are not significant in the computation for their monotonic scenes.

After the view significance map is estimated for areas and routes, we place panoramic views at local maximum positions and route panoramas at streets with rich scenes on two sides ($\Sigma(S)$ is high), as depicted in Figure 3c.

3. PERVASIVE SCENE ACQUISITION

3.1 Street Scanning and Image Properties

After the locations are selected, panoramic views are taken up to a maximum azimuth angle Φ using a fish-eye lens and a high-resolution digital camera. If Φ is close to π, a spherical view is obtained [19]. Orientations are marked in color in the view in order to provide correspondence with that in the map. In addition, discrete images are taken for the scenes that need to be particularly emphasized.

We acquire route panoramas by driving through streets and scanning scenes continuously with a pixel line in the image frame of a video camera. This is much more efficient than taking static images at discrete positions and then merging them. A vertical *plane of scanning* through the camera focus intersects the image frame to determine the slit. Such a plane of scanning preserves the shapes in the route panorama; the projections of vertical structural lines on architecture are vertical in the route panoramas and scene tunnels. The location of a slit is calibrated by using the sampling images in which vertical buildings are included as the vehicle is parked on a horizontal plane [5].

If we choose a plane of scanning non-orthogonal to the camera moving direction, the route panorama and scene tunnel I(s,φ) capture side surfaces in additional to front façades of architectures. Defining the angle between the Plane of Scanning and the vehicle moving direction by α, Figure 5 shows a section of route panorama scanned with α=45°. Such a route panorama not only includes more visual information of crossing streets than an open sky, but also it is suitable for displaying the forward aspect views of the vehicle in visualization. In addition, for a point on the side surface, its depth $Z(s,\varphi)$ along the crossing street is proportional to the distance to the building corner in the route panorama, if the side surface is orthogonal to the street the vehicle is moving on. We can thus measure the depth of scenes stretching in the crossing streets in the route panorama.

For an appropriate vertical FOV, the camera uses a zoom lens and the focal length is adjusted to keep Φ within the image

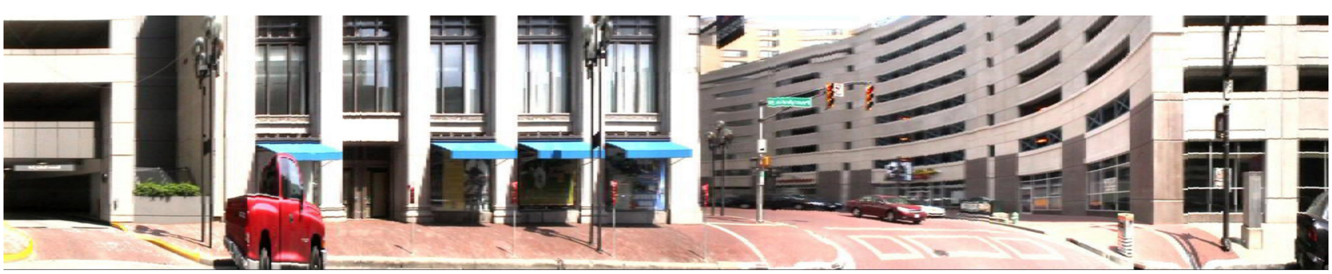


Figure 5 A segment of route panorama with front and side facades of buildings. The Plane of scanning is forward in about 45° from the camera moving direction.

size. If Φ is chosen to be π, a scene tunnel is obtained with a fish-eye lens. The height is kept the same while the real angular coverage of a pixel is extended, which may yield more stationary blur as described below.

An ideal route panorama and scene tunnel employ parallel-perspective projection and parallel-central projection, respectively. An object has an absolute width horizontally regardless of its depth from the path, which is different from perspective projection in which distant objects are smaller than close ones. Thus, distant objects in the route panorama look wider than in normal images, which give a different impression. In addition, because of this horizontal extension of distant objects, high contrast scene features such as vertical edges are horizontally blurred if they are scanned with a low-resolution camera (or equivalently wide-angle lens, large spread cone of a pixel).

