Camera Models and Image Formation

Srikumar Ramalingam

School of Computing

University of Utah

srikumar@cs.utah.edu

Reference

Most slides are adapted from the following notes:

- Some lecture notes on geometric computer vision (available online) by Peter Sturm

3D Street Art

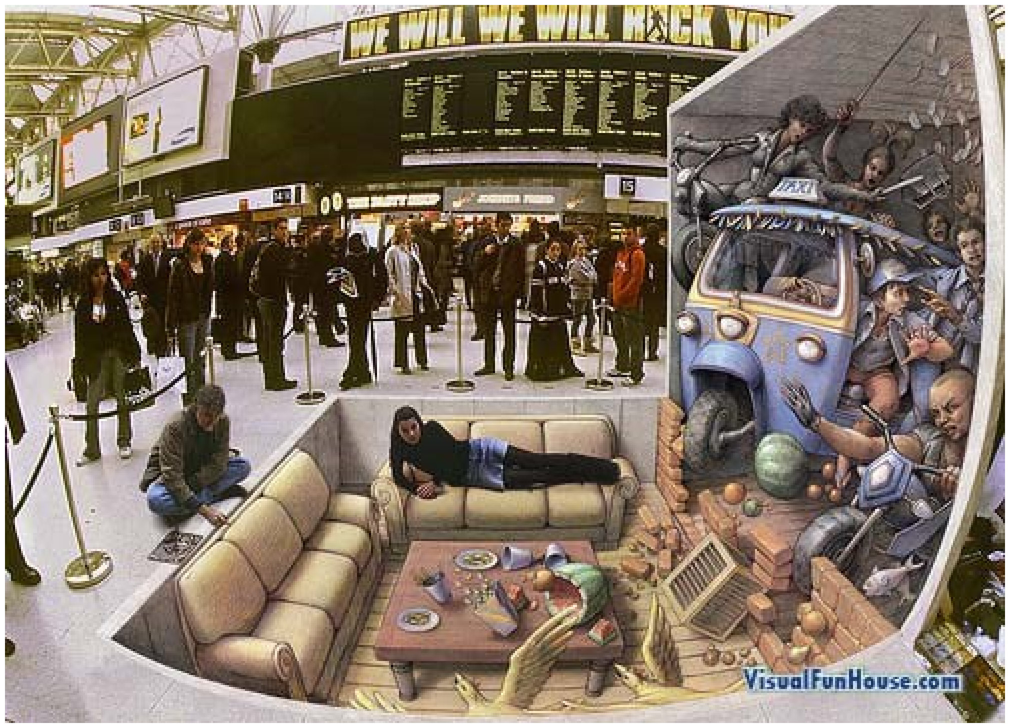


Image courtesy: Julian Beaver (VisualFunHouse.com)

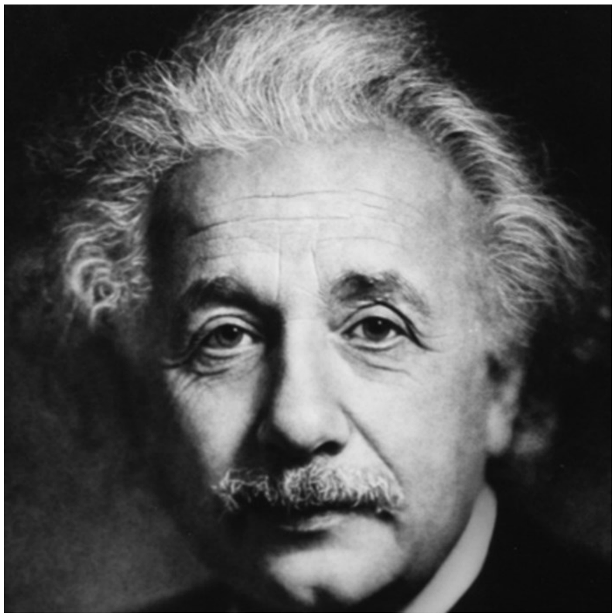
3D Street Art

Image courtesy: Julian Beaver (VisualFunHouse.com)

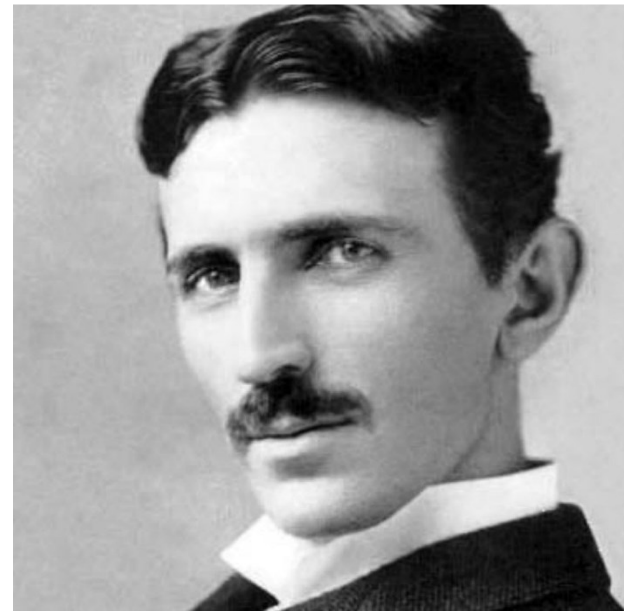
3D Street Art

Image courtesy: Julian Beaver (VisualFunHouse.com)

