# Comparison of Radial and Tangential Geometries for Cylindrical Panorama

Faezeh Amjadi
Université de Montréal
Montréal (Québec), Canada
faezeh.amjadi@umontreal.ca

Sébastien Roy
Université de Montréal
Montréal (Québec), Canada
roys@iro.umontreal.ca

## Abstract

*This paper presents a new approach which builds* 360° *cylindrical panoramic images from multiple cameras. In order to ensure a perceptually correct result, mosaicing typically requires either a planar or near-planar scene, parallax-free camera motion between source frames, or a dense sampling of the scene. When these conditions are not satisfied, various artifacts may appear. There are many algorithms to overcome these problems. We propose a panoramic setup where cameras are placed evenly around a circle. Instead of looking outward, which is the traditional configuration, we propose to make the optical axes tangent to the camera circle, a "tangential" configuration. We will demonstrate that this configuration is very insensitive to depth estimation, which reduces stitching artifacts. This property is only limited by the fact that tangential cameras usually occlude each other along the circle. Beside an analysis and comparison of* radial *and tangential geometries, we provide an experimental setup with real panoramas obtained in realistic conditions.*

## 1. Introduction

People have always been fascinated about capturing view of entire scenes. Panoramic images are a natural way of capturing a wide visual field of view. A panorama is a compact representation of the environment viewed from one 3D position. While a single photograph of a scene is just a static snapshot with a limited field of view captured from a single viewpoint, a panorama combines multiple input images, typically with some overlap, to produce an output with all wide field of views. Many techniques have been proposed to extend the ways in which a scene can be visualized by taking multiple photographs.

A number of techniques have been developed for capturing panoramic images of real-world scenes. Panoramas can be created on an extended planar image surface, on a cylinder [7, 22, 24], or on a sphere [23, 10]. Panoramas can be generated with multiple cameras, possibly combined

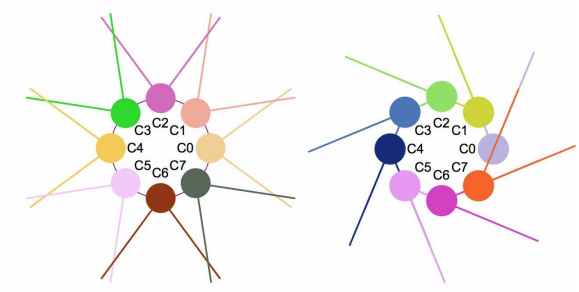



Figure 1: Position of eight camera in two proposed designs, left is Radial model and, right is Tangential model

with mirrors. The simplest mosaics are created by a camera from a set of images whose mutual displacements are pure image-plane translations [7, 19, 18, 22] , or single planar scene in which both center of projection and image sensor are free to translate and rotate [11]. In this method, a camera is mounted on a rotational robotic arm, so the optical center of the camera is offset from the vertical axis of rotation, as shown in figure (1-left). The cameras are looking outward from the rotational center. Panoramic images are generated by repeatedly shifting the rotational arm by an angle which corresponds to a single pixel column of the captured image. By assembling the center columns of these images, we get a mosaic panoramic image. These methods contain little or no parallax effects. Because there is no motion parallax, one can't see the relative depth of points in the scene with rotation, so the images might as well be located on any plane. Another way to build panoramic images is by taking another column out of a captured image and mosaicing the columns. Such panoramic images are called multi-perspective panoramic images [17, 26], because it is possible to change the selected column. The crucial property of two or more multi-perspective panoramic images is that they capture the information about the motion parallax effect since the columns forming the panoramic images are captured from different perspectives. From these panora-

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.81

mas it is possible to recover depth using stereoscopic $3D$ reconstruction and dense sampling, which is time-consuming. Panoramic images can be created by using cameras with a wide field of view (fish-eye) lens. They work for a dynamic scene but the images must undergo substantial distortions, and mapping an entire scene into the limited resolution of a video camera can compromise the image quality[25, 4, 27]. Catadioptric cameras, which rely on mirrors, can also be used for panoramic imaging[15, 14]. Such panoramic cameras are appropriate for low-resolution reconstruction of dynamic scenes and for motion estimation. It is also known that it is possible to keep a camera as a single viewpoint in the cases of hyperbolic mirrors viewed for perspective cameras and of parabolic mirrors viewed from orthographic cameras [21].

The other way to produce a panorama is to capture images simultaneously from multiple cameras, with some overlap. Creating a panorama without overlap require specialized algorithms. These techniques have to decide where to cut the images to remove the overlaps. Since the overlap depends on scene depth, this is a difficult problem. This is why we also need to rely on algorithms to seamlessly blend overlapping images, to make cutting error less visible, and also reduce the problem of scene motion and lens distortion. Early methods estimate a 2D transformation, a homography, between two input images to align them [23, 5]. When the input images have little parallax, a homography works well, but when the input images feature larger parallax, artifacts like ghosting and blind spots happens. Since a homography cannot account for parallax, these methods require images that share a common single viewpoint.

Local warping [20] guided by motion estimation can be used to reduce the overlap problems. Also, advanced image composition techniques, such as seam cutting [3, 12] and blending [6, 16], can reduce the ghosting artifacts. However, they cannot address significant misalignment. Recent image stitching methods use spatially-varying warping algorithms to align input images [13, 28]. While they can handle parallax better than homography, they still have difficulties in the presence of large parallax. The recent dual-homography warping method [9] can stitch images with parallax, but it requires the scene content to be modeled by two planes.

