

Image Cube Trajectory Analysis for Concentric Mosaics

Ingo Feldmann, Peter Kauff, and Peter Eisert

Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institute,
Einsteinufer 37, 10587 Berlin, Germany
{feldmann, kauff, eisert}@hhi.fhg.de
<http://bs.hhi.de>

Abstract. We present a new concept for the extension of epipolar image analysis to more general camera configurations like circular camera movements usually occurring for concentric mosaic acquisition. In this way the robust method for 3D scene depth reconstruction which we call Image Cube Trajectory Analysis (ICT) is no longer restricted to horizontal, linear, and equidistant camera movements. Similar to epipolar image analysis the algorithm uses all available views of an image sequence simultaneously. Instead of searching for straight lines in the epipolar image we explicitly compute the trajectories of particular points through the image cube. Variation of the unknown depth leads to different curves. The best match is assumed to correspond to the true depth. It is selected by evaluating color constancy along the curve. For the example of concentric mosaics we derive an occlusion compatible ordering scheme that guarantees an optimal processing of occluding object points.

1 Introduction

Image-based rendering techniques have received much attention in the computer graphics community in the past decade. These methods are based on the 7D plenoptic function [1] that specifies all possible light rays of a dynamic scene and allows the synthesis of arbitrary views by subsampling the 7D space. For practical systems, this space is usually reduced to 5 (plenoptic modeling [2]), 4 (light fields [3]), or 3 (concentric mosaics [4, 5]) degrees of freedom. In this paper, we focus on concentric mosaics captured by a circularly moving camera. These 3D data sets describe scene light intensity as a function of radius, rotation angle, and vertical elevation. Dependent on the radius of the concentric camera motion and the camera's field of view, arbitrary new views can be reconstructed within a certain range. However, for the view synthesis, it is assumed that all object points have the same distance from the center of rotation. Deviations of the real scene geometry from this assumption lead to interpolation artifacts and vertical distortions [5]. These artifacts can be reduced by a perspective depth correction based on the distances of the 3D points from the camera. Unfortunately, scene geometry is usually not known and its estimation is still a hard vision problem.

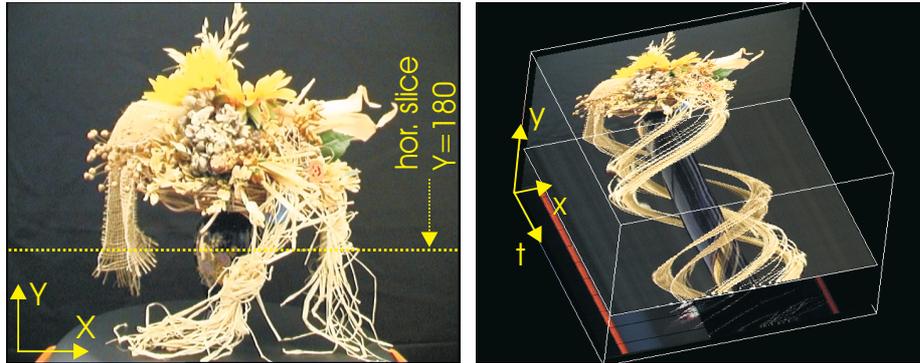


Fig. 1. "Flower" sequence, circular moving camera, **left)** first frame, **right)** image cube representation, horizontal slice for fixed Y-coordinate.

In general, the basic problem of depth estimation from a set of 2D images is the correspondence search [6]. Given a single point in one of the images its correspondences in the other images need to be detected. Depending on the algorithm two or more point correspondences as well as the camera geometry are used to estimate the depth of that point [7]. However, for complex real scenes the correspondence detection problem is still not fully solved. Especially in the case of homogeneous regions, occlusions or noise, it still faces many difficulties. It is now generally recognized that using more than two images can dramatically improve the quality of reconstruction. This is particularly interesting for the depth analysis of concentric mosaics which consist of a very large number of views of a scene.

