Applied Computer Vision

David Vernon
Carnegie Mellon University Africa

vernon@cmu.edu
www.vernon.eu

Lecture 24

3D Vision I

Homogeneous coordinates, perspective projection, camera model and inverse perspective transformation

Homogeneous Coordinates

A 3D vector, $v = ai + bj + ck$, where i, j and k are unit vectors along the X, Y and Z axes is represented in homogenous co-ordinates as

$$v = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

where $a = \frac{x}{w}, b = \frac{y}{w}$ and $c = \frac{z}{w}$

Homogeneous Coordinates

Thus, the additional fourth co-ordinate w is just a scaling factor and means that a single 3D vector can be represented by several homogenous co-ordinates

For example, $3i + 4j + 5k$ can be represented by $\begin{bmatrix} 3 \\ 4 \\ 5 \\ 1 \end{bmatrix}$ or by $\begin{bmatrix} 6 \\ 8 \\ 10 \\ 2 \end{bmatrix}$.

Note that, since division of zero is indeterminate, the vector $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ is undefined.

The Camera Model and the Inverse Perspective Transformation

For any given optical configuration, there are two aspects to the relationship between world point and image point

- the ***camera model***, which maps a 3D world point to its corresponding 2D image point

- the ***inverse perspective transformation***, which is used to identify the 3D world point(s) corresponding to a particular 2D image point.

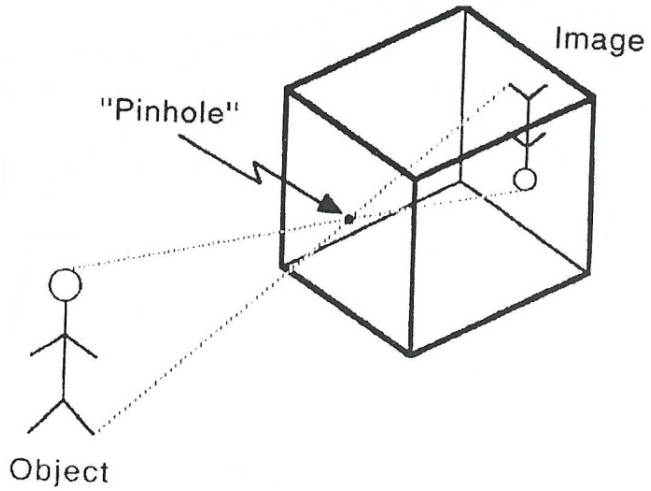
The Camera Model and the Inverse Perspective Transformation

Since the imaging process is a projection (from a 3D world to a 2D image), the inverse process, *i.e.* the inverse perspective transformation, cannot uniquely determine a single world point for a given image point

- the inverse perspective transformation thus maps a 2D image point into a line (an infinite set of points) in the 3D world,
- however, it does so in a useful and well-constrained manner