Another method is Parallax-tolerant Image Stitching [29] which first aligns input images, then use a content-preserving warping and seam cutting algorithm to find a seam to piece aligned images together and explores both image content and geometric alignment and finally employ a multiband blending algorithm to create the final stitching result. This method handles images with large parallax well, but it fails when input images have very large parallax or are full of salient structures.

In this paper, we intend to work on the geometry of cameras instead of developing stitching algorithms. When capturing hand-held panoramas or using an array of cameras, parallax has to be accounted for. A common strategy, which is also the basis for our work, is to change the orientation of cameras to reduce these artifacts. Our goal is to compare various camera geometries and figure out in which conditions we can obtain the highest qualities panorama with the least algorithmic effort. The general design of our system places all cameras on a common circle, symmetrically and all at the same distance from the center of the circle. Each camera optical axis will varies from *radial* configuration to *tangential* configuration (see figure(1-right)).

In the *radial* configuration, optical axes are perpendicular to the camera's circle (they radiate from the center). In the *tangential* configuration, the optical axis of each camera is tangential to camera's circle. In both configurations each camera has the same position, only the optical axis varies. To the best of our knowledge, these issues have not been explored in previous works.

In this paper, we will show that these configurations respond very differently to ghosting, blind spots and other artifacts induced by parallax. Section 2 introduces the camera model. Section 3 will present the calibration of the camera setup. Section 4 explains the panorama generation and section 5 presents experimental result with real cameras.

## 2. Camera Model

In this section, we present our panorama camera models. They are designed to generate a panorama from a number of cameras that are placed on the circumference of a circle, all equidistant from each other and at the same distance from the center. In other words, the angle between two consecutive cameras is the same. The number of cameras must be sufficient to get an overlap between successive images. In this model, all cameras have the same optical axis orientation relative to the normal of the camera circle. We consider two cases. First, the optical axis is parallel to the normal, yielding a *radial* configuration as shown in figure (1-Left). Second, the optical axis is perpendicular to the normal, yielding a *tangential* configuration as shown in figure (1-Right). Various optical axis orientations from the radial to the tangential configuration are also possible.

All the camera configurations provide a similar panorama but mostly differ in their sensitivity to parallax. Our panorama geometry is cylindrical, which is commonly used because of its ease of construction. In practice, a single camera can be rotated around the circle center to provide an image sequence. Also, it is possible to use multiple fixed cameras to provide a similar sequence. For static scenes both approaches work fine, but for dynamic scenes multiple cameras are required because they can capture simultaneously. Assuming each camera is properly calibrated, all perspective images can be warped into cylindrical co-

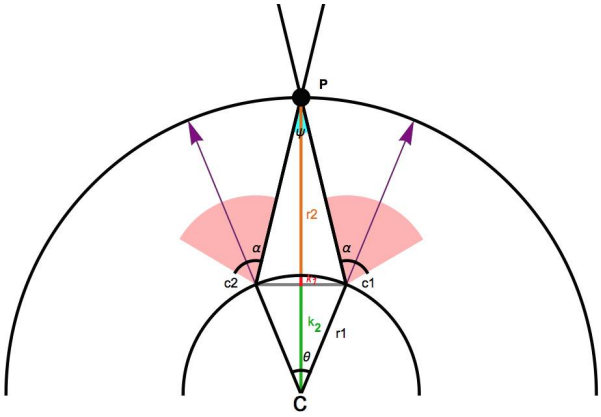



Figure 2: Radial camera model, showing 2 of 8 cameras. The FOV of each camera must be increased from the minimum ($45°$) to account for the position of $P$.

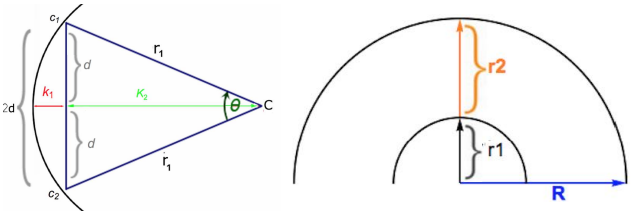
ordinates, so they can be stitched together. Since parallax is present, knowledge of depth will have an effect on the stitching process.

Suppose we have two different circles with the same center $C$. The first circle with smaller radius $r_1$ is where the cameras ($c_1$ ,$c_2$) are located, as shown in figure 2. The second circle of radius $R$, represents the reference depth for stitching. The difference between the two radiuses is $r_2$, such that $r_1 + r_2 = R$, as shown in figure (3-b). Assuming we have $n$ cameras, each one is located on the circle of radius $r_1$, with an angle $\theta = \frac{2\pi}{n}$, so they are equally spaced on the circle. We define the distance between two consecutive cameras as $2d$ and the height of the triangle $< c_2, C, c_1 >$ as $k_2$, as shown in figure (3-a). Also we define $k_1$ such that $k_1 + k_2 = r_1$. The field of view for each camera is $\alpha$ and a reference point $P$ is located between the cameras at distance $R$. The optical axis (The purple arrows are in figure 2) can be rotated from $0°$ to $90°$ to obtain the *radial* and *tangential* configurations respectively. For stitching, we have to cut both images at the projection of point $P$. In practice, we cut along a vertical line on the cylinder which contains $P$.

$$r_1 + r_2 = R. \qquad k_1 + k_2 = r_1. \tag{1}$$

Notice that in Figure 4 moving point $P$ to a different depth along the red line does not have any effect on the stitching. This lack of sensitivity to depth is a very desirable property, which makes stitching free of artifact.