3.2 Scene Data Processing and Image Quality

The horizontal blur in the scanned route panorama and scene tunnel is named *stationary blur* [4][22], which is a temporal blur as a counterpart of the motion blur in a spatial image. Different from the motion blur appearing on close objects during a long image exposure, the temporal stationary blur appears on distant scenes because they move across the scanning pixel line slowly. The stationary blur also appears at the concave side of a curved camera path, and is more obvious when the vehicle speed slows. The sharpness degrading due to the stationary blur adds more atmospheres to distance in addition to the reduction of object size. If we define the vehicle velocity by V, the curvature of the path by κ, the camera focal length by f, the depth of a scene by Z, the original image differential of an edge by I_x, the horizontal differential, $I_s(s,\varphi)$, in the route panorama can be proved to be

$$I_s(s,\varphi) = I_x(s,\varphi) f V(s) \left(\frac{1}{Z(s,\varphi)} - \frac{\kappa(s)}{\sin^2 \alpha} \right) \quad (4)$$

where $Z(s,\varphi)=D(s,\varphi)\cos\varphi\sin\alpha$, k>0 on the concave side and k<0 on the convex side of the path. This implies a short focal length (wide-angle lens) will blur distance scenes (Z is large). Generally, the camera resolution and sampling rate are fixed and f is set before street scanning. If scenes along a route are distant, their heights are normally low. This allows us to select a long focal length to preserve the scene contrast in the route panorama. Figure 5 shows the blur on the side surface as the depth increases.

The vehicle speed is kept as constant as possible. We have succeeded in capturing route panoramas for long distances, with piecewise constant speed. We normalize a route panorama to a length proportional to the real street length according to GPS output or satellite images. For GPS based normalization, the output data are used to rescale vertical patches of the route panorama. Alternatively, by aligning the route panorama sections with satellite images available on Internet web services, we match features such as building rims, crossing streets, trails, trees in both

Figure 6 A route panorama in color taken along an urban route with more than twenty blocks in Indianapolis, IN46202 [24].

types of images to adjust the section length of route panoramas. Features close to the path are more precise than distant ones in the length adjustment, because close features are less influenced from the stationary blur, and the horizontal variation of the vertical plane of scanning due to the vehicle deviation from an ideal path.

Another problem to solve in generating visual archives is to stabilize the route panorama obtained over a bumpy road. View stabilization is applied to the jittered camera to obtain a better route panorama. Figure 6 shows an example of a long route in an urban area. We extract near horizontal edge lines on architectures and use multiple median filters to smooth the vertical coordinates of the lines. We shift jittered parts on the horizontal structures vertically by adjusting the sampled pixel lines. The condition is the existence of man made objects in the scenes.

The parked vehicles are projected as static objects. The moving vehicles, however, have no fixed lengths in the route panorama. A fast passing vehicle in a very short period leaves a narrow width in the route panorama. Such a moving object can be detected through the image flow calculation around the sampling pixel lines. The narrow vehicle shape can be extended to its real size after its shape is lined out. Alternatively, it can be removed in the rendering either by pasting a standard vehicle pattern, or by filling with background colors. If a vehicle stays aside with our scanning vehicle, it will drag a long pattern in the route panorama and occlude many street scenes behind, which ought to be avoided in the scanning.

In our city media database, route panoramas are segmented to images of 1000 pixels and are indexed from streets. The non-redundant route panoramas significantly reduce data from video. It can be further compressed to JPG or GIF images. The heights of input route panorama and the scene tunnel can reach 640 and 1000 pixels, respectively. Considering the loss of quality due to the camera shaking, the limited window size of a PDA, and moderate window size displayed side-by-side with a map, we squeeze images to 240 pixels in height. The lengths of images are proportional to the street length. Even though, the image resolution is about 10 times finer than the highest resolution satellite images in the Google map. A mile route panorama takes about 7MB image data including both side of the street.

4. VISUAL INDEXING OF SPACES

4.1 Map-to-view Indexing

After obtaining continuous scenes, flexible links are created for visitors to go into a building, move through a street, and look around a location, on the web and with a PDA. We display local scenes for viewers to transit from place to place. The route panoramas and panoramic views are extendable from major streets to small roads, and newly added streets are linked to the existing database in the same data format. Here we design a scheme of indexing for incremental area data collection.

We define three types of spaces: spot (0D), route (1D), and area (2D), respectively. They are associated with different views such as digital image, panoramic/spherical view (0D), route panoramas or scene tunnel (1D), and aerial image or distributed images (2D). A high dimensional space includes lower dimensional spaces in it. For example, a route (1D) is connected with street crossings (0D). An area (2D) contains many routes and spots of interest.

Two styles of space access are implemented. One is the in-space traversing and the other is inter-space transition. The space traversing shows continuous route views as the viewer moves along a street, or rotates the panoramic view as the viewer changes his/her orientation. The transition between spaces switches the views completely in the display. *View-to-space* indexing links views to various spaces for transition during the virtual exploration.

