# Digital Route Panoramas

**Jiang Yu Zheng**
*Indiana University–Purdue University, Indianapolis*

**Route panorama is a new image medium for digitally archiving and visualizing scenes along a route. It's suitable for registration, transmission, and visualization of route scenes. This article explores route panorama projection, resolution, and shape distortion. It also discusses how to improve quality, achieve real-time transmission, and display long route panoramas on the Internet.**

In ancient western art, compositions showed the perspective of a scene's projection toward a 2D view plane. In contrast, an ancient oriental painting technique exposed scenes of interest at different locations in a large field of view, which let viewers explore different segments by moving their focus. A typical and famous example is a long painting scroll named *A Cathay City* (see Figure 1), painted 900 years ago. It recorded the prosperity of the capital city of ancient China in the Song Dynasty by drawing scenes and events along a route from a suburb to the inner city.

The invention of paper in ancient China allowed most paintings to be created on foldable and scrollable papers instead of canvases. As opposed to a single-focus perspective projection at one end of a street or a bird's eye view, one benefit of this scrollable style is that a path-oriented projection can display more detailed visual information in an extended visual field than a single-focus perspective projection at one end of a street or a bird's-eye view from the top.

Today we can realize a composition approach similar to *A Cathay City* by creating a new image medium—Route Panorama, a program that contains an entire scene sequence along a route. We generate long route panoramas by using digital image-processing techniques and render route views continuously on the Internet. Now we can capture, register, and display route panoramas for many miles taken from vehicles, trains, or ships. We can use this approach for many practical purposes, including profiles of cities for visitors or for introductory indexes of local hometowns. In an extended manner, we could probably even use it as part of something like an enhanced version of Mapquest for people navigating through large cities like Los Angeles or Beijing. Figure 2 (next page) shows an example of a route panorama from a one-hour video in Venice, a place where people often get lost.

## Route scenes on the Internet

There are many things to consider when creating a quality route panorama. To begin with, we create a route panorama by scanning scenes continuously with a *virtual slit* in the camera frame to form *image memories*. We call it a virtual slit because it isn't an actual lens cover with a slit in it; rather, for each camera image in the video sequence, we're copying a vertical pixel line at a fixed position.

We paste these consecutive slit views (or image memories) together to form a long, seamless 2D *image belt*. We can then transmit the 2D

*Figure 1.* A Cathay City *painted on an 11 m scroll in the Song Dynasty 900 years ago. (From the collection of the National Palace Museum, Taipei, Taiwan, China.)*


*Figure 2. Segment of route panorama recording canal scenes in Venice.*

image belt via the Internet, enabling end users to easily scroll back and forth along a route. The process of capturing a route panorama is as simple as recording a video on a moving vehicle. We can create the route panorama in real time with a portable PC inside the vehicle or by replaying a recorded video taken during the vehicle's movement and inputting it into a computer later. Nevertheless, the generated image belt with its consecutive slit views pieced together has much less data than a continuous video sequence.

My colleagues and I first invented a route panorama 10 years ago for mobile robot navigation.[1-3] We called it a *generalized panoramic view* because we discovered it while creating the first digital panoramic view. In this early study, we mounted a camera on a moving vehicle and it constantly captured slit views orthogonal to the moving direction. The route panorama is a special case of a more general representation called a *dynamic projection image*,[4] which forms the 2D image using a nonfixed, temporal projection. Figure 3 (next page) illustrates how a vertical slit can scan the street scenes displayed in Figure 2 when the camera is positioned sideways along a smooth path. This viewing scheme is an orthogonal-perspective projection of scenes—orthogonal toward the camera path and perspective along the vertical slit. Generally speaking, common modes of transportation such as a four-wheeled vehicle, ship, train, or airplane can provide a smooth path for the camera.

Compared with existing approaches to model a route using graphics models,[5,6] our route panorama has an advantage in capturing scenes. It doesn't require taking discrete images by manual operation or texture mapping onto geometry models. A route panorama can be ready after driving a vehicle in town for a while. It yields a continuous image scroll that other image stitching or mosaic approaches,[7-9] in principle, are impossible to realize. A mosaicing approach works well for stitching images taken by a camera rotating at a static position. For a translating camera, however, scenes with different depths have different disparities (or optical flow vectors) in consecutive images. Overlapping scenes at one depth will add layers and create dissonant scenes at other depths, much like overlapping stereo images.

A route panorama requires only a small fraction of data compared to a video sequence and has a continuous format when accessed. If we pile a sequence of video images together along the time axis, we obtain a 3D data volume called *spatial–temporal volume* that's full of pixels (see Figure 4, p. 60). The route panorama comprises pixel lines in consecutive image frames, which correspond to a 2D data sheet in the spatial–temporal volume. Ideally, if the image frame has a width $w$ (in pixels), a route panorama only has $1/w$ of the data size of the entire video sequence ($w$ is 200~300), since we only extract one pixel line from each frame when viewing through the slit. The route panorama neglects redundant scenes in the consecutive video frames, which we can observe from the traces of patterns in the epipolar plane images (Figure 4 shows one EPI). This shows a promising property of the route panorama as an Internet medium, which can deliver large amounts of information with small data. The missing scenes are objects under occlusion when exposed to the slit. Also, as normal 2D images capturing dynamic objects, a route panorama only freezes instantaneous slit views, rather than capturing object movements as video does.

