Multiple-View Scenes: Reconstruction and Virtual Views

Javier Cruz Mota * Michel Bierlaire *
Jean-Philippe Thiran †

July 10, 2008

Report TRANSP-OR 080710
Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

*Transp-OR, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, javier.cruz@epfl.ch, michel.bierlaire@epfl.ch
†SPL5, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, jp.thiran@epfl.ch

Abstract

The problem of generating a virtual view of a scene, i.e. a view from a point where there is not a physical camera to capture the scene, has received recently a lot of attention from the computer vision community. This is probably due to the increase of the computational power of computers, which allows to deal with multiple view systems (systems composed of multiple cameras) efficiently.

In this document, an introduction to virtual view generation techniques is presented. In a first part, geometric constraints of multiple view systems are presented. This geometric constraints allow to reconstruct the $3D$ information of the observed scene, and therefore they allow to generate virtual views from everywhere (although problems with occlusions will arise). In the second part of the document, the state-of-the-art on Image Based Rendering (IBR) techniques is presented. IBR techniques allow to generate virtual views from some constrained regions of the space without requiring a complete $3D$ reconstruction of the scene. To finish, some concussions are given.

1 Introduction

In the last years, multiple view systems have received a lot of attention of the computer vision research community, probably due to the fact that a future 3DTV system begins to be seen as feasible [AYG+07, BWS+07, ATFC07, SMS+07]. This is very important because since the invention of the television, commercially available since the late 1930s, there has not been any significant change in the way it is seen. The quality of the images has been continuously incremented, going from the very low resolution gray-scale images to the nowadays high definition colour images, but the way they are seen, watching non-interactively to a screen, has not substantially changed. With the introduction of the digital television and DVD's, it is sometimes possible to choose between a few different cameras, adding some interactivity from the user, but it is still far away from an interactive system like a complete 3DTV system, where the user could watch the scene freely from any point and in addition with 3D sensation. Probably, a complete 3DTV system is still far away from now because it would need a change of the user's displays, and in addition the 3D displays available now are still in a very embryonic stage, but a free viewpoint television system seems to be feasible in a short time, since it could be shown in current displays [Tan06]. But 3DTV is not the only field of application of multiple view systems, they have a lot of other applications such as robust tracking [YZC04], depth estimation [WK04], virtual reality and immersive environments [YNK+05] or immersive teleconferencing environments [WW00].

This document covers 2 subjects quite related, multiple view systems and virtual viewpoint generation, and it is organised as follows. In a first part (sections 3, 4, 5, 6, 7 and 8) multiple view systems of planar cameras are introduced, and the geometrical objects that manage the constraints between images taken from different cameras are presented. These geometric objects and algorithms allow to *Reconstruct* a scene, i.e. to obtain the 3D information of the scene and the transformations between images from different cameras. This reconstruction of a scene allows to compute occlusions and depth of points seen from an arbitrary given point, and thus, the reconstruction of a scene allows to generate virtual viewpoints

from anywhere. But it is not necessary to compute a complete reconstruction of a scene to generate a virtual viewpoint, and in section 9, methods to generate virtual viewpoints, most commonly known as *Image-Based Rendering* techniques, will be presented. To finish, some conclusions and future work are give.

2 Notation

In the following paragraphs, the next notation has been adopted:

- x denotes a scalar value
- $\mathbf{x}$ or $\mathbf{X}$ denote a vector
- Points in $3D$ space are denoted by capital letters, like for example $\mathbf{X}$, and image points are denoted by letters in lower-case, like for example $\mathbf{x}$
- X denotes a matrix
- $(A \mid B)$ denotes the concatenation of the matrices A and B
- $\mathbf{x}, \mathbf{x}', \mathbf{x}'', \mathbf{x}''' \ldots$ denote the image into several cameras of the same point $\mathbf{X}$ in $3D$ space

3 Multiple View Systems

Intuitively, it is clear that if "something" is "seen" from different places, additional information is obtained if all this captured information is combined appropriately. The problem is basically what does this "appropriately" mean? When dealing with radiofrequency signals, for example, the phase information and more concretely the difference of phase between a signal received by different antennas, has allowed to develop multiple and very efficient techniques of *Array Processing* to extract a lot of information from one or several received signals. See for example [Sch79], where the *Multiple Signal Composition* (MUSIC) algorithm is presented, or

[RK89] for the *Estimation of Signal Parameters Via Rotational Invariance Techniques* (ESPRIT) algorithm. But what about image signals? In this case, the phase information is not captured, and the problem is not as "simple". While in radiofrequency, the common situation is to be in *distant field* and then the received wave can be assumed to be a planar wave, when dealing with images a displacement of a camera can change completely the received signal. These reasons, together with the short wavelength of light signals, are basically why in multiple view systems instead of exploiting the signal nature, the geometry of the multi-camera system is usually used.

In the following sections, the geometry of systems with 2, 3, 4 and n $(n > 4)$ planar cameras is analysed, obtaining the state-of-the-art multiple view geometric objects for these systems. Even though a system with only one camera is not a multiple view system, it is also analysed, since the description of how a camera captures a scene is used later on the other sections and represents a basic theoretical background. Along these sections, classical notation present in the literature has been used, that is vector and matrix notation for 1 and 2 views and tensorial notation for more than 2 views. However, all this can be homogenised using only tensorial notation, see chapter 17 in [HZ03] or [Hey98] for further details.

4 Single View: Camera Geometry

The function of a camera is to map the 3D world into a 2D image, in general by means of a central projection. All cameras modelling a central projection are specialisations of the general projective camera (see figure 1), whose characteristics can be studied using tools from the projective geometry, since the real world can be seen as the projective space of the image plane. Given a point $\mathbf{x} = (x, y)$ in an image, the set of points that can generate the image point $\mathbf{x}$ is, considering the coordinate system situated in the centre of projection of the camera, $k \cdot (x, y, f)^\top$, $\forall k \in \mathbb{R}$, $k \geq 1$, where f is the focal length of the camera. The set of points $\{k \cdot (x, y, f)^\top \mid k \in \mathbb{R},\ k \geq 1\}$ is called the ray of $\mathbf{x}$. Sometimes, for

convenience, the *Normalised Coordinates* are considered, which consist of normalising the focal length to 1.

Into the family of general projective cameras two subfamilies can be distinguished, the *finite cameras*, those with a "finite" centre of projection, and the *cameras at infinity*, those with a centre of projection "at infinity". Note that, in projective geometry, a point lies at infinity (or it is an ideal point) if in homogeneous coordinates its last coordinate element is 0.

For a general projective camera, given a point $\mathbf{X}$ in homogeneous coordinates, i.e. $\mathbf{X} = (X, Y, Z, 1)^\top$, the image point $\mathbf{x} = (x, y, 1)^\top$, also in homogeneous coordinates, is given by

$$\lambda \mathbf{x} = P\mathbf{X} \tag{1}$$

where P is the projection matrix and λ is a scalar factor. For the sake of conciseness, the λ scalar factor will be ignored from now on if its omission does not change the result, using then the expression

$$\mathbf{x} = P\mathbf{X} \tag{2}$$

Notice that the world coordinate system is not necessarily the same than the camera coordinate system, in which case the matrix P also performs the corresponding rotation and translation. In addition, the camera coordinate system may need a scaling, a skew correction and an offset for the principal point, operations that are usually expressed in a matrix called the *Camera Calibration Matrix K*, which is also part of the projection matrix. The general form of P for a general projective camera is a 3×4 matrix of rank 3

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \tag{3}$$

It has 11 degrees of freedom, and the restriction of rank 3 arise because otherwise the mapping will be to a line or a point and not to the hole plane. Using the projection matrix P, some important definitions can be

done. For these definitions, the projection matrix is considered, without losing of generality, of the form

$$P = (M \mid \mathbf{p}_4) \tag{4}$$

Camera centre: The camera centre is the 1-dimensional right null-space $\mathbf{C}$ of P, i.e. $P\mathbf{C} = \mathbf{0}$. In a finite camera, M is not singular and $\mathbf{C} = (-M^{-1}\mathbf{p}_4 \mid 1)^\top$. In a camera at infinity, M is singular and $\mathbf{C} = (\mathbf{d}^\top \mid 0)^\top$, where $\mathbf{d}$ satisfies $M\mathbf{d} = \mathbf{0}$.

