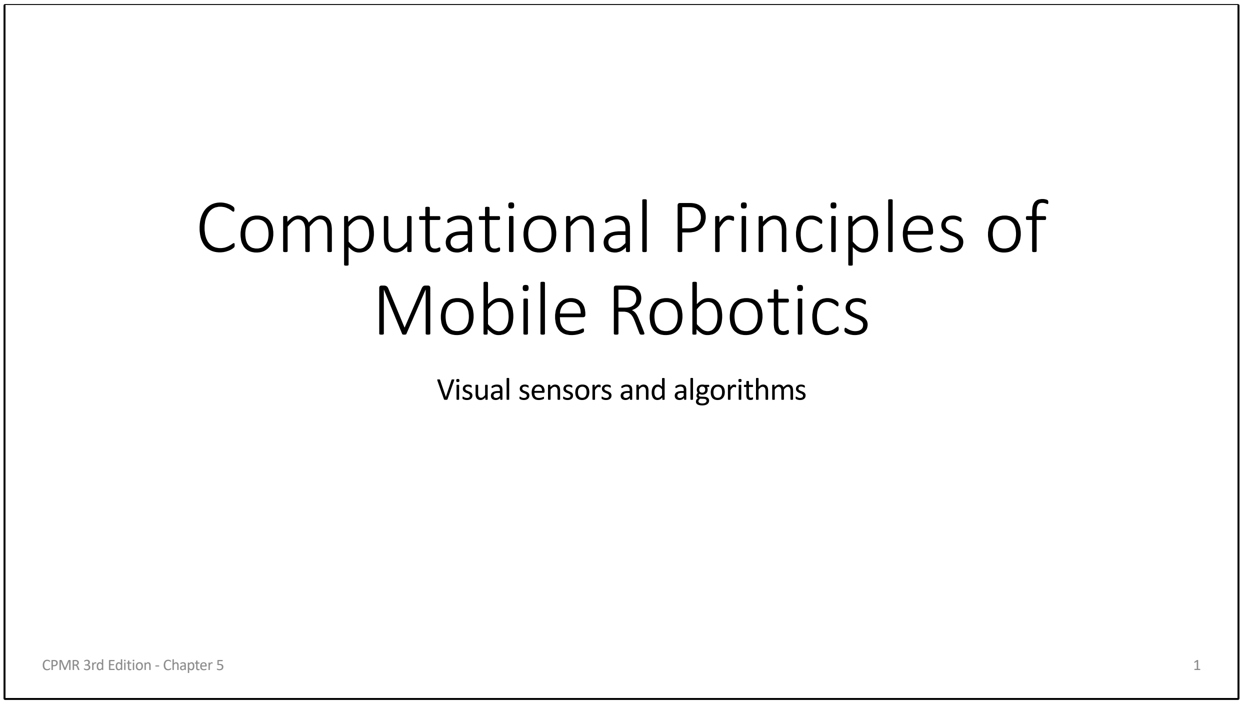
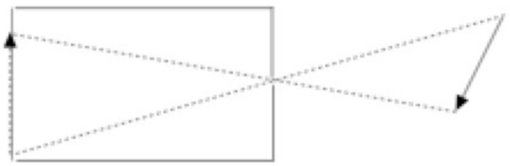


Computational Principles of Mobile Robotics

Visual sensors and algorithms

5.1 Visual sensors

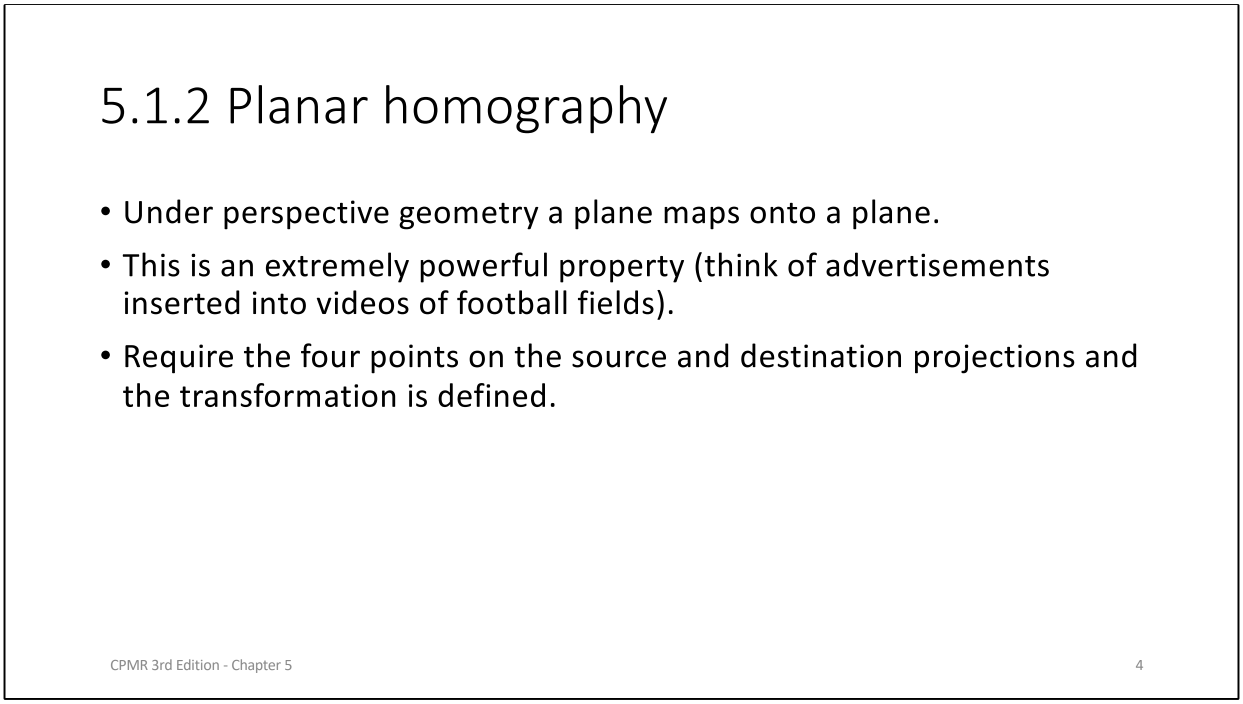
- Simplest model is the pin hole cameras
  - Collects light emitted by/reflected from structures in the environments.
  - Obtains **direction** to these structures
    - Looses depth.



5.1.1 Perspective projection

- Model as perspective projection
  - The point (X_1, X_2, X_3) is converted to homogeneous coordinates $[X_1\ X_2\ X_3\ 1]^T$
  - And we can write the image point as (x_1/x_3, x_2/x_3).
  - Where,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \tilde{P} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$$

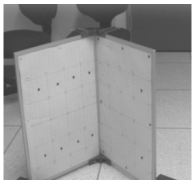


5.1.2 Planar homography

- Under perspective geometry a plane maps onto a plane.
- This is an extremely powerful property (think of advertisements inserted into videos of football fields).
- Require the four points on the source and destination projections and the transformation is defined.

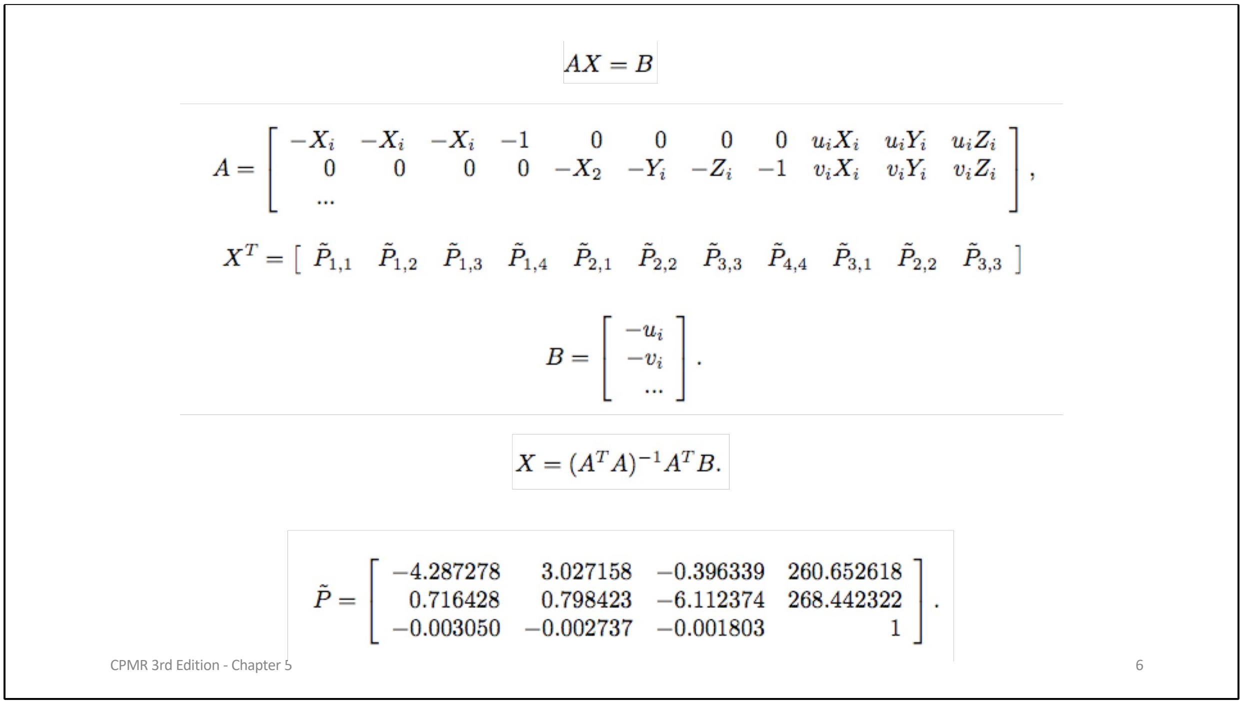
5.1.3 Camera calibration

- Involves computing the matrix P given a set of points {(x,y,z,u,v)} and estimating P
- P is typically decomposed (if necessary) into intrinsic (camera internal) and extrinsic (camera external) parameters.
- Perhaps the simplest method is the *direct method* given in the text
  - Explicitly collect point set and construct linear optimization problem



