Capturing Non-Periodic Omnistereo Motion

Vincent C.-Couture
Université de Montréal
chapdelv@iro.umontreal.ca

Michael S. Langer
McGill University
langer@cim.mcgill.ca

Sébastien Roy
Université de Montréal
roys@iro.umontreal.ca

***Abstract*— An omnistereo pair of images enables depth perception all around the observer. Because omnistereo lenses or mirrors do not yet exist, capturing an omnistereo video would require using several stereo cameras at different baseline orientations. This paper presents a multi-take capture method for creating high-resolution omnistereo videos at an affordable cost: only two standard cameras and a tripod are required. The method combines multiple stereo *takes*, where each take follows some action within the field of view, an actor for instance. Internal camera parameters are first calibrated by capturing a static omnistereo background. Each frame of each take is then pasted on the background at a position estimated by feature tracking. The paper provides a detailed description of the steps of the method. It also provides an analysis of parallax which results from the off-axis camera rotation. Parameters for the method are chosen to minimize artifacts that arise from this parallax. The method is intended to be applied in a relatively controlled space like a movie set, where the action in each take is within the field of view of the cameras.**

I. Introduction

Similarly to the two eyes of the human visual system, a stereo pair of cameras captures two images of a scene from slightly different viewpoints. When these two images are displayed and viewed in stereo, they are fused by the brain of the observer, and the disparities between corresponding points provide important clues for scene depths.

In immersive environments such as in 360^o cylindrical displays, an observer can turn his gaze in any orientation. Moreover, there can be more than one observer and multiple observers can look in different directions at the same time. This is problematic since two traditional cameras cannot capture images for all observer orientations, in particular, because no stereo information is available near directions that are parallel to the baseline.

In contrast to a traditional stereo image, an omnistereo pair of images [9, 12, 8, 14, 13] maximizes stereo perception for all orientations at the same time by gathering light such that there exists stereo information for all orientations. Using extensions of multi-camera systems such as the Ladybug [15] which has five cameras to cover 360°, an omnistereo capture could be used but it would become relatively expensive and data intensive as the number of cameras would double, e.g. from five to ten. In [14], concept ideas of omnistereo Fresnel-like lenses and spiral mirrors were presented, but problems such color aberrations remain to be solved.

Another approach to capture omnistereo images is to gather several small field of view images called *slits* from one or two cameras rotating off-axis, and then mosaic all images together (see Fig. 2). To avoid parallax artifacts which are due to the fact that cameras are rotated off-axis and hence contain a translation component, each slit-image should cover no more than one or two degrees, or approximately 200-400 slits, as done in [14]. This mosaicing method was then extended in [16] to vary the slit position in larger images to produce videos of periodic types of motions, such as waterfalls, and slit shapes to avoid cutting across non-periodic types of motions such as people.

In this paper, we present a method that extends the mosaicing approach of [14] to allow the capture of non-periodic motions. The method is not meant to be an omnistereo capture for arbitrary motion scenes, such as an uncontrolled public space. Rather, it was designed to be used in a controlled environment, such as a movie set.

Our method has two steps. First, a static omnistereo background is captured by mosaicing slit-images taken by a rotating stereo camera, which is auto-calibrated in the process. Second, keeping the tripod in place but allowing rotation of the stereo rig, several non-periodic motions, like a moving actor for instance, are captured within the field of view of the stereo camera. Camera motion for each of these *takes* is then estimated by feature tracking and each frame is pasted on top of the background.

The main technical contribution of the paper is a detailed parallax analysis that aims to minimize parallax artifacts in the second step, namely to minimize parallax between the left and right edges of the foreground frame and the corresponding region of the background mosaic when pasting each foreground frame over the background.

A layout of the paper is as follows. A summary of previous mosaicing methods is first presented in Sec. II. The capture setup as well as an analysis of motion parallax issues are then detailed in Sec. III. Section IV presents the omnistereo static background capture method as well as camera calibration. Sec. V then gives a detailed explanation of the process to add multiple takes of non-periodic motion. Results are presented in Sec. VI, followed by a conclusion in Sec. VII.

II. Previous Work

We review in this section existing mosaicing methods that allow the capture of static scenes, periodic types of motions, and with lesser success non-periodic types of motions.

