Technical report
7 pages, no accompanying disk
Latest amendment: 7 Aug. 2000

Panoramic Images

Panoramic images create a photorealistic, three dimensional, navigable environment. A panoramic image captures the surroundings of a location in a 360° cylindrical or spherical view. The spectator is immersed in this environment and he or she usually has some control over viewport, through panning and zooming and tilting. The viewer transforms the portion of the image currently shown through the viewport so that it shows the correct perspective.

Panoramic imaging presents opportunities to a variety of entertainment and educational uses. Adventure style games with realistic (photographed or scanned) or semi-realistic, high quality backgrounds are one example. Guided tours and real estate images are another (museums, exhibitions, building sites).

When compared with full 3D engines, panoramic images represent a trade-off between image quality, speed of display and the ability to move freely through a scene. Panoramic imaging takes the path of pre-rendering the scenes as much as possible. This allows you to use very detailed, photo-realistic images for the panorama. 3D engines are usually restricted to polygons and textures of limited size and of limited number. The drawback of panoramic imaging is that you can only move along pre-computed paths.

Panoramic images can be combined with a sprite engine to put “motion” into the surrounding panorama.

Overview

There exist at least three types of panoramic images: cylindrical, cubical and spherical. The techniques underlying each kind of panoramic image are quite different. Hence, it is important not to confuse the varieties.

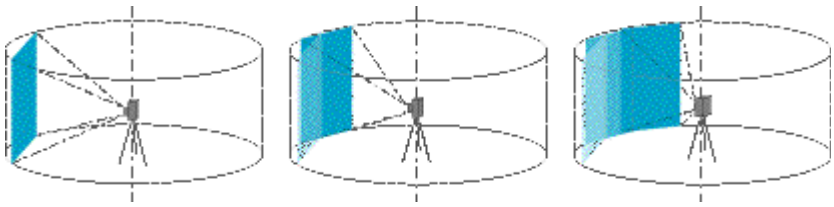
Spherical panoramic images are usually taken with a fisheye lens. The photographer captures two pictures in opposite directions using a camera with a fisheye lens. Each picture covers a hemi-sphere; the combination of both pictures covers the full sphere. There exist cameras, both still picture cameras or video cameras that use two fisheye lenses to capture a full sphere in one “click”. An example of software for spherical panoramic images is IPIX from Perfect World Technologies, which focuses at the high end UNIX market.

The quality of a spherical panorama is uneven, because the source images are much denser along the (circular) circumference than in the centre. That is, the pixels in the centre of the image represent a much smaller area of the photographed scene than the pixels near the circumference. The effect of a fisheye lens is that of a magnifying lens: there is much detail in the centre, but very little detail at a 90° angle.

Cubical panoramic environments consist of 6 images, one for every face of a cube. The images for cubical panoramas are usually generated by a computer, because the angle of a camera lens would need to be 90° to capture the required images (and still produce rectilinear images). Tools and engines for cubical panoramic images are produced, amongst others, by Warp Ltd. Their product, Virtual TV (VTV), is available for Microsoft’s Win32 environment. Wasabi software offers a plug-in for Photoshop and Paint Shop Pro which helps an artist to draw or retouch panoramic images.

Cylindrical panoramas offer only limited rotational freedom around the horizontal axis. That is, one can turn around in a full circle (rotation around the vertocal axis), but the amount by which one can look up and down is restricted. The images for a cylindrical panorama are best captured

with a special *panoramic* camera with a rotating lens and a vertical diaphragm. However, the most common method of image acquiring for panoramic images is to capture a series of pictures with a standard camera that turns around a single point of rotation. After digitizing the pictures into a set of computer image files, these images are *stitched* together using tools like Apple's QTVR tools, or Live Picture's PhotoVista. The source images can be captured with nearly any type of lens and camera combination; even a simple 35mm lens. Panodome, an advanced panoramic image engine for applications supports cylindrical panoramas.



The four main requirements for this stitching approach are:

- The camera must rotate on an axis perpendicular to the axis of the lens.
- The tripod should have a radial scale, or the camera should be corrected for parallax. Otherwise, getting even angle increments while taking pictures becomes very hard. Note that stitching software needs the pictures to overlap, and that the software usually specifies a *minimum* and a *maximum* overlap. Typically, the overlap must be 30% to 50%.
- The lens must produce rectilinear images—that is, straight lines in the input image must appear as straight lines in the resulting image. This rules out fisheye lenses.
- Postprocessing software must warp and stitch the images together in a way that mimics the operation of a panoramic camera.