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \tilde{P} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$$

$$\sum((\tilde{P}X_i)_1 - (\tilde{P}X_i)_3 u_i)^2 + ((\tilde{P}X_i)_2 - (\tilde{P}X_i)_3 v_i)^2$$



$$AX = B$$

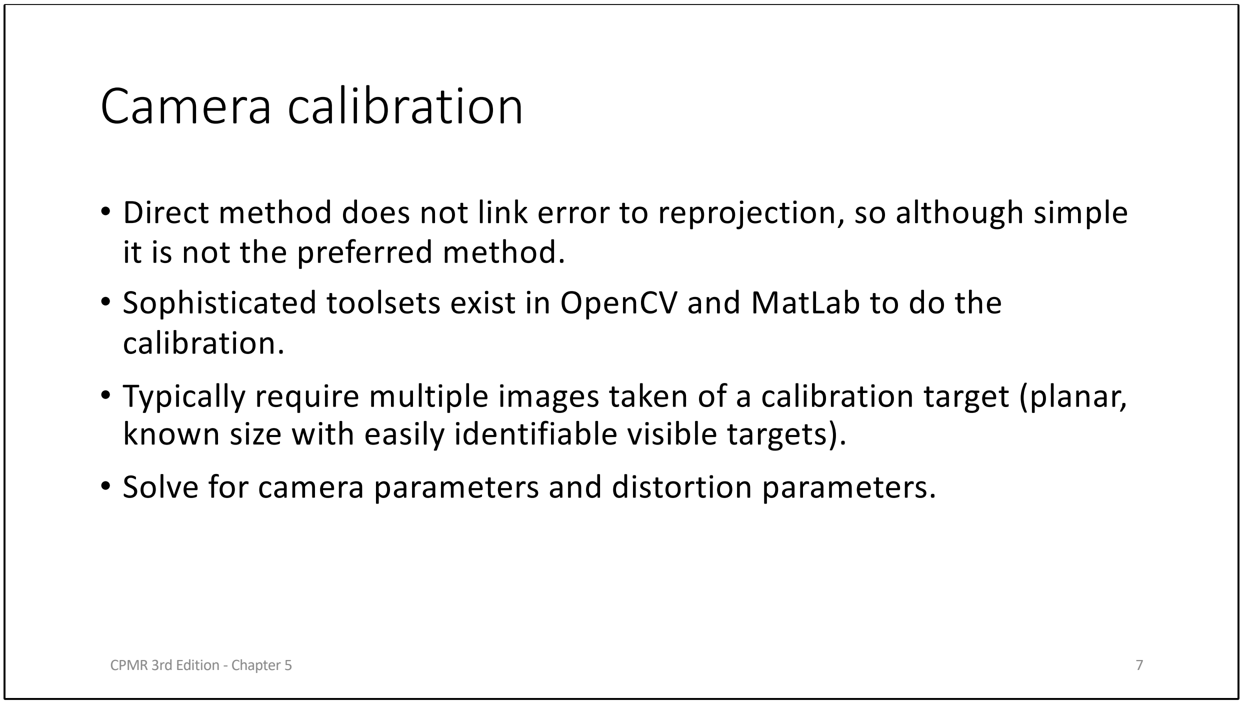
$$A = \begin{bmatrix} -X_i & -X_i & -X_i & -1 & 0 & 0 & 0 & 0 & u_iX_i & u_iY_i & u_iZ_i \\ 0 & 0 & 0 & 0 & -X_2 & -Y_i & -Z_i & -1 & v_iX_i & v_iY_i & v_iZ_i \\ \dots & & & & & & & & & & \end{bmatrix},$$

$$X^T = \begin{bmatrix} \tilde{P}_{1,1} & \tilde{P}_{1,2} & \tilde{P}_{1,3} & \tilde{P}_{1,4} & \tilde{P}_{2,1} & \tilde{P}_{2,2} & \tilde{P}_{3,3} & \tilde{P}_{4,4} & \tilde{P}_{3,1} & \tilde{P}_{2,2} & \tilde{P}_{3,3} \end{bmatrix}$$

$$B = \begin{bmatrix} -u_i \\ -v_i \\ \dots \end{bmatrix}.$$

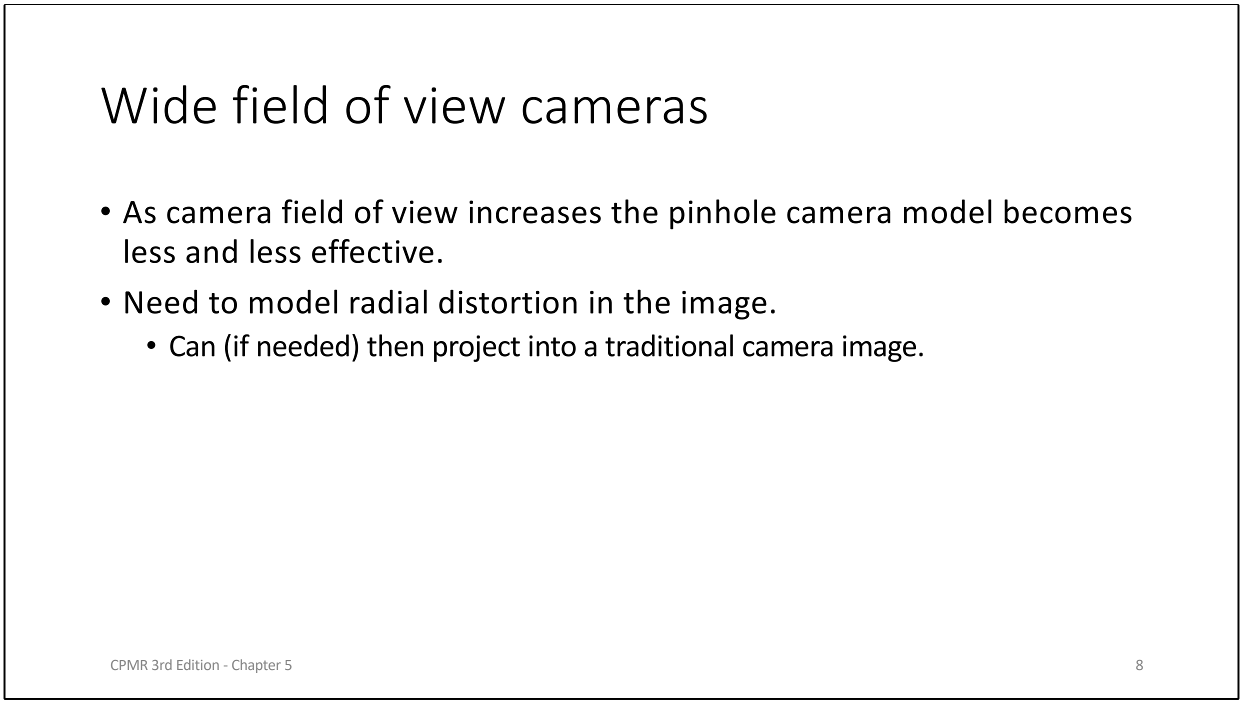
$$X = (A^TA)^{-1}A^TB.$$

$$\tilde{P} = \begin{bmatrix} -4.287278 & 3.027158 & -0.396339 & 260.652618 \\ 0.716428 & 0.798423 & -6.112374 & 268.442322 \\ -0.003050 & -0.002737 & -0.001803 & 1 \end{bmatrix}.$$



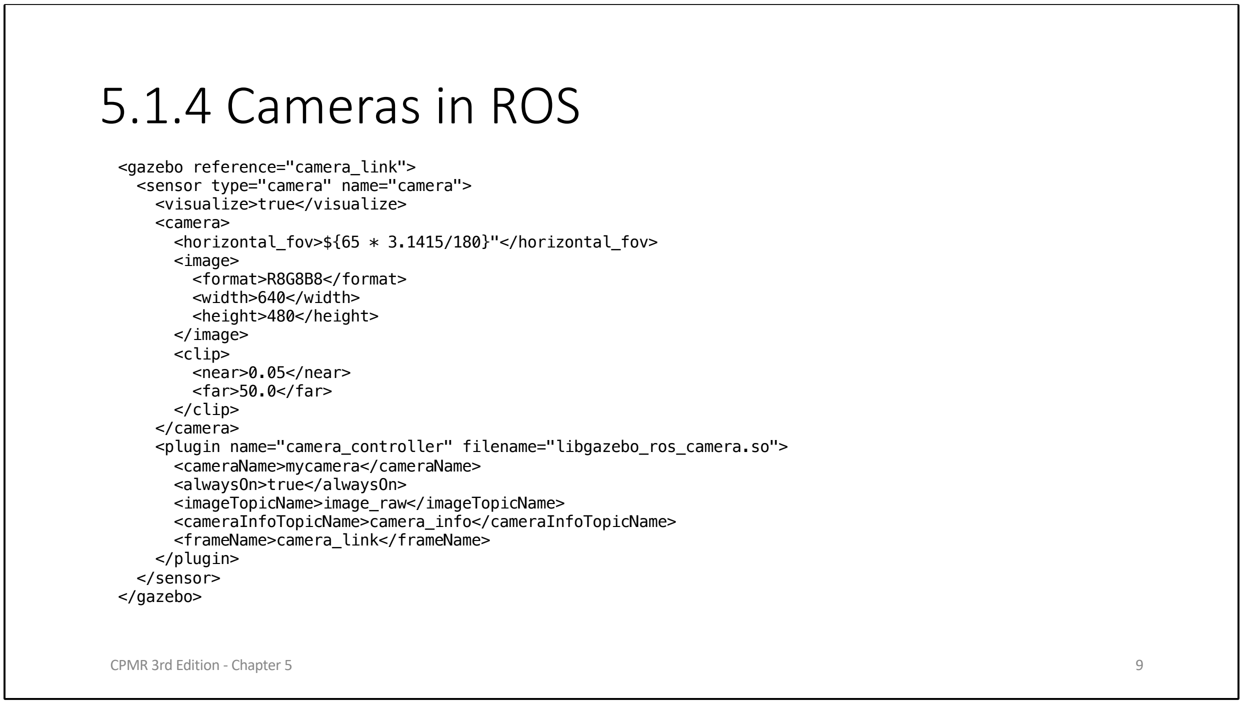
Camera calibration

- Direct method does not link error to reprojection, so although simple it is not the preferred method.
- Sophisticated toolsets exist in OpenCV and MatLab to do the calibration.
- Typically require multiple images taken of a calibration target (planar, known size with easily identifiable visible targets).
- Solve for camera parameters and distortion parameters.



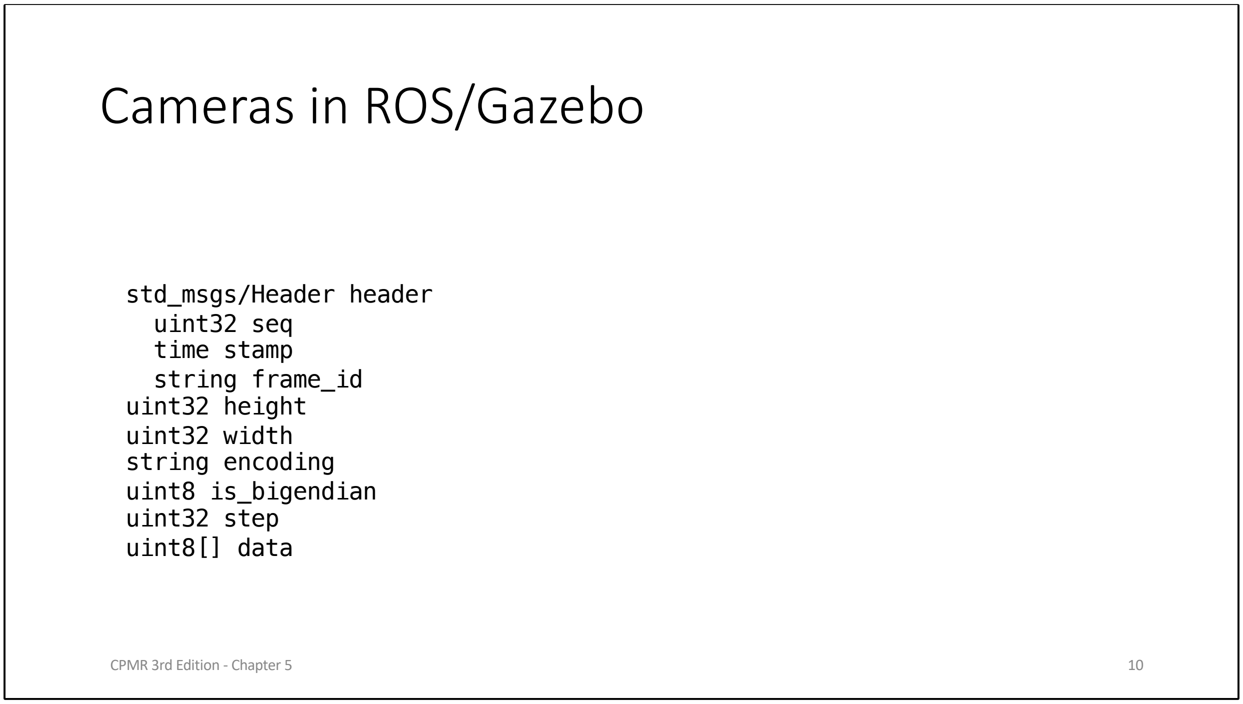
Wide field of view cameras

- As camera field of view increases the pinhole camera model becomes less and less effective.
- Need to model radial distortion in the image.
    - Can (if needed) then project into a traditional camera image.



