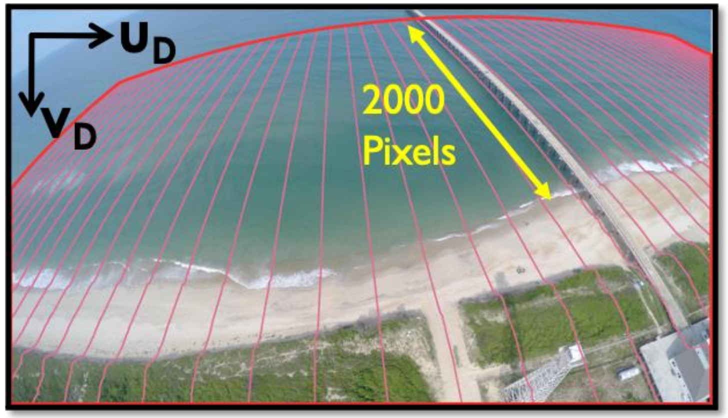
PHOTOGRAMMETRY BASICS

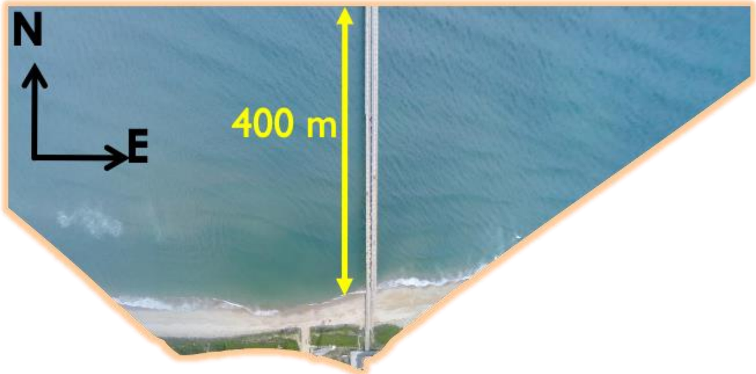
2023

With a few pieces of information & some math, we can exploit how a camera lens works to make quantitative measurements of the coast within the camera field of view

PHOTOGRAMMETRY - DEFINITION

- Word Origins:
  - Photos = 'light'
  - Gramma = 'that which is drawn or written'
  - Metron = 'to measure'
- Definition in Manual of Photogrammetry, 1st ed., 1944, American Society for Photogrammetry:
  - *Photogrammetry is the science or art of obtaining reliable measurement by means of photographs*





MAPPING AND PROVIDING LENGTH SCALES TO IMAGES TO DO MEASUREMENT AND SCIENCE

COASTAL IMAGING

MANY NEARSHORE PROCESSES HAVE OPTICAL SIGNATURES

WAVE FACE
DARK COLOR

SURFACE CURRENTS
MOVING FOAM/PLUME

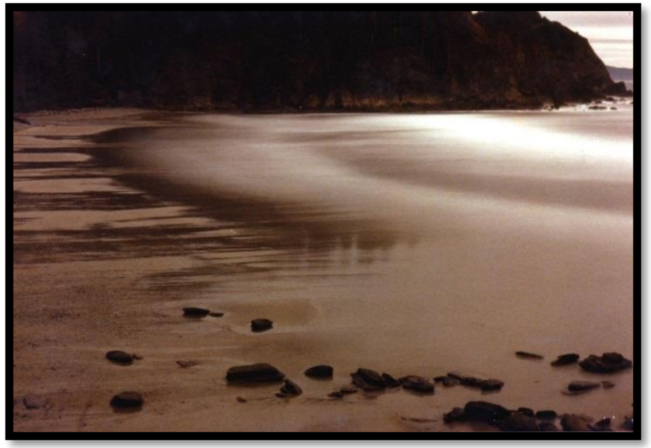
TIDAL LINE
DARK SEDIMENT

RUN-UP
WHITE EDGE

WAVE BREAKING
WHITE FOAM

BUT HOW CAN WE QUANTIFY WHAT WE SEE?

HISTORY OF GROUND-BASED COASTAL IMAGING



- 1910s: Aerial photography & traditional photogrammetry
- 1980s: Time-averaged photography
- 1990s-2000s: Development of a global Argus network w/ automated data products

- 2010 – Present: easy access to action cameras, streaming web-cameras, and drone/UAV-based cameras; many different collection & processing approaches

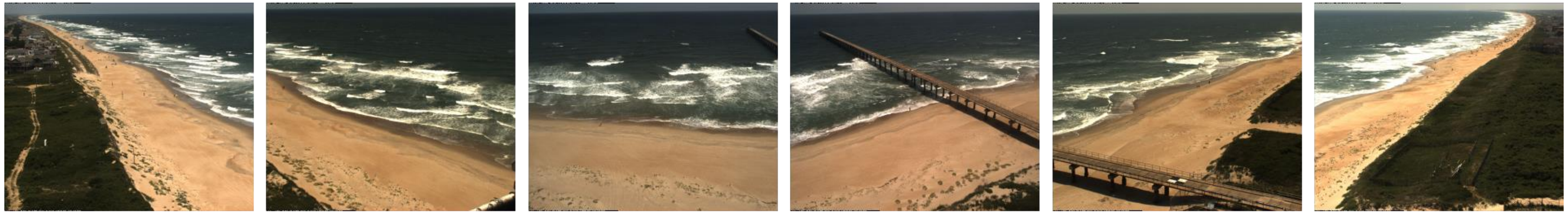
PRESENTATION OUTLINE

- Basic Principles of Photogrammetry & Camera Calibration (Camera Basics, Intrinsics, Extrinsics, Camera Models)
  - The information we exploit
- Detailed Pin-Hole Camera Model
  - Understanding qualitatively how to relate what you see in an image to the world
- Math Principles
  - Geometry, Linear Algebra, Matrices
- Deriving the Camera Equation & Projection Matrix
  - Understanding quantitatively how to relate what you see in an image to the world

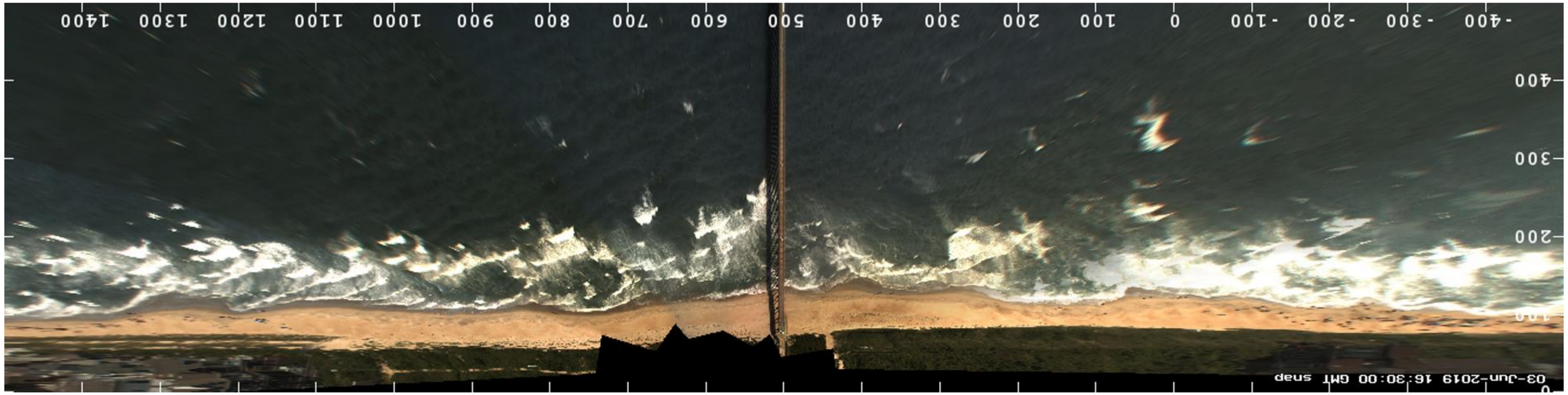
PHOTOGRAMMETRY – BASIC PRINCIPLES

- Cameras are a passive sensor (not active like radar, lidar, acoustic gauges, etc., all of which emit signals and listen for the return)
- Cameras provide a "non-contact" measurement (generally do not interrupt the processes they are taking a picture of)
- Geometric principles (collinearity, similar triangles, homogeneous coordinates) and linear algebra (rigid transformations, projections) allow us to exploit how a camera lens works to understand relationships between the locations of an object in the world and in an image

"Oblique Image"
(pixel space)



"Projected Image"
(geo-rectified maps; can make spatial measurements)

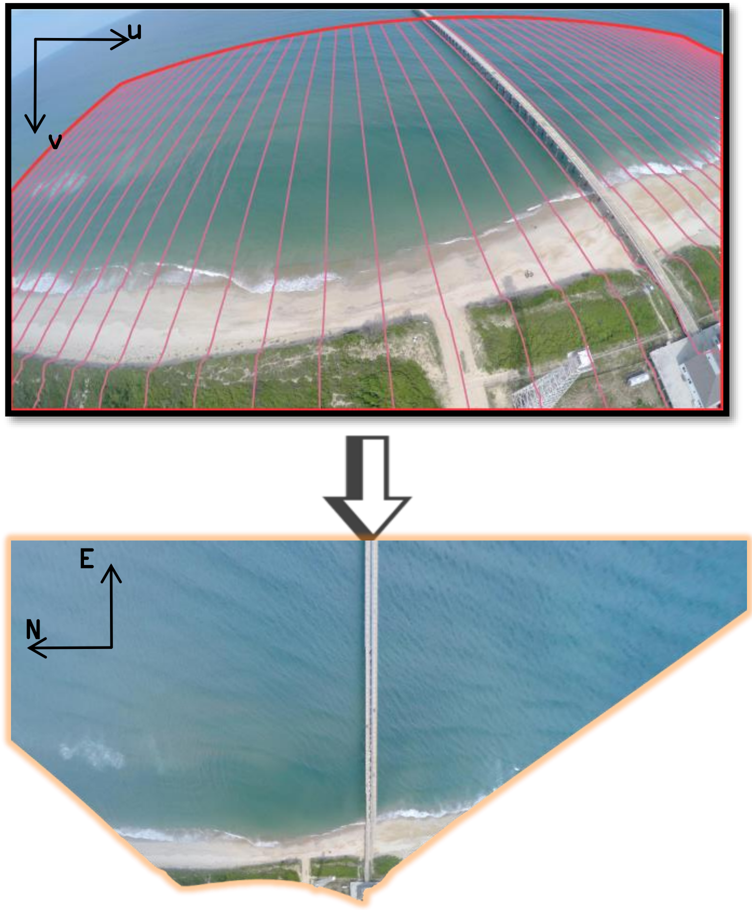


CAMERA BASICS

- **Lens** → focal length determines field of view (FOV)
- **Sensor Size** → controls image dimension, noise level, and the number of pixels in your image (image resolution)
- **Frame Rate** → controls the frequency of your pictures
- **Shutter Type** → rolling shutters (most webcams) will look like "jello" if shaking (high-wind)
- **Exposure** → changes to the shutter speed, aperture, and ISO based on light fluctuations effect the brightness of features in your images (fixed vs. auto-adjusting); if using auto, it may change your frame rate in dimly lit areas

LEARN MORE: How Cameras Work Lesson

PHOTOGRAMMETRY BASICS



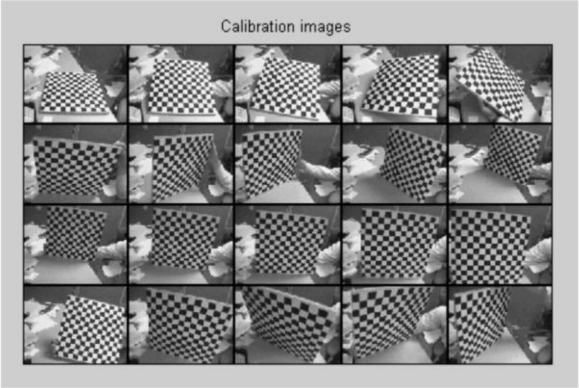
TO MAKE SPATIAL MEASUREMENTS, WE MUST CORRECT FOR:

- *CAMERA LENS PROPERTIES (INTRINSIC CAMERA CALIBRATION)*
- *WHERE THE CAMERA IS LOCATED (Position) AND LOOKING (Orientation) (EXTRINSIC CAMERA CALIBRATION)*

If you forget all of the presentation… remember this!

