Image-Based Rendering Methods

for 3D Video-communications

Andrea Fusiello*

http://profs.sci.univr.it/~fusiello

Brixen, July 5 2007

Contents

1 Introduction

The migration of immersive media towards telecommunication application continues to advance:

- immersive video conferencing
- collaborative virtual environment
- Immersive TV

The development from 2D toward 3D video-communication is a key component for such applications.

Been able to (dynamically) change the view point of the observer/user is the most general and challenging scenario of 3D video-communication.

Image Based Rendering tackle this problem: given images of a real scene, synthesize novel views of the same scene from a virtual camera by processing the real images.

1.1 Image Based Rendering

One of the central themes in the field of computer graphics is the generation of images of artificial environments capable to convince the viewer that they are looking into a real scene (photorealism).

Model-based rendering: The model specify the geometry of scene (usually as 3D mesh) and the surface properties (how a surface interact with light). Images of the scene are generated by render algorithm such as *ray tracing* or *radiosity*.

This works well in synthetic scenes where all elements are well defined.

In the context of video communication all the information that is available consist of a set of video streams, taken by real cameras from different view points.

One way would be to invert the imaging process and to reconstruct a full 3D scene model from real images (Computer Vision). Then follow the model-based rendering pipeline (Computer Graphics).

It turns out that a full 3D reconstruction is not necessary, and sometimes even not desired.

Image-Based Rendering: Images of a real scene are taken from various view points and novel views of the same scene are synthesized from a virtual camera by processing the real images, without the need of a full three-dimensional reconstruction.

IBR greatly simplifies the modelling of real scenes as only a number of example images need to be acquired.

A second advantage of IBR is that the complexity of rendering is decoupled from the complexity of the scene (number of triangles).

Moreover, photorealism is improved, as novel views are generated by re-sampling the real images.

In the following we will

- outline the theory relevant for understanding the imaging process of a 3D scene onto a camera and the geometrical rationale behind view synthesis.
- Then we will survey the IBR techniques,
- and finally we will concentrate on the problem of computing correspondences between images, as they are largely used as a geometry proxy in many IBR methods.

2 Pin-hole Camera

The pin-hole camera is described by its *optical centre* $\mathbf{C}$ (also known as *camera projection centre*) and the *image plane*.

The distance of the image plane from $\mathbf{C}$ is the *focal length* f.

The plane parallel to the image plane containing the optical centre is called the *principal plane* or *focal plane* of the camera.

A 3-D point is projected onto the image plane with the line containing the point and the optical centre (see Figure 1).

By similar triangles it is readily seen that the 3-D point $(x, y, z)^T$ is mapped to the point $(fx/z, fy/z)^T$ on the image plane.

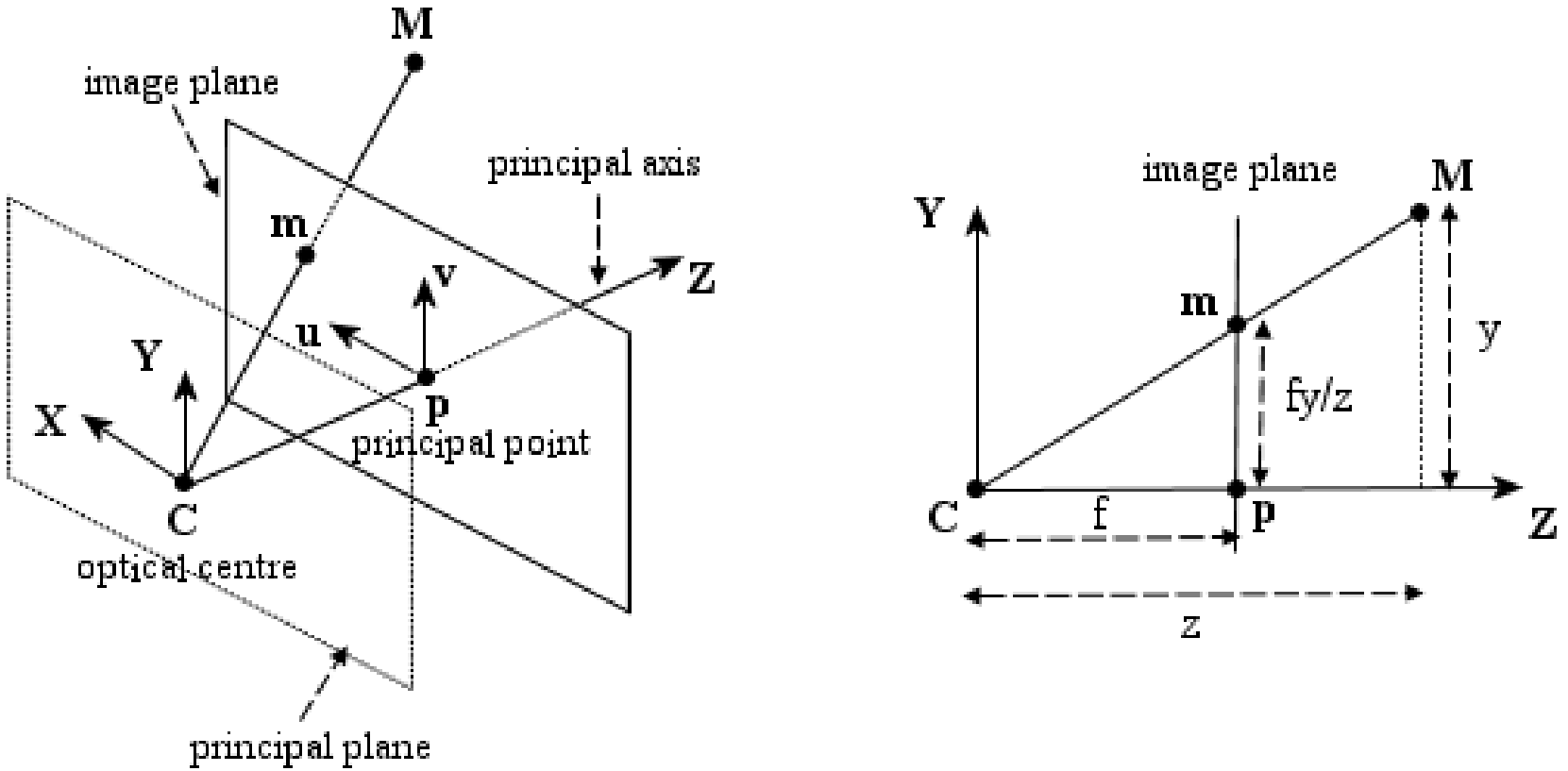


Fig. 1. Pin-hole camera geometry. The left figure illustrates the projection of the point **M** on the image plane by drawing the line through the camera centre **C** and the point to be projected. The right figure illustrates the same situation in the YZ plane, showing the similar triangles used to compute the position of the projected point **m** in the image plane.

2.1 The camera projection matrix

If the world and image points are represented by homogeneous vectors, then perspective projection can be expressed in terms of matrix multiplication as

$$\zeta \mathbf{m} = P\mathbf{M} \tag{1}$$

where

- $\mathbf{M} = (x, y, z, 1)^T$ are the homogeneous coordinates of the 3-D point,
- $\mathbf{m} = (u, v, 1)^T$ are the homogeneous pixel coordinates of the image point,
- ζ is the distance of $\mathbf{M}$ from the focal plane of the camera and
- P is the matrix describing the mapping, called the *camera projection matrix*.

If reference systems are chosen as in Fig. 1 then the camera matrix is

$$\begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2}$$

which further redudes to $P = [I|\mathbf{0}]$ if $f = 1$.

More in general, the camera matrix is the product of two matrices

$$P = K[I|\mathbf{0}]G = K[R|\mathbf{t}] \tag{3}$$

Extrinsic parameters

$$G = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{4}$$

G is composed by a rotation matrix R and a translation vector $\mathbf{t}$. It describes the position and orientation of the camera with respect to an external (world) coordinate system. It depends on six parameters, called *extrinsic* parameters.

The rows of R are unit vectors that, together with the optical centre, define the *camera reference frame*, expressed in world coordinates.

Intrinsic parameters

$$K = \begin{bmatrix} f/s_x & f/s_x \cot\theta & o_x \\ 0 & f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

K is the *camera calibration matrix*; it encodes the transformation in the image plane from the so-called *normalized camera coordinates* to *pixel coordinates*.

It depends on the so-called *intrinsic* parameters:

- focal distance f (in mm),
- principal point (or image centre) coordinates o_x, o_y (in pixel),
- width (s_x) and height (s_y) of the pixel footprint on the camera photosensor (in mm),
- angle θ between the axes (usually $\pi/2$).

The ratio s_y/s_x is the aspect ratio (usually close to 1).

General camera

If P describes a camera, also λP for any $0 \neq \lambda \in \mathbb{R}$ describes the same camera, since these give the same image point for each scene point.

In this case we can also write:

$$\mathbf{m} \simeq P\mathbf{M} \tag{6}$$

where $\simeq$ means "equal up to a scale factor."

In general, the camera projection matrix is a 3×4 full-rank matrix and, being homogeneous, it has 11 degrees of freedom.

Using QR factorization, it can be shown that any 3×4 full rank matrix P can be factorised as:

$$P = \lambda K[R|\mathbf{t}], \tag{7}$$

(λ is recovered from $K(3,3) = 1$).

Projection centre

The camera projection centre $\mathbf{C}$ is the only point for which the projection is not defined, i.e.:

$$P\mathbf{C} = P\begin{pmatrix}\tilde{\mathbf{C}}\\ 1\end{pmatrix} = \mathbf{0} \tag{8}$$

where $\tilde{\mathbf{C}}$ is a 3-D vector containing the Cartesian (non-homogeneous) coordinates of the optical centre.

After solving for $\tilde{\mathbf{C}}$ we obtain:

$$\tilde{\mathbf{C}} = -P_{1:3}^{-1}P_4 \tag{9}$$

where the matrix P is represented by the block form: $P = [P_{1:3}|P_4]$ (the subscript denotes a range of columns).

Optical ray

The projection can be geometrically modelled by a ray through the optical centre and the point in space that is being projected onto the image plane (see Fig. 1).

The *optical ray* of an image point $\mathbf{m}$ is the locus of points in space that projects onto $\mathbf{m}$.

It can be described as a parametric line passing through the camera projection centre $\mathbf{C}$ and a special point (at infinity) that projects onto $\mathbf{m}$:

$$\mathbf{M} = \begin{pmatrix} -P_{1:3}^{-1} P_4 \\ 1 \end{pmatrix} + \zeta \begin{pmatrix} P_{1:3}^{-1} \mathbf{m} \\ 0 \end{pmatrix}, \quad \zeta \in \mathbb{R}. \tag{10}$$

The parameter ζ in Eq. (10) represent the the depth of the point $\mathbf{M}$ only if P has been scaled so that $\lambda = 1$ in Eq. (7).

Knowing the intrinsic parameters is equivalent to being able to trace the optical ray of any image point (with $P = [K|\mathbf{0}]$).

3 Two-View Geometry

The two-view geometry is the intrinsic geometry of two different perspective views of the same 3-D scene (see Figure 2). It is usually referred to as *epipolar geometry*.

The two perspective views may be acquired simultaneously, for example in a stereo rig, or sequentially, for example by a moving camera. From the geometric viewpoint, the two situations are equivalent, provided that that the scene do not change between successive snapshots.