5.1.4 Cameras in ROS

```
<gazebo reference="camera_link">
  <sensor type="camera" name="camera">
    <visualize>true</visualize>
    <camera>
      <horizontal_fov>${65 * 3.1415/180}"</horizontal_fov>
      <image>
        <format>R8G8B8</format>
        <width>640</width>
        <height>480</height>
      </image>
      <clip>
        <near>0.05</near>
        <far>50.0</far>
      </clip>
    </camera>
    <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
      <cameraName>mycamera</cameraName>
      <alwaysOn>true</alwaysOn>
      <imageTopicName>image_raw</imageTopicName>
      <cameraInfoTopicName>camera_info</cameraInfoTopicName>
      <frameName>camera_link</frameName>
    </plugin>
  </sensor>
</gazebo>
```

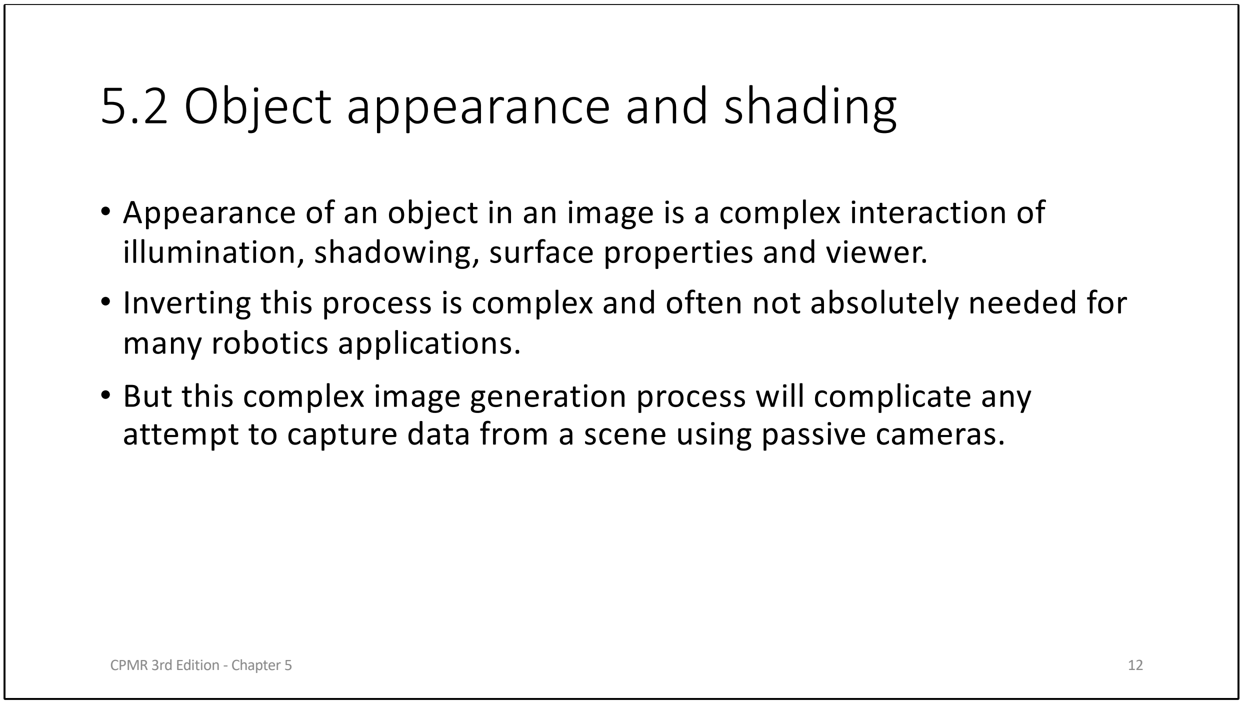


Cameras in ROS/Gazebo

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
uint32 height
uint32 width
string encoding
uint8 is_bigendian
uint32 step
uint8[] data
```

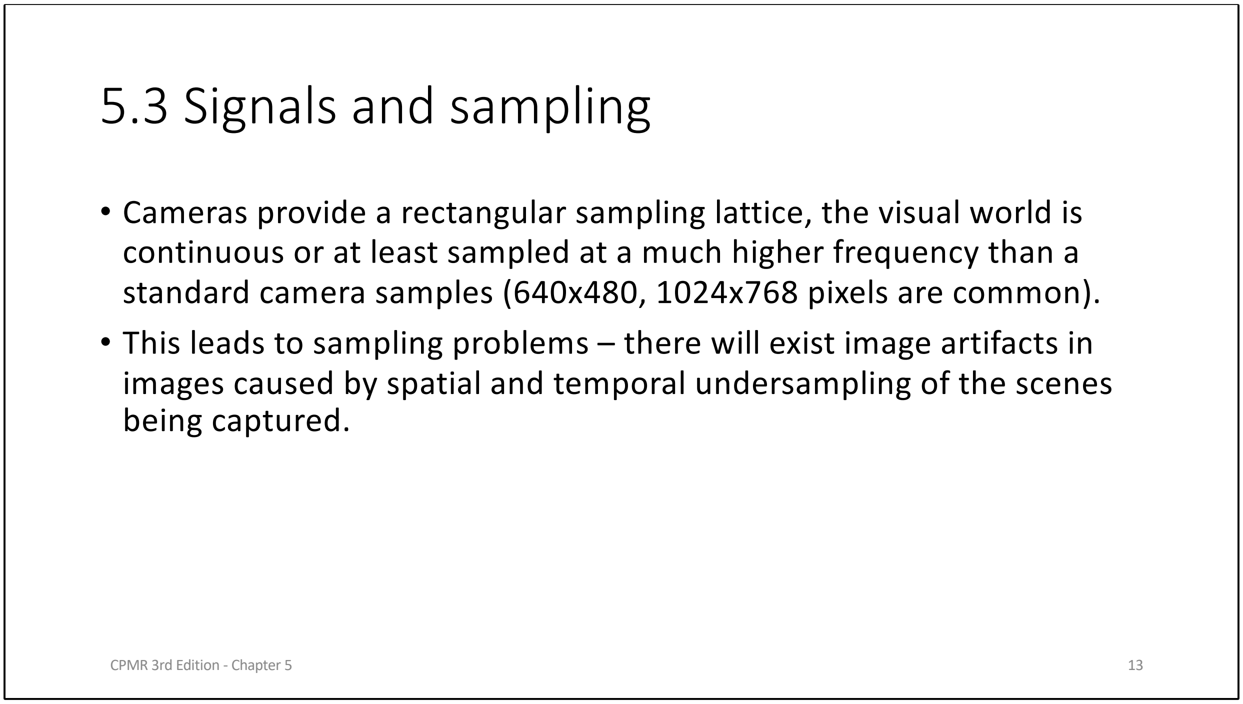
Cameras in ROS/Gazebo