For the simpler case of static scenes, [9, 14] introduced slit-images having a very small horizontal field of view of about 1^o. The method captures light at many orientations such as to cover a complete turnaround, i.e. 360 degrees (see Fig. 2(b)). In practice, a column of pixels is considered in images taken

Fig. 1. Part of a frame of an omnistereo video, shown in red/cyan anaglyph format.

by standard cameras. An omnistereo pair of images is then formed by mosaicing the slit-images of each camera. In the extreme case where a slit-image is the width of a single light ray, parallax artifacts from camera motion which are due to the off-axis camera rotation are non-existent. An omnistereo capture method using a single camera was also described in [13] by considering two slits in a single image, both slits at the same distance from the image center but one towards the left edge of the image and one towards the right. These correspond to two virtual cameras following a circle. The direction of sight of each virtual camera is a line tangent to this circle, but the directions for the two cameras are opposite. Thus, ignoring occlusions, any point in the scene is captured by two points of view, but at a different time.

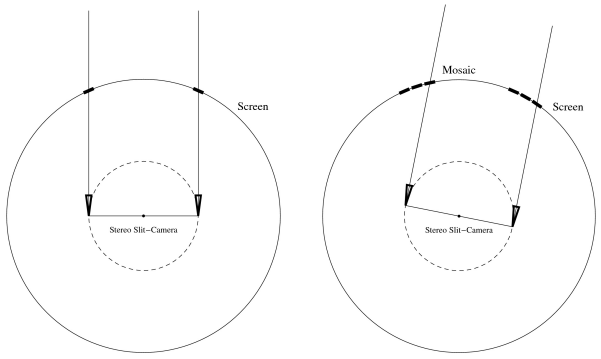


Fig. 2. (a) Previous work on omnistereo images uses a rotating stereo pair of slit-cameras. Here the cameras are parallel. (b) An omnistereo pair of images combines light from several projection centers constrained to lie on a circle in an attempt to allow depth perception all around of an observer. In practice, image slits are stitched together in a mosaicing process to fully cover 360^o.

A related slit image method was presented in [16] for the capture of (monocular) dynamic scenes. Called the dynamosaicing method, this approach varies the slit position with time to create a video of periodic types of motions, such as a waterfall. For non-periodic types of motions, such as walking actor, a 4D graph-cut minimization [2] computes slit positions and shapes to produce a spatially and temporally continuous video. The method attempts to avoid motion at the edge of the slit. This method requires enormous amounts of resources, as the graph to minimize the whole sequence at once is very large.

Another monocular approach was presented in [1] where a graph-cut minimization allows to render, after a manual selection of dynamic and static regions, a continuous video from a fixed viewpoint, but for periodic motions only (*e.g.* small waves, flag in the wind). Parallax artifacts are not addressed in this method because the camera is assumed to perform a pure rotation.

In this paper, a tripod with two cameras rotating off-axis is used. First, a 360° omnistereo background mosaic is captured using slit-images of a static scene. Foreground non-periodic stereo action takes are then captured. These takes can follow action in any direction as long the action is restricted to a safe region within the field of view of the camera in each frame. The foreground action is finally pasted over the background. To ensure there is no seam between the foreground and background, parallax must be avoided. The next section explains how we have achieved this.

III. Camera Setup and Parallax Analysis

Two cameras are setup on a tripod and rotated around a single axis. The distance between the two camera lens centers should be about 6.5 cm, similar to the distance between human eyes. Each camera follows a circular motion that contains both translation and rotation components. If scene points have different depths, each camera then captures an image sequence with motion parallax. In this section, we give a more general definition of *parallax* and we show that this parallax can be reduced under certain conditions.

Consider a pixel p at various horizontal positions in a reference camera. The reference camera is defined to be located at 0^o (see Fig. 3). Pixel p is then projected out to distances z_{min} and infinity to get scene points $P_{z_{min}}$ and P_∞. We use $z_{min} = 2$m. Both points $P_{z_{min}}$ and P_∞ are then reprojected at various other camera orientations θ ranging from $-FOV$ to FOV, where FOV is the field of view of the camera. *Parallax* of p at orientation θ is then defined as the image distance in pixels between these two reprojected points. Note that parallax depends on both p and on θ. In the subsequent plots, we only consider cases for which the reprojected points are both within the image boundaries.