Cylindrical panoramic images: technical details

Three main sources for panoramic images are:

- Still camera on a tripod: the set-up described above.
- A video camera on a tripod; with the right software (VideoBrush), the stitching is automatic and motion is handled transparently. The resolution is lower than that of a still camera.
- Panoramic camera (swing lens): expensive, but required when capturing panoramas at locations with unavoidable movement (e.g. streets with traffic). Alternatives to a swing lens are special parabolic reflectors with postprocessing software. These "panoramic lenses" are also quite expensive.
- Computer generated images. A 3D rendering system can usually mimic the operation of a panoramic camera by rendering the scene as viewed from a rotating narrow vertical shutter. The picture below is an example of a computer generated panoramic image.

Image restoration

Panoramic images only show the proper perspective when viewed as a cylinder. When flattened out, features which would normally be straight lines become warped. Depending on the radius of the (imaginary) cylinder, this curvature can be quite severe.

The geometry of a panoramic image is different from that of a normal (camera) picture in the aspect that all points along a horizontal (scan) line are equidistant from the camera's focal point. This, in effect, creates a cylindrical image that only appears correctly when viewed from the exact centre of the cylinder. When the image is "unrolled" on a flat surface, such as a computer monitor, the image shows heavy distortion.

To remove the distortion, the "viewer" software should mathematically project a portion of the cylindrical source onto a flat surface. The resulting image will then be similar to one that is taken by a normal camera.

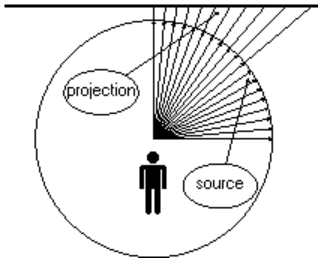


FIGURE 1. *The horizontal projection*

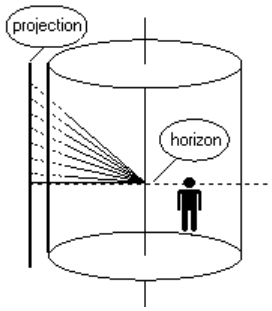


FIGURE 2. *The vertical projection*

A note about the formulae in the analysis below: the intermediate expressions serve only to clarify the subject. The final expression, for each topic, is boxed. Sometimes, such an expression has interesting (or useful) properties; these are added, right-aligned, in the box for the final formula.

The horizontal projection

The image is wrapped on a cylinder with an appropriate radius. That is, the width of the panoramic image determines the radius of the cylinder.

$$R = \frac{image\ width}{2\pi}$$

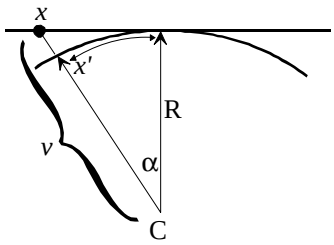


FIGURE 3. *Horizontal projection*

Compare figure 3 to figure 1. Point "C" is the centre of the cylinder. To simplify the calculation, the flat "projection screen" touches the cylinder (in figure 1, there was a slight gap between the cylinder and the screen).

$\alpha = \arctan\left(\frac{x}{R}\right)$ α is in radians (of course)

$$x' = R\alpha = R \cdot \arctan\left(\frac{x}{R}\right)$$ Note that $x' \le x$.

The vertical projection

Compare figure 4 to figures 2 and 3. Figure 4 is the "simplified" or abstracted version of figure 2. Some of the symbols in figure 4 are also in figure 3: x' is where a horizontal "ray" hits the cylinder and x is the position where that ray hits the flat screen; R is the radius of the cylinder and v is the length of the vector between C and x. From figure 3, one can easily determine v:

$$v = \sqrt{x^2 + R^2}$$

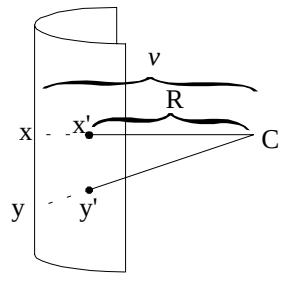


FIGURE 4. *The vertical projection*

The triangles ΔCxy and $\Delta Cx'y'$ are isomorphic. The distance C-x' is the radius R and the distance C–x is v. Therefore:

$$\frac{y}{v} = \frac{y'}{R} \Leftrightarrow y' = y \cdot \frac{R}{v}$$

$$y' = \frac{yR}{\sqrt{x^2 + R^2}}$$ Note that $y' \le y$; $y' = y$ iff $x = 0$.

