Camera Geometry

CSE 6367: Computer Vision

Instructor: William J. Beksi

Introduction

- We will focus on the geometry of a single perspective camera, specifically:

Introduction

- We will focus on the geometry of a single perspective camera, specifically:
  - How the projection of a 3D scene space onto a 2D image plane works

Introduction

- We will focus on the geometry of a single perspective camera, specifically:
    - How the projection of a 3D scene space onto a 2D image plane works
    - How to estimate the camera matrix given the coordinates of a set of corresponding world and image points

Introduction

- We will focus on the geometry of a single perspective camera, specifically:
  - How the projection of a 3D scene space onto a 2D image plane works
  - How to estimate the camera matrix given the coordinates of a set of corresponding world and image points
  - How the internal parameters of the camera matrix are computed for the purpose of calibration

The Basic Pinhole Model

- Consider the central projection of points in a space onto a plane where the center of projection is the origin of a Euclidean coordinate system

The Basic Pinhole Model

- Consider the central projection of points in a space onto a plane where the center of projection is the origin of a Euclidean coordinate system
- We denote the plane $z = f$ as the **image plane** or **focal plane**

The Basic Pinhole Model

- Consider the central projection of points in a space onto a plane where the center of projection is the origin of a Euclidean coordinate system
- We denote the plane $z = f$ as the **image plane** or **focal plane**
- Under the **pinhole camera model**, a point in space with coordinates $\mathbf{X} = [X, Y, Z]^T$ is mapped to the point on the image plane where a line joining $\mathbf{X}$ to the center of projection meets the image plane

The Basic Pinhole Model

- The center of projection is called the **camera center** (it's also known as the **optical center**)

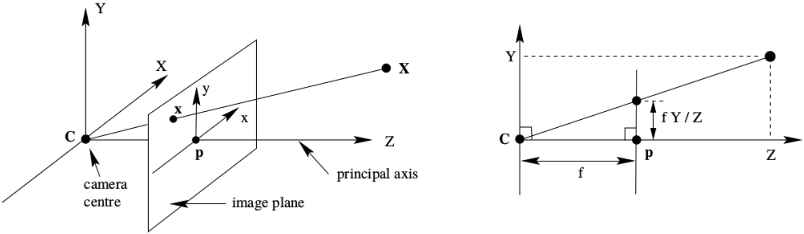
The Basic Pinhole Model

- The center of projection is called the **camera center** (it's also known as the **optical center**)
- The line from the camera center perpendicular to the image plane is called the **principle axis** or **principal ray** of the camera, and the point where the principal axis meets the image plane is called the **principal point**

The Basic Pinhole Model

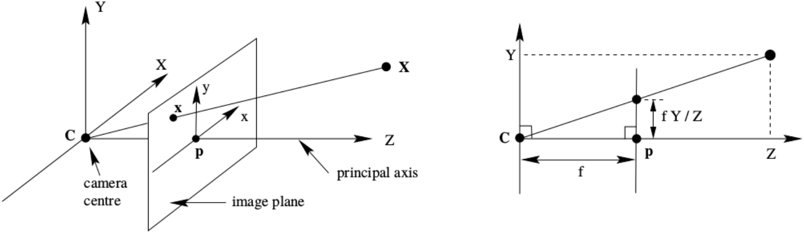
- The center of projection is called the **camera center** (it's also known as the **optical center**)
- The line from the camera center perpendicular to the image plane is called the **principle axis** or **principal ray** of the camera, and the point where the principal axis meets the image plane is called the **principal point**
- The plane through the camera center parallel to the image plane is called the **principle plane** of the camera

Pinhole Camera Geometry



- **C** is the camera center and **p** the principal point, the camera center here is placed at the origin

Pinhole Camera Geometry



- **C** is the camera center and **p** the principal point, the camera center here is placed at the origin
- Note the image plane is placed in front of the camera center

Central Projection Mapping

- By similar triangles, one can quickly compute that the point $[X, Y, Z]^T$ is mapped to the point $[fX/Z, fY/Z, f]^T$ on the image plane

Central Projection Mapping

- By similar triangles, one can quickly compute that the point $[X, Y, Z]^T$ is mapped to the point $[fX/Z, fY/Z, f]^T$ on the image plane
- Ignoring the final image coordinate, we see that

$$[X, Y, Z]^T \mapsto [fX/Z, fY/Z]^T \tag{1}$$

describes the central projection mapping from world to image coordinates

Central Projection Mapping

- By similar triangles, one can quickly compute that the point $[X, Y, Z]^T$ is mapped to the point $[fX/Z, fY/Z, f]^T$ on the image plane
- Ignoring the final image coordinate, we see that

$$[X, Y, Z]^T \mapsto [fX/Z, fY/Z]^T \tag{1}$$

  describes the central projection mapping from world to image coordinates
- This is a mapping from Euclidean 3-space $\mathbb{R}^3$ to Euclidean 2-space $\mathbb{R}^2$

Central Projection using Homogeneous Coordinates

- If the world and image points are represented by homogeneous vectors, then the central projection can be expressed as the following linear mapping

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

Central Projection using Homogeneous Coordinates

- If the world and image points are represented by homogeneous vectors, then the central projection can be expressed as the following linear mapping

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

- The matrix in this expression may be written as $\mathrm{diag}(f, f, 1)[I \mid \mathbf{0}]$ where $\mathrm{diag}(f, f, 1)$ is a diagonal matrix and $[I \mid \mathbf{0}]$ represents a matrix divided up into a 3×3 block plus a column vector

Central Projection using Homogeneous Coordinates

- Let $\mathbf{X}$ be the world point represented by the homogeneous 4-vector $[X, Y, Z, 1]^T$, let $\mathbf{x}$ be the image point represented by a homogeneous 3-vector, and let P be the 3×4 homogeneous **camera projection matrix**, then (2) can be written compactly as

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

which defines the camera matrix for the pinhole model of central projection as

$$P = \mathrm{diag}(f, f, 1)[I \mid \mathbf{0}]$$

Principal Point Offset

- Equation (1) assumes that the origin of the coordinates in the image plane is at the principal point, however in practice this may not be the case

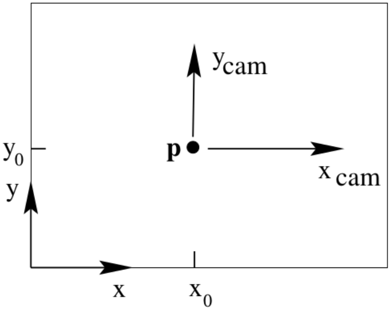
Principal Point Offset

- Equation (1) assumes that the origin of the coordinates in the image plane is at the principal point, however in practice this may not be the case
- Therefore, there is a mapping

$$[X, Y, Z]^T \mapsto [fX/Z + p_x, fY/Z + p_y]^T$$

where $[p_x, p_y]^T$ are the coordinates of the principle point

Principal Point Offset



- Image (x, y) and camera $(x_{\text{cam}}, y_{\text{cam}})$ coordinate systems

Principal Point Offset

- This equation expressing the mapping can be written in homogeneous coordinates as

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

and letting

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$

equation (3) has the concise form $\mathbf{x} = K[I \mid \mathbf{0}]\mathbf{X}_{\mathrm{cam}}$

Principal Point Offset

- The matrix K is called the **camera calibration matrix**

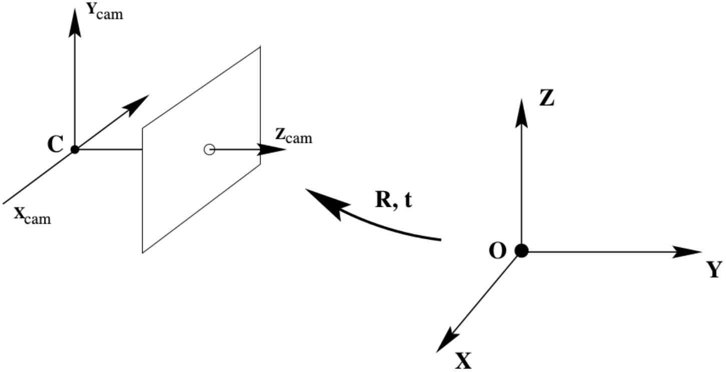
Principal Point Offset

- The matrix K is called the **camera calibration matrix**
- We write $[X, Y, Z, 1]^T$ as $\mathbf{X}_{\text{cam}}$ to emphasize that the camera is assumed to be located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the z-axis