Column points: For i=1,2,3, the column vectors $\mathbf{p}_i$ are the vanishing points in the image of the directions of the world axes X, Y and Z. Column $\mathbf{p}_4$ is the image of the coordinate world origin.

Principal plane: The principal plane of the camera is $\mathbf{P}^3$, the last row of P.

Axis planes: The planes $\mathbf{P}^1$ and $\mathbf{P}^2$, that correspond to the first and the second row of P, are the planes in space that contain the camera centre and the image lines $x = 0$ and $y = 0$ respectively.

Principal point: The principal point is the image point $\mathbf{x}_0 = M\mathbf{m}^3$, where $\mathbf{m}^{3\top}$ is the third row of M.

Principal ray: The principal ray or axis of the camera is the ray passing through the camera centre $\mathbf{C}$ and direction vector $\mathbf{m}^{3\top}$.

4.1 Finite Cameras

As said before, the finite cameras are those with the centre of projection situated in a finite point. A general projection matrix for these cameras can be expressed as

$$P = KR(I \mid -\widetilde{\mathbf{C}}) \tag{5}$$

where R is a 3×3 rotation matrix representing the orientation of the camera coordinate frame, I is the 3×3 identity matrix, $\widetilde{\mathbf{C}}$ are the coordinates

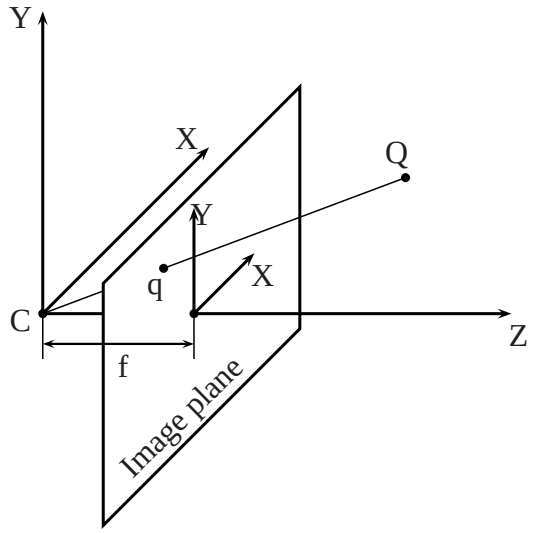


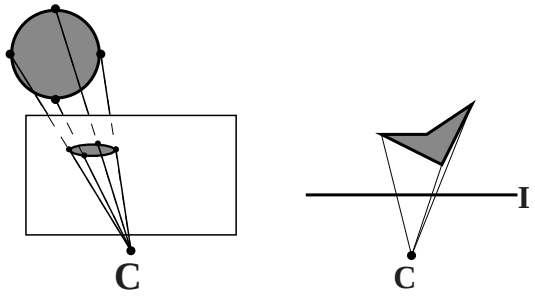
Figure 1: Central projective camera

of the camera centre in the world coordinate system, and K is the camera calibration matrix and it has the form

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

with $\alpha_x = fm_x$ and $\alpha_y = fm_y$ representing the focal length of the camera in terms of pixel dimensions in the x and y directions respectively (m_x and m_y are the number of pixels per distance unit in image coordinates in the x and y directions respectively), s is the skew parameter and (x_0, y_0) is the principal point in terms of pixel dimensions. A projection matrix following the expression 5 will be called from now on a full perspective projection matrix.

The problem when using a *finite projective camera* model is that the map from 3D to 2D is a nonlinear mapping, which can easily make vision problems ill-conditioned, specially when perspective effects are small. For this reason, approximations of the finite projective camera intended



(a) A circle projected into an ellipse

(b) Size of sides is not preserved, nor their relations

Figure 2: Examples of projective deformations

to remove this nonlinearity are often used.

4.2 Infinite Cameras

As commented before, *Infinite Cameras* or *Cameras at Infinity* are cameras with their centre on the plane at infinity, i.e. with M from equation 4 singular. The camera centre can be found exactly as with finite cameras, solving $P\mathbf{C} = \mathbf{0}$. Cameras at infinity can be subdivided into *Affine Cameras* and *Non-Affine Cameras*. Non-affine cameras are cameras with its centre on the plane at infinity but without the hole principal plane being the plane at infinity. This kind of cameras has strange properties, as for example sending, in general, points on the plane at infinity to points not at infinity and viceversa. Non-affine cameras are not widely used and they will not be discussed in the present document.

4.2.1 Affine Cameras

By definition, and affine camera is a camera with a projection matrix P where the last row is $(0,0,0,1)$, or $(0,0,0,k)$, $k \in \mathbb{R}^+$ in general. This means that the principal plane of the camera is the plane at infinity. The general form of the projection matrix of an affine camera is

$$P_A = \begin{pmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

This matrix has 8 degrees of freedom and its sole restriction is that $M_{2\times 3}$, the 2×3 matrix on the top-left, has rank 2. It can be seen also as the concatenation of three transformations: an affine transformation of $3D$ space, an orthographic (or parallel) projection from $3D$ space to an image and an affine transformation of the image

$$P_A = (3 \times 3 \text{ affine transform}) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (4 \times 4 \text{ affine transform}) \tag{8}$$

If we consider $\widetilde{x}_{\text{proj}}$ and $\widetilde{x}_{\text{aff}}$ the image coordinates of a point $\mathbf{X}$ obtained using the full perspective projection matrix and the affine approximation respectively, it can be deduced that

$$\tilde{x}_{\text{aff}} - \widetilde{x}_{\text{proj}} = \frac{\Delta}{d_0}(\widetilde{x}_{\text{proj}} - \widetilde{x}_0) \tag{9}$$

where Δ is the depth of the point with respect to the plane through the world origin and perpendicular to the principal ray, which is at a distance d_0 from the camera centre. Expression 9 means that the affine approximation is good when the depth relief is small compared with the average depth, and the distance from the point to the principal ray is small.

There are 2 main differences between the affine projection and the full perspective projection. The first one is that the canonical projection matrix $(I \mid 0)$ is replaced by the parallel projection matrix $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, and the second one is that the principal point is not defined.

Several special cases of the affine camera are of special relevance and are widely used. The most important and known ones, the *Orthographic Projection*, the *Weak Perspective Projection* and the *Paraperspective Projection* are

introduced in the following paragraphs.

Orthographic Projection

The orthographic projection is the simplest affine projection, it consists of completely ignoring the depth dimension, projecting the objects perpendicularly to the *Image plane*. This projection presents basically two main problems, the "distance effect" and the "position effect". The *distance effect* is the effect caused by orthographic projection which makes that two identical objects have the same image even if one is further to the camera than the other. The *position effect* causes the same deformation than the *distance effect* but when one object is more distant from the optical axis than the other. See figure 3 for an example of orthographic projection and its problems, and figure 6 for an example of the error introduced by this projection with respect to the full perspective projection. The projection matrix for the orthographic projection in normalised coordinates (i.e. $f = 1$) can be expressed as

$$P_{op} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{10}$$

where R and $\mathbf{t}$ are the rotation and translation that move the world coordinate system to the camera coordinate system.

