The Omnipolar Camera: A New Approach to Stereo Immersive Capture*

Vincent Chapdelaine-Couture and Sébastien Roy
Département d'Informatique et recherche opérationnelle
Université de Montréal
Montréal (Québec), Canada
{chapdelv,roys}@iro.umontreal.ca

Abstract

We introduce in this paper a camera setup for stereo immersive (omnistereo) capture. An omnistereo pair of images gives stereo information up to 360 degrees around a central observer. Previous methods to produce omnistereo images assume a static scene in order to stitch together multiple images captured by a stereo camera rotating on a fixed tripod. Our omnipolar camera setup uses a minimum of 3 cameras with fisheye lenses. The multiple epipoles are used as locations to stitch the images together and produce omnistereo images with no horizontal misalignments due to parallax. We show results of using 3 cameras to capture an unconstrained dynamic scene while the camera is travelling. The produced omnistereo videos are formatted to be displayed on a cylindrical screen or dome.

Figure 1. The omnipolar camera setup has no moving parts and is composed of at least 3 cameras with fisheye lens oriented in the same direction.

1. Introduction

Traditional stereo imaging use two cameras separated on a baseline to capture two slightly different viewpoints looking in the same direction. The stereo image pair can then be projected on a stereo display and fused by the human brain to get strong cues to scene depth.

The goal of omnistereo imaging is to provide stereo cues for up to 360 degrees around an observer. Omnistereo images can be used for navigation in a virtual environment such as CAVEs [8, 7], but without the need to track the head orientation [14, 3, 4]. For example, these images could also be used in a stereo-video extension of Google Street View [2] in which a user could pan over the 360 degree field of view using a stereo monitor.

The difficulty in capturing omnistereo images is that capture cannot simply be done using two cameras side by side. Even with fisheye lenses, such a capture would provide maximum stereo information on the median line (perpendicular to the baseline) but not stereo information along the baseline [16].

This paper has three main contributions. First, we introduce the omnipolar[1] camera setup composed of a minimum of 3 cameras with fisheye lens oriented upwards (see Fig. 1) to capture a scene in omnistereo. Every visible point surrounding the setup is captured in stereo by at least 2 cameras in nearly a 180 degree field of view from about the height of the cameras and above. Note that the camera setup could also be oriented downwards to capture a scene under it.

Second, we present a stitching method which takes images of our camera setup and produces an omnistereo pair of images with no horizontal misalignments due to parallax. Stitching is done at vertical planes passing by the epipoles in neighboring camera images.

Finally, we propose to use the idea of "epipolar stitching" to design a high resolution monocular camera setup using standard lenses. Such a setup would reduce misalignments due to parallax usually observed in systems like the Ladybug [17].

*This work was included in the US Provisional patent application 61760383 titled "OMNISTEREO IMAGING".

[1] *Omnipolar* is meant as a reference to *omnistereo* and *epipoles*.

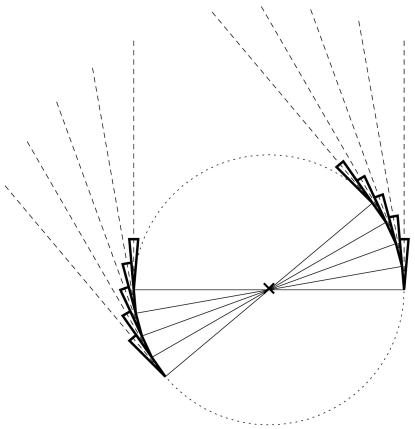


Figure 2. The capture process of the slit-camera setup showing two parallel cameras being rotated. Note that the 5 stereo slits are captured sequentially, i.e. not at the same time.

The paper is organized as follows. Section 2 gives a brief overview of previous work on omnistereo camera setups and lenses. Section 3 presents the omnipolar camera model and how to calibrate it. Section 4 describes a novel stitching method that uses epipoles to produce an omnistereo pair of images. We also discuss issues related to vertical parallax and stereo distortions, and how the stitching method could be applied to a monocular camera setup. Section 5 gives results on a few scenes. We conclude in Section 6.

2. Previous Work

Omnistereo imaging of static scenes was introduced to allow a robot to discover its environment [11]. Standard methods typically capture several vertical *slits* images, i.e. small horizontal field of view images captured from cameras rotating off-axis [10, 11, 14, 16, 19]. All captured rays are tangent to a viewing circle, but in opposite directions for the left and right views (see Fig. 2). Methods for synthesizing omnistereo images typically accumulate 100 or more of these stereo slit images, and stitch them together to produce a larger field of view stereo pair. Each of these stereo slits give maximum stereo information because they capture scene points on the median line perpendicular to the baseline. The use of a very high number of the images also minimizes the change in camera position between consecutive images, thus reducing artifacts due to parallax.