Principal Point Offset

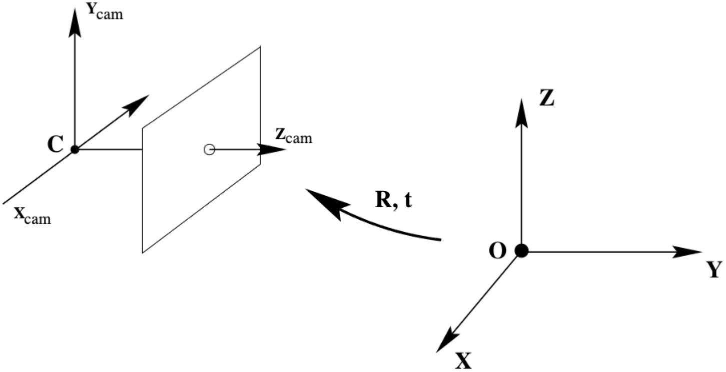
- The matrix K is called the **camera calibration matrix**
- We write $[X, Y, Z, 1]^T$ as $\mathbf{X}_{\text{cam}}$ to emphasize that the camera is assumed to be located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the z-axis
- Such a coordinate system is called the **camera coordinate frame**

Camera Rotation and Translation



- In general, we will express points in space in terms of a different Euclidean coordinate frame known as the **world coordinate frame**

Camera Rotation and Translation



- In general, we will express points in space in terms of a different Euclidean coordinate frame known as the **world coordinate frame**
- The two coordinate frames are related via a rotation and translation

Camera Rotation and Translation

- If $\tilde{\mathbf{X}}$ is an inhomogeneous 3-vector representing the coordinates of a point in the world frame and $\tilde{\mathbf{X}}_{\text{cam}}$ represents the same point in the camera frame, then we can write

$$\tilde{\mathbf{X}}_{\text{cam}} = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$$

where $\tilde{\mathbf{C}}$ represents the coordinates of the camera in the world coordinate frame

Camera Rotation and Translation

- If $\tilde{\mathbf{X}}$ is an inhomogeneous 3-vector representing the coordinates of a point in the world frame and $\tilde{\mathbf{X}}_{\text{cam}}$ represents the same point in the camera frame, then we can write

$$\tilde{\mathbf{X}}_{\text{cam}} = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$$

where $\tilde{\mathbf{C}}$ represents the coordinates of the camera in the world coordinate frame

- In homogeneous coordinates, this equation can be written as

$$\mathbf{X}_{\text{cam}} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}$$

Camera Rotation and Translation

- Thus, the general mapping of a pinhole camera is given by

$$\mathbf{x} = KR[I \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

where $\mathbf{X}$ is now in a world coordinate frame

Camera Rotation and Translation

- Thus, the general mapping of a pinhole camera is given by

$$\mathbf{x} = KR[I \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

  where $\mathbf{X}$ is now in a world coordinate frame

- The general pinhole camera model, $P = KR[I \mid -\tilde{\mathbf{C}}]$, has 9 degrees of freedom: 3 for K (f, p_x, p_y), 3 for R, and 3 for $\tilde{\mathbf{C}}$

Camera Rotation and Translation

- The parameters contained in K are called the **internal parameters** or the **internal orientation** of the camera

Camera Rotation and Translation

- The parameters contained in K are called the **internal parameters** or the **internal orientation** of the camera
- The parameters of R and $\tilde{\mathbf{C}}$ which relate the camera orientation and position to a world coordinate system are called the **external parameters** or the **exterior orientation**

Camera Rotation and Translation

- The parameters contained in K are called the **internal parameters** or the **internal orientation** of the camera
- The parameters of R and $\tilde{\mathbf{C}}$ which relate the camera orientation and position to a world coordinate system are called the **external parameters** or the **exterior orientation**
- It is often convenient not to make the camera center explicit, and instead to represent the world to image transformation as $\tilde{\mathbf{X}}_{\text{cam}} = R\tilde{\mathbf{X}} + \mathbf{t}$ and the camera matrix as

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

where $\mathbf{t} = -R\tilde{\mathbf{C}}$

CCD Cameras

- The pinhole camera model assumes that the image coordinates are Euclidean coordinates having equal scales in both axial directions

CCD Cameras

- The pinhole camera model assumes that the image coordinates are Euclidean coordinates having equal scales in both axial directions
- In the case of CCD cameras, there is the possibility of having non-square pixels

CCD Cameras

- The pinhole camera model assumes that the image coordinates are Euclidean coordinates having equal scales in both axial directions
- In the case of CCD cameras, there is the possibility of having non-square pixels
- If image coordinates are measured in pixels, then this has the extra effect of creating unequal scale factors in each direction

CCD Cameras

- If the number of pixels per unit distance in image coordinates are m_x and m_y in the x and y directions, then the transformation from world to pixel coordinates is obtained by multiplying K by an extra factor $\mathrm{diag}(m_x, m_y, 1)$

CCD Cameras

- If the number of pixels per unit distance in image coordinates are m_x and m_y in the x and y directions, then the transformation from world to pixel coordinates is obtained by multiplying K by an extra factor $\mathrm{diag}(m_x, m_y, 1)$
- Therefore, the general form of the calibration matrix of a CCD camera is

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal length of the camera in terms of pixel dimensions in the x and y directions respectively

Finite Projective Camera

- For added generality, we can consider a calibration matrix of the form

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{4}$$

Finite Projective Camera

- For added generality, we can consider a calibration matrix of the form

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{4}$$

- The added parameter s is referred to as the **skew** parameter and will be zero for most normal cameras

Finite Projective Camera

- A camera

$$P = KR[I \mid -\tilde{\mathbf{C}}] \tag{5}$$

for which the calibration matrix K is of the form (4) is called a **finite projective camera**

Finite Projective Camera

- A camera

$$P = KR[I \mid -\tilde{\mathbf{C}}] \tag{5}$$

for which the calibration matrix K is of the form (4) is called a **finite projective camera**

- A finite projective camera has 11 degrees of freedom, the same as a 3×4 matrix defined up to an arbitrary scale

Finite Projective Camera

- A camera

$$P = KR[I \mid -\tilde{\mathbf{C}}] \tag{5}$$

for which the calibration matrix K is of the form (4) is called a **finite projective camera**

- A finite projective camera has 11 degrees of freedom, the same as a 3×4 matrix defined up to an arbitrary scale

- Letting $M = KR$, P can be written as

$$P = M[I \mid M^{-1}\mathbf{p}_4] = KR[I \mid -\tilde{\mathbf{C}}]$$

where $\mathbf{p}_4$ is the last column of P

General Projective Cameras

- If we remove the non-singularity restriction on the left hand 3×3 matrix KR, then we have the form of a **general projective camera**

General Projective Cameras

- If we remove the non-singularity restriction on the left hand 3×3 matrix KR, then we have the form of a **general projective camera**
- A general projective matrix is represented by an arbitrary 3×4 matrix of rank 3 and has 11 degrees of freedom

General Projective Cameras

- If we remove the non-singularity restriction on the left hand 3×3 matrix KR, then we have the form of a **general projective camera**
- A general projective matrix is represented by an arbitrary 3×4 matrix of rank 3 and has 11 degrees of freedom
- The rank 3 requirement arises because if the rank is less, then the range of the matrix mapping will be a line or point and not the whole plane (i.e. not a 2D image)

The Projective Camera

- A general projective camera P maps world points $\mathbf{X}$ to image points $\mathbf{x}$ according to $\mathbf{x} = P\mathbf{X}$

The Projective Camera

- A general projective camera P maps world points $\mathbf{X}$ to image points $\mathbf{x}$ according to $\mathbf{x} = P\mathbf{X}$
- It can be decomposed into blocks according to $P = [M \mid \mathbf{p}_4]$, where M is a 3×3 matrix

The Projective Camera

- A general projective camera P maps world points $\mathbf{X}$ to image points $\mathbf{x}$ according to $\mathbf{x} = P\mathbf{X}$
- It can be decomposed into blocks according to $P = [M \mid \mathbf{p}_4]$, where M is a 3×3 matrix
- If M is non-singular, then this is a finite camera otherwise it is not

Camera Center

- Consider the line containing $\mathbf{C}$ and any other point $\mathbf{A}$ in 3-space, points on this line may be represented by the join

$$\mathbf{X}(\lambda) = \lambda\mathbf{A} + (1-\lambda)\mathbf{C}$$

Camera Center

- Consider the line containing $\mathbf{C}$ and any other point $\mathbf{A}$ in 3-space, points on this line may be represented by the join

$$\mathbf{X}(\lambda) = \lambda\mathbf{A} + (1-\lambda)\mathbf{C}$$

- Under the mapping $\mathbf{x} = P\mathbf{X}$ points on this line are projected to

$$\mathbf{x} = P\mathbf{X}(\lambda) = \lambda P\mathbf{A} + (1-\lambda)P\mathbf{C} = \lambda P\mathbf{A}$$

  since $P\mathbf{C} = \mathbf{0}$ (i.e. all points on the line are mapped to the same image point $P\mathbf{A}$, which means that the line must be a ray through the **camera center**)

Column Vectors

- The column vectors of the projective camera are 3-vectors which have a geometric meaning as particular image points