Figure 5 (see p. 60) is another example of a route panorama. It displays a segment of the First Main Street of China. We used a small digital video camera on a double-decker bus to record the 5-km route panorama. No image device other than an image grabber is required in processing.

## Projecting scenes

Compared with existing media such as photo, video, and panoramic views, route panorama has its own projection properties. For instance, much can depend on the smoothness of the ride while the camera is recording. In an ideal situation roads are flat and the vehicle has solid suspension, a relatively long wheelbase, and travels at a slow speed. Under these conditions, the camera has less up-and-down translation and can traverse a smooth path along a street.

In this article, we assume the camera axis is set horizontally for simplicity. We define a path-oriented coordinate system *SYD* for the projection of scenes toward a smooth camera trace *S* on the horizontal plane. We denote a point in such a coordinate system as $P(S, Y, D)$, where $S$ is the camera-passed length from a local starting point (see Figure 3b), $Y$ is the height of the point from the plane containing the camera path, and $D$ is the depth of the point from the camera focus. Assume $t$ and $y$ are the horizontal and vertical axes of the route panorama, respectively. The projection of $P$ in the route panorama denoted by $p(t, y)$ is

$$t = S/r \qquad y = Yf/D \tag{1}$$

where $r$ (m/frame) is the slit sampling interval on the camera trace, which is the division of the vehicle's speed $V$ (m/second) by the camera's frame rate per second (fps). The camera's focal length $f$ (in pixels) is known in advance after calibration.

Because the route panorama employs an orthogonal-perspective projection, the aspect ratio of an object depends on its distance from the path. Figure 6 displays a comparison of views between an ordinary perspective image and a route panorama. The further the object, the lower the object height is in the route panorama. Object widths in the route panorama are proportional to their real widths facing the street. In this

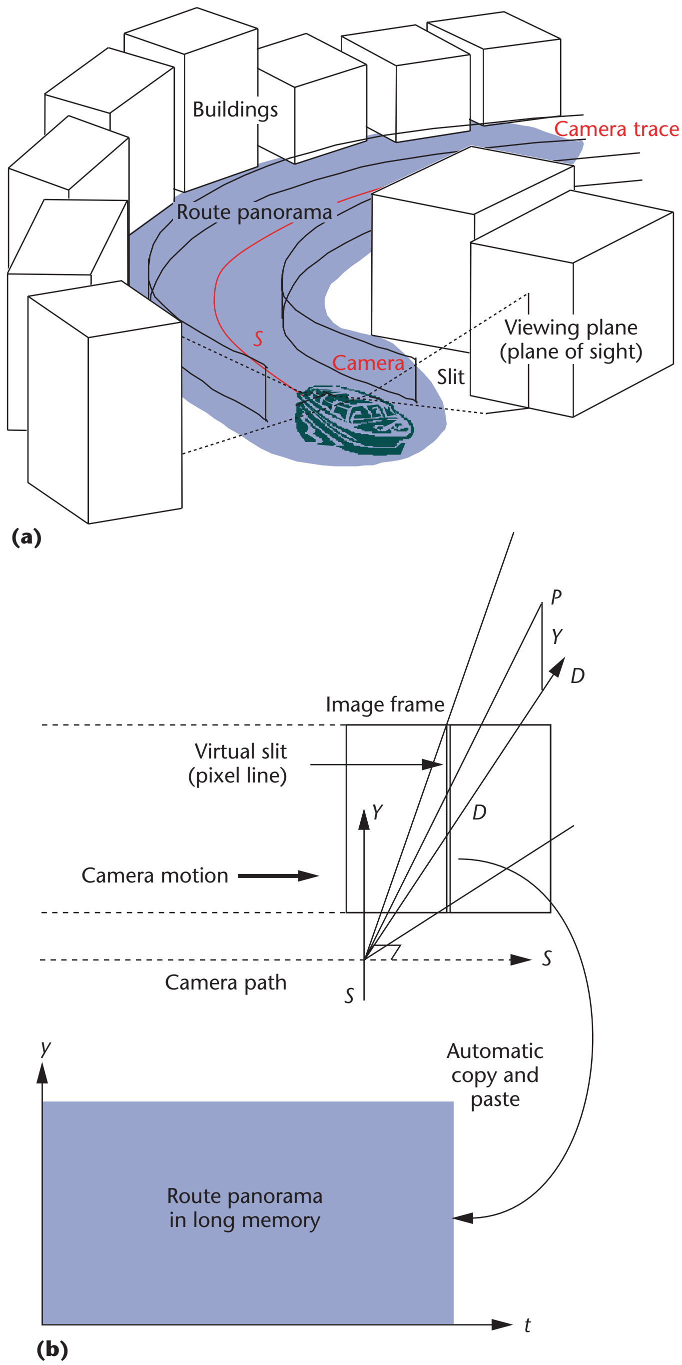


*Figure 3. Route scene scanning by constantly cutting a vertical pixel line in the image frame and pasting it into another continuous image memory when the camera moves along a smooth path on the horizontal plane.*

