Stereo Vision

An introduction with Applications

Bae Gu-hwan
Hong Tae-kwan
Yun Mi-rin

Pusan National University

Outline

- Concept of Stereo Vison
- A brief description of Image
- Fundamentals of Computer Vision
- Implementation of Depth Camera
- Conclusion of this project
- Q&A

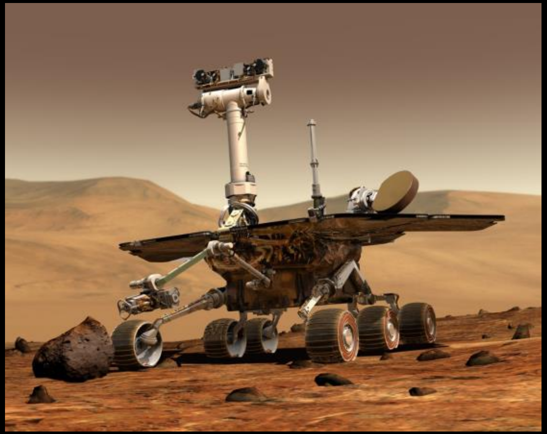
Introduction to Computer Vision

- The ability of computers to see
  - From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

- The extraction of a variety of information from images
  - Extract form, color, and distance of objects from the image.

Introduction to Computer Vision

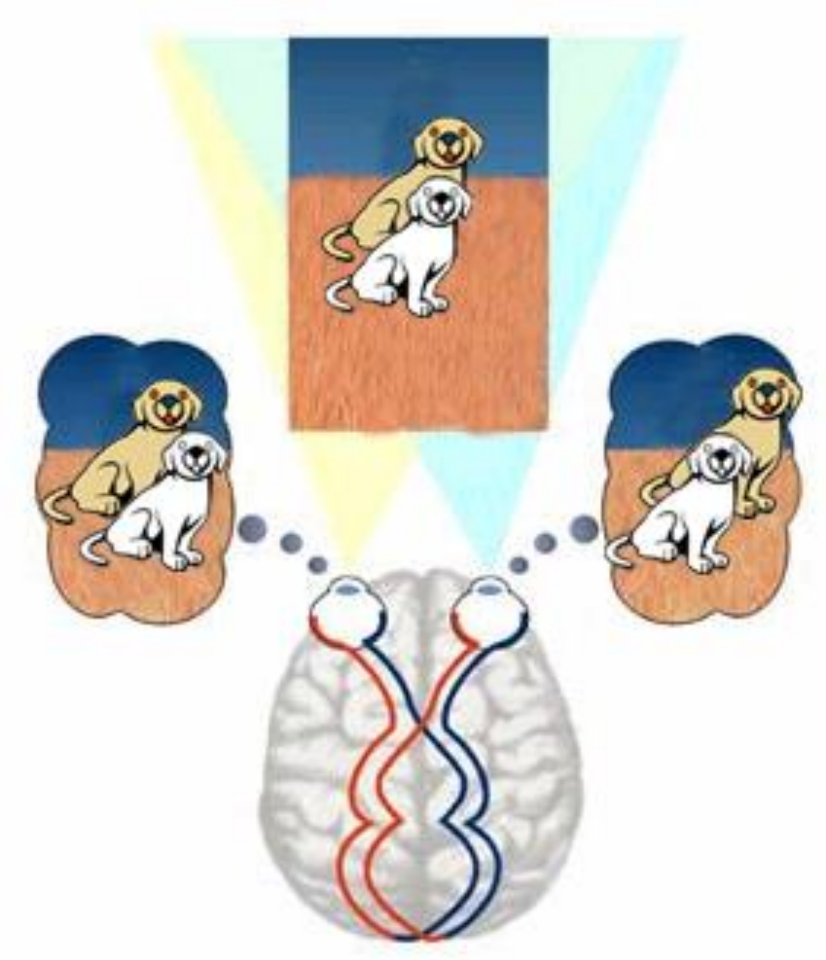
- Application
  - Information from the computer vision can be used in many areas.
  - Image processing, Recognition, Detecting, etc.
  - Photoshop, Robot vision, VR, AR, 3D Graphics of game, etc.
- Background
  - Linear Algebra, Signal Processing, Calculus, Numerical Analysis, etc.



What is Stereo Vision?

- Extraction of 3D information from two images

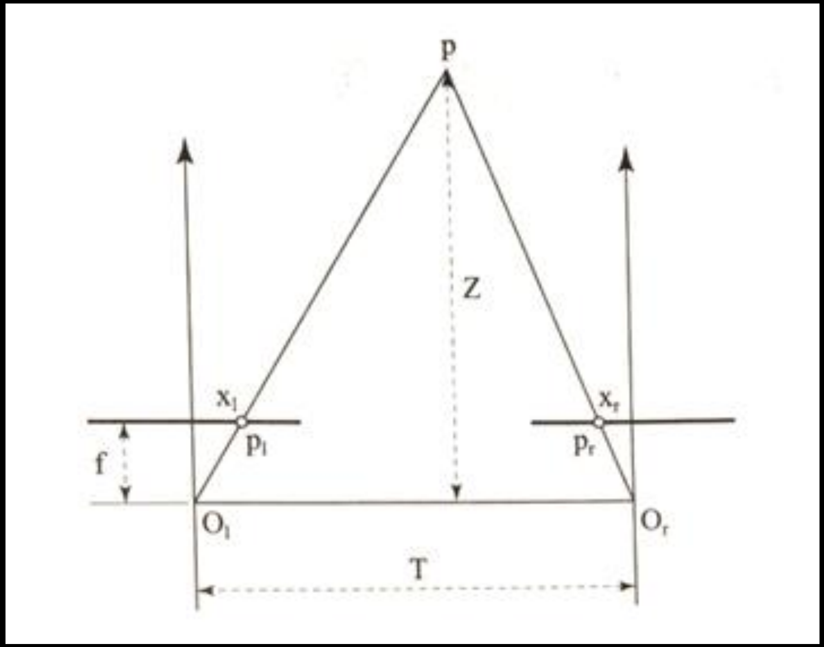
- It is same as how humans look at objects and determine their distance!



What is Stereo Vision?

- Extraction of 3D information from two images

  - If we have two images of same scene, we can get depth information from that in an intuitive way.
  - Next is an image and some simple mathematical formulas which proves that intuition.



$$f : x_l = Z : X \,,\, f : (-x_r) = Z : (T - X)$$

$$Z = \frac{fT}{x_l - x_r} = \frac{fT}{d}$$

Example

- A stereo vision system estimates the disparity of a point as d = 10 pixels

  - what is the depth(Z) of the point, if f= 500 pixels and b= 10 cm?

$$Z = f\frac{b}{d} = 500pix\frac{10cm}{10pix} = 500cm$$

  - what is the uncertainty(standard deviation) of depth, if the standard deviation a feature in each image = 1 pixel?

$$\delta_z = Z\frac{\delta_d}{d} \qquad \delta_d^{\,2} = \delta_{x_L}^{\,2} + \delta_{x_R}^{\,2} = 2 \qquad \therefore \delta_d = \sqrt{2}$$

$$= 500cm\frac{\sqrt{2}pix}{10pix} \cong 70cm$$

What is Image?