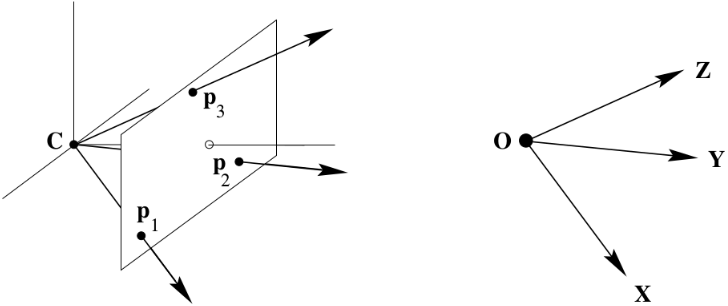
Column Vectors

- The column vectors of the projective camera are 3-vectors which have a geometric meaning as particular image points
- Let the columns of P be $\mathbf{p}_i, i = 1, \ldots, 4$, then $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are the vanishing points of the world coordinate X, Y, and Z axes respectively

Column Vectors

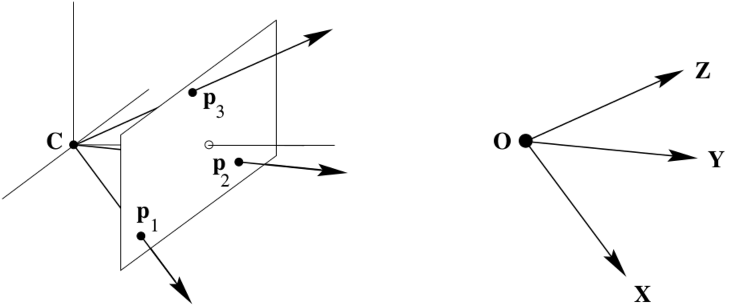
- The column vectors of the projective camera are 3-vectors which have a geometric meaning as particular image points
- Let the columns of P be $\mathbf{p}_i, i = 1, \ldots, 4$, then $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are the vanishing points of the world coordinate X, Y, and Z axes respectively
- This follows because these points are the images of the axes' directions

Column Vectors



- The three image points defined by the columns $\mathbf{p}_i, i = 1, \ldots, 3$, of the projection matrix are the vanishing points of the directions of the world axes

Column Vectors



- The three image points defined by the columns $\mathbf{p}_i, i = 1, \ldots, 3$, of the projection matrix are the vanishing points of the directions of the world axes
- The column $\mathbf{p}_4$ is the image of the world origin

Row Vectors

- The row vectors of the projective camera are 4-vectors which may be interpreted geometrically as particular world planes

Row Vectors

- The row vectors of the projective camera are 4-vectors which may be interpreted geometrically as particular world planes
- We express the rows of P as $\mathbf{P}^{iT}$ so that

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{1T} \\ \mathbf{P}^{2T} \\ \mathbf{P}^{3T} \end{bmatrix}$$

The Principal Plane

- The **principal plane** is the plane through the camera center parallel to the image plane

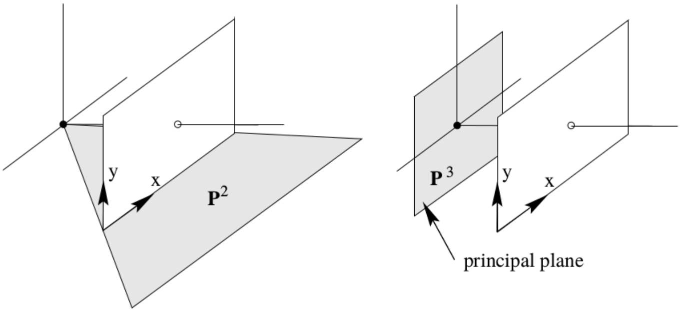
The Principal Plane

- The **principal plane** is the plane through the camera center parallel to the image plane
- It consists of the set of points $\mathbf{X}$ which are imaged on the line at infinity of the image, explicitly $P\mathbf{X} = [x, y, 0]^T$

The Principal Plane

- The **principal plane** is the plane through the camera center parallel to the image plane
- It consists of the set of points $\mathbf{X}$ which are imaged on the line at infinity of the image, explicitly $P\mathbf{X} = [x, y, 0]^T$
- Thus, a point lies on the principal plane of the camera iff $\mathbf{P}^{3T}\mathbf{X} = 0$, e.g. $\mathbf{P}^3$ is the vector representing the principal plane of the camera

The Principal Plane



- Two of the three planes defined by the rows of the projection matrix

Axis Planes

- Unlike the principal plane $\mathbf{P}^3$, the **axis planes** $\mathbf{P}^1$ and $\mathbf{P}^2$ are dependent on the image x- and y-axes, i.e. on the choice of the image coordinate system

Axis Planes

- Unlike the principal plane $\mathbf{P}^3$, the **axis planes** $\mathbf{P}^1$ and $\mathbf{P}^2$ are dependent on the image x- and y-axes, i.e. on the choice of the image coordinate system
- Therefore, they are less tightly coupled to the natural geometry than the principal plane

Axis Planes

- Unlike the principal plane $\mathbf{P}^3$, the **axis planes** $\mathbf{P}^1$ and $\mathbf{P}^2$ are dependent on the image x- and y-axes, i.e. on the choice of the image coordinate system
- Therefore, they are less tightly coupled to the natural geometry than the principal plane
- In particular, the line of intersection of $\mathbf{P}^1$ and $\mathbf{P}^2$ is a line joining the camera center and image origin (i.e. the back-projection of the image origin) and generally this line will not coincide with the camera principal axis

The Principal Point

- The principal axis is the line passing through the camera center $\mathbf{C}$, with direction perpendicular to the principal plane $\mathbf{P}^3$, and intersects the image plane at the **principal point**

The Principal Point

- The principal axis is the line passing through the camera center $\mathbf{C}$, with direction perpendicular to the principal plane $\mathbf{P}^3$, and intersects the image plane at the **principal point**
- In the case of $\mathbf{P^3}$, this point is $[p_{31}, p_{32}, p_{33}, 0]^T$ which we denote by $\hat{\mathbf{P}}^3$

The Principal Point

- The principal axis is the line passing through the camera center $\mathbf{C}$, with direction perpendicular to the principal plane $\mathbf{P}^3$, and intersects the image plane at the **principal point**
- In the case of $\mathbf{P^3}$, this point is $[p_{31}, p_{32}, p_{33}, 0]^T$ which we denote by $\hat{\mathbf{P}}^3$
- The principal point can be computed as $\mathbf{x}_0 = M\mathbf{m}^3$ (where $\mathbf{m}^{3T}$ is the third row of M) and projecting that point using P gives the principal point of the camera $P\hat{\mathbf{P}}^3$

The Principal Axis Vector

- Although any point $\mathbf{X}$ not on the principal plane may be mapped to an image point according to $\mathbf{x} = P\mathbf{X}$, in reality only half the points in space (those that lie in front of the camera) may be seen in an image

The Principal Axis Vector

- Although any point $\mathbf{X}$ not on the principal plane may be mapped to an image point according to $\mathbf{x} = P\mathbf{X}$, in reality only half the points in space (those that lie in front of the camera) may be seen in an image
- Using the equation for projection of a 3D point to an image point given by $\mathbf{x} = P_{\text{cam}}\mathbf{X}_{\text{cam}} = K[I \mid \mathbf{0}]\mathbf{X}_{\text{cam}}$, we observe that the vector $\mathbf{v} = \det(M)\mathbf{m}^3 = [0, 0, 1]^T$ points *towards the front of the camera* in the direction of the principal axis (irrespective of the scaling of P_{cam})

The Principal Axis Vector

- Although any point $\mathbf{X}$ not on the principal plane may be mapped to an image point according to $\mathbf{x} = P\mathbf{X}$, in reality only half the points in space (those that lie in front of the camera) may be seen in an image
- Using the equation for projection of a 3D point to an image point given by $\mathbf{x} = P_{\text{cam}}\mathbf{X}_{\text{cam}} = K[I \mid \mathbf{0}]\mathbf{X}_{\text{cam}}$, we observe that the vector $\mathbf{v} = \det(M)\mathbf{m}^3 = [0, 0, 1]^T$ points *towards the front of the camera* in the direction of the principal axis (irrespective of the scaling of P_{cam})
- If the 3D point is expressed in world coordinates then $P = kK[R \mid -R\tilde{\mathbf{C}}] = [M \mid \mathbf{p}_4]$ where $M = kKR$

Forward Projection

- A general projective camera maps a point in space $\mathbf{X}$ to an image point according to the mapping $\mathbf{x} = P\mathbf{X}$

Forward Projection

- A general projective camera maps a point in space $\mathbf{X}$ to an image point according to the mapping $\mathbf{x} = P\mathbf{X}$
- Points $\mathbf{D} = [\mathbf{d}^T, 0]^T$ on the plane at infinity represent vanishing points and map to

$$\mathbf{x} = P\mathbf{D} = [M \,|\, \mathbf{p}_4]\mathbf{D} = M\mathbf{d}$$

and thus are only affected by M, the first 3×3 submatrix of P

Back-Projection of Points to Rays

- Given a point $\mathbf{x}$ in an image, we want to determine the set of points in space that map to this point