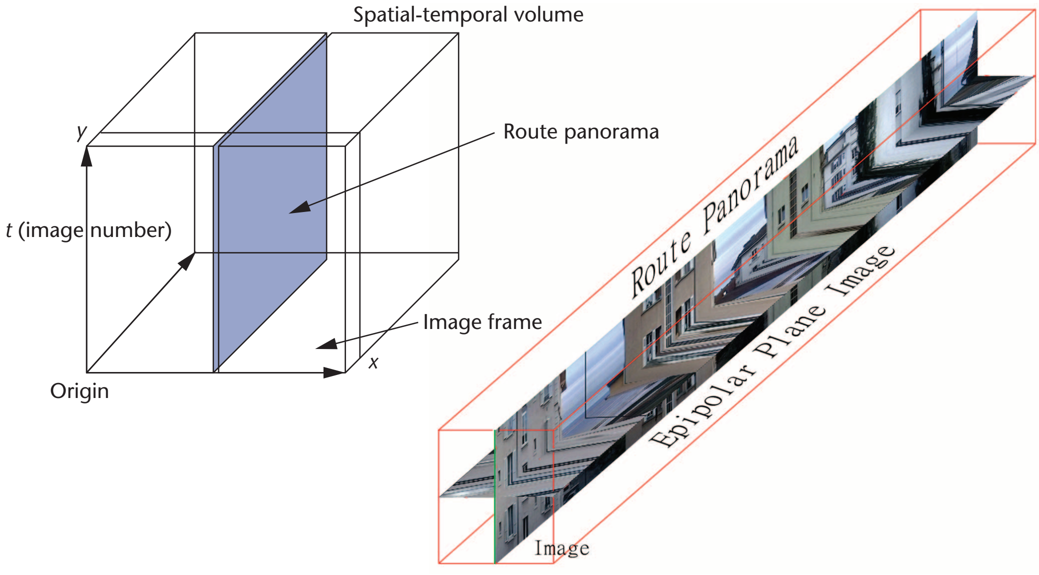


*Figure 4. Data size of a route panorama is one sheet out of a volume of video images.*

*Figure 5. A segment of route panorama generated from a video taken on a bus (before removing shaking components).*

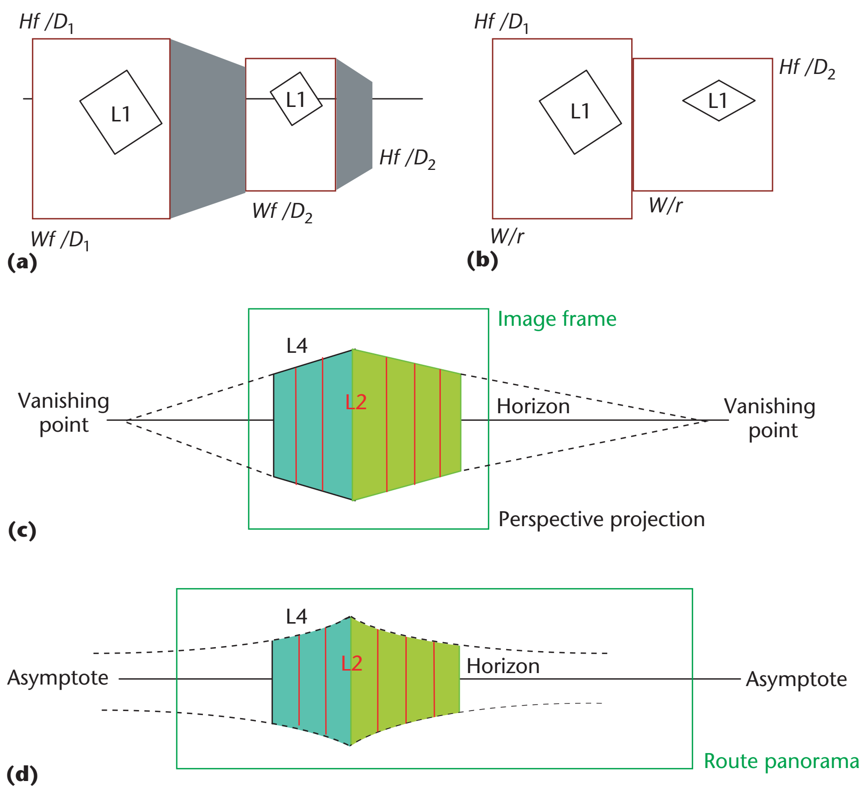


*Figure 6. 2D projections of 3D objects in a route panorama compared with a perspective projection image.* W *stands for the object width,* H *is object height, and* D *is object depth. (a) Ordinary perspective projection. (b) Orthogonal-perspective projection. (c) A typical object in perspective projection and (d) in an orthogonal-perspective projection images.*

sense, distant objects are extended horizontally in the route panorama. Thus, route panoramas aren't likely to miss large architectures. Small objects such as trees, poles, and signboards are usually narrower than buildings along the $t$ axis and might disappear after squeezing in the $t$ direction, while buildings won't disappear. In a normal perspective projection image, a small tree close to the camera may occasionally occlude a large building behind it.

We examine several sets of structure lines typically appearing on 3D architectures (see Figures 6c and 6d) and find their shapes in the route panorama from a linear path. Assuming a linear vector $\mathbf{V} = (a, b, c)$ in the global coordinate system with its $X$ axis parallel to the camera path, the line sets are

- L1 $\{\mathbf{V} \mid c = 0\}$: lines on vertical planes parallel to the camera path. These lines may appear on the front walls of buildings.
- L2 $\{\mathbf{V} \mid a = c = 0\} \subset$ L1: vertical lines in the 3D space. These lines are vertical rims on architectures.
- L3 $\{\mathbf{V} \mid c \neq 0\}$: lines stretching in depth from the camera path.
- L4 $\{\mathbf{V} \mid c \neq 0, b = 0\} \subset$ L3: horizontal 3D lines nonparallel to the camera path.