- Digital images are stored as matrix of numbers

- Sampling

ex) VGA : 640x480

UHD : 3840x2160

- Quantization

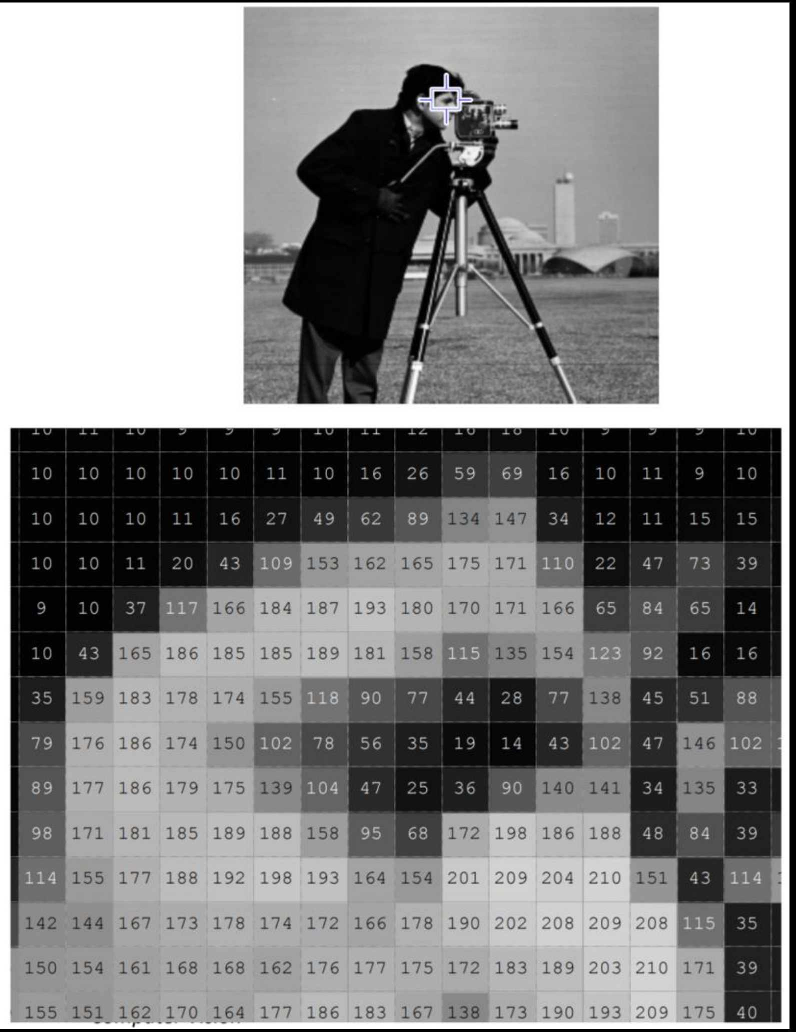
ex) 8bit (gray scale)

24bit (true RGB)

- Image vs Text

ex) UHD(true RGB) Image : 3840x2160x24bit = 23.7MB

Book : 250kB ~ 300kB



What is Image?

- Digital images are stored as matrix of numbers

- Sampling

ex) VGA : 640x480

UHD : 3840x2160

- Quantization

ex) 8bit (gray scale)

24bit (true RGB)