For route traversing, a *route-to-image* index, as a type of *space-to-view* indexing, indicates numbers and locations of the route panorama segments, i.e.,

$$R_j \rightarrow \begin{pmatrix} n_l^j, img_1^{jl}, img_2^{jl} \ldots, img_{n_l^j}^{jl} \\ n_r^j, img_1^{jr}, img_2^{jr} \ldots, img_{n_r^j}^{jr} \end{pmatrix} \qquad j=1,2,3 \ldots J \qquad (5)$$

where n_l^j and n_r^j are the numbers of segments on both sides of the *jth* street, $img_1^{jl} \sim img_{n_l^j}^{jl}$ and $img_1^{jr} \sim img_{n_r^j}^{jr}$ are URL links or relative paths to the image segments. All streets are coded from east to west and from north to south, e.g., *Michigan St.* is as follows.

n_l=15, img_1="RP/Michigan/south/image0001.jpg", ...,

img_{15}="RP/Michigan/south/image0015.jpg"

n_r=16, img_1="RP/Michigan/north/image0001.jpg", ...,

img_{16}="RP/Michigan/north/image0016.jpg"

which are saved in text files for small areas and in database for large areas.

We also use *map-to-space* index to achieve space traversing and transition. When a map is clicked, spot $\boldsymbol{P}_i$(X,Z), $i \in [1, 2, 3, \ldots, n]$ is searched. If the number of spots increases significantly, a quad-tree can be used to reduce the searching speed. A spot is either associated with a panoramic view, or is a street crossing with links to neighboring crossings. Further clicking on an adjacent crossing determines a street $\boldsymbol{R}_j$ $j \in [1,2,3,\ldots,J]$, travel direction, and viewer's position on the route. Selecting consecutive crossings yields a street or even a route connecting multiple streets to traverse. Accordingly, an indexing from spots to adjacent streets and links of panoramic views is prepared

$$P_i \rightarrow [X, Z, panorama\_view_i, R_{j1}, R_{j2}] \qquad (6)$$

where $j_1, j_2 \in [1,\ldots,J]$ and $i \in [1,2,\ldots n]$. If a spot is not at a street crossing but at an open space, R_{j1} and R_{j2} are nearby streets to reach the spot.

Area traversing using map or aerial/satellite images has been realized in map service sites. Clicking a 2D space $\boldsymbol{A_l}$, $\boldsymbol{l} \in [1,2,3\ldots,L]$ in the map, a detailed page of buildings, houses, or sites is displayed. Different from normal facility pages, links from the 2D spaces back to the streets and spots are prepared so that viewers can start a trip or return to a traveled route from the buildings and sites.

4.2 View-to-space Indexing

Conventional multimedia systems have employed *text-to-view*, and *map-to-view* indexes. In this work, *view-to-space* and *view-to-text* indexes are added to the geospatial information system, which has not been widely used because the visual data were not a complete digest of a real space. The sequential route panoramas work as ideal *visual* indexes of information along streets. As illustrated in Figure 7, we embed the links in the route panoramas for information such as:

(i) Geospatial data: address, direction, building name, name of adjacent street, size/area of architecture, function (shop, school,…), as *view-to-text* index in the route space.

(ii) Dynamic data: parking and hotel vacancy, traffic flow, advertisements from a database.

(iii) Links: as *view-to-space* index, connect to high-resolution digital images, restaurant menus, crossing streets, and web pages of facilities, buildings, and houses for flexible space transition.

These data associated to locations on the streets are superimposed onto the views in the dynamic display. Along the *jth*-route, we define *connectors* $\boldsymbol{B}_k$, $k=1,2,3,...,K$ from the route panorama to visible spaces. That is

$$R_j \rightarrow \left(B_1^j, B_2^j, ..., B_k^j, ...B_K^j\right) \tag{7}$$

$$B_k^j \rightarrow \left(side_k^j, img_k^j, s_k^j, e_k^j, text_k^j, link_k^j, loc_k^j, pos_k^j\right)$$

where the scope of $\boldsymbol{B}_k$ in the route panorama is registered by s^j_k and e^j_k in pixels. The information about space k is in $text^j_k$, and displayed in the view frame if the viewer enters scope $[s^j_k, e^j_k]$ in segment img^j_k. If $\boldsymbol{B}_k$ connects an adjacent route (*1D*) with a link registered in $link_k$, e.g.,

B_3 = (*right*, 3, 120, 210, *University Blvd.*, *university, 12, 564*),

the current route display can switch to space $\boldsymbol{B}_k.link$ at destination segment loc_k, and pixel position pos_k. If $\boldsymbol{B}_k$ connects to a space other than routes, an extra page $\boldsymbol{B}_k.link$ is popped up, when the $\boldsymbol{B}_k$ is clicked. An example of $\boldsymbol{B}_k$ can be

B_6 = (*left*, 7, 320, 870, *Indiana Museum of Art*, *www.ima-art.com*)

Similarly, visible spaces around a spot can be linked from a panoramic view as

$$P_i \rightarrow \left(B_1^i, B_2^i, ..., B_m^i, ...B_M^i\right) \tag{8}$$

$$B_m^i \rightarrow \left(s_m^i, e_m^i, text_m^i, link_m^i\right)$$

where $[s_m, e_m]$ is the orientation of the *mth* space visible in the *ith* panoramic view. Inversely, following the links to the arrived spaces, the viewer can always trace back to the previous spaces. The adjacent spaces of a street are organized in a text file. As the viewer transits to a new route (1D), a new *view-to-space* indexing file is loaded with the images for updating the current display.

