Digital Image Processing

Wei Qi Yan

Auckland University of Technology

Table of Contents

Vision

- Our eyes are a very effective sensor for recognition, navigation, obstacle avoidance and manipulation.
- Cameras mimic the function of an eye.
- We make use of cameras to create vision-based competencies for robots.
- We take use of digital images to recognize objects and navigate within the real world.
- Technological development has made it feasible for robots to use cameras as eyes.
- New algorithms, cheap sensors and plentiful computing power make camera as a practical sensor today.

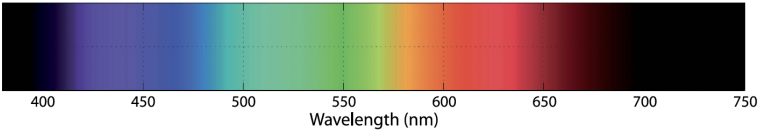


Image Formation

- In robot and human perception, we can deduce the size, shape and position of visual objects as well as other characteristics such as color and texture.
- A simple pinhole is able to create a perfect inverted image on the wall of a darkened room.
- In a digital camera, a glass or plastic lens forms an image using a semiconductor chip with an array of light sensitive devices to convert light to a digital image.
- The process of image formation involves a projection of the 3D world onto a 2D surface.
- The depth information is lost, we can no longer tell from the image whether is a large object in distance or a small closer object.

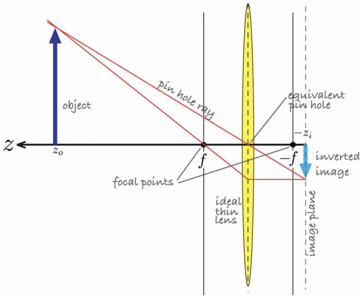
Image Formation

Image Formation

The z-coordinate of an object and its image are related by the lens law.

$$\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$$

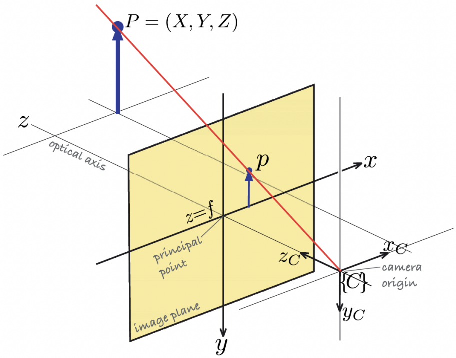
where z_o is the distance to the object, z_i is the distance to the image, and f is the focal length of the lens.



Corke, P. (2023) Robotics, Vision & Control. Springer

Central-Projection Model

The camera's coordinate frame is with the z-axis defining the centre of the field of view.



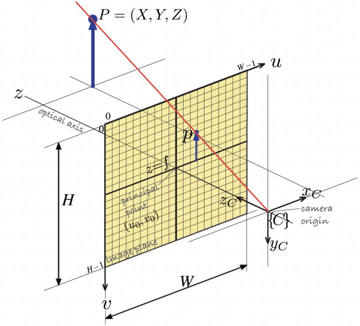
Corke, P. (2023) Robotics, Vision & Control. Springer

Image Formation

Perspective Transformation

A point at the world coordinates $P = (X, Y, Z)$ is projected to the image plane $p = (x, y)$ by using

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}$$

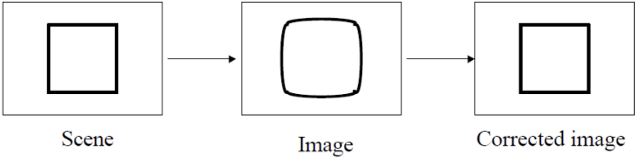


Corke, P. (2023) Robotics, Vision & Control. Springer

Camera Calibration

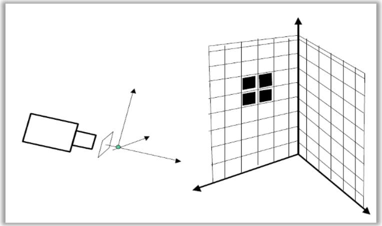
Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.