- Image vs Text

ex) UHD(true RGB) Image : 3840x2160x24bit = 23.7MB

Book : 250kB ~ 300kB

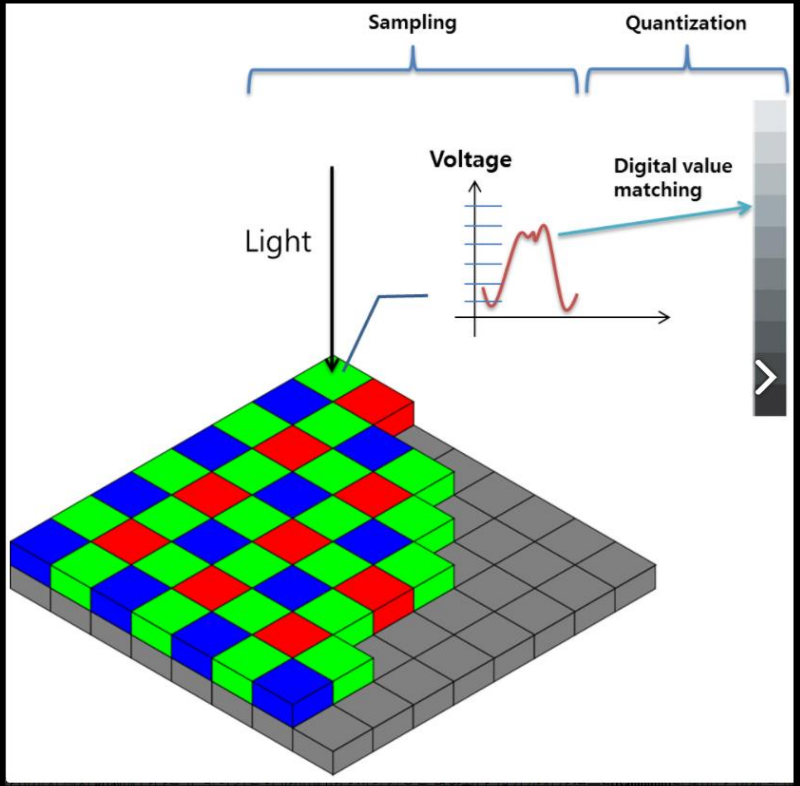
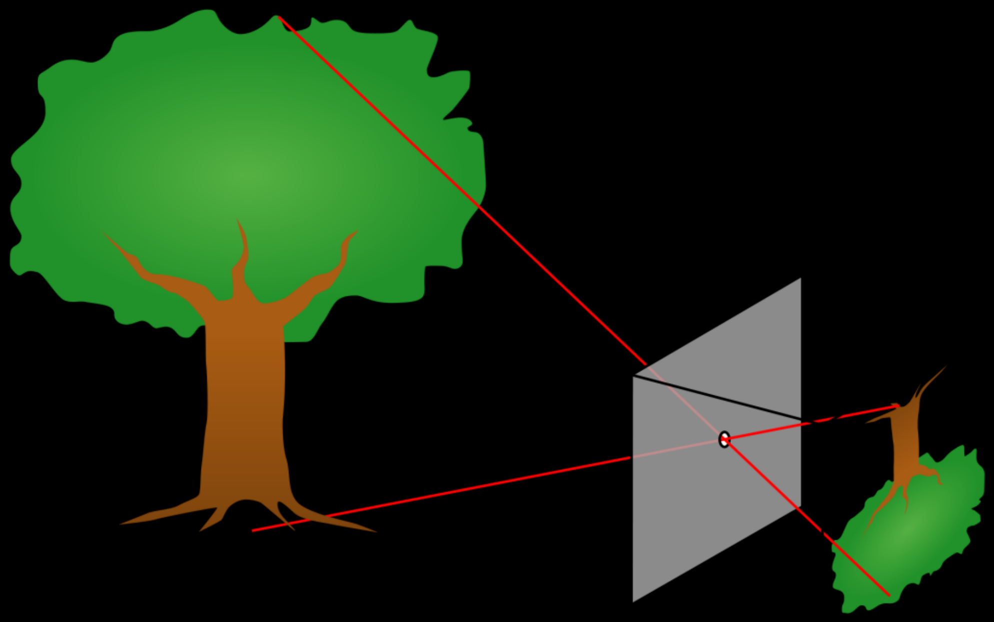
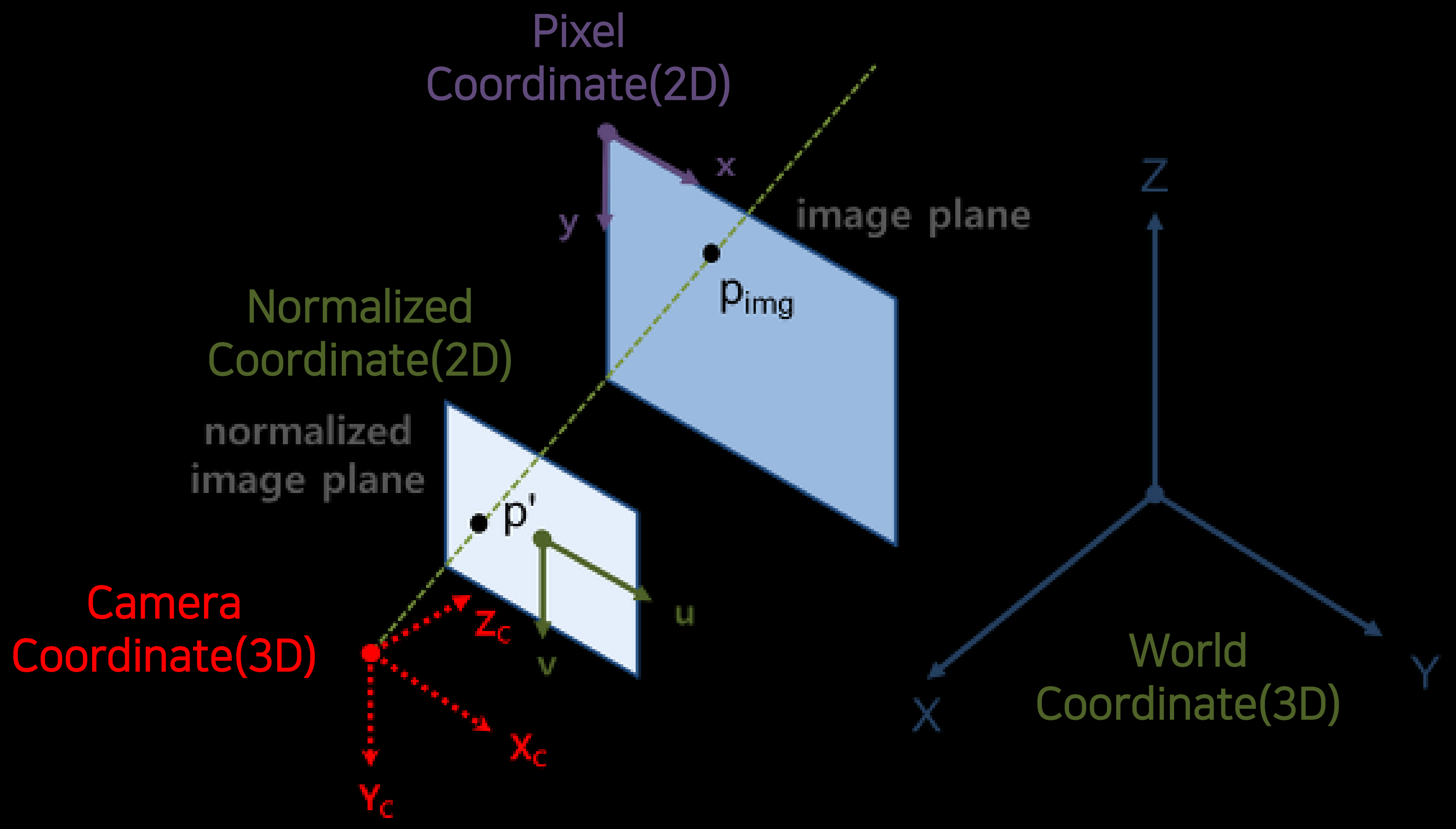


Image Formation

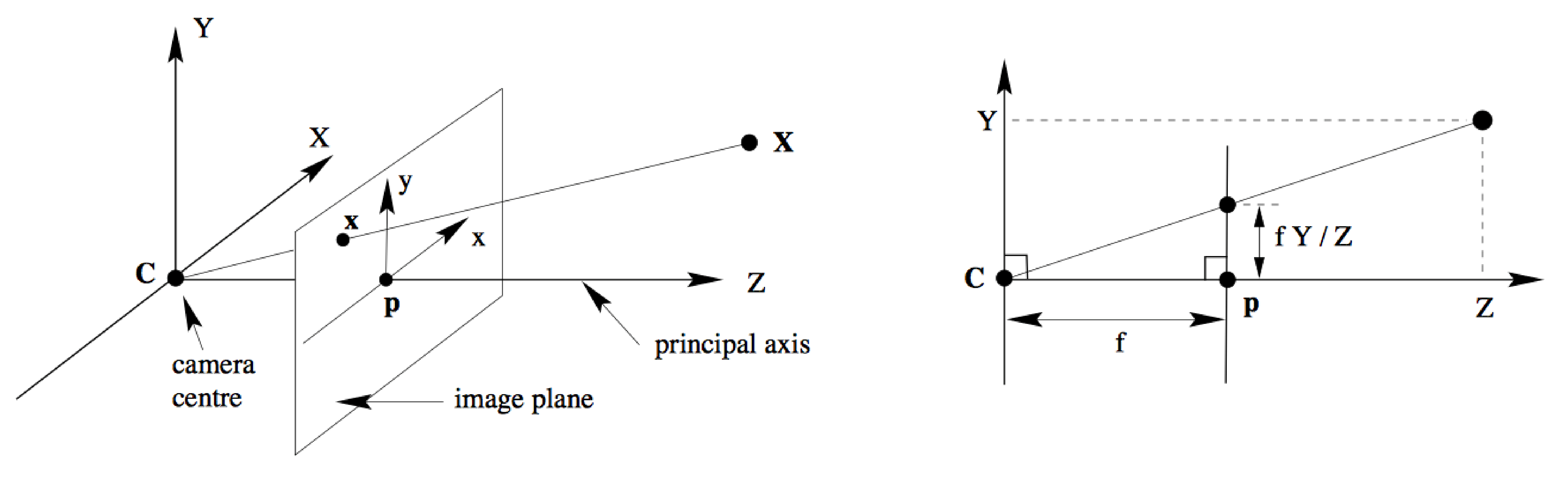


- A good lens can be modeled by a pinhole camera; each ray from the scene passes undeflected to the image plane.
- Simple equations describe projection of a scene point onto the image plane("perspective projection").