Figure 7 Indexing spaces from route panoramas

The *view-to-space* indexes are pervasively embedded in the views so that dense inter-space links facilitate free transitions across spaces. Figure 8 depicts a diagram of bi-directional visual links between different types of spaces. One can turn from street to street through 1D-1D links, enter and exit buildings through 1D-2D links, move to spots through 2D-1D links, etc.

Figure 9 displays a prototype of a pervasive view system implemented in JAVA. Components such as *map*, *view*, and *text* frames are displayed side-by-side, and they are synchronized according to viewer's position $\boldsymbol{P}$(s) interpolated between street crossings $\boldsymbol{P}_i$ and $\boldsymbol{P}_{i+1}$ that are indicated in red dots in the map. The

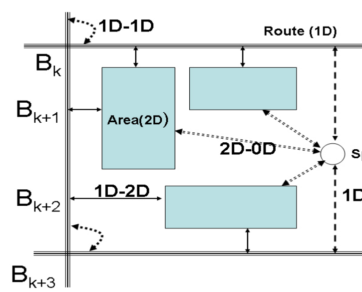
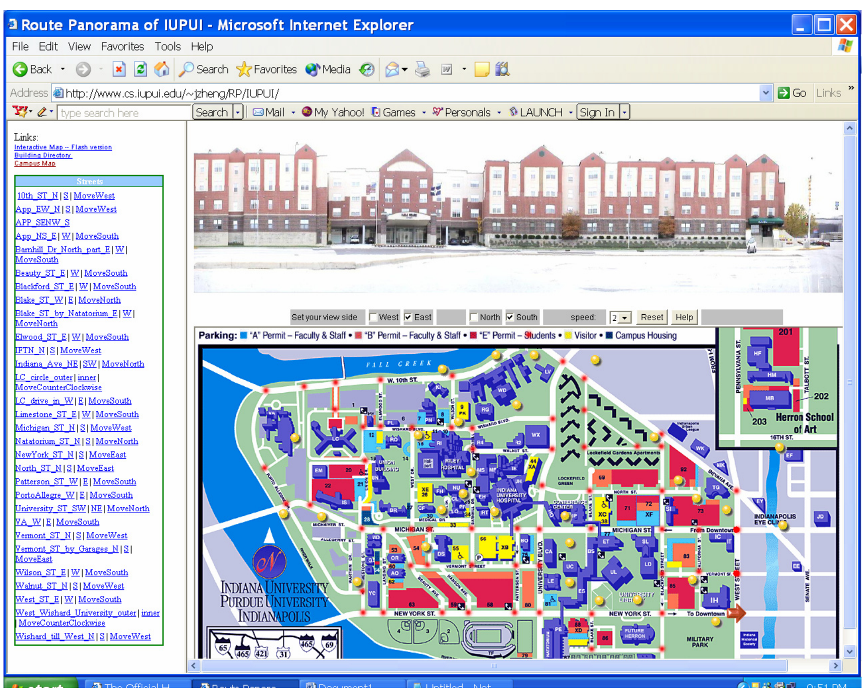


Figure 8 Interconnection of spaces through various visual links.



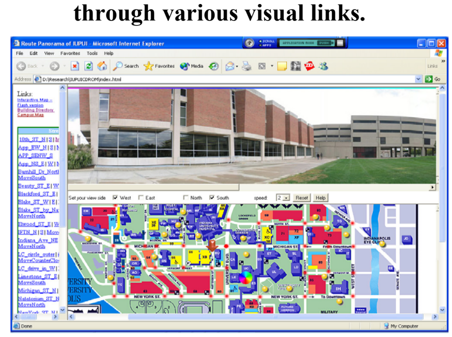


Figure 9 A pervasive view system of a campus with map, text, and view frames [24]. (Left) Route travel in the map synchronized with the corresponding scenes in the view frame. (Right) Clicking a spot in the map (a color disc) displays a panoramic view in the view frame. The orientation is shown in the *view* and *map* in corresponding color.

← Crossing street is notified.