Pixels are permuted and replicated



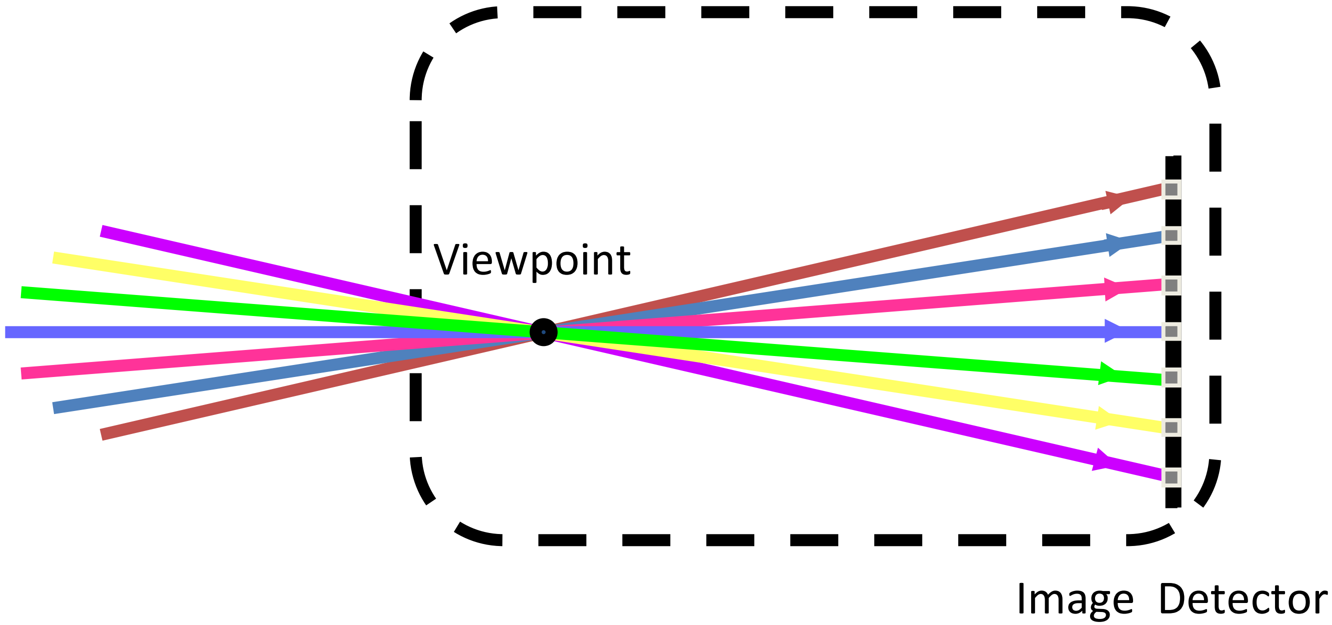
Scene



Output Image

- Is this a camera?