Most 3-D scene points must be visible in both views simultaneously. This is not true in case of occlusions, i.e., points visible only in one camera. Any unoccluded 3-D scene point $\mathbf{M} = (x, y, z, 1)^T$ is projected to the left and right view as $\mathbf{m}_\ell = (u_\ell, v_\ell, 1)^T$ and $\mathbf{m}_r = (u_r, v_r, 1)^T$, respectively (see Figure 2).

Image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are called *corresponding points* (or conjugate points) as they represent projections of the same 3-D scene point $\mathbf{M}$.

The knowledge of image correspondences enables scene reconstruction from images.

The concept of correspondence is a cornerstone of multiple-view vision. In this notes we assume *known correspondences*, and explore their use in geometric algorithms. Techniques for computing dense correspondences are surveyed in [21, 3].

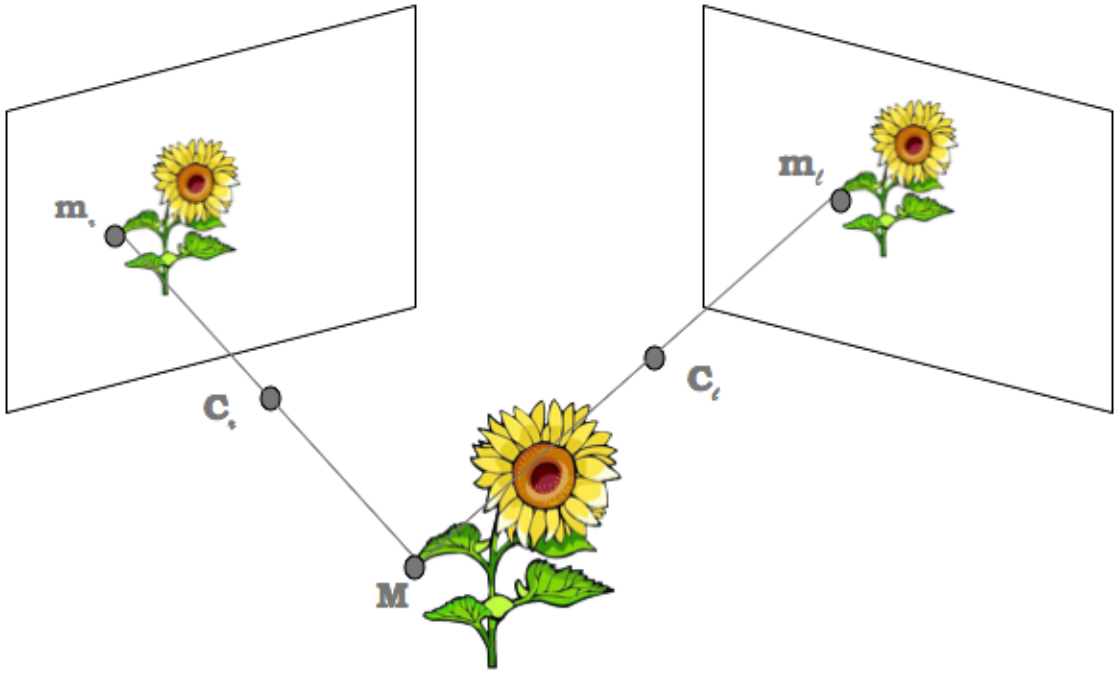


Fig. 2. Two perspective views of the same 3-D scene. $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are corresponding points, as they are the projection of the same 3-D point, $\mathbf{M}$.

We will refer to the camera projection matrix of the left view as P_ℓ and of the right view as P_r. The 3-D point $\mathbf{M}$ is then imaged as (11) in the left view, and (12) in the right view:

$$\zeta_\ell \mathbf{m}_\ell = P_\ell \mathbf{M} \tag{11}$$

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M}. \tag{12}$$

Geometrically, the position of the image point $\mathbf{m}_\ell$ in the left image plane I_ℓ can be found by drawing the optical ray through the left camera projection centre $\mathbf{C}_\ell$ and the scene point $\mathbf{M}$. The ray intersects the left image plane I_ℓ at $\mathbf{m}_\ell$.

Similarly, the optical ray connecting $\mathbf{C}_r$ and $\mathbf{M}$ intersects the right image plane I_r at $\mathbf{m}_r$.

The relationship between image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ is given by the epipolar geometry, described in Section 3.1.

3.1 Epipolar Geometry

The epipolar geometry describes the geometric relationship between two perspective views of the same 3-D scene.

The key finding, discussed below, is that *corresponding image points must lie on particular image lines*, which can be computed without information on the calibration of the cameras.

This implies that, given a point in one image, one can search the corresponding point in the other along a line and not in a 2-D region, a significant reduction in complexity.

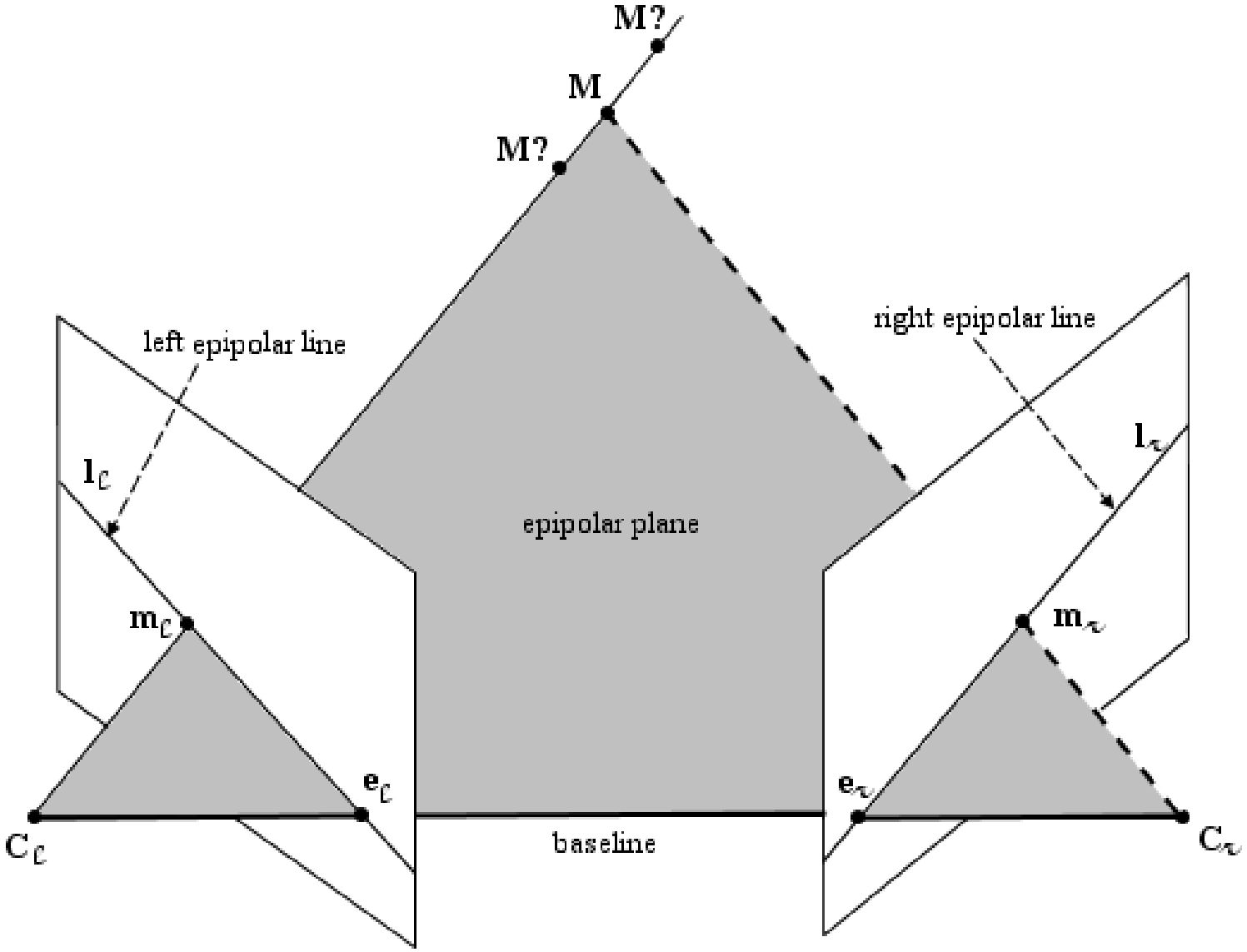


Fig. 3. The epipolar geometry and epipolar constraint.

Any 3-D point $\mathbf{M}$ and the camera projection centres $\mathbf{C}_\ell$ and $\mathbf{C}_r$ define a plane that is called *epipolar plane*.

The projections of the point $\mathbf{M}$, image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$, also lie in the epipolar plane since they lie on the rays connecting the corresponding camera projection centre and point $\mathbf{M}$.

The conjugate epipolar lines, $\mathbf{l}_\ell$ and $\mathbf{l}_r$, are the intersections of the epipolar plane with the image planes. The line connecting the camera projection centres $(\mathbf{C}_\ell, \mathbf{C}_r)$ is called the *baseline*.

The baseline intersects each image plane in a point called *epipole*.

By construction, the left epipole $\mathbf{e}_\ell$ is the image of the right camera projection centre $\mathbf{C}_r$ in the left image plane. Similarly, the right epipole $\mathbf{e}_r$ is the image of the left camera projection centre $\mathbf{C}_\ell$ in the right image plane.

All epipolar lines in the left image go through $\mathbf{e}_\ell$ and all epipolar lines in the right image go through $\mathbf{e}_r$.

The epipolar constraint.

An epipolar plane is completely defined by the camera projection centres and one image point.

Therefore, given a point $\mathbf{m}_\ell$, one can determine the epipolar line in the right image on which the corresponding point, $\mathbf{m}_r$, must lie.

The equation of the epipolar line can be derived from the equation describing the optical ray. As we mentioned before, the right epipolar line corresponding to $\mathbf{m}_\ell$ geometrically represents the projection (Eq. (1)) of the optical ray through $\mathbf{m}_\ell$ (Eq. (10)) onto the right image plane:

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M} = \underbrace{P_r \begin{pmatrix} -P_{\ell_{1:3}}^{-1} P_{\ell_4} \\ 1 \end{pmatrix}}_{\mathbf{e}_r} + \zeta_\ell P_r \begin{pmatrix} P_{\ell_{1:3}}^{-1} \mathbf{m}_\ell \\ 0 \end{pmatrix} \tag{13}$$

If we now simplify the above equation we obtain:

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell \underbrace{P_{r_{1:3}} P_{\ell_{1:3}}^{-1} \mathbf{m}_\ell}_{\mathbf{m}'_\ell} \tag{14}$$

This is the equation of a line – parametrized by the depth ζ_ℓ – through the right epipole $\mathbf{e}_r$ and the image point $\mathbf{m}'_\ell$, which represents the projection onto the right image plane of the point at infinity of the optical ray of $\mathbf{m}_\ell$.

The equation for the left epipolar line is obtained in a similar way.