Zoom in	Move Forward Speed up	Zoom out
View left	Stop	View right
Turn left	Move Backward Speed up	Turn right

Figure 10 Traversing display of a route shows dynamic scenes in the view frame with functions of rotation and translation.

viewer's location and moving direction are obtained through clicks on neighboring crossings on the map, and the viewer's direction is selected with buttons. Clicking consecutive crossings provides a continuous move along a route at a speed of 2~64 pixels per frame cycle. An arrow on the map indicates the viewer's position and orientation during the smooth movement, and route scenes are scrolled in the view frame. The viewer can move back and forth as long as the selected street crossings are consecutive. This realizes a free *map-to-view* indexing. Besides route traversing, yellow balls on the map show spots where panoramic/spherical views are associated. The route can also be selected from text menu, where the travel speed and viewing direction can be controlled freely.

This system is extendable to larger areas if the data are prepared in the same format. The synchronized position and the view help the viewer understand environments in their virtual exploration. The visual data provides an intuitive way of finding scenes and locations, and the text indexes further enhance presentation of city information.

5. SEAMLESS CITY EXPLORATION

5.1 Streaming Scenes on a Network

An important aspect of this work is to transmit route scenes progressively over the network. Unlike a video played with fixed speeds, the transmission of route images using JAVA is more flexible in changing display speed, direction, and route.

In order to increase the portability of the system on a variety of platforms, a pseudo 3D rendering approach is employed here [5]. As a viewer moves along a route, the route scenes are scrolled in the view frame according to viewer's interaction. The viewer can select one side of a route for display as shown in Figure 9a, or two sides as in Figure 10. On a two-side display, route panoramas are warped onto a cylindrical view of half period (180 degree), and its opened form is displayed in a wide view for navigation tasks. This also emphasizes the side scenes scanned correctly from sideways. If the route panoramas covers up to an average height of scenes, some building tops will be cut off. The view is like a bus window with ceiling part available for text display. For scene tunnels with entire heights, a display with a global panoramic view is shown in Figure 11. Viewers can move forward or backward, change translation speed, turn left or right, and look left or right by clicking sub-regions in the view frame. These actions are rendered simultaneously so that the viewing direction can be different from the moving direction, and the angle between them is kept during the turning action. The viewer is a rider or walker rather than a driver. Spaces along the route are also displayed with names or addresses, and viewers can enter them by clicking the turning action.

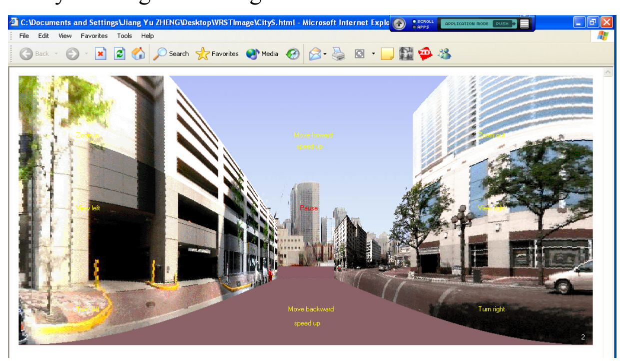


Figure 11 A scene tunnel display of a street.

During the route traversing, consecutive segments are transmitted. The arrived image segments are connected again for seamless display. Figure 12 shows the streaming segments on one side of a route. At any instance, four image segments are downloaded to memory sections T0, T1, T2, and T3. Each image segment is longer than the view frame. Hence, at most two image segments (in T1, T2) are involved in rendering at any time. As T1, T2 sections shift in the view frame, the following section T3 is buffering by another thread. It will move in the view frame, after confirming it has finished buffering. Otherwise, the display waits for completion of the image transmission into T3. Similarly, moving in the inverse direction is implemented simply by buffering T0 at the other end.

The rendering program uses an intermediate layer to store the temporal image warped from memory sections T1, T2. The rotation and zooming is rendered by scaling and translating a view port of the intermediate layer. We denote the viewer's speed, orientation, zoom (scaled to view frame) by *v*, *ϕ*, and *f*, respectively. The diagram of streaming data for a forward moving is as follows:

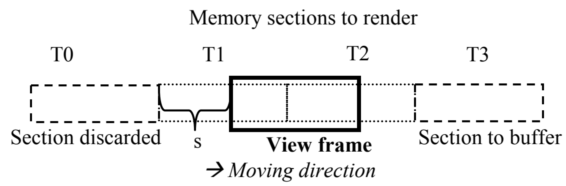


Figure 12 Streaming route panorama on the Internet

```
Set_viewer_parameter(v, ϕ, f)
Mainloop:
While (v≠0){
    s ←s+v;    // s: passed distance of the beginning position of T1.
    if(T1.width = s){       // section change, leftward move is similar
        T1 ←T2, T2 ←T3, seg_num ← seg_num+1
        Buffering(T3, seg_num)
        s ← 0;
    }
    visible_area ← clipping(T1, T2, ϕ, f);
    Rendering(T1.[s, T1.width] & visible_area);
    Rendering(T2.[0, frame.width - T1.width+ s] & visible_area);
    if(s∈range_of[B] & seg_num=B.seg)
            show(B.text)                    // B: scene to space connector
    if(s move_out_range_of[B])
            B = Spacequeue[space_pointer++];
}
```