While these slit-based methods work well for static scenes, they do not generalize well to dynamic scenes. The main issue with this approach is that all images are captured at different times. This can lead to monocular shearing of motion and wrong stereo disparities. Each camera's slit captures each visible scene point at some time, but there is no guarantee that each scene point is captured in stereo *at the same time*. Indeed this simultaneity condition is only met if the scene point happens to be where the slits are converging in stereo.

A few ideas on producing omnistereo lenses were presented in [16]. One idea involves using a catadioptric mirror [15] with a Fresnel-like lens built with thousands of lens segments whose diacaustic yields a set of rays tangent to a viewing circle (similar to Fig. 2). No prototypes have been produced as practical engineering issues remain, such as chromatic aberration for instance. Another idea combines a catadioptric mirror with a spiral-shaped mirror to make all captured rays to be tangent to the viewing circle. By using two such spiral mirrors, one could capture an omnistereo view covering up to 132 degrees of field of view without self-occlusion. Thus, three such camera setups are necessary to cover a full 360 degrees, although making a seamless omnistereo image from these is not obvious.

In this paper, we expand on the method presented in [6] that use larger field of view images to produce omnistereo videos with localized motions. Misalignments due to horizontal parallax are removed by stitching left/right borders of two images when these borders are collinear with both corresponding camera positions. More generally, stitching can be done where the baseline joining two camera positions intersects the images because there is no parallax at these points. These intersections are called the epipoles. This principle leads us to present our camera setup in the next section. This camera setup, which has no moving parts, will be used to capture arbitrary motions in omnistereo.

3. Omnipolar Camera Setup

This section introduces the omnipolar camera model composed of N cameras spaced equally around a circle, and then describes the calibration process.

3.1. Projection model

The projection model defines how points in the world are mapped to camera pixels. We assume that each camera uses single viewpoint fisheye lens and that all cameras look in the same direction (see Fig. 1). Let (o_x, o_y) be the principal point (image center of a camera). Each pixel $p = (x, y)$ is mapped onto a unit sphere using angle ϕ around the image center and angle θ related to the lens field of view:

$$\phi = \arctan(y - o_y, x - o_x) \quad (1)$$

$$\theta_d = \frac{||(x - o_x, y - o_y)||}{f}. \quad (2)$$

Angle θ_d represents a distorted value of θ in an equidistant projection model for which pixels are directly proportional to angles. We model θ as a function of θ_d and distortion coefficients k_1, k_2 in a polynomial function [18, 5]:

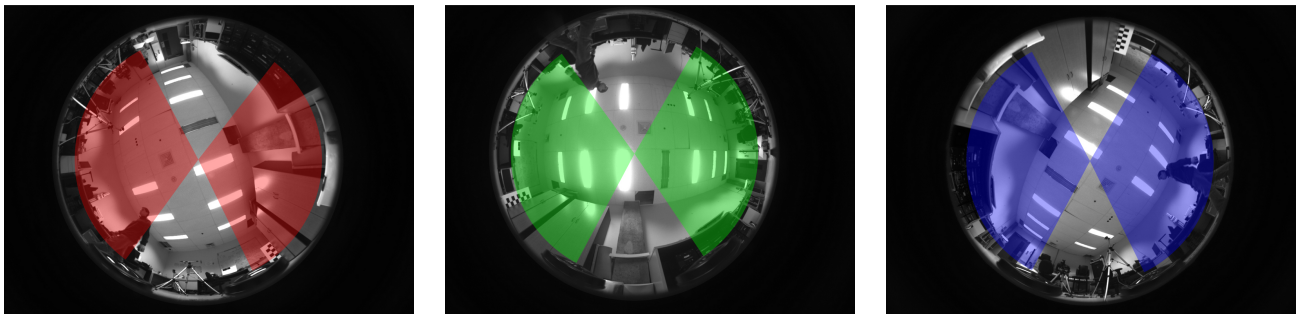


Figure 3. An input image of each camera in an omnipolar setup composed of 3 cameras. In each view, the other two cameras can be seen at the bottom of the field of view. The colored bowties indicate the regions that are used to produce an omnistereo pair of images (the left/right hand side is used to produce the image for the left/right eye). Each side of the bowtie spans about 120 degrees.

$$\theta \quad = \quad \theta_d(1 + \theta_d^2 k_1 + \theta_d^4 k_2). \tag{3}$$

Pixel values can be mapped to rays in camera space as:

$$p_c \quad = \quad \mathbf{\Omega}_y(\phi)\mathbf{\Omega}_z(\pi - \theta)(1, 0, 0, 1)^T \tag{4}$$

where $\mathbf{\Omega}_y$ and $\mathbf{\Omega}_z$ are rotation matrices w.r.t. the y-axis and the z-axis respectively. Furthermore, we model that the camera positions lie on a unit circle in the x-z plane, and that each camera's up vector is normal to this circle. A unit ray in camera space is thus mapped to a world point using the following relation:

$$p_w \quad = \quad \mathbf{R}_y\mathbf{T}_z\mathbf{R}_{xz}Zp_c \tag{5}$$

where $\mathbf{R}_y$ is a rotation matrix w.r.t. the y-axis that defines the position on the circle, $\mathbf{T}_z$ is a translation along the z-axis that models the circle radius, $\mathbf{R}_{xz}$ is a combination of two rotations w.r.t. the x and z-axis respectively. The Z parameter is the depth seen at pixel (x, y) which is unknown *a priori*.