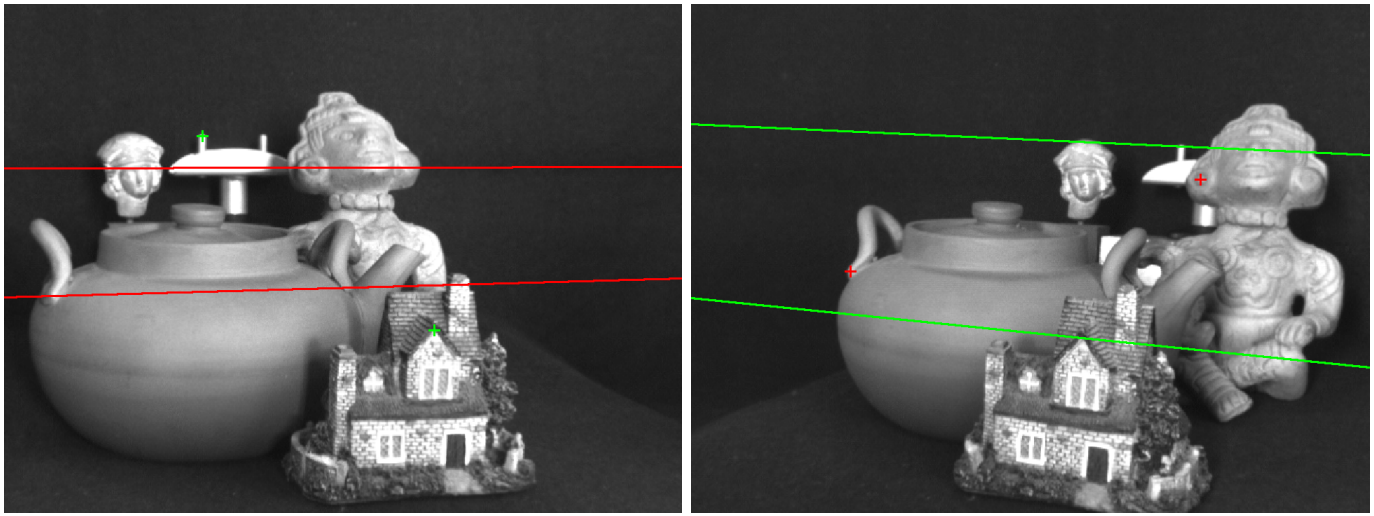


Fig. 4. Left and right images with epipolar lines.

If we take the first camera reference frame as the world reference frame, we can write the following two general camera matrices:

$$P_\ell = K_\ell[I|\mathbf{0}] = [K_\ell|\mathbf{0}] \quad P_r = K_r[R|\mathbf{t}] \tag{15}$$

Then, two corresponding points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are related by

$$\zeta_r \mathbf{m}_r = \zeta_\ell K_r R K_\ell^{-1} \mathbf{m}_\ell + K_r \mathbf{t}. \tag{16}$$

Depth-based warping If the depth of a pixel is given, it can be forward-mapped from the real (left) view into the virtual view using Eq. (16). Indeed:

$$\mathbf{m}_v \simeq \zeta_\ell K_v R K_\ell^{-1} \mathbf{m}_\ell + K_v \mathbf{t}. \tag{17}$$

where R and $\mathbf{t}$ specify the position and orientation of the virtual camera with respect to the real one.

Fig. 5. The image and depth information were acquired simultaneously using a laser-based 3D scanner. The warped image is shown on the right. From [20].

Problems with warping:

- Image folding: more than one pixel in the reference view maps into a single pixel in the extrapolated view
- Holes: information missing (not visible) in the reference view is required in the extrapolated view.
- Magnification: the projected area of a surface increases in the extrapolated view.

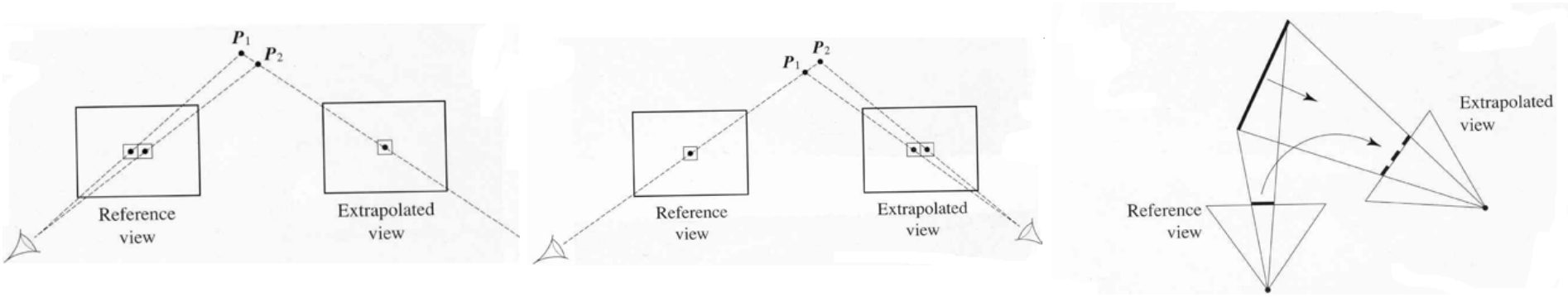


Fig. 6. Depth-based warping artifacts. (From A. Watt.)

Image folding can be avoided by following a suitable evaluation order [18], that guarantee that pixels closer to the viewer are warped after the others, thereby overwriting them (e.g., if synthetic camera is translated to the left, pixel must be processed from right to left.)

Magnification artifacts occurs also in texture mapping or 2D image warping. They are typically solved by interpolation or by drawing "fat" pixels (*splatting*).

Holes are more difficult to solve. In the lack of more information one can only guess the missing values. *Layered Depth Images* [23] solve the occlusion problem by associating to each pixel many depths, namely the depth of each surface (layer) that the optical ray through the pixel would intersect. Rendering is done back-to-front, by processing one layer at a time.

Epipolar transfer

If the depth is not known, but two images with a dense correspondence map are given instead, point transfer is still possible.

If the epipolar geometry of three cameras I_r, I_ℓ and I_v is known, given two corresponding points $\mathbf{m}_r$ and $\mathbf{m}_\ell$ in the two real views, the position of the corresponding point $\mathbf{m}_v$ in view I_v is completely determined (Figure 7). Indeed, $\mathbf{m}_v$ belongs simultaneously to the epipolar line of $\mathbf{m}_r$ and to the epipolar line of $\mathbf{m}_\ell$.

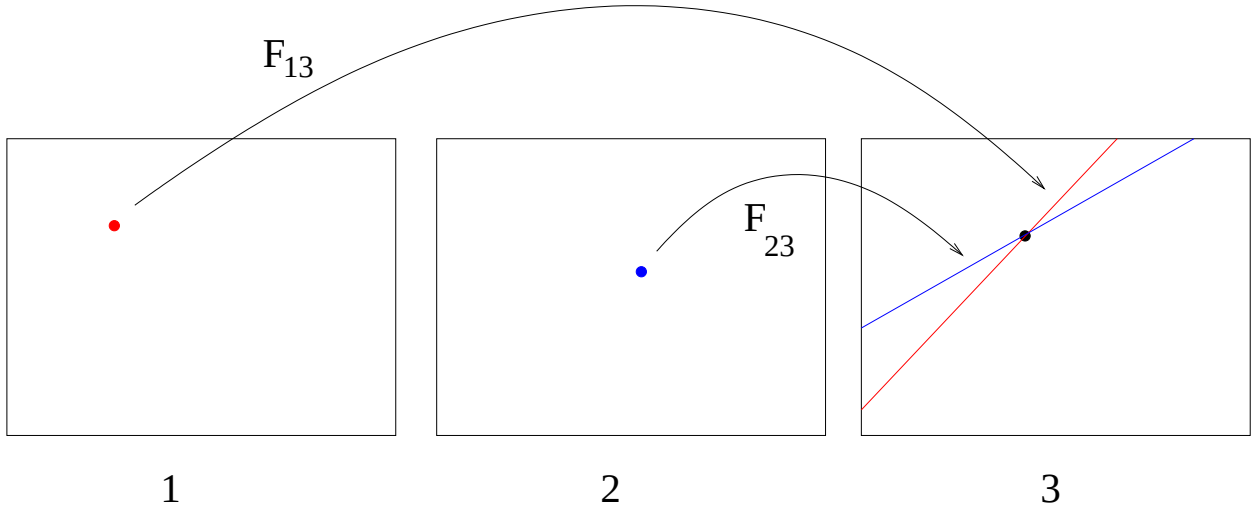


Fig. 7. Point transfer using epipolar constraints between three views.

Fig. 8. From left to right: The two real views and the predicted view. (from [13]).

This description fails when the three optical rays are coplanar.

A more compact and stable description of the geometry of three views is given by the *Trifocal tensor* [2], which has been used as well for point transfer.

3.2 Rectification

Given a pair of stereo images, *epipolar rectification* (or simply *rectification*) determines a transformation of each image plane such that *pairs of conjugate epipolar lines become collinear and parallel to one of the image axes* (usually the horizontal one).

The rectified images can be thought of as acquired by two new virtual cameras, obtained by rotating the actual cameras and possibly modifying the intrinsic parameters.

The important advantage of rectification is that computing stereo correspondences is made simpler, because search is done along the horizontal lines of the rectified images.

We assume here that *the stereo pair is calibrated*, i.e., the cameras' intrinsic parameters, mutual position and orientation are known. This assumption is not strictly necessary [9, 17, 11], but leads to a simpler technique and less distorted images.

Specifying virtual cameras.

Given the actual camera matrices P_{or} and $P_{o\ell}$, the idea behind rectification is to define two new *virtual* cameras P_{nr} and $P_{n\ell}$ obtained by rotating the actual ones around their optical centers until focal planes becomes coplanar, thereby containing the baseline (Figure 9). This ensures that epipoles are at infinity, hence epipolar lines are *parallel*.

To have *horizontal* epipolar lines, the baseline must be parallel to the x-axis of both virtual cameras. In addition, to have a proper rectification, conjugate points must have the *same vertical coordinate*.

In summary: positions (i.e, optical centers) of the virtual cameras are the same as the actual cameras, whereas the orientation of both virtual cameras differs from the actual ones by suitable rotations; intrinsic parameters are the same for both cameras.

Therefore, the two resulting virtual cameras will differ only in their optical centers, and they can be thought as a single camera translated along the x-axis of its reference system.