```
header:
  seq: 73442
  stamp:
    secs: 56872
    nsecs: 773000000
  frame_id: "camera_link"
height: 480
width: 640
distortion_model: "plumb_bob"
D: [0.0, 0.0, 0.0, 0.0, 0.0]
K: [502.3182365128262, 0.0, 320.5, 0.0, 502.3182365128262, 240.5, 0.0, 0.0, 1.0]
R: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
P: [502.3182365128262, 0.0, 320.5, -0.0, 0.0, 502.3182365128262, 240.5, 0.0, 0.0, 0.0, 1.0, 0.0]
binning_x: 0
binning_y: 0
roi:
  x_offset: 0
  y_offset: 0
  height: 0
  width: 0
  do_rectify: False
```



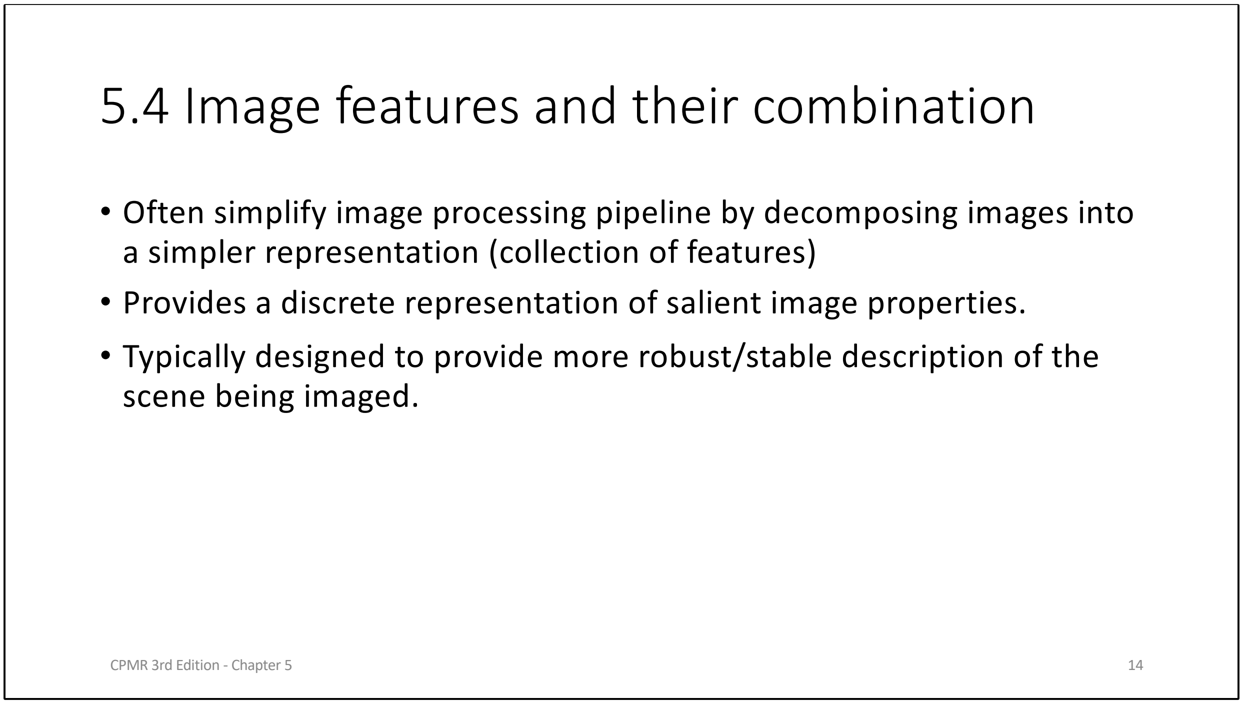
5.2 Object appearance and shading

- Appearance of an object in an image is a complex interaction of illumination, shadowing, surface properties and viewer.
- Inverting this process is complex and often not absolutely needed for many robotics applications.
- But this complex image generation process will complicate any attempt to capture data from a scene using passive cameras.



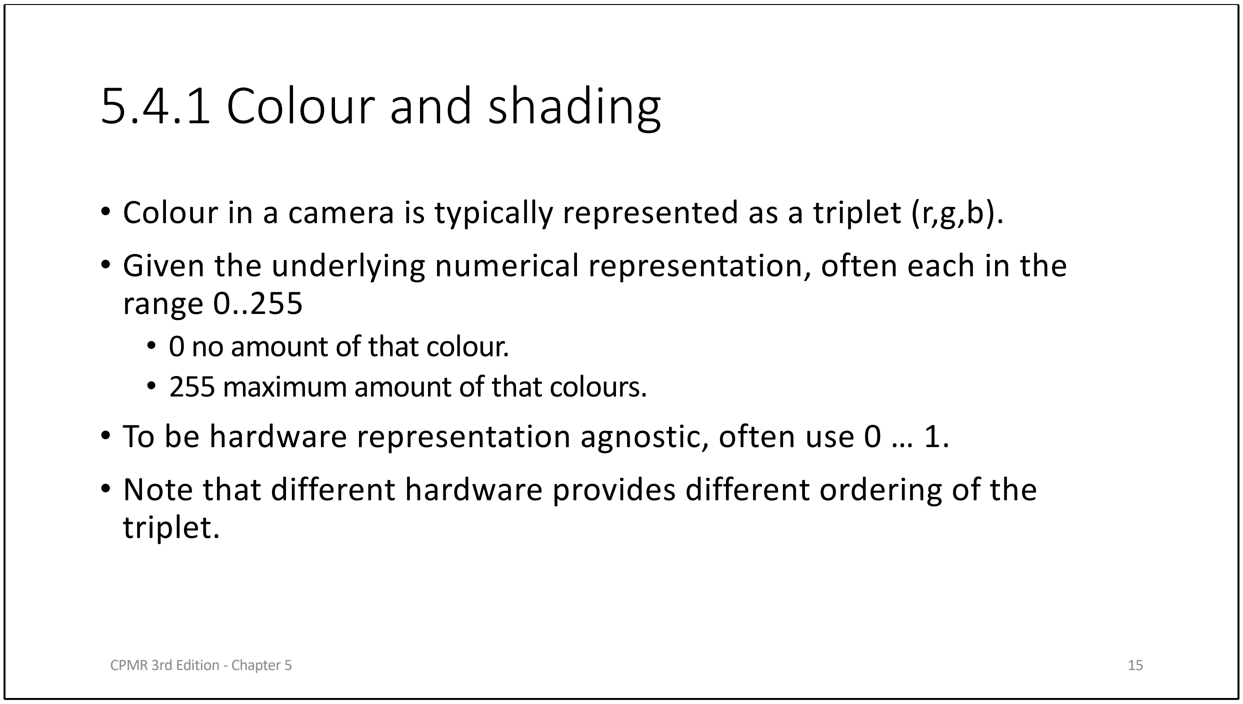
5.3 Signals and sampling

- Cameras provide a rectangular sampling lattice, the visual world is continuous or at least sampled at a much higher frequency than a standard camera samples (640x480, 1024x768 pixels are common).
- This leads to sampling problems – there will exist image artifacts in images caused by spatial and temporal undersampling of the scenes being captured.



5.4 Image features and their combination

- Often simplify image processing pipeline by decomposing images into a simpler representation (collection of features)
- Provides a discrete representation of salient image properties.
- Typically designed to provide more robust/stable description of the scene being imaged.



5.4.1 Colour and shading

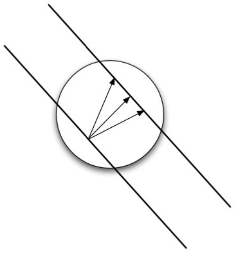
- Colour in a camera is typically represented as a triplet (r,g,b).
- Given the underlying numerical representation, often each in the range 0..255
  - 0 no amount of that colour.
  - 255 maximum amount of that colours.