3.2. Calibration

To calibrate the N cameras, we first manually locate the center (o_{x_i}, o_{y_i}) in each circular fisheye image i, for $i \in 1, 2, \ldots, N$. Fig. 3 shows an example of input images for $N = 3$. The colored bowties, centered at (o_{x_i}, o_{y_i}), indicate the regions that will be used to produce the omnistereo pair of images (see Sec. 4). It remains to calibrate 11 parameters, namely the internal parameters f, k_1 and k_2 which we fix to be the same for all lens, and rotation angles r_{x_i}, r_{y_i} and r_{z_i}, except r_{y_1} which we force to be 0. These parameters are estimated in a bundle minimization using an over-constrained system of about $M = 20$ features $p_i^j = (x_i^j, y_i^j)$ for $j \in 1, 2, \ldots, M$. These features are manually located in all N images. The f parameter is initialized as the ratio of the image circle radius in pixels over π, and rotation angles r_{y_2}, r_{y_3} are initialized to $\frac{2\pi(i-1)}{N}$. Other parameters are initialized to 0. To increase stability of the f estimation, we also manually locate one epipole e_i^k in each image and minimize the distance of their projection in the world to the x-z plane. Thus, we minimize the following sum using the Levenberg-Marquardt method [12]:

$$\sum_{j=1}^{M}\sum_{i=1}^{N} ||f_i(p_w^j) - (x_i^j, y_i^j)|| \quad + \quad \sum_{i=1, k\neq i}^{N} ||g(e_i^k)[y]|| \tag{6}$$

where $f_i()$ is a function which maps a point in the world to a pixel position in image i, and $g()$ is a function which takes a pixel and maps it to the world at $Z = 1$. See equations 4 and 5. At each iteration of the bundle adjustment, the location of the features in the world p_w^j are estimated in Euclidean space by triangulation using the updated camera parameters:

$$p_w^j = \left(\sum_i \mathbf{I} - p_{c_i}^j p_{c_i}^{j^T}\right)^{-1} \left(\sum_i (I - p_{c_i}^j p_{c_i}^{j^T}) c_i\right) \tag{7}$$

where $\mathbf{I}$ is a 3×3 identity matrix, $p_{c_i}^j$ is the pixel feature p_i^j mapped to the camera i space, and c_i is the position of camera i.

4. Stitching the omnistereo view

We want to render an omnistereo pair of images for a dome or cylindrical screen centered at the 'x' shown in Fig. 4. Let α_i be the angles between two consecutive baselines. The external matrices (rotation and translation matrices) of this output image are set to be identity matrices. The output distortion coefficients k_1 and k_2 are also set to 0.