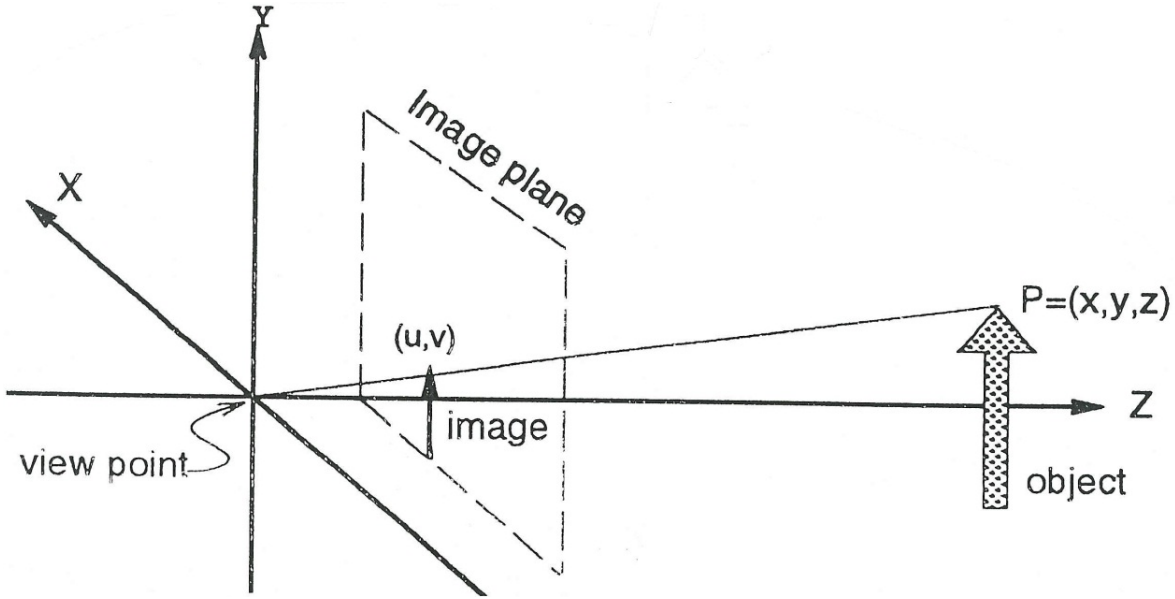
Perspective Projection

Pinhole model of a camera



Perspective Projection

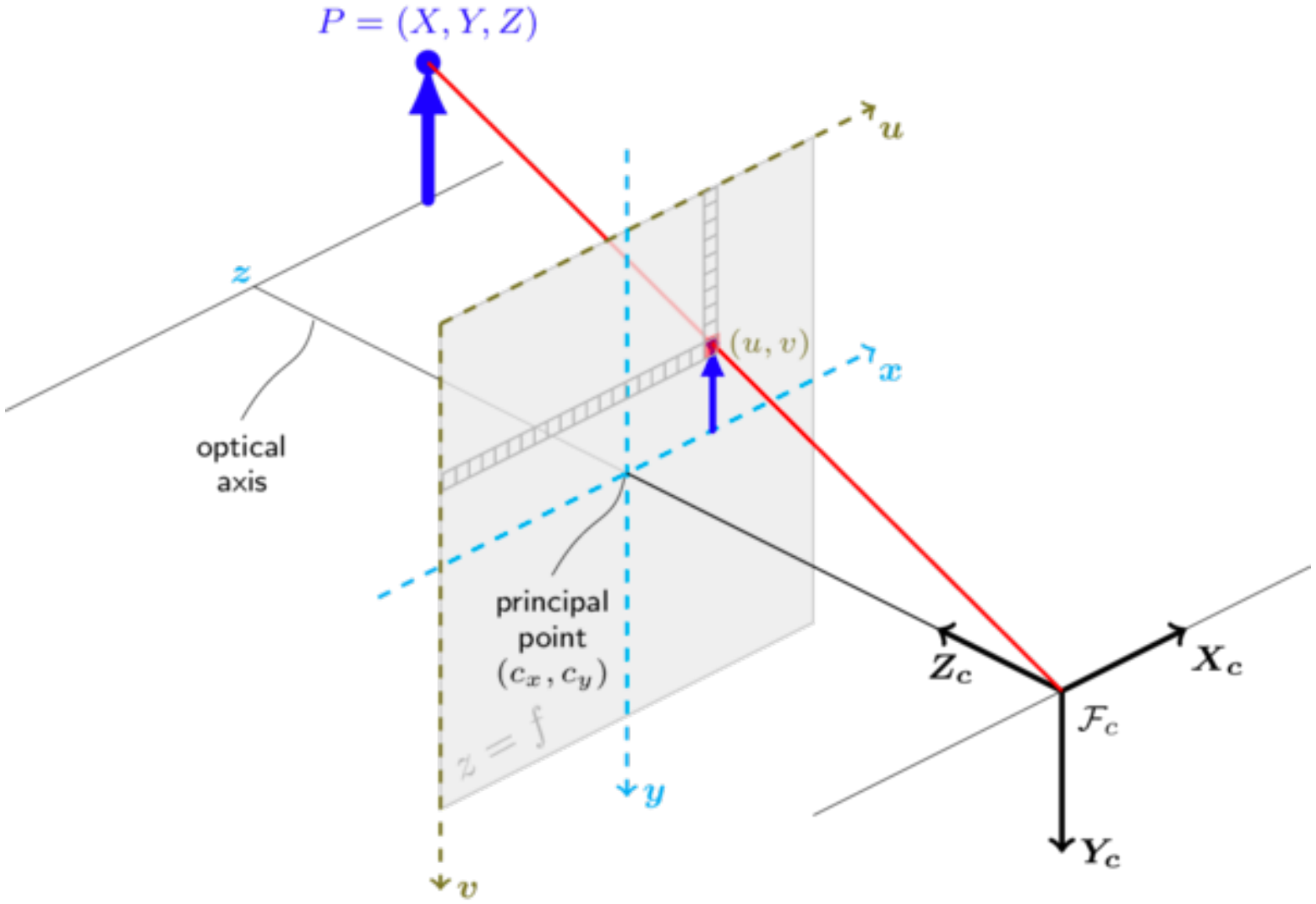
Pinhole model of a camera



$$u = \frac{f}{z}x \qquad v = \frac{f}{z}y$$

Source: Markus Vincze, Technische Universität Wien

Perspective transformation



http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

The Camera Model and the Inverse Perspective Transformation

For the following, we will assume that the camera model (and, hence, the inverse perspective transformation) is linear

This treatment closely follows that of [Ballard and Brown 1982]

The Camera Model

Let the image point in question be given by the co-ordinates $\begin{bmatrix} U \\ V \end{bmatrix}$ which, in homogenous co-ordinates is written $\begin{bmatrix} u \\ v \\ t \end{bmatrix}$.

Thus, $U = \frac{u}{t}$

and $V = \frac{v}{t}$

The Camera Model

Let the desired camera model, a transformation which maps the 3D world point to the corresponding 2D image point, be C.

Thus,

$$C\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ t \end{bmatrix}$$

The Camera Model

Hence C must be a 3*4 (homogenous) transformation

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix}$$

and

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ t \end{bmatrix}$$

The Camera Model

Expanding this matrix equation, we get :

$$C_{11}x + C_{12}y + C_{13}z + C_{14} = u \qquad (1)$$
$$C_{21}x + C_{22}x + C_{23}x + C_{24} = v \qquad (2)$$
$$C_{31}x + C_{32}x + C_{33}x + C_{34} = t \qquad (3)$$

The Camera Model

but $u = Ut$ and $v = Vt$

so $u - Ut = 0$ (4)

$v - Vt = 0$ (5)

Substituting (1) and (3) for u and t, respectively, in (4) and substituting (2) and (3) for v and t, respectively, in (5) :

$$C_{11}x + C_{12}y + C_{13}z + C_{14} - UC_{31}x - UC_{32}y - UC_{33}z - UC_{34} = 0 \quad (6)$$

$$C_{21}x + C_{22}y + C_{23}z + C_{24} - VC_{31}x - VC_{32}y - VC_{33}z - VC_{34} = 0 \quad (7)$$

The Camera Model

If we establish this association

i.e. if we measure the values of x, y, z, U and V

we will have two equations in which the only unknowns are the twelve camera model coefficients (and which we require)

Since a single observation gives rise to two equations, six observations will produce twelve simultaneous equations which we can solve for the required camera coefficients C_{ij}.

The Camera Model

Remember that these two equations arose from the association of a particular world point $\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

and a corresponding image point $\begin{bmatrix} u \\ v \\ t \end{bmatrix}$

The Camera Model

Before we proceed, however, we need to note that the overall scaling of C is irrelevant due to the homogenous formulation and, thus, the value of c_{34} may be set arbitrarily to 1 and we can re-write (6) and (7), completing the equations so that terms for each coefficient of C is included, as follows

$$C_{11}x + C_{12}y + C_{13}z + C_{14} + C_{21}0 + C_{22}0 + C_{23}0 + C_{24}0 - UC_{31}x - UC_{32}y - UC_{33}z = U$$
$$C_{11}0 + C_{12}0 + C_{13}0 + C_{14}0 - C_{21}x + C_{22}y + C_{23}z + C_{24} - VC_{31}x - VC_{32}y - VC_{33}z = V$$

This reduces the number of unknowns to eleven

The Camera Model

For six observations, we now have twelve equations and eleven unknowns: *i.e.* the system of equations is over-determined

Re-formulating the twelve equations in matrix form, we can obtain a least-square-error solution to the system using the pseudo-inverse method