- To be hardware representation agnostic, often use 0 … 1.
- Note that different hardware provides different ordering of the triplet.

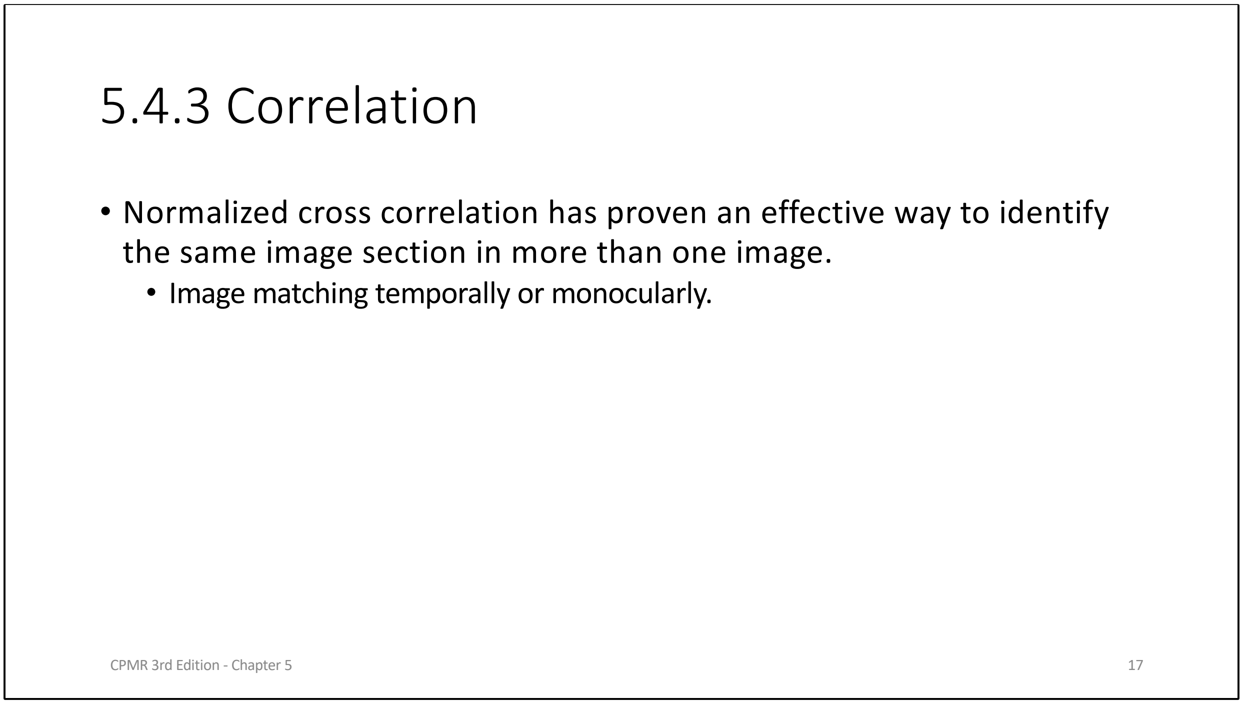
5.4.2 Image brightness constraint

- For a sufficiently small image patch moving over time
  - f(x,y,t)=f(f+dx,y+dy,t+dt).
- Can solve this to obtain

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}.$$

- Leads to the aperture problem.



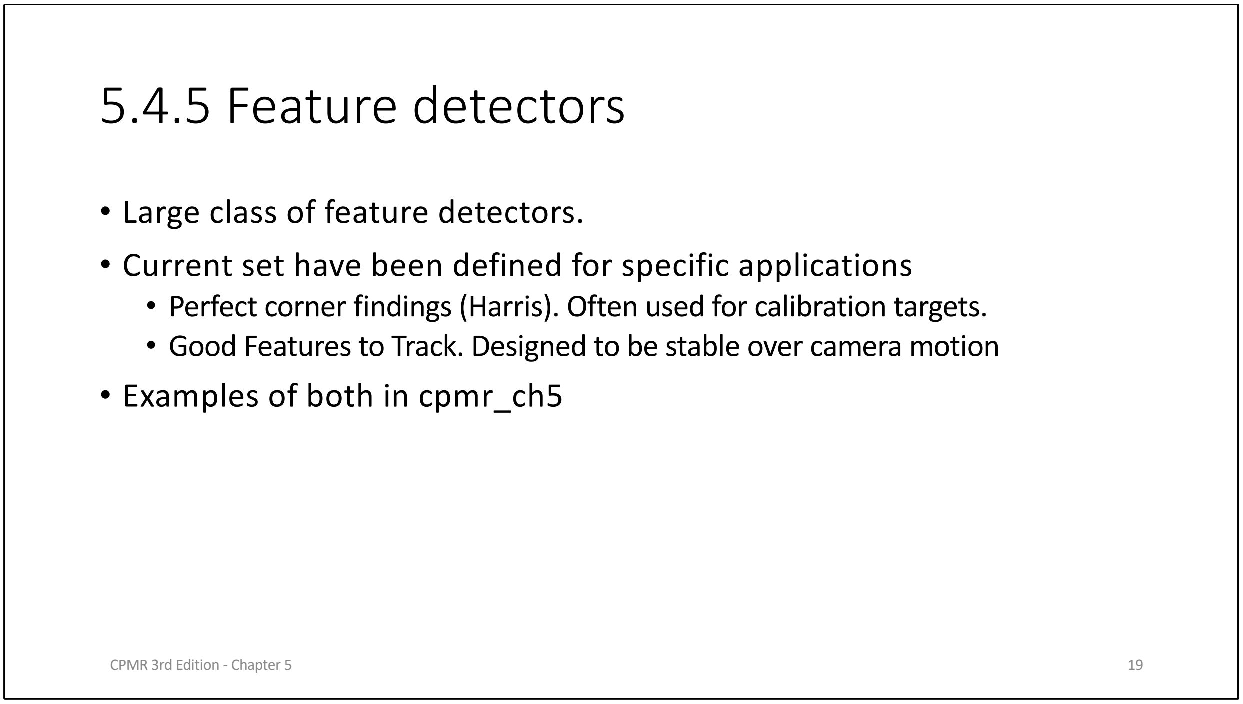


5.4.3 Correlation

- Normalized cross correlation has proven an effective way to identify the same image section in more than one image.
  - Image matching temporally or monocularly.

5.4.4 Fourier methods

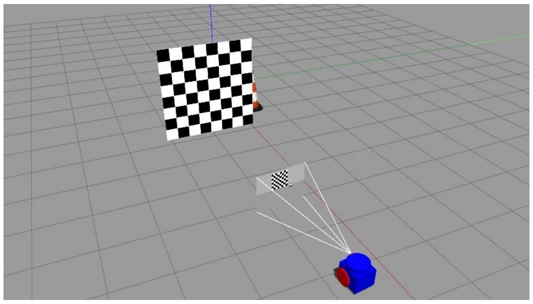
- Large cadre of effective signal processing operators that can be exploited in image processing.
- Fundamentally can represent an image as a sum of sine waves.
  - Can exploit a large literature here.

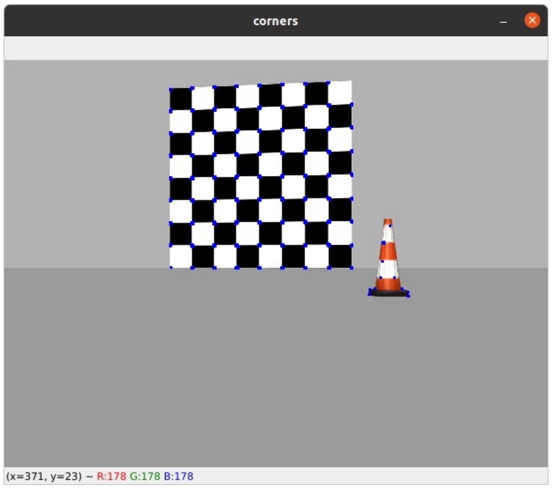


5.4.5 Feature detectors

- Large class of feature detectors.
- Current set have been defined for specific applications
  - Perfect corner findings (Harris). Often used for calibration targets.
  - Good Features to Track. Designed to be stable over camera motion
- Examples of both in cpmr_ch5

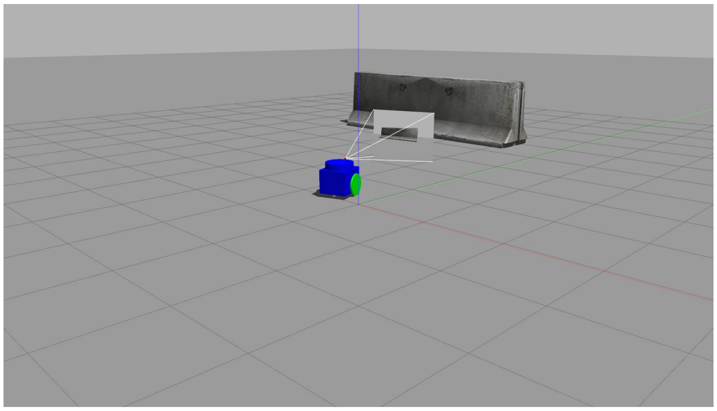
Harris Corner Detector

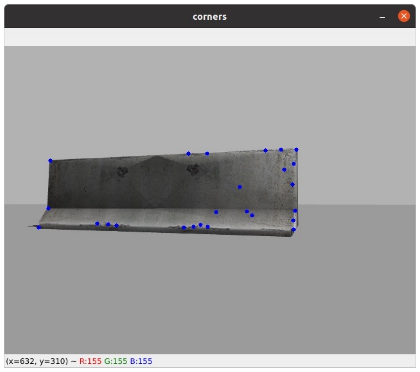




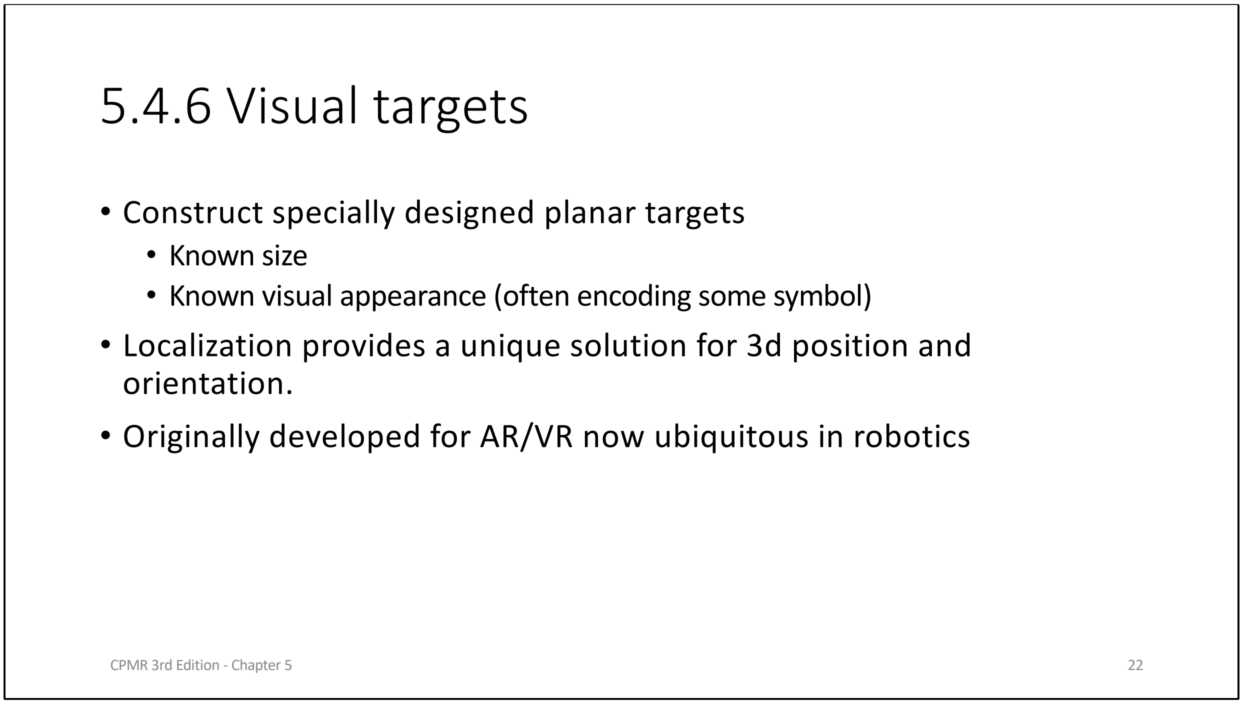
(x=371, y=23) ~ R:178 G:178 B:178

Good features to track





Features designed to be stable under (small) camera motion

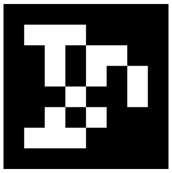


5.4.6 Visual targets

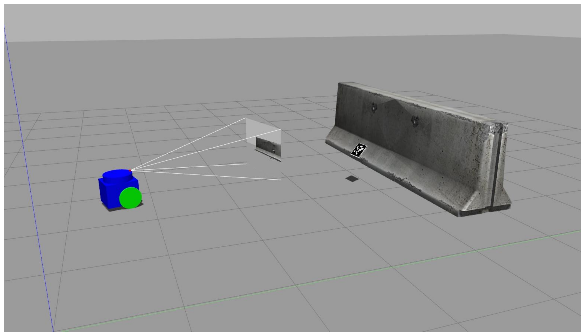
- Construct specially designed planar targets
  - Known size
  - Known visual appearance (often encoding some symbol)
- Localization provides a unique solution for 3d position and orientation.
- Originally developed for AR/VR now ubiquitous in robotics

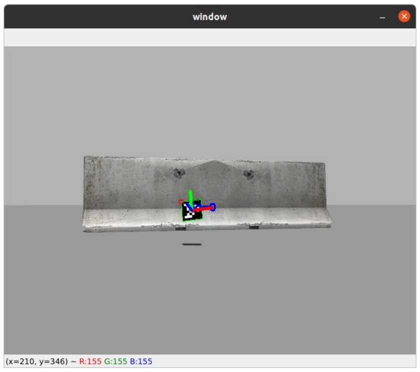
Aruco targets

- Supported in OpenCV (3rd party, but free to use)