Weak Perspective Projection

The orthographic projection approximation is too strong in general, for this reason, a more feasible approximation is the *Weak Perspective* projection, especially when the object is small compared to its distance to the camera. The weak perspective projection considers a common depth Z_c for an object and, firstly, the object is projected orthographically to that plane $Z = Z_c$ and secondly from that plane to the image plane, but using now in this second step the full perspective projection. In general, the

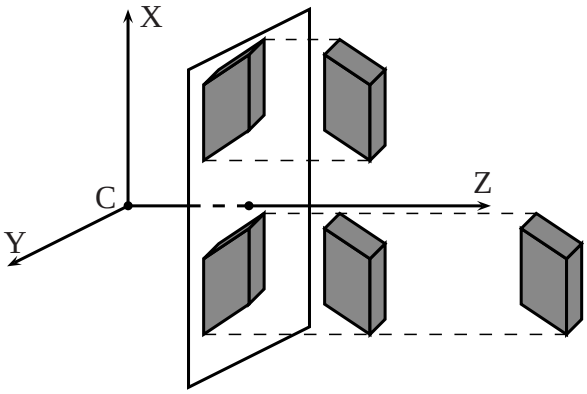


Figure 3: Orthographic projection and its problems

common depth Z_c for an object is the depth of its centroid (see figure 4 for an example). The projection matrix for the weak perspective projection in normalised coordinates is

$$P_{wp} = \begin{pmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_c \end{pmatrix} \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{11}$$

where a hypothetical different scale in x and y image coordinates can be captured by α_x and α_y. As it can be easily seen, when Z_c is equal to 1, the weak perspective projection is the orthographic projection (with a different scale in x and y image coordinates if $\alpha_x \neq \alpha_y$). In figure 6 the error introduced by the weak perspective projection with respect to the full perspective projection and other affine projections can be seen. As noticed, if the value of Z_c is accurately chosen, the error can be very small.

Paraperspective Projection

The main drawback of the weak perspective projection is the error introduced in the projection of objects that are far from the optical axis (big X/Z and/or Y/Z). The *Paraperspective* projection tries to minimise this error projecting into the average depth plane with rays that are parallel to the central projecting ray, i.e. parallel to $\overline{CG}$, where $G = (X_c, Y_c, Z_c)^\top$

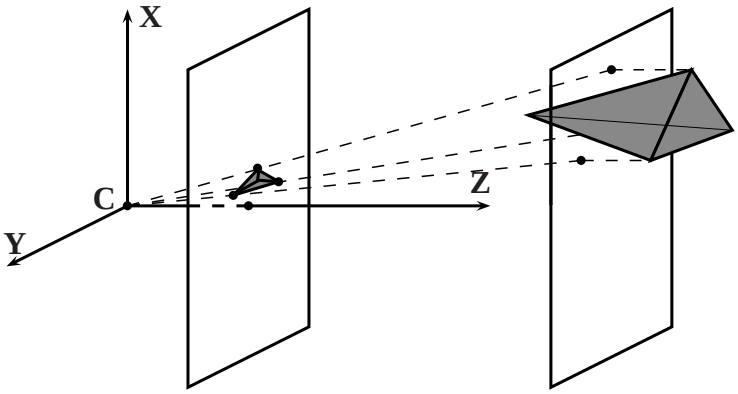


Figure 4: Example of weak perspective projection

is the centroid of the object (see figure 5 for an example and figure 6 for a comparison between the errors of each perspective approximation with respect to the full perspective projection). The projection matrix for the paraperspective projection under normalised coordinates is

$$P_{pp} = \begin{pmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -X_c/Z_c & X_c \\ 0 & 1 & -Y_c/Z_c & Y_c \\ 0 & 0 & 0 & Z_c \end{pmatrix} \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{12}$$

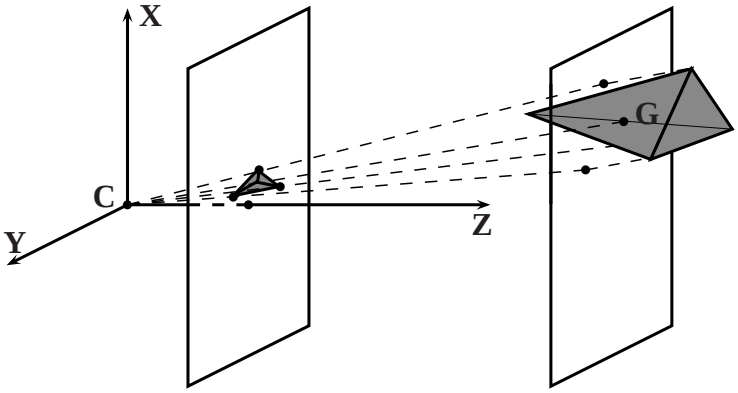


Figure 5: Example of paraperspective projection

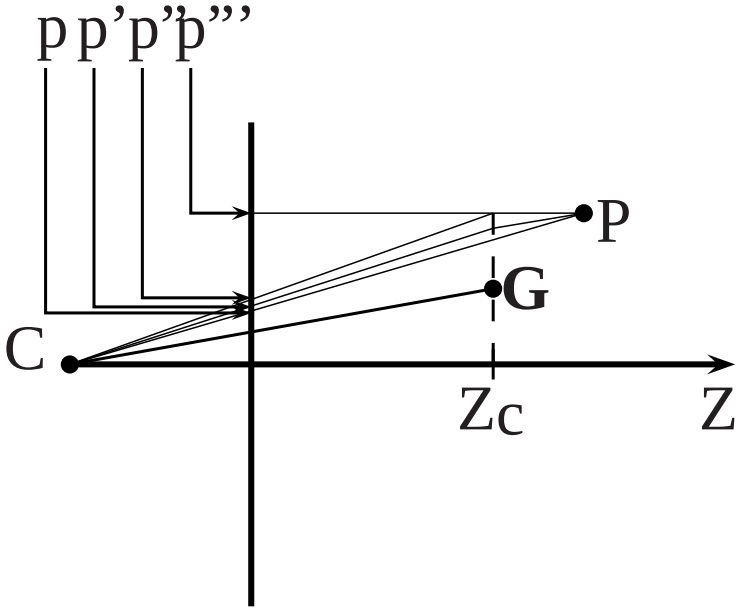


Figure 6: Errors produced in the projection of point P to the image plane by the orthographic projection p''', by the weak perspective projection p'' and by the paraperspective projection p' with respect to the full perspective projection p

5 2-Views: Epipolar Geometry

Epipolar geometry is the most commonly used geometric link between two views of the same scene. The epipolar geometry is the geometry of the intersection of the image planes of each camera with the pencil of planes that have the line linking the camera centres, or baseline, as one of its axis. It is a useful tool when searching correspondences of points, called *Homography* if seen as a function, in stereo vision (see figure 7) since it reduces the two-dimensional search to a one-dimensional search.

Given a point $\mathbf{X}$ in $3D$ space, i.e. in $\mathbb{R}^3$, its corresponding image points in two cameras with centres at $\mathbf{C}$ and $\mathbf{C}'$, are $\mathbf{x}$ and $\mathbf{x}'$ respectively. It is clear that $\mathbf{X}$, $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{C}$ and $\mathbf{C}'$ are coplanar (see figure 8), let's call this plane π. Suppose now that only $\mathbf{x}$ is known, then the question that epipolar geometry answers is how the point $\mathbf{x}'$ is constrained given $\mathbf{x}$ (see figure

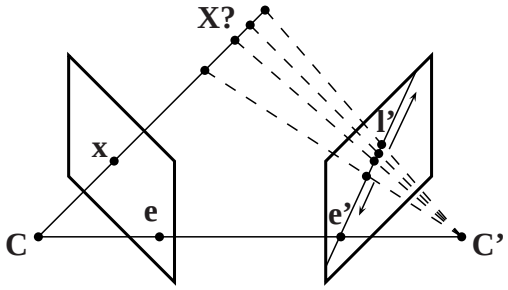


Figure 7: Epipolar geometry in stereo vision. **C** and $\mathbf{C}'$ are the centres of projection of each camera, **p** is the projection of a point **P** in the first camera, **e** and $\mathbf{e}'$ are the epipoles, $\overline{ee'}$ is the baseline and $\mathbf{l}'$ is the epipolar line corresponding to point **P** in the second camera.