4-Coordinate System



A Simple Pinhole Camera Model



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

We can see the relation between a point in 3D space and to a point on the focal plane.

Image Geometry

- This **3x4 matrix** P is known as the Camera Projection Matrix. To adjust for the reference frame in the image plane – origin at corner v/s origin at center, we add a translation component to the left 3x3 matrix of P.
- This 3x3 matrix K is called the Camera Calibration Matrix.

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

$$x = PX$$

In the Image plane

In world coordinate

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

Consider
Nomalization & distortion factor

Image Geometry

- In most cases, the camera coordinates will be with respect to a different coordinate system where it is not at the origin in the world coordinate system.
- Hence, any coordinate can be shifted from the world coordinate space to the camera coordinate space (where the camera is the origin) with the equation.

$$X_{camera} = R(X_{world} - C)$$

Image Geometry

$$X_{camera} = R(X_{world} - C)$$

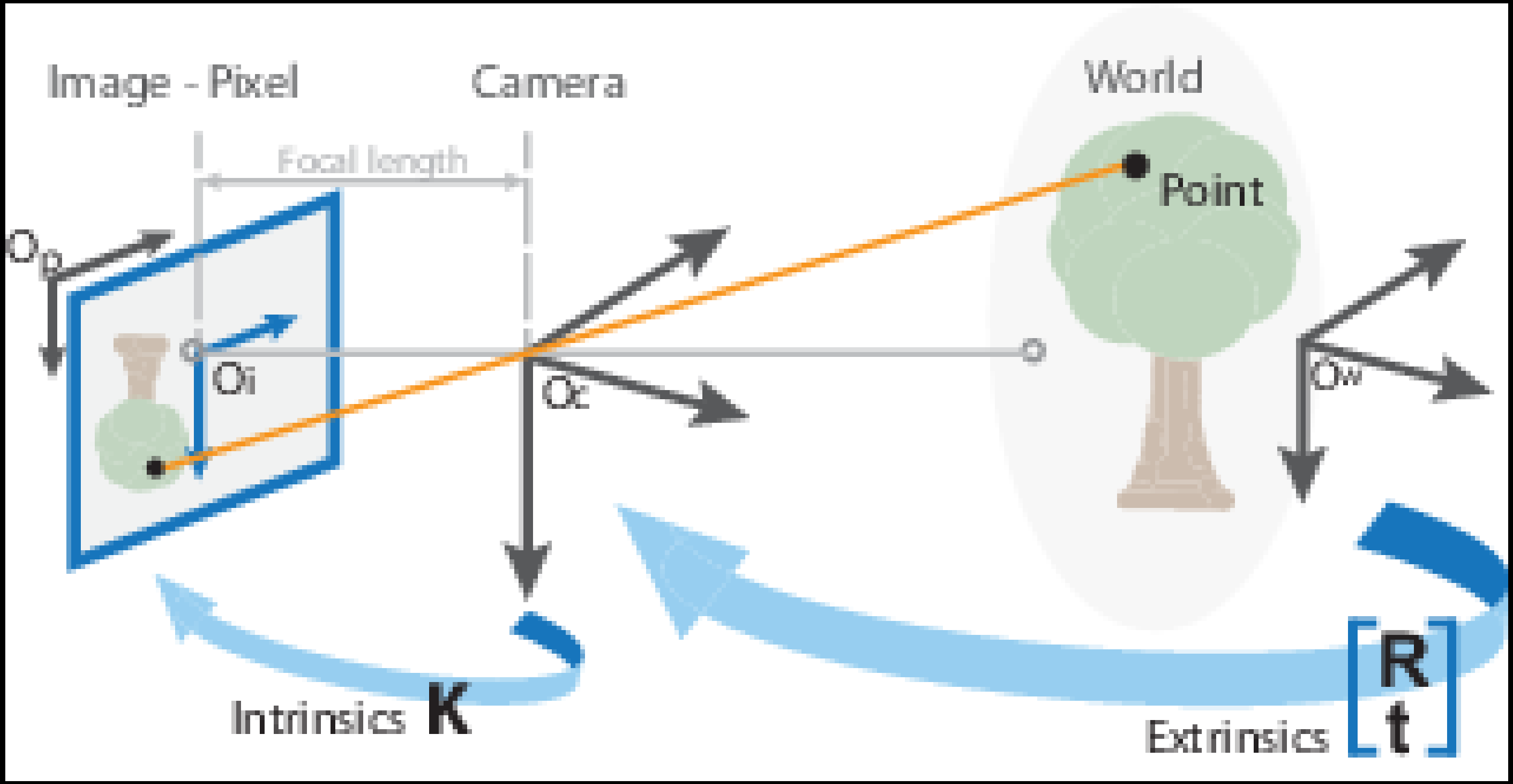
- where R is the rotation matrix and $-C$ represents a translation with respect to the camera center to make it the origin.
- Combining the *camera calibration matrix* and the coordinate frame shifting above, we can draw a mapping between the point in world coordinate system X and the point x in the image plane, represented by

$$x = KR[I| - C]X$$

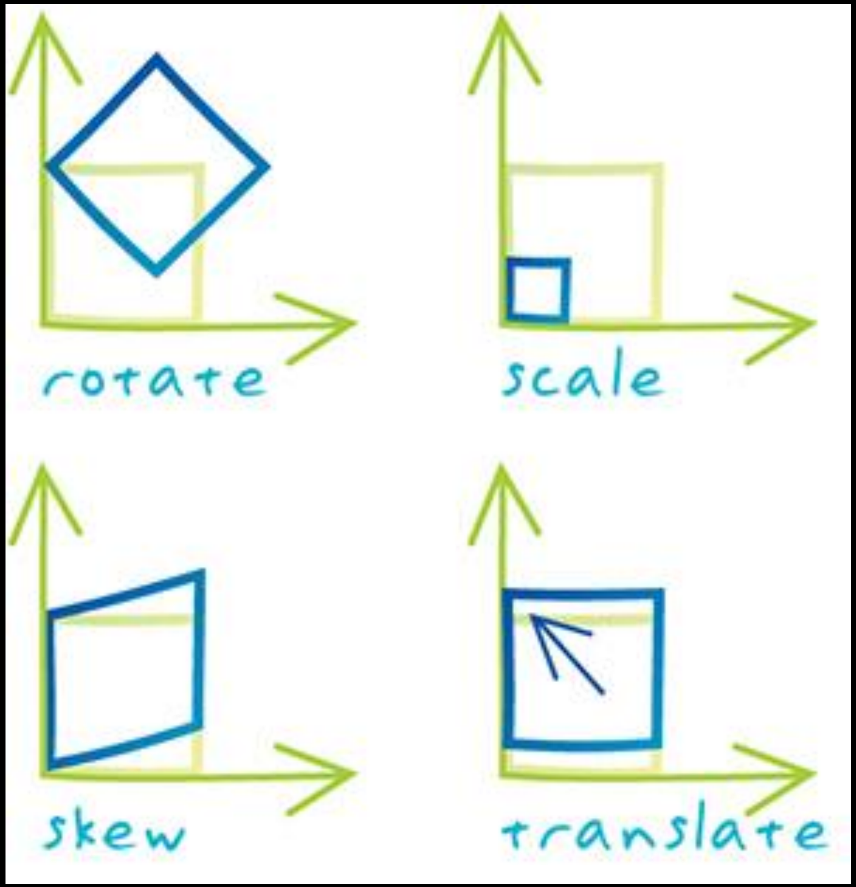
$$P = KR[I| - C]$$

Parameters in K are called the camera intrinsics (they never change) and the remaining parameters R and C are called the camera extrinsics.