The backward movement is symmetric to this operation by changing the position increment to decrement, and the opposite side of the street is scrolled by changing the parameters accordingly. For two-side route panorama streaming, five sections are involved on each side. Three middle sections T1, T2, T3 are for rendering and another two sections, T0, T4 on both ends are for buffering, since more distance scenes are visible along the street when the viewer translates back and forth. Using the same data sequences and the viewing left and right functions ($\phi>0$ or $\phi<0$), along with the translation speed ($v>0$ or $v<0$), the viewer can move on the same street in the opposite direction. The viewing direction change is realized by pasting the view port from the intermediate layer to the view frame according to the viewer's interaction as

```
    v, ϕ, f ←Event_processing(interaction)    //non-turning
    if((ϕ  or f ≠0)  and  v=0){
        visible_area ← clipping(T1, T2, ϕ, f);
        Viewport(T1.[s, T1.width] & visible_area);
        Viewport(T2.[0, frame.width - T1.width+ s] & visible_area);
    }
    Retern to Mainloop      //updating screen
```

If a viewer selects a turning action at a crossing, the route image segments of the adjacent street will be loaded. Through the index from *crossings* to *streets* and then to image segments, the image segments of the next street start transmission to memory sections T1, T2, T3 and T4 by the data streaming thread. Meanwhile, the main display performs the viewing direction change at the crossing and optional zooming in. Such an action helps the viewer understand the direction and make a smooth connection between routes. The rendering is from the intermediate layer to the screen so that the buffering of the route segments to T1~T4 is not visible. If a turn action is not at a crossing, but at the front of a space, e.g., at a building front, the same turning action is performed and a window pops off to show the web contents connected from $\boldsymbol{B}_k$.*link*.

```
If(s∈range_of[B]){
    If(B is a separate space) Pop off a new webpage(B.link);
    If(B is a route){
        crossing ← GetCrossing(route, B)
        next_route← FindNextRoute(B.link)
        seg_num← RouteToImageIndexing(crossing, next_route)
        Buffering(T1,T2,T3,T4, seg_nums)
        StepWiseDirectionChange(ϕ ± 90°)  //± according to turn direction
        Loop(f) to zoom in
    //Switching to new route rendering from sections to view frame
        If(T1,T2, T3 are buffered) Retern to Mainloop // display new street.
    }
}
```

Generally, we can change the setting to increase the number of image segments to buffer if the machine or terminal has sufficient memory. This reduces the waiting time for buffering and realizes a smooth traversing. On the other hand, the initial loading of all the sections becomes slow and it affects the speed of turning to a new route.

5.2 Continuous Route Guidance

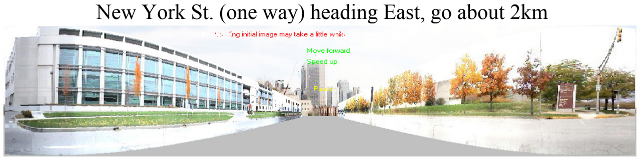
With the complete visual archive of routes, we can generate view sequences to a destination by concatenating the image segments of route panoramas, panoramic views at important locations and discrete images on landmarks. Referring to the map services on the Internet, a visitor can further check the scenes visible from the ground along the searched route. These scenes can be saved as guidance to a destination. For the entire area in Figure 9, we can arbitrarily pursue a route connecting street segments. Figure 13 shows an example to a destination, where real scenes are rendered dynamically in the view frame, using the saved script and the images saved in the server.

Using a PDA with Wi-Fi wireless or possibly GPS functions, real trip guidance can be carried out. As a visitor moves around the campus, he/she can update scenes while walking through the area. Because of the limitations of small portable devices in memory, screen size and rendering power, the campus system shown in Figure 9 has to be modified. Map, view and text are switched interchangeably in the same window. Some dynamic display functions using JAVA have to be replaced with pre-coded webpage. The interface design is shown in Figure 14.