The insensitivity to depth only occur at the epipole but it will increase with the distance from the epipole in the image. Since we cut along a vertical line, some vertical parallax will occur on that line, but the *radial* configuration will produce more displacement everywhere since all the points on the cutting line are further from the epipole compared to the *tangential* configuration. Obviously, we only care about the magnitude of the displacement, not its direction. In monoscopic panoramas, stitching becomes more difficult when corresponding pixels are farther apart, regardless of the direction of the separation. Also, the perception of stitching artifact is conditioned on the magnitude of the error, not its direction. In stereo, the perception of depth will be distorted mostly by the parallax between the left and right panoramas, not by the parallax induced by successive cameras in the panorama for a single eye.




(a) (b)

Figure 3: Closeup of the circle of the camera models

In the *radial* setup, the horizontal fov of the cameras has to be at least $\frac{2\pi}{n}$ degrees, which represents the minimum achievable, when everything is at infinity. For *tangential*, the angle is the same but works for all object depth. This is because when we cut along a vertical epipolar line, all points visible on one side of that cut will stay on that side regardless of their depth, so $\frac{2\pi}{n}$ is the maximum fov needed for the *tangential* setup. This property is truly ideal but there is a problem: you can't see epipole without seeing the other camera. The next section discusses this camera hiding problem.

## 2.1. Camera hiding Problem

In the ideal *tangential* model of figure 4 we assume that cameras are invisible or of negligible size. In practice, cameras will take up space and become visible from the next

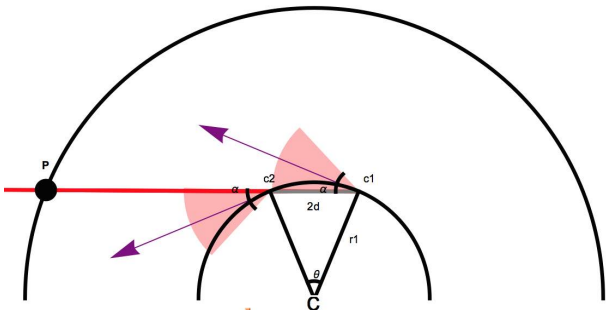



Figure 4: Ideal tangential model (assuming negligible camera size) for eight camera setup. The FOV is set to the minimum achievable( $45°$ for $n = 8$).

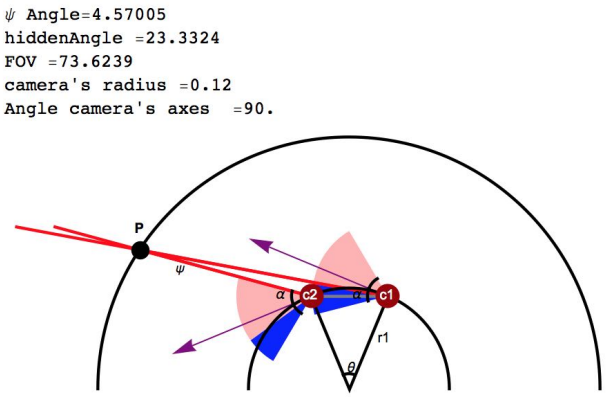



Figure 5: Non ideal tangential model, for realistic camera size. The fov has to be increased from the minimum ($45°$ for $n = 8$) to account for $P$. Note that all angles in top of image are in degree

camera at the cutting line . Figure 5 illustrates what happens when camera size is taken into account.

To remediate this, the cutting point $P$ must be moved until the camera is not visible anymore. This makes the model less than ideal and will weaken the depth insensitivity property. To accomplish this, the fov of the camera must be increased to make point $P$ visible on the right side of the image. At the left of the camera image, the other camera is visible but not needed (blue area in figure 5). Cutting at point $P$ will remove it and allow reducing the field of view, as illustrated in figure 7. This compensates the increase of fov on the right, so the resulting fov is only slightly increased.

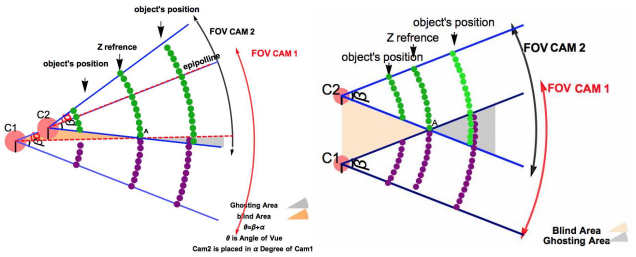
In the *radial* model, it is necessary to have an fov larger than $45°$ (for eight cameras) to cut point $P$. In our example of figure 2, it is set to $73°$. In an ideal *tangential* model, as shown in figure 4, cameras are rotated $90°$ and the fov is $45°$, but in order to account for camera size, the fov is set to $73.6°$ as shown in figure 5. Notice the hidden area that is defined by blue color and called hidden angle. This angle depends on the camera size (radius $r_c$) and distance $d$ between two cameras :

$$\sigma = \arcsin(\frac{r_c}{d}).$$

We can calculate the size of the hidden region (in blue in figure 5) from the horizontal hidden pixel width $w$ in the image and the focal length $f$:

$$\psi = 2\arctan(\frac{w}{2f}).$$

We have the option to rotate the cameras until the hidden angles are removed, as shown in figure 7. The cameras are rotated to $77°$ instead of $90°$ and the fov becomes $46.6°$, for the same camera size as in figure 5.