Obviously, lines in L2 are projected as vertical lines in the route panorama through the vertical slit. Denoting two points $P_1(S_1,Y_1,D_1)$ and $P_2(S_2,Y_2,D_2)$ on the line, where $P_1P_2 = \tau V$, their projections in the route panorama are $p_1$ and $p_2$. The projection of the line is then

$$\mathbf{v} = p_2 - p_1 = \left( \frac{S_2 - S_1}{r} \quad \frac{fY_2}{D_2} - \frac{fY_1}{D_1} \right)$$

according to the projection model in Equation 1. For a line in L1 where $\Delta D = D_2 - D_1 = 0$ and $\Delta Y/\Delta S$ is constant $b/a$, its projection becomes

$$\mathbf{v} = \left( \frac{\Delta S}{r} \quad \frac{f\Delta Y}{D_1} \right) = \left( \frac{a\tau}{r} \quad \frac{fb\tau}{D_1} \right) = \tau \left( \frac{a}{r} \quad \frac{fb}{D_1} \right)$$

which is linear in the route panorama. Therefore, a line in L1 is still projected as a line in the route panorama for a linear path.

**We can obtain reasonably good visual indexes of route scenes as long as we allow for smooth curves in the route panorama.**

The most significant difference from perspective projection is a curving effect on line set L3 (Figure 6d). For a line in L3, its projection in the route panorama is a curve, because point $P_2$ in the route panorama is

$$(t_2, y_2) = \left( \frac{S_2}{r} \quad \frac{fY_2}{D_2} \right) = \left( \frac{\Delta S + S_1}{r} \quad \frac{f(Y_1 + \Delta Y)}{D_1 + \Delta D} \right) = \left( \frac{a\tau + S_1}{r} \quad \frac{f(Y_1 + b\tau)}{D_1 + c\tau} \right)$$

which is a hyperbolic function of $\tau$. This curve approaches a horizontal asymptotic line $y = fb/c$ from $p_1$ when $\tau \to \infty$. Particularly for lines in L4, their projections are curves approaching toward the projection of horizon ($y = 0$) in the route panorama.

The path curvature also defines the lines' curving effect if the camera moves on a curved trace. Because of space, we omit the analysis here. Nevertheless, we can obtain reasonably good visual indexes of route scenes as long as we allow for smooth curves in the route panorama.

Another interesting property of the route panorama is the common asymptote for a set of parallel lines stretching in depth (parallel lines in L3). Under perspective projection, we project parallel lines with a depth change in the 3D space onto the image plane as nonparallel lines and their extensions on the image plane cross at a common point called the *vanishing point* (according to the principle in computer vision). In the route panorama obtained from a linear camera path, however, a set of 3D parallel lines stretching in depth has a common asymptotic line in the route panorama. This is because parallel lines in L3 have the same direction $(a, b, c)$, and their projections in the route panorama all approach the same horizontal asymptotic line $y = fb/c$ when $\tau \to \infty$.

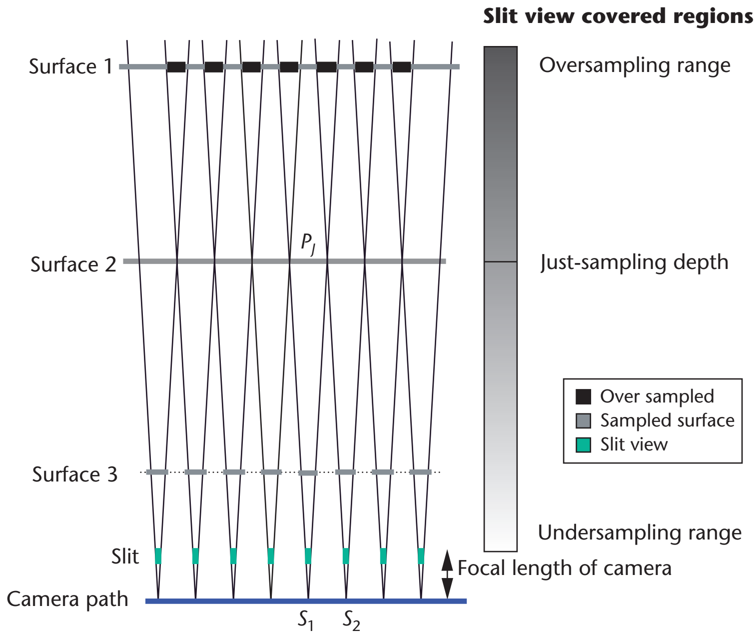


*Figure 7. Oversampling range, just-sampling depth, and undersampling range of the route panorama.*

If we fix the camera axis so that the plane of sight through the slit isn't perpendicular to the camera path, we obtain a parallel-perspective projection along a linear path because all the perspective planes of sight are parallel. We can further extend this to a bended-parallel-perspective projection when the camera moves along a curved path. We can extend most properties of the orthogonal-perspective projection similarly.