Perspective Imaging



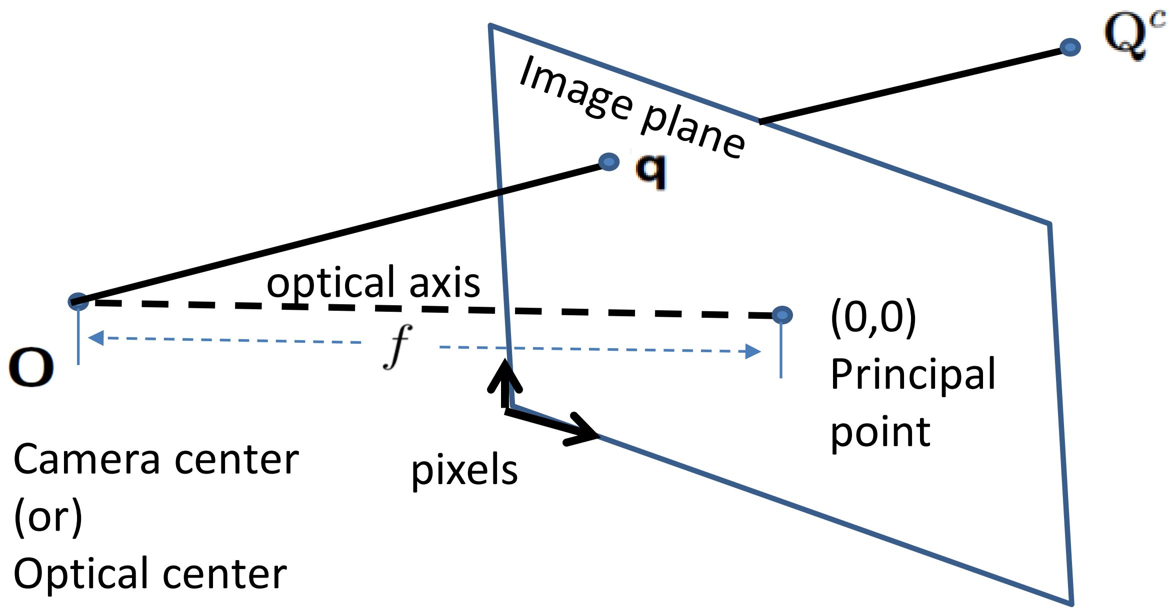
Many Types of Imaging Systems



Pinhole model

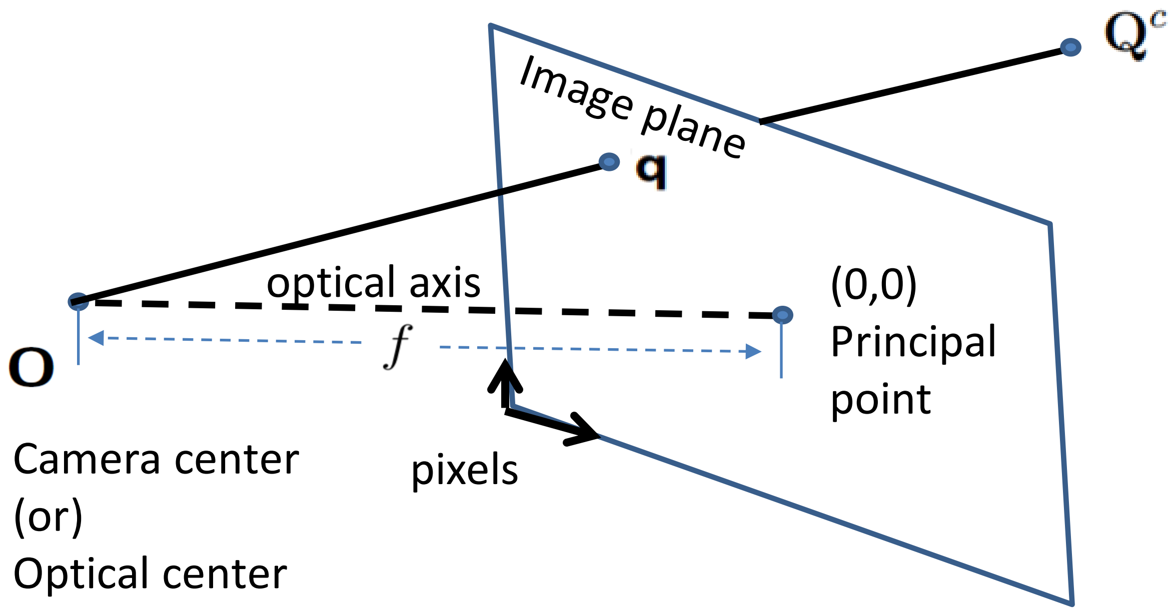
- A camera maps the 3D world to a 2D image.
- Many such mappings or camera models exist.
- A pinhole model is a good approximation for many existing cameras.

Perspective projection



- A pinhole model can be expressed using an optical center $\mathbf{O}$ and an image plane. We treat the optical center to be the origin of the camera coordinate frame.
- A 3D point $\mathbf{Q}^c$ gets projected along the line of sight that connects it with the optical center $\mathbf{O}$. Its image point $\mathbf{q}$ is the intersection of this line with the image plane.

Perspective projection



- We are given the 3D point in the camera coordinate system (exponent "c" denotes the camera coordinate system).

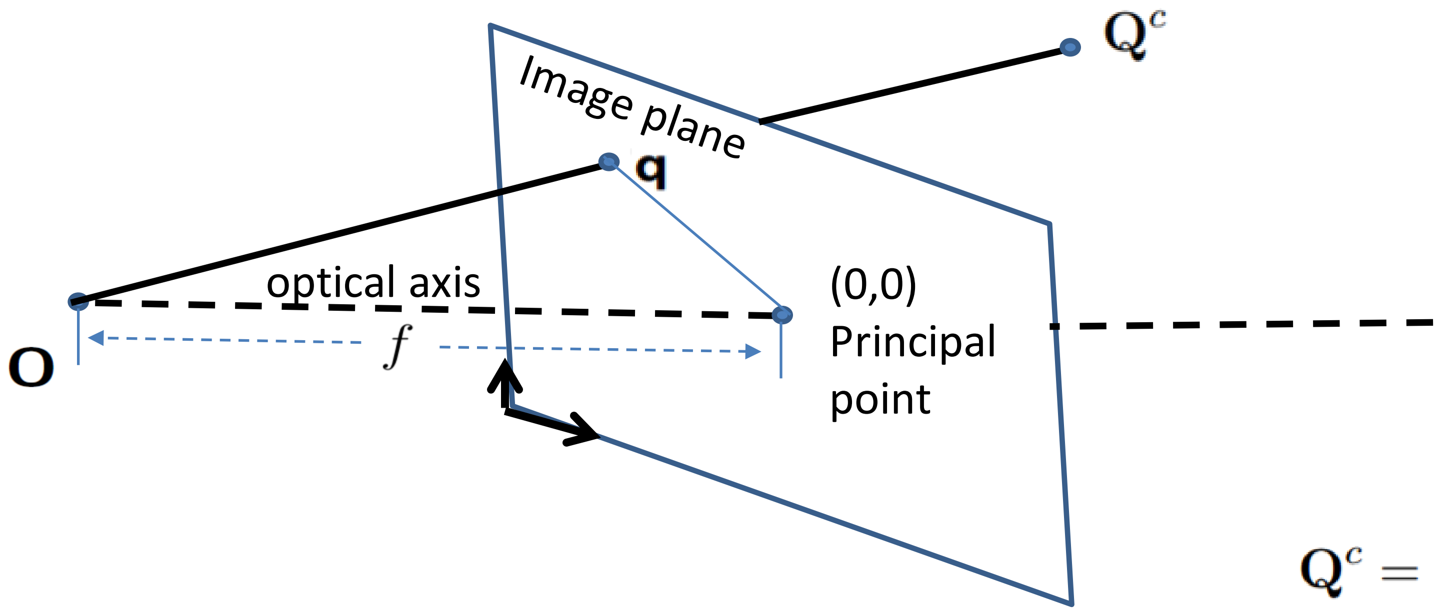
$$\mathbf{Q}^c = \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}$$

Why do we have the "1"?

Why homogenous coordinates?

- Allows us to express common transformations in matrix form:
- Simplifies the concepts such as points at infinity, etc.