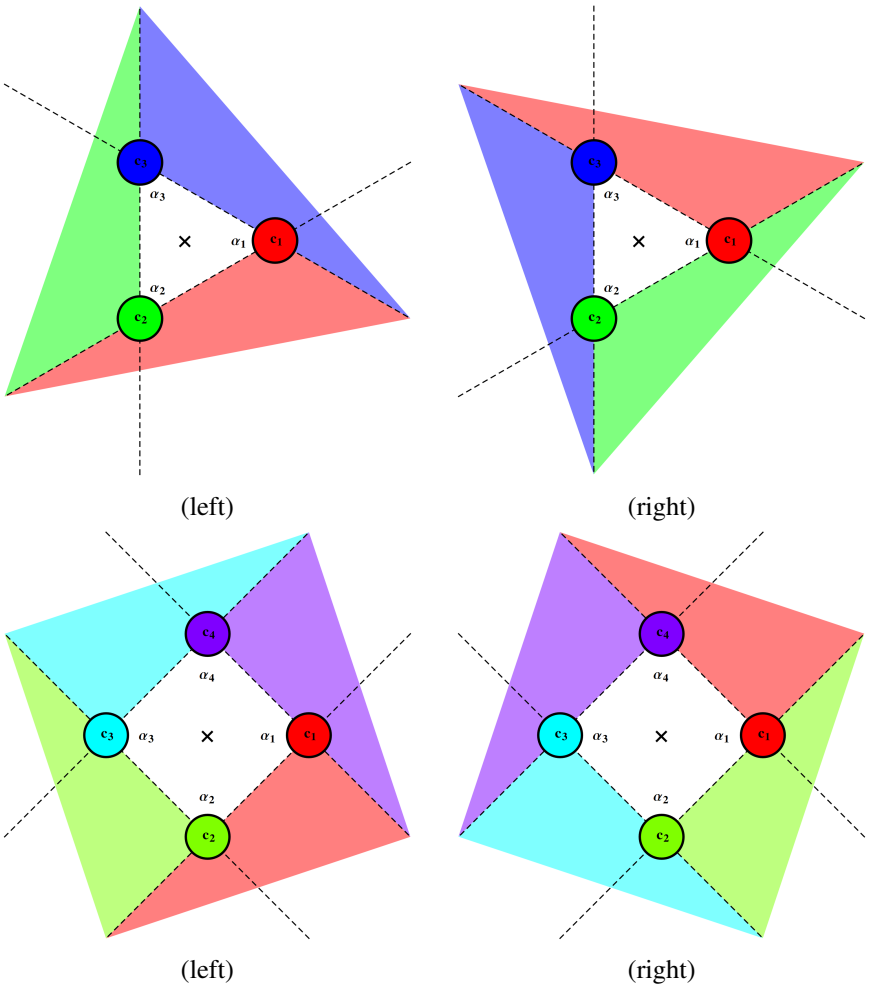


Figure 4. These figures show how left/right omnistereo images are assembled following the baselines (dashed lines), for an omnipolar setup using 3 cameras (top row) and 4 cameras (bottom row) viewed from above. The 360 degree fisheye images are drawn using colored circles labelled c_i. The field of view seen by a selected image region is indicated by the corresponding color. The 'x' represents the position of the ideal observer at the center of the screen, and the α_i are angles between two consecutive baselines.

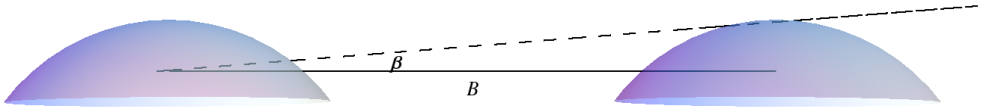


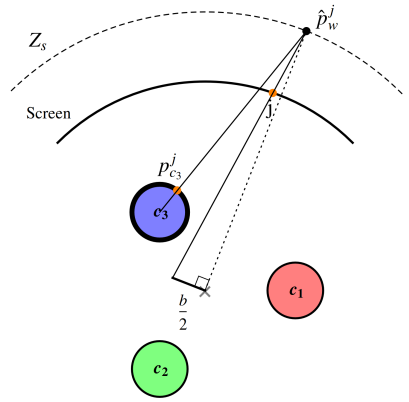
Figure 5. The vertical field of view free of lens occlusion is typically a bit smaller than 180 degrees. Here, angle β depends on the baseline B and the curvature of the lens.

We next consider the image for the left eye in an omnistereo dome. Assuming a fixed scene depth of Z_s, each pixel j is projected to a sphere of radius Z_s to get an estimated world point position $\hat{p}_w^j$ (see Fig. 6(a)). This projection depends on Z_s, the screen radius and the eye separation b which we fix to 6.5cm as this is the average human eye separation [9]. We will discuss in Sec. 4.1 possible values of Z_s. Consider ω_i^j, the horizontal angular position of point $\hat{p}_w^j$ in camera i, given by :

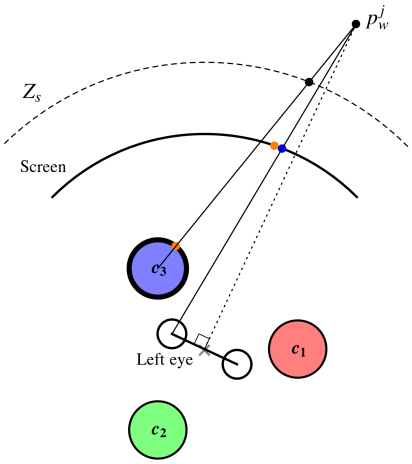
$$\omega_i^j = \arctan(\hat{p}_w^j[z] - c_i[z], \hat{p}_w^j[x] - c_i[x]). \quad (8)$$

The pixel sample of point $\hat{p}_w^j$ reprojected in camera i is used only if $\omega_i^j$is within$[\gamma_{i-1}, \gamma_i]$ where γ_i are angles defined as:

$$\gamma_i = \gamma_{i-1} + \pi - \alpha_i \quad (9)$$



(a)



(b)

Figure 6. (a) The process to render on the screen the left image of the omnistereo pair: pixel j is projected at depth Z_s to get $\hat{p}_w^j$ which is seen by the selected camera c_3 at pixel $p_{c_3}^j$. Thus, the image color at $p_{c_3}^j$ is copied on the screen at pixel j. (b) When a world point p_w^j is not located at depth Z_s, distortions occur because the rendered position on the screen (orange point) does not correspond to the position it should be seen at when an observer at the center is looking straight at it (blue point).

with $\gamma_0 = 0$, corresponding to the direction of the line joining c_1 and c_N. To render the image for the right eye, an offset of π is added to the γ_i values.