One method for the simultaneous consideration of all available views is Epipolar Image (EPI) analysis [8]. An epipolar image can be thought of being a horizontal slice (or plane) in the so called *image cube* [9] that can be constructed by collating all images of a sequence. Each EPI represents a single horizontal line ($Y=\text{constant}$) of all available camera views. For a linear camera movement parallel to the horizontal axis of the image plane, all projections of 3D object points remain in the same EPI throughout the entire sequence. Thus, the EPI represents the trajectories of object points. If the camera is moved equidistantly, the path of an arbitrary 3D point becomes a straight line, called *EPI line*. The slope of the line represents the depth. The principle of EPI analysis is the detection of all EPI-lines (and their slopes) in all available EPIs.

The advantage of the EPI analysis algorithm is the joint detection of point correspondences for all available views. Occlusions as well as homogeneous regions can be handled efficiently [9]. The big disadvantage of the algorithm is its restriction to linear equidistant camera movements which prevents its usage for the analysis of concentric mosaics. One idea to overcome this problem is the piecewise linear EPI analysis where small segments of the object point trajectory are approximated by straight lines. This approach can also be applied to circular

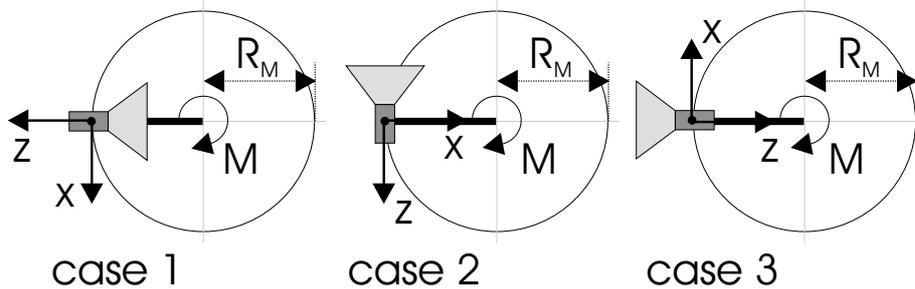


Fig. 2. Camera configurations for different cases. **left (case 1)** turn-table scenario with inwards pointing camera, see [11], **middle (case 2)** concentric mosaic (tangential direction) **right (case 3)** concentric mosaic (normal direction).

camera movements [10] but significantly reduces the amount of reference images and thus robustness of the 3D reconstruction.

In [11], we have proposed a new concept called *image cube trajectory (ICT) analysis* that overcomes this restriction and is able to jointly exploit all available views also for circular camera configurations. For the special case of an inwards facing, circularly moving camera, we derived the analytical shape of the almost sinusoidal object point trajectories (see fig. 1) and proposed a new ICT matching method for robust depth estimation. For concentric mosaics, where the rotating camera usually faces in tangential or outwards normal direction, modifications of the ICT calculation and the optimal occlusion compatible ordering scheme [11] are necessary. This extension to concentric mosaics is addressed in this paper.

2 Image Cube Trajectory Analysis

For the proposed ICT analysis algorithm we suggest an inverse approach to the conventional way of EPI analysis where usually in a first step the EPI lines are detected in the EPI using some kind of segmentation algorithm [8, 10]. In a second step, the corresponding depth is reconstructed from the slopes of the lines. In contrast, our idea is to specify an ICT for an assumed 3D point by determining its parameters from its assumed 3D position. We will call this the *reference ICT*. In a second step we check if this assumption is valid and the reference ICT fits to the image cube. This is done by evaluating color constancy along the entire trajectory in the image cube for different parameter sets. From the best matching ICT the 3D position of the corresponding object point is derived.

For a concentric circular camera motion an arbitrary 3D point may be described in terms of its radius to the center of rotation R , its rotation angle ϕ and its height y . We will show in the following that the ICTs have a well defined structure in the image cube which depends on these 3 parameters. The structure can be exploited to define an efficient and simple occlusion compatible 3D search

strategy. In order to create the above mentioned reference ICTs we need to calibrate the camera system such that all camera positions are known in advance. Robust camera self calibration systems are well known in the literature [12].

