FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
Comenius University
Bratislava

3D Vision

Lecture 2: Rigid Body Motion and Pinhole Camera Model

Ing. Viktor Kocur, PhD.

21.2.2023

Contents

Basic Pinhole Camera Model

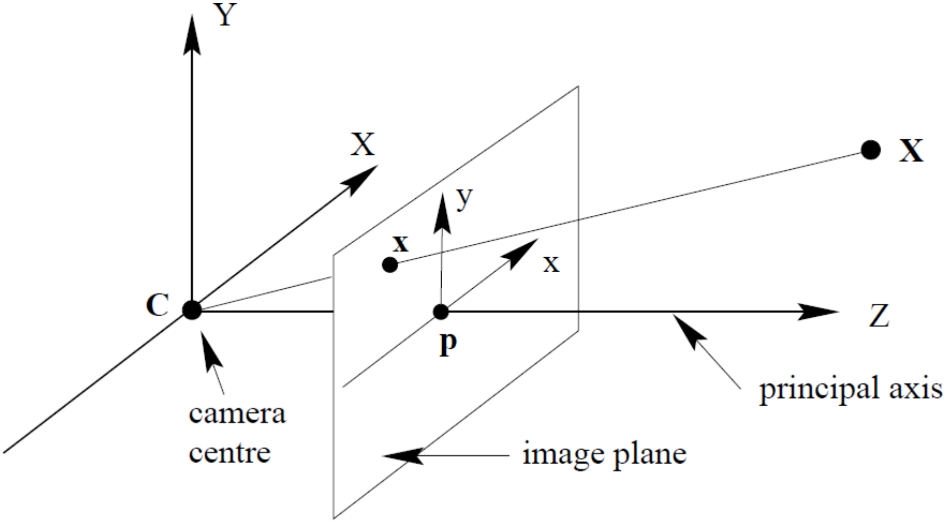


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Basic Pinhole Camera Model

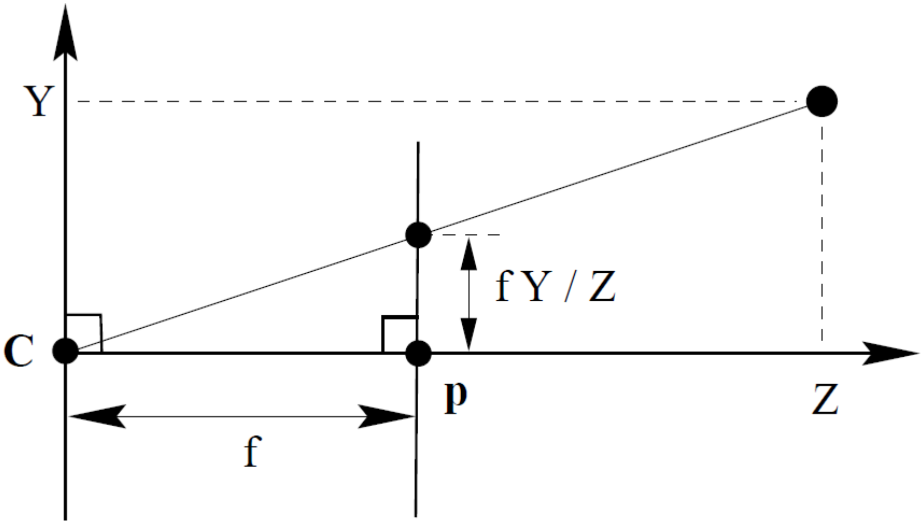


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Projection of points

A point $\boldsymbol{X} = (X, Y, Z)^T$ in world coordinates is projected onto the imaging plane to obtain coordinates $\boldsymbol{x} = (x, y)^T$ using the following equation:

$$\boldsymbol{x} = \pi(\boldsymbol{X}) = \begin{pmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{pmatrix}, \tag{1}$$

where f is the focal length.

Homogeneous Coordinates

Homogeneous coordinates can be used to represent points and vectors. In 2D, a point with homogeneous coordinates (x, y, w) represents the point $(x/w, y/w)$ in Cartesian coordinates.