Thus, an omnistereo pair of images is produced covering the full 360 degree field of view horizontally. Note that the disparities can be adjusted by rotating the right image about its center so that zero image disparities on the screen roughly correspond, for instance, to the distance of interest in the scene. Although real-time stitching was not implemented for this paper, it can be achieved by calibrating the cameras beforehand and computing stitching maps in pixel shaders. The output resolution roughly correspond to the original resolution of input images, but the total number of pixels can be increased by zooming in so that the unused

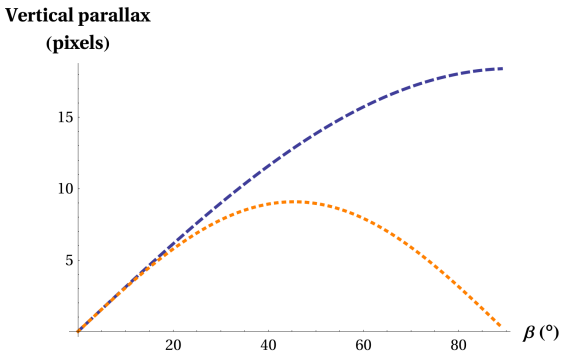


Figure 7. Maximum vertical misalignment at the seams w.r.t. angular elevation for points on a dome (blue) and a vertical wall (orange). See text for details.

image parts go out of the top and bottom field of view while the used parts remain visible (recall Fig. 3).

The vertical field of view free of self-occlusion is nearly 180 degrees. The amount of occlusion depends on the baseline and lens curvature (see Fig. 5). For highly curved lenses, the baseline can be slightly increased to get a wider vertical field of view without self-occlusions. Note that for an odd number of cameras, these occlusions create non-adjacent monocular regions as they are not located in the same image regions in the left and right omnistereo images.

4.1. Adjusting vertical misalignments

Each image in an omnistereo pair has a number of monocular seams equal to the number of cameras. By using vertical stitching planes passing by the epipoles, there is no horizontal misalignments at the seams induced by parallax, regardless of the scene depths. However, there is *vertical* misalignments for points outside the x-z plane. The magnitude of these vertical misalignments depends on the range of depths and the angular elevation w.r.t. the x-z plane.

One can choose a single depth for which no vertical parallax is produced, namely points at depth Z_s. It is therefore important to choose Z_s carefully. For instance, one could set Z_s to be the distance of the ceiling above the cameras, as this is where vertical parallax is worse.

We next quantify vertical parallax by considering $Z_s = \infty$ at different angular elevations β. In particular, we consider two sets of scene points, namely points on a dome:

$$p_w^{\text{dome}}(\beta) = c_1 + \mathbf{R}_y(30°)\mathbf{R}_x(\beta)(-Z_r, 0, 0)^T$$

and points on a wall:

$$p_w^{\text{wall}}(\beta) = c_1 + \mathbf{R}_y(30°)(-Z_r, \tan(\beta)Z_r, 0)^T$$

where $30°$ correspond to the orientation of the seam between cameras c_1 and c_2, and Z_r is the radius of the dome

or the wall distance. We consider $Z_r = 230$cm as this corresponds roughly to the radius of our cylindrical screen and that no scene point should ideally be closer to avoid window violations [13]. Vertical parallax is then computed by taking the difference between projections w.r.t. to cameras c_1 and c_2 on a dome having a 2048×2048 pixel resolution:

$$\text{Parallax}(p_w^{\text{dome,wall}}) = ||f_1(p_w^{\text{dome,wall}}) - f_2(p_w^{\text{dome,wall}})||.$$

Vertical parallax in pixels is shown in Fig. 7. It is 0 for points in the x-z plane and increases to about 20 pixels at the top of the dome (blue curve), and to about 9 pixels for points on a vertical wall at a $45°$ elevation (orange curve). Thus, although vertical is an issue, this experiment shows that it is typically small and varies with the kind of scene observed.

4.2. Perceptual stereo distortions

In this section, we are interested by stereo distortions at the center of the visual system for points in the x-z plane. For any scene point p_w^j, we assume that an observer located at the center of a dome or cylindrical screen is looking straight at it. We measure where this point is distorted as follows. The location of the eyes is given by $\mathbf{R}_y(\alpha)(\pm\frac{b}{2}, 0, 0)^T$, where α is the orientation of the eye baseline given by $\alpha = \arctan(p_w^j[x], p_w^j[z])$. The left/right rendering positions on the screen are computed by considering $\hat{p}_w^j$ at depth Z_s instead of its true position at p_w^j (see Fig. 6). The distorted position of a point in the world is triangulated using the rays joining the eyes and the left/right rendering positions.

Fig. 9 shows how points are distorted at different depths, namely 100cm, 200cm, 400cm and 800cm. Distortions are shown when using 3, 4 and 5 cameras, for $Z_s = \infty$ in the left column and $Z_s = 400$cm on the right. Observe that using an odd number of cameras distorts a circle of points into $2N$ straight lines or arcs, while using an even number of cameras leads to a distortion into N straight lines or arcs. This can be explained by looking at figure 4. Say that an observer located at the center of screen is looking in the direction of camera c_4. The stereo view is captured by cameras c_1 and c_3 as they correspond exactly to the positions of the eyes. In fact, the camera pairs (c_1, c_3) and (c_2, c_4) each give two stereos views with no stereo distortion at their center.