- Supports a number of families of targets
  - Provides different numbers of symbols, different sizes



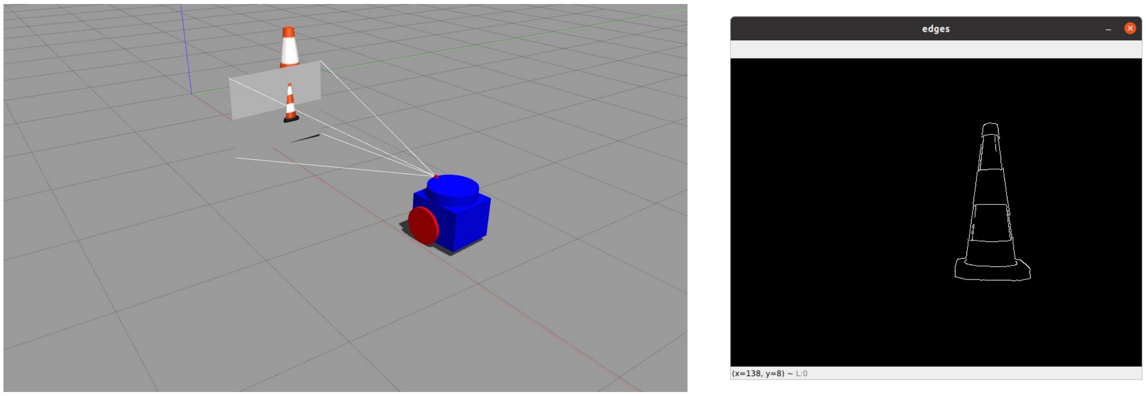
Aruco targets





(x=210, y=346) ~ R:155 G:155 B:155

5.4.7 Edges



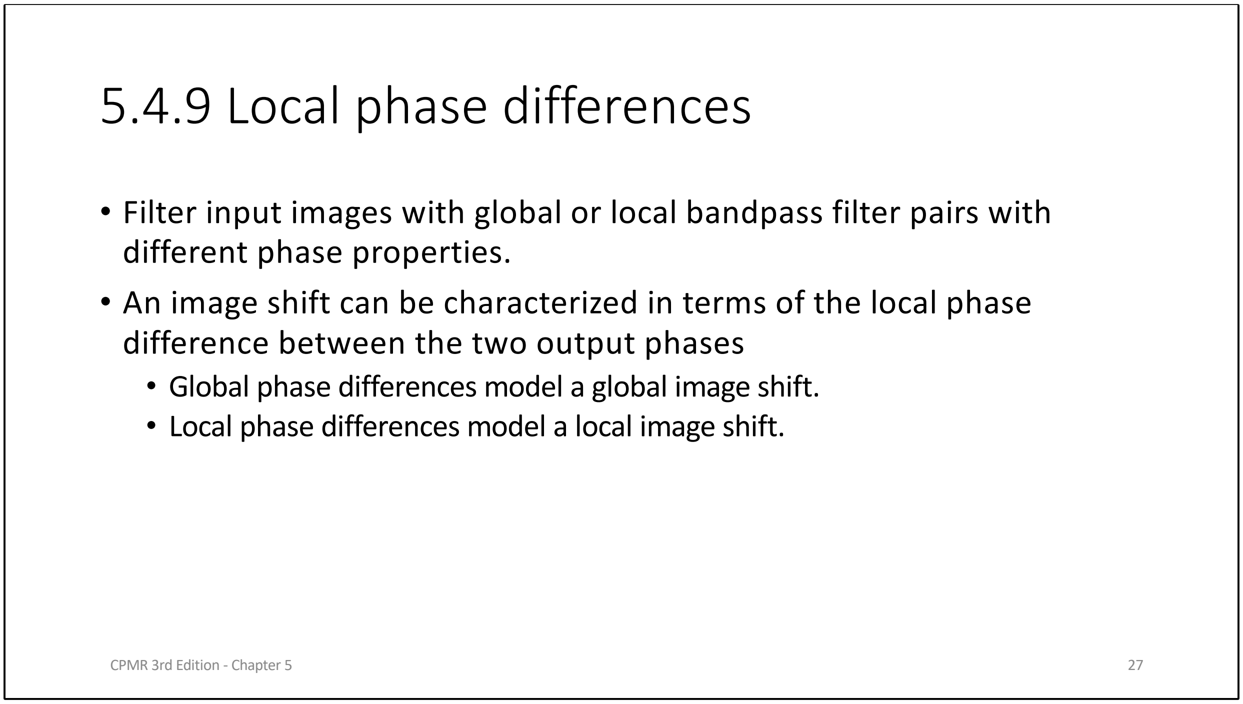
Canny Edge Detection

5.4.8 Image filters

- The output of different image filters can be used to model/enhance specific image features.
- Given k filters, two image points can be compared by minimizing

$$e_m = \sum_k |F_k * I_r(i,j) - F_k * I_l(i+h, j+v)|$$

For a given image shift (h,v)

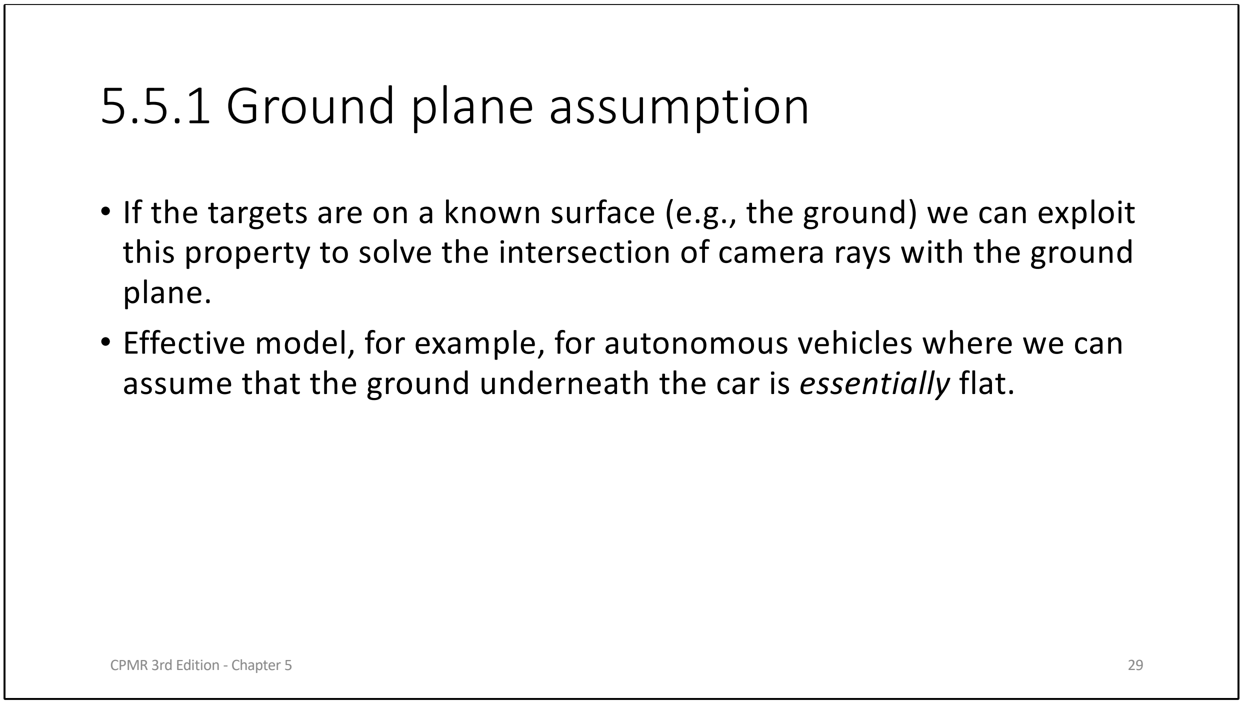


5.4.9 Local phase differences

- Filter input images with global or local bandpass filter pairs with different phase properties.
- An image shift can be characterized in terms of the local phase difference between the two output phases
  - Global phase differences model a global image shift.
  - Local phase differences model a local image shift.

5.5 Obtaining depth

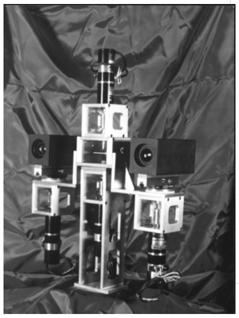
- If we don’t have special purpose targets and we wish to obtain depth more generally, we cannot rely easily on a single view of a single image.
- We can exploit other properties to help solve this problem.

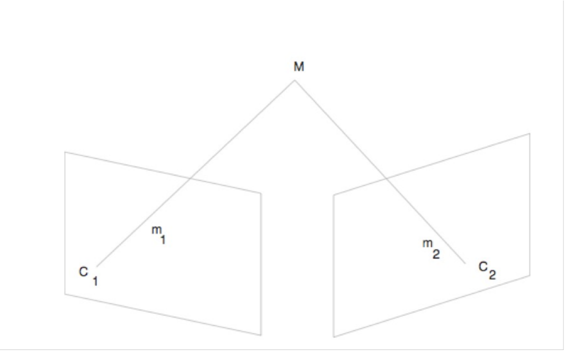


5.5.1 Ground plane assumption

- If the targets are on a known surface (e.g., the ground) we can exploit this property to solve the intersection of camera rays with the ground plane.
- Effective model, for example, for autonomous vehicles where we can assume that the ground underneath the car is *essentially* flat.

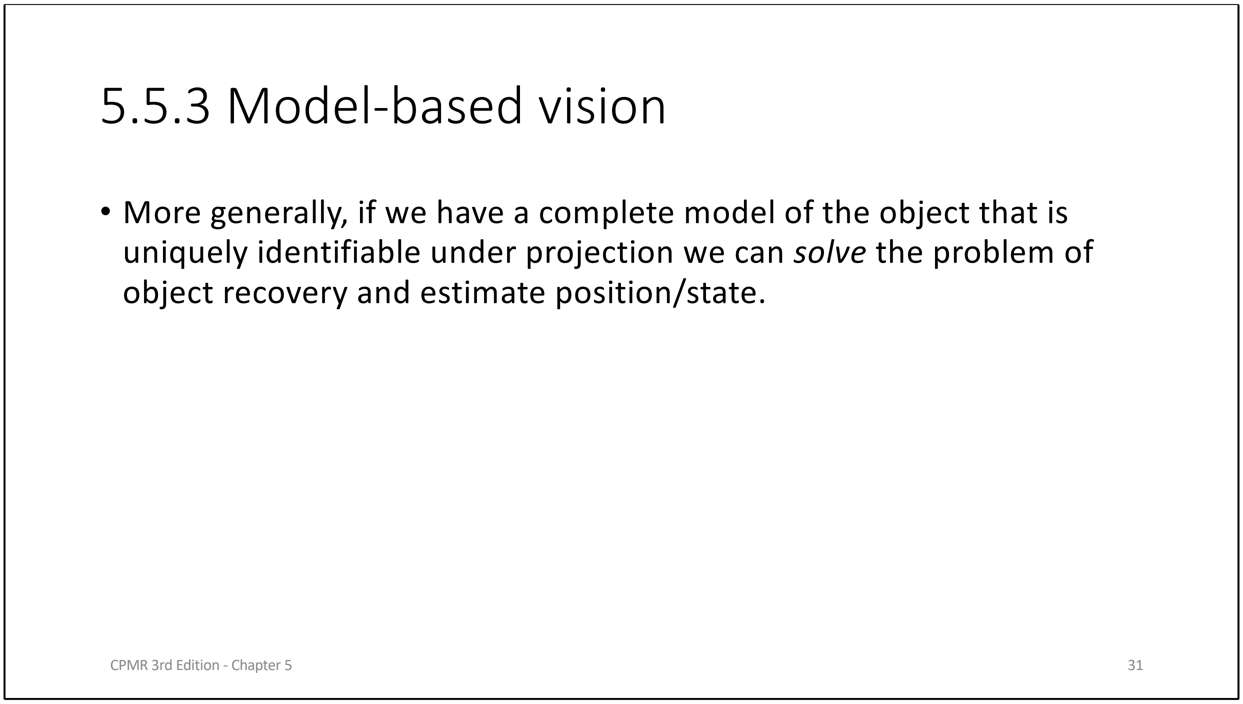
5.5.2 Multiple cameras





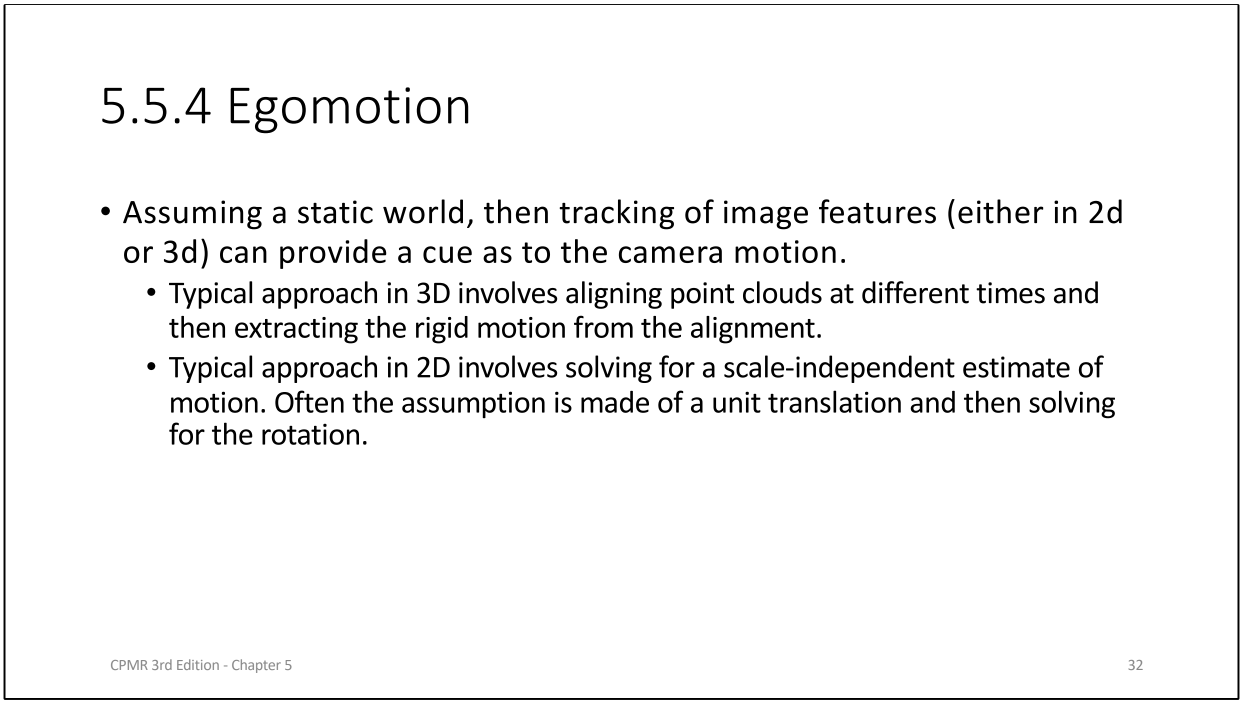
$$(u_i, v_i) = \left(\frac{(\tilde{P}_i X)_1}{(\tilde{P}_i X)_3}, \frac{(\tilde{P}_i X)_2}{(\tilde{P}_i X)_3} \right)$$

$$\begin{bmatrix} (u_i \tilde{P}_i)_3 - (\tilde{P}_i)_1 \\ (v_i \tilde{P}_i)_3 - (\tilde{P}_i)_2 \\ \ldots \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \ldots \\ 0 \end{bmatrix}$$