## Stationary image blur and close object filtering

When we're actually recording a route panorama, we obtain slit views by cutting a pixel line in the image frame of a video camera. Every slit view itself then is a narrow-perspective projection. The sampling rate of the slit has a limit lower than the video rate. If the vehicle speed isn't slow enough, it's reflected in the route panorama, because the panorama is actually the connection of narrow perspective projections at discrete positions along the route (as Figure 7 depicts). Scenes contained in a narrow wedge are projected onto the one-pixel line at each time instance. We examine surfaces that can appear in three depth ranges from the camera path. First, at the depth where surface 2 in Figure 7 is located, each part of the surface is taken into consecutive slit views without overlapping, just as a normal perspective projection does. We call this the *just-sampling range* (*depth*).

Second, for a surface closer than the just-sampling range, the consecutive slit views don't cover every fine part of the surface (surface 3 in Figure 7). Surfaces in this range are undersampled in the route panorama. If the spatial frequency of intensity distribution is low on the surfaces—that is, the surface has relatively homogeneous intensity—we can recover the original intensity distribution from the sampled slit views (the route panorama), according to the Nyquist theorem in digital signal processing. Otherwise, we may lose some details within that range. Therefore, route panorama has a function of filtering out close objects such as trees, poles, people, and so forth. By reducing the camera's sampling rate, this filtering effect becomes clearer and more distinct. This is helpful when we're mainly interested in architectures along a street.

Third, if a surface is farther than the just-sampling range, the camera could oversample the surface points. Because a slit view accumulates intensities in the narrow perspective wedge, a point on surface 1 may be counted in the overlapped wedges of the consecutive slit views. Therefore, a distant object point retaining at a position in the perspective projection may cause a blur horizontally in the route panorama. We call this phenomenon *stationary blur*, since it's converse to the motion blur effect in a dynamic image where a fast translating point wipes across several pixels during the image exposure.

We can give the just-sampling depth's numerical computation in a more general form to include bended parallel-perspective projection. If we set a global coordinate system *O-XYZ*, we can describe the smooth camera path by $S[X(t), Z(t)]$. If the vehicle is moving on a straight lane without obvious turns, the camera path almost has zero curvature ($\kappa \cong 0$). Selecting a camera observing side, we can divide a path roughly as linear, concave, or convex segments, depending on the sign of curvature. For simplicity, we assume the radius of curvature $R(t)$ of a curved camera path is constant between two consecutive sampling positions $S_1$ and $S_2$, where $R(t) > 0$ for a concave path, $R(t) < 0$ for a convex path, and $R(t) = \infty$ for a linear path. The curve length between $S_1$ and $S_2$ is $r$. The wedge's angle is $2\theta$ where $f \times \tan\theta = 1/2$, because we cut one pixel as the slit width (see Figure 8).

For bended parallel-perspective projection, the plane of sight—which is the wedge's central plane—has an angle $\alpha$ from the camera transla-

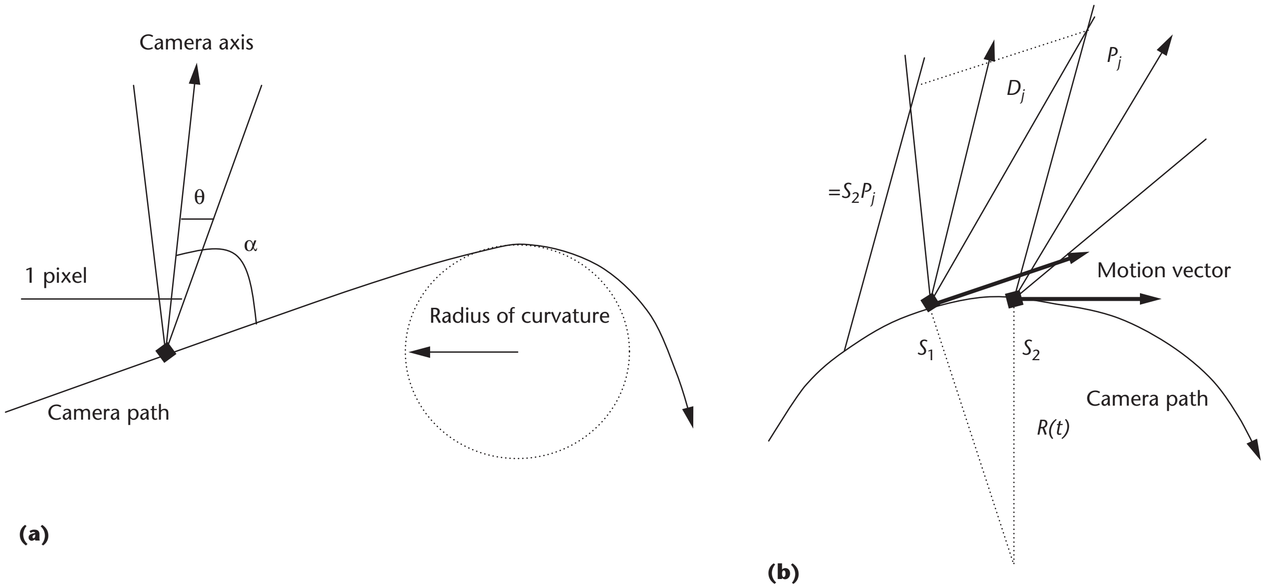


***Figure 8. Just-sampling range for a curved camera path. (a) Curved camera path and a physical sampling wedge. (b) Two consecutive slit views and just-sampling depth.***