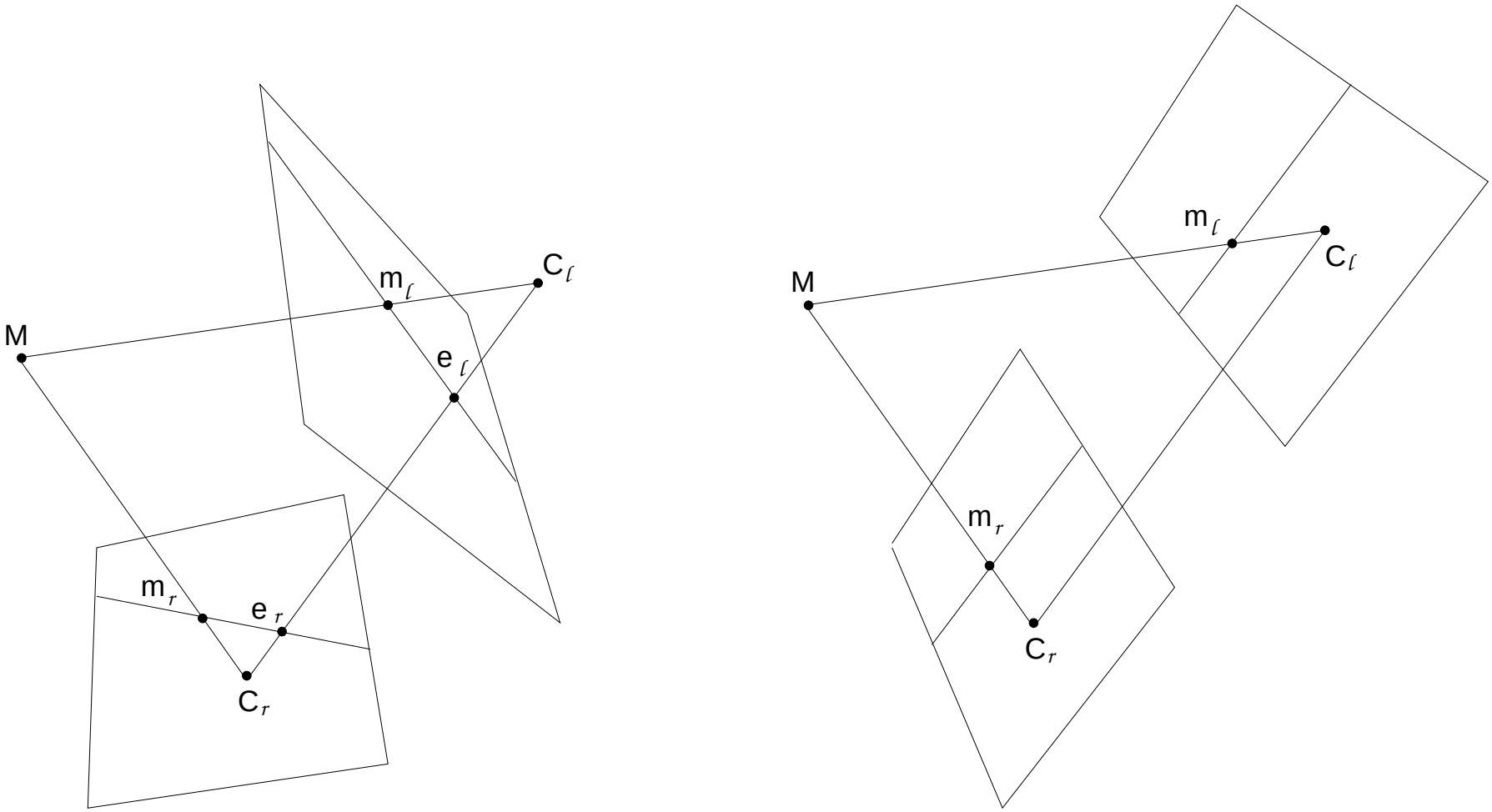


Fig. 9. Epipolar geometry before and after rectification.

Using Eq. (7) and Eq. (9), we can write the virtual cameras matrices as:

$$P_{n\ell} = K[R \mid -R\,\tilde{\mathbf{C}}_\ell], \qquad P_{nr} = K[R \mid -R\,\tilde{\mathbf{C}}_r]. \tag{18}$$

In order to define them, we need to assign $K, R, \tilde{\mathbf{C}}_\ell, \tilde{\mathbf{C}}_r$

The optical centers $\mathbf{C}_\ell$ and $\mathbf{C}_r$ are the same as the actual cameras. The intrinsic parameters matrix K can be chosen arbitrarily. The matrix R, which gives the orientation of both cameras will be specified by means of its row vectors:

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \tag{19}$$

that are the x, y, and z-axes, respectively, of the virtual camera reference frame, expressed in world coordinates.

According to the previous comments, we take:

(i) The x-axis parallel to the baseline: $\mathbf{r}_1 = (\tilde{\mathbf{C}}_r - \tilde{\mathbf{C}}_\ell)/||\tilde{\mathbf{C}}_r - \tilde{\mathbf{C}}_\ell||$

(ii) The y-axis orthogonal to x (mandatory) and to an arbitrary unit vector $\mathbf{k}$: $\mathbf{r}_2 = \mathbf{k} \times \mathbf{r}_1$

(iii) The z-axis orthogonal to xy (mandatory) : $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$

In point 2, $\mathbf{k}$ fixes the position of the y-axis in the plane orthogonal to x. In order to ensure that the virtual cameras look in the same direction as the actual ones, $\mathbf{k}$ is set equal to the direction of the optical axis of one of the two actual cameras.

We assumed that both virtual cameras have the same intrinsic parameters. Actually, the horizontal components of the image centre (v_0) can be different, and this degree of freedom might be exploited to "center" the rectified images in the viewport by applying a suitable horizontal translation.

The rectifying transformation.

In order to rectify the images, we need to compute the transformation mapping the image plane of P_o onto the image plane of P_n.

According to the equation of the optical ray, if $\mathbf{M}$ projects to $\mathbf{m}_o$ in the actual image and to $\mathbf{m}_n$ in the rectified image, we have:

$$\begin{cases} \tilde{\mathbf{M}} = \tilde{\mathbf{C}}_+ \zeta_o P_{o_{1:3}}^{-1} \mathbf{m}_o \\ \tilde{\mathbf{M}} = \tilde{\mathbf{C}}_+ \zeta_n P_{n_{1:3}}^{-1} \mathbf{m}_n \end{cases} \tag{20}$$

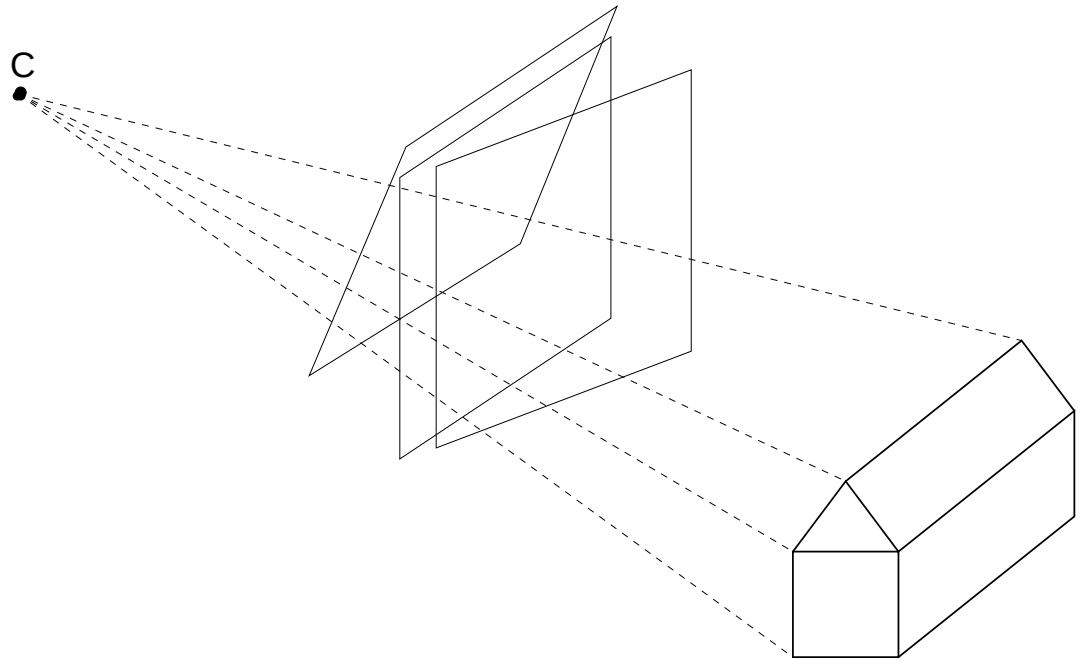
hence

$$\mathbf{m}_n = \frac{\zeta_o}{\zeta_n} \underbrace{P_{n_{1:3}} P_{o_{1:3}}^{-1}}_{H} \mathbf{m}_o \tag{21}$$

The rectifying transformation is a linear transformation of the projective plane (a *collineation*) given by the 3×3 matrix H.

It is understood that this has to be applied to the left and right images.

It is useful to think of an image as the intersection of the image plane with the cone of rays between points in 3-D space and the optical centre. We are moving the image plane while leaving fixed the cone of rays.



Rectification is actually an instance of **view synthesis**, where the rectified views are obtained with a virtual rotation of the real cameras.

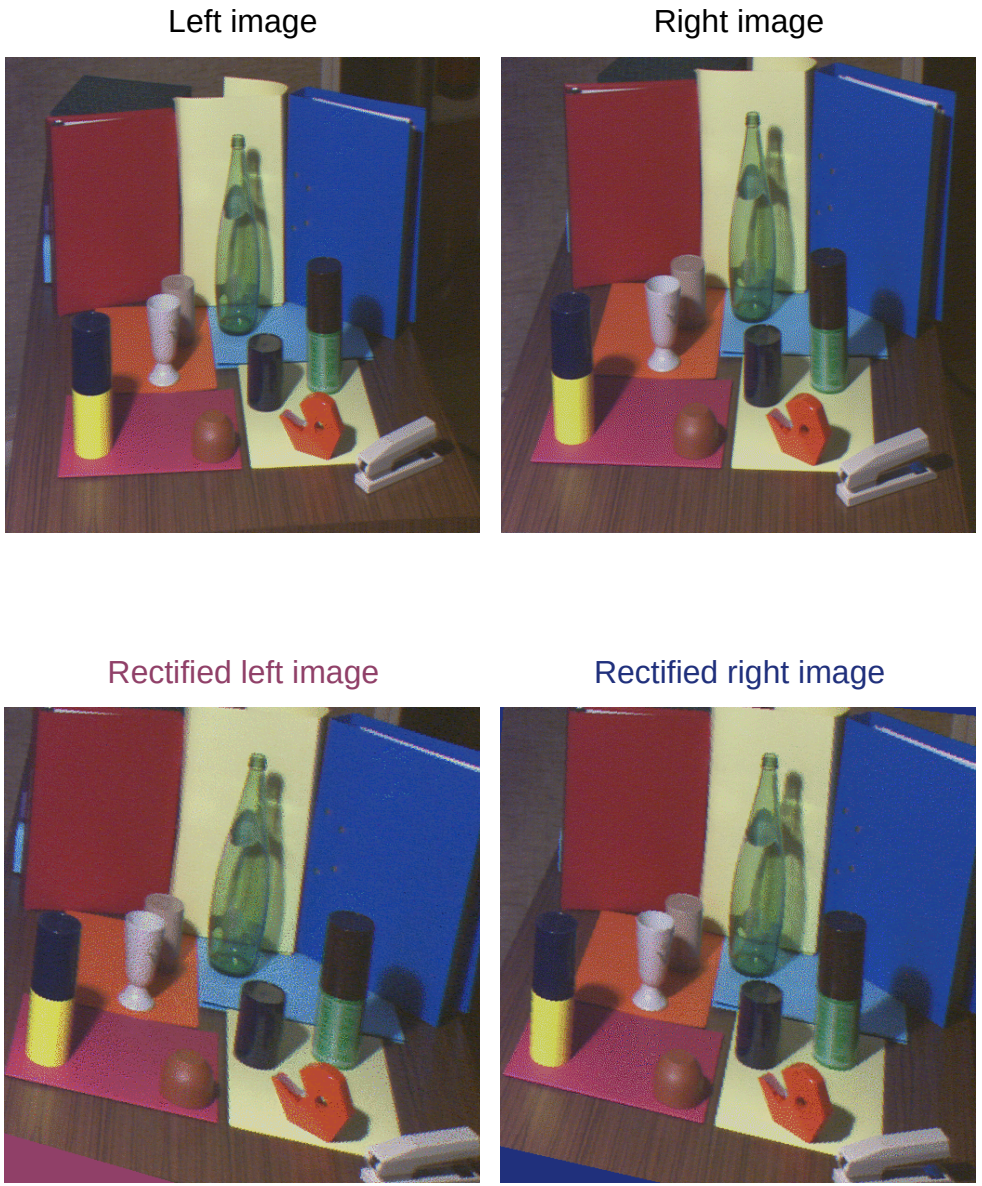


Fig. 10. Original and rectified stereo pair.