- Points with $w \neq 0$ represent finite points in the Euclidean plane. In this case, the coordinates (x, y, w) are proportional to the Cartesian coordinates $(x/w, y/w)$ of the corresponding point in 2D space.
- Points with $w = 0$ represent points at infinity. In this case, the coordinates $(x, y, 0)$ represent the direction of the line that passes through the point at infinity.
- The triple $(0, 0, 0)$ is not valid in 2D homogeneous coordinates.

This also works in 3D coordinates where (X, Y, Z, W) in homogeneous coordinates represents the point $(X/W, Y/W, Z/W)$ in Cartesian coordinates.

Homogeneous Coordinates

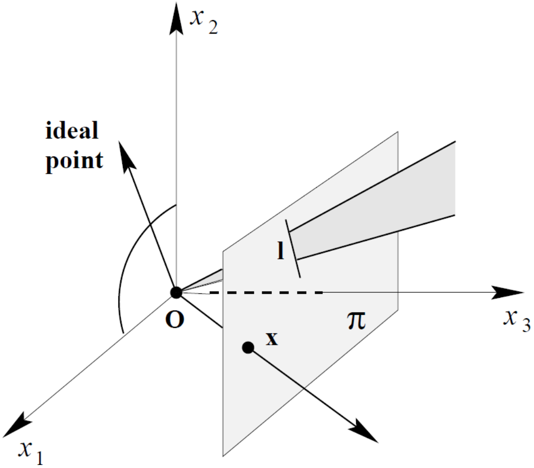


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Homogeneous Coordinates

Note that the homogeneous coordinates for points are not unique since $\boldsymbol{x}$ and $\alpha\boldsymbol{x}$ represent the same point if $\alpha \neq 0$. This creates an equivalence class defined by relation $\sim$:

$$\boldsymbol{p} \sim \boldsymbol{q} \iff (\exists\alpha \neq 0)(\alpha\boldsymbol{p} = \boldsymbol{q}). \tag{2}$$

Note that this sort of equivalence will also hold for most of the matrices we will work with as for $\boldsymbol{y} = A\boldsymbol{x}$ $\alpha\boldsymbol{y} = \alpha(A\boldsymbol{x}) = (\alpha A)\boldsymbol{x}$.

Lines in 2D

We can represent a 2D line by a vector $\boldsymbol{h} = (a, b, c)^T$ such that it represents the line given by the following equation:

$$ax + by + c = 0. \tag{3}$$

If we have a point in 2D expressed in homogeneous coordinates in a vector $\boldsymbol{x}$ and a line $\boldsymbol{h}$. Then the point lies on the line iff $\langle \boldsymbol{x}, \boldsymbol{h} \rangle = \boldsymbol{x}^T \boldsymbol{h} = 0$.

We can also find the intercept of two lines $\boldsymbol{h}$ and $\boldsymbol{h}$ as a point $\boldsymbol{y} = \boldsymbol{h} \times \boldsymbol{h} = [\boldsymbol{l}]_\times \boldsymbol{h}$. We can also find a line $\boldsymbol{k}$ connecting two points $\boldsymbol{p}$ and $\boldsymbol{q}$ in the same fashion as $\boldsymbol{k} = \boldsymbol{p} \times \boldsymbol{q} = [\boldsymbol{p}]_\times \boldsymbol{q}$. Note that same as with homogeneous coordinates, multiplying by non-zero scalar still gives the same line.

Planes in 3D

Similarly to 2D we can represent a plane in 3D space by a vector $\boldsymbol{\pi} = (a, b, c, d)^T$ such that it represents the plane given by the following equation:

$$aX + bY + cZ + d = 0. \tag{4}$$

Again if we multiply by a non-zero scalar we get the same plane. A 3D point $\boldsymbol{X}$ expressed in homogeneous coordinates lies on the plane if $\boldsymbol{\pi}^T \boldsymbol{X} = 0$. The vector $\boldsymbol{h} = (a, b, c)^T$ is in the direction of the plane normal and we can get the normal by calculating $\boldsymbol{n} = \frac{\boldsymbol{h}}{|\boldsymbol{h}|}$.