The Camera Model

Let

$$X = \begin{bmatrix} x^1 & y^1 & z^1 & 1 & 0 & 0 & 0 & 0 & -U^1x^1 & -U^1y^1 & -U^1z^1 \\ 0 & 0 & 0 & 0 & x^1 & y^1 & z^1 & 1 & -V^1x^1 & -V^1y^1 & -V^1z^1 \\ x^2 & y^2 & z^2 & 1 & 0 & 0 & 0 & 0 & -U^2x^2 & -U^2y^2 & -U^2z^2 \\ 0 & 0 & 0 & 0 & x^2 & y^2 & z^2 & 1 & -V^2x^2 & -V^2y^2 & -V^2z^2 \\ x^3 & y^3 & z^3 & 1 & 0 & 0 & 0 & 0 & -U^3x^3 & -U^3y^3 & -U^3z^3 \\ 0 & 0 & 0 & 0 & x^3 & y^3 & z^3 & 1 & -V^3x^3 & -V^3y^3 & -V^3z^3 \\ x^4 & y^4 & z^4 & 1 & 0 & 0 & 0 & 0 & -U^4x^4 & -U^4y^4 & -U^4z^4 \\ 0 & 0 & 0 & 0 & x^4 & y^4 & z^4 & 1 & -V^4x^4 & -V^4y^4 & -V^4z^4 \\ x^5 & y^5 & z^5 & 1 & 0 & 0 & 0 & 0 & -U^5x^5 & -U^5y^5 & -U^5z^5 \\ 0 & 0 & 0 & 0 & x^5 & y^5 & z^5 & 1 & -V^5x^5 & -V^5y^5 & -V^5z^5 \\ x^6 & y^6 & z^6 & 1 & 0 & 0 & 0 & 0 & -U^6x^6 & -U^6y^6 & -U^6z^6 \\ 0 & 0 & 0 & 0 & x^6 & y^6 & z^6 & 1 & -V^6x^6 & -V^6y^6 & -V^6z^6 \end{bmatrix}$$

The Camera Model

$$c = \begin{bmatrix} C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \\ C_{21} \\ C_{22} \\ C_{23} \\ C_{24} \\ C_{31} \\ C_{32} \\ C_{33} \end{bmatrix} \qquad y = \begin{bmatrix} U^1 \\ V^1 \\ U^2 \\ V^2 \\ U^3 \\ V^3 \\ U^4 \\ V^4 \\ U^5 \\ V^5 \\ U^6 \\ V^6 \end{bmatrix}$$

thus,

$$X\,c = y$$

The trailing superscript denotes the observation number

The Camera Model

Then :

$$c = \left(X^T X\right)^{-1} X^T y$$

$$= X^{\dagger} y$$

We assumed above that we make six observations to establish the relationship between six sets of image co-ordinates and six sets of real world co-ordinates.

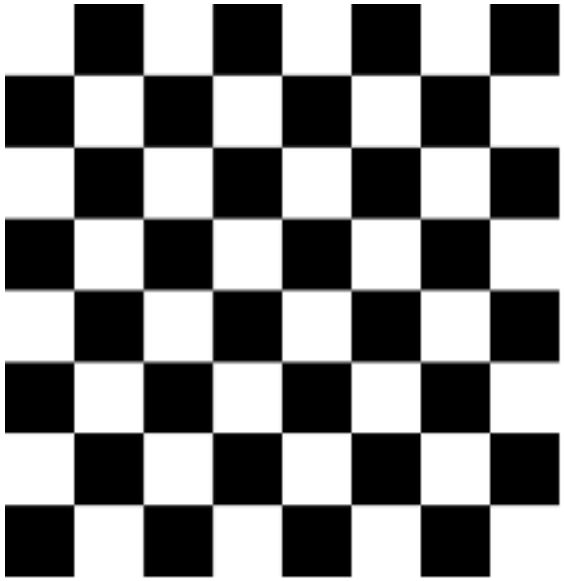
In general, it is better to significantly over-determine the system of equations by generating a larger set of observations than the minimal six

The Camera Model

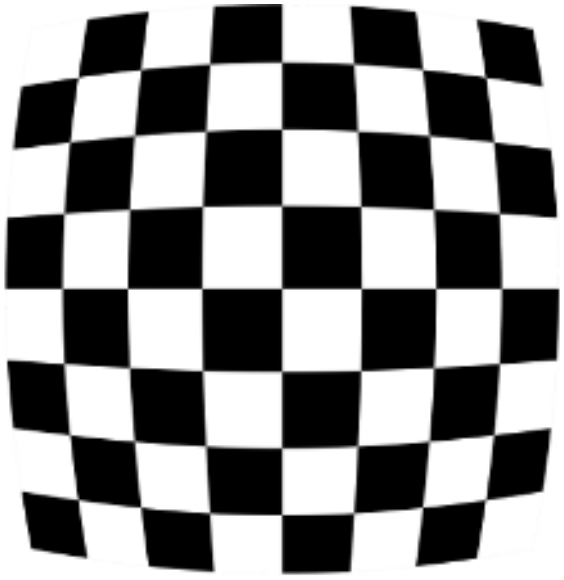
- This is, in fact, the central issue in the derivation of the camera model, that is, the identification of a set of corresponding ***control points***

- There are several approaches.
  - we could present the imaging system with a calibration grid
  - empirically measure the positions of the grid intersections
  - identifying the corresponding points in the image, either interactively or automatically