Perspective projection

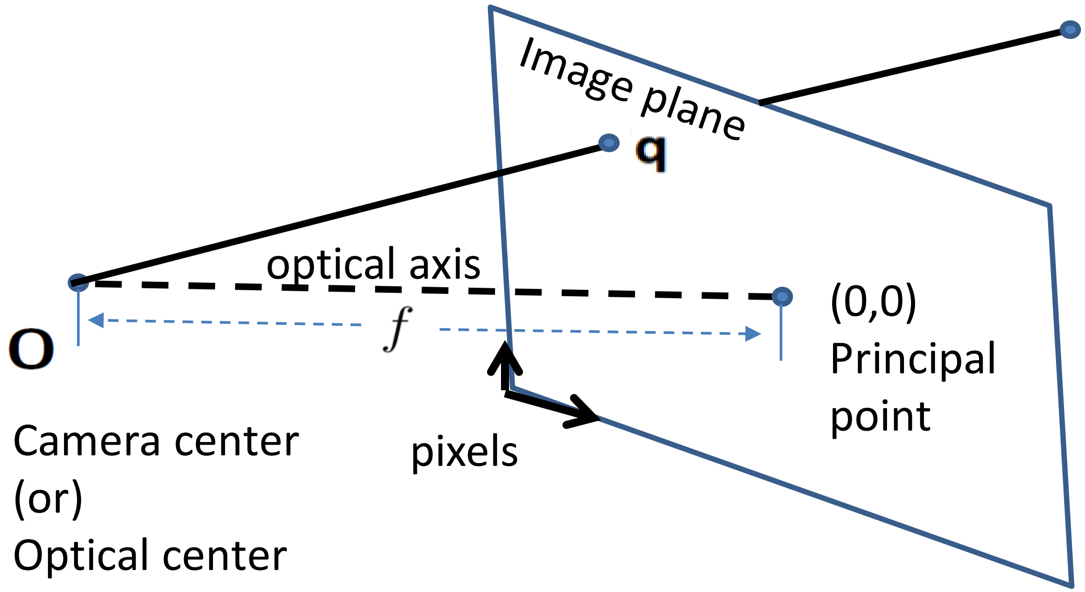


$$\mathbf{Q}^c = \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}$$

- The image point $\mathbf{q}(x, y)$ can be computed using similar triangles:

$$x = f\frac{X^c}{Z^c} \qquad y = f\frac{Y^c}{Z^c}$$

Perspective projection



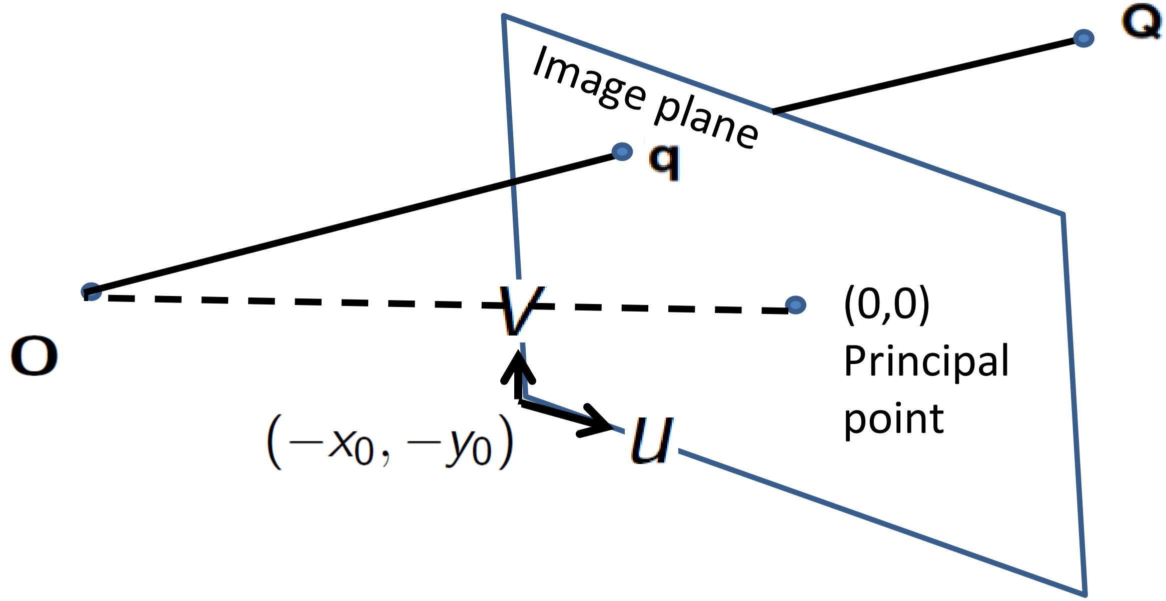
$$\mathbf{Q}^c = \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}$$

- In homogeneous coordinates (by adding 1 at the end of a vector), these equations can be written in the form of matrix-vector product:

"up to a scale" - scalar multiplication does not change the equivalence.