Simple Pinhole Camera Model

Using homogeneous coordinates we can express the simple pinhole camera model using the following equation:

$$\boldsymbol{x} \sim w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = P\boldsymbol{X} = K_f \Pi_0 \boldsymbol{X} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{5}$$

Coordinate Shift

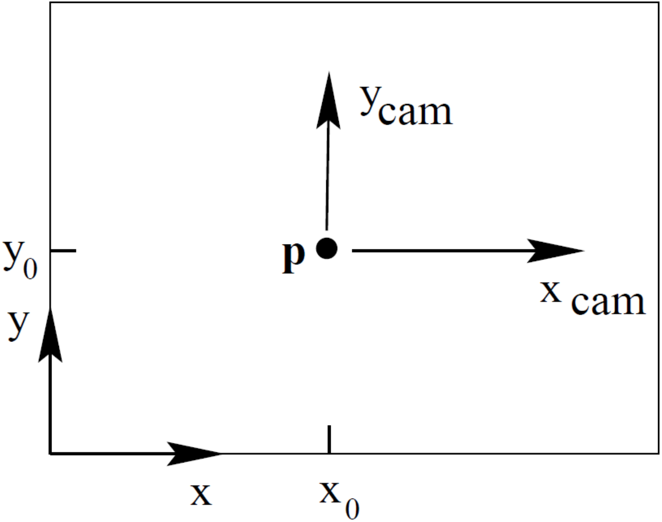


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Coordinate Shift

To shift the coordinates we modify the equation to obtain:

$$P\boldsymbol{X} = K\Pi_0\boldsymbol{X} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{6}$$

where (x_0, y_0) is the **principal point** (*hlavný bod*).

Camera Rotation and Translation

So far we have consider a coordinate system where the camera center is in the origin, the optical axis is the z axis and the remaining two axes are aligned with the image coordinates. However, we will often need to consider camera which has its own rotation R and translation $\boldsymbol{t}$ w.r.t the world coordinate system. In such case we may rewrite the equation into the form:

$$P\boldsymbol{X} = K[R|\boldsymbol{t}]\boldsymbol{X}. \tag{7}$$

We then call the matrix K the **intrinsic matrix** (*matica vnútorných parametrov*) or also the calibration matrix or the camera matrix and $[R|\boldsymbol{t}]$ the **extrinsic matrix** (*matica vonkajších parametrov*).

Intrinsic Camera Matrix

In a more general case K can be slightly more complicated:

$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{8}$$

where s is skew of the lens and f_x and f_y are different focal lengths for their corresponding directions. However, for most modern cameras we can assume that $f_x = f_y$, $s = 0$ and (x_0, y_0) lies in the center of the image.

Radial Distortion

The pinhole camera model may not account for some distortions of the image arising from the geometry of the camera lens. Most common such distortion is the radial distortion. As a result of this distortion straight lines from the imaged scene are curved in the image. The rate at which the lines are distorted is approximately symmetric around the principal point. Thus it is possible to model this type of distortion as a function of the distance of a pixel (x, y) from the principal point (x_0, y_0):

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2}. \tag{9}$$

Radial Distortion

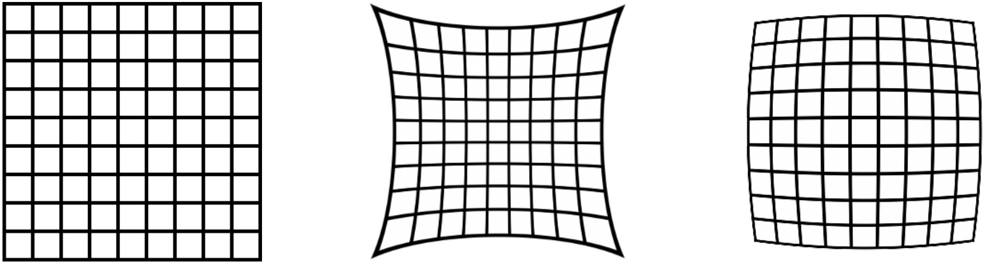
To correct for the distortion we can calculate the undistorted position of the pixels $(\hat{x}, \hat{y})$:

$$\hat{x} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{10}$$

$$\hat{y} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{11}$$

where k_1, k_2, k_3 are the parameters of the model. In some cases it is sufficient to assume that k_3 or even k_2 is zero. The mappings (10) and (11) can be used to transform the captured image to remove the effects of the distortion. Radial distortion is usually most pronounced in cheaper cameras or cameras with special lens.