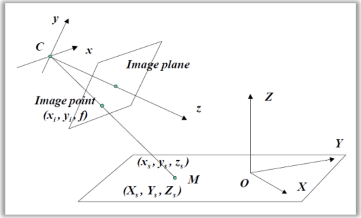
- Calibration relies on sets of world points whose relative coordinates are known and whose corresponding image-plane coordinates are also known.
- The intrinsic parameters (including distortion parameters) can be estimated as well as the relative pose of the chessboard in each image.
- Classical calibration requires of a 3D target.



Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.





Hence, $[x, y, z]^\tau = R \cdot [X, Y, Z]^\tau + T$; $[x', y']^\tau = [x/z, y/z]^\tau$; $[u, v]^\tau = [f_x \cdot x' + c_x, f_y \cdot y' + c_y]^\tau$

Camera Calibration

Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Hence, $[x, y, z]^\tau = R \cdot [X, Y, Z]^\tau + T$; $[x', y']^\tau = [x/z, y/z]^\tau$; $[u, v]^\tau = [f_x \cdot x' + c_x, f_y \cdot y' + c_y]^\tau$

Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Hence, $s \cdot \mathbf{m}' = \mathbf{A} \cdot [\mathbf{R}|\mathbf{T}]\mathbf{M}'$

Camera Calibration

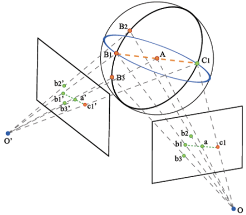
Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.

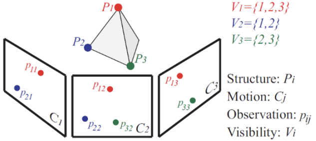
- (X, Y, Z) is the coordinate of a 3D point in the world coordinate space;
- (u, v) is the coordinate of the projection point in pixels;
- (c_x, c_y) is a principal point (that is usually at the image center);
- (f_x, f_y) is the focal lengths expressed in pixel-related units;
- $\mathbf{A}$ is called a camera matrix, or a matrix of intrinsic parameters;
- The joint rotation-translation matrix $[\mathbf{R}|\mathbf{T}]$ is called a matrix of extrinsic parameters.

Camera Calibration

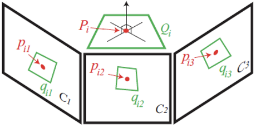
Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.





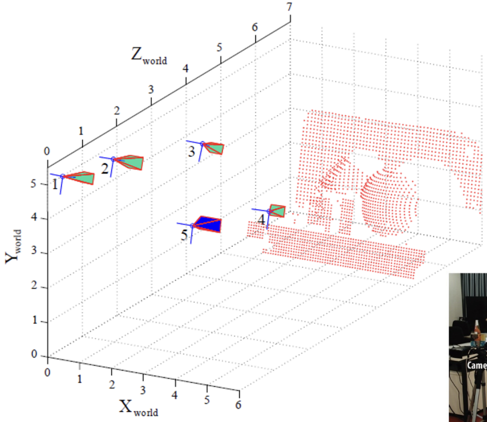
Notation: Three points P_1, P_2, P_3 are observed by three cameras C_1, C_2, C_3.

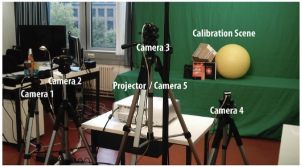


Given a patch (P_i, Q_i) and the visibility information V_i, we initialize matching images patches (p_{ij}, q_{ij}).

Camera Calibration

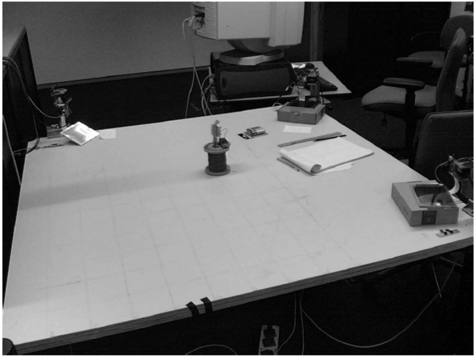
Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.





Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.