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}$$

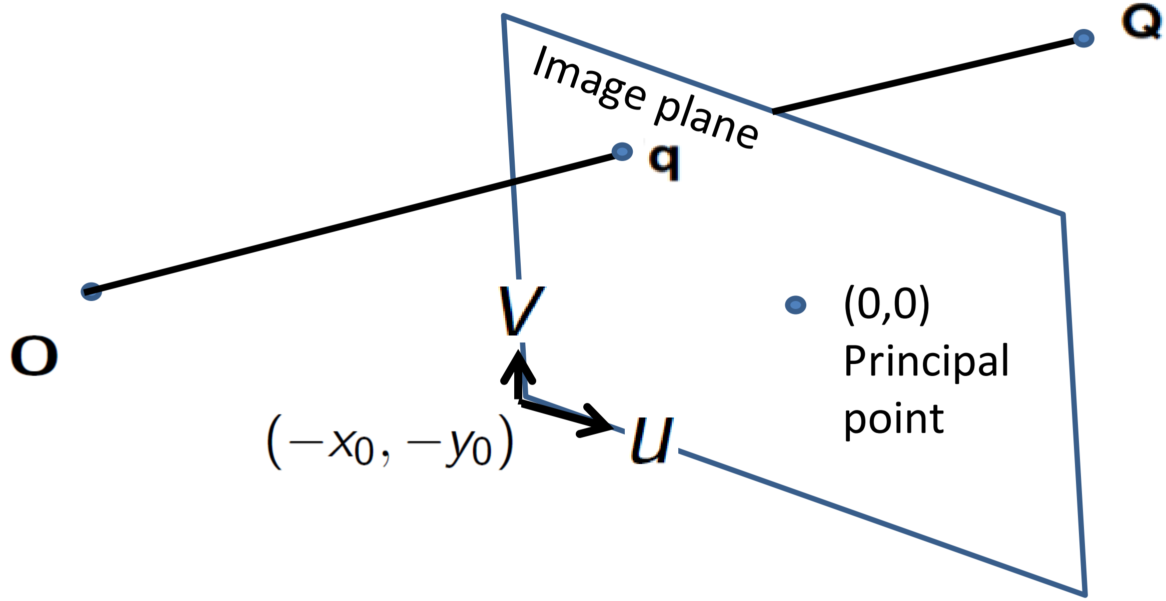
From image plane to pixel coordinates



Pixels need not be squares (especially the older cameras):

- k_u - density of pixels along u direction.
- k_v - density of pixels along v direction.

From image plane to pixel coordinates



- In homogeneous coordinates we have the following for the image point $\mathbf{q}(x, y)$:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

3D-to-2D projection

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}$$

3D to image plane projection

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Image plane to pixel system

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}$$

World coordinate frame

- We assume that the 3D point is given in the world coordinate system.
- We model the pose of the camera using a 3x1 translation vector $\mathbf{t}$ and a 3x3 rotation matrix R.
- Let us assume that the superscript “m” denotes 3D points in the world coordinate frame, and the transformation to camera frame is given below:

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \end{pmatrix} = \mathsf{R}\left(\begin{pmatrix} X^m \\ Y^m \\ Z^m \end{pmatrix} - \mathbf{t}\right) = \mathsf{R}\begin{pmatrix} X^m \\ Y^m \\ Z^m \end{pmatrix} - \mathsf{R}\mathbf{t}$$

World coordinate frame

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \end{pmatrix} = \mathsf{R}\left(\begin{pmatrix} X^m \\ Y^m \\ Z^m \end{pmatrix} - \mathbf{t}\right) = \mathsf{R}\begin{pmatrix} X^m \\ Y^m \\ Z^m \end{pmatrix} - \mathsf{R}\mathbf{t}$$

- Rewriting the above equation in homogeneous coordinates to denote the mapping from world to camera coordinate frame:

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}}_{\text{4x4 matrix}} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

$\mathbf{0}^\mathsf{T}$ = (0 0 0)

Cross-checking

- We want to ensure that the optical center is the origin of the camera coordinate system.
- In the world coordinate system, the optical center is given by $\mathbf{t}$.

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R}\mathbf{t} - \mathsf{R}\mathbf{t} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$$

Rotation matrices

- Rotations are orthonormal matrices:
  - their columns are mutually orthogonal 3-vectors of norm 1.
- For a rotation matrix, the determinant value should be equal to +1. For reflection matrix, the determinant value will be -1.
- The inverse of a rotation matrix is its transpose:

$$RR^T = I$$

Rotation matrices

- Each rotation can be decomposed into three base rotations:

$$\mathsf{R} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

- The Euler angles α, β and γ are associated with X, Y and Z axes respectively.

Complete Model

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

- The following 3x3 matrix is the camera calibration matrix:

$$\mathsf{K} = \begin{pmatrix} k_u f & 0 & k_u x_0 \\ 0 & k_v f & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Projection Matrix

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{K} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix}$$

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{KR} & -\mathsf{KR}\mathbf{t} \end{pmatrix}$$

$$\mathsf{P} \sim \mathsf{KR} \begin{pmatrix} \mathtt{I} & -\mathbf{t} \end{pmatrix}$$

- The 3×4 matrix is called projection matrix. It maps 3D points to 2D image points, all expressed in homogeneous coordinates.

Camera Parameters

- Extrinsic parameters – the rotation matrix R and the translation vector $\mathbf{t}$.
- Intrinsic parameters – that explains what happens inside a camera - f, k_u, k_v, x_0 and y_0 .

Calibration matrix

$$\mathsf{K} = \begin{pmatrix} k_u f & 0 & k_u x_0 \\ 0 & k_v f & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

There used to be a skew parameter in old cameras, that are not necessary in modern cameras.

- We have 4 parameters defined by 5 coefficients.

Reparameterization

$$\left.\begin{aligned}\alpha_u &= k_u f \\ \alpha_v &= k_v f\end{aligned}\right\}\ \text{Focal lengths in pixels}$$

$$\left.\begin{aligned}u_0 &= k_u x_0 \\ v_0 &= k_v y_0\end{aligned}\right\}\ \text{Principal point in pixels}$$