(a) Tangential Model (b) Radial Model

Figure 6: Blind spot and Ghosting area in two models according to a depth reference. The gray color is ghosting area that seen by both cameras and light orange parts are blind spot that not seen by these cameras.

## 2.2. Artifact Angle

The reference depth, where the point $P$ is located, is an important parameter in our configuration. If we observe an object at a depth that corresponds to the reference depth, its location in the panorama will be coherent between cameras and easy to stitch. If the reference depth is farther or closer than the real depth, ghosting or blind spot will occur, respectively, in the panorama image, as illustrated in the gray and orange areas in figure 6. We refer to the angle of this area as the "artifact angle". If this angle is zero then there is no parallax and therefor no impact on stitching. If this angle is small, then the effect of parallax will be small, so the impact of a bad depth estimate will be less visible. The magnitude of the artifact angle is related to the optical axis orientation. It is maximal in the *radial* configuration and minimal in the *tangential* as shown in figure 6.

The artifact angle can be computed directly from the camera model. First, we present how to derive it for the *radial* configuration, as shown in figure 2. The artifact angle $\psi$ is obtained as follows :

$$\begin{aligned} \sin(\frac{\theta}{2}) &= \frac{d}{r_1}. \\ \tan(\frac{\omega}{2}) &= \frac{d}{r_2 + k_1}. \end{aligned} \quad (2)$$

From equations 1 and 2,

$$\begin{aligned} R &= \frac{d}{\sin(\frac{\theta}{2})} + \frac{d}{\tan(\frac{\psi}{2})} - k_1. \\ \psi &= 2.\arctan(\frac{d}{R + k_1 - \frac{d}{\sin(\frac{\theta}{2})}}). \end{aligned}$$

As $\theta$ and $d$ are constant, $\psi$ only depends on $r_2$ and $r_1$.

For the *tangential* model as shown in figure 7, we can also compute $\psi$ using additional parameters. We define length $L$ as the distance between point $P$ and first camera

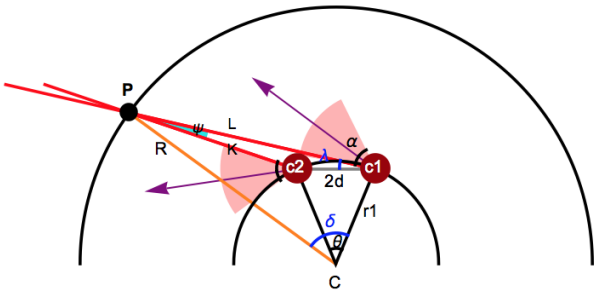



Figure 7: Tangential model (eight cameras) with minimal camera FOV (46.6°), with rotated cameras (77°). The artifact angle $\psi$ is (4.57°)

$c_1$, length $K$ as the distance between point $p$ and second camera $c_2$, angle $\delta$ between $r_1$ and $R$, and angle $\lambda$ between $L$ and the line connecting $c_1$ and $c_2$. Suppose we have a triangle $< P, C, c_1 >$, and we know the angle of rotation of point $P$, then $\delta$ is known, so it is possible to compute $L$ and $\lambda$ from equation 3 and 4 respectively:

$$\cos(\delta) = \frac{r_1^2 + R^2 - L^2}{2Rr_1}. \quad (3)$$

$$\cos(\lambda + (\frac{\pi}{2} - \frac{\theta}{2})) = \frac{r_1^2 + L^2 - R^2}{2Lr_1} \quad (4)$$

Once $\lambda$ is obtained from equation 4, then we can compute $\psi$ with the help of equation 5 from the triangle $< P, c_1, c_2 >$ to find $K$. In our example in figure 2, artifact angle is 27.29°. Because of the rotation of the camera, the blinding and ghosting areas in *tangential* model are smaller than *radial* model. We have:

$$\cos(\lambda) = \frac{(2d)^2 + L^2 - K^2}{2(2d)L}. \quad (5)$$

$$\cos(\psi) = \frac{K^2 + L^2 - (2d)^2}{2KL}. \quad (6)$$

$$\psi = \arccos(\frac{K^2 + L^2 - (2d)^2}{2KL}). \quad (7)$$

In fact the two derivation of $\psi$ are related. In our example of figure 7 the artifact angle is 4.52°. In the tangential derivation, if point $P$ is allowed to rotate around center $C$, then both derivation are equal when $P$ is above $c_1$ and $c_2$ (rotated zero degree) as in figure 2. When point $P$ is rotated until it is collinear with $c_1$ and $c_2$, then $\psi$ is zero. This is illustrated in figure (8-a), where we observe that the *radial* configuration is the most sensitive to parallax and this sensitivity decrease as the geometry becomes more tangential. Notice that point $P$ can be collinear to $c_1$ and $c_2$ (which happens near 160°in the example of figure (8-a)). This situation does not occur in practice since near that position one camera will hide the point $P$ from the other camera. Also the impact of camera size on the artifact angle $\psi$ can be seen in figure(8-b), so for practical camera sizes, we observe that the angle $\psi$ is very small.