Back-Projection of Points to Rays

- Given a point $\mathbf{x}$ in an image, we want to determine the set of points in space that map to this point
- This set will constitute a ray in space passing through the camera center

Back-Projection of Points to Rays

- Given a point $\mathbf{x}$ in an image, we want to determine the set of points in space that map to this point
- This set will constitute a ray in space passing through the camera center
- The form of the ray may be specified in several ways (depending on how we want to represent a line in 3-space), here we'll represent the line as the join of two points

Back-Projection of Points to Rays

- We know two points on the ray: the camera center $\mathbf{C}$ (where $P\mathbf{C} = 0$) and $P^{\dagger}\mathbf{x}$ (where $P^{\dagger}$ is the pseudoinverse of P)

Back-Projection of Points to Rays

- We know two points on the ray: the camera center $\mathbf{C}$ (where $P\mathbf{C} = 0$) and $P^{\dagger}\mathbf{x}$ (where $P^{\dagger}$ is the pseudoinverse of P)
- The pseudoinverse of P is the matrix $P^{\dagger} = P^T(PP^T)^{-1}$, for which $PP^{\dagger} = I$

Back-Projection of Points to Rays

- We know two points on the ray: the camera center $\mathbf{C}$ (where $P\mathbf{C} = 0$) and $P^\dagger \mathbf{x}$ (where $P^\dagger$ is the pseudoinverse of P)
- The pseudoinverse of P is the matrix $P^\dagger = P^T(PP^T)^{-1}$, for which $PP^\dagger = I$
- $P^\dagger \mathbf{x}$ lies on the ray because it projects to $\mathbf{x}$, since $P(P^\dagger \mathbf{x}) = I\mathbf{x} = \mathbf{x}$

Back-Projection of Points to Rays

- We know two points on the ray: the camera center $\mathbf{C}$ (where $P\mathbf{C} = 0$) and $P^{\dagger}\mathbf{x}$ (where $P^{\dagger}$ is the pseudoinverse of P)
- The pseudoinverse of P is the matrix $P^{\dagger} = P^T(PP^T)^{-1}$, for which $PP^{\dagger} = I$
- $P^{\dagger}\mathbf{x}$ lies on the ray because it projects to $\mathbf{x}$, since $P(P^{\dagger}\mathbf{x}) = I\mathbf{x} = \mathbf{x}$
- Then the ray is the line formed by the join of these two points

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

Back-Projection of Points to Rays

- In the case of finite cameras an alternative expression can be developed; writing $P = [M \,|\, \mathbf{p}_4]$, the camera center is given by $\tilde{\mathbf{C}} = -M^{-1}\mathbf{p}_4$

Back-Projection of Points to Rays

- In the case of finite cameras an alternative expression can be developed; writing $P = [M \mid \mathbf{p}_4]$, the camera center is given by $\tilde{\mathbf{C}} = -M^{-1}\mathbf{p}_4$
- An image point $\mathbf{x}$ back-projects to a ray intersecting the plane at infinity at the point $D = [(M^{-1}\mathbf{x})^T, 0]^T$, and provides a second point on the ray

Back-Projection of Points to Rays

- In the case of finite cameras an alternative expression can be developed; writing $P = [M \,|\, \mathbf{p}_4]$, the camera center is given by $\tilde{\mathbf{C}} = -M^{-1}\mathbf{p}_4$
- An image point $\mathbf{x}$ back-projects to a ray intersecting the plane at infinity at the point $D = [(M^{-1}\mathbf{x})^T, 0]^T$, and provides a second point on the ray
- Writing the line as the join of two points on the ray we have

$$\mathbf{X}(\mu) = \mu \begin{bmatrix} M^{-1}\mathbf{x} \\ 0 \end{bmatrix} + \begin{bmatrix} M^{-1}\mathbf{p}_4 \\ 1 \end{bmatrix} = \begin{bmatrix} M^{-1}(\mu\mathbf{x} - \mathbf{p}_4) \\ 1 \end{bmatrix}$$

Depth of Points

- We want to determine the distance a point lies in front of or behind the principal plane of the camera

Depth of Points

- We want to determine the distance a point lies in front of or behind the principal plane of the camera
- Consider a camera matrix $P = [M \mid \mathbf{p}_4]$, projecting a point $\mathbf{X} = [X, Y, Z, 1]^T = [\tilde{\mathbf{X}}, 1]^T$ in 3-space to the image point $\mathbf{x} = w[x, y, 1]^T = P\mathbf{X}$

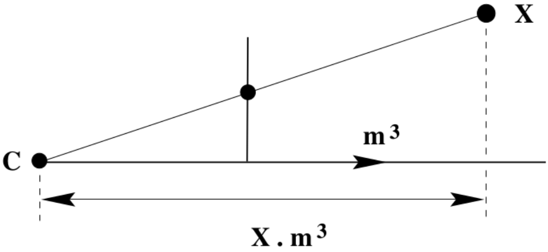
Depth of Points

- We want to determine the distance a point lies in front of or behind the principal plane of the camera
- Consider a camera matrix $P = [M \,|\, \mathbf{p}_4]$, projecting a point $\mathbf{X} = [X, Y, Z, 1]^T = [\tilde{\mathbf{X}}, 1]^T$ in 3-space to the image point $\mathbf{x} = w[x, y, 1]^T = P\mathbf{X}$
- Then $w = \mathbf{P}^{3T}\mathbf{X} = \mathbf{P}^{3T}(\mathbf{X} - \mathbf{C})$ since $P\mathbf{C} = 0$ for the camera center $\mathbf{C}$

Depth of Points

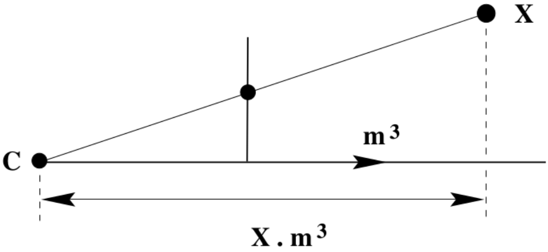
- We want to determine the distance a point lies in front of or behind the principal plane of the camera
- Consider a camera matrix $P = [M \,|\, \mathbf{p}_4]$, projecting a point $\mathbf{X} = [X, Y, Z, 1]^T = [\tilde{\mathbf{X}}, 1]^T$ in 3-space to the image point $\mathbf{x} = w[x, y, 1]^T = P\mathbf{X}$
- Then $w = \mathbf{P}^{3T}\mathbf{X} = \mathbf{P}^{3T}(\mathbf{X} - \mathbf{C})$ since $P\mathbf{C} = 0$ for the camera center $\mathbf{C}$
- However, $\mathbf{P}^{3T}(\mathbf{X} - \mathbf{C}) = \mathbf{m}^{3T}(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$ where $\mathbf{m}^3$ is the principal ray direction, so $w = \mathbf{m}^{3T}(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$ can be interpreted as the dot product of the ray from the camera center to the point $\mathbf{X}$, with the principal ray direction

Depth of Points



- If the camera matrix is normalized so that $\det(M) > 0$ and $||\mathbf{m}^3|| = 1$, then $\mathbf{m}^3$ is a unit vector pointing in the *positive* axial direction

Depth of Points



- If the camera matrix is normalized so that $\det(M) > 0$ and $||\mathbf{m}^3|| = 1$, then $\mathbf{m}^3$ is a unit vector pointing in the *positive* axial direction
- Thus w may be interpreted as the depth of the point $\mathbf{X}$ from the camera center $\mathbf{C}$ in the direction of the principal ray

Depth of Points

- Any camera matrix may be normalized by multiplying it by an appropriate factor

Depth of Points

- Any camera matrix may be normalized by multiplying it by an appropriate factor
- However, to avoid having to always deal with normalized camera matrices, the depth of a point can be computed as follows

Depth of Points

- Any camera matrix may be normalized by multiplying it by an appropriate factor
- However, to avoid having to always deal with normalized camera matrices, the depth of a point can be computed as follows
- Let $\mathbf{X} = [X, Y, Z, T]^T$ be a 3D point, $P = [M \mid \mathbf{p}_4]$ be a camera matrix for a finite camera, and suppose $P[X, Y, Z, T]^T = w[x, y, 1]^T$, then

$$\text{depth}(\mathbf{X}; P) = \frac{\text{sign}(\det(M))w}{T\|\mathbf{m}^3\|}$$

  is the depth of the point $\mathbf{X}$ in front of the principal plane of the camera

Finding the Camera Center

- The camera center $\mathbf{C}$ is the point for which $P\mathbf{C} = 0$

Finding the Camera Center

- The camera center $\mathbf{C}$ is the point for which $P\mathbf{C} = 0$
- Numerically, this right null vector can be obtained from the SVD of P