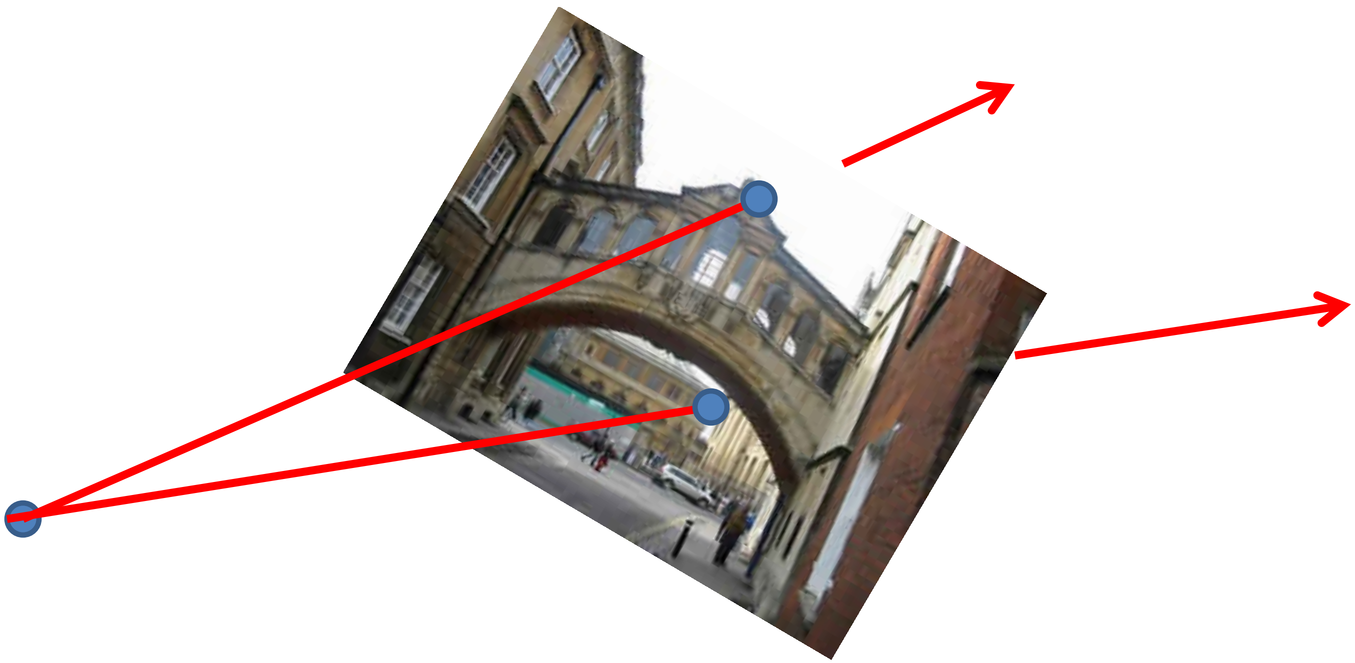
Calibration Parameters

- Focal length typically ranges to several hundreds of millimeters.
- The photosensitive area of a camera is typically a rectangle with several millimeters side length and the pixel density is usually of the order of one or several hundreds of pixels per mm
- For cameras with well mounted optics, the principal point is usually very close to the center of the photosensitive area.

What is Camera Calibration?

- The task refers to the problem of computing the calibration matrix.
- We compute the focal length, principal point, and aspect ratio.

Why do you need calibration?



- For a given pixel, the camera calibration allows you to know the light ray along which the camera samples the world.

Calibration Toolbox

MATLAB

- https://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html

OpenCV

- http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

Visual SFM

- http://ccwu.me/vsfm/
- https://www.youtube.com/watch?v=SHa_LBIzDac

Forward and backward projection

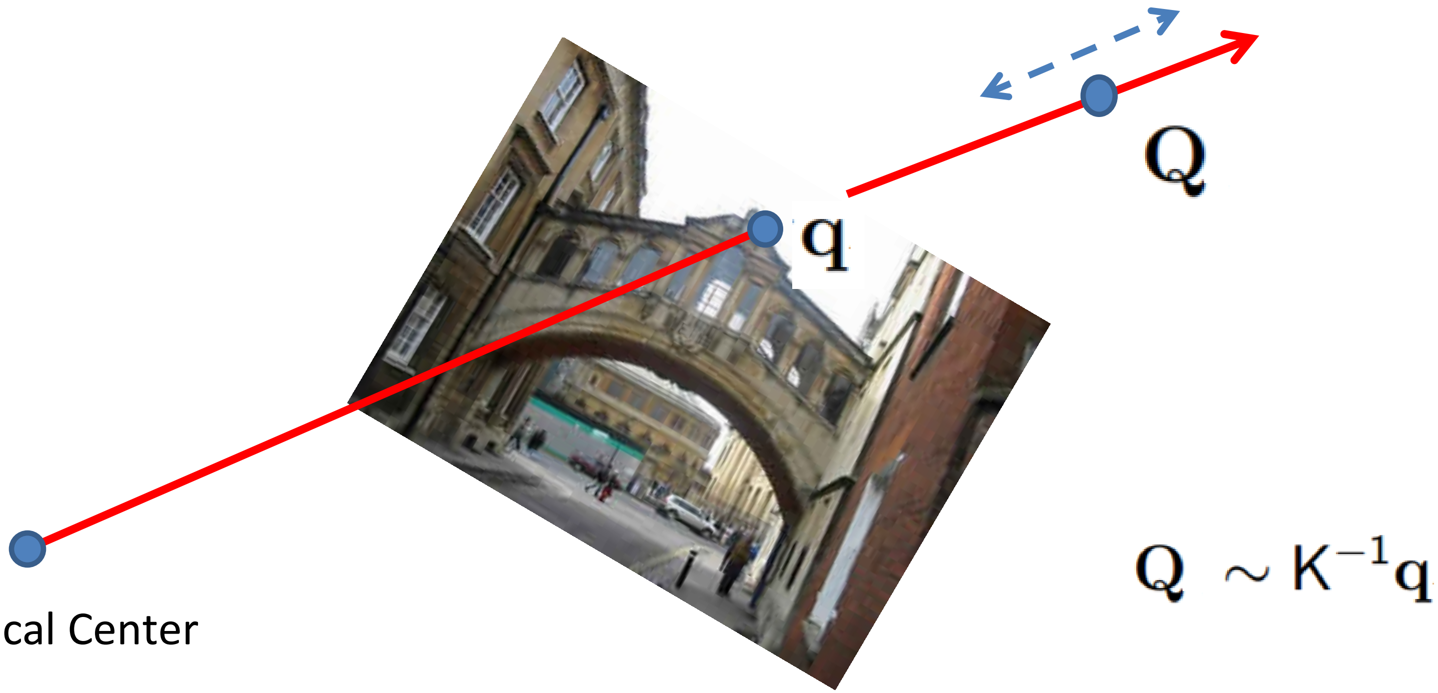
- Forward: The projection of a 3D point on the image:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{KR} \begin{pmatrix} \mathtt{I} & -\mathbf{t} \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{P} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix}$$