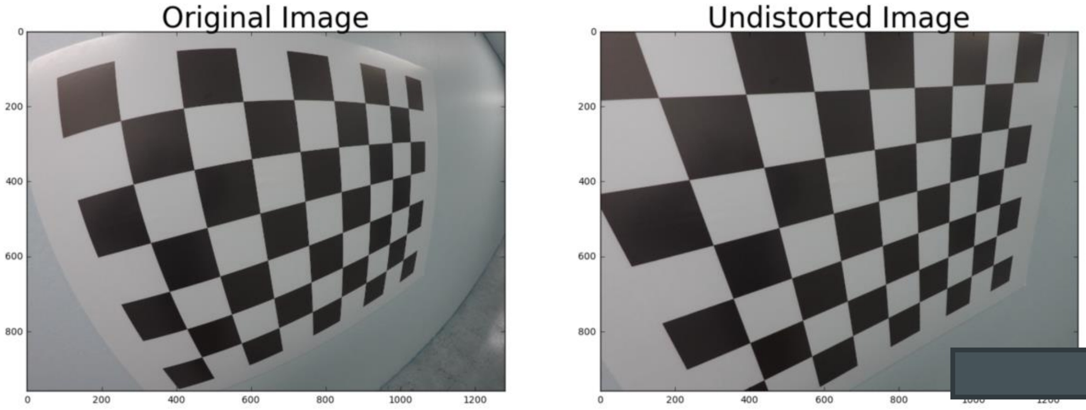
INTRINSIC CALIBRATIONS (ALSO CALLED INTERIOR PARAMETERS)

- Intrinsic calibration finds calibration coefficients that are functions only of the camera and lens, so can be found in the lab prior to field installation.
- Different distortion models exist (find different number of coefficients)



Taking pictures of objects of known size (like a checkerboard) allows us to solve for focal length, principle points, pixel skewness and distortion coefficients

Distorted → Undistorted



Take Home Points:

- Simple calibration to determine focal length, principle point, skewness, and distortion coefficients (must always do even if low distortion lens!)
- The wider your field-of-view, the more distorted your imagery and the more spatial variability in your ground sample distance (size of a pixel in the real world)

LEARN MORE:
Lens Distortion Lesson

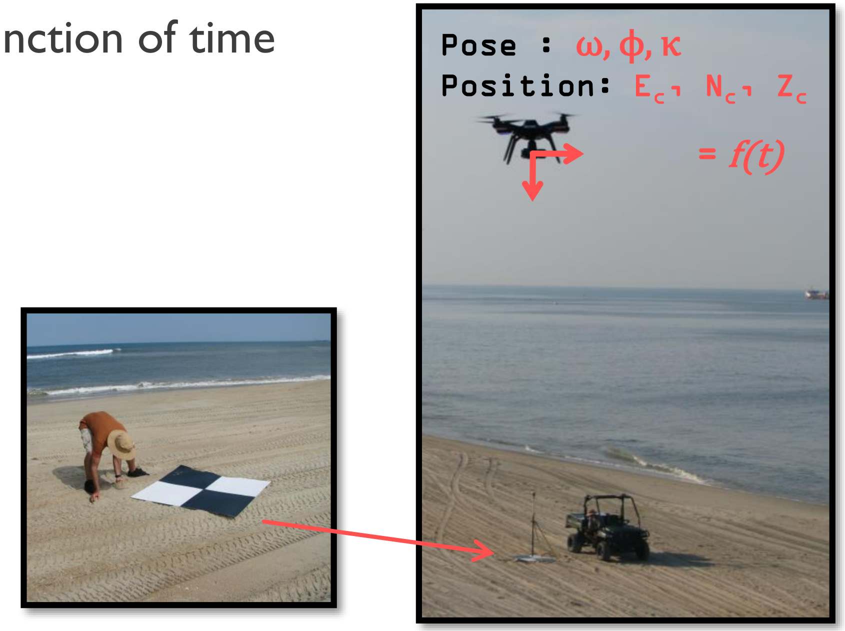
CiRN

EXTRINSIC CALIBRATION (ALSO CALLED EXTERIOR PARAMETERS)

- Extrinsic calibration finds the six coefficients that describe the location and viewing angles of a camera once installed (fixed station) or as a function of time (UAS)

  Position: X, Y, Z of camera
  Orientation: Heading, Roll, Pitch

  - Sometimes these can be difficult to measure;
  - instead we use Ground Control Points (GCP)s, and survey their location to solve for position and orientation



LEARN MORE: Determining Camera Geometries Lesson

Take Home Points:

- Knowing the position & orientation of your camera is critical for making maps with your images
- Need to re-do every-time camera moves OR use a correction algorithm to match features between images to remove movement

EXTRINSIC CALIBRATION (ALSO CALLED EXTERIOR PARAMETERS)

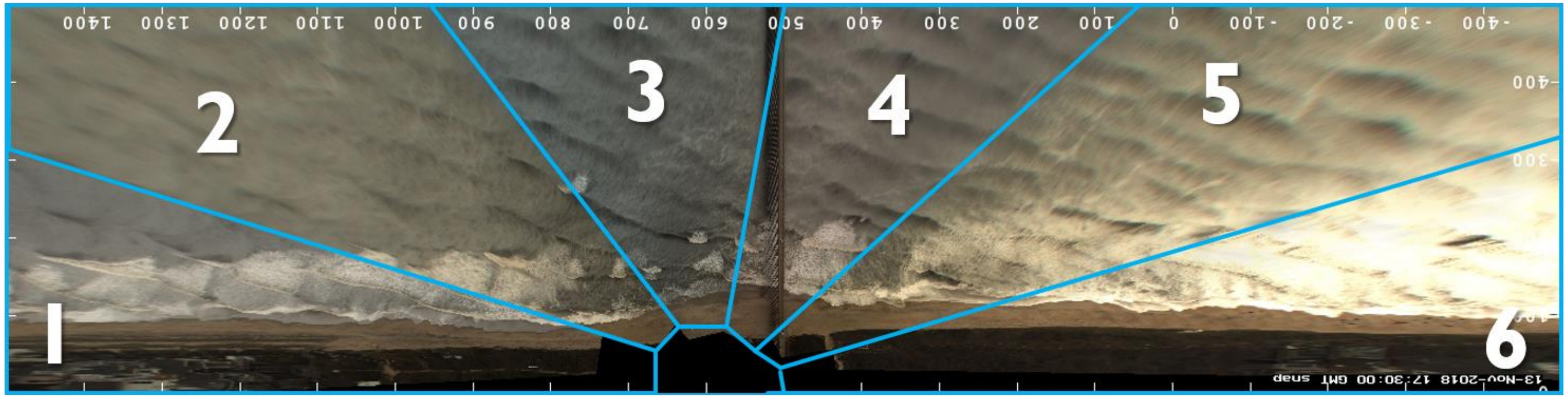
LEARN MORE: Determining Camera Geometries Lesson

Take Home Points:

- Knowing the position & orientation of your camera is critical for making maps with your images
- Need to re-do every-time camera moves OR use a correction algorithm to match features between images to remove movement
- Need known features (at least 4 GCPs; more is better) in FOV and must have a good distribution of GCPs throughout your FOV

IMAGE PROJECTION & GEORECTIFICATION OVERVIEW

- Image projection uses intrinsics & extrinsics in combination with known elevations in the field of view to project and rectify the image into map space

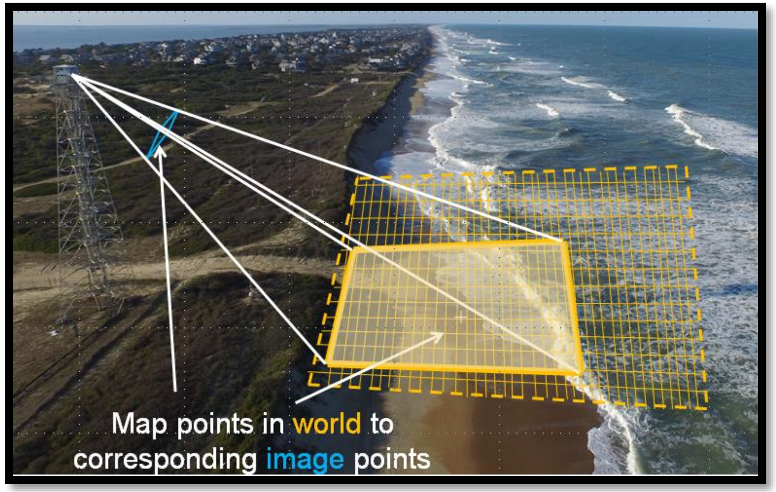


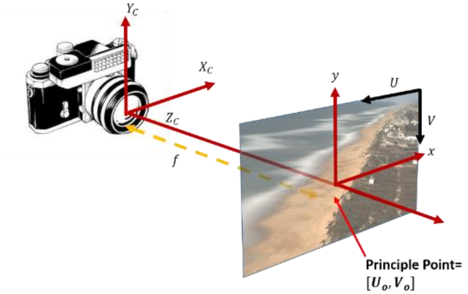
Take Home Points:

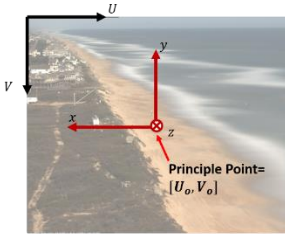
- In order to project an image from a single camera (solve the system of equations), we must know the elevation of features in the field of view
- Can assume a tide level for water pixels (introduces more error the lower your camera is)
- Projection is accomplished through a series of matrix multiplication operations

LEARN MORE:
Image Rectification Lesson

IMAGE PROJECTION (P-MATRIX): PIN-HOLE CAMERA FUNDAMENTAL PROCESS (IMAGE RESECTION)







- Make **grid** in World Coordinates (XYZ_w)
- Identify the corresponding Undistorted Pixel Coordinates (UV) for the grid points using camera projection matrix P = [K]*[R]*[T]
    - Convert World Coordinates (XYZ_w) to Camera Coordinates (XYZ_c) using *R & T* ***(EXTRINSICS)*** *in world units*
    - Convert and Homogenize (Scale) Camera Coordinates (XYZ_c) to Undistorted Pixel Coordinates (UV) in pixels using K ***(INTRINSICS)***
- Convert Undistorted Pixel Coordinates (UV) to Distorted Pixel Coordinates (UV_D) ***(INTRINSICS)***
- Extract Image intensities (RGB) from UV_D coordinates
- Assign to XYZ_w and plot grid as image!

Can be a local or world coordinate system

P

Distortion

Grid

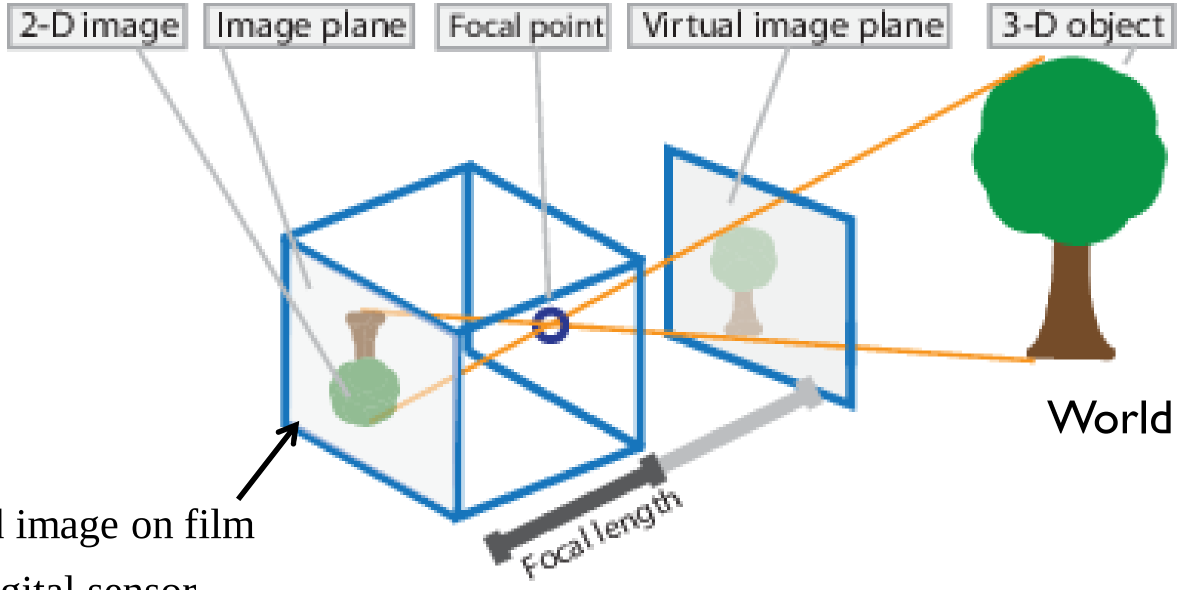
$$\begin{bmatrix} U_D \\ V_D \\ 1 \end{bmatrix} \leftrightarrow \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} \overset{*}{=} \underbrace{\begin{bmatrix} -f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{22} & r_{33} \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix}}_{T} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

Homogenization

MATH PRINCIPLES & CAMERA MODELS

- Geometric principles (collinearity, similar triangles, homogeneous coordinates) and linear algebra (rigid transformations, projections) allow us to exploit how a camera lens works to understand relationships between the locations of an object in the world and in an image
- Pin-Hole Camera Model – text & photo from mathworks "What is Camera Calibration?"