When using an odd number of cameras, there is never a pair of cameras that correspond exactly to the eye positions. There is no distortion for points at Z_s, whereas points behind Z_s appear slightly closer in depth, and points in front of Z_s appear further away in depth. However, the number of straight lines or arcs is increased for a better circle approximation. Overall stereo distortions are reduced considerably when using 5 cameras[2].

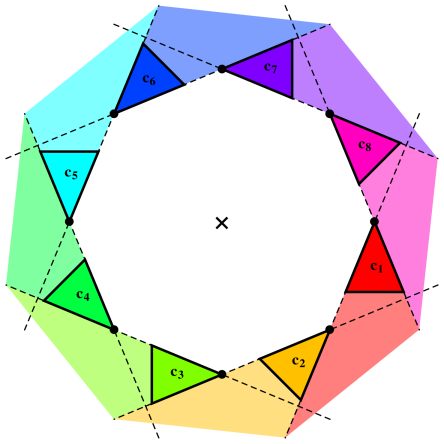


Figure 8. An omnipolar setup for monocular capture using 8 cameras. Each camera is represented using a small disk (center of projection) and a colored triangle labelled c_i. The respective field of views are indicated using dashed lines and corresponding colors.

On a related note, there is perceptual singularity for scene points on the y-axis because the observer can look directly upwards with an arbitrary eye orientation. A dome can provide at its zenith only one scene depth and it has to be at screen distance (i.e. with zero disparity).

4.3. Application to monocular panoramas

The stitching method using epipoles of neighboring cameras could also be used in the context of monocular camera systems to reduce misalignments at seams due to parallax.

For instance, a camera system such as the Ladybug [17] typically use a few cameras to cover a full 360 degree field of view with high resolution. Unfortunately, not all cameras can be located exactly at the same position. Thus, there is parallax from one image to its neighboring image. This creates misalignments at the seams. These misalignments can be observed in systems such as Google Street View.

Horizontal parallax could be eliminated by orienting cameras instead of outwards (see Fig. 8). Note that the realization of such a setup would have to avoid that a camera occludes another, as cameras do not have a triangular shape as shown in Fig. 8. Also note that there is no constraint on the radius of the circle on which lies the cameras. This makes practical the use of large camera equipment.

5. Results

We present two scenes, namely `Lab` and `Driveway`. Both scenes were captured using an omnipolar setup using 3 cameras with fisheye lens lying on a circle with a diameter

[2]Note that we do not assess in this paper whether these distortions are perceived or not by the human visual system.