Image Geometry

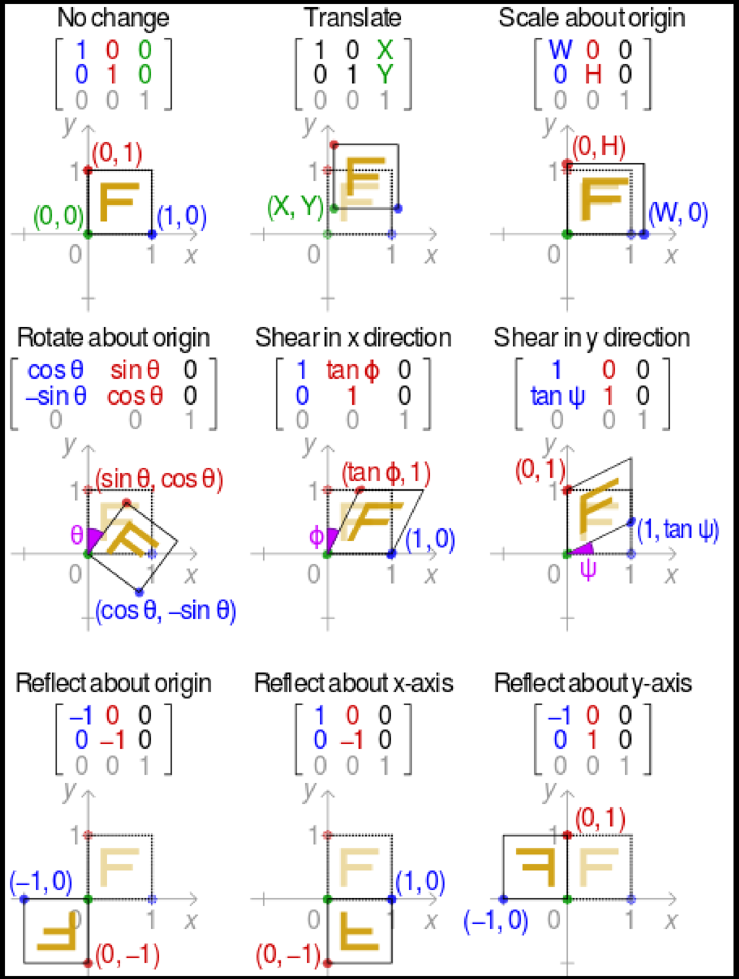


Geometric Transformations of Images

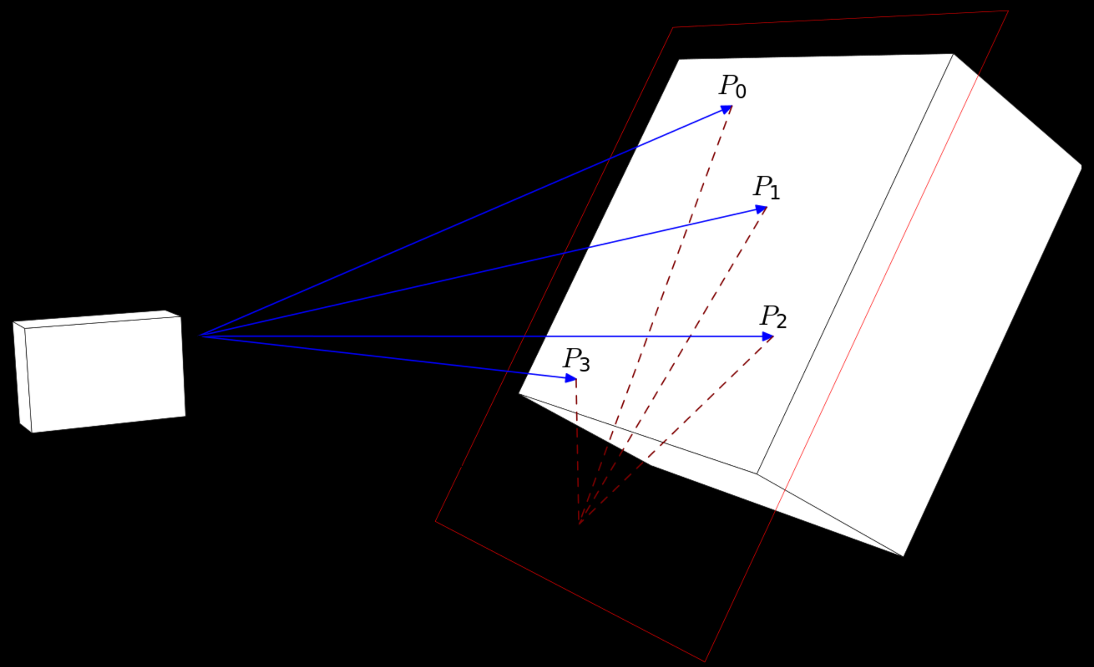


- Also called spatial transformations, 2D transformations
- Include: translation, rotation, scaling, nonlinear warping, or perspective angle adjustment.
- It is actually a rearrangement of pixels on the image plane.
- The coordinates of input image are transformed into coordinates of output image using a transformation function.

Transformations Matrix



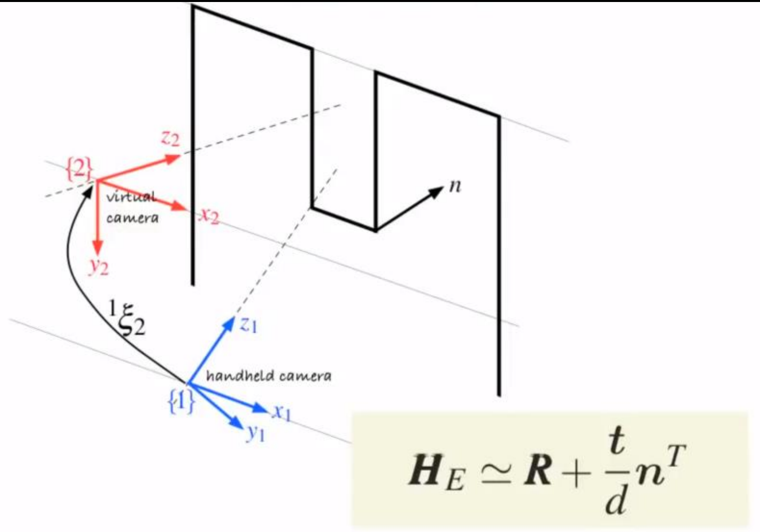
Homography



- For a Planer surface object, you can model a random perspective projection transformation from 3D space to 2D images as a homography between two images.