Tilt

The given formulae are valid only when looking at the horizon. When moving the picture so that the horizon of the image is not at centre of the viewport, the formulae for the horizontal projection needs to be adjusted. This effect is referred to as “tilt”.

Tilt mimics looking up and down. When the spectator in figure 2 does looks at the lower part of the image, the new “horizon” is no longer a horizontal line. If one would really build the set-up in figure 2, with a large cylinder in which a person could view his or her surroundings, no correction is necessary. Human vision is able to correct the changes in perspective when looking at an image in different angles. Psychologists refer to this as the “robustness of perspective” (Michael Kubovy; *The psychology of perspective and renaissance art*; Cambridge U.P.; 1986).

On the computer screen, the spectator’s head does not really move when he or she “virtually” looks up or down the panoramic image. Instead the viewport —the computer monitor— stays at the same spot. Therefore the “robustness of perspective” principle does not apply, and one must correct for the distortion of the projection screen due to tilt.

The perspective correction is independent from the cylindrical correction (“warping”). That is, it does not make a difference whether you change the preceding formulae for the horizontal and vertical projections to map a skewed cylinder to the perpendicular screen, or you first map the cylinder to the screen (as described earlier) and then skew the screen and map to a second (perpendicular) screen.

Perspective images are a visual illusion; we have a sensation of depth on a flat surface. This illusion can be quite strong too, as Michael Kubovy and others have discovered. The goal in perspective correction, then, is first and foremost to maintain the illusion. Being correct is an objective of less importance. Approximate formulae are acceptable as long as the illusion is not destroyed.

The reason to bring this up is because the “correct” perspective correction formulae require a fair amount of goniometric and trigonometric calculations. One will want to avoid to perform these calculations in real-time.

Tilt is the vertical displacement of the horizon of the panoramic image from the centre of the viewport.

$\tan\gamma = \frac{\Delta y}{R}$ where Δy is the “tilt”.

The general correction for the horizontal projection is:

$$x'' = x' \cdot \left(D \big/ R + 1 \right)$$

where $D \approx y' \cdot \tan\gamma$, giving:

$$x'' = x' \cdot \left(\frac{\tan\gamma \cdot y'}{R} + 1 \right)$$

Replacing tan(γ) by its definition results in:

$$x'' = x' \cdot \left(\frac{\Delta y \cdot y'}{R^2} + 1 \right)$$ R is constant, Δy is constant for a frame

A better approximation of D is:

$$D \approx (y' + \tfrac{1}{2} tilt) \cdot \tan\gamma$$

However, since we only adjust the horizontal coordinate, the original definition of D gives less distortion.

Moving off-centre

When the spectator steps towards the screen, the effect is similar to zooming on a flat picture. However, the horizontal and vertical projections change significantly when they must be corrected

for an off-centre point of view. Although the exact mappings can be calculated, the calculations are probably too costly for real-time operation. Fortunately, a low cost approximate correction gives very acceptable results. Simple zooming is also an option.

Finally: the projection

For every (x,y) in the viewport, calculate (x',y'). Then read the pixel at (x',y') from the panoramic source image and store it at coordinate pair (x,y) in the viewport. In other words:

$$viewport(x, y) \leftarrow source(x', y')$$

This projection ignores tilt and acentric viewing positions. To add tilt, calculate x'' from x' and map from *that* coordinate; an general purpose "zoom factor" (z in the formula below) is usually required as well to make the panoramic image fit a particular viewport size.

$$viewport(z \cdot x, z \cdot y) \leftarrow source(x'', y')$$

As is apparent, this is a "reverse projection": the pixels in the destination (the projection screen) are mapped to the relating pixels in the source image. Reverse mapping avoids holes in the projection, and it avoids "overwrite" in the destination—at the cost of having "overread" in the source. On several systems, overwrite is more costly than overread, due to hardware restrictions (video memory) or memory protection mechanisms in the operating system.

Other considerations

Horizon correction after cropping

In addition to the general projection formulae, a viewer should also correct the horizon. In the original panoramic image, the horizon lies at the vertical centre (because of the requirement that "The camera must rotate on an axis perpendicular to the axis of the lens", see the introduction). After scanning and cropping the image, this need no longer be true, however.

Viewport size & acentric viewing positions

The viewport cannot have any size. The viewport can never display over a 180° view angle, but for practical purposes (and to avoid distortion around the edges), it is best to restrict the view angle to about 90°. The view angle, and thereby the maximum viewport size, depends on the distance that the viewer is from the cylindrical screen. In other words, given a fixed viewport size, the amount by which a viewing position may move off-centre has an upper bound. (For practical purposes, it should also have a lower bound.)