Finding the Camera Center

- The camera center $\mathbf{C}$ is the point for which $P\mathbf{C} = 0$
- Numerically, this right null vector can be obtained from the SVD of P
- Algebraically, the center $\mathbf{C} = [X, Y, Z, T]^T$ can be obtained as

$$X = \det([\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]) \, Y = -\det([\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4])$$
$$Z = \det([\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4]) \, T = -\det([\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3])$$

Finding the Camera Orientation and Internal Parameters

- In the case of a finite camera

$$P = [M \mid -M\tilde{\mathbf{C}}] = K[R \mid -R\tilde{\mathbf{C}}]$$

Finding the Camera Orientation and Internal Parameters

- In the case of a finite camera

$$P = [M \mid -M\tilde{\mathbf{C}}] = K[R \mid -R\tilde{\mathbf{C}}]$$

- We can easily find both K and R by decomposing M as $M = KR$ using the **RQ-decomposition**

Finding the Camera Orientation and Internal Parameters

- In the case of a finite camera

$$P = [M \mid -M\tilde{\mathbf{C}}] = K[R \mid -R\tilde{\mathbf{C}}]$$

- We can easily find both K and R by decomposing M as $M = KR$ using the **RQ-decomposition**
- The matrix R gives the orientation of the camera whereas K is the calibration matrix

Finding the Camera Orientation and Internal Parameters

- The matrix K has the form

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

where

Finding the Camera Orientation and Internal Parameters

- The matrix K has the form

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

where

  - α_x is the scale factor in the x-coordinate direction

Finding the Camera Orientation and Internal Parameters

- The matrix K has the form

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

  where

  - α_x is the scale factor in the x-coordinate direction
  - α_y is the scale factor in the y-coordinate direction

Finding the Camera Orientation and Internal Parameters

- The matrix K has the form

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

  where

  - α_x is the scale factor in the x-coordinate direction
  - α_y is the scale factor in the y-coordinate direction
  - s is the skew

Finding the Camera Orientation and Internal Parameters

- The matrix K has the form

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

  where

  - α_x is the scale factor in the x-coordinate direction
  - α_y is the scale factor in the y-coordinate direction
  - s is the skew
  - $[x_0, y_0]^T$ are the coordinates of the principal point

Finding the Camera Orientation and Internal Parameters

- The matrix K has the form

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

  where
  - α_x is the scale factor in the x-coordinate direction
  - α_y is the scale factor in the y-coordinate direction
  - s is the skew
  - $[x_0, y_0]^T$ are the coordinates of the principal point

- The *aspect ratio* is α_x/α_y

Givens Rotations and RQ Decomposition

- A 3D **Givens rotation** is a rotation about one of the three coordinate axes

Givens Rotations and RQ Decomposition

- A 3D **Givens rotation** is a rotation about one of the three coordinate axes
- The three Givens rotations are

$$Q_x = \begin{bmatrix} 1 & & \\ & c & -s \\ & s & c \end{bmatrix}, \quad Q_y = \begin{bmatrix} c & & s \\ & 1 & \\ -s & & c \end{bmatrix}, \quad Q_z = \begin{bmatrix} c & -s & \\ s & c & \\ & & 1 \end{bmatrix}$$

where $c = \cos\theta$ and $s = \sin\theta$ for some angle θ and blank entries represent zeros

Givens Rotations and RQ Decomposition

- A 3D **Givens rotation** is a rotation about one of the three coordinate axes
- The three Givens rotations are

$$Q_x = \begin{bmatrix} 1 & & \\ & c & -s \\ & s & c \end{bmatrix}, \quad Q_y = \begin{bmatrix} c & & s \\ & 1 & \\ -s & & c \end{bmatrix}, \quad Q_z = \begin{bmatrix} c & -s & \\ s & c & \\ & & 1 \end{bmatrix}$$

  where $c = \cos\theta$ and $s = \sin\theta$ for some angle θ and blank entries represent zeros
- The strategy of the **RQ algorithm** is to clear out the lower half of a matrix A one entry at a time by multiplication by Givens rotations

RQ Algorithm

<u>Objective</u>

Carry out the RQ decomposition of a 3×3 matrix $\mathtt{A}$ using Givens rotations.

<u>Algorithm</u>

(i) Multiply by $\mathtt{Q}_x$ so as to set $\mathtt{A}_{32}$ to zero.
(ii) Multiply by $\mathtt{Q}_y$ so as to set $\mathtt{A}_{31}$ to zero. This multiplication does not change the second column of $\mathtt{A}$, hence $\mathtt{A}_{32}$ remains zero.
(iii) Multiply by $\mathtt{Q}_z$ so as to set $\mathtt{A}_{21}$ to zero. The first two columns are replaced by linear combinations of themselves. Thus, $\mathtt{A}_{31}$ and $\mathtt{A}_{32}$ remain zero.

- The algorithm for performing the RQ decomposition of a 3×3 matrix

RQ Decomposition in MATLAB

- The RQ decomposition can be obtained from the QR decomposition

$$M = (QR)^{-1} = R^{-1}Q^{-1}$$

RQ Decomposition in MATLAB

- The RQ decomposition can be obtained from the QR decomposition

$$M = (QR)^{-1} = R^{-1}Q^{-1}$$

- In MATLAB, we can implement this as

```
function [R,Q] = rq(M)
  [Q,R] = qr(rot90(M,3));
 R = rot90(R,2)';
 Q = rot90(Q);
```

Example: Computing K and R

- Let the camera matrix be

$$P = \begin{bmatrix} 3.5355e+2 & 3.3964e+2 & 2.7774e+2 & -1.4495e+6 \\ -1.0353e+2 & 2.3321e+1 & 4.5961e+2 & -6.3252e+5 \\ 7.0711e-1 & -3.5355e-1 & 6.1237e-1 & -9.1856e+02 \end{bmatrix}$$

with $P = [M \,|\, -M\tilde{\mathbf{C}}]$ and center $\tilde{\mathbf{C}} = [1000.0, 2000.0, 1500.0]^T$

Example: Computing K and R

- Let the camera matrix be

$$P = \begin{bmatrix} 3.5355e+2 & 3.3964e+2 & 2.7774e+2 & -1.4495e+6 \\ -1.0353e+2 & 2.3321e+1 & 4.5961e+2 & -6.3252e+5 \\ 7.0711e-1 & -3.5355e-1 & 6.1237e-1 & -9.1856e+02 \end{bmatrix}$$

  with $P = [M \mid -M\tilde{\mathbf{C}}]$ and center $\tilde{\mathbf{C}} = [1000.0, 2000.0, 1500.0]^T$

- The matrix M decomposes as

$$M = KR = \begin{bmatrix} 468.2 & 91.2 & 300.0 \\ & 427.2 & 200.0 \\ & & 1.0 \end{bmatrix} \begin{bmatrix} 0.41380 & 0.90915 & 0.04708 \\ -0.57338 & 0.22011 & 0.78917 \\ 0.70711 & -0.35355 & 0.61237 \end{bmatrix}$$

Where is the Decomposition Required?

- If the camera P is constructed from $P = KR[I \mid -\tilde{\mathbf{C}}]$, then the parameters are known and a decomposition is clearly unnecessary, therefore where would one obtain a camera for which the decomposition is not known?

Where is the Decomposition Required?

- If the camera P is constructed from $P = KR[I \mid -\tilde{\mathbf{C}}]$, then the parameters are known and a decomposition is clearly unnecessary, therefore where would one obtain a camera for which the decomposition is not known?
- In fact, cameras can be computed in many different ways and decomposing an unknown camera is a frequently used tool in practice

Where is the Decomposition Required?