By construction, there is no parallax when $\theta = 0$ since this is the location of the reference camera. Fig. 3 shows that there is also zero parallax at another orientation, namely $\theta = \theta_0$, that can be derived from the following two relations

$$\gamma = \alpha_r + \beta \tag{1}$$

and

$$\pi = \alpha_s + \beta + \gamma \tag{2}$$

where α_r is the angular orientation of p with respect to the optical axis of the reference camera at $\theta = 0^o$. Plugging (1)

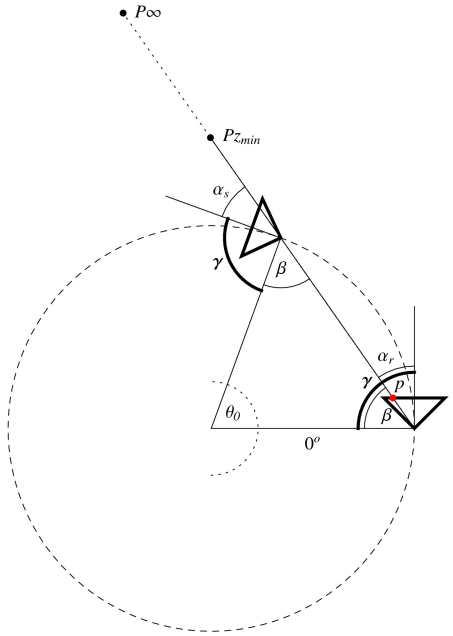


Fig. 3. Parallax is measured by projecting a pixel p at Z_{min} and infinity to get $P_{Z_{min}}$ and P_∞, and reprojecting these 2 points at various camera orientations, here ranging from -90^o to 90^o for a 90^o field of view. The figure shows that there is no parallax at $\theta = 0^o$ and at $\theta = \theta_0$, when the camera position lies on the line passing by p, $P_{Z_{min}}$ and P_∞. See Eq. 3 for a derivation of θ_0. Note that $P_{Z_{min}}$ and P_∞ are not to scale.

into (2) yields:

$$\begin{aligned}
\pi = \alpha_s + \beta + \alpha_r + \beta &\Rightarrow \pi - 2\beta = \alpha_r + \alpha_s \\
&\Rightarrow \theta_o = \alpha_r + \alpha_s \\
&\Rightarrow \theta_o = \alpha_r + \pi - \beta - \gamma \quad \text{using (2)} \\
&\Rightarrow \theta_o = 2\alpha_r + \pi - 2\gamma \quad \text{using (1)} \\
&\Rightarrow \theta_o = 2\alpha_r + \nu \qquad (3)
\end{aligned}$$

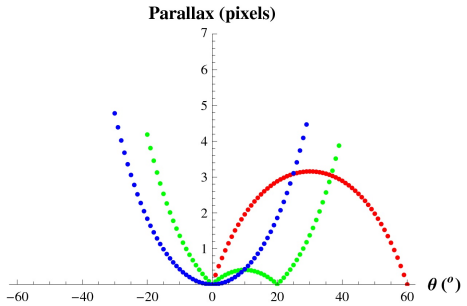
where ν is the stereo vergence, defined to be the angle between the optical axes of the two cameras in the stereo rig.

Fig. 4(a) shows the absolute value of parallax for various positions of pixel p when the stereo vergence ν is 0^o. Observe that the three curves reach 0-parallax at $\theta = 0^o$ and at $\theta_o = 2\alpha_r$, following Eq. 3. For example, if p is located at the center of the image, such as for the blue curve, then 0-parallax is reached only there. Another special case is the red curve, namely when p is located at the left (or right) image border *i.e.* $\alpha_r = \frac{FOV}{2}$. In this case, $\theta_0 = FOV$. They leads to the following.

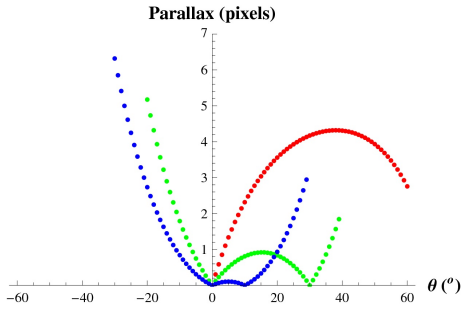
Key Observation: *If there is 0-parallax when entering the field of view, then there is also 0-parallax when leaving the field of view.*

This observation will be used in the background mosaicing step (Sec. IV-B) to choose, by a weighting function, the location of the slit such that when we paste the foreground over the background mosaic, parallax is zero between the edge of the foreground frame and the corresponding region of the background mosaic.

Note that if vergence ν is non-zero, then the key observation no longer holds, namely although there is 0-parallax at the left border, 0-parallax is reached at ν degrees away from the right border rather than exactly at the right border (see red curve in Fig. 4(b)).