3 ICT Analysis for Concentric Mosaics

In this chapter we will derive an analytical description of the ICT structure for concentric mosaics as a function of their three degrees of freedom. Without loss of generality, we will restrict our discussion to the three cases illustrated in fig. 2.

The first case refers to an inwards looking camera (turn-table configuration) while the latter two describe concentric mosaic scenarios with cameras in tangential and normal direction, respectively. For the derivation of the image cube trajectories, consider a 3D object point \mathbf{x} in the scene. Viewed from the camera which rotates around M at a distance of R_M , the object point \mathbf{x} has a circular trajectory with center M and radius R . The location of center M in camera coordinates is given by

$$\mathbf{x}_M = \mathbf{R}^T \cdot \begin{bmatrix} 0 \\ 0 \\ -R_M \end{bmatrix}, \quad (1)$$

where \mathbf{R} specifies the rotation of the camera relative to the camera arm. For an inwards pointing camera, matrix \mathbf{R} is the identity matrix. With this definition, the object point trajectory in camera coordinates x , y , and z can be specified by

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}^T \cdot \begin{bmatrix} R \sin \phi \\ y \\ -R_M + R \cos \phi \end{bmatrix}. \quad (2)$$

Dependent on the camera configuration, different trajectories occur. If the camera points to the rotation center M , a turn-table scenario as described in [11] is modeled. \mathbf{R}_1^T then reduces to the identity matrix. For concentric mosaics, the camera usually points tangentially or outwards (normal direction). For these two cases, the corresponding rotation matrices are

$$\mathbf{R}_2^T = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{R}_3^T = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (3)$$

Perspective projection (with focal length f) of the 3D object point trajectory into the image plane leads to the desired ICT. The 2D image coordinates X and Y become

$$\begin{aligned} X_1 &= -f \frac{x}{z} = \frac{f \frac{R}{R_M} \sin \phi}{1 - \frac{R}{R_M} \cos \phi} = \frac{fq \sin \phi}{1 - q \cos \phi} \\ Y_1 &= -f \frac{y}{z} = \frac{f \frac{y}{R_M}}{1 - \frac{R}{R_M} \cos \phi} = \frac{fp}{1 - q \cos \phi} \end{aligned} \quad (4)$$

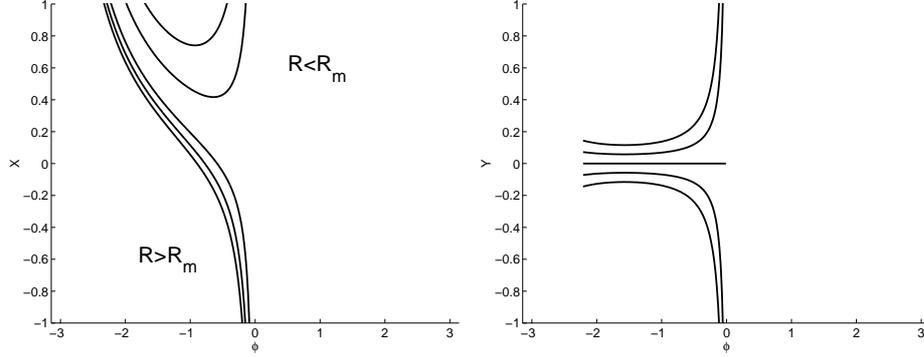


Fig. 3. Trajectory for concentric mosaics with tangentially looking camera. **left)** X coordinate for points with varying radius, **right)** Y coordinate for points with varying height.