Perspective rectification

$$^{2}p = H\ ^{1}p$$



By using Homography matrix H, We can get an image that it seems from virtual camera view.

Demo #1

1) Image Alignment
2) Virtual Ads on buildings

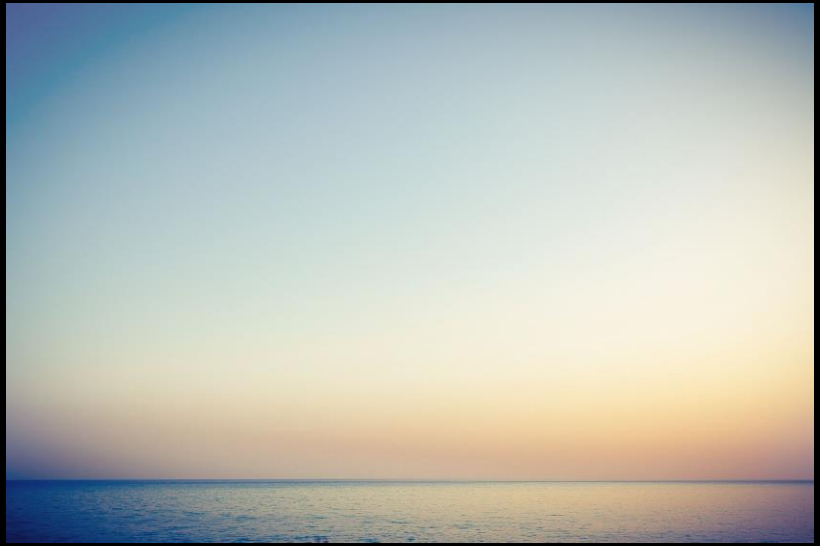
Features in Image

- Correlation-based approaches
  - Matches image patches using correlation.
  - Assumes only a translational difference between the two local patches. (no rotation, or differences in appearance due to perspective)
  - A good assumption if patch covers a single surface, and surface is far away compared to baseline between cameras.
  - Works well for scenes with lots of texture.

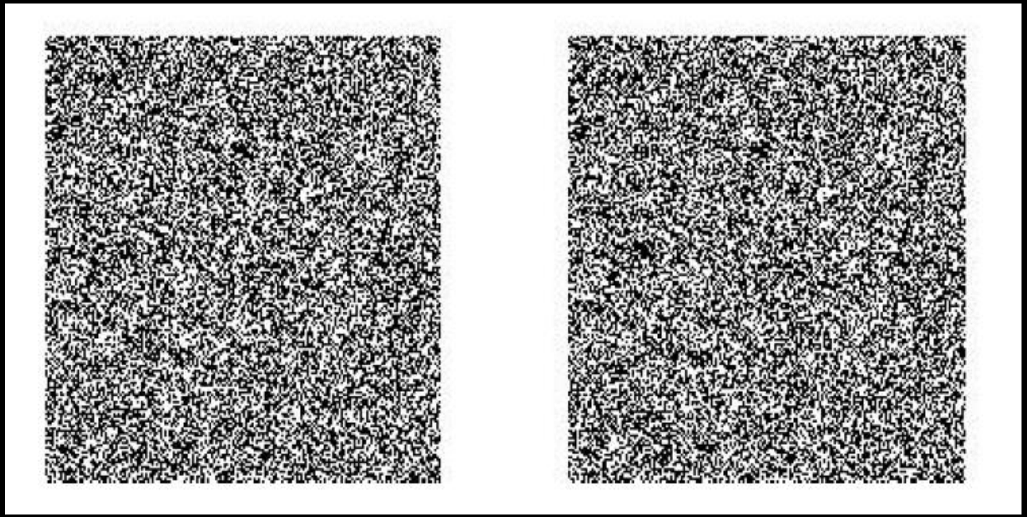
- Feature-based approaches
  - Matches edges, line, or corners
  - Gives a sparse reconstruction

Correspondence Problem

- Worst case scenarios



No features

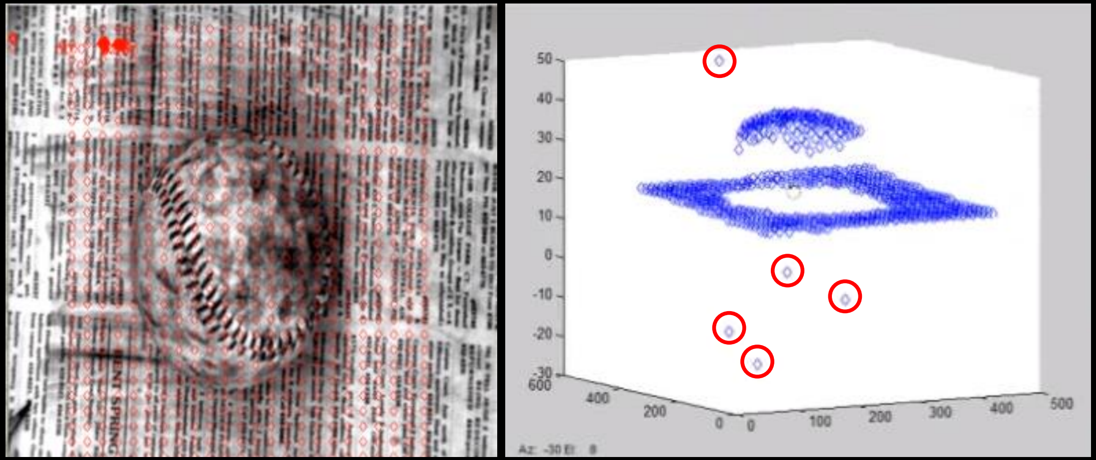


Ambiguous matches

Correspondence Problem

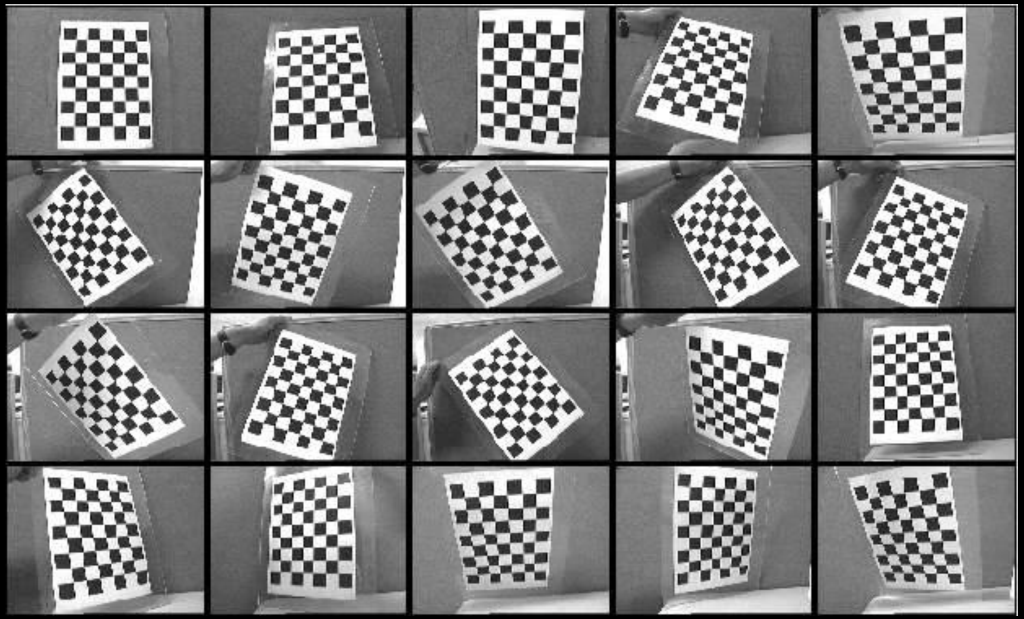
- Mismatching makes errors when calculating Depth-map.