tion direction that is the curve's tangent vector. The two consecutive wedges of thin perspective projection meet at a vertical line through point $P_j$ in Figure 8. We have a vector relation on the horizontal plane as

$$\mathbf{S_1P}_j = \mathbf{S_1S}_1 + \mathbf{S_1P}_j$$

in triangle $S_1S_2P_j$. Then we obtain $S_1P_j$ and $S_2P_j$ as

$$S_1P_j = \frac{2R(t)\sin\frac{r}{2R(t)}\sin\left(\alpha+\theta+\frac{r}{2R(t)}\right)}{\sin\left(2\theta+\frac{r}{R(t)}\right)}$$

$$S_2P_j = \frac{2R(t)\sin\frac{r}{2R(t)}\sin\left(\alpha-\theta-\frac{r}{2R(t)}\right)}{\sin\left(2\theta+\frac{r}{R(t)}\right)}$$

by using sine theorem. It isn't difficult to further calculate the just-sampling depth $D_j$ in the plane of sight to the just-sampled surface. Figure 9 shows the equation for this.

It's important to note that the just-sampling range not only relies on the camera's sampling rate, vehicle speed, image resolution, and camera's focal length, but also depends on the camera path's curvature. The just-sampling range tends to be close when the camera moves on a concave path and far on a linear path. When the camera path changes from linear to convex ($R(t)$ varies from $-\infty$ to 0–), $D_j$ extends to the infinity ($D_j \to +\infty$) and then starts to yield negative values ($P_j$ flips to the path's other side). This means that the consecutive perspective wedges won't intersect when the convex path reaches a high curvature and the entire depth range toward infinity is undersampled.

$$D_j = \frac{S_1P_j \times S_2P_j \cos\theta}{S_1P_j + S_2P_j}$$
$$= \frac{2R(t)\sin\frac{r}{2R(t)}\sin\left(\alpha+\theta+\frac{r}{2R(t)}\right)\sin\left(\alpha-\theta-\frac{r}{2R(t)}\right)\cos\theta}{\sin\left(2\theta+\frac{r}{R(t)}\right)\left(\sin\left(\alpha+\theta+\frac{r}{2R(t)}\right)+\sin\left(\alpha-\theta-\frac{r}{2R(t)}\right)\right)}$$

***Figure 9. Calculating the just-sampling depth $D_j$ in the plane of sight to the just-sampled surface.***

For a simple orthogonal-perspective projection toward a linear path, we can simplify the equation in Figure 9 to

$$D_j = \frac{r}{2\tan\theta}$$

by setting $\alpha = \pi/2$ and $R(t) \to \infty$. Overall, there are differently sampled regions in a route panorama depending on the subjects' depths. We can select a sampling rate so that the just-sampling range is

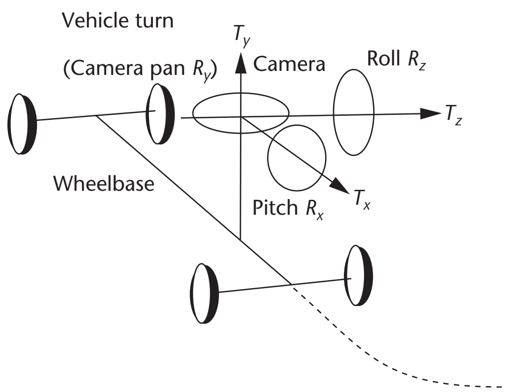


***Figure 10. Vehicle and camera model in taking a route panorama. Note that ($T_x$, $T_y$, $T_z$, $R_x$, $R_y$, and $R_z$) = (forward translation, up-and-down translation, translation sideways, pitch, pan, and roll), respectively. Also, translation sideways doesn't occur for a vehicle movement.***

approximately at the front surfaces of the buildings of interest.

The collection of slit views is a process of smoothing spatial intensity distribution: output value at a point by averaging intensities over a space around it. We can estimate the degree of stationary blur as follows. At depth $D$, the width of the perspective wedge is $W = D \tan\theta$. We can average the color distribution at depth $D$ over $W$ to produce a pixel value for the slit view, which is the convolution between the intensity distribution and a rectangular pulse function with width $W$ and height $1/W$. If we set a standard test pattern at depth $D$ that's a step edge with unit contrast or sharpness, we can easily verify that the convoluted result is a triangular wave with the contrast reduced to $1/(D \tan\theta)$. Therefore, an edge's sharpness is inversely proportional to its depth. This is important when estimating objects' depth in a route panorama.

If a segment of route panorama mainly contains objects far away, we can squeeze it along the $t$ axis to reduce the stationary blurring effect and, at the same time, reduce shape distortion. This scaling may visually improve the objects' appearances in the route panorama if there's no requirement to keep the exact scale or resolution of the route panorama horizontally.

## Dealing with camera shaking

Improving image quality is a crucial step toward the real application of route panoramas in multimedia and the Internet. In Figure 5, we pushed a small video camera firmly on a window frame of the bus to avoid uncontrollable accidental camera movement. We still can observe severe zigzags on horizontal structural lines in the route panorama. This is because the camera shook when the vehicle moved over an uneven road. To cope with the camera shaking, some have tried to compensate by using a gyroscope. However, adding special devices might increase the difficulty of spreading this technology. Our approach was to develop an algorithm to rectify distortion according to some constraints from scenes and motion.