for an inwards pointing camera (turn-table scenario),

$$\begin{aligned}
 X_2 &= -f \frac{x}{z} = \frac{-f(1 - \frac{R}{R_M} \cos \phi)}{\frac{R}{R_M} \sin \phi} = \frac{-f(1 - q \cos \phi)}{q \sin \phi} \\
 Y_2 &= -f \frac{y}{z} = \frac{-f \frac{y}{R_M}}{\frac{R}{R_M} \sin \phi} = \frac{-fp}{q \sin \phi}
 \end{aligned} \tag{5}$$

for a concentric mosaic scenario with tangentially pointing camera, and

$$\begin{aligned}
 X_3 &= -f \frac{x}{z} = \frac{f \frac{R}{R_M} \sin \phi}{1 - \frac{R}{R_M} \cos \phi} = \frac{fq \sin \phi}{1 - q \cos \phi} \\
 Y_3 &= -f \frac{y}{z} = \frac{-f \frac{y}{R_M}}{1 - \frac{R}{R_M} \cos \phi} = \frac{-fp}{1 - q \cos \phi}
 \end{aligned} \tag{6}$$

for a concentric mosaic scenario with outwards facing camera. The abbreviations q and p are defined as follows

$$q = \frac{R}{R_M} \quad p = \frac{y}{R_M}. \tag{7}$$

The shapes of the resulting object point trajectories in the image cube are illustrated in figs. 3 and 4 for the two concentric mosaic configurations. The left hand side of fig. 3 shows the X-coordinates plotted over different rotation angles ϕ of a tangentially looking camera. The object point distance from the center specified by q is varied over values 0.6, 0.8, 1.2, 1.4, and 1.6. Object points outside the circle performed by the camera cross the entire image in horizontal direction while points within the inner circle enter the image on one side and leave them on the same side again.

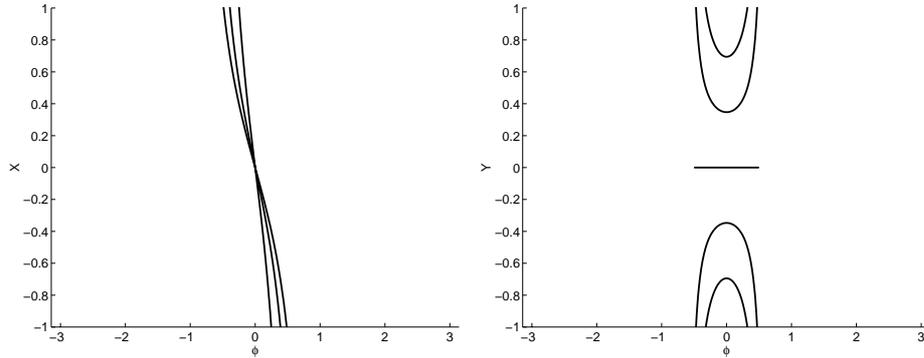


Fig. 4. Trajectory for concentric mosaics with outwards looking camera. **left)** X coordinate for points with varying radius, **right)** Y coordinate for points with varying height.

The image coordinate in vertical direction Y is also altered for different rotation angles ϕ as depicted on the right hand side of fig. 3 for points of different height p . Points that get closer to the camera are shifted towards the horizontal image borders.

Similarly, fig. 4 illustrates the behavior of object points moving in the image cube for the case of a concentric mosaic with an outwards looking camera. Again, the trajectories differ significantly from straight lines. Please note that objects points which are closer to the center M than the camera are not visible.

3.1 Occlusion Handling

As mentioned in chapter 2 the idea of our ICT analysis algorithm is to search within the entire 3D space of the image cube for optimally matching trajectories. In this chapter, we will define the order in which the 3D space has to be analyzed (varying the three parameters R , ϕ , and y) to avoid occlusion problems. Due to the parameterized circular movement of the camera it is possible to derive explicit rules for an occlusion compatible search order.

In general, two points occlude each other if they are both lying on the same line of sight and one of them is closer to the camera than the other one. Depending on the camera orientation, i.e. the rotation matrix \mathbf{R} , different cases occur for these lines of sight which need to be handled differently. The first case is an inwards looking camera, i.e. no camera rotation around y axis exists (fig. 2 left). A detailed description of this case can be found in [11]. In the following, we will – without loss of generality – specialize our discussion to the tangential camera orientation (see fig. 2 middle). It automatically includes the cases of occlusion handling for purely inwards or outwards facing cameras.