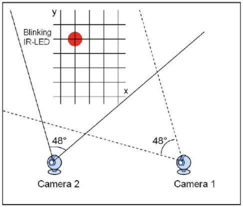


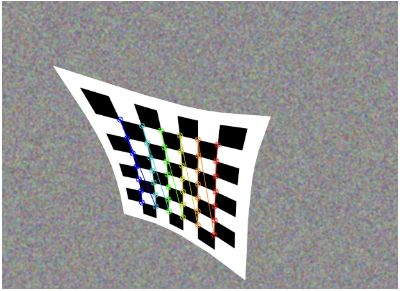
Fig. 2. Schematic of multiple camera setup.

Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system.

Steps of the algorithm:

1. Corner extraction
2. Point ordering
3. Point correspondences
4. Bundle adjustments



Questions?

Questions?

From a single image, we can tell:

1. An object is larger
2. An object is smaller
3. An object is larger or smaller
4. None of the given options

The right answer is:___

Questions?

Digital Image Processing

Digital Image Processing

- An image is a rectangular array of picture pixels.
- Robots will always gather imperfect images of the world due to noise, shadows, reflections and uneven illumination.
- The image processing algorithms operate pixel-wise on a single image or a pair of images, or on local groups of pixels within an image.

- *Monadic operations*: Each output pixel is a function of the corresponding input pixel. (e.g., histogram normalized, etc.)
- *Spatial operations*: Each pixel in the output image is a function of all pixels in a region surrounding the corresponding pixel in the input image. (e.g., convolution, etc.)

Spatial Operations

Each pixel in the output image is a function of all pixels in a region surrounding the corresponding pixel in the input image.

- Convolution is a linear spatial operator.

$$\mathbf{O}[u,v] = \sum_{(i,j)\in W} \mathbf{I}(u+i, v+j)\mathbf{K}(i,j), \forall (u,v) \in \mathbf{I}$$

where $\mathbf{K}$ is the convolution kernel, W is the image window. Hence,

$$\mathbf{O} = \mathbf{I} \otimes \mathbf{K}.$$

where $\otimes$ is convolution operator.

Spatial Operations

Each pixel in the output image is a function of all pixels in a region surrounding the corresponding pixel in the input image.

- Gaussian kernel is symmetric:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}}$$

where σ is the standard deviation parameter. For an example, the 5×5 matrix is:

1	4	7	4	1
4	20	33	20	4
7	33	55	33	7
4	20	33	20	4
1	4	7	4	1

Edge Detection

Filters can be designed to respond to edges at any arbitrary angle of digital images.

Sobel kernel can be considered as an image edge detector. The overall result is a weighted sum of the gradients.

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \qquad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

where $\mathbf{A}$ as the source image, $\mathbf{G}_x$ and $\mathbf{G}_y$ are two images which at each point contain the horizontal and vertical derivative approximations respectively, '$*$' denotes 2D convolution operation.

Template Matching

In template matching, we will find which parts of the input image are most similar to the template.

Each pixel in the output image is given by

$$\mathbf{O}(u, v) = s(\mathbf{T}, W)$$

where $\mathbf{T}$ is the template, W is the window centred at (u, v) in the input image. The function $s(I_1, I_2)$ is a scalar measure that describes the similarity of two equally sized images I_1 and I_2.

Digital Image Processing

Template Matching

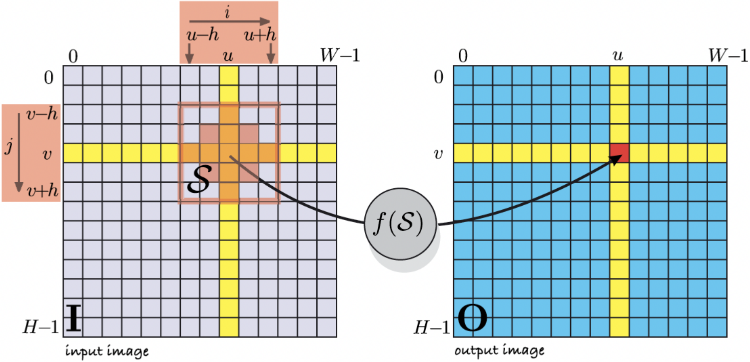
The difference between two images is the sum of the absolute differences (SAD) or the sum of the squared differences (SSD). These metrics are zero if the images are identical and increase with dissimilarity.