Scattering reflection of light.



Error

Camera Calibration

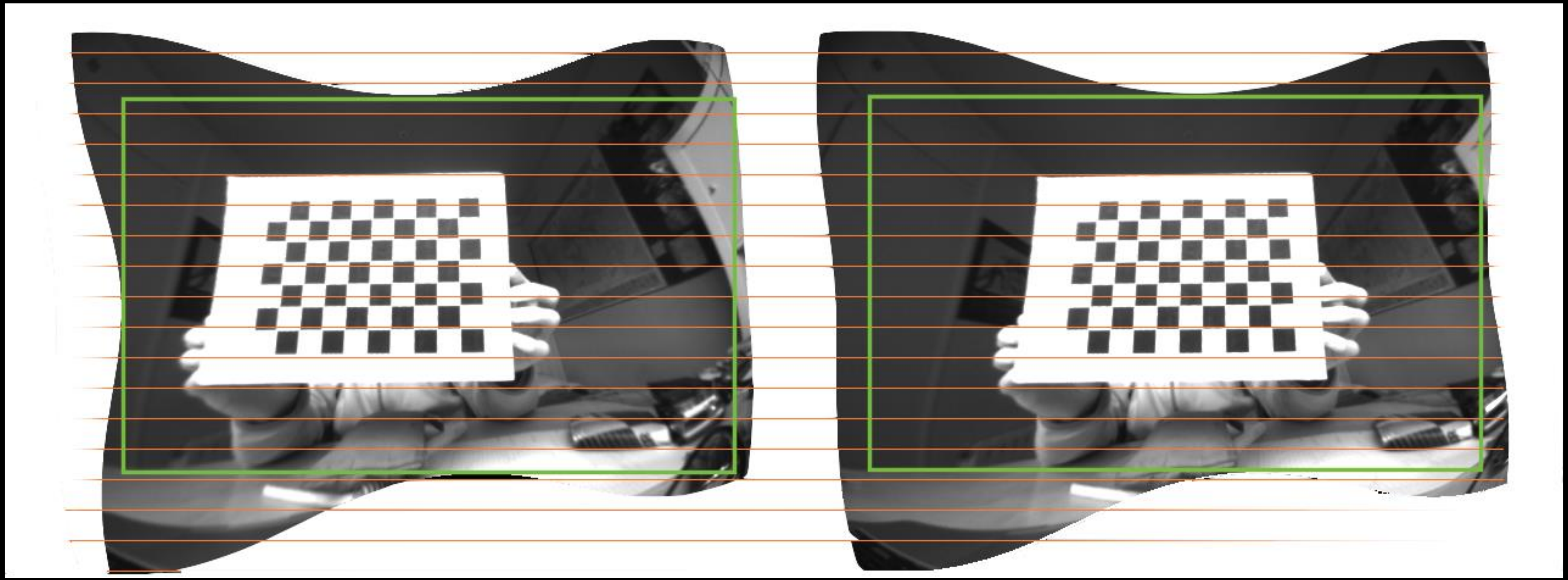


- Geometric **camera calibration**, also referred to as camera resectioning, estimates the parameters of a lens and image sensor of an image or video camera.
- You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene.
- These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3D scene reconstruction.

Camera Calibration

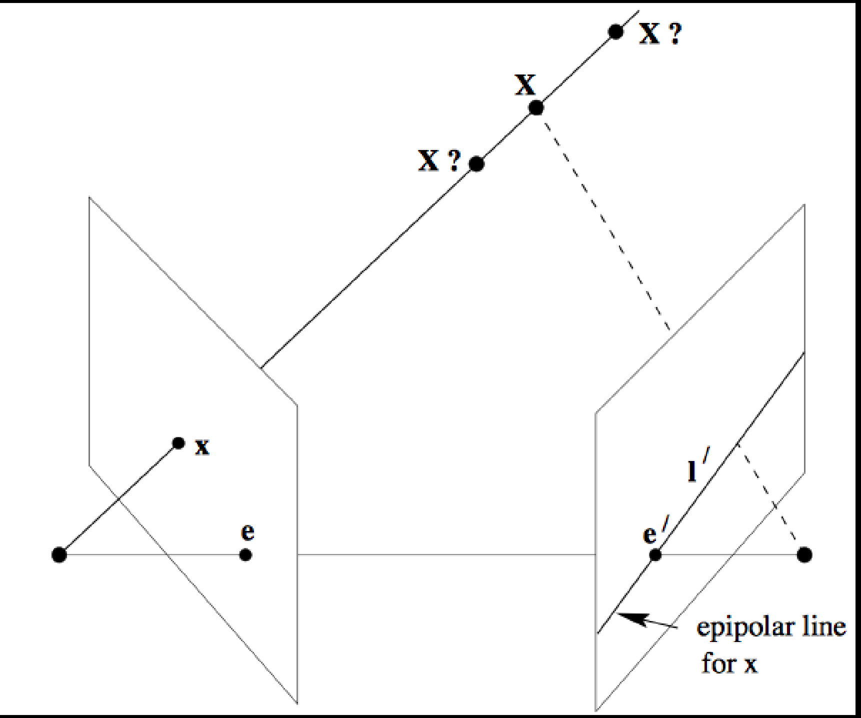
- You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene.
- These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3D scene reconstruction.