Firstly, occlusions for points at different heights y are considered. This is illustrated in fig. 5 left where point P_3 occludes point P_2 . Due to perspective projection points with a smaller distance to plane $y = 0$ (locating the camera

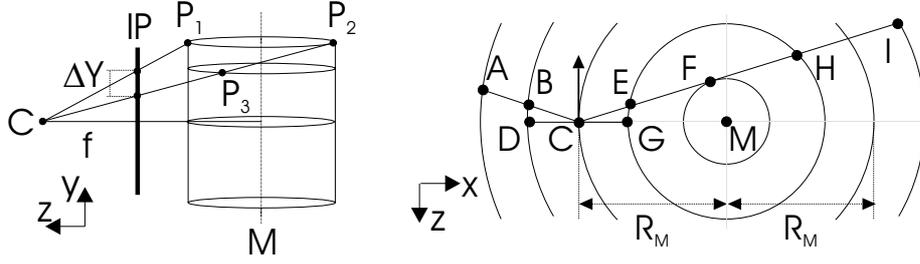


Fig. 5. Projection of 3D points to camera C rotating at radius R_M around M , **left)** vertical occlusions, **right)** horizontal occlusions

centers) may occlude those with a larger one. Therefore, an occlusion compatible search algorithm has to start at the height of $y = 0$ varying parameters R and ϕ as described later. Afterwards, this process is repeated for increasing distances from the camera plane $y = 0$.

The occlusion handling for all points on one common plane $y = const$ is more complicated. Consider a camera C rotating on a circular path with radius R_M around the midpoint M (see fig. 5 right). For object points with radius R it is necessary to distinguish between two cases. Firstly, $R < R_M$ and, secondly, $R > R_M$. In the first case the line of sight intersects any circle twice. Therefore, points with different radii (points E and F) as well as points with the same radius occlude each other (points E and H) if they are located on the same line of sight. For an occlusion compatible ICT search it is necessary to consider the four quadrants of each circle separately. Again, a detailed description of this case can be found in [11]. If $R > R_M$ the two points B and D on the same circle never occlude each other. A point B occludes a point A only if both are lying on the same line of sight and $R_B < R_A$ as shown in fig. 5 right. Therefore, an occlusion compatible ICT search needs to consider all points with smaller radius first. This holds for the case of inwards looking cameras as well. In the figure point I will always be occluded by points H , F or E if they are all on the same line of sight ($R_I > R_M > R_{H,E,F}$).

Following those rules the occlusion compatible search algorithm described in chapter 2 must start with points in the inner circle ($R < R_M$) varying the rotation angle ϕ for each considered radius as described in [11]. Afterwards, all points $R > R_M$ need to be considered. For each radius R the rotation angle ϕ must be varied within the range $0 \leq \phi < 2\pi$. As mentioned earlier, the whole process needs to be repeated for different heights y starting at $y = 0$ increasing the distance to the plane $y = 0$. Note, that similar to the EPI analysis technique successfully detected trajectories are masked in the image cube and therefore excluded from the subsequent search.

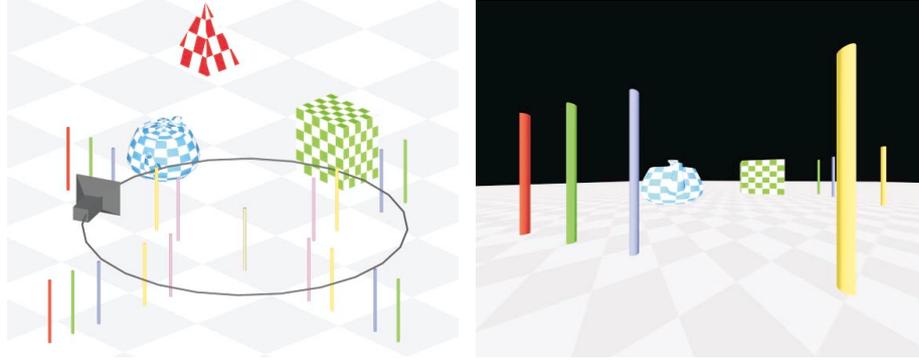


Fig. 6. **left)** 3D scene used for generation of synthetic concentric mosaics, **right)** one view of the virtual camera