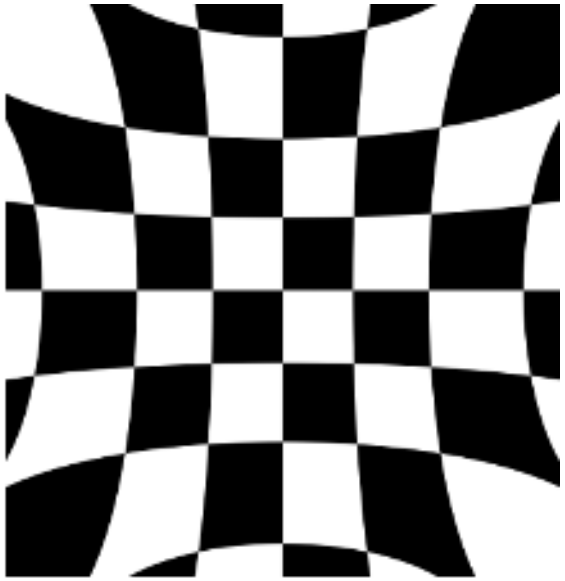
The Camera Model



No distortion

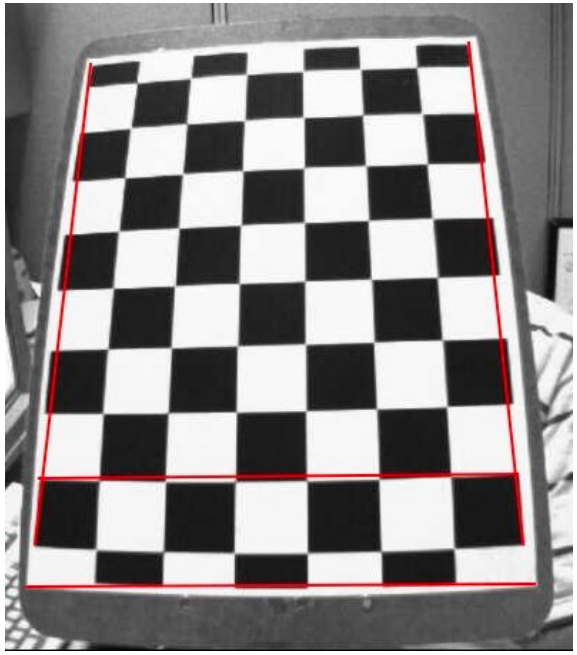


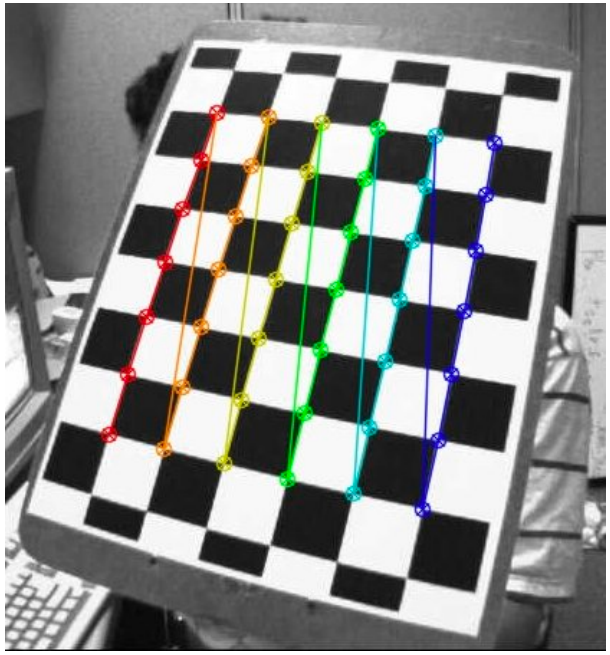
Positive radial distortion
(Barrel distortion)



Negative radial distortion
(Pincushion distortion)

The Camera Model



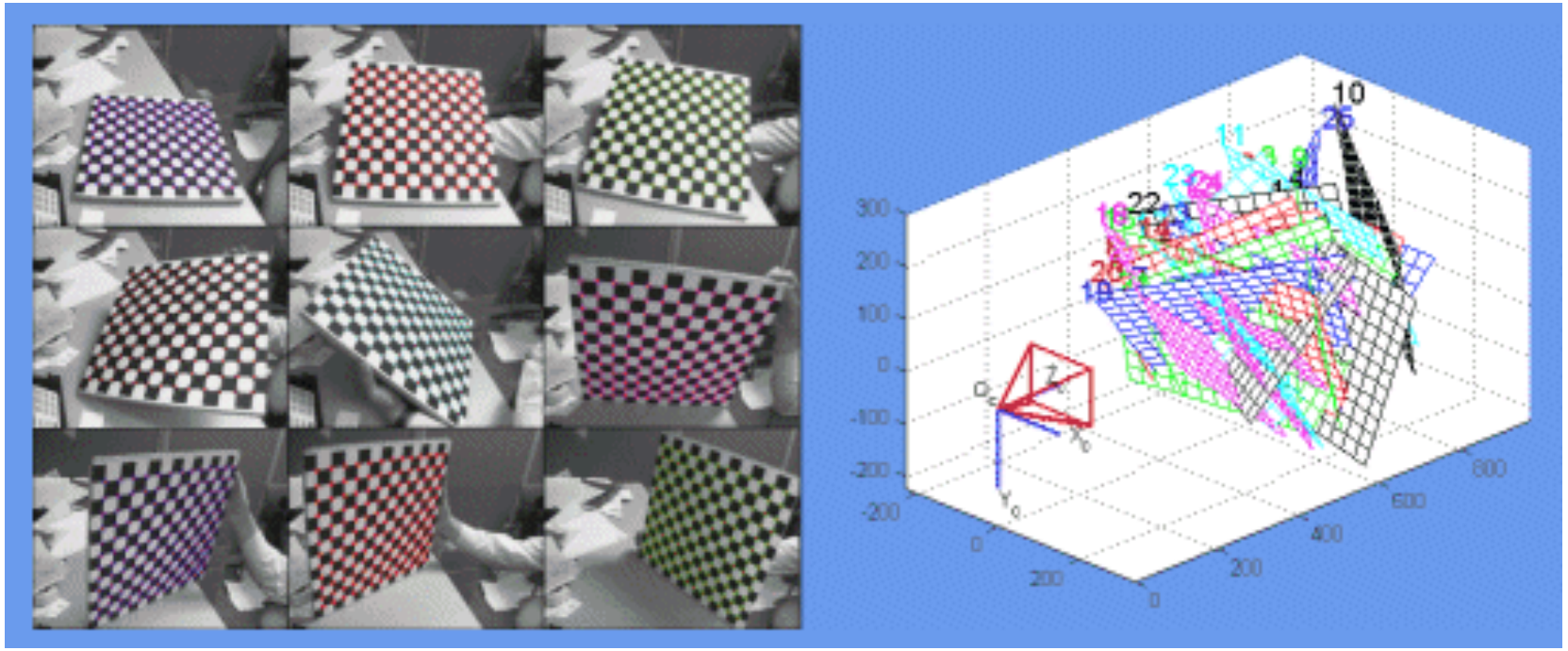


http://docs.opencv.org/3.1.0/dc/dbb/tutorial_py_calibration.html

The Camera Model

- Camera parameters from many planar points
  - Checkerboard or ellipse patterns on flat plate

- Camera pose from three or more points



http://www.vision.caltech.edu/bouguetj/calib_doc/

The Camera Model

- When used for robot manipulation, the empirical measurement of these real-world co-ordinates may be prone to error and this error will be manifested in the resultant camera model

- It is better practice to get the robot itself to calibrate the system

  – by fitting it with an end-effector with an accurately located calibration mark (*e.g.* a cross-hair or a surveyor's mark)

  – by programming it to place the mark at a variety of (known) positions in the field of view of the camera system

The Camera Model

Main benefit:

The two components of the manipulation environment,

the robot and the vision system

both of which are reasoning about co-ordinates in the 3D world, are effectively coupled

So, if the vision system “sees” something at a particular location, that is where the robot manipulator will go