- A pinhole camera is a simple camera without a lens and with a single small aperture.
- Light rays pass through the aperture and project an inverted image on the opposite side of the camera (on the sensor).
- Think of the virtual image plane as being in front of the camera and containing the upright image of the scene.
- Image resectioning, is basically the process of figuring out which incoming light ray is associated with which image pixel



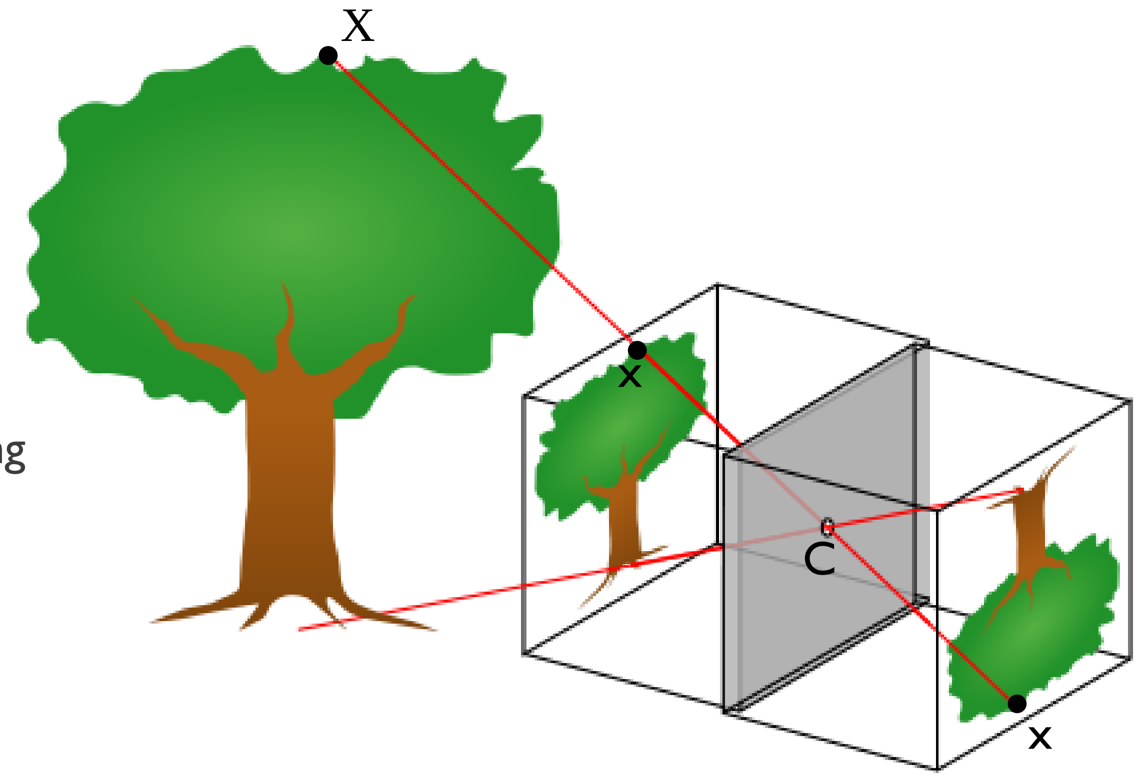
Inverted image on film plane/digital sensor

→ Focal length controls proper scale

BUILDING THE K MATRIX CONCEPT: COLLINEARITY

PinHole model assumes collinearity

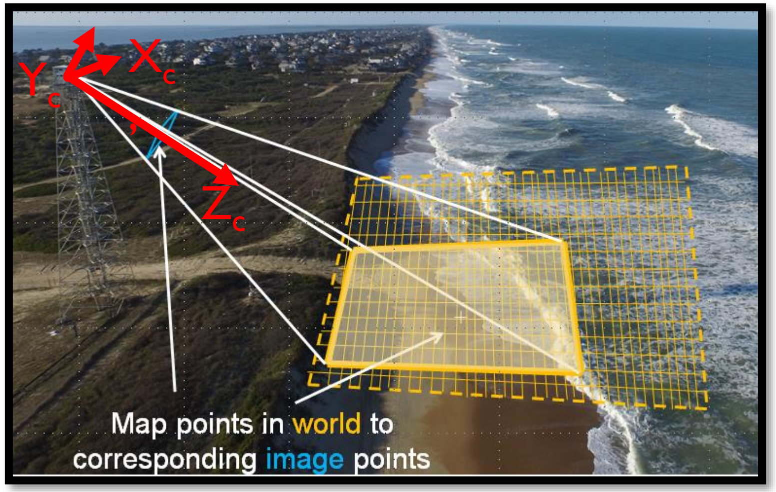
- Collinearity assumes:
  - the object point, X, in the world
  - the camera projection center, C, and
  - the projected point, x,
- are collinear (i.e. lie on a straight line)
- these "straight lines" are the light rays reflecting from objects in the field of view that travel through the pinhole to the camera sensor

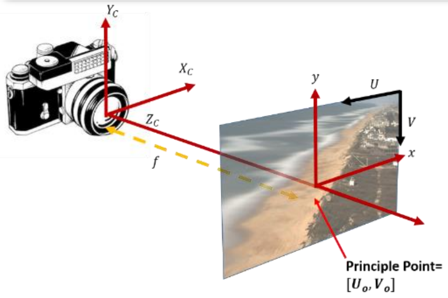


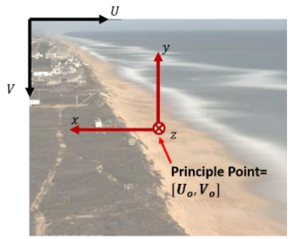
Note: the coordinates of the projected point, x, will be identical if a negative sensor is placed behind the camera or if a positive sensor is mirrored and placed in front of the camera center … basically defines relative direction of UV and xyz, and signage of fx, fy

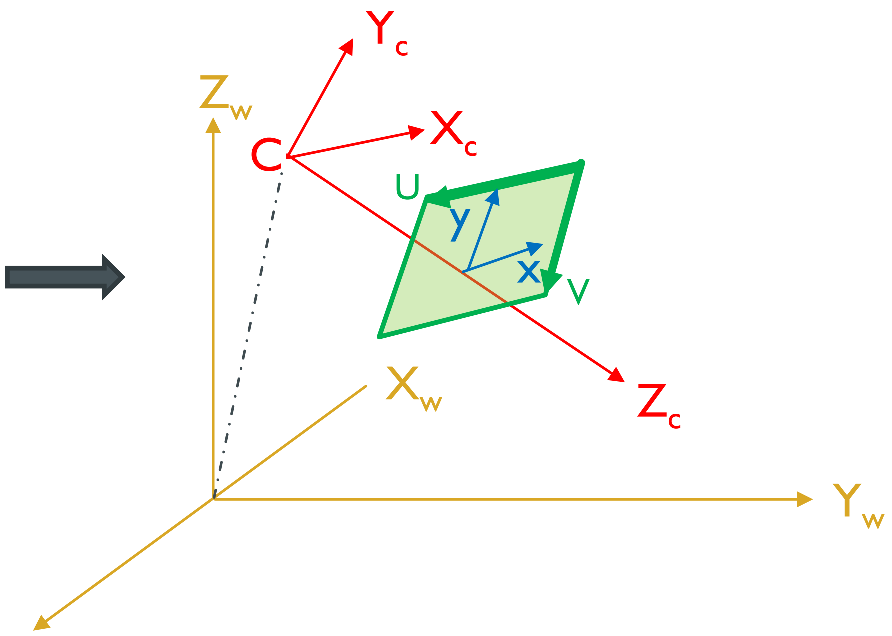
BUILDING THE K MATRIX
COORDINATE SYSTEMS & INITIAL ASSUMPTIONS

- We have four coordinate systems to deal with: **world** (X,Y,Z), **camera** (X_c,Y_c, Z_c), **image** (x, y), and **pixel** **(u, v)**