Disparity-based interpolation (view morphing.) This technique interpolates novel perspective views from a given pair of real views and a dense correspondence map.

In the rectified images, two corresponding points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are related by

$$\mathbf{m}_r = \mathbf{m}_\ell + \frac{1}{\zeta} K[t_x, 0, 0]^T. \tag{22}$$

The difference $\mathbf{m}_r - \mathbf{m}_\ell$ is called *disparity* d (as only the first component is different from zero, we deem d to be a scalar.) The disparity map is computed from the dense correspondence map.

The point transfer equation into the virtual view (I_v) is

$$\mathbf{m}_v = \mathbf{m}_\ell + [\alpha d, 0, 0]^T \quad \alpha \in [0, 1]. \tag{23}$$

It is easy to verify that interpolating the disparity is equivalent to placing the virtual camera at intermediate positions along the baseline from 0 to t_x.

The algorithm for rectified views was introduced by [6], and it was then extended to tilted cameras by [22]. They simply rectify the real cameras, do the interpolation and eventually de-rectify the resulting virtual view.

Fig. 11. The left and right images are the real ones, the central view is interpolated (from [22].)

3.3 Planes and collineations

When observing a plane, we obtain an interesting specialization of the epipolar geometry of two views.

First, let us establish that the map between a world plane and its perspective image is a collineation of $\mathbb{P}_2$. The easiest way to see it is to choose the world coordinate system such that the plane has equation $z = 0$.

Expanding the projection equation gives:

$$\zeta \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = [P_1|P_2|P_4] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{24}$$

Points are mapped from the world plane to the image plane with a 3×3 (non-singular) matrix, which represents a collineation of $\mathbb{P}_2$.

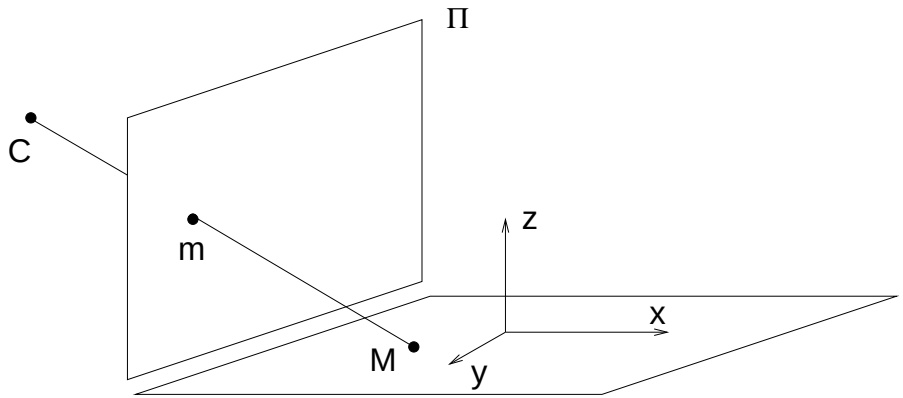


Fig. 12. The map between a world plane Π and a perspective image is a collineation.

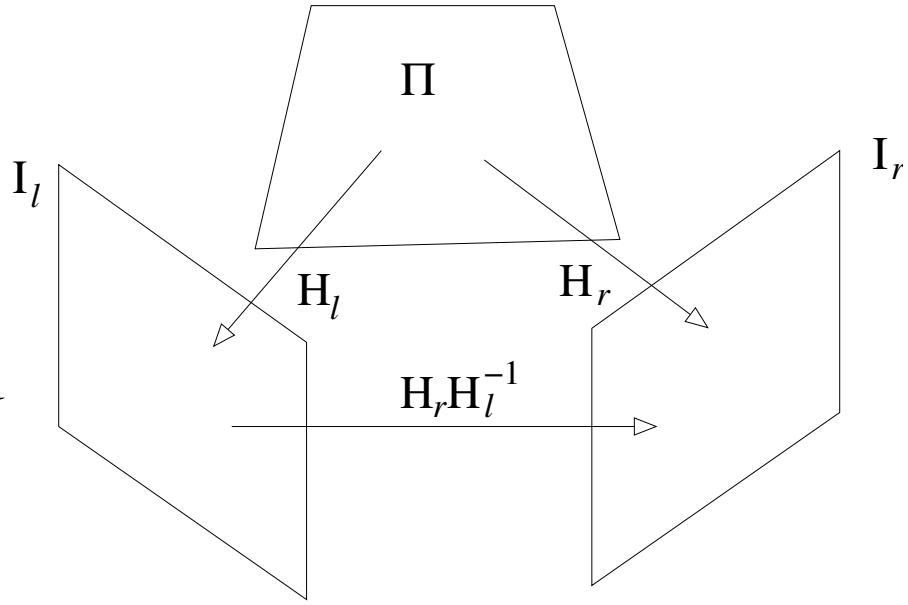


Fig. 13. The plane Π induces a collineation between two views.

Next, we prove that: images of points on a plane are related to corresponding image points in a second view by a *collineation* (or homography) of $\mathbb{P}_2$.

We have one collineation from Π to the left image plane, and another collineation from Π to the right image plane. By composing the inverse of the first with the second, we define a collineation from the image plane of the left camera to the image plane of the right camera.

The plane Π *induces* a collineation H_Π between the views, which transfers points from one view to the other:

$$\mathbf{m}_r \simeq H_\Pi \mathbf{m}_\ell \quad \text{if } \mathbf{M} \in \Pi. \tag{25}$$

where H_Π is a 3×3 non-singular matrix.

3.3.1 Homography induced by a plane

If the 3-D point $\mathbf{M}$ lies on a plane Π with equation $\mathbf{n}^T\mathbf{M} = d$, Eq. (16) can be specialized, obtaining (after elaborating):

$$\frac{\zeta_r}{\zeta_\ell}\mathbf{m}_r = K_r \left(R + \frac{\mathbf{t}\ \mathbf{n}^T}{d} \right) K_\ell^{-1}\mathbf{m}_\ell. \tag{26}$$

Therefore, the collineation induced by Π is given by:

$$H_\Pi = K_r \left(R + \frac{\mathbf{t}\ \mathbf{n}^T}{d} \right) K_\ell^{-1} \tag{27}$$

This is a three-parameter family of collineations, parametrized by $\mathbf{n}/d$.

3.3.2 Infinite homography

The infinite homography H_∞ is the collineation induced by the plane at infinity; it maps vanishing points to vanishing points (a vanishing point is where all the lines that shares the same direction meet).

It can be derived by letting $d \to \infty$ in (26), thereby obtaining:

$$H_\infty = K_r R K_\ell^{-1} \tag{28}$$

The infinity homography does not depend on the translation between views.

In other terms, the vanishing points are fixed under camera translation.

3.3.3 Plane induced parallax

In general, when points are not on the plane, the homography induced by a plane generates a virtual parallax. This gives rise to an alternative representation of the epipolar geometry and scene structure [24].

First, let us rewrite Eq. (16), which links two general conjugate points, as:

$$\frac{\zeta_r}{\zeta_\ell}\mathbf{m}_r = H_\infty \mathbf{m}_\ell + \frac{1}{\zeta_\ell}\mathbf{e}_r, \tag{29}$$

The mapping from one point to its conjugate can be seen as composed by a transfer with the infinity homography $(H_\infty \mathbf{m}_\ell)$ plus a parallax correction term $(\frac{1}{\zeta_\ell}\mathbf{e}_r)$.

Note that if $\mathbf{t} = \mathbf{0}$, then the parallax vanishes. Thus H_∞ not only relates points at infinity when the camera describes a general motion, but it also relates image points of any depth if the camera rotates about its centre.

We want to generalize this equation to any plane. To this end we substitute

$$H_\infty = H_\Pi - K_r \left(\frac{\mathbf{t}\ \mathbf{n}^T}{d} \right) K_\ell^{-1} \tag{30}$$

into Eq. (29), obtaining

$$\frac{\zeta_r}{\zeta_\ell} \mathbf{m}_r = H_\Pi \mathbf{m}_\ell + \gamma \mathbf{e}_r \tag{31}$$

with $\gamma = \left(\frac{a}{d\ \zeta_\ell} \right)$, where a is the distance of $\mathbf{M}$ to the plane Π.

When $\mathbf{M}$ is on the 3-D plane Π, then $\mathbf{m}_r \simeq H_\Pi \mathbf{m}_\ell$. Otherwise there is a residual displacement, called *parallax*, which is proportional to γ and oriented along the epipolar line.

The magnitude parallax depends only on the left view and the plane. It does not depend on the parameters of the right view.

From Eq. (31) we derive $\mathbf{m}_r^T (\mathbf{e}_r \times H_\Pi \mathbf{m}_\ell) = 0$, hence