(a) camera vergence of 0^o



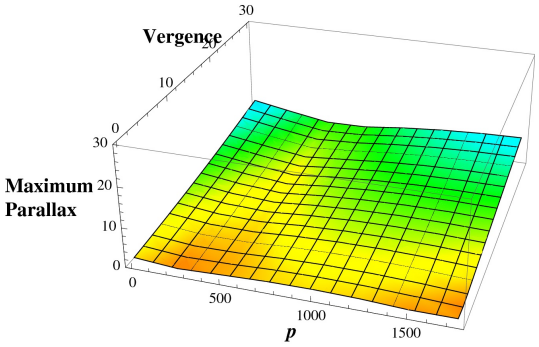
(b) camera vergence of 10^o

Fig. 4. For the right HD camera with 60^o field of view and a vergence of (a) 0^o (b) 10^o. Parallax is shown at angles $\alpha_r = 0^o$ (blue) which is the center of the image, $\alpha_r = 10^o$ (green) and $\alpha_r = 30^o$ (red) which is the border of the image. These correspond to pixels $p = 960$ (blue), $p = 650$ (green) and $p = 0$ (red).

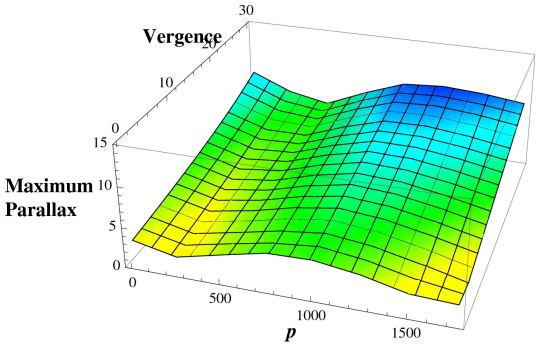
Parallax is not only an issue at the edge of the foreground frame, but it is also an issue in the interior when the location of the foreground frame is moving, *i.e.* when the camera is panning or tracking during action take. In the case of the red curve in Fig. 4(a), parallax is maximal near center of the image. This parallax causes static objects in the scene to move slightly from one frame to the next. One might ask whether this interior parallax could be reduced by using a non-zero vergence ν. The answer is no, however. As shown in Fig. 4(b) for a specific case $\nu = 10°$, all three curves have a maximum parallax that is greater than in Fig. 4(a). More generally, Fig. 5 shows the maximum parallax as a function the location of pixel p, the camera vergence ν and three different fields of view. These show that maximum parallax generally increases with vergence and field of view.

From the above observations, we conclude that a stereo vergence angle of 0^o is desirable and so this is what we use in the remainder of this paper. The choice of field of view size remains, and here we face a trade-off. A larger field of view allows one to capture wider actions, but it also introduces more parallax in the middle of the foreground frames when panning the stereo camera. A wide field of view also reduces the horizontal omnistereo resolution. In our results, we used

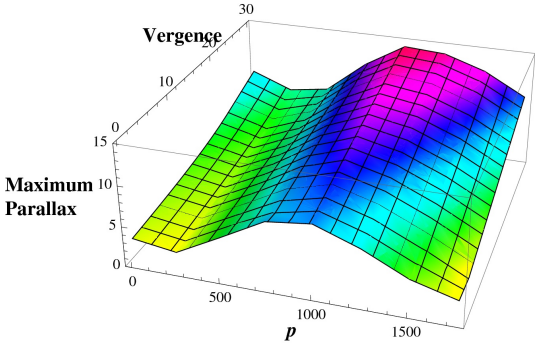
a field of view of about 55^o (see Sec. VI).



(a) 30^o field of view



(b) 60^o field of view



(c) 90^o field of view

Fig. 5. Maximum parallax for a varying vergence ν and pixel position p. Plots are shown for cameras having field of views of (a) 30^o (b) 60^o (c) 90^o. Note that the camera resolution is the same in all cases, namely 1920×1080.

IV. Omnistereo Mosaicing of Static Background

The first step of the method is to capture a static omnistereo background by rotating two cameras on a tripod. From the captured images, both cameras need to be calibrated as precisely as possible. This calibration is done as part of the mosaicing (hence, autocalibration) for convenience. Images for each camera can then be stitched using standard panoramic mosaicing techniques [4].