7), and the answer is the *epipolar line*.

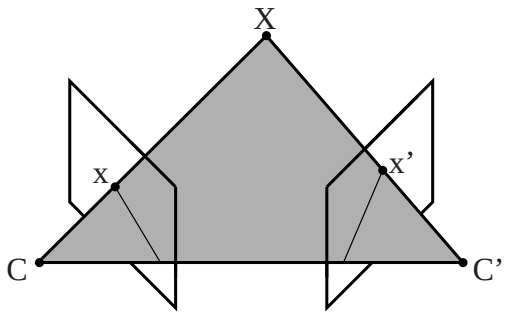


Figure 8: Epipolar geometry.

Here there are some important definitions (see figure 8):

Baseline: The baseline is the line that joins the camera centres.

Epipole: The epipoles are the points where the baseline intersects the image planes, or equivalently, the epipole is the image in one of the cameras of the camera centre of the other camera.

Epipolar Plane: An epipolar plane is a plane containing the baseline. There is a one-parameter family, or pencil, of epipolar planes.

Epipolar Line: An epipolar line is the intersection of an epipolar plane with one of the image planes. Each epipolar plane intersects both image planes at the same time, defining the correspondence between epipolar lines in both cameras. All the epipolar lines in one camera intersect at the epipole of this camera.

All the information of the epipolar geometry is algebraically encapsulated into the *Fundamental Matrix*. The fundamental matrix is a 3×3 matrix of rank 2 that links the image coordinates of a point in 2 different cameras. Given a point $\mathbf{X}$ in $3D$ space, its image coordinates in 2 different cameras, $\mathbf{x}$ and $\mathbf{x}'$, must satisfy

$$\mathbf{x}'^{\top} F \mathbf{x} = 0 \tag{13}$$

where F is the fundamental matrix for this given pair of cameras. Let's deduce its expression in terms of the 2 camera projection matrices (see section 4) P and P'. For further details see [XZ96]. Given an image point $\mathbf{x}$ in the first camera, its back-projected ray is a function of a scalar parameter λ

$$\mathbf{X}(\lambda) = P^{+}\mathbf{x} + \lambda \mathbf{C} \tag{14}$$

where P^{+} satisfies $PP^{+} = I$ and it is called the pseudoinverse of P. There are two particular points in that ray, $P^{+}\mathbf{x}$ at $\lambda = 0$ and $\mathbf{C}$ when $\lambda \to \infty$, that are seen by the second camera at $P'P^{+}\mathbf{x}$ and $P'\mathbf{C}$ respectively. The epipolar line is the line that joins these two points, i.e. $\mathbf{l}' = (P'\mathbf{C}) \times (P'P^{+}\mathbf{x}) = \mathbf{e}' \times (P'P^{+}\mathbf{x}) = [\mathbf{e}']_{\times}(P'P^{+})\mathbf{x}$ where $\mathbf{e}'$ is the epipole and $[\cdot]_{\times}$ denotes a map from a vector to its corresponding *Skew Symmetric Matrix*

$$[\mathbf{x}]_{\times} = \left[\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \right]_{\times} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{15}$$

Then, the fundamental matrix is

$$F = [\mathbf{e}']_{\times}(P'P^{+}) \tag{16}$$

and it satisfies

Transpose: If F is the fundamental matrix of the pair of cameras (P, P'), then $F^\top$ is the fundamental matrix of the pair of cameras (P', P)

Epipolar lines: For any given image point $\mathbf{x}$ in the first camera, its epipolar line is $\mathbf{l}' = F\mathbf{x}$. Similarly, given $\mathbf{x}'$, its epipolar line is $\mathbf{l} = F^\top \mathbf{x}'$.

Epipole: The epipoles $\mathbf{e}$ and $\mathbf{e}'$ satisfy $\mathbf{e}'^\top F = \mathbf{0}$ and $F\mathbf{e} = \mathbf{0}$, i.e. they are the left and right null-vector of F.

The scene reconstruction can be obtained by performing transfers of points by homographies using planes at different depth distances. This way, the depth of each point correspond to the depth of the plane giving a focused image of the point in the transferred image.

6 3-Views: Trifocal Tensor

When a scene is seen by three different cameras, there is a new multiple view object that plays the role of the fundamental matrix in two views, the *Trifocal Tensor* [SA90, Sha94, SW95]. One of the easiest ways of introducing the trifocal tensor is by means of the incidence relationship of three corresponding lines: given the image in three different cameras of a line in $3D$ space, the intersection of planes produced by back-projection of each one of the views of that line, must intersect in a line, the line in $3D$ space. Since in general, the intersection of 3 planes is not a single line, this geometric incidence condition provides a constraint on sets of corresponding lines (see figure 9). In the following, when dealing with tensors, the *Einstein Summation Convention* will be used. For an introduction to tensor analysis, see [Sim97] or [AMR93].

The trifocal tensor $\mathcal{T}$ is a $(2+1)$-rank tensor (2 contravariant and 1 covariant indexes) $\mathcal{T}_i^{jk}$. Given three general camera matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, $\mathbf{a}^k$, $\mathbf{b}^k$ and $\mathbf{c}^k$ represent the k-th row of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ respectively, and $\sim \mathbf{a}^l$ represents the matrix $\mathbf{A}$ without the l-th row, then the trifocal tensor can be expressed as (for the complete deduction of this expression see [HZ03], chapters $15-17$)

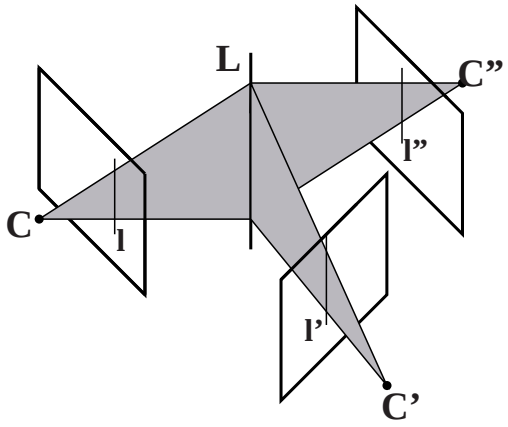


Figure 9: Constraint for the trifocal tensor

$$\mathcal{T}_i^{jk} = (-1)^{i+1} \det \begin{pmatrix} \sim \mathbf{a}^i \\ \mathbf{b}^j \\ \mathbf{c}^k \end{pmatrix} \tag{17}$$

It is easy to see that for the case where one of the camera matrices, for example $\mathbf{A}$, is $\mathbf{A} = (I \mid \mathbf{0})$, then the equation 17 can be expressed as

$$\mathcal{T}_i^{jk} = b_i^j c_4^k - b_4^j c_i^k \tag{18}$$

This trifocal tensor allows to do three important things, transfers by homographies, trilinear incidence relations (or trilinearities) and the retrieval of fundamental and camera matrices.

As said before, one of the most interesting things that the trifocal tensor allows to do is the transfer by homographies in two of the three views, i.e. given a point or line correspondence over 2 views, to determine its position in the third one. These transfers are obtained by using the trifocal tensor as an operator which takes a line and produces a homography matrix by means of the expression

$$h_i^k = l'_j \mathcal{T}_i^{jk} \tag{19}$$

which is the expression of the homography between the first and the third view obtained from the tensor by contraction with a line. Then,

using equation 19, transfers can be calculated as

Line transfer:

$$l_i = l'_j l''_k \mathcal{T}_i^{jk} \tag{20}$$

Point transfer:

$$x''^k = h_i^k x^i = (l'_j \mathcal{T}_i^{jk}) x^i \tag{21}$$

The second application commented before was trilinear incidence relations. Trilinear incidence relations, or trilinearities, are the relations of the coordinates of image points and lines. They are called trilinearities because each relation involves three image elements and all they are linear in the arguments of the tensor. There are 5 trilinearities, *line-line-line*, *point-line-line*, *point-line-point*, *point-point-line* and *point-point-point*.