$$F \simeq [\mathbf{e}_r]_\times H_\Pi \tag{32}$$

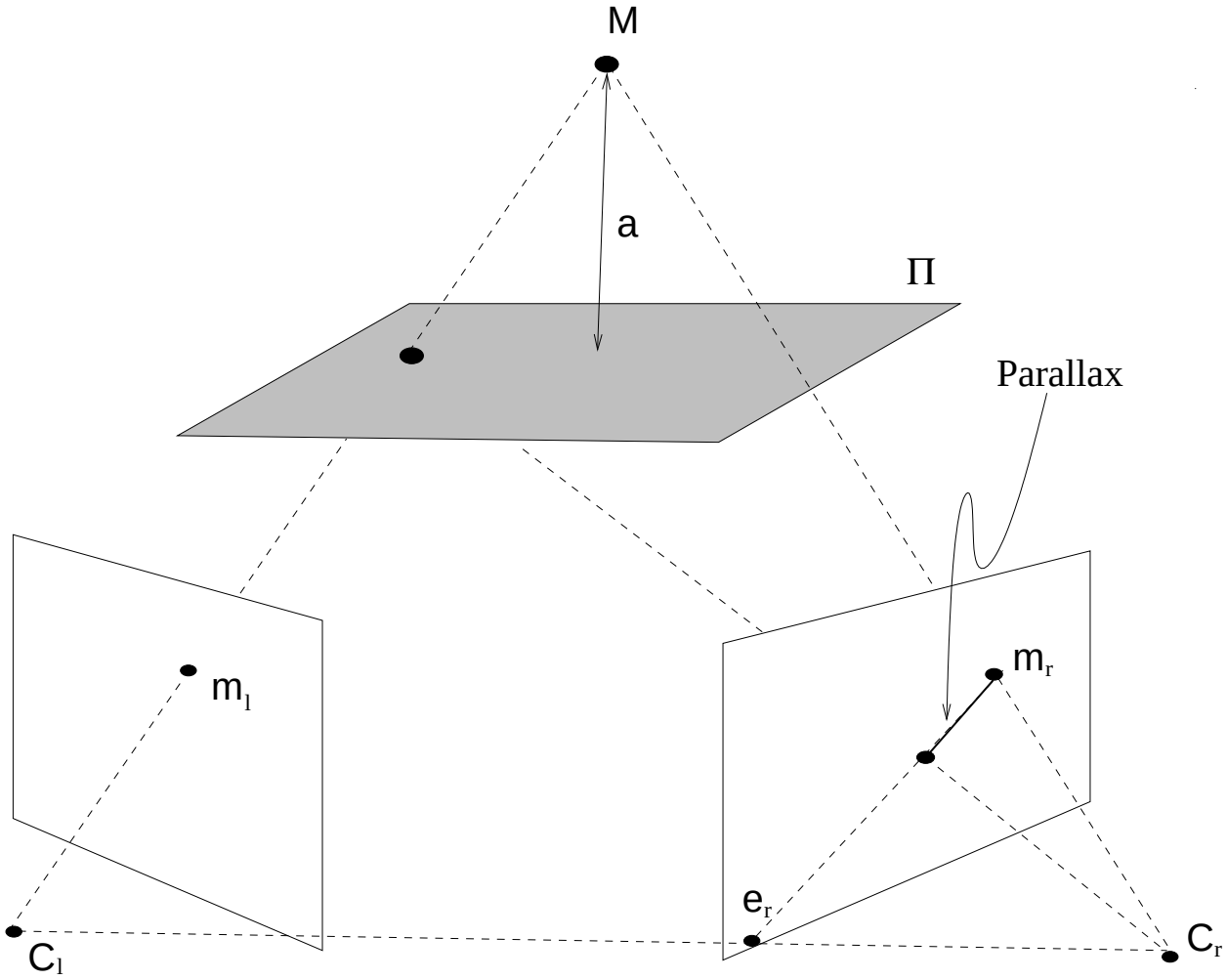


Fig. 14. Plane induced parallax.

Fig. 15. Left and right images. The leftmost image is a superposition of the warped left image and the right image. The reference plane exactly coincide. However, points off the plane (such as the bottle) do not coincide.

3.3.4 Applications

Mosaics. Image mosaicing is the automatic alignment (or registration) of multiple images into larger aggregates [25]. There are two types of mosaics. In both cases, it turns out that images are related by homographies, as we discussed previously.

Planar mosaic: result from the registration of different views of a planar scene.

Panoramic mosaic result from the registration of views taken by a camera rotating around its optical centre (typically panning).

In order to cope with large rotations (> 180 deg), the images are converted to cylindrical (or spherical) coordinates.

Fig. 16. Planar mosaic with components location shown as white outlines.

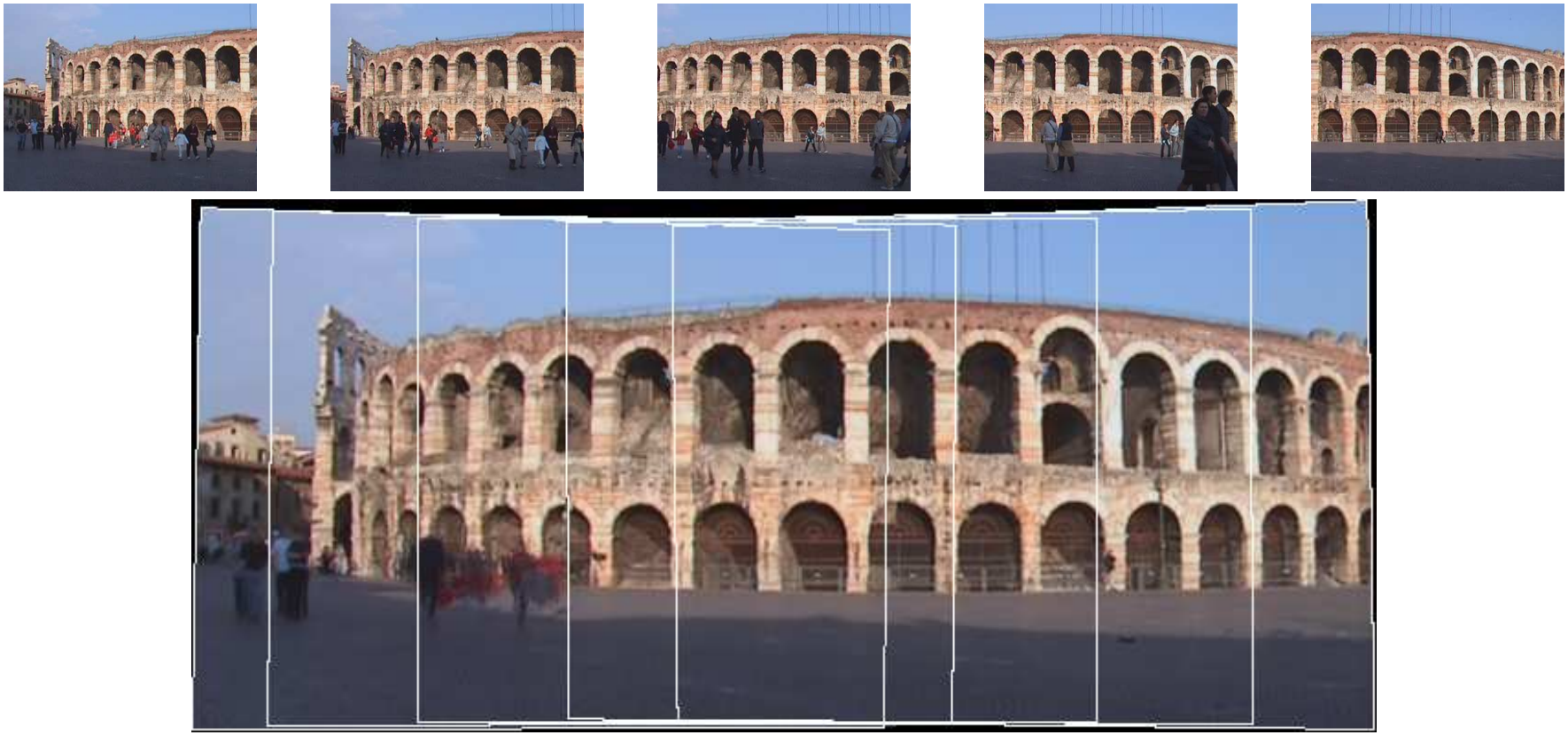


Fig. 17. Selected frames from "Arena" sequence (top) and panoramic mosaic (bottom). Components location shown as white outlines.

Fig. 18. Planar panoramic mosaic (right) and its projection onto a cylinder (left).

Orthogonal rectification. The map between a world plane and its perspective image is an homography. The world-plane to image-plane homography is fully defined by four points of which we know the relative position in the world plane. Once this homography is determined, the image can be back projected (warped) onto the world plane. This is equivalent to synthesize an image as taken from a fronto-parallel view of the plane. This is known as *orthogonal rectification* [15] of a perspective image.

Fig. 19. A perspective image and a ortho-rectified image of the floor plane

Parallax-based transfer Since the relative affine structure is invariant on the choice of the right view, arbitrary virtual "right views" can be synthesized, starting from the the left view and the parallax field [24].

Let us rewrite Eq. (31) without the unknown depths:

$$\mathbf{m}_r \simeq H_\Pi \mathbf{m}_\ell + \gamma \mathbf{e}_r. \tag{33}$$

Given a certain number (> 6) of corresponding pairs $(\mathbf{m}_1^k\,;\,\mathbf{m}_2^k) \quad \forall k = 1, \ldots, m$ the homography H_Π and the epipole $\mathbf{e}_r$ can be easily computed.

Then, the parallax of each point is obtained by solving for γ in (33):

$$\gamma = \frac{(\mathbf{H}_\Pi \mathbf{m}_\ell \times \mathbf{m}_r)^T (\mathbf{m}_r \times \mathbf{e}_r)}{||\mathbf{m}_r \times \mathbf{e}_r||^2} \tag{34}$$

Please note that the epipole and the homography can be computed from images only up to an unknown scale factor. It follows that the magnitude of the parallax as well is known only up to a scale factor.

Once the parallax has been recovered, a point can be forward-mapped from the real (left) view into the virtual one using

$$\mathbf{m}_v \simeq H'_\Pi \mathbf{m}_\ell + \gamma \mathbf{e}_v. \tag{35}$$

Where H'_Π is the homography induced by Π between the left and the virtual image and $\mathbf{e}_v$ is the epipole in the virtual image. They specify the position and orientation of the virtual camera in a projective frame.

This technique is similar to the depth-based warping, but it uses the parallax instead of the depth, therefore the internal camera parameters are not needed.

On the other hand, working in a projective frame, as opposed to a Euclidean frame, is not intuitive.

t	-1.0	-0.5	0
			
t	0.5	1	1.5
			

Fig. 20. Some frames from the "Porta" sequence. The values $t = 0$ and $t = 1$ correspond to the reference images.

In summary...

... we have seen several geometrical ways of performing view synthesis:

- Depth-based warping. Given the depth of a point use Eq. (14) Eq. (16) or to map it to its corresponding point;
- Epipolar transfer. Given correspondences in two views and the epipolar geometry linking them and a third synthetic view, intersect the epipolar lines in the virtual view;
- Disparity-based interpolation. Interpolate disparity to synthesize in-between views.
- Panoramic mosaics. Given several views taken with a rotating camera build a spherical (or cylindrical) mosaic and then generate synthetic views (rotation and zoom only) by grabbing a portion of the mosaic;
- Parallax-based transfer. Given correspondences in two views, compute parallax and use it to a third view;

- Please also note that orthogonal rectification and epipolar rectification produces a virtual rotation of the camera.

We shall now see

- how these view synthesis instances fit in a bigger picture (taxonomy of IBR), and finally
- how correspondences – which are essential to many IBR techniques – are obtained (stereo matching).

4 Image based rendering

Some definitions of IBR:

- Techniques to generate novel views by re-sampling one or more example images, using suitable warping functions.
- Use of photographs to enhance realism in Computer Graphics.
- Use of pre-computed images to speed-up rendering.

4.1 Plenoptic function

The plenoptic function [1], is the 5-dimensional function representing the intensity (or radiance) of the light observed from every position and direction in 3-d space.

$$L = P(\theta, \phi, V_x, V_y, V_z)$$

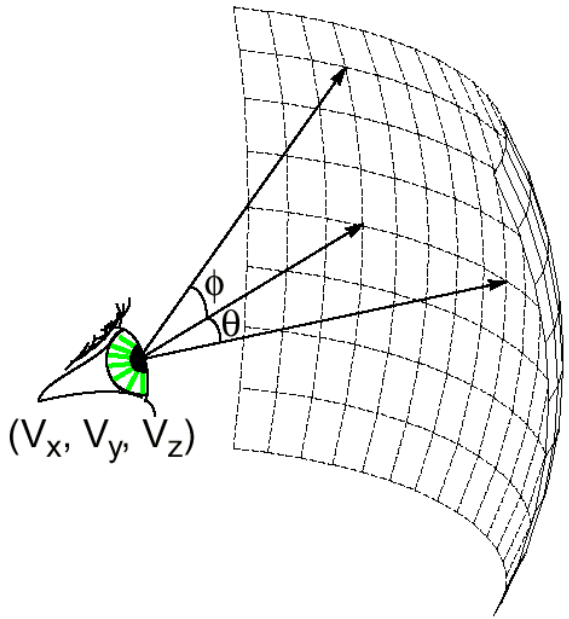


Fig. 21. Plenoptic function (from [19]).