## 3. Calibration

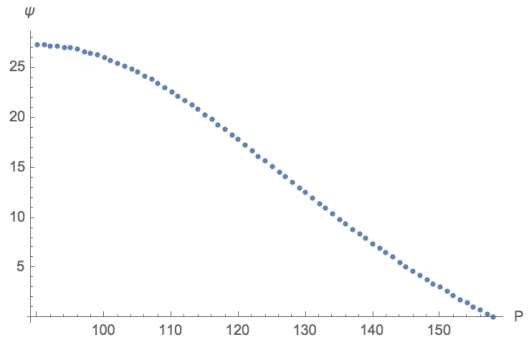
Proper calibration is an important part of using any multiple cameras setup, where we estimate the intrinsic and extrinsic parameters of the cameras. Can be found through this process[30].

First, the intrinsic parameters are estimated using the classical checkerboard method [30], applied to each camera individually.

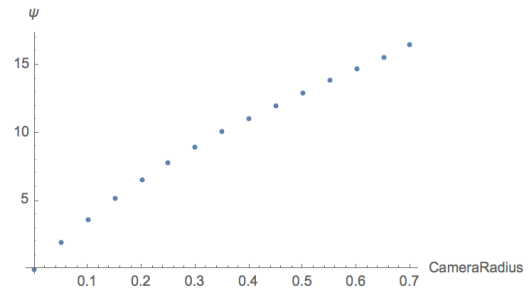
Determining the extrinsic parameters is more difficult, mostly because of the small overlap between views, and because the cameras form a closed loop which can't be solved linearly in a single step.

In order to solve the small overlap problem, we rely on Structured-light on a Lcd monitor to replace the checkerboard and establish a dense correspondence between views [8]. While the density is very good, the correspondences can only be established between two successive cameras.

To obtain a full calibration, we first estimate the extrinsic parameters between two successive cameras ($i$ and $i+1$) which yield rotation $R_i$ and translation $T_i$, with $i$ from 1

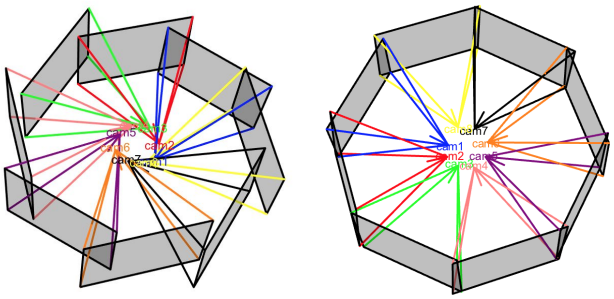



(a)




(b)

Figure 8: Artifact angle $\psi$ as a function of the angle (in degrees) of point $P$ (a) and as a function of camera size relative to the inter-camera distance (b)



(a) Tangential Model (b) Radial Model

Figure 9: Position of all cameras after calibration.

to 8. Because we have a closed loop, $R_8$ and $T_8$ are the rotation and translation from camera 8 to camera 1.

The reference system is camera 1, so the final transformation $M$ for each camera is

$$\begin{aligned} M_1 &= K_1[I|0] \\ M_2 &= K_2[R_1|t_1] \\ M_3 &= K_3[R_2|t_2].[R_1|t_1] \\ &\dots \\ M_8 &= K_8[R_8|t_8].[R_7|t_7].\dots.[R_1|t_1] \end{aligned}$$

However theses transformations are inaccurate and must be optimized globally. For each camera, we define a correction transformation. Using the various correspondences obtained from structured light, we can estimate the reprojection error in pixels, and optimize this error globally over the correction transformations. This yields the optimal camera geometry, shown in figure 9, which we use for stitching. Note that we do not optimize the internal camera parameters, since they are obtained first and assumed of good precision.

## 4. Panorama generation

After estimating the relative camera poses, a panorama can be generated by projecting all images into a common surface. A traditional way to do this is to choose either a cylindrical or spherical map. Cylindrical panoramas are commonly used because of their ease of construction. When the cameras are calibrated, each perspective image can be warped into cylindrical coordinates and the color is associated with each pixel is computed by first deriving a 3D ray from the pixel position, and then mapping this ray into each input image through our known transformation. The cylinder parameters (center, axis, radius) are important. The obvious choice for the center is the average of all camera centers. The average $Y$ axis of all cameras can be used as the cylinder axis. Since the radius represents depth, its selection is the most difficult aspect of stitching. Estimating depth by stereo is impossible because of the small overlap between cameras. However, at the overlap, a wrong depth estimate will induce a visible artifact which magnitude is related to the artifact angle (see section 2.2). The *tangential* geometry will be much more resilient to a bad estimate of depth, thereby making it possible to use a single depth for a whole panorama without much visible artifact. This not only simplifies stitching but can make it real-time.

Each pixel in a panorama can be classified as visible or not visible from the cameras, and when it is visible, it can be from a single camera or two cameras. Usually, overlaps between cameras are visible by two cameras, but in a tangential panorama part of the overlap is obstructed by a camera body, so it effectively becomes partially visible as a single camera. This is illustrated in figure 10 for *tangential* model and figure 11 for *radial* model.