The Camera Model

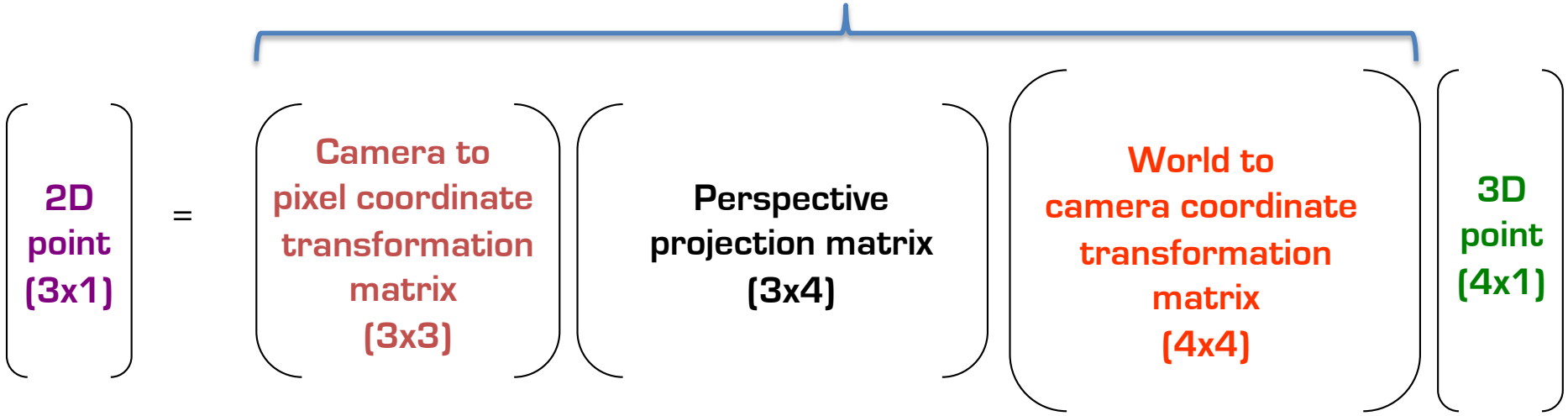
So far:

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ t \end{bmatrix}$$

The Camera Model

Intrinsic and extrinsic camera parameters

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ t \end{bmatrix}$$

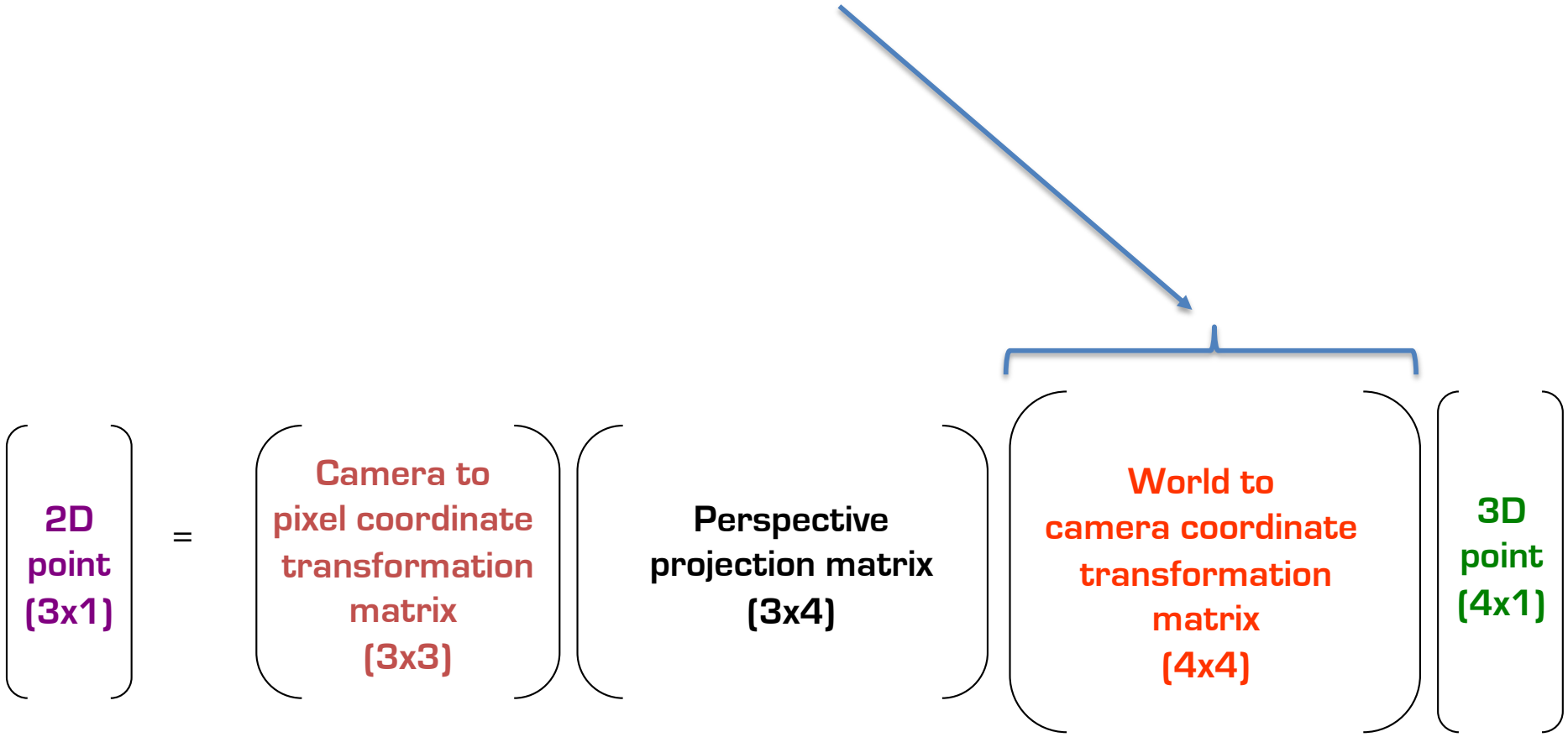


Credit: Markus Vincze, Technische Universität Wien

The Camera Model

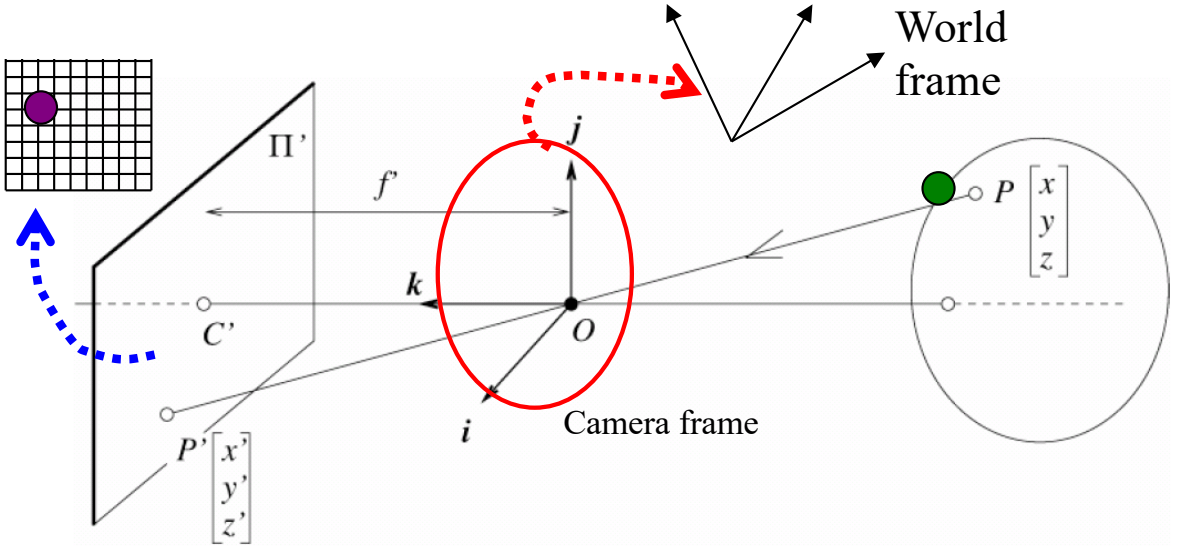
Extrinsic camera parameters

Transformation from world frame of reference to camera frame of reference



Credit: Markus Vincze, Technische Universität Wien

The Camera Model

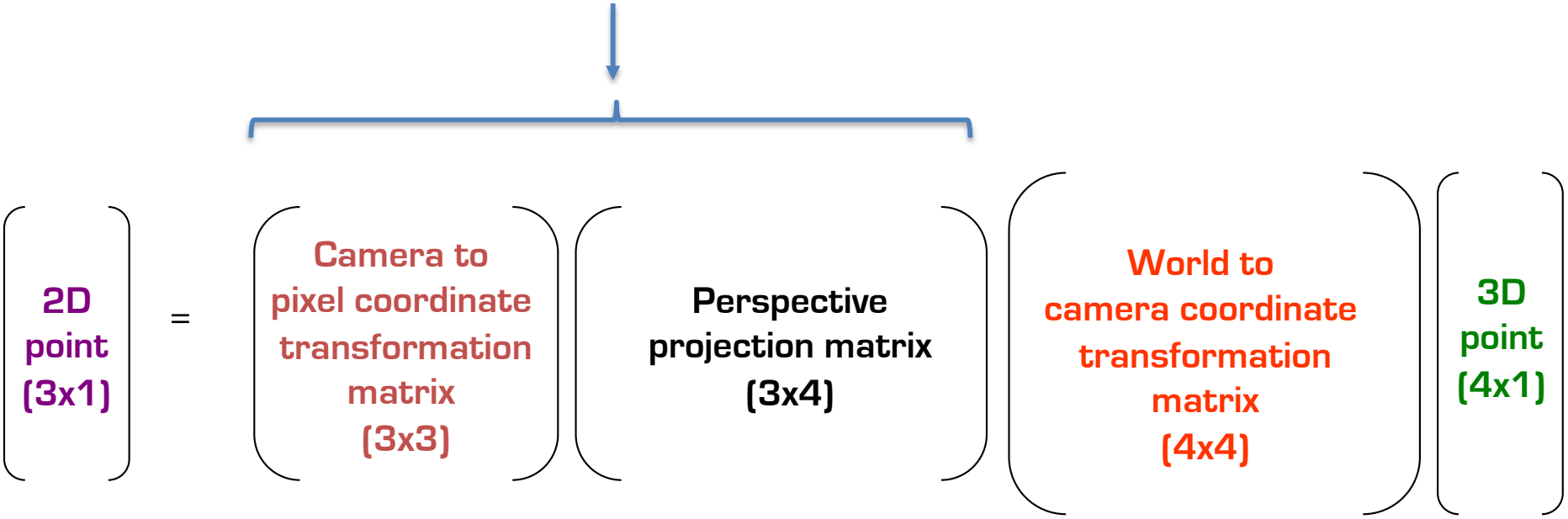


Credit: Markus Vincze, Technische Universität Wien

The Camera Model

Intrinsic camera parameters

- The perspective projection (only parameter is the focal length f)
- The transformation between camera frame coordinates and pixel coordinates
- The geometric distortion introduced by the optics

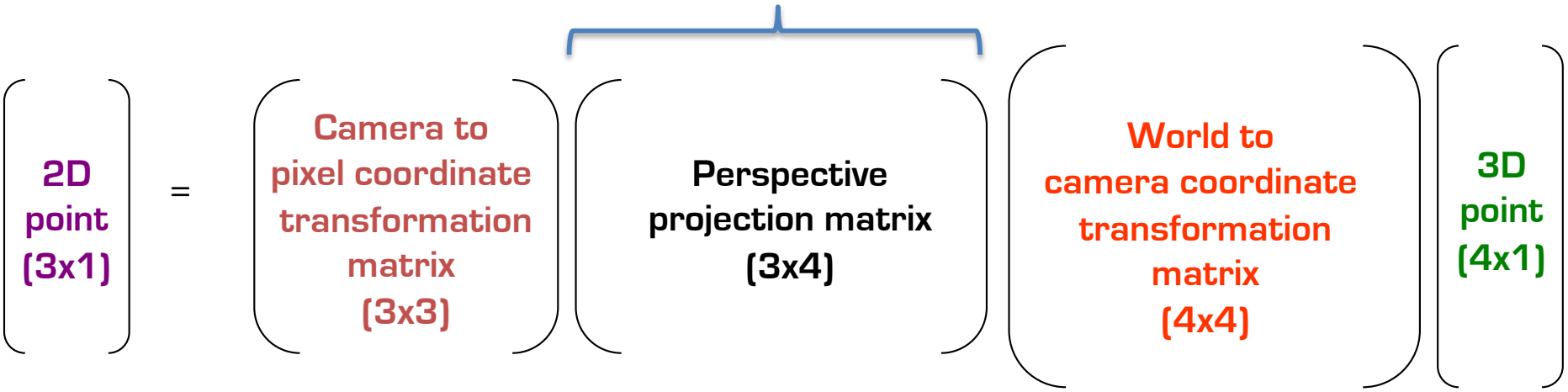


Credit: Markus Vincze, Technische Universität Wien

The Camera Model

Perspective Projection

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow (f\frac{x}{z}, f\frac{y}{z})$$



Credit: Markus Vincze, Technische Universität Wien

The Camera Model

Transformation from camera to pixel coordinates

$$(x, y) \rightarrow (x_{im}, y_{im})$$

$$x = - (x_{im} - c_x) s_x$$

$$y = - (y_{im} - c_y) s_y$$

(c_x, c_{iy})
coordinates of the image centre / principal point
(intersection of principal ray with image)

(s_x, s_{iy})
effective size of the pixel, in millimetres,
in the horizontal and vertical directions

$$\begin{pmatrix} \text{2D point (3x1)} \end{pmatrix} = \begin{pmatrix} \text{Camera to pixel coordinate transformation matrix (3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective projection matrix (3x4)} \end{pmatrix} \begin{pmatrix} \text{World to camera coordinate transformation matrix (4x4)} \end{pmatrix} \begin{pmatrix} \text{3D point (4x1)} \end{pmatrix}$$