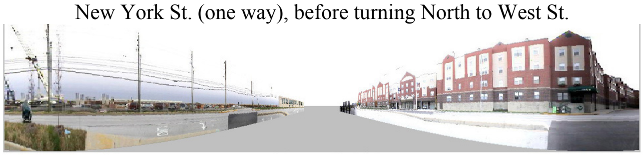
- Spot: the *view frame* displays a *compass view,* warped from a panoramic view in the polar coordinate system, to help the viewer to find orientations.
- Route: two side route panoramas of streets are aligned oppositely for display in landscape style. The route can be scrolled back and forth by viewers.
- Area or facility: a separate page is prepared with detailed description, and links to adjacent routes and spots.

To use the existing browser on the PDA, the campus data are reorganized as web pages with various types of hyperlinks (map, bookmark, page, etc.) and scrolling, in order to realize the pervasive view display and *map-to-view*, *text-to-view*, and *view-to-view* indexing. A route panorama is segmented and coded according to the street crossings and visible landmarks (building, sites, etc.). Links are embedded into the segments overlapped with the scene areas to realize the transitions to the related spaces and connected streets. These segments (320~500 pixel long) also become targets or bookmarks for the transitions back to this route from adjacent spaces; the display jumps to the right locations of the street.

New York St. (one way) heading East, go about 2km



New York St. (one way), before turning North to West St.



Turn left onto West St. heading North, go 500m

Turn left onto Michigan St. (one way) heading West, go 500m

Turn right to Blackford St from Michigan St. Garage is on the right.

Going North one block on Blackford St.

Turn left to North St. The garage entrance is on the left.

Walk 600m West along Michigan St. until destination is on left

Move (transit) to spot in a square with a panoramic view. Destination (a building) is marked in the view.

Figure 13 Screen shots of 180° scenes from continuous route display toward a destination

6. SYSTEMS AND EXPERIMENTS

In the route scanning, busy traffic is a problem affecting the stable collection of route panoramas. To extend the scene archiving to a real long street in a metropolitan city, smooth movements over even roads and less passing vehicles are required. Otherwise, post processing on shaking removing and length normalization of the route panoramas are heavy. Clouded traffics leave distorted shapes in the route panorama occluding background scenes.

The total storage for the campus system of 1.6×1 mile^2 area shown in Figure 9 is about 80MB, including 50 panoramic views and 34 route panoramas of both sides, and 44 buildings and sites (2D spaces). The height of image segments is 240 pixels and their lengths are around 1000 pixels. To facilitate the image editing, the lengths are not uniform. However, they must be longer than that of the view frame, in order to avoid noticeable pasting of image patches across the frame. The length is not set very long to reduce unnecessary data transmission. The view frame is set at 500~800 pixel wide on most PC screen. For two-side street scenes, at least ten image segments are kept by JAVA program.

Our JAVA based program renders 800×240 pixel screen of two side route panoramas at 5~10 frames per second on a 2.4GH PC without using special graphics card. Most of the time is spent on rendering view translation that deforms the scenes as the street stretches ahead. The viewer's rotation is rendered much faster than translation. A compiled C program using OpenGL realizes the same function at a frame rate higher than the video rate. The virtual translation speed is determined mainly from the image transmission speed. Buffering two-side views can reach a speed of 32pixels/second, which briefly corresponds to a bicycle or car speed. Loading from our campus internet or a local hard disc allows a viewer to travel at a fast car or train speed. If the Internet transmission is slow, the viewer may experience a virtual traffic jam in the route traversing, due to the waiting for buffering. The transmitted image segments are cached in temporal memory and this saves the transmission time if the route is traversed again. For street transition, a new set of image segments is transmitted or loaded. If the new image sizes are standard and consistent with the current ones, the memory is reused and the switching time is short (typically 1~2second). Otherwise, a new set of memory is allocated dynamically by JAVA program and underlying operation may take a longer time depending on system resources.

Figure 14 Map, spot, and route displayed on PDA

For the mobile platform, the sizes of route panorama and panoramic views are further reduced to half in order to display them in typical PDA screens of 320×240 pixels. We have tested an HP iPAD (520Hz, 128MB) to access the data in a website through campus wireless connection [24]. The total size of the campus is about 40MB, which is possible to be stored in a PDA memory card in case the wireless connection is unstable. The wireless transmission is sufficient to cope with walking speed in the area. The temporal loss of views of far locations on a street has less influence on the current location visualization. Because of switching between map and view, and transition between views, users have to pay more attention to route tracking. Panoramic views at street crossings may improve the smooth street transitions. Therefore, taking panoramic views right at path crossings should be achieved with some special camera or lens.

The limitations of the system are (i) shape deformation from projection and vehicle shaking in the route panoramas, (ii) slow rendering speed by JAVA program, and (iii) limited wireless coverage in navigating large areas.