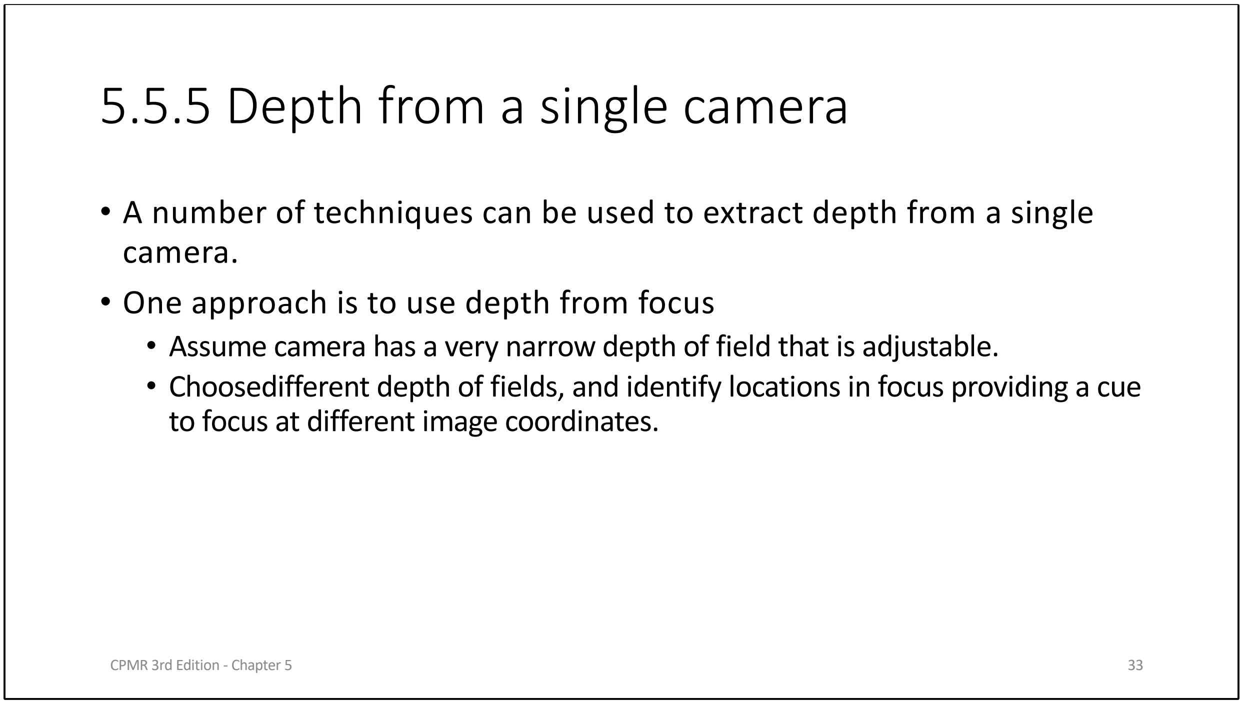
5.5.3 Model-based vision

- More generally, if we have a complete model of the object that is uniquely identifiable under projection we can *solve* the problem of object recovery and estimate position/state.



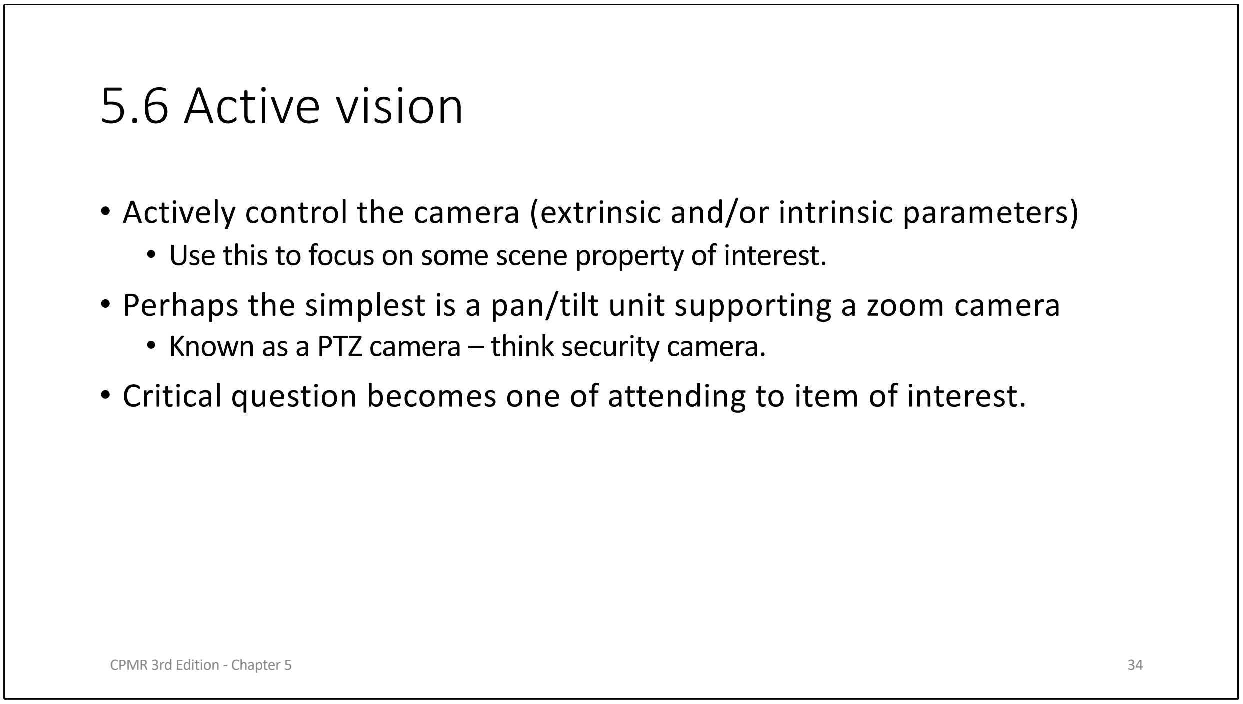
5.5.4 Egomotion

- Assuming a static world, then tracking of image features (either in 2d or 3d) can provide a cue as to the camera motion.
  - Typical approach in 3D involves aligning point clouds at different times and then extracting the rigid motion from the alignment.
  - Typical approach in 2D involves solving for a scale-independent estimate of motion. Often the assumption is made of a unit translation and then solving for the rotation.



5.5.5 Depth from a single camera

- A number of techniques can be used to extract depth from a single camera.
- One approach is to use depth from focus
  - Assume camera has a very narrow depth of field that is adjustable.
  - Choosedifferent depth of fields, and identify locations in focus providing a cue to focus at different image coordinates.



5.6 Active vision

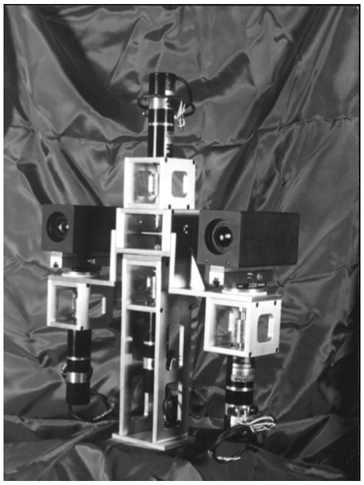
- Actively control the camera (extrinsic and/or intrinsic parameters)
  - Use this to focus on some scene property of interest.
- Perhaps the simplest is a pan/tilt unit supporting a zoom camera
  - Known as a PTZ camera – think security camera.
- Critical question becomes one of attending to item of interest.

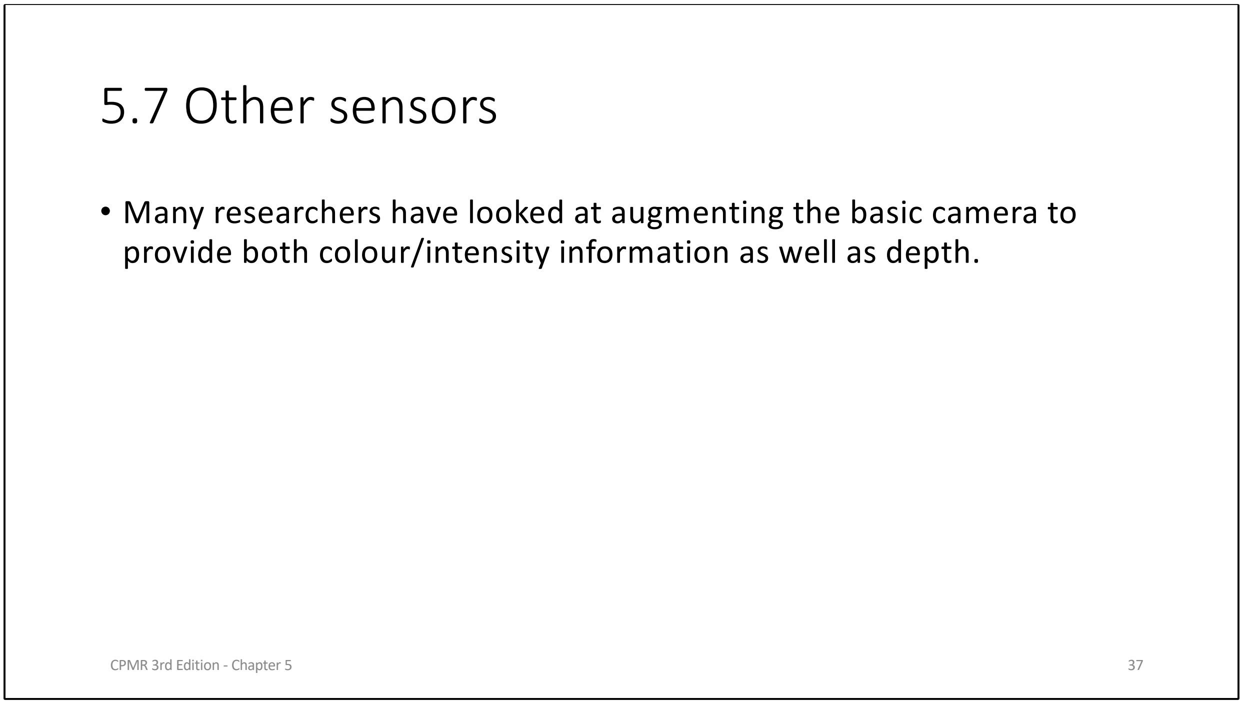
5.6.1 Foveated sensors

- Typical camera has uniform sampling.
  - Good for computers, but not good for attending to tasks and/or modelling sampling in the human retina.
- Very expensive to build but has been done.
- Can occur when using mirrors/other structures to manipulate optical pathways.

5.6.2 Stereo heads

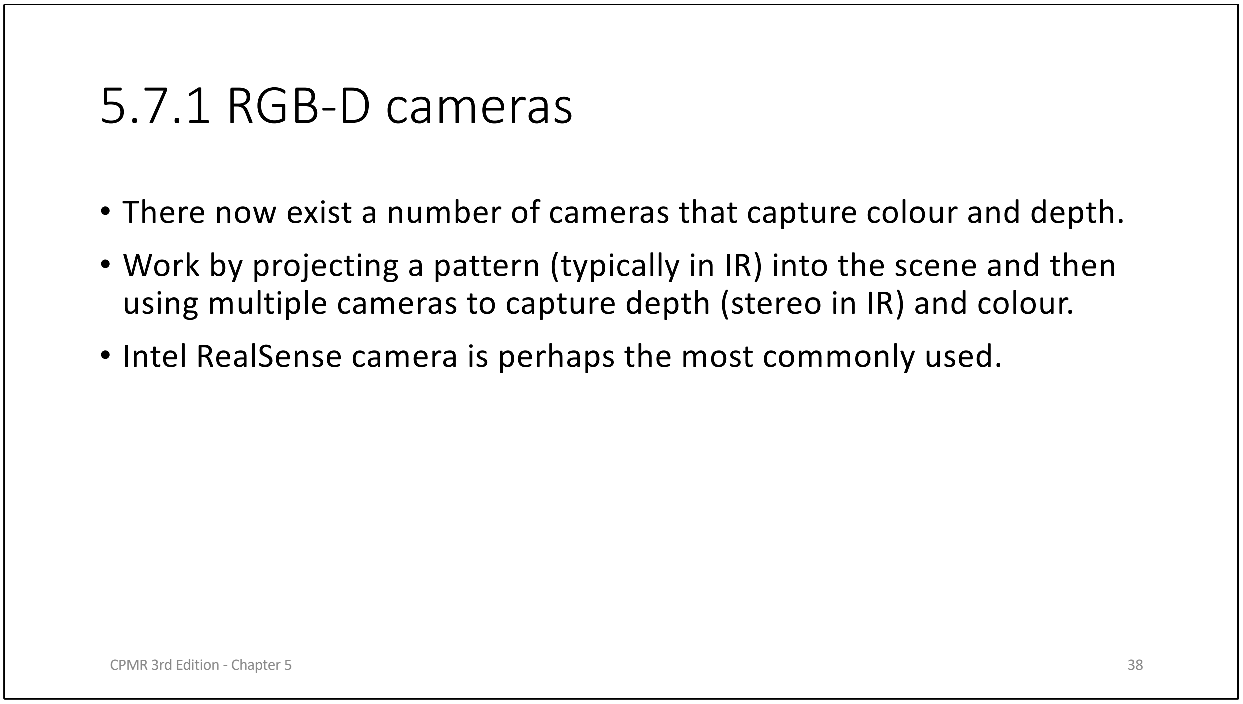
- Active sensor with stereo sensors.
- Seeks to model biological image capture