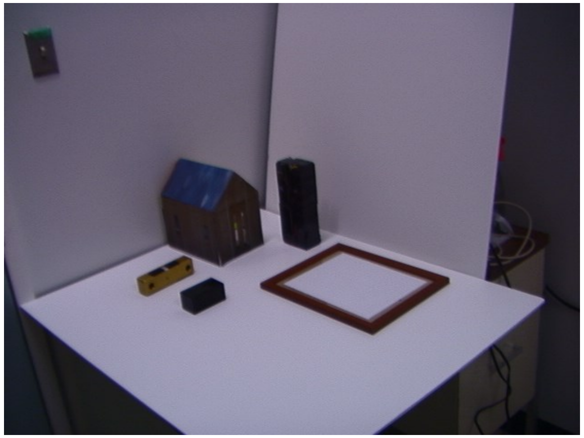
Forward and backward projection

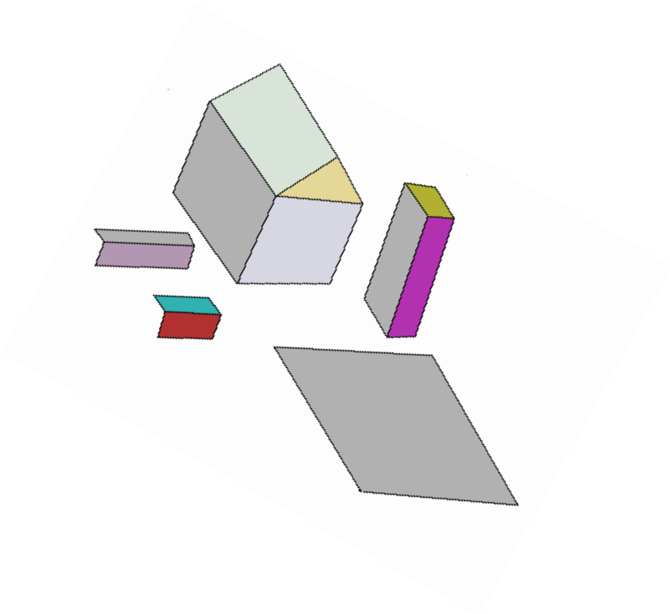
- Backward projection: Given a pixel in the image, we determine the set of points in space that map to this point.



Pose estimation

- Given an image of an object whose structure is perfectly known, is it possible find the position and orientation of the camera?