Radial Distortion



a b c

Projection of planes

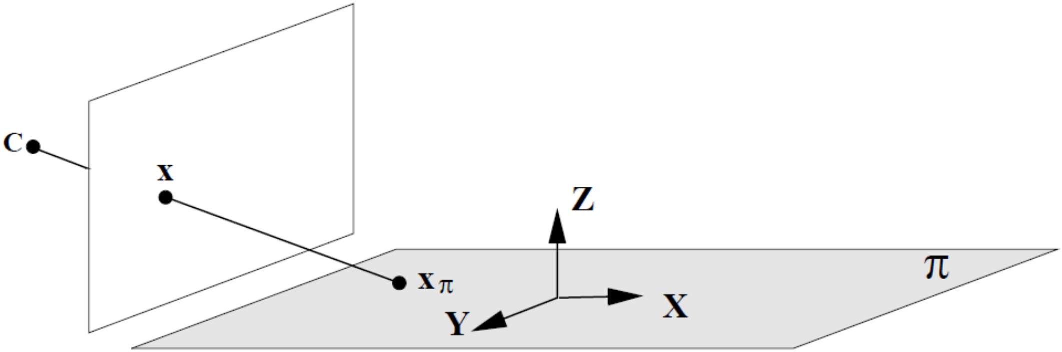


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Projections of Planes

We have discussed how points in the world are projected. Now we will consider how planes get projected onto an image. Suppose we have a plane in 3D space defined by $Z = 0$. All points on the plane then have coordinates $(X, Y, 0, 1)$. This leads to images in the form:

$$\boldsymbol{x} \sim P\boldsymbol{X} = K[R|\boldsymbol{t}]\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}. \tag{12}$$

Equation (12) can be simplified to:

$$\boldsymbol{x} \sim K[\boldsymbol{r}_{:,1}, \boldsymbol{r}_{:,2}, \boldsymbol{t}] \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \tag{13}$$

where $\boldsymbol{r}_{:,i}$ is the i-th column of the rotation matrix. We can observe that $H = K[\boldsymbol{r}_{:,1}, \boldsymbol{r}_{:,2}, \boldsymbol{t}]$ is a 3×3 matrix. If $[\boldsymbol{r}_{:,1}, \boldsymbol{r}_{:,2}, \boldsymbol{t}]$ is regular then also H is also regular. This is also known as a **homography**. H can be singular if the camera center lies on the plane. In that case H is not a homography.

Homographies preserve straight lines. A line $\boldsymbol{h}$ is transformed to $\boldsymbol{h}' \sim H^{-T}\boldsymbol{h}$.

Using Homographies

Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Projected Lines

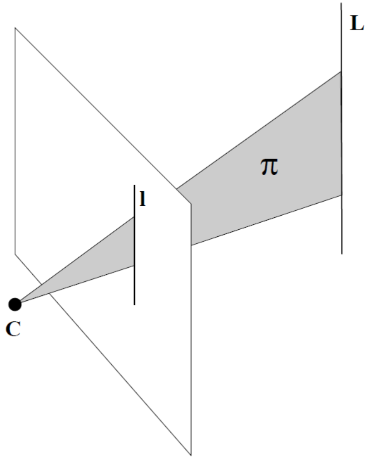


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Projected Lines

Unless the camera center lies on a line, the line in the world is projected to a line in the image. Points imaged to a line $\boldsymbol{h}$ have to lie on a plane $\boldsymbol{\pi}$ given by

$$\boldsymbol{\pi} = P^T \boldsymbol{h}. \tag{14}$$

Transformations with Fixed Camera Center

Suppose our coordinate system is such that we have $\boldsymbol{t} = \mathbf{0}$ for two different cameras. $P = K[R|\mathbf{0}]$ and $P' = K'[R'|\mathbf{0}]$. We can then show a relationship between real point $\boldsymbol{X}$ and its images in $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the two cameras:

$$\boldsymbol{x}' \sim P'\boldsymbol{X} \sim (K'R')(KR)^{-1}P\boldsymbol{X} \sim (K'R')(KR)^{-1}\boldsymbol{x}. \quad (15)$$

Note that the resulting matrix is a homography. This result can be useful for creating synthetic views with different camera paramaters from the same camera center. Note that this also means that it is not possible to recover the scene structure from two cameras with the same center.