In image-based modelling the aim is to reconstruct the plenoptic function from a set of examples images.

Once the plenoptic function has been reconstructed, images are generated by indexing the appropriate light rays.

If the plenoptic function is only constructed for a single point in space (V_x, V_y, V_z) then its dimensionality is reduced from 5 to 2.

This is the principle used in reflection mapping (also known as environment mapping) where the view of the environment from a fixed position is represented by a 2-dimensional texture map.

A spherical panorama can be viewed as a sample the plenoptic function at a fixed position.

Cylindrical panorama and conventional perspective images can be modelled as projection from the sphere onto a cilynder or a plane respectively.

Sampling the plenoptic funcion with real sensors introduces discretization at two levels:

1. Angular sampling (θ, ϕ) of a single plenoptic sample due to the finite pixel resolution
2. Spatial sampling (V_x, V_y, V_z) due to the discrete positioning of the sensor.

It is necessary to obey the sampling theorem to avoid aliasing

Angular sampling is usually not a problem, but patial sampling may be critical. Therefore IBR systems have to distinguish between dense and sparse sampling.

Dense sampling obey the the sampling theorem. [4] give bounds for the sampling density as a function of the depth variation in the scene.

Sparse sampling violate the sampling theorem, hence additional information – usually in the form of depth, parallax or disparity – it is necessary to render correct views, or to *compensate*.

4.2 Taxonomy

The categorization of IBR systems follows [12], and its organized along three axes:

Geometry: the "amount" of geometry needed for compensation, ranging from systems with no compensation at all (dense sampling) to full 3D information.

Samples: the plenoptic samples density (dense vs sparse) and the spatial samples arrangement (unstructured, structured 1D, structured 2D).

Motion: amount of freedom in selecting the virtual viewpoint. Three categories: predetermined discrete positions, constrained in some way, unrestricted.

In the following we will briefly survey IBR methods following the "Geometry" axis.

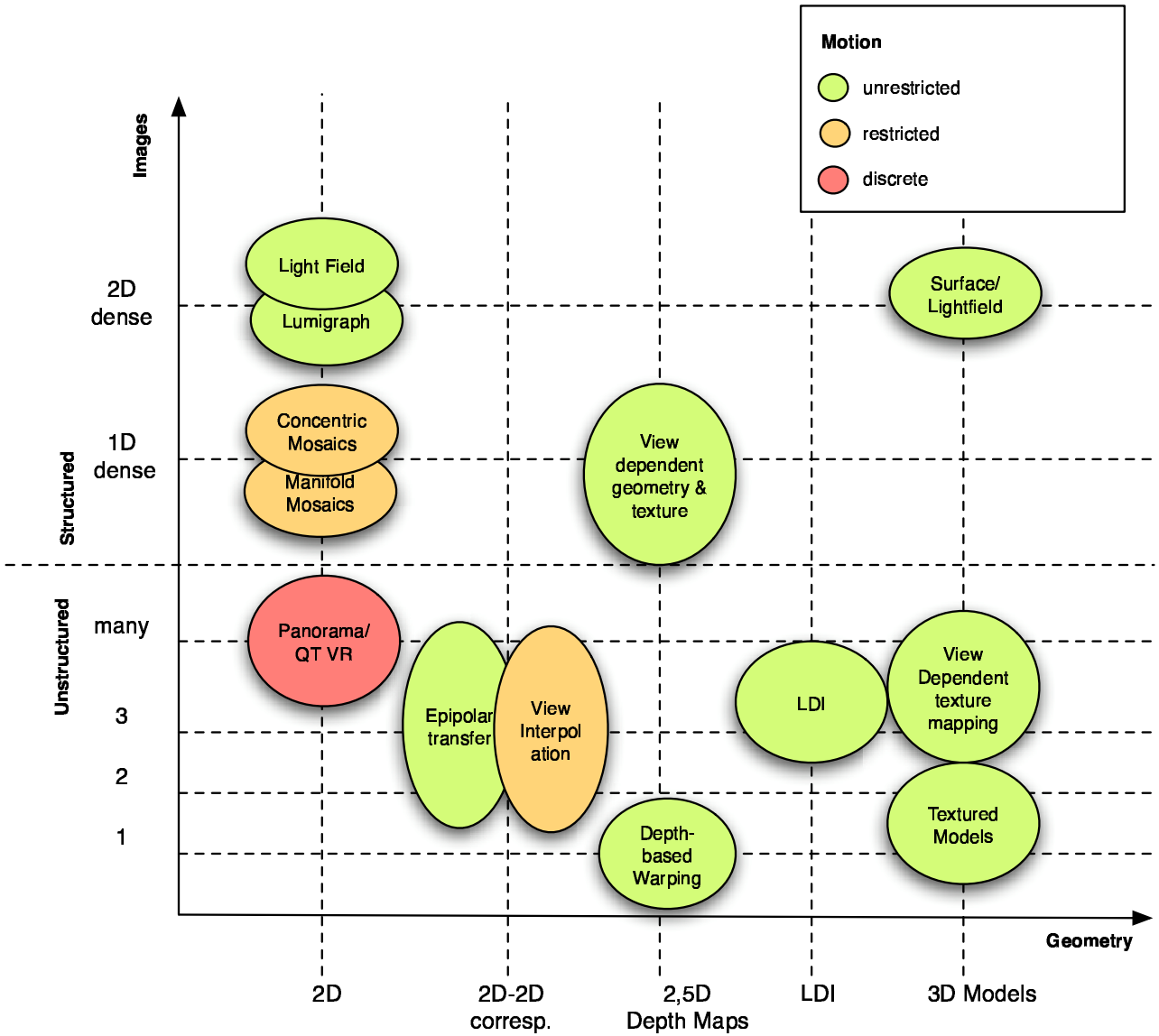


Fig. 22. Taxonomy of IBR methods. The "Motion" axis is colour coded. Adapted from [12].

4.3 Rendering without geometry

Methods within this class use no geomety information at all. Since no compensation is possible, the sampling must be either very dense, or the possible motion is restricted.

- Movie Map
- Panoramic Mosaic
- Concentric Mosaic
- Light Field/Lumigraph

4.3.1 Movie Map

Movie Map [16] was the earliest system to obtain restricted interactive look-around capabilities, based on dense sampling with four orthogonal cameras mounted on a car.

The streams were captured on video disks for interactive playback, allowing the user to select the route at street intersections.

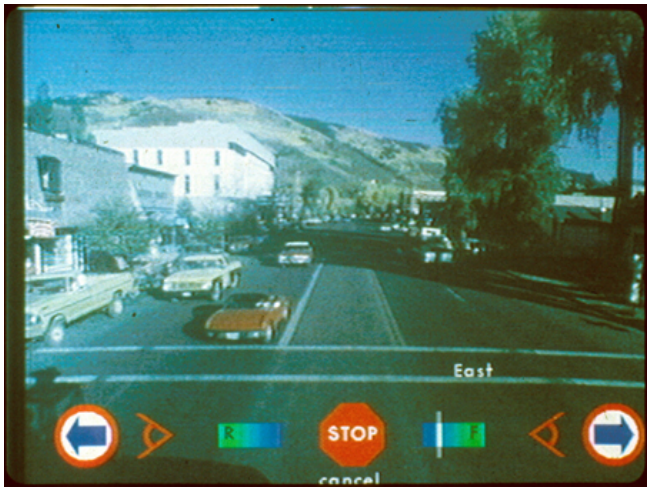


Fig. 23. A screenshot from the Aspen Movie Map. (http://en.wikipedia.org/wiki/Image:Aspen.jpg)

4.3.2 Panoramic mosaics

A camera is rotated (on a tripod) at a fixed position and all images are stitched together to form a cylindrical or spherical panoramic view of the scene (mosaic). Any image of the scene from the capturing position can then be rendered using the inverse mapping.

This is the principle used in **QuickTimeVR** which enables an environment to be viewed in any direction from a discrete set fixed positions [5].

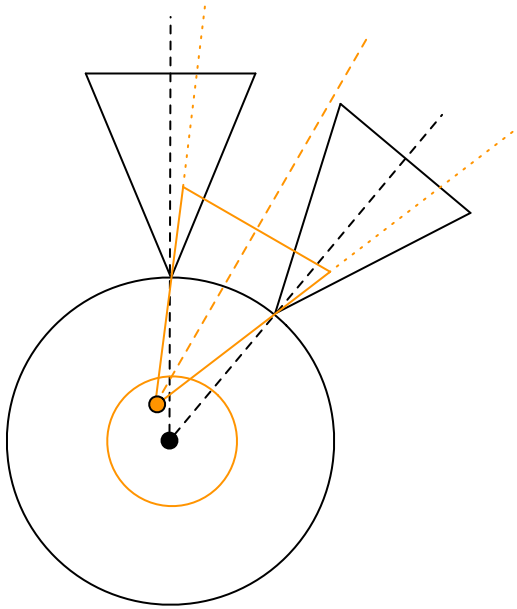
For videocommunications, panoramic images can serve to create and visualize static background, but due to the missing parallax they cannot be used to generate novel views of nearby objects.

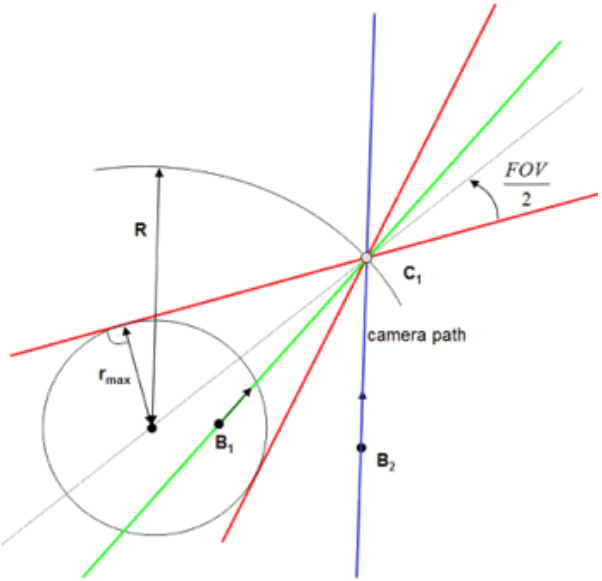
4.3.3 Concentric mosaics

Camera motion on a circle (this is *not* a pure rotation).

Novel views are rendered by assembling vertical line (slits) coming from the recorded images (interpolating when needed).

Viepoint can move inside a circle with the same centre as the camera path and a radius $r_{max} = R\sin(FOV/2)$





4.3.4 Light Field/Lumigraph.

Trade sample density for geometric complexity.

If one considers only the subset of light rays leaving the convex hull of a bounded object, the fact the radiance along any ray remains constant allows to reduce the 5-dimensional plenoptic function to a 4-dimensional function.

Two similar methods (Lumigraph [7] and Light Field [14]) for representing this 4-d function and for constructing the function from example images have been proposed.

Both of these methods allow scenes and objects to be rendered very efficiently from novel viewpoints but even the 4-d functions requires very large amounts of storage.

The principle of the **Lightfield** can be briefly addressed as follows. By placing the object in its bounding box which is surrounded by another larger box, the Lightfield indexes all possible light rays entering and exiting one of the six parallel planes of the double bounding boxes.

The Lightfield data is thus composed of six 4D functions, where the plane of the inner box is indexed with coordinate (u, v) and that of the outer box with coordinate (s, t).