  - Exploit changing camera geometry to solve specific tasks \.





5.7 Other sensors

- Many researchers have looked at augmenting the basic camera to provide both colour/intensity information as well as depth.

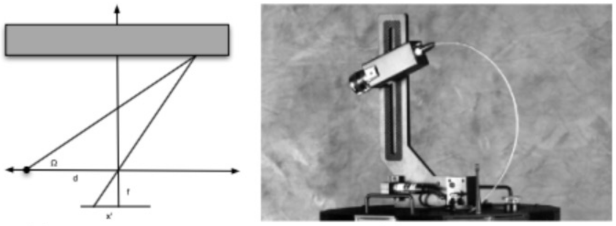


5.7.1 RGB-D cameras

- There now exist a number of cameras that capture colour and depth.
- Work by projecting a pattern (typically in IR) into the scene and then using multiple cameras to capture depth (stereo in IR) and colour.
- Intel RealSense camera is perhaps the most commonly used.

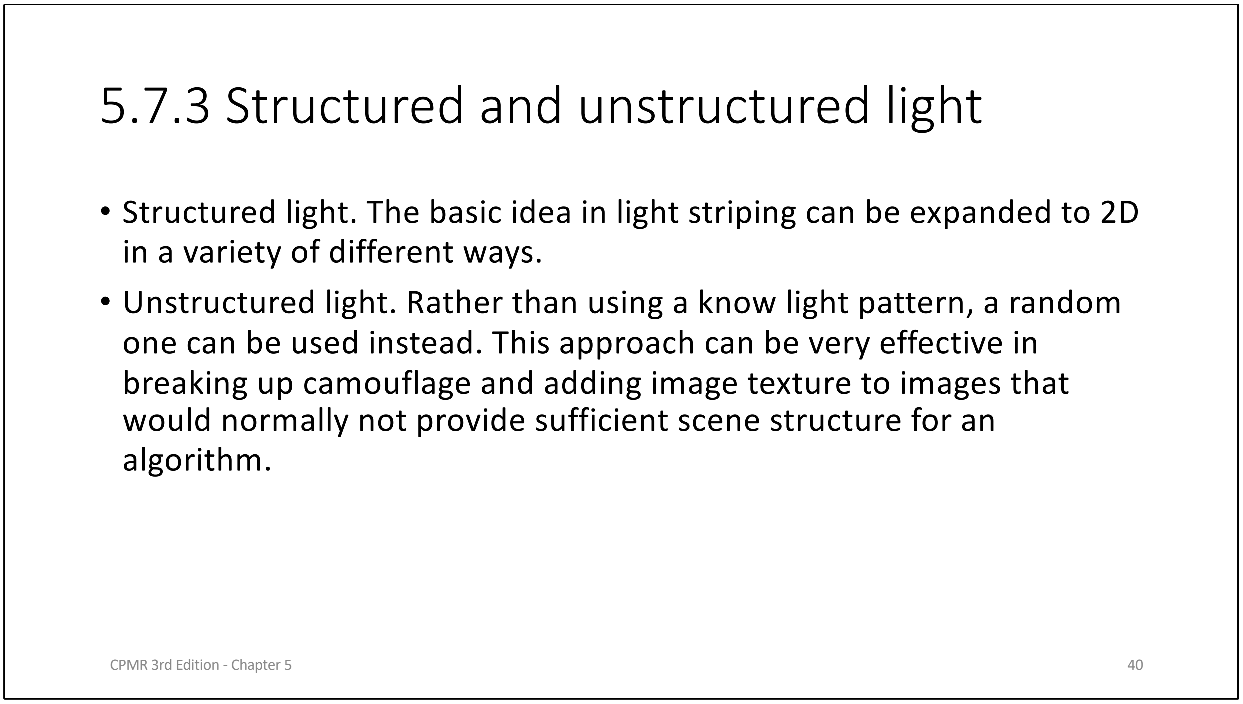
5.7.2 Light striping

- Using a laser or regular light provide a visible line in the scene.
- Known geometry between the line and the camera and proper camera calibration, enables a limited amount of 3D information to be recovered.



(a) Line stripe geometry

(b) Line striper

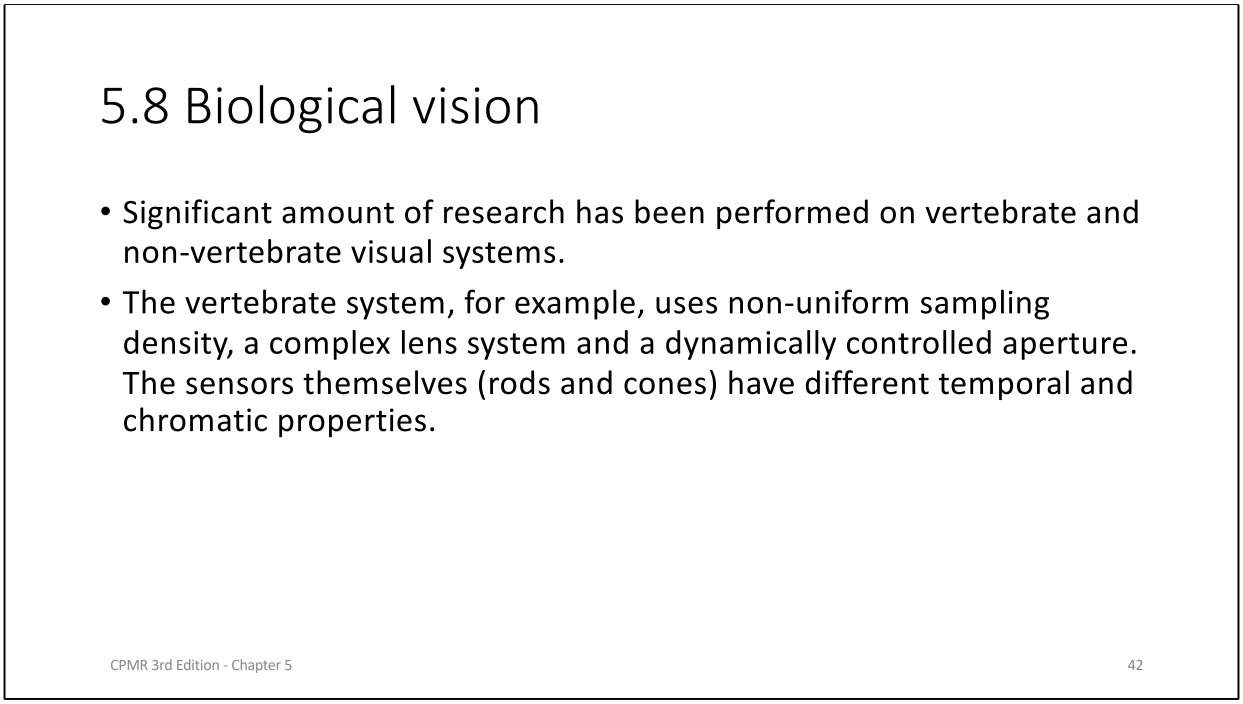


5.7.3 Structured and unstructured light

- Structured light. The basic idea in light striping can be expanded to 2D in a variety of different ways.
- Unstructured light. Rather than using a know light pattern, a random one can be used instead. This approach can be very effective in breaking up camouflage and adding image texture to images that would normally not provide sufficient scene structure for an algorithm.

5.7.4 Omnidirectional sensors

- To overcome the limited field of view of traditional cameras, extreme wide-angle lenses or conic-like mirrors can be used to provide a wider field of view.
- This approach often introduces a variation in image sampling that may not be appropriate for a given task.



5.8 Biological vision

- Significant amount of research has been performed on vertebrate and non-vertebrate visual systems.
- The vertebrate system, for example, uses non-uniform sampling density, a complex lens system and a dynamically controlled aperture. The sensors themselves (rods and cones) have different temporal and chromatic properties.