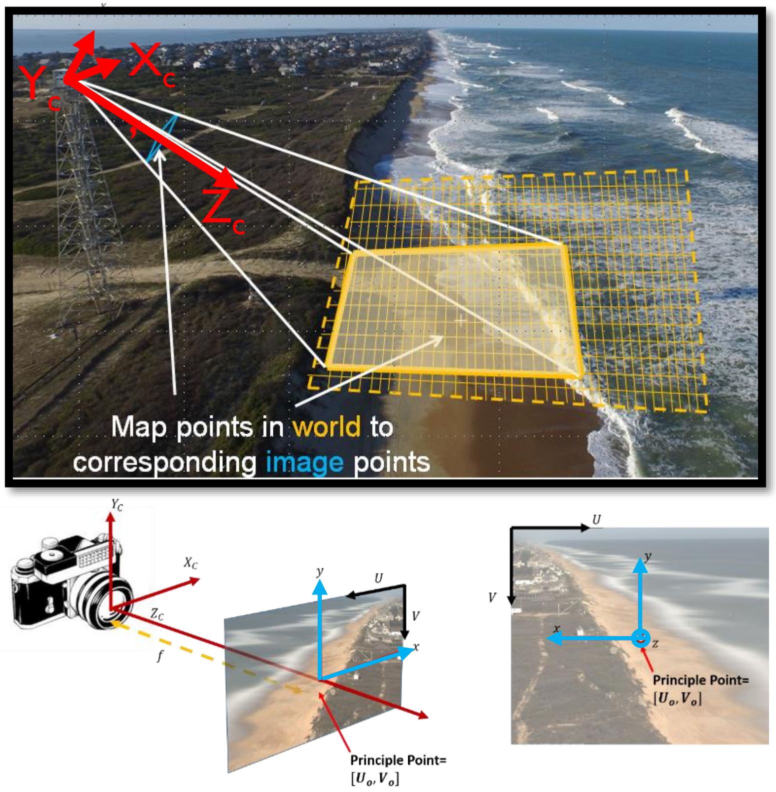


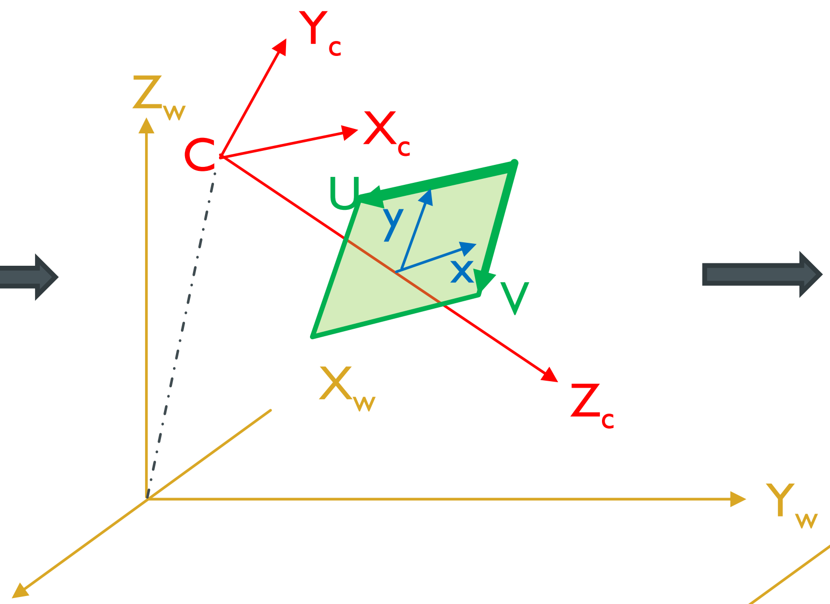




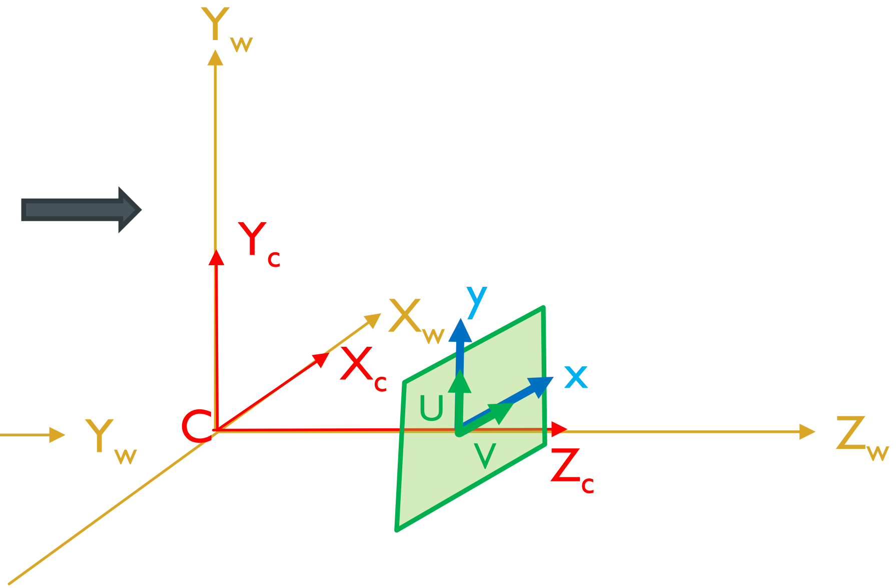
BUILDING THE K MATRIX
COORDINATE SYSTEMS & INITIAL ASSUMPTIONS

- Let's start with the assumptions that (1) our world and camera are aligned, and (2) our image and pixel are aligned to simplify:



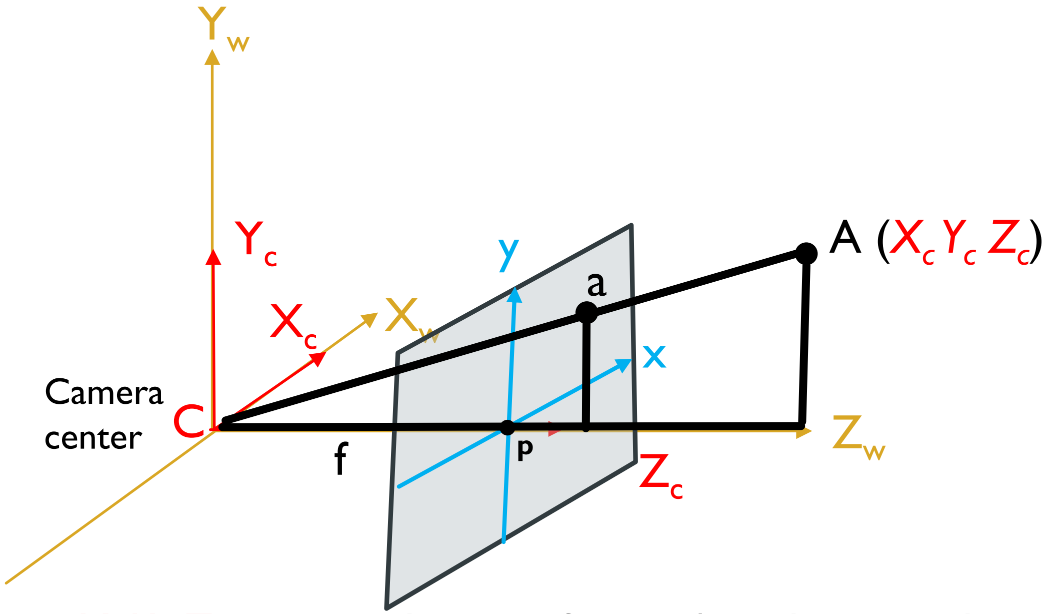


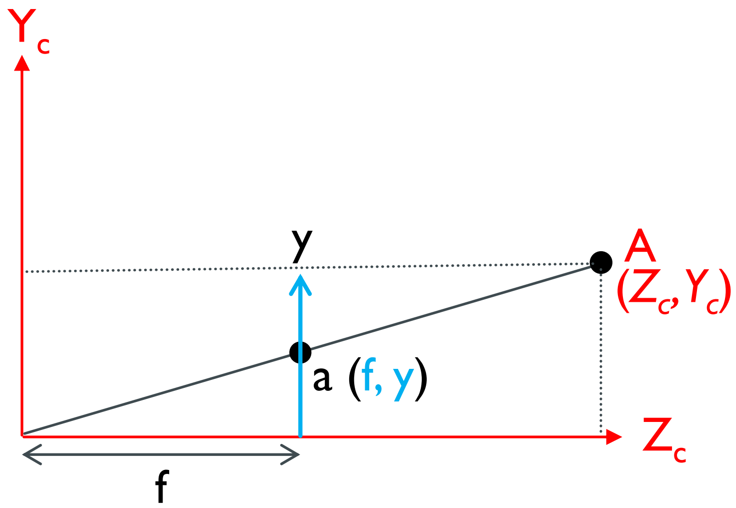
Typical Setup



Simplified Scenario

BUILDING THE K MATRIX
APPLYING COLLINEARITY & SIMILAR TRIANGLES CONCEPT





$$\frac{y}{f_y} = \frac{Y_c}{Z_c}$$

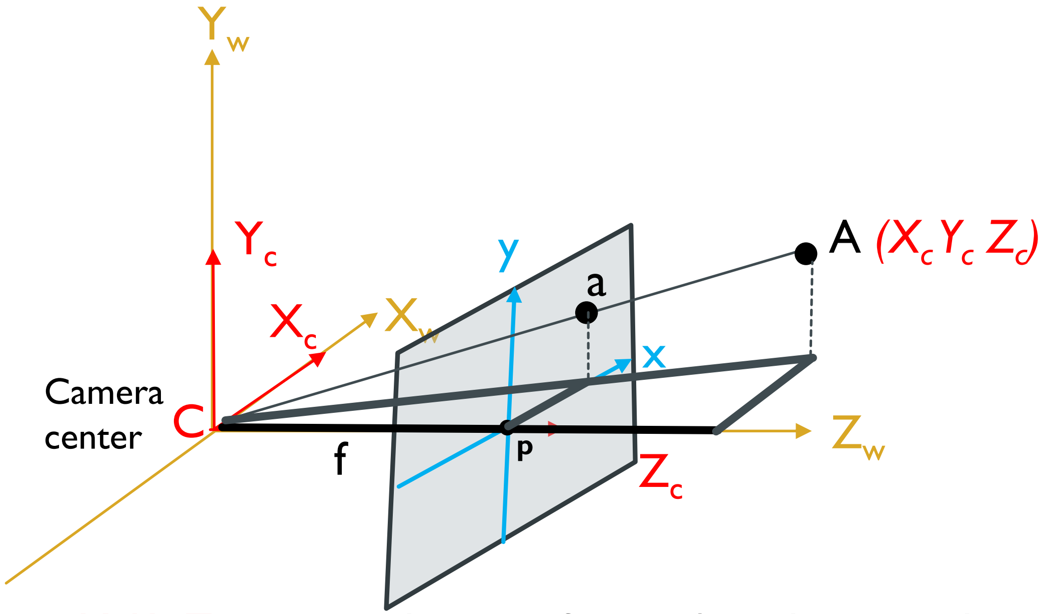
and

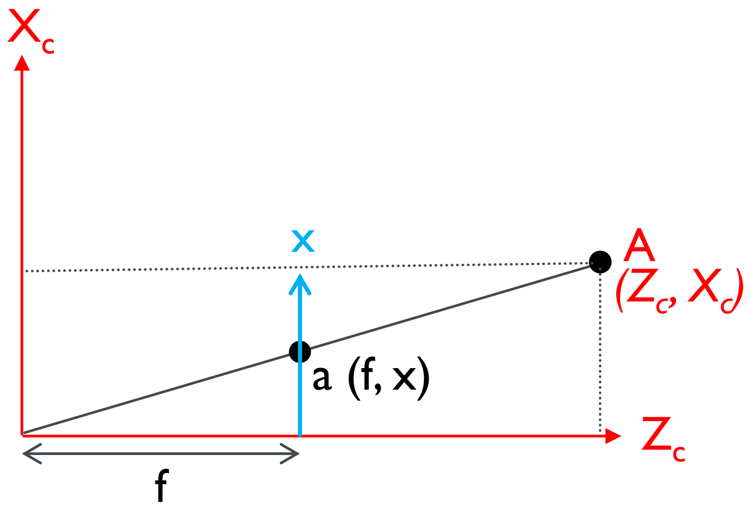
$$y = \frac{f_y Y_c}{Z_c}$$

- X_c, Y_c, Z_c are coordinates of point A in the aligned world and camera coordinate system
  - Note: In this case our world & camera are aligned (we'll talk later about what to do when they aren't)
- x, y, are image coordinates of corresponding point a along the same light ray in the image coordinate system;
- the image plane is at distance Z = f (i.e. image is located at f units out in the camera Z optics axis)
- f is the focal length measured in pixels; because at Z = f, 1 pixel = 1 image coordinate

From Hartley and Zisserman, Multiple View Geometry, 2003

BUILDING THE K MATRIX
APPLYING COLLINEARITY CONCEPT





$$\frac{x}{f_x} = \frac{X_c}{Z_c}$$

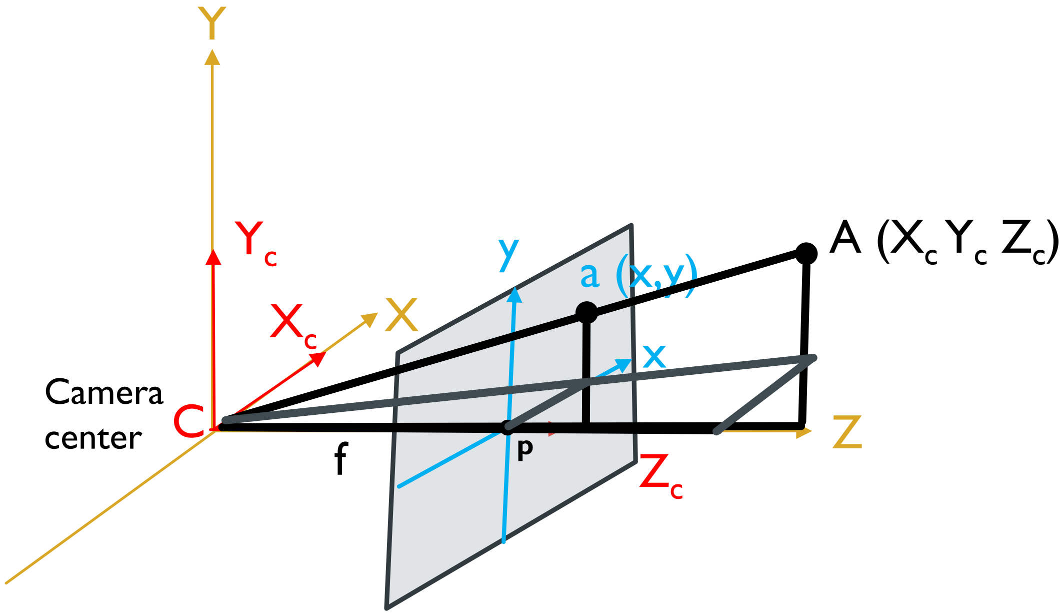
and

$$x = \frac{f_x X_c}{Z_c}$$

- X_c, Y_c, Z_c are coordinates of point A in the aligned world and camera coordinate system
  - Note: In this case our world & camera are aligned (we'll talk later about what to do when they aren't)
- x, y, are image coordinates of corresponding point a along the same light ray in the image coordinate system;
- the image plane is at distance Z = f (i.e. image is located at f units out in the camera Z optics axis)
- f is the focal length measured in pixels; because at Z = f, 1 pixel = 1 image coordinate