Fixed Camera Center

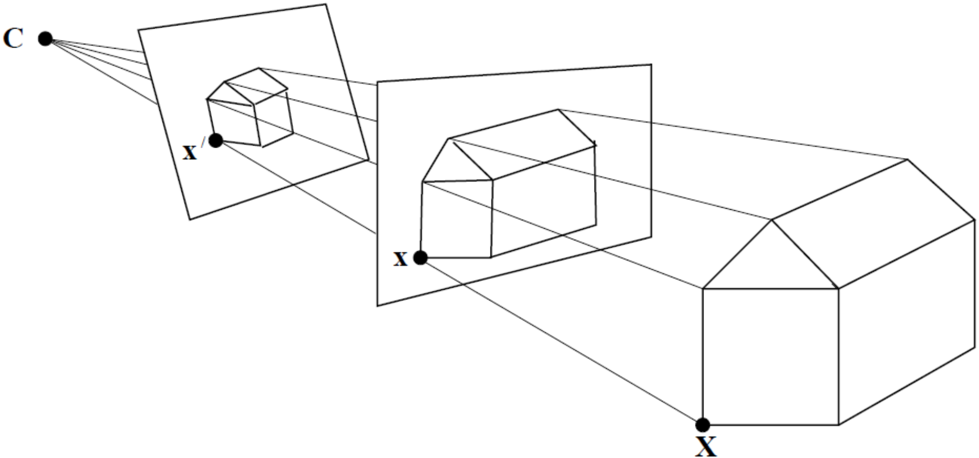
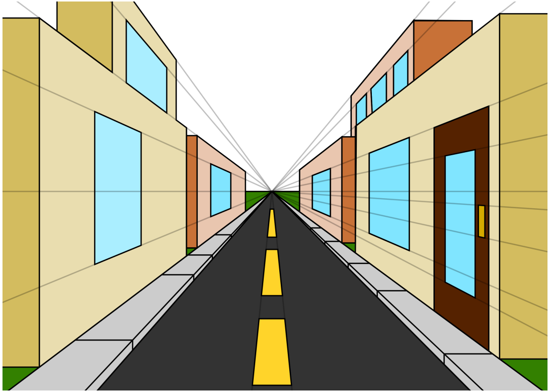


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Vanishing Points



Lines which are parallel in the real scene are imaged such that they all coincide into a single point called the **vanishing point** (*úbežník*).

Image adopted from: *Wikipedia*.

Vanishing Points

Let us consider a line in the world coordinates defined by a point $\boldsymbol{A}$ in homogeneous coordinates and a direction $\boldsymbol{D} = (\boldsymbol{d}^T, 0)$ with a world coordinate system aligned with the camera (e.g. $R = I$ and $\boldsymbol{t} = \boldsymbol{0}$). Let us parametrize the points on the line as $\boldsymbol{X}(\lambda) = \boldsymbol{A} + \lambda \boldsymbol{D}$. Using the camera model we can derive the image of the point:

$$\boldsymbol{x}(\lambda) = K\,[I|\boldsymbol{0}]\,\boldsymbol{X}(\lambda) = \boldsymbol{a} + \lambda K \boldsymbol{d}, \tag{16}$$

where $\boldsymbol{a}$ is the image of $\boldsymbol{A}$. Then the vanishing point $\boldsymbol{v}$ is obtained as the limit:

$$\boldsymbol{v} \sim \lim_{\lambda \to +\infty} \boldsymbol{x}(\lambda) \sim \lim_{\lambda \to +\infty} \boldsymbol{a} + \lambda K \boldsymbol{d} = K \boldsymbol{d}. \tag{17}$$

Note that we can also use (17) to calculate $\boldsymbol{d} = K^{-1} \boldsymbol{v}$.