We can describe the camera movement at any instance by three degrees of translations and three degrees of rotations (as displayed in Figure 10). Among them, the camera pitch caused by left-and-right sway and the up-and-down translation caused by the vehicle shaking on uneven roads are the most significant components affecting the image quality. The latter only yields small up-and-down optical flow in the image if the vehicle bumping is less than several inches.

Overall, camera pitch influences image quality the most and luckily we can compensate for it with an algorithm that reduces camera shaking (many algorithms along these lines exist).[10,11] Most of these algorithms work by detecting a dominant motion component between consecutive frames in the image sequence. For a route panorama, however, we only need to deal with shaking components between consecutive slit lines. The idea is to filter the horizontal zigzagged lines in the route panorama to make them smooth, which involves spatial processing. We do this by using the following criteria:

- smooth structural lines in the 3D space should also be smooth in a route panorama, and
- vertical camera shaking (in pitch) joggles the slit view instantly to the opposite direction.

As Figure 11 illustrates, we estimate the camera motion from the joggled curves in the route panorama and then align vertical pixel lines accordingly to recover the original straight structural lines. The way to find an instant camera shaking is to check if all the straight lines joggle simultaneously at that position. We track line segments horizontally in the route panorama after edge detection and calculate their consecutive vertical deviations along the $t$ axis. At each $t$ position, we use a median filter to obtain the common vertical deviation of all lines to yield the camera's shaking component. The median filter prevents taking an original curve in the scene as a camera-joggled line and resulting in a wrong pitch value. After obtaining the sequence of camera parameters along the $t$ axis, we prepare a window shifting along the horizontal axis and apply another median filter to the camera sequence in the window, which eliminates disturbances from abrupt vehicle shaking.

Suppose the original structure lines in the scenes are horizontal in an ideal route panorama (Figure 11a). The camera, however, shakes vertically over time (Figure 11b), which joggles the structure lines inversely in the captured route panorama (Figure 11c). By shifting all the vertical pixel lines according to the estimated camera motion sequence, we can align curved lines properly to make horizontal structure lines smooth in the route panorama (Figure 11d).

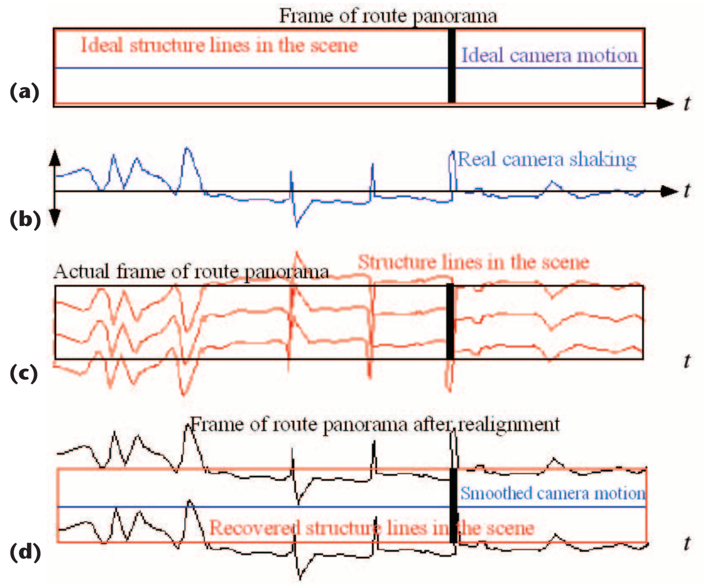


*Figure 11. Recovering straight structure lines from camera-joggled curves in a route panorama.*

*Figure 12. Recovering smooth structure lines in the route panorama by removing the camera-shaking components.*

Figure 12 shows the route panorama of Figure 5 after removing the camera shakes in the pitch. This algorithm is good at removing small zigzags on the structure lines to produce smooth curves. Once we apply the algorithm, it's easy to modify the route panorama to make major lines smooth and straight.


*Figure 13. Real-time streaming data transmission over the Internet to show route scenes.*

## Real-time transmission and display

Our next step is to transmit a long route panorama on the Internet and to seamlessly scroll it back and forth. Displaying and streaming route panoramas gives users the freedom to easily maneuver along the route.

We developed three kinds of route panorama displays to augment a virtual tour. The first type is a long, opened form of route panorama (see Figure 2). The second type is a side-window view (see Figure 13) that continuously reveals a section of the route panorama back and forth. We call it a *route image scroll*. You can control the direction of movement and the scrolling speed with a mouse. The third type is a forward view of the vehicle for street traversing; we call it a *traversing window* (see Figure 14, next page).

We can combine these different displays in various ways. In rendering a traversing window, we map both side-route panoramas onto two sidewalls


*Figure 14. The traversing window dynamically displays two sides of route panoramas in an open, cylindrical panoramic view for virtual navigation along a street.*

along the street and then project to a local panoramic view (a cylindrical image frame). We then display the opened 2D form of this panoramic view so that users can observe the street stretching forward as well as architectures passing by, while traversing the route. We render the traversing window continuously according to the moving speed specified by the mouse. Although the traversing window isn't a true 3D display, major portions in the traversing window have an optical flow that resembles real 3D scenes. As another form of use, it's even possible to display these pseudo-3D routes within a car's navigation system.

As a route panorama extends to several miles, it's unwise to download the whole image and then display it. We developed a streaming data transmission function in Java that can display route image scrolls and traversing windows during download. Because of the route panoramas' small data sizes, we achieved much faster transmission of street views than video.