From Hartley and Zisserman,
Multiple View Geometry, 2003

BUILDING THE K MATRIX
APPLYING HOMOGENEOUS COORDINATES & COLLINEARITY



From before,

$$y = \frac{f_y Y_c}{Z_c} \qquad x = \frac{f_x X_c}{Z_c}$$

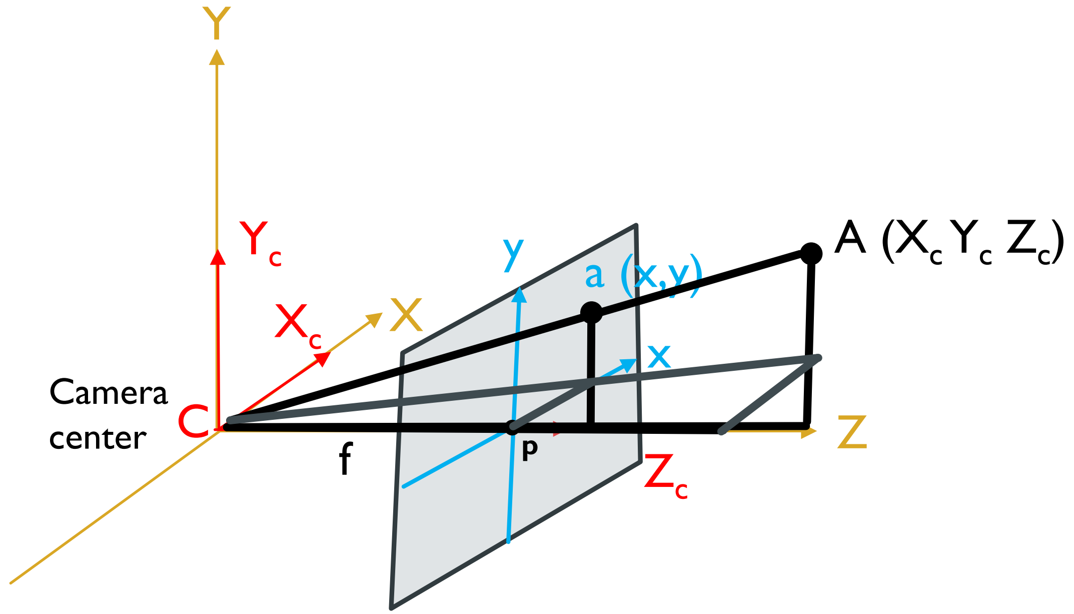
In Matrix Form

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{f_x X_c}{Z_c} \\ \frac{f_y Y_c}{Z_c} \end{bmatrix}$$

which maps the camera coordinates into image coordinates

But…..

BUILDING THE K MATRIX
APPLYING HOMOGENEOUS COORDINATES & COLLINEARITY



This 2x1 Matrix, i.e. 2 dimensions (xy) wont work with
Our 3x1 Matrices, i.e. 3 dimensions (XcYcZc)

So we rewrite it as this…

$$y = \frac{f_y Y_c}{Z_c} \qquad x = \frac{f_x X_c}{Z_c}$$

In Matrix Form

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x X_c \\ f_y Y_c \\ Z_c \end{bmatrix} \frac{1}{Z_c}$$

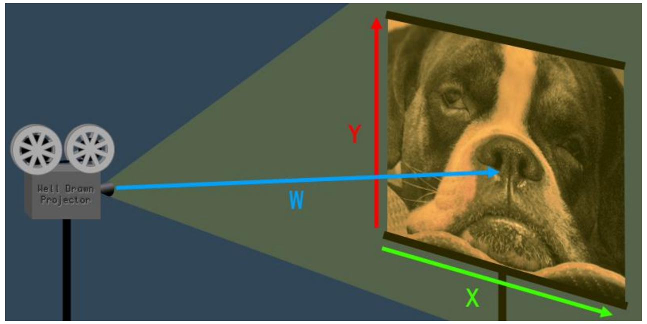
We call this homogenization

BUILDING THE K MATRIX
CONCEPT: HOMOGENOUS COORDINATES

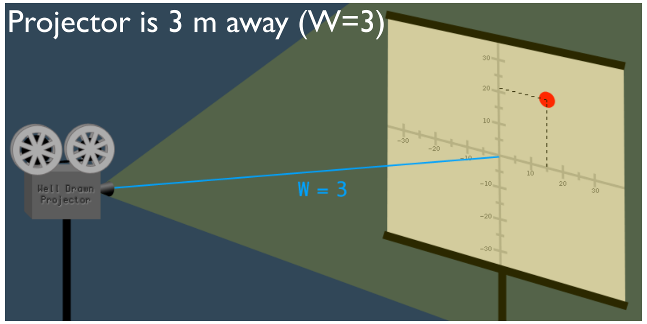
- Projective Geometry uses "Homogenous Coordinates" to describe points & lines
- To transform a Cartesian 2D point $[x, y]^T$ you simply add a unit value as an extra coordinate $[x, y, 1]^T$
- This third coordinate controls the "scale" of the image
  - If you think in 2D; the third coordinate changes the scale of the original $[x, y]^T$ based on how far away your image is (projector example at right)
  - We always want the third dimension to be 1, because we want the world to be at the right scale (i.e. accounting for the focal length)!

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \overbrace{\underbrace{\begin{bmatrix} -f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{22} & r_{33} \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix}}_{T}}^{P} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

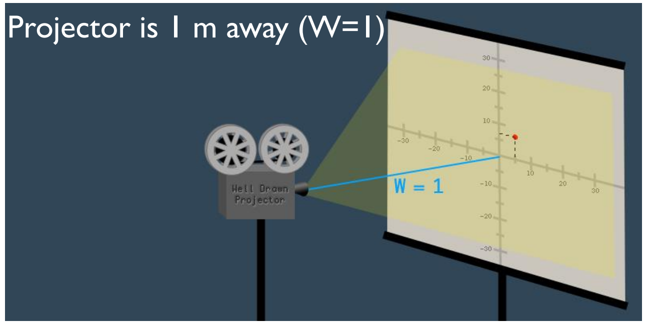
* Homogenization



x, y are image coordinates
w controls the "scale" of the image



Dot at (15, 21) (2D Coordinates)
(15, 21, 3) (Projected Coordinates)



Dot at (5, 7) (2D Coordinates)
(15/3, 21/3, 3/3) (Projected Coordinates)
(5, 7, 1) (Homogeneous Coordinates)

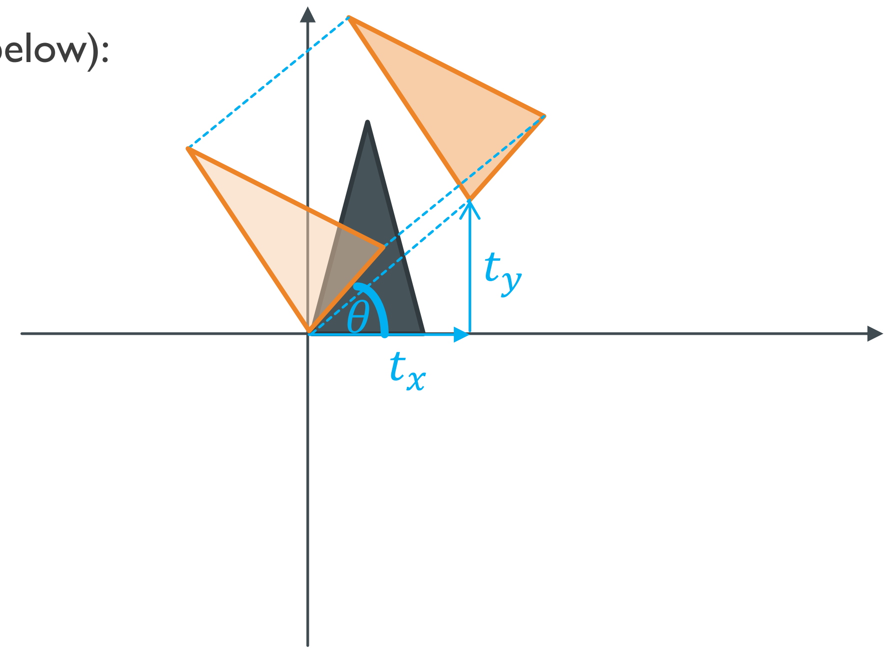
https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/

BUILDING THE K MATRIX
CONCEPT: HOMOGENEOUS COORDINATES & RIGID TRANSFORMATIONS

- Homogeneous coordinates allow us to do rotation and translation calculations (a rigid transformation) as one matrix multiplication step (makes the math easier)
  - Transformation matrix translates and rotates in one step (2D example below):

$$x\,cos\theta + y\,sin\theta + \ t_x = x'$$
$$-x\,sin\theta + y\,cos\theta + \ t_y = y'$$

$$\begin{pmatrix} cos(\theta) & sin(\theta) & t_x \\ -sin(\theta) & cos(\theta) & t_x \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$



SIMPLIFIED K MATRIX (ASSUMES CAMERA & IMAGE ALIGNED)

- So, if $x = f_x X_c / Z_c$ and $y = f_y Y_c / Z_c$, *then* $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} fX_c \\ fY_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \frac{fX_c}{Z_c} \\ \frac{fY_c}{Z_c} \\ 1 \end{bmatrix}$ which maps the camera coordinates into image coordinates

- Which can also be written in matrix form as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} fX_c \\ fY_c \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Image Coordinates

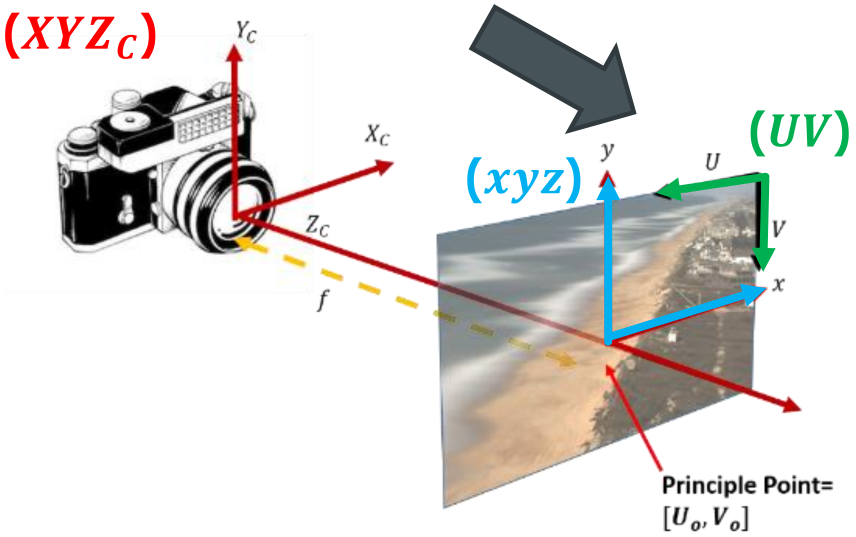
Basic Camera Matrix (Simplified K Matrix)

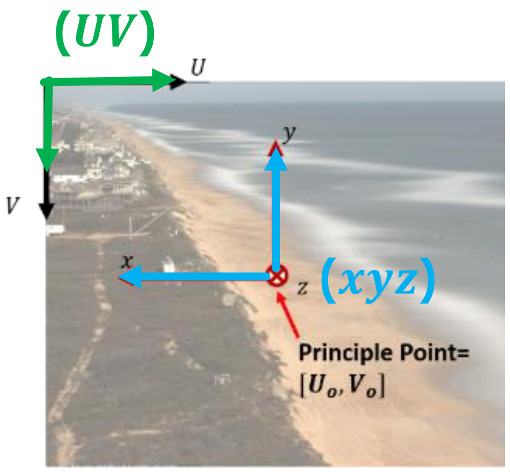
Camera Coordinates

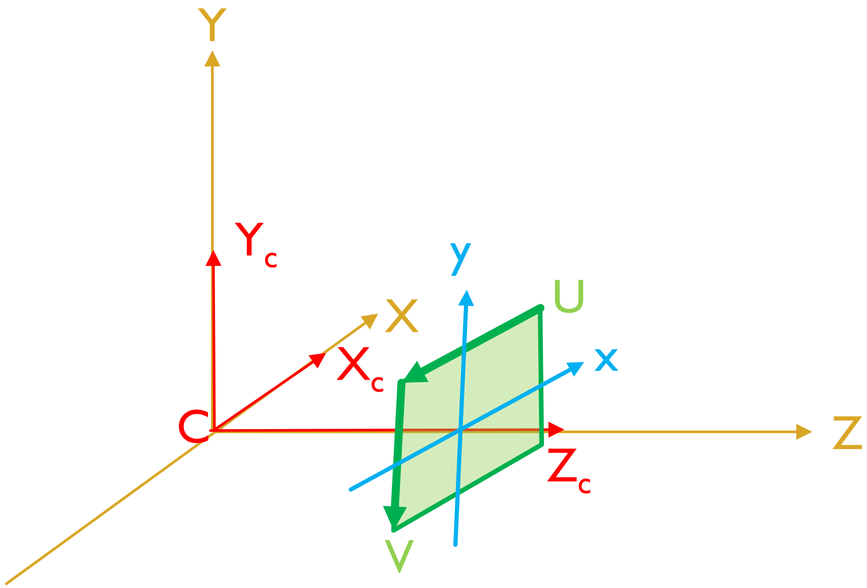
Note: this camera matrix made simplifying assumptions; so we'll add a few more parts to make it practical.