Vanishing Points

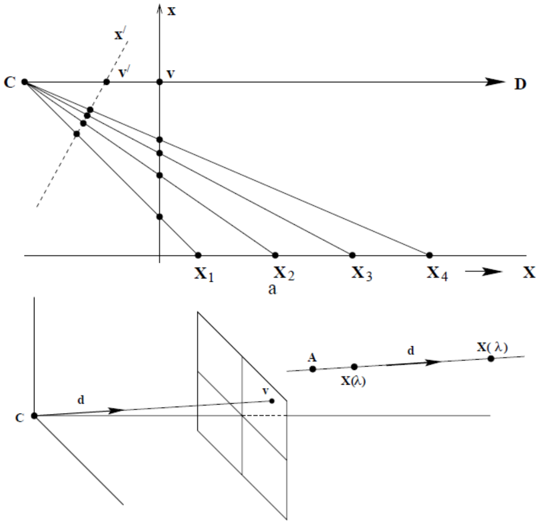


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

Vanishing Points - Angles

Given two vanishing points $\boldsymbol{u}$ and $\boldsymbol{v}$ imaged the same camera we can calculate the angle θ between them:

$$\cos\theta = \frac{\boldsymbol{u}^T(KK^T)^{-1}\boldsymbol{v}}{\sqrt{\boldsymbol{u}^T(KK^T)^{-1}\boldsymbol{u}}\sqrt{\boldsymbol{u}^T(KK^T)^{-1}\boldsymbol{v}}} \tag{18}$$

.

Note that this is especially useful when the vanishing points correspond to orthogonal directions. Then we can reduce (18) to:

$$0 = \boldsymbol{u}^T(KK^T)^{-1}\boldsymbol{v}. \tag{19}$$

This is useful as it places one constraint on K which can be used to find the intrinsic camera parameters which is also known as **camera calibration**.

Vanishing Lines

When considering a plane in the real world we may investigate the images of all of the directions parallel to the plane. The vanishing points for these directions form a line in the image. This line is called the **vanishing line** or sometimes also the **horizon** of the plane. Note that all real-world planes which are parallel to one another share the same vanishing line.

When the intrinsic parameters of the camera are known it is possible to use the vanishing line $\boldsymbol{h}$ of a plane to calculate the plane normal $\boldsymbol{n}$ in the real-world camera coordinate system as

$$\boldsymbol{n} = K^T \boldsymbol{h}. \tag{20}$$

Vanishing Lines

We can also calculate the angle θ between two planes with vanishing lines $\boldsymbol{k}$ and $\boldsymbol{h}$:

$$\cos\theta = \frac{\boldsymbol{k}^T(KK^T)\boldsymbol{h}}{\sqrt{\boldsymbol{k}^T(KK^T)\boldsymbol{k}}\sqrt{\boldsymbol{h}^T(K^TK)\boldsymbol{h}}}. \tag{21}$$

A vanishing point $\boldsymbol{v}$ corresponding to the direction of the normal of a plane which has the vanishing line $\boldsymbol{h}$ impose two constraints on the intrinsic matrix via the equation

$$\boldsymbol{h} \times \left((KK^T)^{-1}\boldsymbol{v}\right) \sim \boldsymbol{0}. \tag{22}$$

Vanishing Lines

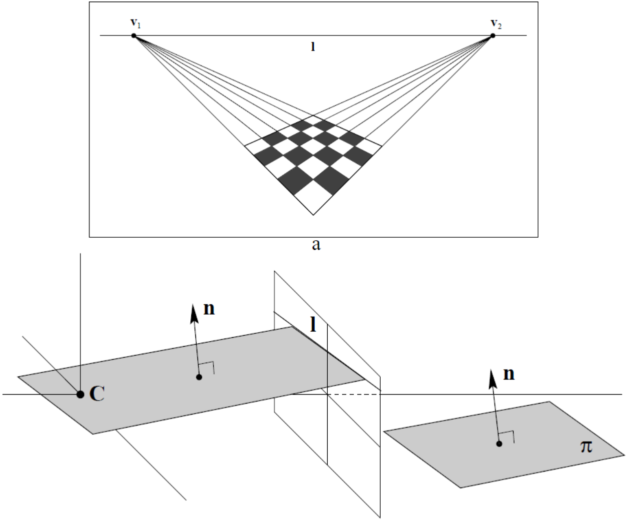


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003