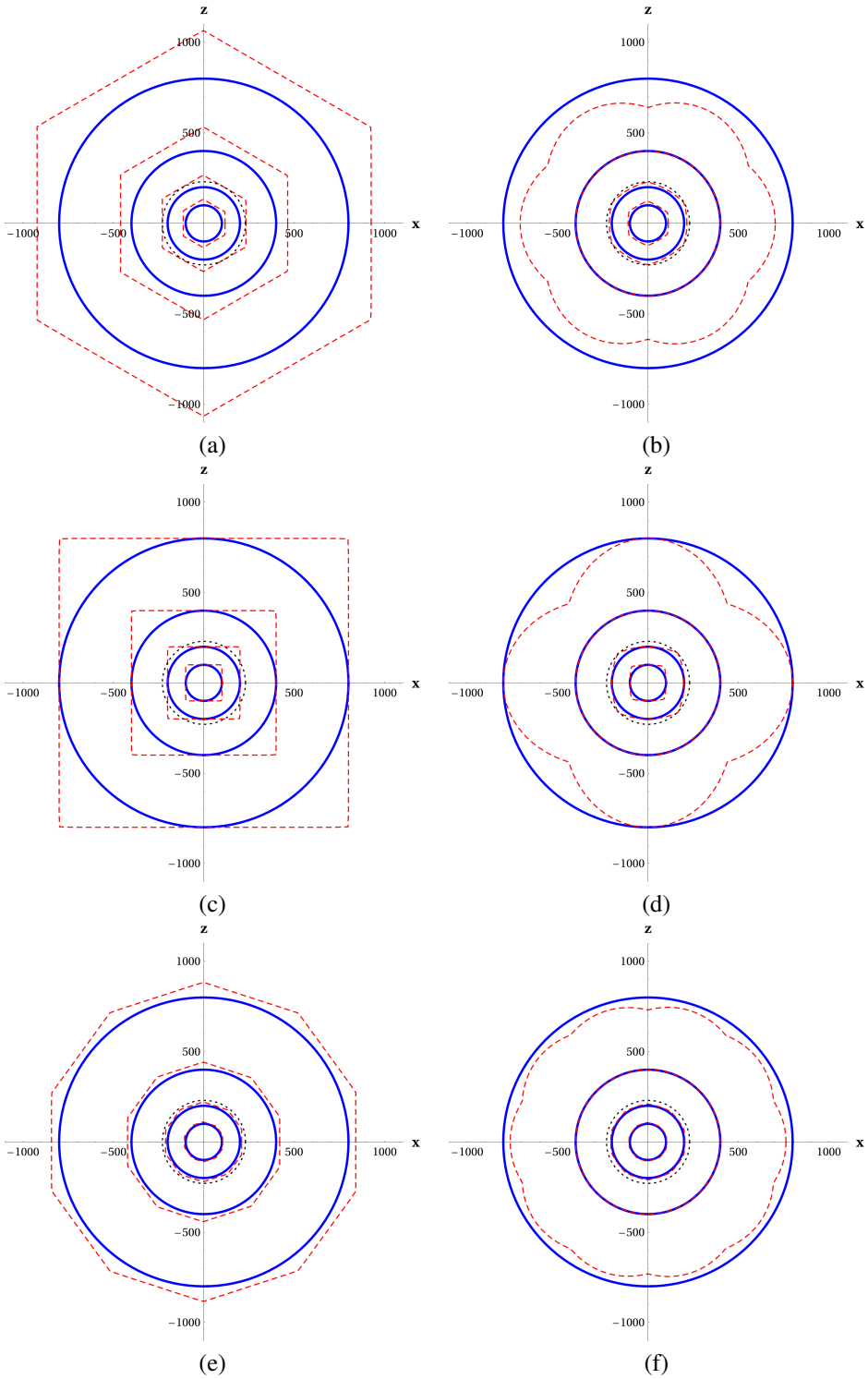


Figure 9. Depth distortions at the center of the visual field for points in the x-z plane at eye level when using an omnipolar setup using (a-b) 3 cameras (c-d) 4 cameras (e-f) 5 cameras. The true position of the points are shown in blue. They lie on 4 circles with the radius ranging from 100cm to 800cm. The distorted positions are shown in dotted red. Vertical misalignments were adjusted for (left) points at $Z_s = \infty$ (right) points at $Z_s = 400$cm. Notice that, on the right, there is no distortion for points on the circle of radius 400cm because it is equal to Z_s. The radius of the screen we use is indicated by a black dotted circle.

of about 12cm. The camera setup was fixed on a rail on the ground to produce a travelling of about 1.5m while a person is walking by. See online videos at [1].

For the `Lab` scene, we tested a camera setup composed of 3 Prosilica 1380 cameras (1360×1024 pixel resolution), each with a Fujinon C-mount fisheye lens. Vertical parallax was adjusted so that the ceiling above the cameras was correctly aligned ($Z_s \approx 2.4$m). See Fig. 10(a,b). Omnistereo field of views were cut so that lenses cannot be seen.

The `Driveway` scene was shot with 3 Canon HFS11 cameras (1920×1090 pixel resolution), each with an Opteka Vortex fisheye lens. Camera clocks were synchronized using the LANC protocol. Vertical parallax was adjusted to align the electrical wires above the cameras ($Z_s \approx$ 6m). See Fig. 10(c,d). Because of the high curvature of the lenses, large occlusions can be visible in the produced omnistereo images unless the field of view is cut significantly.

6. Conclusion

This paper introduced a camera setup for capturing omnistereo videos using a minimum of 3 cameras with fisheyes with no moving parts. Its field of view covers a dome in stereo. The baseline between pairs of cameras define epipoles which are used for stitching without any horizontal misalignments due to parallax. Applications range from omnistereo cinema to providing stereo immersion for systems like Google Street View. Future work could involve recovering scene depth from these omnistereo images.

References

[1] `http://vision3d.iro.umontreal.ca/en/projects/omnistereo/`.

[2] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google Street View: Capturing the world at street level. *Computer*, 43(6):32–38, June 2010.

[3] P. Bourke. Synthetic stereoscopic panoramic images. *Lecture Notes in Computer Science (VSMM 2006)*, 4270:147–155, 2006.

[4] P. Bourke. Omni-directional stereoscopic fisheye images for immersive hemispherical dome environments. *Computer Games and Allied Technology*, pages 136–143, May 2009.

[5] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In *International Conference on Intelligent Robots and Systems*, pages 1683–1688. IEEE, October 2007.

[6] V. Couture, M. S. Langer, and S. Roy. Panoramic stereo video textures. *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011.

[7] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *ACM Conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 135–142, New York, NY, USA, 1993.

[8] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–72, 1992.

[9] I. P. Howard and B. J. Rogers. *Seeing in Depth*. Oxford University Press, USA, 2002.

[10] 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.

[11] H. Ishiguro, M. Yamamoto, and S. Tsuji. Omnidirectional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):257–262, 1992.

[12] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics (SIAP)*, 11:431–441, 1963.