A. Camera Autocalibration

The cameras are calibrated separately. Each camera is assumed to be rotating around the y-axis. Let the image sequence be $I_{\theta_1}, I_{\theta_2}, ..., I_{\theta_N}$, where $\theta_1 = 0$ and $\theta_N = 2\pi$, and where

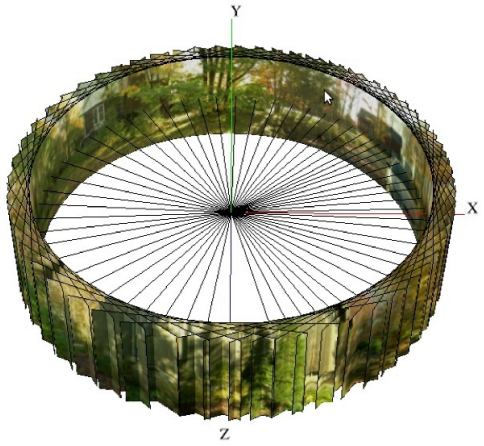


Fig. 6. The autocalibration process estimates the rotation angle between consecutive frames and, for each camera of a stereoscopic pair, the focal length of the camera and radial distortion. Motion parallax artifacts are ignored, as if the projection center of both cameras was centered on the rotation axis.

each image is of size $W \times H$ pixels. We use N = 60, and $\theta_{i+1} - \theta_i \approx \frac{2\pi}{N}$ radians. Internal camera parameters $\mathbf{K}$ are assumed to be constant over all images, giving:

$$p_i = \mathbf{K}\ \mathbf{R_0}\ \mathbf{T}\ \mathbf{R}_y(\theta_i) P \tag{4}$$

where $p_i = (x, y, 1)$ is a pixel in the image i, $P = (X, Y, Z)$ is a point in the world, $\mathbf{R}_y(\theta_i)$ is a rotation matrix around the y-axis, $\mathbf{T}$ is the camera translation matrix that offsets the two cameras from the origin by distance b in the x, z-plane, and $\mathbf{R}_0$ is a rotation that allows the optical axis of the cameras to have an arbitrary orientation.

Hence, the total number of parameters is $N + 3$: the focal length f, the three rotations of $\mathbf{R}_0$ and the $N - 1$ rotation angles $\theta_{i,i+1}$ between consecutive images.

As a first approximation, camera parameters were estimated by ignoring camera translation, *i.e.* setting $\mathbf{T}$ to be the identity matrix, and using standard techniques, namely robustly matching SIFT features between frames using a homography model and RANSAC, followed by an initial estimate of the camera parameters and then bundle adjustment, taking radial distortions into account [7]. The estimates can be improved by allowing for non-zero translation, namely performing another bundle adjustment that triangulates features in 3D [19].

B. Mosaic Blending

Once the cameras have been calibrated, a mosaic is rendered by projecting each image onto a cylindrical surface. To hide vignetting and calibration errors caused by motion parallax, the images are blended using the composition method of [3]. For a pixel of the mosaic, the method selects high frequency components from the image corresponding to a maximum weighting function, and adds the average of low frequencies of all images. The observation in Sec. III that parallax is zero at the image boundary leads us to use a weight function higher near the background image borders instead of the center (see Fig. 7). This removes parallax differences between the

(a) (b)

Fig. 7. The blending weight function for the (a) left camera (b) right camera. The function puts more weight near the right or left border instead of the center to minimize the parallax difference between the background and foreground action frames.

background and a foreground frame near the borders of the latter (see Sec. V).

C. Adjusting Omnistereo Disparity

It is usually preferable that the zero image disparities roughly occur at the distance of the projection screen. To ensure this, the right omnistereo background is shifted to the right until objects located at this distance have zero disparity. A vertical shift is also sometimes necessary to make objects align vertically.

V. Adding Omnistereo Foreground Motion

The previous section captured a static omnistereo pair of images, using two cameras rotating around a single axis on a tripod. By keeping this setup fixed, the calibration of Sec. IV can be used to warp new images on the omnistereo image space, using the same projection onto a cylinder surface as in Sec. IV-B. In this section, foreground action is captured in stereo and pasted over the static omnistereo background.

A. Stereo Capture Constraints

Foreground action is captured by rotating the stereo rig. The only constraint is that the action cannot go out of the left/right frame borders. For an HD capture of 1920×1080 pixels, this constraint is ensured by having static left and right margins. These margins need to be at least as large as the maximum pixel motion between consecutive frames. In this work, margins between 30 and 60 pixels are used. The image margins will be very important in the rendering process described in Sec. V-C. Fig. 8 shows an example, with the static left/right margins indicated in red.

It is essential that motion blur caused by camera rotation be avoided. Both cameras should be set in manual mode at a small exposure time $\frac{1}{125}$ or $\frac{1}{250}$, depending on how fast the camera will be rotated. For lower-end cameras for which there is no manual exposure mode, the *Sports* mode should be used. We also recommend to use manual focus to avoid any instabilities by the autofocus mode. Gain should be set at a minimum. Aperture can be set in manual or automatic mode (see Sec. V-C, for brightness and contrast adjustments).