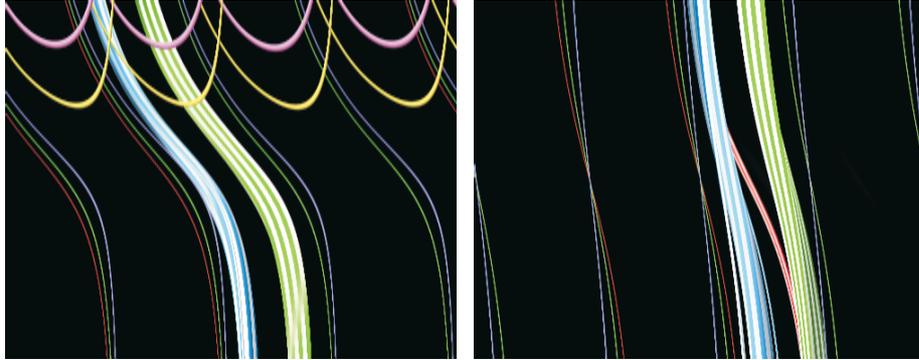


Fig. 7. **left)** ICT for a concentric mosaic with tangential camera setting, **right)** ICT for the corresponding normal camera setup

4 Simulations and experimental results

In order to prove the theoretical insights, simulations on synthetic concentric mosaics were performed. First, a synthetic 3D scene as depicted on the left hand side of fig. 6 is created. A camera moves concentrically facing in tangential (case 2) or outwards in radial direction (case 3), respectively. The right hand side of fig. 6 illustrates one camera frame for the tangential case that is rendered from the 3D scene using computer graphics techniques. From these camera views, image cubes were constructed. Fig. 7 shows two horizontal slices of those cubes ($y = const$) for the two different concentric mosaic scenarios illustrating object point trajectories for different radii R . For better clarity, the scene consists of vertical bars without y -dependency for the considered range. For a tangentially facing camera (left hand side of fig. 7), the trajectories have different behavior for object point radii R larger or smaller than R_M . Generally, the rendered trajectories show the same properties as the curves computed in section 3 and

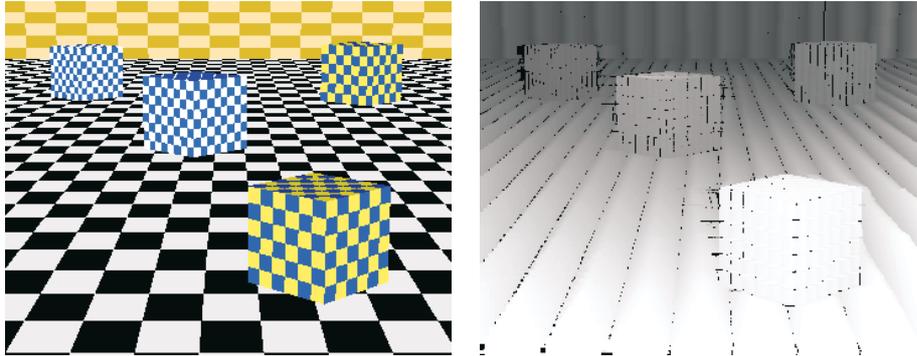


Fig. 8. "Boxes", virtual sequence with linear horizontal camera movement, **left)** original image, **right)** estimated depth

depicted on the left side of figs. 3 and 4 validating the insights of this ICT analysis approach. These figures also prove the rules for occlusion compatible ordering described in section 3.1.