Figure 12 illustrates the impact of depth estimation. Consider an object that is seen by two cameras in a tangential panorama (top of figure 12-a), and radial panorama (bottom of figure 12-a). The object is highlighted by a red rectangle. Starting from the correct depth of this object, we varied the depth of the panorama point from much closer to much further. The reprojections in both images, relative to the correct point match, are displayed superimposed together in figure (12-b) for tangential, and figure (12-c) for radial panoramas. Clearly, the wrong depth estimation induces much more separation between the reprojected points in the *radial* case than the *tangential* case. The pixel distance is also illustrated in figure (12-d), where the pixel separation is clearly larger for the radial panorama. In this example, the camera image dimension is $1296 \times 972$.

## 5. Experiments

We implemented the proposed panorama system using 8 raspberry pi cameras (figure 14-a) setup in a tangential and radial arrangement illustrated in figure (14-b) and c, respectively. The cameras feature a focal length of 3.6 mm, a $1.4 \times 1.4 \mu m$ pixel size. The image resolution is $2592 \times 1944$ pixels with a horizontal field of view of about $53°$, and vertical field of view of $40°$. The camera itself has a size of $8 \times 8 \times 4$ mm [2]. The disposition of the cameras is a circle about $10cm$ diameter. In practice, the *radial* configuration could use a smaller diameter, but the *tangential* can't since reducing the inter-camera space increase the relative size of the camera occlusion. However, using a large diameter has potential applications for stereoscopic panoramas, so it is relevant [1].

Sample panoramas are provided in figure 13. Both are built using a single depth for the whole scene, without any adjustments. Moreover, the stitching in overlap areas is a simple average of both images. This is done on purpose

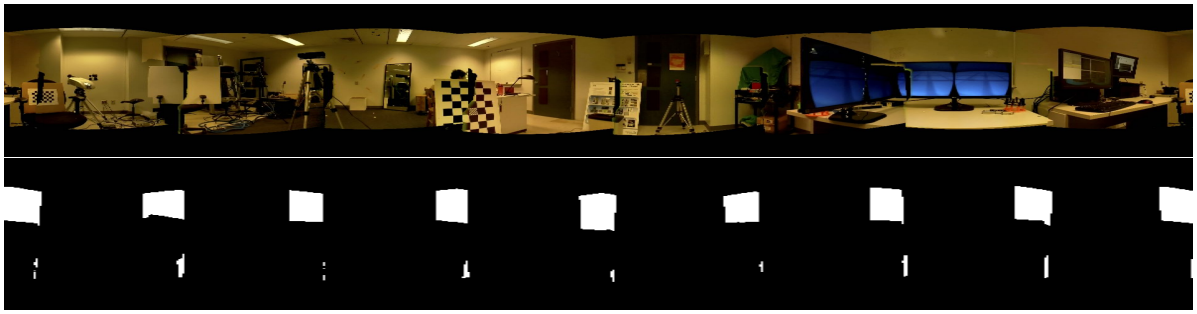


Figure 10: Sample tangential panorama (top) with mask (bottom) illustrating overlapping pixels (seen by two cameras). Smaller overlap is caused by camera occlusion.

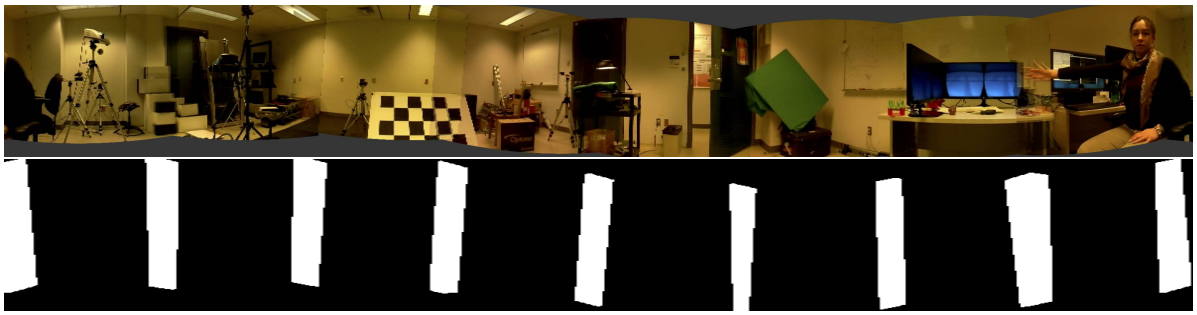
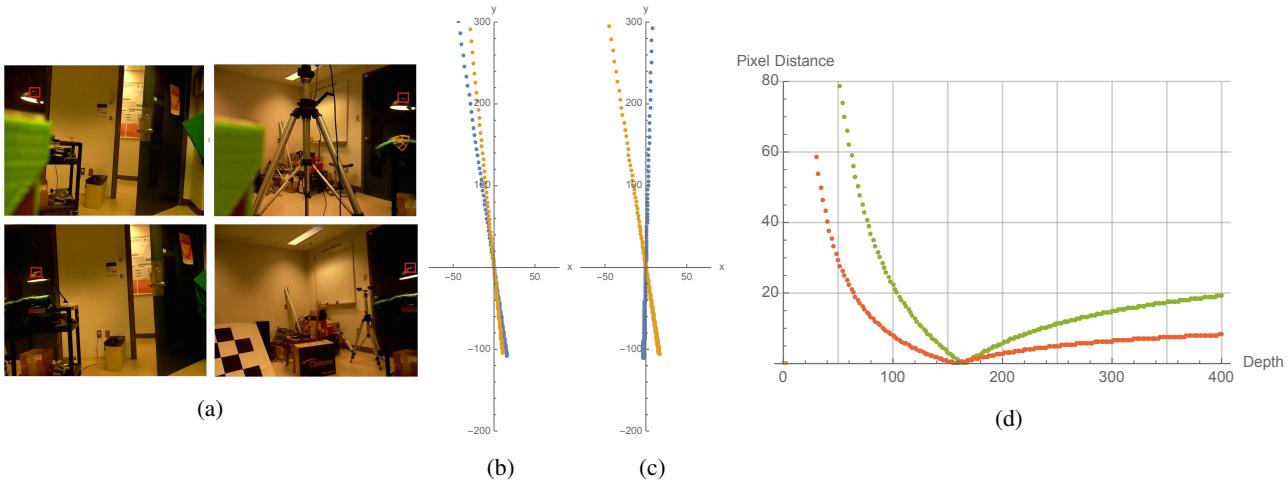