Camera Calibration



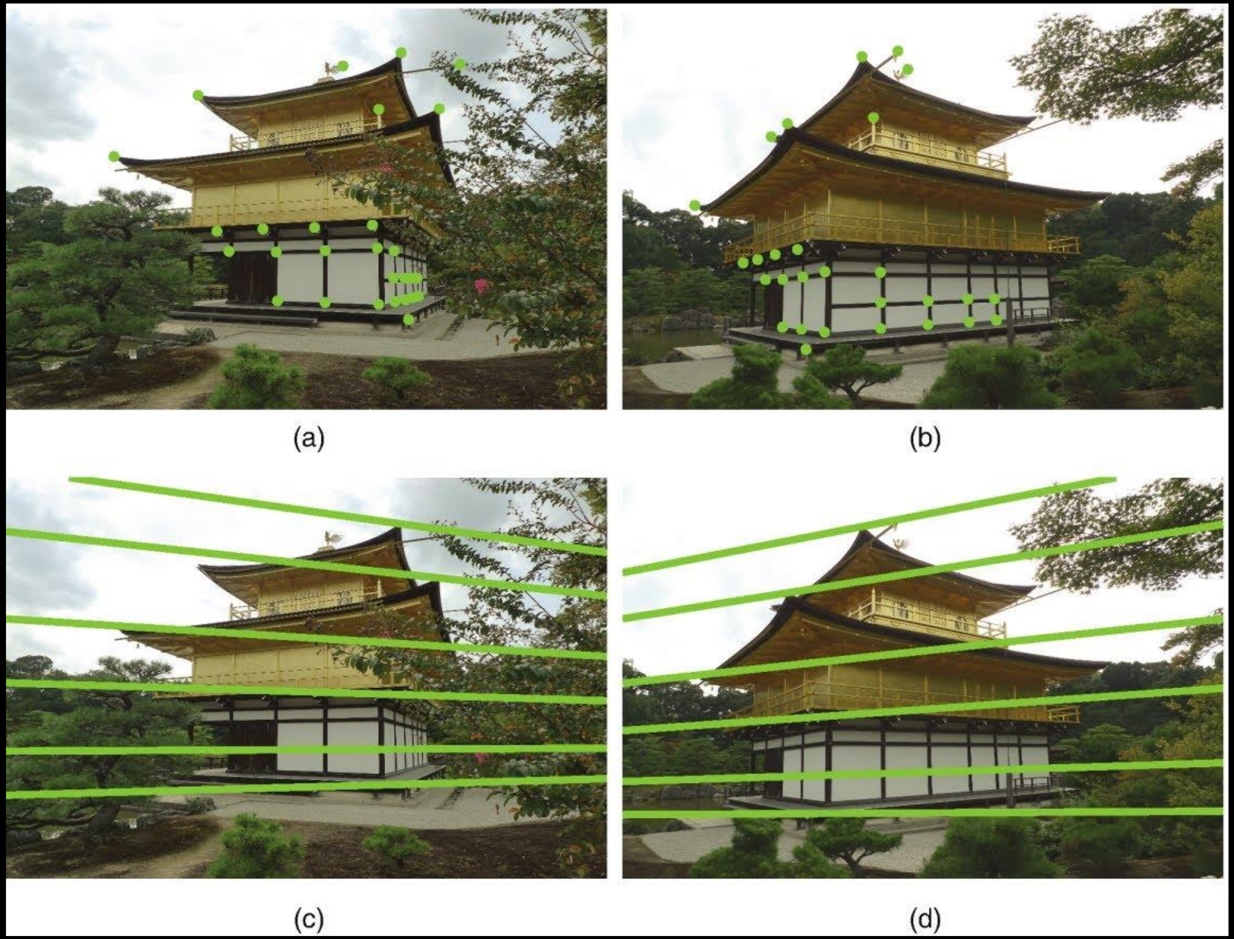
- Corrected parameters reduce errors to accurately build Depth-map.
- These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, autonomous driving system and 3D scene reconstruction.

Epipolar Geometry



When two cameras view a 3D scene from two distinct positions, there are a number of geometric relations between the 3D points and their projections onto the 2D images that lead to constraints between the image points.

Epipolar Geometry



Green points are matching point from features in image and Green lines are epipole lines.

If the epipole lines in the two images are parallel to each other, the most accurate Depth-map can be calculated.

Demo#2
– 3D Reconstruction with Stereo Image

Calibrated Left Image

Calibrated Right Image

What is disparity map?

Conclusion of this project

Use open-source

- Ubuntu
  - Free and well made GUI Linux package
  - Good Portability
  - License : GNU Public license
  - Platform : Linux
  - Supports open sources much better than Windows (OpenGL, ffmpeg, Tensorflow etc.)

Conclusion of this project

Use open-source

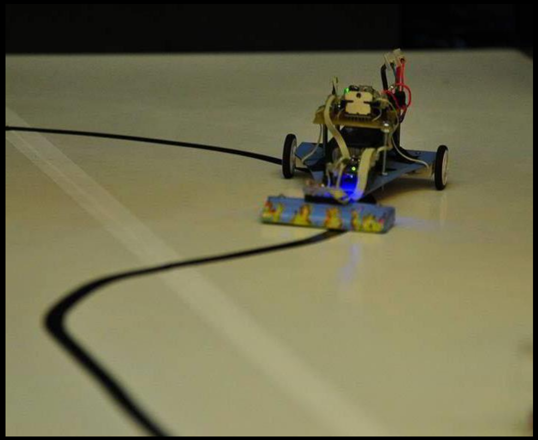
- OpenCV
  - Free Software package
  - Image Processing & Computer Vision Libraries
  - License : Modified BSD lisence
  - Platform : Linux/Mac/Windows/etc.
  - Language : C++/Python/Java/etc.

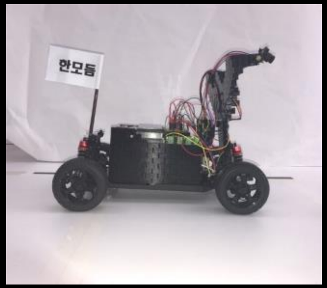
My Project Repository : github.com/ghbae92

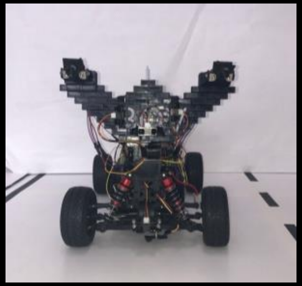
Team Repository : github.com/teamHMD

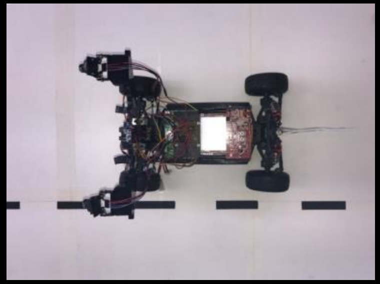
Conclusion of this project

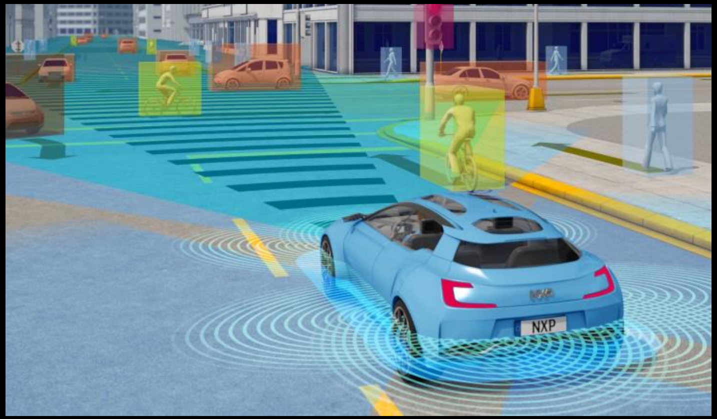
What is next?











Q&A

End of this presentation

Thanks for your attention