- If the camera P is constructed from $P = KR[I \mid -\tilde{\mathbf{C}}]$, then the parameters are known and a decomposition is clearly unnecessary, therefore where would one obtain a camera for which the decomposition is not known?
- In fact, cameras can be computed in many different ways and decomposing an unknown camera is a frequently used tool in practice
- For example, cameras can be computed directly by **calibration** where the camera is computed from a set of world to image correspondences, and indirectly by computing a multiple view relation and subsequently computing projection matrices from this relation

Cameras at Infinity

- Cameras at infinity can be broadly classified into two types: **affine cameras** and **non-affine cameras**

Cameras at Infinity

- Cameras at infinity can be broadly classified into two types: **affine cameras** and **non-affine cameras**
- This means that the left hand 3×3 block of P is singular and the camera center may be found from $P\mathbf{C} = 0$ just as with finite cameras

Cameras at Infinity

- Cameras at infinity can be broadly classified into two types: **affine cameras** and **non-affine cameras**
- This means that the left hand 3×3 block of P is singular and the camera center may be found from $P\mathbf{C} = 0$ just as with finite cameras
- The affine class of cameras are the most important in practice

Affine Cameras

- An affine camera is one that has a camera matrix P in which the last row $\mathbf{P}^{3T}$ is of the form $[0, 0, 0, 1]$ and is called an affine camera because points at infinity are mapped to points at infinity

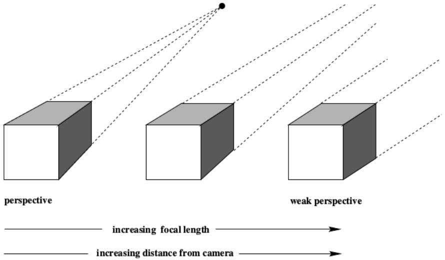
Affine Cameras

- An affine camera is one that has a camera matrix P in which the last row $\mathbf{P}^{3T}$ is of the form $[0, 0, 0, 1]$ and is called an affine camera because points at infinity are mapped to points at infinity
- To understand the affine camera, consider what happens as we apply a cinematographic technique of tracking back while zooming in, in such as way as to keep the objects of interest the same size

Affine Cameras

- An affine camera is one that has a camera matrix P in which the last row $\mathbf{P}^{3T}$ is of the form $[0, 0, 0, 1]$ and is called an affine camera because points at infinity are mapped to points at infinity
- To understand the affine camera, consider what happens as we apply a cinematographic technique of tracking back while zooming in, in such as way as to keep the objects of interest the same size
- We are going to model this process by taking the limit as both the focal length and principal axis distance of the camera from the object increase

Affine Cameras



- As the focal length increases and the distance between the camera and object also increases, the image remains the same size but perspective effects diminish

Affine Cameras

- In analyzing this technique, we start with a finite projective camera (5) where the camera matrix may be written as

$$P_0 = KR[I \mid -\tilde{\mathbf{C}}] = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} & -\mathbf{r}^{3T}\tilde{\mathbf{C}} \end{bmatrix} \tag{6}$$

where $\mathbf{r}^{iT}$ is the i-th row of the rotation matrix

Affine Cameras

- In analyzing this technique, we start with a finite projective camera (5) where the camera matrix may be written as

$$P_0 = KR[I \mid -\tilde{\mathbf{C}}] = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} & -\mathbf{r}^{3T}\tilde{\mathbf{C}} \end{bmatrix} \tag{6}$$

  where $\mathbf{r}^{iT}$ is the i-th row of the rotation matrix

- This camera is located at $\tilde{\mathbf{C}}$ and has orientation denoted by R and internal parameters K of the form given in (4)

Affine Cameras

- In analyzing this technique, we start with a finite projective camera (5) where the camera matrix may be written as

$$P_0 = KR[I \mid -\tilde{\mathbf{C}}] = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} & -\mathbf{r}^{3T}\tilde{\mathbf{C}} \end{bmatrix} \tag{6}$$

  where $\mathbf{r}^{iT}$ is the i-th row of the rotation matrix

- This camera is located at $\tilde{\mathbf{C}}$ and has orientation denoted by R and internal parameters K of the form given in (4)

- The principal ray of the camera is the direction of $\mathbf{r}^3$, and $d_0 = -\mathbf{r}^{3T}\tilde{\mathbf{C}}$ is the distance of the world origin from the camera center in the direction of the principal ray

Affine Cameras

- Now we consider what happens if the camera center is moved backwards along the principal ray at unit speed for a time t so that the camera is moved to $-\tilde{\mathbf{C}} - t\mathbf{r}^3$

Affine Cameras

- Now we consider what happens if the camera center is moved backwards along the principal ray at unit speed for a time t so that the camera is moved to $-\tilde{\mathbf{C}} - t\mathbf{r}^3$
- Replacing $\tilde{\mathbf{C}}$ by $\tilde{\mathbf{C}} - t\mathbf{r}^3$ gives the camera matrix at time t

$$P_t = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}(\tilde{\mathbf{C}} - t\mathbf{r}^3) \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}(\tilde{\mathbf{C}} - t\mathbf{r}^3) \\ \mathbf{r}^{3T} & -\mathbf{r}^{3T}(\tilde{\mathbf{C}} - t\mathbf{r}^3) \end{bmatrix} = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} & d_t \end{bmatrix}$$

where the terms $\mathbf{r}^{iT}\mathbf{r}^3$ are zero for $i = 1, 2$ because R is a rotation matrix

Affine Cameras

- Now we consider what happens if the camera center is moved backwards along the principal ray at unit speed for a time t so that the camera is moved to $-\tilde{\mathbf{C}} - t\mathbf{r}^3$
- Replacing $\tilde{\mathbf{C}}$ by $\tilde{\mathbf{C}} - t\mathbf{r}^3$ gives the camera matrix at time t

$$P_t = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}(\tilde{\mathbf{C}} - t\mathbf{r}^3) \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}(\tilde{\mathbf{C}} - t\mathbf{r}^3) \\ \mathbf{r}^{3T} & -\mathbf{r}^{3T}(\tilde{\mathbf{C}} - t\mathbf{r}^3) \end{bmatrix} = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} & d_t \end{bmatrix}$$

  where the terms $\mathbf{r}^{iT}\mathbf{r}^3$ are zero for $i = 1, 2$ because R is a rotation matrix
- The scalar $d_t = -\mathbf{r}^{3T}\tilde{\mathbf{C}} + t$ is the depth of the world origin w.r.t the camera center in the direction of the principal ray $\mathbf{r}^3$

Affine Cameras

- Next, we consider zooming such that the focal length is increased by a factor k (this magnifies the image by k)

Affine Cameras

- Next, we consider zooming such that the focal length is increased by a factor k (this magnifies the image by k)
- The effect of zooming by k is to multiply the calibration matrix K on the right by $\mathrm{diag}(k, k, 1)$

Affine Cameras

- Next, we consider zooming such that the focal length is increased by a factor k (this magnifies the image by k)
- The effect of zooming by k is to multiply the calibration matrix K on the right by $\text{diag}(k, k, 1)$
- Suppose that $k = d_t/d_0$ so that the image size remains fixed, the resulting camera matrix at time t is

$$P_t = K \begin{bmatrix} \frac{d_t}{d_0} & & \\ & \frac{d_t}{d_0} & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} & d_t \end{bmatrix} = \frac{d_t}{d_0} K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{r}^{3T} d_0/d_t & d_0 \end{bmatrix}$$

and one can ignore the factor d_t/d_0

Affine Cameras

- When $t = 0$ the camera matrix P_t corresponds with (6), and as the limit d_t tends to infinity this matrix becomes

$$P_\infty = \lim_{t \to \infty} P_t = K \begin{bmatrix} \mathbf{r}^{1T} & -\mathbf{r}^{1T}\tilde{\mathbf{C}} \\ \mathbf{r}^{2T} & -\mathbf{r}^{2T}\tilde{\mathbf{C}} \\ \mathbf{0}^T & d_0 \end{bmatrix} \tag{7}$$

  which is just the original camera matrix (6) with the first three entries of the last row set to zero

Decomposition of $P\infty$

- The camera matrix (7) may be written as

$$P_\infty = \begin{bmatrix} K_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & d_0 \end{bmatrix}$$

where $\hat{R}$ consists of the first two rows of a rotation matrix, $\hat{\mathbf{t}}$ is the vector $[-\mathbf{r}^{1T}\tilde{\mathbf{C}}, -\mathbf{r}^{2T}\tilde{\mathbf{C}}]^T$, and $\hat{\mathbf{0}}$ the vector $[0, 0]^T$

Decomposition of $P\infty$

- The camera matrix (7) may be written as

$$P_\infty = \begin{bmatrix} K_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & d_0 \end{bmatrix}$$

  where $\hat{R}$ consists of the first two rows of a rotation matrix, $\hat{\mathbf{t}}$ is the vector $[-\mathbf{r}^{1T}\tilde{\mathbf{C}}, -\mathbf{r}^{2T}\tilde{\mathbf{C}}]^T$, and $\hat{\mathbf{0}}$ the vector $[0, 0]^T$

- The matrix $K_{2\times 2}$ is upper triangular

Decomposition of $P\infty$

- By verifying that

$$P_\infty = \begin{bmatrix} K_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & d_0 \end{bmatrix} = \begin{bmatrix} d_0^{-1} K_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

  we can replace $K_{2\times 2}$ by $d_0^{-1} K_{2\times 2}$ and assume that $d_0 = 1$

Decomposition of $P\infty$

- By verifying that

$$P_\infty = \begin{bmatrix} K_{2\times2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & d_0 \end{bmatrix} = \begin{bmatrix} d_0^{-1} K_{2\times2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

  we can replace $K_{2\times2}$ by $d_0^{-1} K_{2\times2}$ and assume that $d_0 = 1$
- Multiplying out this product gives

$$P_\infty = \begin{bmatrix} K_{2\times2}\hat{R} & K_{2\times2}\hat{\mathbf{t}} + \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} = \begin{bmatrix} K_{2\times2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} + K_{2\times2}^{-1}\tilde{\mathbf{x}}_0 \\ \mathbf{0}^T & 1 \end{bmatrix}$$

$$= \begin{bmatrix} K_{2\times2} & K_{2\times2}\hat{\mathbf{t}} + \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{0}} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Decomposition of $P\infty$