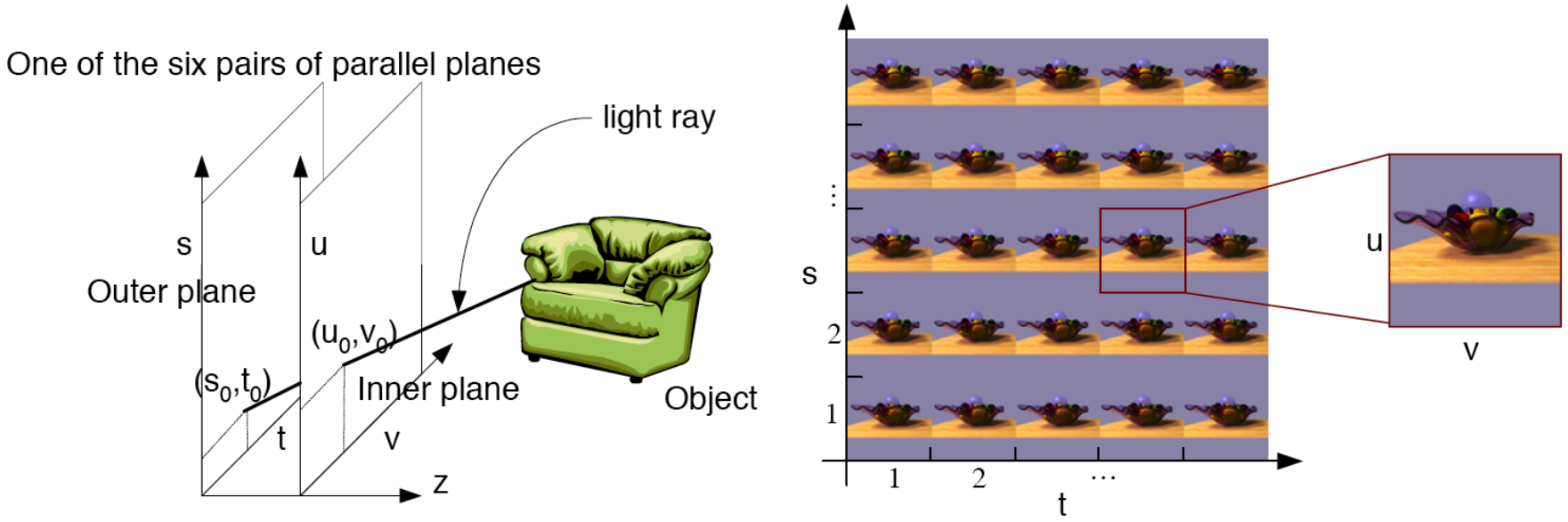


Fig. 24. From [26].

Alternatively, the Lightfield can be considered as six twodimensional image arrays, with all the light rays coming from a fixed (s, t) coordinate forming one image. This is equivalent to setting a camera at each coordinate (s, t) and taking a picture of the object with the imaging plane being the (u, v) plane.

To create a new view of the object, we just split the view into its light rays, which are then calculated by interpolating existing nearby light rays in the image arrays.

4.4 Rendering with geometry compensation

If dense sampling is not viable and a free view point is desired, the aliasing effect must be compensated using additional information such as depth or disparity.

- Disparity-based interpolation (calibrated, viewpoint in-between)
- Image transfer (uncalibrated, free viewpoint)
- Depth-based warping (calibrated, free viewpoint)

4.4.1 Disparity-based interpolation

The relatively small number of reference images required by Disparity-based interpolation together with the absence of explicit geometry are their principal characteristics. However, the view point is constrained to lie in-between the original views. All such techniques rely on establishing dense correspondences between the source views and on the knowledge of the internal parameters of the camera.

Chen and Williams [6] introduced **view interpolation**. They arranged the set of original images in a graph. The nodes of the graph are the images. Each arc in the graph represents a correspondence mapping, which is bi-directional, and two maps are associated with each arc. The user can then move continuously around the space represented by the images by interpolating from one node to the next.

To synthesize views in-between a pair of images the displacement vectors are linearly interpolated and the pixels in the reference images are moved by the interpolated vector to their destination. However, linear interpolation only yields a valid reprojection if the source and the new image planes are parallel.

To overcome this limitation Seitz et al. [22] proposed a three step algorithm called **view morphing**. Initially they pre-warp (rectify) the source images so that their image planes are aligned. By linearly interpolating positions and colours on the reference images a new intermediate view along the line segment connecting the two camera centres is generated. A post-warping (de-rectification) process finally transforms the image plane of the new view to its desired position and orientation.

Generation of new views from perspective source images has been extended to cylindrical panoramic images. **McMillan and Bishop** [19] describe a *cylindrical epipolar geometry* that determines the possible positions of a point given its location in some other cylinder.

This constraint is used to establish dense correspondences between cylindrical reference pairs. A warp function subsequently combines the transformation of the disparity values from the known reference pair to the new cylinder and its reprojection as a planar image for viewing.

4.4.2 Image transfer methods

Image transfer methods are based on the observation that certain relationships exist between the positions of pixels representing the same points in space observed from different viewpoints.

Uncalibrated transfer techniques utilise image to image constrains such as the fundamental matrix and the trifocal tensor – without knowing the camera parameters – to transfer image pixels from a small number of reference images to a virtual image, with unconstrained view point.

For example, we have seen that, thanks to epiplar transfer [13], any third view can be predicted given only the pixel correspondences and the epipolar geometry for the two example views.

A more stable geometric constraint than the epipolar geometry is the trilinear tensor (see [8] for example). The use of trilinearities as a warping function from model views to novel synthesised images has been presented in [2]. A seed tensor is computed from three reference images. For every new view with known camera motion parameters relative to one of the reference images, a tensor is computed between the remaining two images and this new view. The tensor is subsequently used to render the image.

Another way of linking corresponding points is the plane+parallax [24, 10] paradigm, that we discussed in some detail.

4.4.3 Depth-based warping

Given a depth map (range image) and the camera parameters, it is easy to obtain the position of each pixel in the synthetic view, i.e., computing the disparity (or parallax) between the real view and the synthetic views.

Depth-based warping can extrapolate from just one view, but in this case the information for filling holes is missing (occlusions). This is not the case when interpolating views.

Layered Depth Images [23] solve the occlusion problem by associating to each pixel many depths, namely the depth of each surface (layer) that a ray through the pixel would intersect. Rendering is done back-to-front, by processing one layer at a time, following McMillan's ordering algorithm [18].

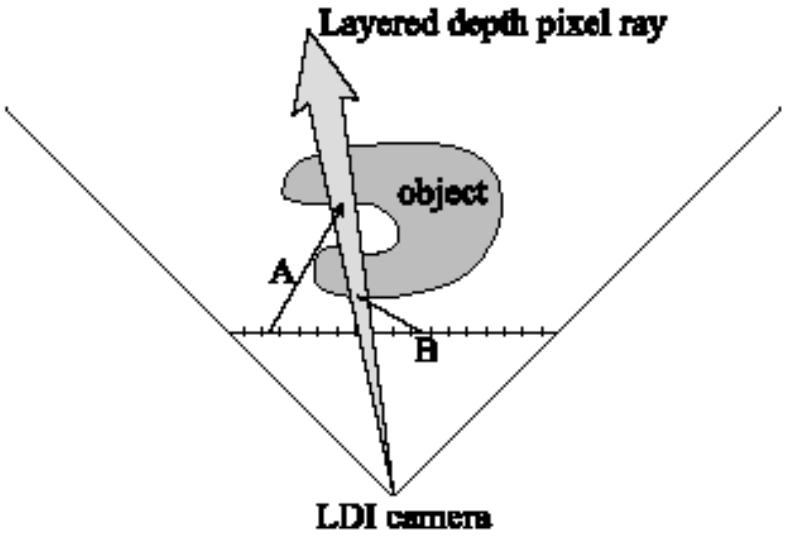


Fig. 25. Intersections from sampling rays A and B are added to the same layered depth pixel (from [23]) .

4.5 Rendering from (approximate) geometry

This is the class of methods closer to the tradition model-based rendering, because a static 3D model is created, albeit approximate.

View-Dependent Texture Mapping. 3D surface model available (low detail). The same polygon is seen from different positions. At rendering-time, the real view nearest to the virtual view is texture-mapped onto the polygon.

Surface Lightfield. The inner box (s, t) is replaced by the parametric equation of a surface patch.

References

[1] E. Adelson and J. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. A. Movshon, editors, *Computation Models of Visual Processing*, chapter 1, pages 3–20. MIT Press, 1991.

[2] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, 1997.

[3] Myron Z. Brown, Darius Burschka, and Gregory D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):933–1008, August 2003.

[4] Jin-Xiang Chai, Shing-Chow Chan, Heung-Yeung Shum, and Xin Tong. Plenoptic sampling. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[5] Shenchang Eric Chen. Quicktime VR - an image-based approach to virtual environment navigation. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 29–38. ACM SIGGRAPH, Addison Wesley, August 1995. Los Angeles, California, 06-11 August 1995.

[6] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 279–288, August 1993.

[7] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 43–54. ACM SIGGRAPH, Addison Wesley, August 1996.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.

[9] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):1–16, November 1999.

[10] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *Proceedings of the European Conference on Computer Vision*, pages 17–30, 1996.

[11] F. Isgrò and E. Trucco. Projective rectification without epipolar geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:94–99, Fort Collins, CO, June 23-25 1999.

[12] R. Koch and J.-F. Evers-Senne. View syntheis and rendsering methods. In O. Schreer, P. Kauff, and T. Sikora, editors, *3D Videocommunication. Algorithms, concepts and real-time systems in human centered communication*, chapter 69. John Wiley & Sons, 2005. ISBN: 0-470-02271-X.

[13] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images and foundamental matrices. Technical Report 2205, INRIA, Institut National de Recherche en Informatique et an Automatique, February 1994.

[14] Marc Levoy and Pat Hanrahan. Light field rendering. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 31–42. ACM SIGGRAPH, Addison Wesley, August 1996.

[15] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, 1998.

[16] A. Lippman. Movie-maps: An application of the optical videodisc to computer graphics. *Computer Graphics (SIGGRAPH '80 Proceedings)*, 14(3):32–42, July 1980.

[17] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:125–131, Fort Collins, CO, June 23-25 1999.

[18] L. McMillan and G. Bishop. Head-tracked stereo display using image warping. In *Stereoscopic Displays and Virtual Reality Systems II*, number 2409 in SPIE Proceedings, pages 21–30, San Jose, CA, 1995.

[19] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH 95 Conference Proceedings*, pages 39–46, August 1995.

[20] L. McMillann and S. Gortler. Image-based rendering:a new interface between computer vision and computer graphics. *SIGGRAPH Comput. Graph.*, 33(4), November 1999.

[21] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.

[22] Steven M. Seitz and Charles R. Dyer. View morphing: Synthesizing 3D metamorphoses using image transforms. In *SIGGRAPH 96 Conference Proceedings*, pages 21–30, August 1996.

[23] J. Shade, S. Gortler, L. He, and R. Szeliski. Layered depth images. In *SIGGRAPH 98 Conference Proceedings*, pages 231–242, 1998.

[24] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, September 1996.

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

[26] C. Zhang and T. Chen. Generalized plenoptic sampling. Technical report, AMP01-06, Carnegie Mellon, 2001.