B. Tracking Camera Orientation

The next step is to estimate angular orientation of each foreground action frame relative to each frame on the background. Note that the tracking operation should give identical

Fig. 8. Each foreground action frame is constrained to have static left and right margins (indicated in red).

Fig. 9. Motion is rendered by superimposing the current frame on all previous frames of the sequence in order. This rendering trick retains changes such as the car no longer parked in the driveway. Images are shown for the left camera only.

results for corresponding left and right images. Assuming that a close enough estimation of the camera orientation is given for the first frame, the orientation of subsequent frames is estimated by tracking features on the background using the KLT method [18]. In practice, the position of the first frame can be obtained by testing all possible orientations with respect to the background and selecting the one with minimum SSD. A RANSAC method [5] makes the estimation process robust to outliers, if there is wind in branches for instance, or if there are features in the foreground action that are not in the background (or vice-versa). We assume that 75% of samples are inliers within a 0.25 pixel distance. At each frame, KLT features of the background are selected on a cropped background having the same width as the frame, and starting at the previous frame position.

The above orientation estimate is affected by parallax and is used only as a first approximation. The result is refined by using features from the two closest original background frames. Note that the orientation of only 60 background frames is calibrated. The orientation of intermediate background frames is estimated by frame registration using pixels near the center of the frame, much less affected by parallax. This can be seen by looking at the motion field for forward camera motion, where parallax is 0 at the image center and increases quadratically towards the image borders [11]. Accumulated drift is adjusted so that all the frames cover exactly the calibrated mosaic background.