- Thus, making appropriate substitutions for $\hat{\mathbf{t}}$ or $\tilde{\mathbf{x}}_0$, we can write the affine camera matrix in one of two forms

$$P_\infty = \begin{bmatrix} K_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} K_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{0}} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Decomposition of $P\infty$

- Thus, making appropriate substitutions for $\hat{\mathbf{t}}$ or $\tilde{\mathbf{x}}_0$, we can write the affine camera matrix in one of two forms

$$P_\infty = \begin{bmatrix} K_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} K_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{0}} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

- Since the value of $\tilde{\mathbf{x}}_0$ is dependent on the particular choice of world coordinates (and therefore is not an intrinsic property of the camera) it is preferable to use the first decomposition

$$P_\infty = \begin{bmatrix} K_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{\mathbf{t}} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{8}$$

Parallel Projection

- The essential differences between P_∞ and a finite camera are:

Parallel Projection

- The essential differences between P_∞ and a finite camera are:
  - The parallel projection matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

    replaces the canonical projection matrix $[I \mid \mathbf{0}]$ of a finite camera

Parallel Projection

- The essential differences between P_∞ and a finite camera are:
  - The parallel projection matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

    replaces the canonical projection matrix $[I \mid \mathbf{0}]$ of a finite camera
  - The calibration matrix $\begin{bmatrix} K_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix}$ replaces K of a finite camera

Parallel Projection

- The essential differences between P_∞ and a finite camera are:
  - The parallel projection matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

    replaces the canonical projection matrix $[I \mid \mathbf{0}]$ of a finite camera
  - The calibration matrix $\begin{bmatrix} K_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^T & 1 \end{bmatrix}$ replaces K of a finite camera
  - The principal point is not defined

Properties of a Projective Camera

Camera centre. The camera centre is the 1-dimensional right null-space $\mathbf{C}$ of $\mathtt{P}$, i.e. $\mathtt{P}\mathbf{C} = \mathbf{0}$.

- **Finite camera** ($\mathtt{M}$ is not singular) $\mathbf{C} = \begin{pmatrix} -\mathtt{M}^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix}$
- **Camera at infinity** ($\mathtt{M}$ is singular) $\mathbf{C} = \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$ where $\mathbf{d}$ is the null 3-vector of $\mathtt{M}$, i.e. $\mathtt{M}\mathbf{d} = \mathbf{0}$.

Column points. For $i = 1, \ldots, 3$, the column vectors $\mathbf{p}_i$ are vanishing points in the image corresponding to the $\mathtt{X}$, $\mathtt{Y}$ and $\mathtt{Z}$ axes respectively. Column $\mathbf{p}_4$ is the image of the coordinate origin.

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

Axis planes. The planes $\mathbf{P}^1$ and $\mathbf{P}^2$ (the first and second rows of $\mathtt{P}$) represent planes in space through the camera centre, corresponding to points that map to the image lines $x = 0$ and $y = 0$ respectively.

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

Principal ray. The principal ray (axis) of the camera is the ray passing through the camera centre $\mathbf{C}$ with direction vector $\mathbf{m}^{3\mathsf{T}}$. The principal axis vector $\mathbf{v} = \det(\mathtt{M})\mathbf{m}^3$ is directed towards the front of the camera.

- A summary of the properties of a projective camera P

Properties of a Projective Camera

Camera centre. The camera centre is the 1-dimensional right null-space $\mathbf{C}$ of $\mathtt{P}$, i.e. $\mathtt{P}\mathbf{C} = \mathbf{0}$.

- **Finite camera** ($\mathtt{M}$ is not singular) $\mathbf{C} = \begin{pmatrix} -\mathtt{M}^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix}$
- **Camera at infinity** ($\mathtt{M}$ is singular) $\mathbf{C} = \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$ where $\mathbf{d}$ is the null 3-vector of $\mathtt{M}$, i.e. $\mathtt{M}\mathbf{d} = \mathbf{0}$.

Column points. For $i = 1, \ldots, 3$, the column vectors $\mathbf{p}_i$ are vanishing points in the image corresponding to the $\mathtt{X}$, $\mathtt{Y}$ and $\mathtt{Z}$ axes respectively. Column $\mathbf{p}_4$ is the image of the coordinate origin.

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

Axis planes. The planes $\mathbf{P}^1$ and $\mathbf{P}^2$ (the first and second rows of $\mathtt{P}$) represent planes in space through the camera centre, corresponding to points that map to the image lines $x = 0$ and $y = 0$ respectively.

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

Principal ray. The principal ray (axis) of the camera is the ray passing through the camera centre $\mathbf{C}$ with direction vector $\mathbf{m}^{3\mathsf{T}}$. The principal axis vector $\mathbf{v} = \det(\mathtt{M})\mathbf{m}^3$ is directed towards the front of the camera.

- A summary of the properties of a projective camera P
- The matrix is represented by the block form $P = [M \,|\, \mathbf{p}_4]$

Estimating the Camera Projection Matrix

- We will study numerical methods for estimating the camera projection matrix from corresponding 3-space and image entities

Estimating the Camera Projection Matrix

- We will study numerical methods for estimating the camera projection matrix from corresponding 3-space and image entities
- This computation of the camera matrix is known as **resectioning** and the simplest such correspondence is that between a 3D point $\mathbf{X}$ and its image $\mathbf{x}$ under the unknown camera mapping

Estimating the Camera Projection Matrix

- We will study numerical methods for estimating the camera projection matrix from corresponding 3-space and image entities
- This computation of the camera matrix is known as **resectioning** and the simplest such correspondence is that between a 3D point $\mathbf{X}$ and its image $\mathbf{x}$ under the unknown camera mapping
- Given sufficiently many correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ P may determined, and similarly P may be determined from sufficiently many corresponding world and image lines

Computation of the Camera Matrix P

- Assume that we are given a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$

Computation of the Camera Matrix P

- Assume that we are given a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$
- Our goal is to find a 3×4 camera matrix P such that $\mathbf{x}_i = P\mathbf{X}_i$ for all i

Computation of the Camera Matrix P

- For each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^T & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & \mathbf{0}^T & -x_i\mathbf{X}_i^T \\ -y_i\mathbf{X}_i^T & x_i\mathbf{X}_i^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix} = \mathbf{0} \qquad (9)$$

where each $\mathbf{P}^{iT}$ is a 4-vector, the ith row of P

Computation of the Camera Matrix P

- Alternatively, one may choose to use only the first two equations

$$\begin{bmatrix} \mathbf{0}^T & -w_i \mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & \mathbf{0}^T & -x_i \mathbf{X}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix} = \mathbf{0} \qquad (10)$$

since the three equations of (9) are linearly dependent

Computation of the Camera Matrix P

- Alternatively, one may choose to use only the first two equations

$$\begin{bmatrix} \mathbf{0}^T & -w_i \mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & \mathbf{0}^T & -x_i \mathbf{X}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix} = \mathbf{0} \qquad (10)$$

  since the three equations of (9) are linearly dependent

- From n point correspondences we obtain a $2n \times 12$ matrix A by stacking up the equations (10) for each correspondence

Computation of the Camera Matrix P

- Alternatively, one may choose to use only the first two equations

$$\begin{bmatrix} \mathbf{0}^T & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & \mathbf{0}^T & -x_i\mathbf{X}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix} = \mathbf{0} \qquad (10)$$

  since the three equations of (9) are linearly dependent

- From n point correspondences we obtain a $2n \times 12$ matrix A by stacking up the equations (10) for each correspondence

- P is computed by solving the set of equations $A\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the vector containing the entries of P

Minimal Solution

- Since P has 12 entries, and (ignoring scale) 11 degrees of freedom, it is necessary to have 11 equations to solve for P

Minimal Solution

- Since P has 12 entries, and (ignoring scale) 11 degrees of freedom, it is necessary to have 11 equations to solve for P
- Each point correspondence leads to two equations, therefore at a minimum $5\frac{1}{2}$ such correspondences are required to solve for P

Minimal Solution

- Since P has 12 entries, and (ignoring scale) 11 degrees of freedom, it is necessary to have 11 equations to solve for P
- Each point correspondence leads to two equations, therefore at a minimum $5\frac{1}{2}$ such correspondences are required to solve for P
- The $\frac{1}{2}$ indicates that only one of the equations is used from the sixth point, so one needs to know the x-coordinate (or alternatively the y-coordinate) of the sixth image point

Minimal Solution

- Given this minimum number of correspondences, the solution is exact, i.e. the space points are projected exactly onto their measured images