7. CONCLUSION

This work creates a new model to record and visualize scenes pervasively in an urban space. It uses extended image media such as route panoramas, scene tunnels, panoramic view and spherical view, which significantly improved the visual information representation and distribution in digital cities. This model enables an entire area to be "visible" on the web through wired or wireless networks. The compactness of the extended images allows continuous city traversing. View-to-space indexing has facilitated visual and spatial information access. The obtained visual data can be further loaded to car navigation systems or handheld terminals as portable navigation tools. The technique will also allow viewers to walk and drive through an urban setting in a VR environment and extract quantitative information such as building heights, walking distance, traffic flow, etc. The future work will (a) evaluate usability of the systems for urban activities such as navigation, planning, shopping and advertising, (b) look at dynamic and temporal scanning of scenes for effects of traffic, natural and manmade disasters in urban environments, (c) employ real time transmission with adapting packet loss to cope with various moving speed in the area.

ACKNOWLEDGMENTS

The project is supported by the Department of Computer Science, IUPUI, and SBC Ameritech Fellow Program. The authors would express their thanks to Michelle Boshears, Yasaswy Bhupalam, Yu Zhou, Min Shi, Shebanov Eli, Jarrod Lannan and many other undergraduate students for their involvements in data collection and processing.

REFERENCES

[1] Zheng J. Y., Tsuji S. Panoramic Representation for route recognition by a mobile robot. *IJCV*, Vol.9, no.1, pp.55-76, 1992.

[2] Lippman A., Movie-map: an application of the optical video disc to computer graphics *ACM SIGGRAPH80*.

[3] Frueh C., Zakhor A., Constructing 3D city models by merging ground-based and airborne views, *IEEE CVPR03*, 562-569, 2003.

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

[5] Zheng J. Y., Zhou Y., Shi M., Scene Tunnels for Seamless Virtual Tour, *ACM Conf. Multimedia* 2004, 448-451.

[6] Zhu Z. G., Riseman E., Hanson A., Parallel-perspective Stereo Mosaics, Generalized parallel-perspective stereo mosaics from airborne video. *IEEE PAMI*, 26(2), 226-237, 2004.

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

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

[9] Zomet A., Feldman D., Peleg S., and Weinshall D., Mosaicing New Views: The crossed-slits projection, *IEEE Trans. on PAMI*, June 2003, pp. 741-754.

[10] Aliaga D. G., Carlbom I., Plenoptic Stitching: A scalable method for reconstructing 3D interactive walkthroughs, *SIGGRAPH01*, 2001.

[11] McMillan L., Bishop G., Plenoptic modeling: an image based rendering system, *ACM SIGGRAPH95*.

[12] Li S., Hayashi A., "Robot navigation in outdoor environments by using GPS information and panoramic views", *IEEE/RSJ Int. Conf. IROS,* pp. 570-575, 1998.

[13] Coorg S., Master N., Teller S., Acquisition of a large pose-mosaic dataset, *IEEE CVPR98*, 23-25, 1998.

[14] Kawanishi T., Yamazawa K., Iwasa H., Takemura H., Yokoya N., Generation of high-resolution stereo panoramic images by omnidirectional imaging sensor using hexagonal pyramidal mirrors, *14th ICPR*, Vol. 1, pp. 485-489, 1998.

[15] Ishida T., Digital City Kyoto: Social Information Infrastructure for Everyday Life. *ACM Communications* (CACM), 2002.

[16] Matsumura K., Usui K., Kai K., Ishikawa K., "Location-aware data broadcasting- an application for digital mobile broadcasting in Japan", *ACM Multimedia Conf. 2003*, pp. 271-274, 2003.

[17] Yoyama K., Logan R., Roseway A., Anadan P., Geographic location tags on Digital Images, *ACM Multimedia Conf.* 2003, pp.156-166.

[18] Zhao H., Shibasaki R., A Vehicle-borne urban 3D acquisition system using single-row laser range scanners, *IEEE Trans. on SMC*, B: vol. 33, no. 4, 658-666, 2003.

[19] Uyttendaele M. *et al.* Image-based interactive exploration of real-world environments. *IEEE Computer Graphics and Application,* 24(3), 2004

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

[21] Rottensteiner F., Briese C., Automatic generation of building models from LIDAR data and the integration of aerial images. *ISPRS, Vol. XXXIV*, 2003.

[22] Shi M., Zheng J. Y., A slit scanning depth of route panorama from stationary blur, *IEEE CVPR*, 2005.

[23] Zheng J. Y., Tsuji S. Generating Dynamic Projection Images for Scene Representation and Understanding, *CVIU*, Academy Press, Vol. 72, No. 3, pp. 237-256, 1998.

[24] http://www.cs.iupui.edu/~jzheng/digitalcity.html