Line-line-line:

$$(l_r \epsilon^{ris}) l'_j l''_k \mathcal{T}_i^{jk} = 0^s \tag{22}$$

Point-line-line:

$$x^i l'_j l''_k \mathcal{T}_i^{jk} = 0 \tag{23}$$

Point-line-point:

$$x^i l'_j (x''^k \epsilon_{kqs}) \mathcal{T}_i^{jq} = 0_s \tag{24}$$

Point-point-line:

$$x^i (x'^j \epsilon_{jpr}) l''_k \mathcal{T}_i^{pk} = 0_r \tag{25}$$

Point-point-point:

$$x^i (x'^j \epsilon_{jpr}) (x''^k \epsilon_{kqs}) \mathcal{T}_i^{pq} = 0_{rs} \tag{26}$$

where ϵ_{rst} (or ϵ^{rst}) is the *Levi-Civita permutation symbol*, i.e.

$$\epsilon_{rst} = \epsilon^{rst} = \begin{cases} +1 & \text{if } rst \text{ is an even permutation of } \{1,2,3\} \\ -1 & \text{if } rst \text{ is an odd permutation of } \{1,2,3\} \\ 0 & \text{if } r = s,\ s = t \text{ or } r = t \end{cases} \tag{27}$$

Finally, the trifocal tensor can also be employed to obtain the fundamental matrices or the camera projection matrices. If the trifocal tensor $\mathcal{T}$ is

considered as a $3 \times 3 \times 3$ "matrix", i.e. $\mathcal{T} = (T_1 \mid T_2 \mid T_3)$ where $T_i = \mathcal{T}_i^{jk}$, $j,k \in \{1,2,3\}$, then the epipoles $\mathbf{e}'$ and $\mathbf{e}''$ can be calculated as

$$\mathbf{e}'^{\top}(\mathbf{u}_1 \mid \mathbf{u}_2 \mid \mathbf{u}_3) = \mathbf{0} \tag{28}$$
$$\mathbf{e}''^{\top}(\mathbf{v}_1 \mid \mathbf{v}_2 \mid \mathbf{v}_3) = \mathbf{0} \tag{29}$$
$$\tag{30}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the left and right null-vectors, respectively, of T_i. Then, the fundamental matrices F_{21} and F_{31} can be calculated as

$$F_{21} = ([\mathbf{e}']_\times T_1 \mathbf{e}'' \mid [\mathbf{e}']_\times T_2 \mathbf{e}'' \mid [\mathbf{e}']_\times T_3 \mathbf{e}'') \tag{31}$$
$$F_{21} = ([\mathbf{e}'']_\times T_1 \mathbf{e}' \mid [\mathbf{e}'']_\times T_2 \mathbf{e}' \mid [\mathbf{e}'']_\times T_3 \mathbf{e}') \tag{32}$$

Camera matrices can be also recovered from the trifocal tensor, but since this one is independent of 3D projective transformations, the camera matrices can be computed only up to a projective ambiguity. See [HZ03] pages $374 - 376$ for further details.

7 4-Views: Quadrifocal Tensor

Going one step further, the *Quadrifocal Tensor* is found. Given a point correspondence across 4 views of a point $\mathbf{X}$, $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}'' \leftrightarrow \mathbf{x}'''$, with camera matrices P_a, P_b, P_c and P_d, according to equation 1 projection equations can be written as

$$\begin{pmatrix} P_a & \mathbf{x} & & & \\ P_b & & \mathbf{x}' & & \\ P_c & & & \mathbf{x}'' & \\ P_d & & & & \mathbf{x}''' \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ -\lambda \\ -\lambda' \\ -\lambda'' \\ -\lambda''' \end{pmatrix} = \mathbf{0} \tag{33}$$

The matrix on the left has at most rank 7 and so, all 8×8 determinant are zero. The determinant built using two rows from each projection matrix defines a quadrilinear relationship of the form

$$x^i x'^j x''^k x'''^l \epsilon_{ipw} \epsilon_{jqx} \epsilon_{kry} \epsilon_{lsz} \mathcal{Q}^{pqrs} = 0_{pqrs} \tag{34}$$

where the quadrifocal tensor, denoted by $\mathcal{Q}^{pqrs}$, is defined by

$$\mathcal{Q}^{pqrs} = det \begin{pmatrix} \mathbf{P}_a^p \\ \mathbf{P}_b^q \\ \mathbf{P}_c^r \\ \mathbf{P}_d^s \end{pmatrix} \tag{35}$$

In the quadrifocal tensor case, all the indices are contravariant and there is no distinguished view as in the trifocal tensor case. The quadrifocal tensor can be written in the case of 4 corresponding lines, $l \leftrightarrow l' \leftrightarrow l'' \leftrightarrow l'''$ as

$$l_p l'_q l''_r l'''_s \mathcal{Q}^{pqrs} = 0 \tag{36}$$

but, according to equation 34, the condition holds as long as there is a single point in space that projects onto the four image lines and so, it is not necessary that the 4 image lines correspond to the same line in space.

Like with the trifocal tensor, it is possible to extract the camera matrices or the epipoles from the quadrifocal tensor, but the calculations needed to obtain them as well as the calculations to obtain the quadrifocal tensor itself, begin to be too costly. For further details on the quadrifocal tensor and its properties and calculations, see [Hey98] or [Har98].

8 n-Views $(n > 4)$: What a Hard Problem!

At this point, it could be thought that there is a (multi)linear constraint, of the style of the constraints presented until now, for each given number of cameras observing a scene, but this is not the case. All multiple view relations that exist between homogeneous coordinates of image points and lines in five or more views of a static scene can be expressed as combinations of the epipolar constraints between any pair, the trifocal constraints between any triple and the quadrifocal constraints between any quadruple of views [Tri95, Moo98, Hey98]. In spite of this, there are several

algorithms that combine the information given by each image in order to reconstruct the scene.

Given a set of 3D points $\mathbf{X}_j$ viewed by a set of cameras with projection matrices P^i, where $\mathbf{x}_j^i$ denotes the image coordinates of point j seen by camera i, the reconstruction problem consists of finding the camera matrices P^i and the points in 3D $\mathbf{X}_j$ only knowing the image coordinates and the correspondences of the points. There are 3 algorithms of special relevance that solve this problem, the *Bundle Adjustment* algorithm, the *Factorisation* algorithm and the *Projective Factorisation* algorithm. These algorithms are briefly introduced in the following paragraphs.

The bundle adjustment algorithm involves an adjustment of the bundle of rays between each camera centre and the set of 3D points by means of the following minimisation

$$\min_{\hat{P}^i, \hat{\mathbf{X}}_j} \sum_{i,j} d(\hat{P}^i \hat{\mathbf{X}}_j, \mathbf{x}_j^i)^2 \tag{37}$$

where $d(\mathbf{x}, \mathbf{y})$ is the geometric image distance between the homogeneous points $\mathbf{x}$ and $\mathbf{y}$. This method is tolerant with missing data and provides a ML estimate, but it requires a good initialisation and it can become an extremely large minimisation problem. This is why it is usually a good idea to use it as the final step of another reconstruction algorithm.