[13] B. Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009.

[14] T. Naemura, M. Kaneko, and H. Harashima. Multiuser immersive stereo. *IEEE International Conference on Image Processing*, 1:903, 1998.

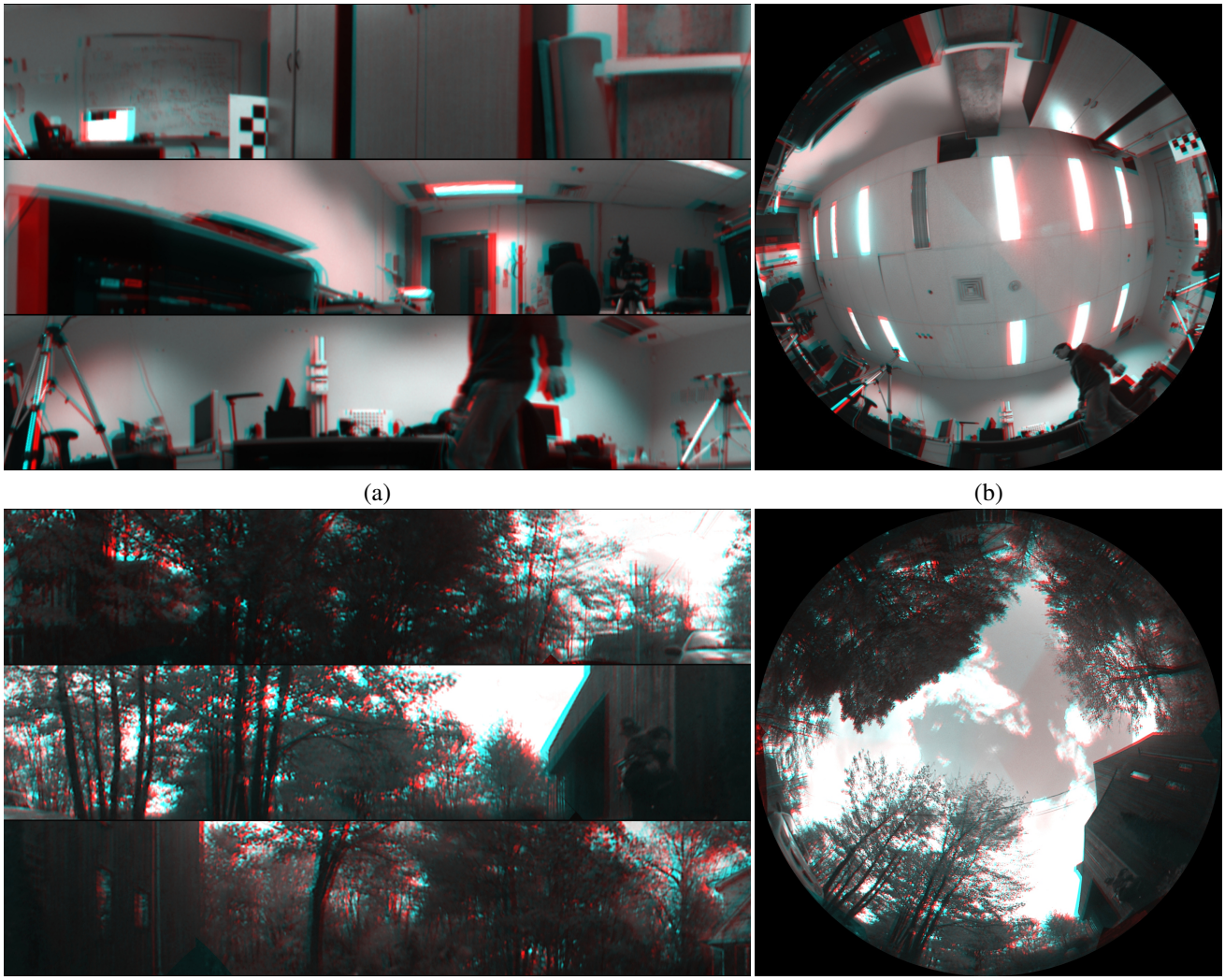
[15] S. K. Nayar. Catadioptric omnidirectional camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–, Washington, DC, USA, 1997. IEEE Computer Society.

[16] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, 2001.

[17] Point Grey Research. *Ladybug3*, 2008.

[18] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *IEEE International Conference on Computer Vision Systems*, ICVS '06, pages 45–, Washington, DC, USA, 2006. IEEE Computer Society.

[19] K. Tanaka and S. Tachi. Tornado: Omnistereo video imaging with rotating optics. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):614–625, Nov. 2005.



(a) (b)

(c) (d)

Figure 10. Resulting omnistereo images in anaglyph format for the `Lab` scene (top) and the `Driveway` scene (bottom). Full 360 degree cylindrical images are shown on the left in three 120 degree parts (these parts do not correspond to the 120 degree field of view used in each camera for the left or right eye). Images formatted for a dome are shown on the right.