Figure 11: Sample radial panorama (top) with mask (bottom) illustrating overlapping pixels (seen by two cameras).

to make visual artifact standout. The tangential panorama (top) is generally cleaner, with less visual artifacts than the radial panorama (middle). Notice that some pixels in the tangential panorama are blacked out. This is caused by the field of view which was slightly smaller than required for this geometry. Since we used a perfect *tangential* geometry, it is possible to rotate the cameras slightly to reduce the field of view requirement and solve the issue (see section 2.2). Since the amount of rotation depends on the apparent camera size, which was unknown when building the camera support, we preferred perfect tangential rotation .

One closeup example of stitching error is provided at the




(a) (b) (c) (d)

Figure 12: A) Object seen from two cameras in a tangencial setup (top) and radial setup (bottom), highlighted by a red rectangle. B) Image reprojections for various depth, relative to the correct match, for tangencial setup and C) for radial setup. D) Pixel distances as a function of depth, for tangencial (red) and radial (green). Correct depth is 160.

bottom of figure 13. Clearly, the *tangential* geometry (left) provides a sharper result.

## 6. Conclusion

This paper presented a new approach to generating a multi-camera cylindrical panorama. The proposed geometry is called *tangential* because each camera has its optical axis tangent to the circle of cameras. Beside the fact that this configuration will induce occlusion from one camera to the next one along the camera circle, we demonstrate that this configuration is much less sensitive to the wrong estimate of scene depth. This allows the stitching process to be simplified enough to provide very good panoramas in real-time using a single depth estimate for a whole scene. Examples are provided for an experimental setup of 8 cameras, but we anticipate that stereoscopic panoramas will be easy to build with this configuration, and would provide real-time images.

## References


[1] Jump camera rig. https://vr.google.com/jump/. 6

[2] Raspberry Pi information and products for custom applications and engineering. http://www.truetex.com/raspberrypi. 6

[3] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 294–302. ACM, 2004. 2

[4] A. Basu and S. Licardie. Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16(4):433–441, 1995. 2

[5] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007. 2

[6] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4):217–236, 1983. 2

[7] S. E. Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38. ACM, 1995. 1

[8] V. Couture, N. Martin, and S. Roy. Unstructured light scanning to overcome interreflections. In *2011 International Conference on Computer Vision*, pages 1895–1902. IEEE, 2011. 5

[9] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 49–56. IEEE, 2011. 2

[10] S. Gumustekin and R. W. Hall. Mosaic image generation on a flattened gaussian sphere. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pages 50–55. IEEE, 1996. 1

[11] H.-C. Huang and Y.-P. Hung. Panoramic stereo imaging system with automatic disparity warping and seaming. *Graphical Models and Image Processing*, 60(3):196–208, 1998. 1

[12] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003. 2

[13] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 345–352. IEEE, 2011. 2

[14] S. K. Nayar and A. Karmarkar. 360× 360 mosaics. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 388–395. IEEE, 2000. 2

[15] S. K. Nayar and V. Peri. Folded catadioptric cameras. In *Panoramic vision*, pages 103–119. Springer, 2001. 2

[16] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. 2

[17] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 199–206. ACM, 1998. 1

[18] H. S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, and V. Paragano. Videobrush tm: Experiences with consumer video mosaicing. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 56–62. IEEE, 1998. 1

[19] H.-Y. Shum and R. Szeliski. Panoramic image mosaics. Technical report, Citeseer, 1997. 1

[20] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Computer Vision, 1998. Sixth International Conference on*, pages 953–956. IEEE, 1998. 2

[21] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1):23–37, 2002. 2

[22] R. Szeliski. Video mosaics for virtual environments. *IEEE computer Graphics and Applications*, 16(2):22–30, 1996. 1

[23] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997. 1, 2

[24] R. Tsai and T. Huang. Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1147–1152, 1981. 1

[25] J. Weng, P. Cohen, M. Herniou, et al. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on pattern analysis and machine intelligence*, 14(10):965–980, 1992. 2

[26] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin. Multiperspective panoramas for cel animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 243–250. ACM Press/Addison-Wesley Publishing Co., 1997. 1