The image belt provides a visual profile of a long street for its compactness. By adding clicking regions in the image, the route panorama becomes a flexible visual index of routes for Web page linking. On the other hand, we can automatically scroll a route panorama in a small window to give viewers the feeling that they're viewing architectures and shops from a sightseeing bus. With two cameras directing left and right sides of the vehicle, we can establish two side views of a route by synchronizing the route panoramas. If we drive vehicles passing every street in a town for all the route panoramas, we can generate a visual map of the town for virtual tourism on the Internet.

With the tools discussed here, we can register and visualize an urban area using panoramic views, route panoramas, route image scrolls, and maps. All these images have much less data compared to video and look more realistic than 3D computer-aided design models. We can link areas within a city map on the Web to corresponding areas in the route panoramas so that clicking a spot on the map can a update route image scroll accordingly (and vice versa). Eventually, we can use these tools to visualize large-scale spaces such as a facility, district, town, or even city.

## Conclusion

Streets have existed for thousands of years and there are millions of streets in the world now. These streets have a tremendous amount of information including rich visual contexts that are closely related to our lifestyles and reflect human civilization and history. Registering and visualizing streets in an effective way is important to our culture and commercial life. With the Route Panorama software, when you click a map to follow a certain route, the route panorama will also be scrolled accordingly, showing real scenes along the route. This will greatly enhance the visualization of Geographic Information Systems (GIS).

Because of the 2D characteristics of captured route panoramas, we can use them in a variety of ways. For instance, we can display and scroll them on wireless phone screens or handheld terminals for navigation in cities or facilities. By connecting a route panorama database with the Global Positioning System, we can locate our position in the city and display the corresponding segment of a route panorama on a liquid crystal display. Displaying route panoramas and panoramic images are basically raster copy of sections of images. Hence, the proposed techniques are even applicable for 2D animation and game applications, potentially providing richer, more realistic content than before.

Currently, we're working on several innovations for our Route Panorama software. This includes constructing 3D streets from route panoramas, sharpening distance scenes due to the stationary blur, establishing route panoramas with a flexible camera setting, capturing high rises into route panoramas, combining route panoramas with local panoramic views in cityscape visualization, and linking route panoramas to existing GIS databases. **MM**

## References

1. J.Y. Zheng, S. Tsuji, and M. Asada, "Color-Based Panoramic Representation of Outdoor Environment for a Mobile Robot," *Proc. 9th Int'l Conf. Pattern Recognition*, IEEE CS Press, vol. 2, 1988, pp.801-803.
2. J.Y. Zheng and S. Tsuji, "Panoramic Representation of Scenes for Route Understanding," *Proc. 10th Int'l Conf. Pattern Recognition*, IEEE CS Press, vol. 1, 1990, pp.161-167.
3. J.Y. Zheng and S. Tsuji, "Panoramic Representation for Route Recognition by a Mobile Robot," *Int'l J. Computer Vision*, Kluwer, vol. 9, no.1, 1992, pp. 55-76.
4. J.Y. Zheng and S. Tsuji, "Generating Dynamic Projection Images for Scene Representation and Recognition," *Computer Vision and Image Understanding*, vol. 72, no. 3, Dec.1998, pp. 237-256.
5. T. Ishida, "Digital City Kyoto: Social Information Infrastructure for Everyday Life," *Comm. ACM*, vol. 45, no. 7, July 2002, pp. 76-81.
6. G. Ennis and M. Lindsay, "VRML Possibilities: The Evolution of the Glasgow Model," *IEEE MultiMedia*, vol. 7, no. 2, Apr.–June 2000, pp. 48-51.
7. Z. Zhu, E.M. Riseman, and A.R. Hanson, "Parallel-Perspective Stereo Mosaics," *Proc. Eighth IEEE Int'l Conf. Computer Vision*, IEEE CS Press, vol. I, 2001, pp. 345-352.
8. H.S. Sawhney, S. Ayer, and M. Gorkani, "Model-Based 2D and 3D Motion Estimation for Mosaicing and Video Representation," *Proc. 5th Int'l Conf. Computer Vision*, IEEE CS Press, 1995, pp. 583-590.
9. S.E. Chen and L. Williams, "Quicktime VR: An Image-Based Approach to Virtual Environment Navigation," *Proc. Siggraph 95*, ACM Press, 1995, pp. 29-38.
10. Z. Zhu et al., "Camera Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering," *Proc. IEEE Int'l Conf. Intelligent Vehicles*, IEEE CS Press, vol. 2, 1998, pp. 329-334.
11. Y.S. Yao and R. Chellappa, "Selective Stabilization of Images Acquired by Unmanned Ground Vehicles," *IEEE Trans. Robotics and Automation*, vol. RA-13, 1997, pp. 693-708.


**Jiang Yu Zheng** is an associate professor at the Department of Computer and Information Science at Indiana University–Purdue University, Indianapolis. His current research interests include 3D modeling, dynamic image processing, scene representation, digital museums, and combining vision, graphics, and human interfaces. Zheng received a BS degree from Fudan University, China, and MS and PhD degrees from Osaka University, Japan. He received the 1991 Best Paper Award from the Information Processing Society of Japan for generating the first digital panoramic image.


Readers may contact Jiang Yu Zheng at jzheng@cs.iupui.edu.