Sum of absolute differences	
SAD	$s = \Sigma_{(u,v)\in I} \left\lvert I_1[u,v] - I_2[u,v] \right\rvert$
ZSAD	$s = \Sigma_{(u,v)\in I} \left\lvert (I_1[u,v] - \bar{I}_1) - (I_2[u,v] - \bar{I}_2) \right\rvert$
Sum of squared differences	
SSD	$s = \Sigma_{(u,v)\in I} \left(I_1[u,v] - I_2[u,v]\right)^2$
ZSSD	$s = \Sigma_{(u,v)\in I} \left((I_1[u,v] - \bar{I}_1) - (I_2[u,v] - \bar{I}_2)\right)^2$
Cross correlation	
NCC	$s = \dfrac{\Sigma_{(u,v)\in I} I_1[u,v] \cdot I_2[u,v]}{\sqrt{\Sigma_{(u,v)\in I} I_1^2[u,v] \cdot \Sigma_{(u,v)\in I} I_2^2[u,v]}}$
ZNCC	$s = \dfrac{\Sigma_{(u,v)\in I} (I_1[u,v] - \bar{I}_1) \cdot (I_2[u,v] - \bar{I}_2)}{\sqrt{\Sigma_{(u,v)\in I} (I_1[u,v] - \bar{I}_1)^2 \cdot \Sigma_{(u,v)\in I} (I_2[u,v] - \bar{I}_2)^2}}$

Corke, P. (2023) Robotics, Vision & Control. Springer

Image Morphology

Image morphology is concerned with the form or shape of objects in the image (Binary).



Corke, P. (2023) Robotics, Vision & Control. Springer

Image Morphology

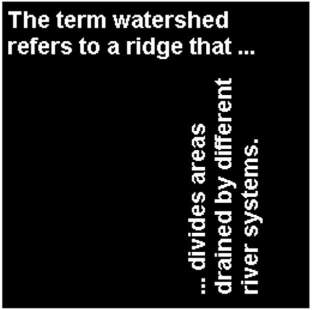
In morphological operations: Erosion.

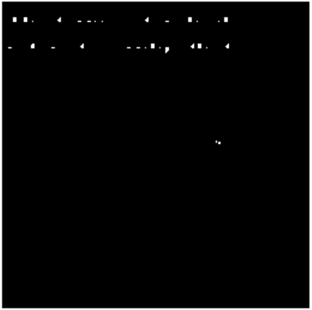
$$
\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&0&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1\\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

Erosion (Blue): If B (Green) is completely contained by A (Red), the pixel is retained, else deleted.

Image Morphology

MATLAB morphological operations: Erosion (Binary image).





https://au.mathworks.com/help/images/ref/imerode.html

Image Morphology

MATLAB morphological operations: Erosion (Grayscale image).

(a)

(b)

https://au.mathworks.com/help/images/ref/imerode.html

Digital Image Processing

Image Morphology

In morphological operations: Dilation.

0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	1	1	1	0
0	1	1	1	1	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	0
0	1	1	0	0	0	1	1	1	1	0
0	1	1	0	0	0	1	1	1	1	0
0	1	1	0	0	0	1	1	1	1	0
0	1	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0

1	1	1
1	1	1
1	1	1

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	0	0

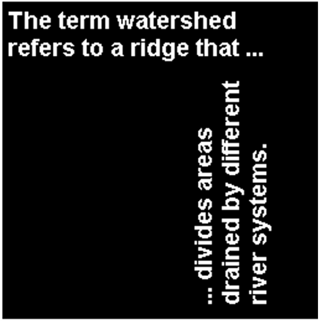
Dilation (Blue): Each pixel in **A** (Red) with a value 1 will be superimposed with *B* (Green). All pixels after superimposed with **B** (Green) is included in the dilation (Blue).

Digital Image Processing

Image Morphology

In morphological operations: Dilation (Binary image).

Original



Dilated

https://au.mathworks.com/help/images/ref/imdilate.html

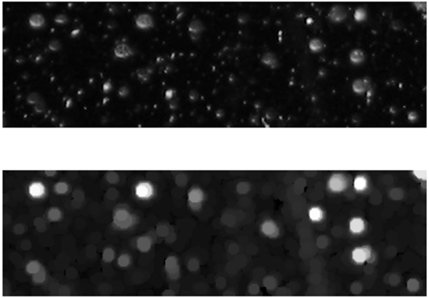
Image Morphology

In morphological operations: Dilation (Grayscale image).

https://au.mathworks.com/help/images/ref/imdilate.html

Image Morphology

In morphological operations, the sequence of operations, erosion then dilation, is known as opening. The operations in the inverse order, dilation then erosion is closing.