Credit: Trucco and Verri 1998

The Camera Model

Correction of radial distortion

$$x = x_d (1 + k_1 r^2 + k_2 r^4)$$

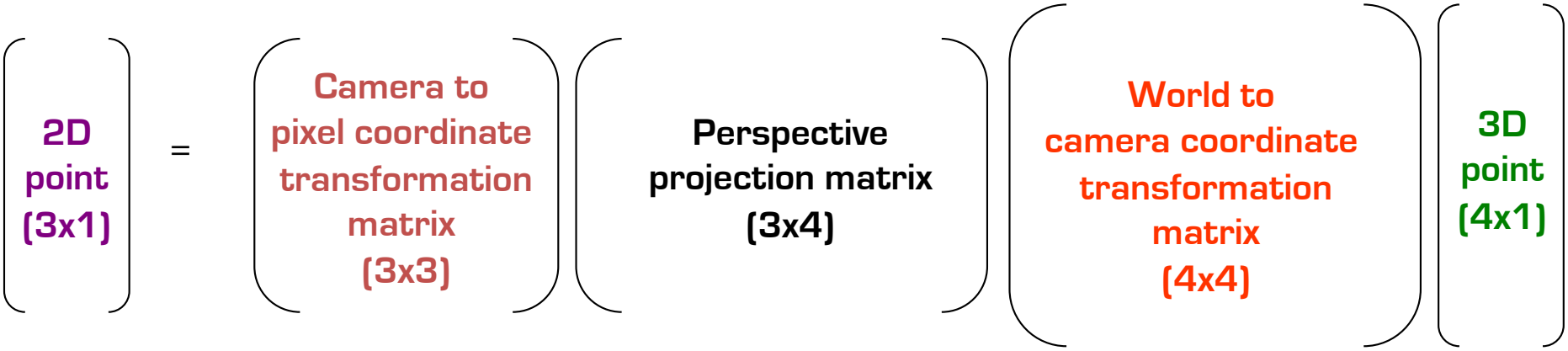
$$y = y_d (1 + k_1 r^2 + k_2 r^4)$$

$$r^2 = x_d^2 + y_d^2$$

(x_d, y_d)
coordinates of the distorted point

k_1 and k_2 are further intrinsic parameters

The magnitude of the geometric distortion depends on the quality of the lens

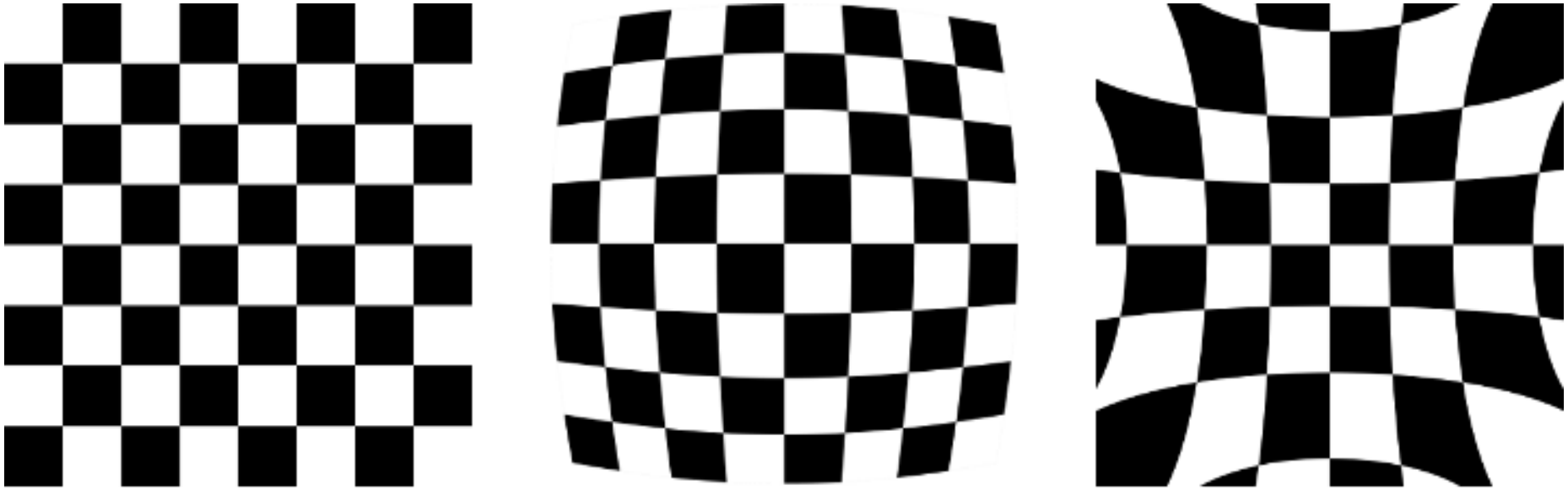


Credit: Trucco and Verri 1998

The Camera Model

$k_1 > 0$

$k_1 < 0$



No distortion

Positive radial distortion
(Barrel distortion)

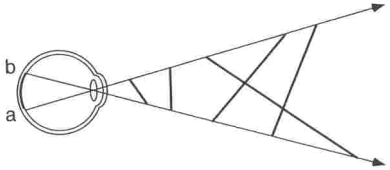
Negative radial distortion
(Pincushion distortion)

Credit: Markus Vincze, Technische Universität Wien

Depth Perception: The Inverse Problem

Estimate depth (3D) of world from images

Inverse perspective transformation



Credit: Markus Vincze, Technische Universität Wien

The Inverse Perspective Transformation

- Once the camera model *C* has been determined, we are now in a position to determine an expression for the co-ordinates of a point in the real world in terms of the co-ordinates of its imaged position

- Recalling equations (1) - (5) :

$$C_{11}x + C_{12}y + C_{13}z + C_{14} = u = Ut$$
$$C_{21}x + C_{22}x + C_{23}x + C_{24} = v = Vt$$
$$C_{31}x + C_{32}x + C_{33}x + C_{34} = t$$

The Inverse Perspective Transformation

Substituting the expression for t into the first two equations gives

$$U\left(C_{31}x + C_{32}y + C_{33}z + C_{34}C_{11}x\right) = C_{11}x + C_{12}y + C_{13}z + C_{14}$$
$$V\left(C_{31}x + C_{32}y + C_{33}z + C_{34}C_{11}x\right) = C_{21}x + C_{22}y + C_{23}z + C_{24}$$

Hence

$$\left(C_{11} - UC_{31}\right)x + \left(C_{12} - UC_{32}\right)y + \left(C_{13} - UC_{33}\right)z + \left(C_{14} - UC_{34}\right) = 0$$
$$\left(C_{21} - VC_{31}\right)x + \left(C_{22} - VC_{32}\right)y + \left(C_{23} - VC_{33}\right)z + \left(C_{24} - VC_{34}\right) = 0$$