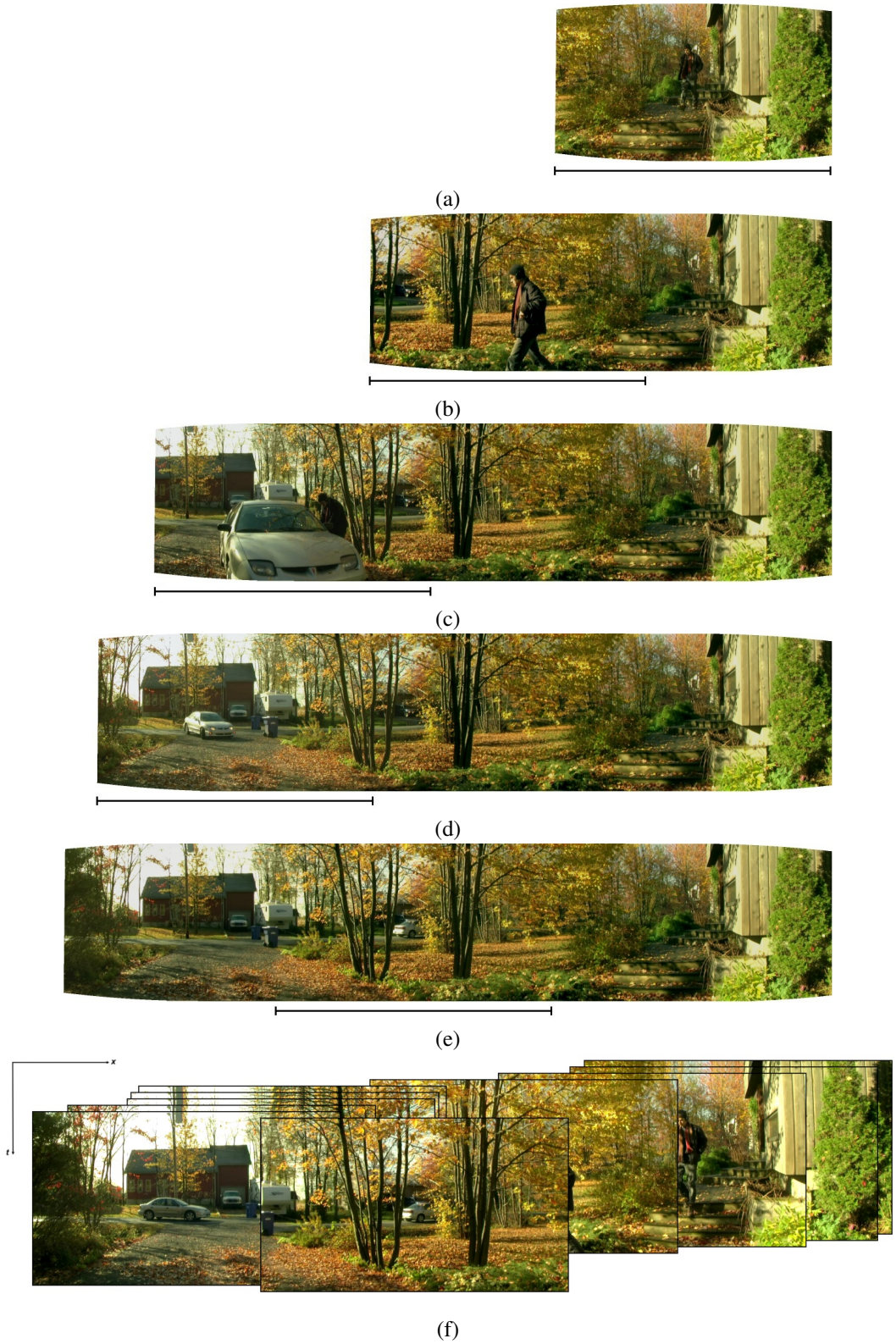


Fig. 10. From (a) to (e), current frame is pasted over the static background and a static part from the still visible image margins of the previous frames. The latter retains any changes made to the original static background (notice the car leaving the driveway). The location of the current frame is shown by a line underneath each image. Images are shown for the left camera only. In (f), the process of superimposing the frames from (a)-(e) is shown. Only a small subset of all the frames is shown.

C. Rendering an Action Take

Once the orientation of each foreground action frame has been estimated, the action is rendered on top of the background by superimposing the current foreground frame on all previous frames of the sequence in order (see Fig. 9). The reason we do this is as follows. If only the current foreground frame were pasted over the original background, then foreground objects that were not present in the background would disappear when going out of frame. For instance, if an actor were to put an object on the ground and walk away leaving that object behind and the camera were to track the actor, then the object would disappear when no longer in frame since it would not have been part of the original background. Similarly, if an actor were to take an object that was part of the background and then walk away with it, the object would reappear as the cameras follow the actor.

By assuming that the left/right static margins are at least as wide as the maximum displacement between consecutive frames, any change to the background is maintained as foreground frames write over it. See Fig. 10. For example, in (d), when the current frame follows the car leaving towards the right, the car does not reappear where it was parked on the background. This is because the foreground action frame has written over the background. Recall also from Sec. III that parallax is near zero at the left and right image boundary. Thus we expect minimal differences between the borders of the current frame and the background over which the current frame is written.

Note that the background is no longer composed of margin slits where the last foreground action frame is pasted. If a foreground frame of a subsequent action take overlaps this area, then there will be parallax between its margins and the background. A solution is to end the previous take by panning the stereo cameras by an angle greater than the FOV after the action has stopped (for instance, after the actor exits through a door or the moving object disappears behind an occluder). The left or right margin of these last foreground frames can then be pasted on the background, thus rendering an omnistereo background composed of margin slits only.

D. Blending Images

Although parallax is minimized, other differences may exist at the margin between the foreground and background, and we would like to hide these differences as much as possible. First, each frame is color calibrated as lighting conditions might change from the time the background was captured, *e.g.* if capture is performed outdoors and sunlight is changing, or if the cameras themselves are not in complete manual mode and somehow adjust differently. Color calibration may also be necessary just within the foreground layer. For instance, if an actor is followed by the cameras going from left to right, then right to left back where the take started, any change in lighting will give away the location of the current frame as it will be lighter or darker than the superimposed margins of the first frames.

From color samples at successfully tracked features, let μ_f and μ_b be the mean RGB intensity of the current frame and the background respectively, and let σ_f and σ_b be their color standard deviation. The intensity of a pixel in the current frame is adjusted for each color channel using the following equation, similar to [17]:

$$\hat{I} = \mu_b + \frac{\sigma_b}{\sigma_f}(I - \mu_f) \tag{5}$$

Vignetting, which makes images darker near the borders, is also another important phenomenon that has to be considered. Because vignetting makes static margins in a motion layer darker, the brighter image center of the current frame makes the camera motion obvious. In this work, vignetting was removed by cutting about 50 pixels from the left and right borders of each frame. However, in the case where this solution cannot be applied without cutting a moving object in frame, the vignetting correction function can be estimated during the autocalibration section of Sec. IV using the method described in [6] that was applied to mosaics. The method, however, assumes fixed aperture which is guaranteed only if the cameras are in manual mode and not in Shutter Priority mode as discussed in Sec. III. One can also use the approach described in [21], a method based on texture segmentation and analysis and requiring a single frame only.

Finally, another important cue about the location of the current frame is the small dynamic noise of the foreground video in contrast to the background which is static. Every pixel intensity $I(x, y)$ of the background is replaced by a normal function $N(I(x, y), \sigma)$, where noise deviation σ is estimated by shooting a static scene with the cameras being stationary and fitting a Gaussian to the intensity variations. The noise is updated for each frame of the final video.