(a)

(b)

https://au.mathworks.com/help/images/ref/imopen.html

Image Morphology

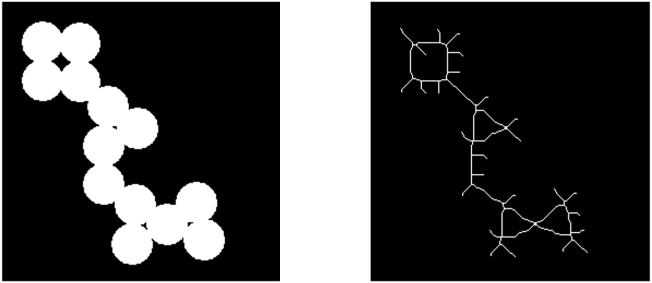
In morphological operations, the sequence of operations, erosion then dilation, is known as opening. The operations in the inverse order, dilation then erosion is closing.

(a) (b)

https://au.mathworks.com/help/images/ref/imclose.html

Image skeletonisation

Image skeletonisation extracts the centerline while preserving the topology of the objects.

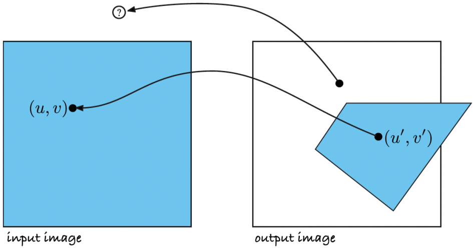


(a) (b)

https://au.mathworks.com/help/images/ref/bwmorph.html

Image Warping

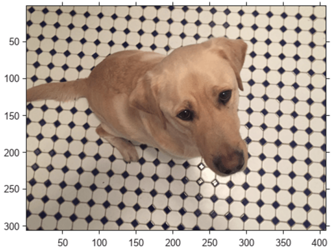
- Image warping is a transformation of the pixel coordinates.
- The pixel (u', v') in the output image is sourced from the pixel at (u, v) in the input image as indicated by the arrow.

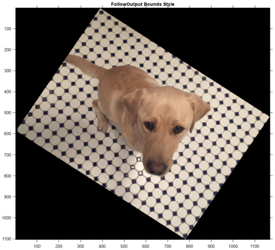


Corke, P. (2023) Robotics, Vision & Control. Springer

Image Warping

- Image warping is a transformation of the pixel coordinates.
- The pixel (u', v') in the output image is sourced from the pixel at (u, v) in the input image as indicated by the arrow.





https://au.mathworks.com/help/images/ref/imwarp.html

Questions?

Questions?

Robots will always gather imperfect images of the real world, due to:

1. noise
2. shadows
3. reflections
4. none of the given options

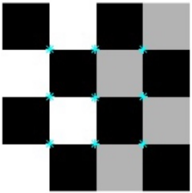
The wrong answer is:___

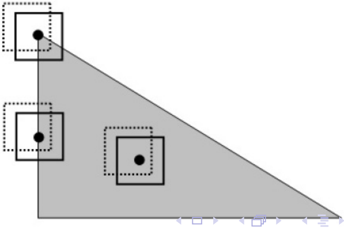
Questions?

Visual Features

In terms of visual features such as object size, position, and shape related to robotics, we have:

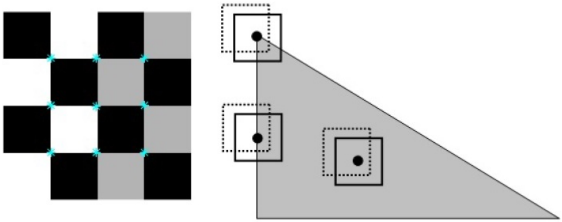
- **Bounding Boxes**: The smallest rectangle that encloses the region.
- **Moments**: A computationally cheap class of image features which can describe region size and location as well as shape.
- **Invariance**: The shape of an object is invariant to translation, rotation and scale.