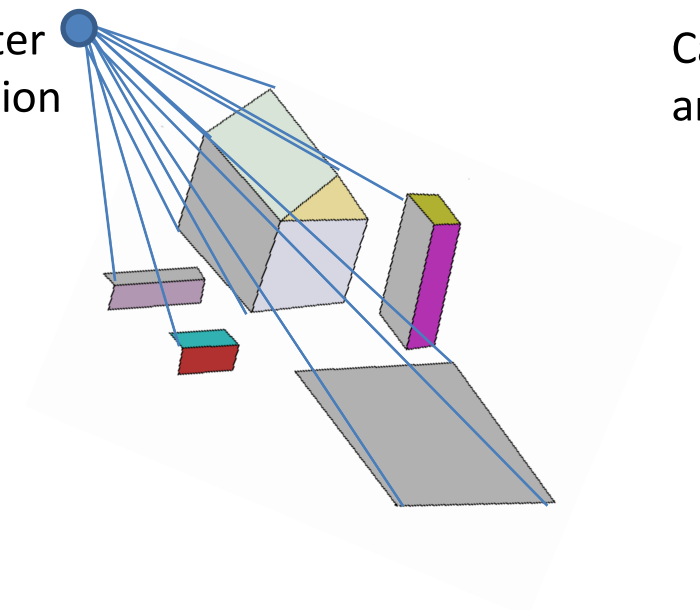




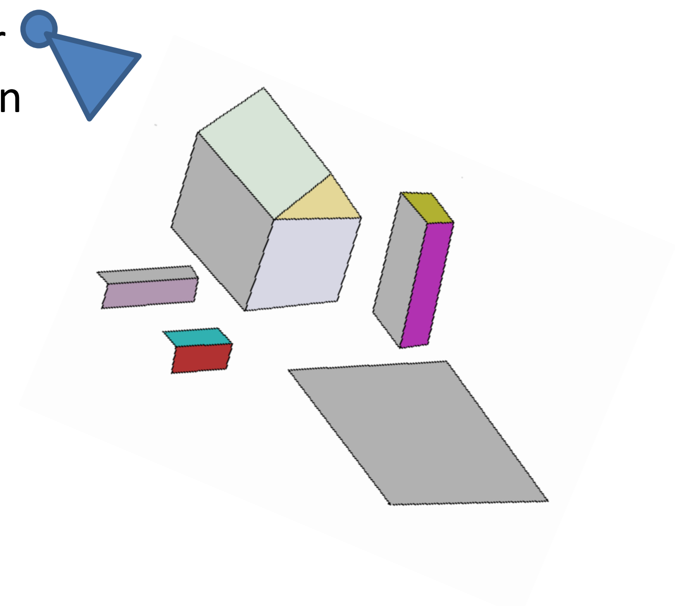
Pose estimation

Compute the pose using constraints from the 3D points

Camera center and orientation

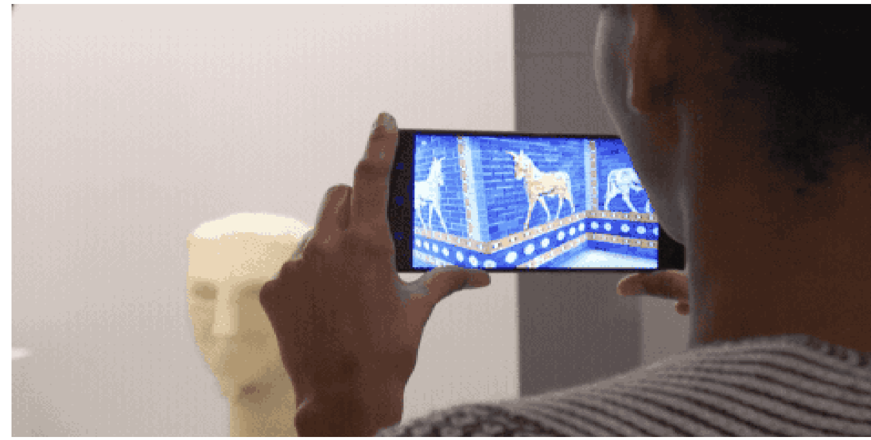


Camera center and orientation



Where do you find applications for pose estimation?

Image courtesy: Oculus Rift



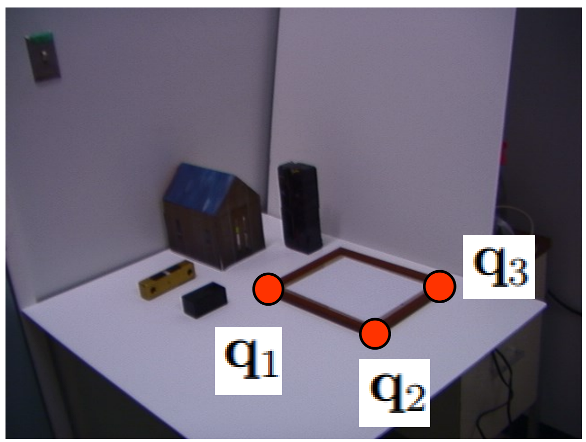
Google Tango

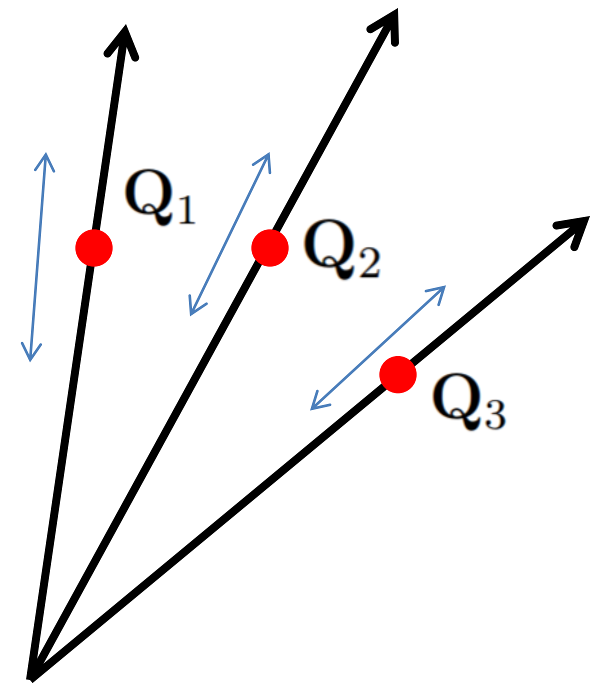
Microsoft Hololens

Where do you find applications for pose estimation?

Magic Leap

Pose estimation

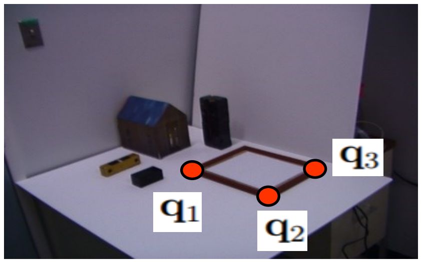


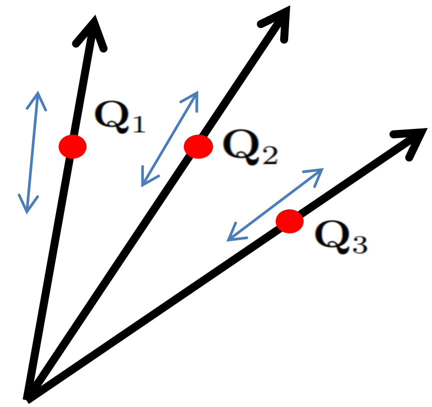


Camera Center

What other constraints can you use to find the pose?

Pose estimation





We are given 3 points on the image and their corresponding 3D points.

$$\mathbf{q}_1 = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \quad \mathbf{q}_2 = \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \quad \mathbf{q}_3 = \begin{pmatrix} u_3 \\ v_3 \\ 1 \end{pmatrix}$$

We compute the 2D distances between pairs of 2D points on the image: d_{12}, d_{13} and d_{23}

Thank You!