The Inverse Perspective Transformation

Letting

$$a_1 = C_{11} - UC_{31}$$
$$b_1 = C_{12} - UC_{32}$$
$$c_1 = C_{13} - UC_{33}$$
$$d_1 = C_{14} - UC_{34}$$

and

$$a_2 = C_{21} - VC_{31}$$
$$b_2 = C_{22} - VC_{32}$$
$$c_2 = C_{23} - VC_{33}$$
$$d_2 = C_{24} - VC_{34}$$

we have

$$a_1x + b_1y + c_1z + d_1 = 0$$
$$a_2x + b_2y + c_2z + d_2 = 0$$

The Inverse Perspective Transformation

These are the equations of two planes

The intersection of these planes determines a line comprising the set of real-world points which project onto the image point $\begin{bmatrix} U \\ V \end{bmatrix}$

Solving these plane equations simultaneously (in terms of z)

$$x = \frac{z\left(b_1 c_2 - b_2 c_1\right) + \left(b_1 d_2 - b_2 d_1\right)}{\left(a_1 b_2 - a_2 b_1\right)}$$

$$y = \frac{z\left(a_2 c_1 - a_1 c_2\right) + \left(a_2 d_1 - a_1 d_2\right)}{\left(a_1 b_2 - a_2 b_1\right)}$$

The Inverse Perspective Transformation

Thus, for any given z_0, U and V, we may determine the corresponding x_0 and y_0, *i.e.* the real-world co-ordinates

The camera model and the inverse perspective transformation which we have just discussed allow us to compute the x and y real-world co-ordinates corresponding to a given position in the image

However, we must assume that the z coordinate, *i.e.* the distance from the camera, is known

The Inverse Perspective Transformation

- However, we must assume that the z coordinate, *i.e.* the distance from the camera, is known
- For some applications, e.g. where objects lie on a table at a given and constant height (*i.e.* at a given z_0), this is sufficient
- In general, however, we will not know the coordinate of the object in the third dimension and we must recover it

The Inverse Perspective Transformation

How we can compute z_0 ?

- If we have a second image of the scene, taken from another viewpoint
- If we know the image coordinates of the point of interest in this image
- Then we have two camera models and, hence, two inverse perspective transformations
- Instead of solving two plane equations simultaneously, we solve four plane equations

The Inverse Perspective Transformation

In particular, we have

$$a_1 x + b_1 y + c_1 z + d_1 = 0$$
$$a_2 x + b_2 y + c_2 z + d_2 = 0$$
$$p_1 x + q_1 y + r_1 z + s_1 = 0$$
$$p_2 x + q_2 y + r_2 z + s_2 = 0$$

where

$$a_1 = C1_{11} - U1C1_{31} \qquad a_2 = C1_{21} - V1C1_{31}$$
$$b_1 = C1_{12} - U1C1_{32} \qquad b_2 = C1_{22} - V1C1_{32}$$
$$c_1 = C1_{13} - U1C1_{33} \qquad c_2 = C1_{23} - V1C1_{33}$$
$$d_1 = C1_{14} - U1C1_{34} \qquad d_2 = C1_{24} - V1C1_{34}$$
$$p_1 = C2_{11} - U2C2_{31} \qquad p_2 = C2_{21} - V2C2_{31}$$
$$q_1 = C2_{12} - U2C2_{32} \qquad q_2 = C2_{22} - V2C2_{32}$$
$$r_1 = C2_{13} - U2C2_{33} \qquad r_2 = C2_{23} - V2C2_{33}$$
$$s_1 = C2_{14} - U2C2_{34} \qquad s_2 = C2_{24} - V2C2_{34}$$

$C1_{ij}$ and $C2_{ij}$ are the coefficients of the camera model for the first and second images, respectively.
$U1, V1$ and $U2, V2$ are the co-ordinates of the point of interest in the first and second images, respectively.

The Inverse Perspective Transformation

Since we now have four equations and three unknowns, the system is over-determined

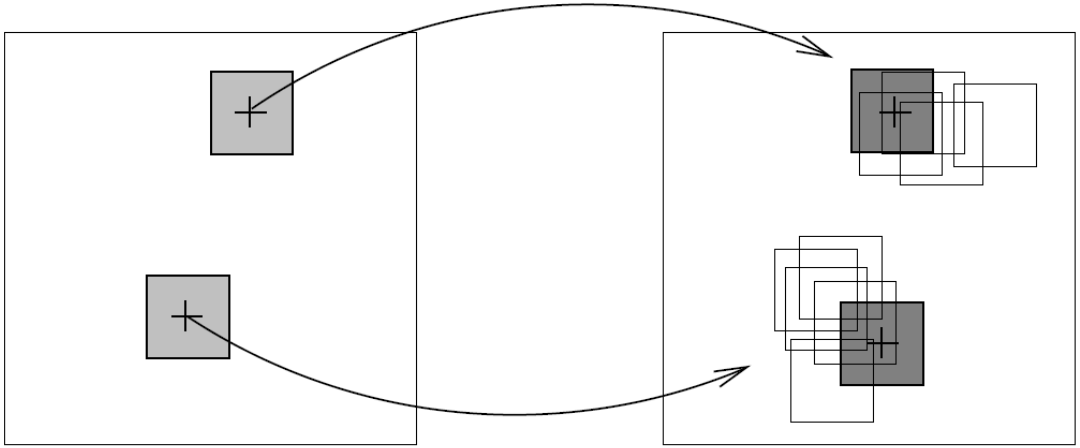
So, we compute a least-square-error solution using the pseudo-inverse technique

The Inverse Perspective Transformation

- It should be noted that the key here is not so much the mathematics which allows us to compute x_0 , y_0 and z_0 but, rather, the image analysis by which we identify the corresponding point of interest in the two images

- It is this correspondence problem which lies at the heart of most of the difficulties in recovery of depth information

The Inverse Perspective Transformation



[Young]

Credit: Markus Vincze, Technische Universität Wien

Demos

The following code is taken from the cameraCalibration project in the lectures directory of the ACV repository

See:

```
cameraCalibration.h
cameraCalibrationImplementation.cpp
cameraCalibrationApplication.cpp
```

Reading

R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

Section 2.1.5 3D to 2D projections

Section 6.3 Geometric intrinsic calibration

Vernon, D. 1991. Machine Vision: Automated Visual Inspection and Robot Vision, Prentice-Hall International; Section 8.6

Dawson-Howe, K. and Vernon, D. 1995. "Simple Pinhole Camera Calibration", The International Journal of Imaging Systems and Technology, Vol. 5, No. 1, pp. 1-6, Wiley, UK.

OpenCV documentation on camera calibration:

http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html