K MATRIX – ACCOUNTING FOR IMAGE COORDINATES (U,V)

- First, the prior slides assumed that the camera and image and **pixel** coordinate systems were aligned and centered on the principle point, BUT:
  - We tend to refer to pixel coordinates as (U,V) with U increasing to the right, and V increasing down





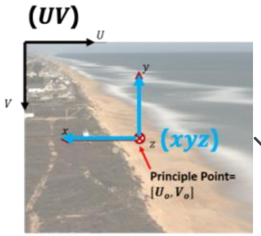


CiRN

K MATRIX – ACCOUNTING FOR PRINCIPLE POINTS & SKEWNESS

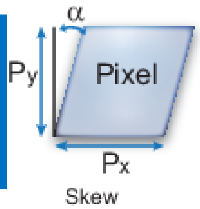
- Second, the prior slides also assumed the lens was centered on the sensor and that sensor pixels were perfectly square:
  - If the principle point (U_0,V_0) is not at the origin of the image coordinate system, we need to adjust our image coordinates relative to that origin and account for the reversal of coordinates:

Negatives account for the rotation between image and pixel coordinates



$$x = U_0 - U$$
$$y = V_0 - V$$

Note: sometimes the pixels also aren't truly orthogonal, so we can solve for pixel skewness too if desired



- SO:

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} -f_x & s & U_0 \\ 0 & -f_y & V_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Pixel Coordinates

Camera Matrix [K]

Camera Coordinates

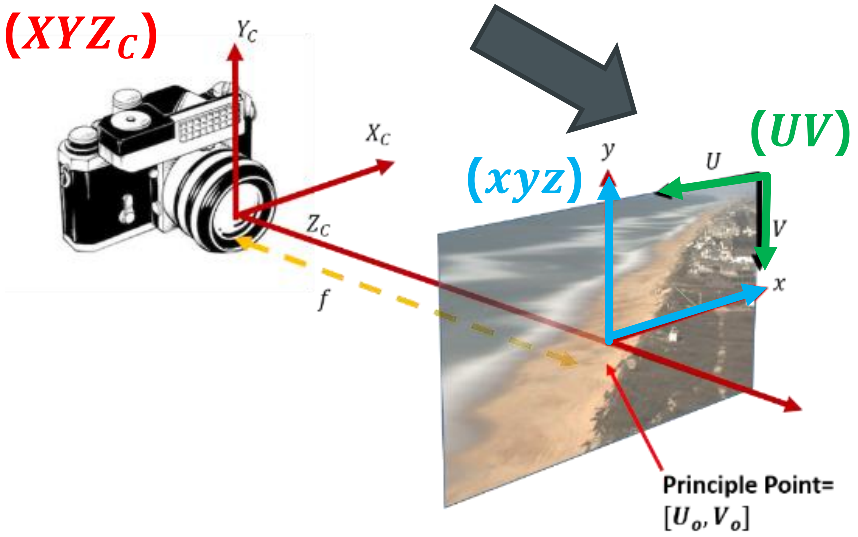
LEARN MORE:
The Lens Distortion Lesson will teach us how to solve for the inputs to K (intrinsics) for your specific camera/lens combination

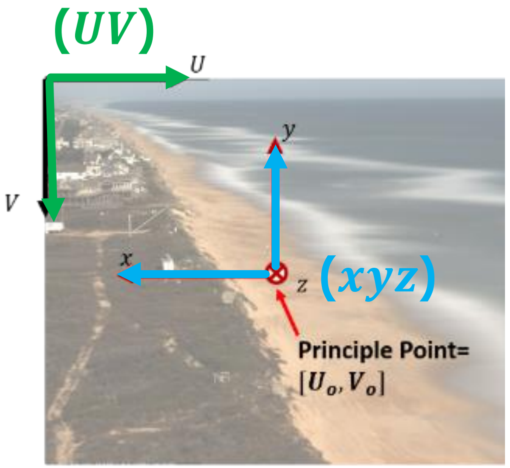
FINAL K MATRIX

- Converts Camera (XYZ_C) to Image (xyz) and Undistorted Pixel Coordinates (UV) using K

$$\begin{bmatrix}U\\V\\1\end{bmatrix} = \frac{1}{Z_c}\begin{bmatrix}-f_x & s & U_0\\0 & -f_y & V_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix}$$

Pixel Coordinates | Camera Matrix [K] | Camera Coordinates





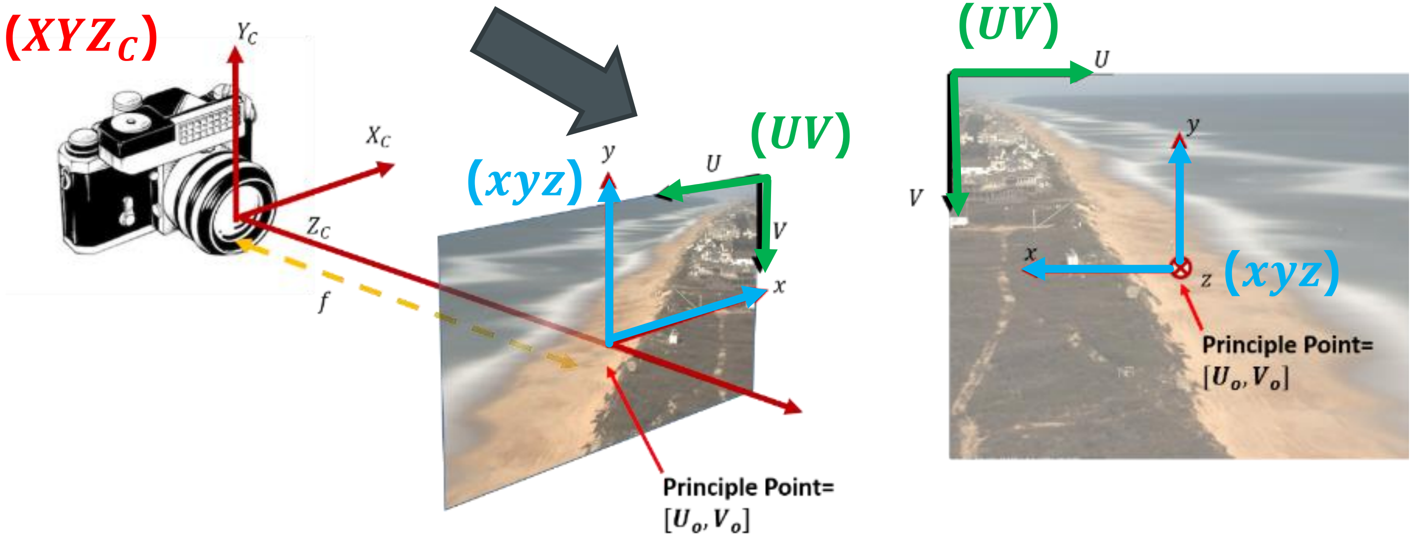
- $[U_o, U_v]$ are principle point coordinates
- $-f_x, -f_y$ are focal lengths in pixel units
- Values are defined by **intrinsics**
- Note, image coordinates must be homogenized to get final units of U,V in pixels

FINAL K MATRIX

- Converts Camera (XYZ_C) to Image (xyz) and Undistorted Pixel Coordinates (UV) using K

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} -f_x & s & U_0 \\ 0 & -f_y & V_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Pixel Coordinates — Camera Matrix [K] — Camera Coordinates



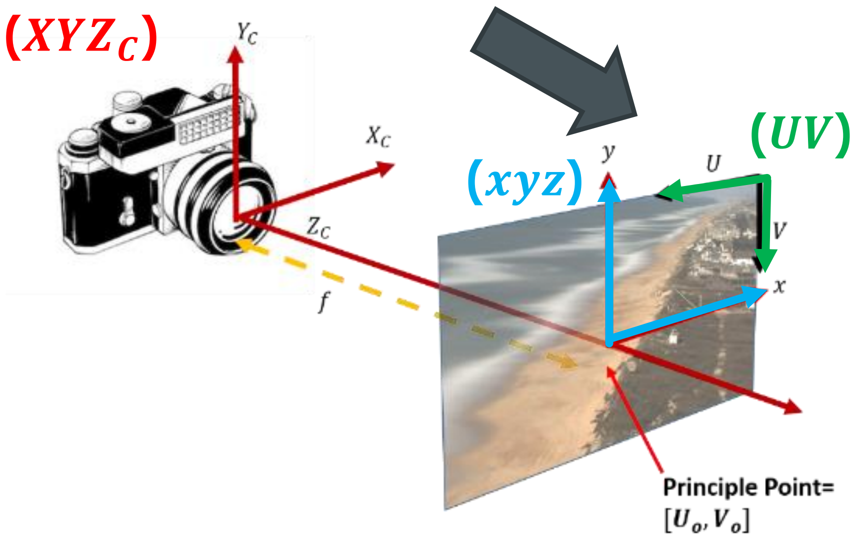
LEARN MORE:
The Lens Distortion Lesson will teach us how to solve for the inputs to K (intrinsics) for your specific camera/lens combination

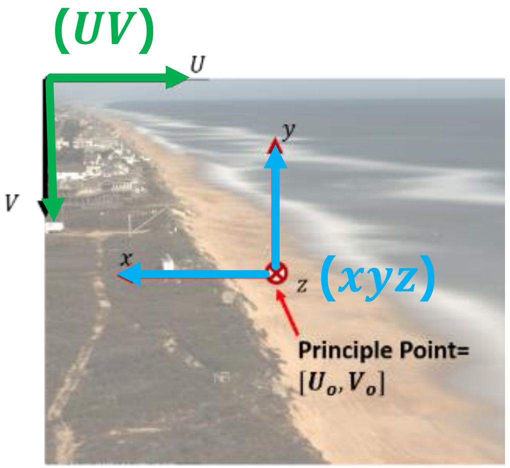
FINAL K MATRIX

- Converts Camera (XYZ_C) to Image (xyz) and Undistorted Pixel Coordinates (UV) using K

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =* \begin{bmatrix} -f_x & s & U_0 \\ 0 & -f_y & V_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Pixel Coordinates | Camera Matrix [K] | Camera Coordinates

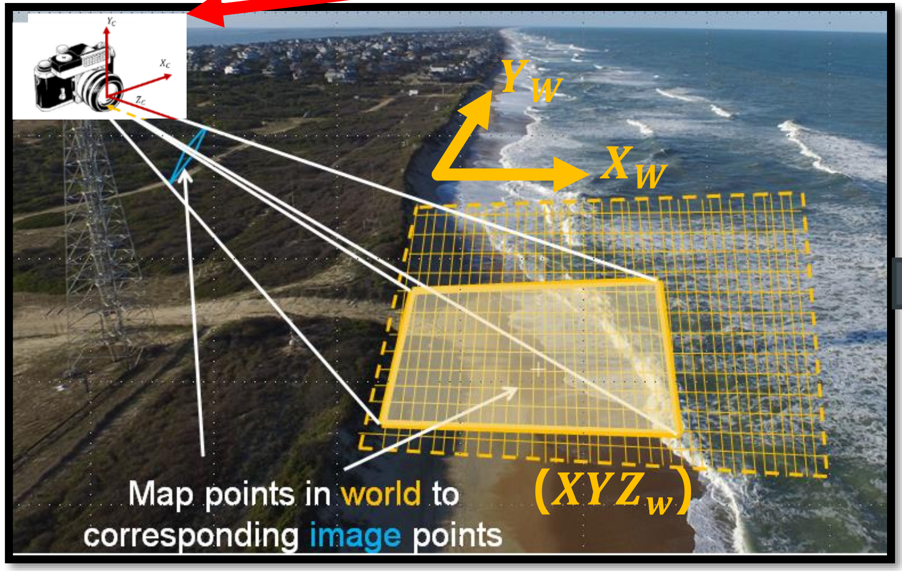


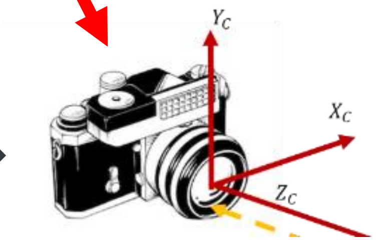


*** Note, as our equations become more complicated--To deal with homogeny, we typically carry out all RHS multiplication and divide answer by (3,1) entry in final 3x1 Matrix, homogenization will be represented by * for now on.**

BUILDING R & T MATRICES
WHAT ABOUT WHEN OUR WORLD & CAMERA AREN'T ALIGNED?