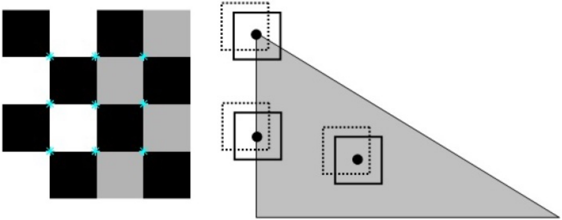
Visual Features

- An interest point of images is a point that has a high image gradient in orthogonal directions.
- Corner detector is computed from image gradients and robust to offsets in illumination, the structure are invariant to rotation.



Corners

- A corner in an image is given at a pixel where two edges of different directions intersect.
- Corners usually lie on high-contrast regions of the image.
- Relative positions between corners in the original scene shouldn't change.



Vision Intelligence II: Visual Information Retrieval

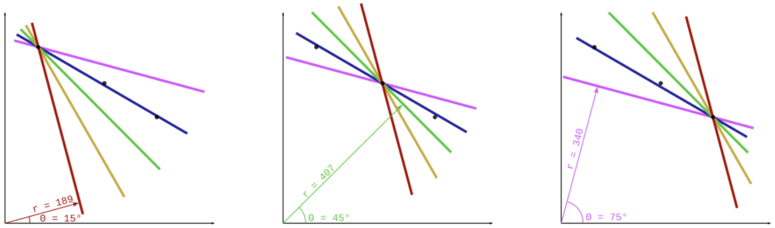
Corners

- Corners are invariant to scaling, orientation, and distortions.
- The best match for each pair of corners is found by identifying its nearest neighbor of corners.
- The nearest neighbors are defined as the corners with the minimum distance from the given descriptor vector.

Visual Features

Line Features: Hough transform estimates the direction of the line by fitting lines to the edge pixels.

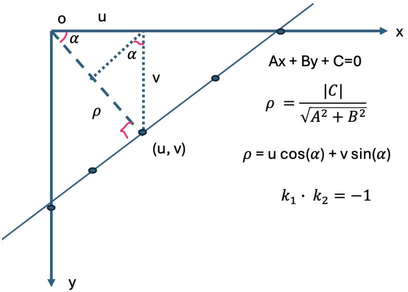
There are numerous lines that pass through that point. If the point could vote for these lines, then each possible line passing through the point would receive one vote.



Visual Features

Line Features: Hough transform estimates the direction of the line by fitting lines to the edge pixels.

There are numerous lines that pass through that point. If the point could vote for these lines, then each possible line passing through the point would receive one vote.

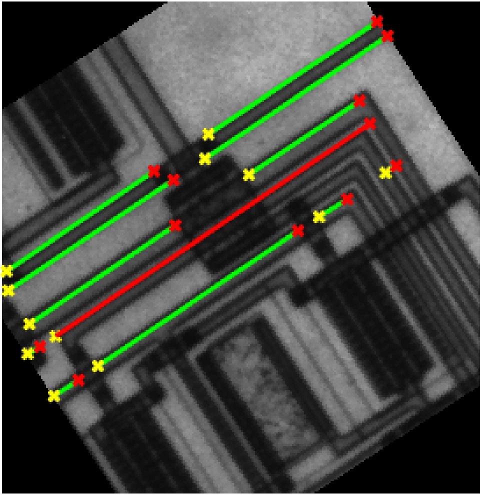


MATLAB Hough Transform

- Read the image.
- Find the edges in the image using the edge function.
- Compute the Hough transform of the binary image returned by edge.
- Find the peaks in the Hough transform matrix using the hough peaks function.
- Find lines in the image using the hough lines function.
- Create a plot that displays the original image with the lines superimposed on it.

https://au.mathworks.com/help/images/ref/houghlines.html

MATLAB Hough Transform



Questions?

Questions?

Corners are invariant to:

1. scaling
2. orientation
3. distortions
4. none of the given options

The wrong answer is:___

Questions?

Learning Objectives

- Derive solutions for particular robotic vision and visual control tasks characterised by specifics of image data and deep learning algorithms.
- Critically evaluate the performance of robotic vision with deep learning algorithms, bench mark data, performance measures, and ways to define ground truth.