The human eye has a very wide angle, but it can only see details in a much smaller region. So for the human eye, the "angle of vision" is a fuzzy concept. A camera has a well-defined angle. The view angle of a camera that produces rectilinear images rarely exceeds 60°, and is usually closer to 35° to 45°. Therefore, relating to camera, a maximum viewing angle of 90°, as proposed in the previous paragraph, is an unrealistically large angle. These "ultra-wide angles" do not disturb the image, however; in fact, they suggest a greater depth (perspective) in the image.

With a zoom factor of 1.0, and a maximum viewing angle of 90°, the maximum viewport width is ¼ of the image width. The maximum viewport height is the height of the image (assuming that you cannot create images with a vertical angle of more than 90° with a normal lens).

Hot spots (click positions)

It is common to add "hotspots" on a panoramic image, to add interaction to the immersive environment that the panoramic image forms. With the transforms that this paper presents, it is trivial to transform a coordinate pair in the viewport to the related pixel position in the source image. The operations (checking) on hotspots then occur on the source image.

Panoramas and animated objects

When combining panoramic images with sprite animation, it is probably most convenient to first build a new "source image" from the background (panoramic) image and all sprites, and then to "warp" the new source image to the viewport. This ensures that the sprites are properly scaled and warped.

Large (and wide) sprites should be "reverse warped" (like is done by stitching software). The formulae for reverse warping are (trivially) derived from the ones in this paper. For small sprite images, no reverse warping is needed. For sprites that are high but narrow (for example human figures like avatars), simple vertical scaling is sufficient.

Partial cylinders

Since the radius is first derived from the width of the source image (see "The horizontal projection"), the value of x' can never fall outside the image… unless the image is tilted. The easiest way to protect the system from mapping to x'' values that fall outside of the source image is to add a margin to the image. The margin can then be set to any solid colour or pattern that least disturbs the viewing.

The required width of the margin depends on the possible tilt, which at its turn depends on the height of the source image and the height of the viewport.

Doing it quickly...

The introduction claimed that panoramic imaging trades in freedom of movement for speed of display. Nevertheless, the analysis lists several fairly complex computations that every pixel must walk through, including a transcendental function (**arctan**) and a square root. Surely, if these calculations are done for every pixel in a 640×480 viewport (over 300,000 pixels), the refresh rate would be in the order of one frame per 10 seconds, even on a moderately fast PC.

The trick is to extent the formulae given here to *incremental* calculations, where the (x_i,y_i) position forms the basis for the calculation of the neighbouring pixel at (x_{i+1},y_{i+1}), thereby reusing previously calculated results. When making extensive use of tables that hold pre-calculated results and DDA algorithms (DDA = discrete digital analyser) for the zoom and tilt effect, the operations for every pixel can be reduced to addition, subtraction and table look-up operations, all operating on integer data. The simple integer operations can fully exploit the dual pipeline architecture of the Intel Pentium processor family.

A file format proposal

A panorama viewer needs more information than only the bits of the source pictures. At least the viewing angle (the section of the circle that the picture represents) and the horizon offset must be saved with each picture. Hotspot information and the margin size of a partially cylindrical panoramic image are another candidates to be stored in an image. Several file formats, like TIFF, permit the storage of additional (user) information blocks in the image file. Other file formats lack extensibility.

Our proposal is to store the additional information in human readable form in either a separate file or in a "comment" or "description" field of the picture file. Image file formats that support comments or other text strings, or that support general purpose extensions can integrate the extra panoramic information. Image file formats that do not support (text) extensions need an additional (ASCII) file.

The information is stated in a simple syntax with a keyword and a list of parameters between parentheses. The following keywords are currently defined:

horizon(*value*)	in pixels from the centre of the image
angle(*value*)	in degrees

margin(*value*)	in pixels, the margin is assumed to be on the right edge of the image
hotspot(x1, y1, x2, y2, label)	a rectangular hotspot in the image; the label is the next image or an “action” label
orientation(value)	the “compass position” of the left margin of the image in degrees; 0 is north, 90 is east, 180 is south, 270 is west

Conclusion

We judge it feasible to show panoramic images with 256 colours at a resolution of 640×480 at a speed of 15 frames per second on a 90 MHz Pentium equipped PC.

For photographic content, cylindrical panoramas work best. The image quality of spherical panoramas is uneven (if taken with a fish-eye lens). Capturing cubical panoramas requires a lens angle of 90° and at the same time the lens should not distort straight lines. Possibly, scenes captured with a fisheye lens could be postprocessed to create rectilinear images, but this requires further investigation.