- Convert **World** to Camera Coordinates (XYZ_C) using $\boldsymbol{R,T}$





Distortion

$$\begin{bmatrix} U_D \\ V_D \\ 1 \end{bmatrix} \leftrightarrow \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = {}^{*}\underbrace{\begin{bmatrix} -f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{22} & r_{33} \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix}}_{T} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

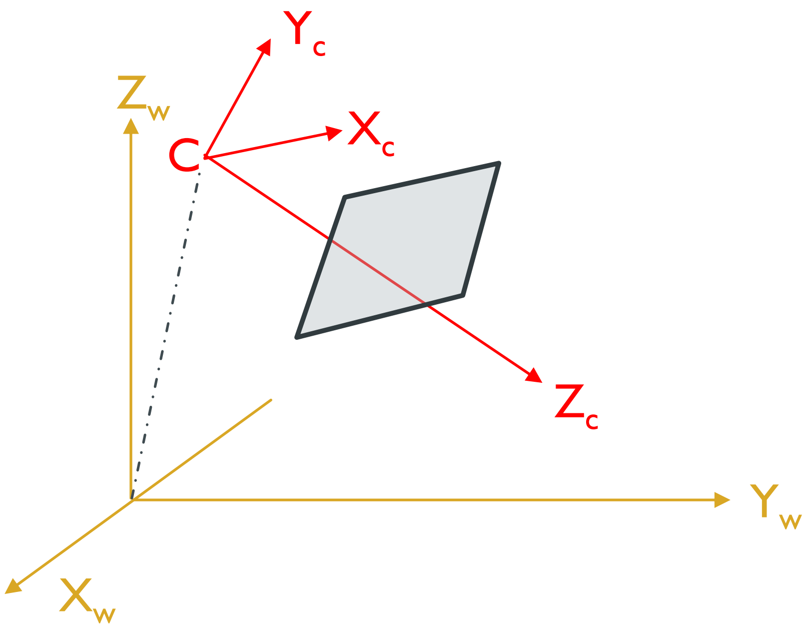
P Grid

* Homogenization

- T translation matrix of camera position $[C_{XW,}C_{YW,}C_{ZW,}]$ in World coordinates (XYZ_w)
- R is rotation matrix; aligns orientation of real world coordinates (XYZ_w) to camera coordinates (XYZ_C)
- User can enter orientation (azimuth, pitch, and swing) of camera, toolbox calculates corresponding R
- Camera Position & Orientation are defined by **extrinsics**
- Note, after conversion to camera coordinates, units are still in world units (e.g. meters)

BUILDING R & T MATRICES
STEPPING THROUGH TRANSLATION & ROTATION

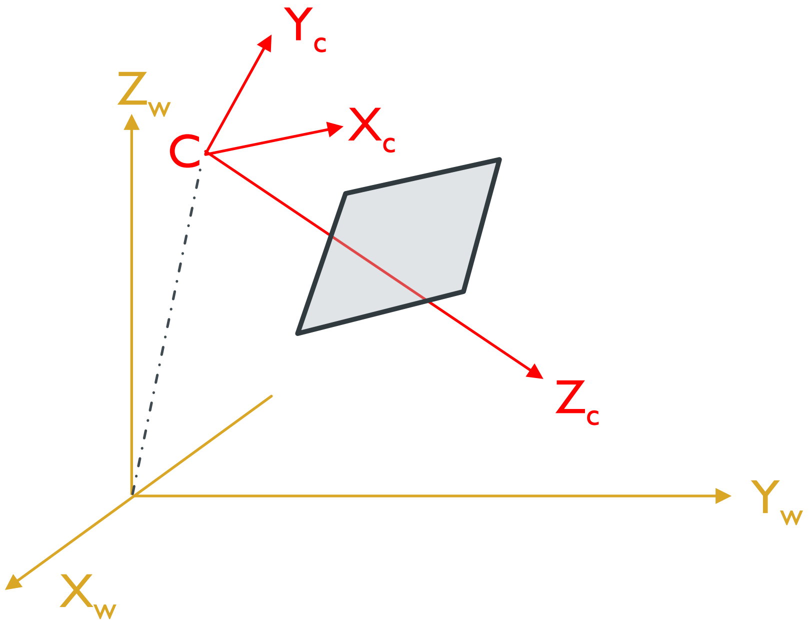
- Need to transform our world coordinate system (X_w, Y_w, Z_w) to align with the camera system (X_c, Y_c, Z_c) through a rigid transformation (rotation & translation), so we can use those principles of collinearity and similar triangles to apply the pinhole camera model:



To do this, we need a rotation, R, and a translation where C = (C_{xw}, C_{yw}, C_{zw}) is the camera center in world coordinates:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \left[\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - \begin{bmatrix} C_{xw} \\ C_{yw} \\ C_{yz} \end{bmatrix} \right]$$

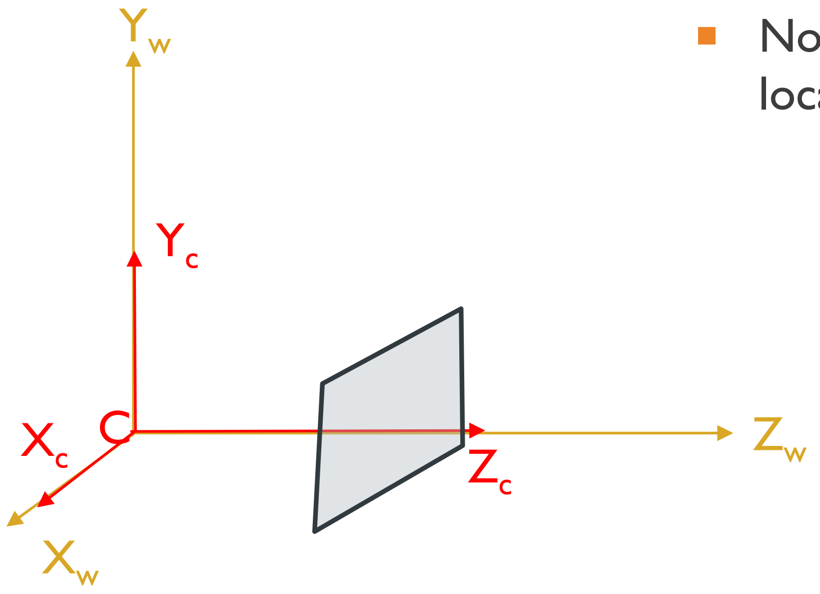
BUILDING R & T MATRICES
ROTATION & TRANSLATION – ALIGNING WORLD & CAMERA



- First we translate (make (0, 0) in world coordinates equal to (0, 0) in camera coordinates) by subtracting the world coordinates of the camera location from our grid.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \left[\underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - \begin{bmatrix} C_{xw} \\ C_{yw} \\ C_{yz} \end{bmatrix}}_{\text{translation}} \right]$$

BUILDING R & T MATRICES
ROTATION & TRANSLATION – ALIGNING WORLD & CAMERA



- Now that **world** & **camera** are centered on the same location, we can rotate to align the coordinate systems

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \underbrace{R}_{\text{rotation}} \left[\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - \begin{bmatrix} C_{xw} \\ C_{yw} \\ C_{yz} \end{bmatrix} \right]$$

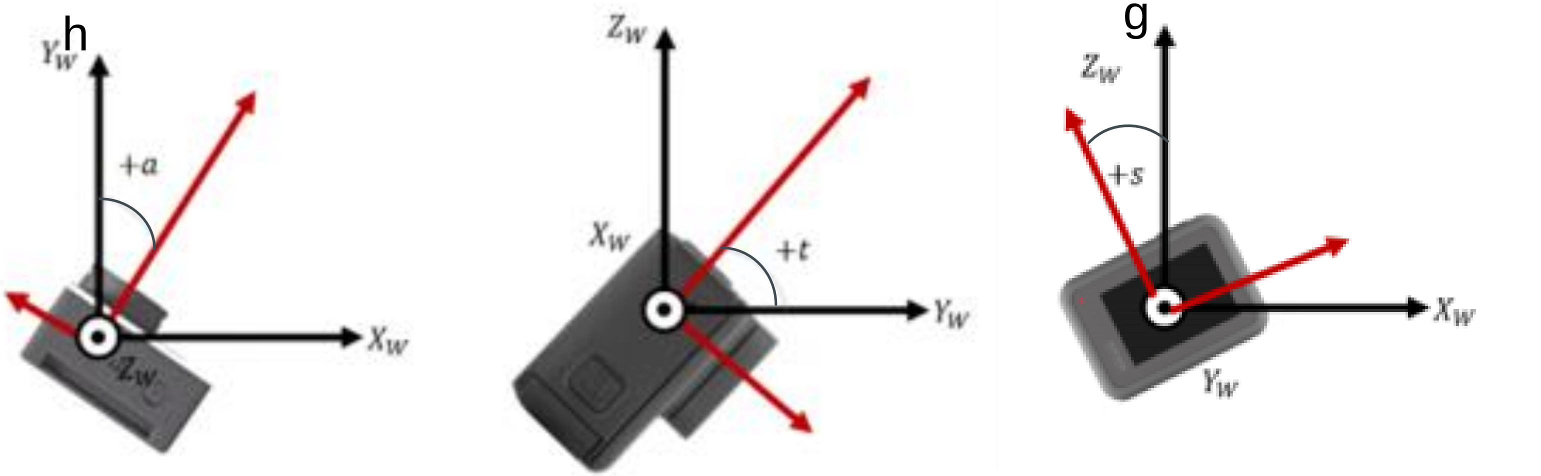
BUILDING R & T MATRICES
EXTRINSIC ROTATION DEFINITION

Can define R by three rotation angles: azimuth, tilt, swing (a,t,s) or (A,T,S)

Azimut
h

Tilt

Swin
g



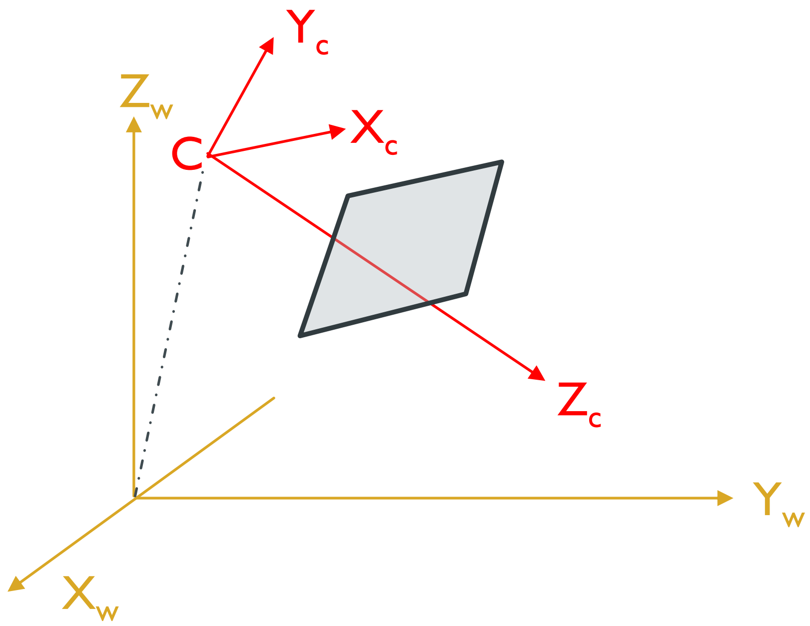
$R_{W\rightarrow C}$=

$$\begin{bmatrix} -\sin(\mathrm{A})\sin(\mathrm{S})\cos(\mathrm{T}) - \cos(S)\cos(A) & \cos(\mathrm{S})\,\mathrm{Sin}(\mathrm{A}) - \mathrm{Sins}(\mathrm{S})\mathrm{Cos}(\mathrm{T})\cos(\mathrm{A}) & -\sin(T)\sin(S) \\ \sin(\mathrm{A})\cos(\mathrm{T})\,\mathrm{Cos}(\mathrm{S}) - \cos(\mathrm{A})\sin(\mathrm{s}) & \sin(\mathrm{S})\,\mathrm{Sin}(\mathrm{A}) + \cos(\mathrm{A})\,\mathrm{Cos}(\mathrm{T})\cos(\mathrm{S}) & \cos(S)\sin(T) \\ \sin(A)\sin(T) & \sin(T)\cos(A) & -\cos(T) \end{bmatrix}$$

So this $R_{W\rightarrow C} = R$ takes world coordinates (XYZ, and transforms them to Camera Coordinates XcYc,Zc).