Further on, in order to validate the general ICT analysis algorithm we applied the proposed inverse search strategy described in section 2 to the case of horizontally moving cameras where the trajectories can be modeled as straight lines. Following the mentioned occlusion compatible ordering depth was varied from small to large values corresponding to decreasing slopes of ICTs that are matched with the image cube. To detect a reference ICT in the image cube, we are using a straightforward approach: The color variation along the whole ICT is compared to a given threshold. If it is less than this value we consider the ICT as detected. Although the described matching algorithm is very primitive, it still provides reasonable results and demonstrates the general idea of our approach quite well. Fig. 8 shows the result of a 3D reconstruction of a synthetic scene. The depth maps are generated without any filtering, interpolation, or postprocessing. As we can see from the figure, it is possible to reconstruct the depth maps with reasonable quality.

5 Conclusions and Future Work

In this paper we have discussed the concept for an extension of conventional epipolar image analysis to the case of concentric circular camera movements. We call our approach ICT analysis. The cases of tangential and normal camera orientations have been analyzed in detail. We have shown that well defined trajectory structures in the image cube exist which can be used to derive rules for a systematic occlusion compatible search algorithm. The basic concept was validated by reconstructing depth maps for a simple synthetic scene and a linear camera movement. It is left to future work to apply the matching algorithm to other camera movements and to improve it using more sophisticated methods

for path detection. The proposed algorithm is not restricted to circular camera movements only. Our extension to EPI analysis can be used to define rules for other parameterized camera movements such as parabolic camera paths etc. as well. Therefore, we are working on more generalized rules and algorithms to solve this problem.

References

1. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In Landy, M., Movshon, J.A., eds.: Computational Models of Visual Processing. MIT Press, Cambridge, Mass (1991) 3–20
2. McMillan, L., Bishop, G.: Plenoptic modelling: An image-based rendering system. In: Proc. Computer Graphics (SIGGRAPH), New Orleans, USA (1995) 31–42
3. Levoy, M., Hanrahan, P.: Light field rendering. In: Proc. Computer Graphics (SIGGRAPH), New Orleans, LA, USA (1996) 31–42
4. Peleg, S., Ben-Ezra, M.: Stereo panorama with a single camera. In: Proc. Computer Vision and Pattern Recognition, Ft. Collins, USA (1999) 395–401
5. Shum, H.Y., He, L.W.: Rendering with concentric mosaics. In: Proc. Computer Graphics (SIGGRAPH), Los Angeles, USA (1999) 299–306
6. Mellor, J.P., Teller, S., Lozano-Perez, T.: Dense depth maps from epipolar images. Technical Report AIM-1593, MIT (1996)
7. Beardsley, P., Torr, P., Zisserman, A.: 3D model acquisition from extended image sequences. In: Proc. European Conference on Computer Vision (ECCV), Cambridge, UK (1996) 683–695
8. Bolles, R.C., Baker, H.H., Marimont, D.H.: Epipolar image analysis: An approach to determine structure from motion. International Journal of Computer Vision (1987) 7–55
9. Criminisi, A., Kang, S.B., Swaminathan, R., Szeliski, R., Anandan, P.: Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. Technical Report MSR-TR-2002-19, Microsoft Research (2002)
10. Li, Y., Tang, C.K., Shum, H.Y.: Efficient dense depth estimation from dense multiperspective panoramas. In: Proc. International Conference on Computer Vision (ICCV), Vancouver, B.C., Canada (2001) 119–126
11. Feldmann, I., Eisert, P., Kauff, P.: Extension of epipolar image analysis to circular camera movements. In: Proc. International Conference on Image Processing (ICIP), Barcelona, Spain (2003)
12. Rothwell, C., Csurka, G., Faucher, O.D.: A comparison of projective reconstruction methods for pairs of views. Technical Report 2538, INRIA (1995)