In the case of affine cameras, the *Factorisation Algorithm*, introduced by Tomasi and Kanade in [TK92], is of special relevance. It was demonstrated in [RM96] by Reid and Murray that under isotropic zero-mean Gaussian noise, independent and equal for each measured point, the factorisation algorithm achieves a *Maximum Likelihood* affine reconstruction. The basic idea of the algorithm is to decompose a $2m \times n$ matrix W composed of the image coordinates of n points ($n \geq 4$) seen by m cameras, remember that $\mathbf{x} = P\mathbf{X}$, by means of the SVD decomposition ($W = UDV^\top$). At the output, the camera projection matrices of each camera and the 3D information of each point is obtained up to multiplication by a common matrix A. The algorithm can also be relaxed to deal with deformable objects by modelling them as a linear combination over basis sets, such as

for example an *Active Appearance Model* [CET01].

The affine factorisation algorithm does not apply to projective reconstruction, but Sturm and Triggs showed in [ST96] that if the *projective depth* (λ in equation 1) is known for each of the used points, then a factorisation algorithm similar to the affine one can be applied. The problem is that, since the real depths are unknown, an initial estimation must be done in order to obtain an estimation of the real depths by means of an iterative process that it is not guaranteed to converge to a global optimum. Nevertheless, with a good initialisation, the algorithm obtains the projection matrices and the 3D information of the points up to a common projective transformation.

9 Virtual Views

Humans have two eyes but only one image, that seems to come from the middle point between the eyes, is perceived. This means that somehow, the brain combines these two images in order to obtain only one that seems to come from a point where there is no eye to capture it. But it also means that it must be possible to generate a view of a scene where there is no camera, by means of the combination of several views of that scene from other points. This is what is known as *Virtual Viewpoint*, from a camera point-of-view, or *View Rendering* from a scene point-of-view.

Using techniques presented in the previous section, new points of view can be generated since the 3D information of the scene is obtained, but it is not necessary to solve completely the reconstruction of the scene to generate (render) new views of the scene. Techniques allowing to render new views without a previous reconstruction of the scene are known as *Image-Based Rendering* (IBR) techniques. One of the main advantages of IBR in front of complete scene reconstruction, in addition to the fact that they are usually less computational intensive, is that there are in general no geometric artifacts introduced in the scene and the output has already by itself a photorealistic aspect.

IBR techniques can be divided into different categories, following different criterions. For example, in [CSN07], three subgroups are proposed according to the amount of geometry information required from the scenes or objects: *Rendering with No Geometry*, *Rendering with Implicit Geometry* and *Rendering with Explicit Geometry*. In the present document, another division, proposed in [ZC04], will be considered. This division is based on the assumptions made by each algorithm in order to reduce the dimension of the considered plenoptic function l, introduced in [AB91], which is a 7-dimensional function ($l : \mathbb{R}^7 \longrightarrow \mathbb{R}$) that gives the radiance, i.e. the amount of light, of a scene from any view point (x, y, z), at any viewing angle (θ, ϕ), for any wavelength λ and at any time t (see figure 10).

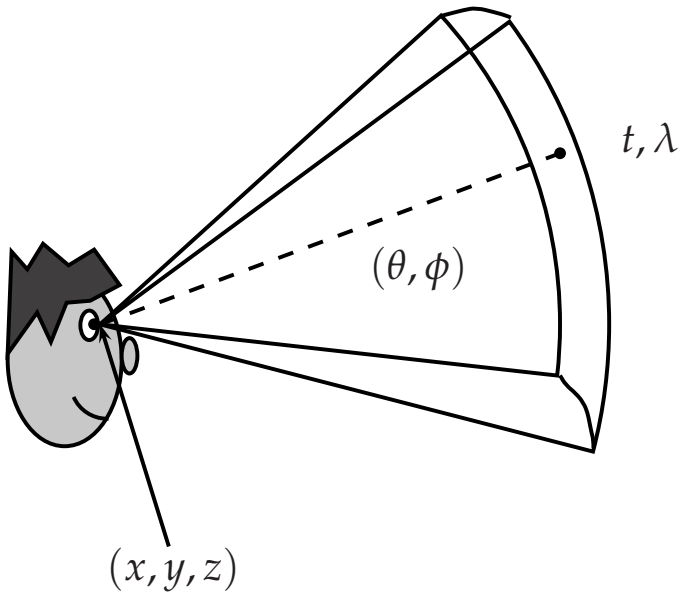


Figure 10: The 7 dimensions of the *Plenoptic Function*

9.1 Common assumptions to restrain de viewing space

IBR techniques are based on the interpolation of the plenoptic function from several samples. The plenoptic function, as explained before, has 7

dimensions and the direct interpolation of such a function would need a huge number of samples. This is why some assumptions are usually done in order to reduce the dimensionality of the function, allowing a practical implementation of an IBR system based on the sampling of the plenoptic function by means of one or several fixed or moving cameras. According to [ZC04], the most common IBR approaches, assume one or several of the following simplifications in order to reduce the dimensionality of the plenoptic function:

A1 The most common assumption, assumed by almost all IBR techniques, consists of not considering all the possible wavelengths but only 3, the corresponding ones to red, green and blue.

A2 Another very common assumption consists of supposing that the radiance along a light ray is constant, this way, the plenoptic function can be represented by its values on any surface that surrounds the scene. Although it is very reasonable, it can have some undesirable effects due to the finite resolution of a camera.

A3 A further assumption, that spares a lot of problems, is to assume a static scene, i.e. to ignore the time dimension.

A4 The restriction of the viewer freedom of moving to be on a surface reduces the plenoptic function one dimension. This assumption is reasonable since the eyes of a person are usually at an almost constant height-level and that the human beings are less sensitive to vertical parallax than to horizontal parallax.

A5 The viewer can also be restricted to move along a fixed path, which reduces the dimensionality of the plenoptic function in 2 dimensions.

A6 Finally, the viewer can also be restricted to be in a fixed position, which reduces the dimensionality of the plenoptic function in 3 dimensions. This is the assumption made for example by the popular QuickTime VR technology [Che95].

Note that in general, the dimensionality reduction achieved by each one of the assumptions is not addable. This is evident between assumptions

A4, *A5* and *A6*, but it is also valid for assumption *A2* when one of the assumptions *A4*, *A5* or *A6* is done. In table 9.1 we can observe the list of the most common IBR techniques with their corresponding plenoptic function dimensionality and the taken assumptions.

Dimension	**IBR Technique**	**Assumptions**
6D	Surface Plenoptic Function	A2
5D	Plenoptic Modelling	A1, A3
	Light Field Video	A1, A2
4D	Light Field	A1, A2, A3
	Lumigraph	A1, A2, A3
	Plenoptic Video	A1, A2, A5
3D	Concentric Mosaics	A1, A2, A3, A4
	Panoramic Video	(A1, A6) or (A1, A3, A5)
2D	Image Mosaicing	A1, A3, A6

Table 1: IBR techniques with their corresponding plenoptic function dimensionality and assumptions

9.2 6D Representations

Surface Plenoptic Function

The Surface Plenoptic Function (SPF), introduced in [ZC03], is a simplification of the 7D plenoptic function taking into account the assumption *A2*. As commented before, assuming *A2* allows to represent the plenoptic function by its values on a surrounding surface. The SPF approach considers the surrounding surface to be the scene surface itself. It is difficult to capture real scenes with unknown geometry using this technique, but SPF was used in [ZC03] for analysing the Fourier spectrum of IBR and how it can be used to sample IBR data more efficiently.

9.3 5D Representations

Plenoptic Modelling

The plenoptic modelling [MB95] simplifies the plenoptic function assuming *A1* and *A3*, obtaining as result a 5-dimensional approximation $l(x, y, z, \theta, \phi)$. The approach consists of recording a static scene by means of a set of cameras that make a continuous panning, making a cylindrical projection of the captured images along the panning. The rendering of new views is done by warping the nearby cylindrical projected images to the new view-point based on their epipolar relationship and visibility tests.