Minimal Solution

- Given this minimum number of correspondences, the solution is exact, i.e. the space points are projected exactly onto their measured images
- The solution is obtained by solving $A\mathbf{p} = \mathbf{0}$ where A is 11×12 in this case

Minimal Solution

- Given this minimum number of correspondences, the solution is exact, i.e. the space points are projected exactly onto their measured images
- The solution is obtained by solving $A\mathbf{p} = \mathbf{0}$ where A is 11×12 in this case
- In general, A will have rank 11 and the solution vector $\mathbf{p}$ is the 1-dimensional right null space of A

Overdetermined Solution

- If the data is not exact because of noise in the point coordinates and $n \geq 6$ point correspondences are given, then there will not be an exact solution to $A\mathbf{p} = \mathbf{0}$

Overdetermined Solution

- If the data is not exact because of noise in the point coordinates and $n \geq 6$ point correspondences are given, then there will not be an exact solution to $A\mathbf{p} = \mathbf{0}$
- One approach is to minimize $||A\mathbf{p}||$ subject to $||\mathbf{p}|| = 1$ where the residual $A\mathbf{p}$ is known as the **algebraic error**

Overdetermined Solution

- If the data is not exact because of noise in the point coordinates and $n \geq 6$ point correspondences are given, then there will not be an exact solution to $A\mathbf{p} = \mathbf{0}$
- One approach is to minimize $||A\mathbf{p}||$ subject to $||\mathbf{p}|| = 1$ where the residual $A\mathbf{p}$ is known as the **algebraic error**
- Using this approach, the DLT algorithm for computing P proceeds in the same manner as that for H

Data Normalization

- It is important to carry out some sort of data normalization, i.e. the points should be translated so that their centroid is at the origin and scaled so that their RMS (root mean squared) distance from the origin is $\sqrt{2}$

Data Normalization

- It is important to carry out some sort of data normalization, i.e. the points should be translated so that their centroid is at the origin and scaled so that their RMS (root mean squared) distance from the origin is $\sqrt{2}$
- In the case where the variation in depth of the 3D points from the camera is relatively slight the centroid of the points is translated to the origin, and their coordinates are scaled so that the RMS distance from the origin is $\sqrt{3}$ (thus the "average" point has coordinates of magnitude $[1, 1, 1, 1]^T$)

Line Correspondences

- The DLT algorithm can be extended to take into account line correspondences as well

Line Correspondences

- The DLT algorithm can be extended to take into account line correspondences as well
- A 3D line may be represented by two points $\mathbf{X}_0$ and $\mathbf{X}_1$ through which the line passes and the plane formed by back-projecting from the image line $\mathbf{l}$ is equal to $P^T\mathbf{l}$

Line Correspondences

- The DLT algorithm can be extended to take into account line correspondences as well
- A 3D line may be represented by two points $\mathbf{X}_0$ and $\mathbf{X}_1$ through which the line passes and the plane formed by back-projecting from the image line $\mathbf{l}$ is equal to $P^T\mathbf{l}$
- The condition that the point $\mathbf{X}_j$ lies on this plane is then

$$\mathbf{l}^T P\mathbf{X}_j = 0 \text{ for } j = 0, 1$$

  where each choice of j gives a single linear equation in the entries of P so that two equations are obtained for each 3D to 2D line correspondence

Geometric Error

- Suppose that the world points $\mathbf{X}_i$ are known far more accurately than the measured image points, e.g. the points $\mathbf{X}_i$ might arise from an accurately machined calibration object

Geometric Error

- Suppose that the world points $\mathbf{X}_i$ are known far more accurately than the measured image points, e.g. the points $\mathbf{X}_i$ might arise from an accurately machined calibration object
- Then the **geometric error** in the image is

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$$

where $\mathbf{x}_i$ is the measured point and $\hat{\mathbf{x}}_i$ is the point $P\mathbf{X}_i$, i.e. the point which is the exact image of $\mathbf{X}_i$ under P

Geometric Error

- If the measurement errors are Gaussian then the solution of

$$\min_P \sum_i d(\mathbf{x}_i, P\mathbf{X}_i)^2$$

is the Maximum Likelihood estimate of P

Geometric Error

- If the measurement errors are Gaussian then the solution of

$$\min_{P} \sum_{i} d(\mathbf{x}_i, P\mathbf{X}_i)^2$$

is the Maximum Likelihood estimate of P

- Minimizing the geometric error requires the use of iterative techniques (such as Levenberg-Marquardt) where the DLT solution, or a minimal solution, may be used as a starting point for the iterative minimization

The Gold Standard Algorithm

Objective

Given $n \geq 6$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood estimate of the camera projection matrix P, i.e. the P which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$.

Algorithm

(i) **Linear solution.** Compute an initial estimate of P using a linear method such as algorithm 4.2(*p*109):

(a) **Normalization:** Use a similarity transformation T to normalize the image points, and a second similarity transformation U to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$.

(b) **DLT:** Form the $2n \times 12$ matrix A by stacking the equations (7.2) generated by each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$. Write $\mathbf{p}$ for the vector containing the entries of the matrix $\tilde{\mathtt{P}}$. A solution of $\mathtt{A}\mathbf{p} = \mathbf{0}$, subject to $\|\mathbf{p}\| = 1$, is obtained from the unit singular vector of A corresponding to the smallest singular value.

(ii) **Minimize geometric error.** Using the linear estimate as a starting point minimize the geometric error (7.4):

$$\sum_i d(\tilde{\mathbf{x}}_i, \tilde{\mathtt{P}}\tilde{\mathbf{X}}_i)^2$$

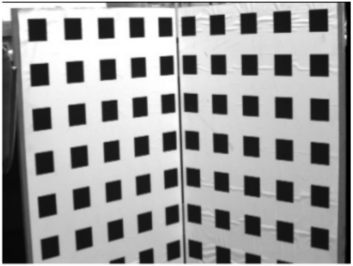
over $\tilde{\mathtt{P}}$, using an iterative algorithm such as Levenberg–Marquardt.

(iii) **Denormalization.** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}$ as

$$\mathtt{P} = \mathtt{T}^{-1}\tilde{\mathtt{P}}\mathtt{U}.$$

- The Gold Standard algorithm for estimating P from world to image point correspondences in the case that the world points are very accurately known

Example: Camera Estimation from a Calibration Object



- The black and white checkerboard pattern is designed to enable the positions of the corners of the imaged squares to be obtained to high accuracy

Example: Camera Estimation from a Calibration Object

- The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:

Example: Camera Estimation from a Calibration Object

- The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:
    - Canny edge detection

Example: Camera Estimation from a Calibration Object

- The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:
  - Canny edge detection
  - Straight line fitting to the detected linked edges

Example: Camera Estimation from a Calibration Object

- The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:
  - Canny edge detection
  - Straight line fitting to the detected linked edges
  - Intersecting the lines to obtain the imaged corners

Example: Camera Estimation from a Calibration Object

- The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:
  - Canny edge detection
  - Straight line fitting to the detected linked edges
  - Intersecting the lines to obtain the imaged corners
- If sufficient care is taken, then $\mathbf{x}_i$ can be obtained to a localization accuracy of far better than 1/10 of a pixel

Example: Camera Estimation from a Calibration Object

- The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:
  - Canny edge detection
  - Straight line fitting to the detected linked edges
  - Intersecting the lines to obtain the imaged corners
- If sufficient care is taken, then $\mathbf{x}_i$ can be obtained to a localization accuracy of far better than 1/10 of a pixel
- A rule of thumb is that for good estimation the number of constraints (point measurements) should exceed the number of unknowns (the 11 camera parameters) by a factor of five (i.e. at least 28 points should be used)

Errors in the World Points

- It may be the case that the world points are measured with "infinite" accuracy, thus one may choose to estimate P by minimizing a 3D geometric error, or an image error, or both

Errors in the World Points

- It may be the case that the world points are measured with "infinite" accuracy, thus one may choose to estimate P by minimizing a 3D geometric error, or an image error, or both
- If only errors in the world plane are considered, then the 3D geometric error is defined as

$$\sum_i d(\mathbf{X}_i, \hat{\mathbf{X}}_i)^2$$

where $\hat{\mathbf{X}}_i$ is the closest point in space to $\mathbf{X}_i$ that maps exactly onto $\mathbf{x}_i$ via $\mathbf{x}_i = P\hat{\mathbf{X}}_i$

Errors in the World Points

- More generally, if errors in both the world and image points are considered, then a weighted sum of world and image errors is minimized

$$\sum_{i=1}^{n} d_{\mathrm{Mah}}(\mathbf{x}_i, P\hat{\mathbf{X}}_i)^2 + d_{\mathrm{Mah}}(\mathbf{X}_i, \hat{\mathbf{X}}_i)^2$$

  where d_{Mah} represents the Mahalanobis distance w.r.t the known error covariance matrices for each of the measurements $\mathbf{x}_i$ and $\mathbf{X}_i$