VI. Results

The method presented in this paper was tested by shooting a four minute video. Two Canon HFS11 cameras were used on a fixed tripod which allowed camera rotation around an almost vertical axis only. The distance between the centers of both lens was about 6.5 cm, similarly to the eyes. To synchronize frame capture, as well as zoom, focus and color balance, both cameras were controlled through the LANC protocol (a LANC Shepherd was connected to the cameras by RA-V1 remote control adapters).

Both cameras were set to their widest field of view (about 55^o), at fixed manual focus and at an exposure time of $\frac{1}{250}$ seconds. The stereo rig was first rotated 360° for background capture. The rig was then fixed for a few seconds for noise analysis. Each action take was finally shot in succession following a planned scenario.

Both for the morning and the evening shoots, ambient lighting changed considerably, and a small wind often moved branches and leaves slightly. While this wind often slightly breaks the assumption that margins should be static, these conditions were good to test the robustness of both camera tracking and color calibration. To speed up the experiments,

Fig. 11. The setup used in this paper consisted of two cameras in parallel on a tripod. The distance between the projection center of both lens was about 6.5 cm. The cameras were synchronized using the LANC protocol.

the original content of 1920×1080 resolution was down-sampled to 960×540 pixels. The final video is high-resolution at about 6500×540 pixels.

Processing time was divided about evenly between tracking and rendering, about 1 second per frame for each operation on a 1.8GHz laptop with 2GB of RAM. However, this can be greatly improved, as the tracking operation for each take can be done in parallel, and the rendering operation which basically repeatedly pastes an image on a background could potentially be executed on the GPU.

The video was also shown in our lab on a cylindrical screen made of a silver fabric that maintains light polarisation. The screen is about $1.5m$ high with a $4.5m$ diameter. A multiprojection system [10, 20] was setup with half the projectors polarized horizontally and the other half polarized vertically, and viewed with glasses for polarized projection. We have also generated an anaglyph version that can be downloaded from author VC's webpage.

VII. Conclusion

This paper has introduced a novel multi-take method to capture non-periodic omnistereo content using two cameras only on a fixed tripod. The resulting video is composed of several stereo takes that capture action within the field of view, which are pasted on top of a static omnistereo background. A motion parallax analysis is also presented that aims to minimize parallax artifacts due to the cameras rotating off-axis. The only restrictions of the method as it stands are that the tripod is fixed at a single location, and that action of a take has to be within the field of view of the cameras.

Future work could involve merging our approach with existing mosaicing methods for so-called periodic motions (or motion textures) such as waterfalls. This extension would allow both the background and foreground to be dynamic.

References

[1] Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. Panoramic video textures. *ACM Transactions on Graphics*, 24(3):821–827, 2005.

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 2001.

[3] Matthew Brown and David G. Lowe. Recognising panoramas. *IEEE International Conference on Computer Vision*, 2:1218–1225, 2003.

[4] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Dan B. Goldman and Jiun-Hung Chen. Vignette and exposure calibration and compensation. In *IEEE International Conference on Computer Vision*, volume 1, pages 899–906, Washington, DC, USA, 2005.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[8] Ho-Chao Huang and Yi-Ping Hung. Panoramic stereo imaging system with automatic disparity warping and seaming. *Graphical Models and Image Processing*, 60(3):196–208, 1998.

[9] Hiroshi Ishiguro, Masashi Yamamoto, and Saburo Tsuji. Omni-directional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):257–262, 1992.

[10] Sébastien Roy J.-P. Tardif and Martin Trudeau. Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. *International Conference on 3-D Digital Imaging and Modeling*, pages 217–224, 2003.

[11] H.C. Longuet Higgins and K. Prazdny. The interpretation of a moving retinal image. *Royal Society of London*, B-208:385–397, 1980.

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

[13] Shmuel Peleg and Moshe Ben-Ezra. Stereo panorama with a single camera. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:1395, 1999.

[14] Shmuel Peleg, Moshe Ben-Ezra, and Yael Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, 2001.

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

[16] Alex Rav-Acha, Yael Pritch, Dani Lischinski, and Shmuel Peleg. Dynamosaicing: Mosaicing of dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1789–1801, 2007.

[17] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[18] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[19] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press.

[20] J.-P. Tardif and Sébastien Roy. A mrf formulation for coded structured light. *International Conference on 3-D Digital Imaging and Modeling*, pages 22–29, 2005.

[21] Yuanjie Zheng, Stephen Lin, and Sing Bing Kang. Single-image vignetting correction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 461–468, Washington, DC, USA, 2006.