9.4 4D Representations

Light Field and Lumigraph

The Light Field [LH96] and the Lumigraph [GGSC96] are probably the most well-known IBR techniques. Both are based on the assumptions *A1*, *A2* and *A3*, i.e. they ignore the wavelength, they assume a constant radiance along a line in "free space" and they ignore the time dimension. With these assumptions, the resulting plenoptic function can be parameterised as $l(s, t, u, v) : \mathbb{R}^4 \mapsto \mathbb{R}$, where (s, t) and (u, v) are the intersecting coordinates of an incident light ray with two parallel planes, the camera plane and the focal plane, parameterised as the st-plane and the uv-plane (see figure 11). Connecting each of the discrete points of camera plane with all the points of the focal plane, a 2D array of images is obtained, the 2D array of captured light rays. With this 2D array of images, to create a new view of the scene the rays received at this point are calculated. These rays are calculated by quadrilinear interpolation of the nearby recorded light rays (see grey dots in figure 11). This rendering can be done in real time, and that is one of the most important features of these IBR techniques.

The main difference between Light Field and Lumigraph is that Light Field assumes no knowledge about the geometry of the scene while Lu-

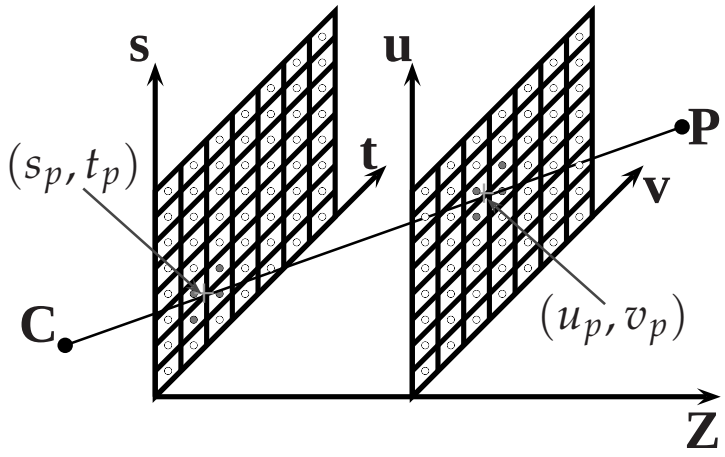


Figure 11: Parameterisation of a light ray using two parallel planes. Since the coordinates in each plane are discrete, an incident ray can affect more than one recorded ray.

migraph reconstructs a rough estimation of the geometry of the scene. This implies that the number of required samples of the Light Field approach is higher than in the Lumigraph approach, and that in this second, an irregular sampling of the scene with for example a tracked hand-held camera, as proposed in [GGSC96], can be done. Nevertheless, Lumigraph requires a re-sampling process to place each captured image onto a uniform sampling grid. An unstructured Lumigraph approach that allows a non-uniform sampling grid is proposed in [BBM+01]. Regarding Light Field, in [CNG+05] a dynamic version known as Plenoptic Video is presented. This approach does not assume *A3* but it uses *A4*, remaining this way in a 4-dimensional approximation of the plenoptic function. There exists also a dynamic extension (i.e. not assuming *A3* and obtaining then a 5-dimensional approximation) of the Light Field approach, introduced in [WSLH01] and called the Light Field Video. The approach is based on an array of 128 CMOS cameras that records a synchronised video flow of 640×480 pixel images at 30fps.

9.5 3D Representations

Concentric Mosaics

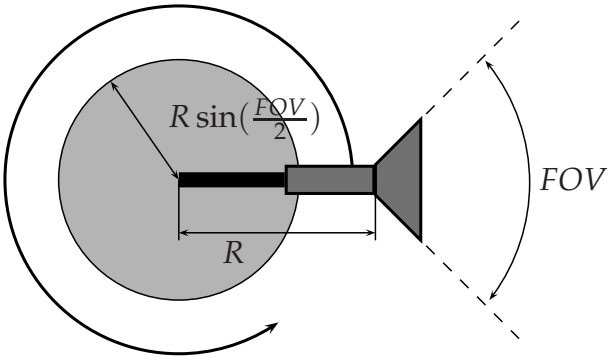


Figure 12: Sketch of a concentric mosaics capturing system.

Concentric Mosaics [SH99] is another well-known IBR approach, based on assumptions *A1*, *A2* and *A3* (like the Light Field) and *A4*, restricting the cameras and the viewers to be on a plane and obtaining a 3-dimensional approximation of the plenoptic function. The method consists of capturing the scene by a camera mounted at the end of a beam that rotates. The images are captured at regular intervals of rotation and the captured light rays are indexed by the beam rotation angle α and the pixel location (u, v), obtaining $l(\alpha, u, v)$. The virtual viewpoint can be placed freely inside a rendering circle with the centre at the centre of rotation of the capturing camera and a radius of $R\sin(FOV/2)$, where R is the camera path radius and FOV is the *Field Of View* of the cameras (see figure 12). The rendering process is based on the splitting of the captured images into slits and the later reassembling of interpolated neighboring slits.

This approach presents an important vertical distortion, independent of the capture density on the camera path. In order to minimise this distortion, in [SH99] a depth correction was used, but this implies that a rough knowledge of the scene geometry must be known. The origin of this distortion is that, as commented before, the dimensionality reductions done

by each assumption are not addable in general. In this case, assumptions *A2* and *A4* are not addable and that makes that the light ray space in the Concentric Mosaics approach is 4-dimensional, but the approach assumes it to be 3-dimensional. An easy extension of Concentric Mosaics to 4D consists of using a vertical array of cameras on the rotating beam.

Panoramic Video

Panoramic Video is another popular 3D IBR representation [Che95, FK00, Nay97, XT97]. It has 2 modalities of operation, one for dynamic scenes (using assumptions *A1* and *A6*) and another one for static scenes (with assumptions *A1*, *A3* and *A5*). In panoramic video, the field of view is often 360^o, allowing the viewer to pan and zoom freely and, in the case of static scenes, also to place freely the viewpoint.

The capture of a panoramic video is an easy task, consisting only of capturing a video sequence with a multi-camera system [FK00], an omnidirectional camera [Nay97] or a camera with a fisheye lens [XT97]. The rendering process consist only of warping from cylindrical or spherical projected images to a planar projection of the region of interest. Due to the simplicity of this approach and the acceptable obtained quality, it is widely used.

9.6 2D Representations

Image Mosaicing

Image mosaicing uses a 2D representation of the plenoptic function and, depending on how the light rays are recorded, image mosaicing techniques can be classified into *Single Centre of Projection Mosaics* or *Multiple Centre of Projection Mosaics*. The former techniques, that only allow the user to change his view direction, are known as *Panoramic Mosaics*, or simply *Panorama*, and index the light rays only according to their directions, i.e. $l(\theta, \phi)$, since the centre of projection does not change during the

registration. Then, the input images are related by 2D projective transformations that can be already known [GH86] or recovered from images [SS]. In the most general case of multiple centres of projection, that allows the user to move along a path or surface, the light rays are usually indexed by the position of the camera in a manifold where the camera moves and captures images usually perpendicullary or tangentially to the manifold. These techniques are also called *Manifold Mosaics* [ZT90, PH97, PBE99].

The rendering of image mosaicing is, in general, very simple. For panoramic mosaics it is usually enough to perform a warping from a cylindrical or spherical projected mosaic to a planar projected image, but in a general manifold mosaic it is a little more difficult since the warping could be unknown, requiring to use small regions of the mosaic directly for rendering, as long as the field of view of the rendered image is small enough [WFH+97].