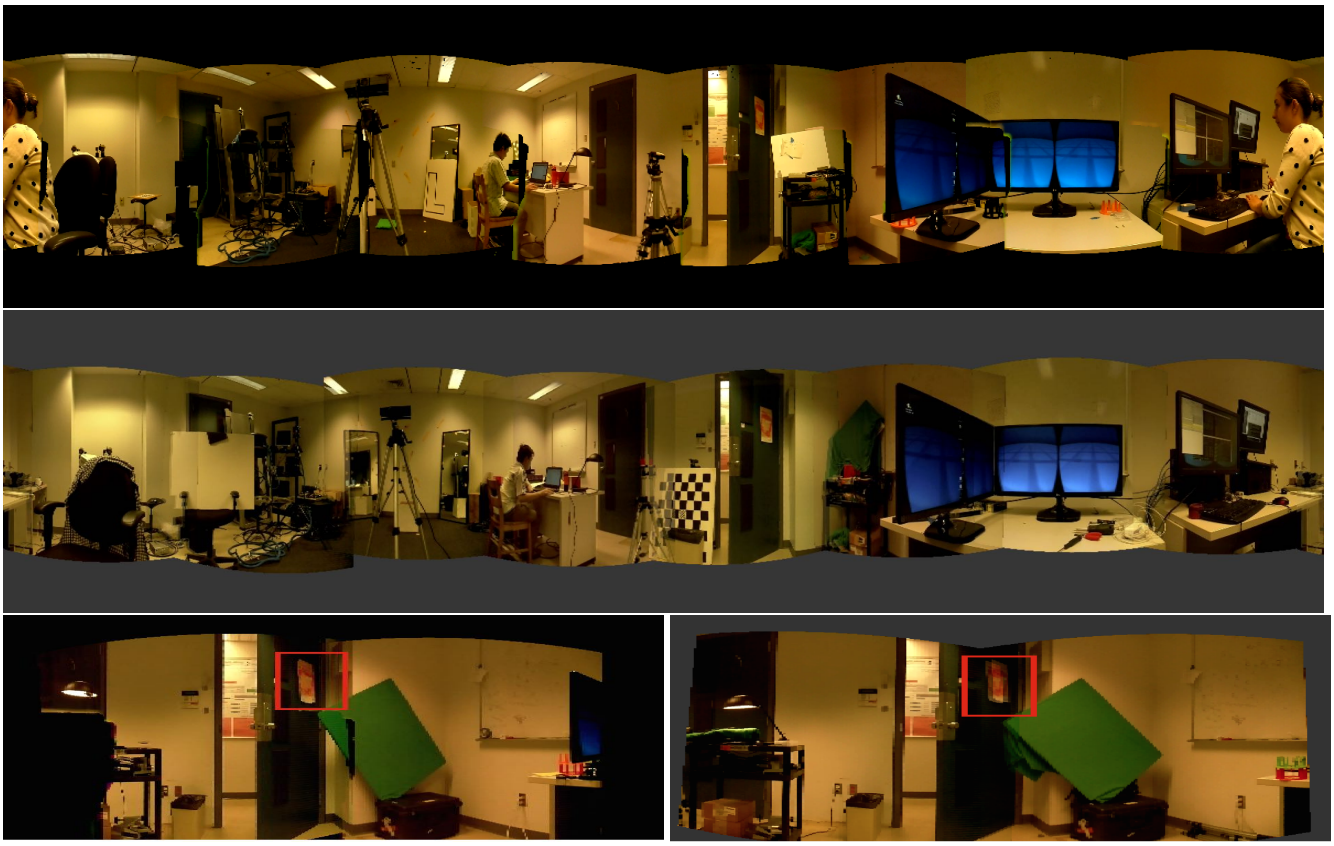
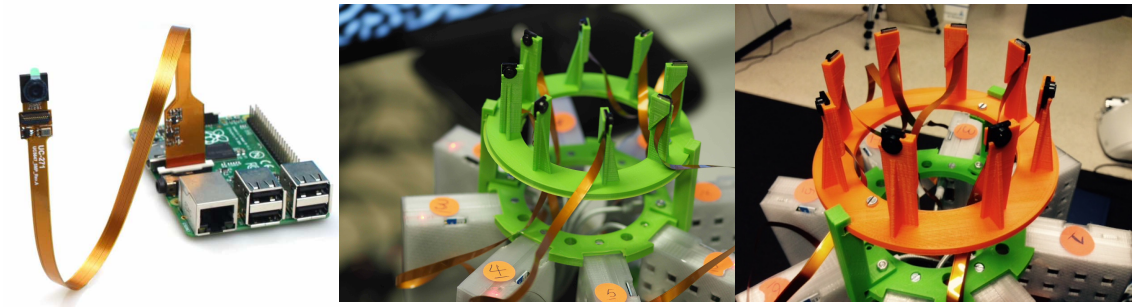


Figure 13: Sample panoramas with tangential model (top) and radial model (middle). At bottom, a single object highlighted with a red rectangle illustrates the difference between tangential model (left) and radial model (right).



(a) Raspberry Pi (b) Tangenciel model (c) Radial model

Figure 14: Panoramic camera setup.


[27] Y. Xiong and K. Turkowski. Creating image-based vr using a self-calibrating fisheye lens. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 237–243. IEEE, 1997. 2

[28] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving dlt. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2339–2346, 2013. 2

[29] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3269, 2014. 2

[30] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 5