10 Conclusions and Future Work

In the last paragraphs, geometric constraints of multiple view systems and image based rendering techniques have been presented. With these geometric constraints, the $3D$ reconstruction of a scene can be obtained and thus a virtual view of the scene from any desired point can be generated. This way of generating virtual views presents the advantage that it gives all the $3D$ information of the scene (excepting, of course, for occluded regions) and therefore, a virtual view can be generated from "everywhere". But it generates visual artifacts due to errors in the $3D$ reconstruction (above all in occluded regions), it is very computationally intensive and in addition this computation depends on the scene complexity. Due to this, less computationally expensive processes of generating virtual views have been proposed and studied by the computer vision community, obtaining the image based rendering techniques. These techniques allow the generation of virtual views without a complete reconstruction of the $3D$ scene or even without any $3D$ information, obtaining algorithms that are independent of the scene complexity and based on the

sampling and interpolation of the plenoptic function. IBR techniques generate outputs without the visual artifacts introduced by the reconstruction methods but the problem with them is the difficulty of sampling a high-dimensional function as the plenoptic function.

As future work, it could be interesting to study the application into IBR techniques of hybrid interpolation techniques (in the sense of the combination of continuous and discrete nature) like for example B-splines or wavelets, instead of pure discrete interpolation techniques. Another crucial point on IBR techniques that needs more research is the sampling process of the plenoptic function and how it can be optimised using the minimum number of cameras. For this purpose, the use of models of the scene, containing information about the objects and/or the dynamics of the scene, could be very interesting. To finish, also the study of image based rendering when dealing with non-lambertian surfaces would be interesting, since this kind of surfaces are widely present on real scenes.

References

[AB91] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20, 1991.

[AMR93] R. Abraham, J.E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer, 1993.

[ATFC07] G. B. Akar, A. M. Tekalp, C. Fehn, and M. R. Civanlar. Transport methods in 3DTV: A survey. *IEEE Trans. Circuits and Systems for Video Technology*, 17(11):1622–1630, November 2007.

[AYG+07] A. A. Alatan, Y. Yemez, U. Goedodkbay, X. Zabulis, K. Moeller, C. E. Erdem, C. Weigel, and A. Smolic. Scene representation technologies for 3DTV: A survey. *IEEE Trans. Circuits and Systems for Video Technology*, 17(11):1587–1605, November 2007.

[BBM+01] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432, 2001.

[BWS+07] P. Benzie, J. Watson, P. Surman, I. Rakkolainen, K. Hopf, H. Urey, V. Sainov, and C. von Kopylow. A survey of 3DTV displays: Techniques and technologies. *IEEE Trans. Circuits and Systems for Video Technology*, 17(11):1647–1658, November 2007.

[CET01] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[Che95] S. E. Chen. Quicktime VR: an image-based approach to virtual environment navigation. In *SIGGRAPH*, pages 29–38. ACM, 1995.

[CNG+05] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan, and H. Y. Shum. The plenoptic video. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(12):1650–1659, dec 2005.

[CSN07] S.C. Chan, Heung-Yeung Shum, and King-To Ng. Image-based rendering and synthesis. *IEEE Signal Processing Magazine*, 24(6):22–33, 2007.

[FK00] J. Foote and D. Kimber. Flycam: practical panoramic video. In *MULTIMEDIA: Proceedings of the 8th ACM International Conference on Multimedia*, pages 487–488, New York, NY, USA, 2000. ACM.

[GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54, 1996.

[GH86] N. Greene and P. S. Heckbert. Creating raster omnimax images from multiple perspectives views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications*, 6(6):21–27, June 1986.

[Har98] R. I. Hartley. Computation of the quadrifocal tensor. In *European Conference on Computer Vision*, pages 20–35. Springer-Verlag, 1998.

[Hey98] A. Heyden. A common framework for multiple view tensors. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 3–19, London, UK, 1998. Springer-Verlag.

[HZ03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[LH96] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, pages 31–42, 1996.

[MB95] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering approach. In *SIGGRAPH*, pages 39–46, 1995.

[Moo98] T. Moons. A guided tour through multiview relations. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 304–346. Springer, 1998.

[Nay97] S. K. Nayar. Catadioptric omnidirectional camera. In *CVPR*, pages 482–488. IEEE Computer Society, 1997.

[PBE99] S. Peleg and M. Ben-Ezra. Stereo panorama with a single camera. In *Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition*, pages 395–401. IEEE Computer Society, June 1999.

[PH97] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition*, page 338. IEEE Computer Society, 1997.

[RK89] R. Roy and T. Kailath. ESPRIT - estimation of signal parameters via rotational invariance techniques. *IEEE Trans. ASSP*, ASSP-37(7):984–995, 1989.

[RM96] I. D. Reid and D. W. Murray. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1):41–60, 1996.

[SA90] M. E. Spetsakis and Y. Aloimonos. A unified theory of structure from motion. In *Image Understanding Workshop*, pages 271–283, 1990.

[Sch79] R. Schmidt. Multiple emitter location and signal parameter estimation. In *Proc. RADC Spectrum Estimation Workshop*, pages 243–258, 1979.

[SH99] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In Alyn Rockwood, editor, *SIGGRAPH*, Annual Conference Series, pages 299–306, Los Angeles, 1999. Addison Wesley Longman.

[Sha94] A. Shashua. Trilinearity in visual recognition by alignment. *Lecture Notes in Computer Science*, 800:479–484, 1994.

[Sim97] J.G. Simmonds. *A Brief on Tensor Analysis*. Undergraduate Texts in Mathematics. Springer, 1997.

[SMS+07] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz. Coding algorithms for 3DTV: A survey. *IEEE Trans. Circuits and Systems for Video Technology*, 17(11):1606–1621, November 2007.

[SS] R. Szeliski and H.-Y. Shum. Creating full view panoramic mosaics and environment maps. In *Proceedings of the ACM SIGGRAPH Conference*, pages 251–258.

[ST96] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. *Lecture Notes in Computer Science*, 1065:709–720, 1996.

[SW95] A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geometry. In *ICCV '95: Proceedings of the 5th International Conference on Computer Vision*, pages 920–925, Cambridge, MA, 1995.

[Tan06] M. Tanimoto. Overview of free viewpoint television. *Signal Processing: Image Communication*, 21(6):454–461, July 2006.

[TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[Tri95] B. Triggs. Matching constraints and the joint image. In *International Conference on Computer Vision*, pages 338–343, 1995.

[WFH+97] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin. Multiperspective panoramas for cel animation. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 243–250, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[WK04] J. Woetzel and R. Koch. Multi-camera real-time depth estimation with discontinuity handling on PC graphics hardware. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 741–744, August 2004.

[WSLH01] B. S. Wilburn, M. Smulski, H.-H. K. Lee, and M. A. Horowitz. Light field video camera. volume 4674, pages 29–36. SPIE, 2001.

[WW00] R. S. Wang and Y. Wang. Multiview video sequence analysis, compression, and virtual viewpoint synthesis. *IEEE Trans. Circuits and Systems for Video Technology*, 10(3):397–410, April 2000.

[XT97] Y. Xiong and K. Turkowski. Creating image based VR using a self-calibrating fisheye lens. In *CVPR*, pages 237–243. IEEE Computer Society, 1997.

[XZ96] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition, A Unified Approach*. Kluwer Academic Publishers, 1996.

[YNK+05] Z. Yang, K. Nahrstedt, Y. Kui, B. Yu, J. Liang, S.-H. Jung, and R. Bajscy. TEEVE: The next generation architecture for tele-immersive environments. In *Seventh IEEE International Symposium on Multimedia*, December 2005.

[YZC04] Z. Yue, S.K. Zhou, and R. Chellappa. Robust two-camera tracking using homography. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1–4, May 2004.

[ZC03] Cha Zhang and Tsuhan Chen. Spectral analysis for sampling image-based rendering data. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1038–1050, 2003.

[ZC04] C. Zhang and T. H. Chen. A survey on image-based rendering–representation, sampling and compression. *Signal Processing: Image Communication*, 19(1):1–28, 2004.

[ZT90] J. Y. Zheng and S. Tsuji. Panoramic representation of scenes for route understanding. In *Proceedings of the International Conference on Pattern Recognition*, pages 161–167, 1990.