More information about rotation matrix construction can be found in the user manual.

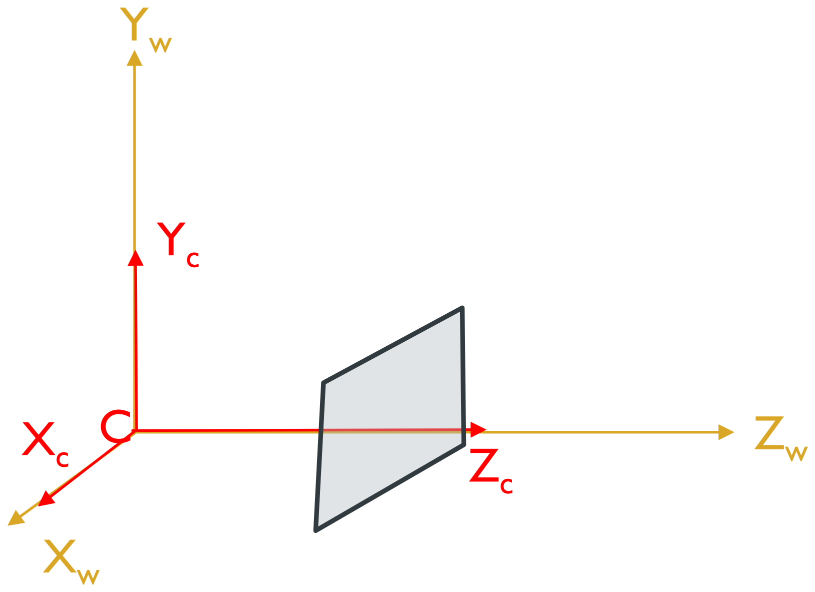
PROJECTIVE MAPPING – REVIEW OF STEPS



- 1. Translate world reference frame to camera-centric world coordinates (move origin to camera location)

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =* KR \overbrace{\begin{bmatrix} 1 & 0 & 0 & -C_{xw} \\ 0 & 1 & 0 & -C_{yw} \\ 0 & 0 & 1 & -C_{zw} \end{bmatrix}}^{[T]} \overbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}^{\text{Grid}}$$

PROJECTIVE MAPPING – REVIEW OF STEPS



- 2. Rotate camera centric world coordinates into camera orientation (align world & camera reference systems) to get camera coordinates (X_c, Y_c, Z_c)

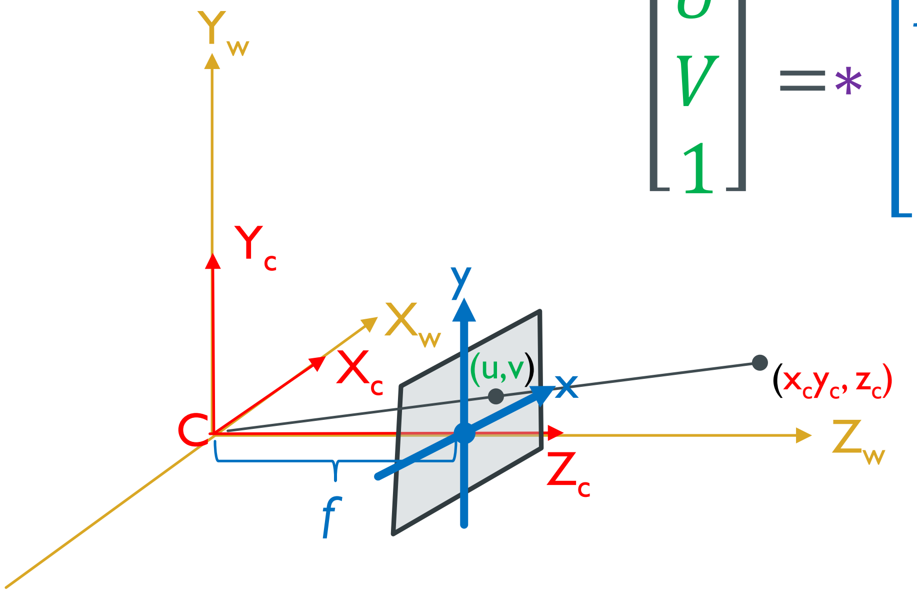
$$
\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =* K \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{[\,\mathbf{R}\,]} \underbrace{\begin{bmatrix} 1 & 0 & 0 & C_x \\ 0 & 1 & 0 & C_y \\ 0 & 0 & 1 & C_z \end{bmatrix}}_{[\,\mathbf{T}\,]} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\mathbf{Grid}}
$$

PROJECTIVE MAPPING – REVIEW OF STEPS

- 3. Convert camera coordinates (X_c, Y_c, Z_c) to Image Coordinates and homogenized and undistorted **Pixel coordinates (U,V)**

Undistorted Pixel Coordinates

$$\underbrace{\begin{bmatrix} U \\ V \\ 1 \end{bmatrix}}_{\text{Undistorted Pixel Coordinates}} =* \underbrace{\begin{bmatrix} f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{[\,K\,]} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{[\,R\,]} \underbrace{\begin{bmatrix} 1 & 0 & 0 & C_x \\ 0 & 1 & 0 & C_y \\ 0 & 0 & 1 & C_z \end{bmatrix}}_{[\,T\,]} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{Grid}}$$



CiRN

PROJECTIVE MAPPING – THE FINAL PROJECTION MATRIX (P)

- Calculate the un-distorted pixel coordinates of an object in the real world as:

P=[K][R][T]

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =* \begin{bmatrix} f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix} \begin{bmatrix} X_w \\ Y_W \\ Z_w \\ 1 \end{bmatrix}$$

System of equations written out

$$U = U_0 - f_x \frac{r_{11}(X_w - C_{xw}) + r_{12}(Y_w - C_{yw}) + r_{13}(Z_w - C_{zw})}{* \, r_{31}(X_w - C_{xw}) + r_{32}(Y_w - C_{yw}) + r_{33}(Z_w - C_{zw})}$$

$$V = V_0 - f_y \frac{r_{21}(X_w - C_{xw}) + r_{22}(Y_w - C_{yw}) + r_{23}(Z_w - C_{zw})}{* \, r_{31}(X_w - C_{xw}) + r_{32}(Y_w - C_{yw}) + r_{33}(Z_w - C_{zw})}$$

$(X_w, Y_w\ Z_w)$ = world coordinates of object
C = camera center in world coordinates (C_{XW}, C_{YW}, C_{ZW})
R = rotation matrix with 9 elements (3 rotation angles)
K = intrinsic parameters (U_0, V_0, f_x, f_y)
(U,V) = undistorted pixel coordinates of object
* Denominator is homogenization part

PROJECTIVE MAPPING – THE FINAL PROJECTION MATRIX (P)

- Calculate the un-distorted pixel coordinates of an object in the real world as:

P=[K][R][T]

Distortion

$$\begin{bmatrix} U_D \\ V_D \\ 1 \end{bmatrix} \longleftrightarrow \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =* \begin{bmatrix} f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

System of equations written out

$$U = U_0 - f_x \frac{r_{11}(X_w - C_{xw}) + r_{12}(Y_w - C_{yw}) + r_{13}(Z_w - C_{zw})}{* \, r_{31}(X_w - C_{xw}) + r_{32}(Y_w - C_{yw}) + r_{33}(Z_w - C_{zw})}$$

$$V = V_0 - f_y \frac{r_{21}(X_w - C_{xw}) + r_{22}(Y_w - C_{yw}) + r_{23}(Z_w - C_{zw})}{* \, r_{31}(X_w - C_{xw}) + r_{32}(Y_w - C_{yw}) + r_{33}(Z_w - C_{zw})}$$

$(X_w, Y_w\ Z_w)$ = world coordinates of object
C = camera center in world coordinates (C_{XW}, C_{YW}, C_{ZW})
R = rotation matrix with 9 elements (3 rotation angles)
K = intrinsic parameters (U_0, V_0, f_x, f_y)
(U,V) = undistorted pixel coordinates of object
* Denominator is homogenization part

PROJECTIVE MAPPING – THE FINAL PROJECTION MATRIX (P)

* Homogenization

P=[K][R][T]

Distortion

$$\begin{bmatrix} U_D \\ V_D \\ 1 \end{bmatrix} \longleftrightarrow \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =^* \begin{bmatrix} f_x & 0 & x_p \\ 0 & -f_y & y_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix} \begin{bmatrix} X_w \\ Y_W \\ Z_w \\ 1 \end{bmatrix}$$

System of equations written out

K = Camera Calibration Matrix (Intrinsics; 4-5 unknowns solved for by lens calibration)

R = Rotation Matrix (Camera Orientation) (Extrinsics; 3 unknowns solved for with GCPs)

T = Translation Matrix (Camera Position) (Extrinsics; 3 unknowns solved for with GCPs or measured)

COORDINATE DISTORTION

Convert Image to Distorted Image Coordinates (UV_D)

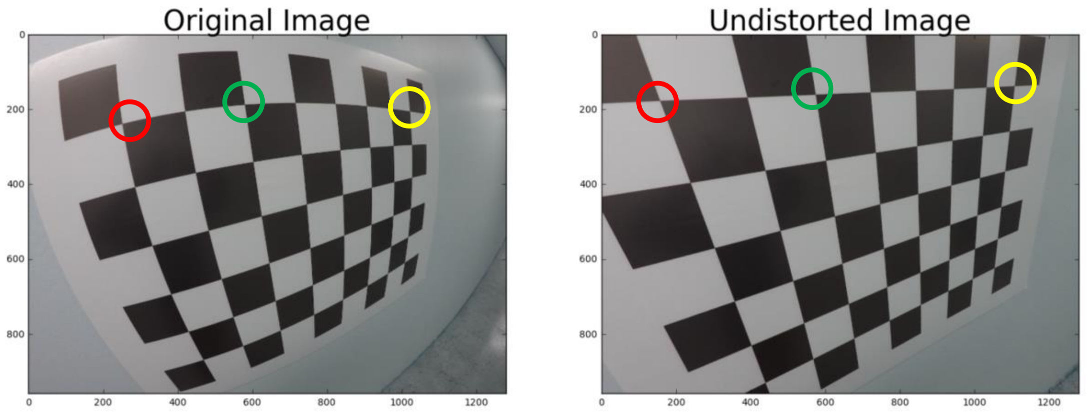
P=[K][R][T]

Distortion

$$\begin{bmatrix} U_D \\ V_D \\ 1 \end{bmatrix} \leftrightarrow \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} =* \begin{bmatrix} f_x & 0 & x_p \\ 0 & -f_y & y_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

Projection matrix uses a Pin Hole model and assumes NO Distortion

Distorted (Ud,Vd) ← Undistorted (U,V)



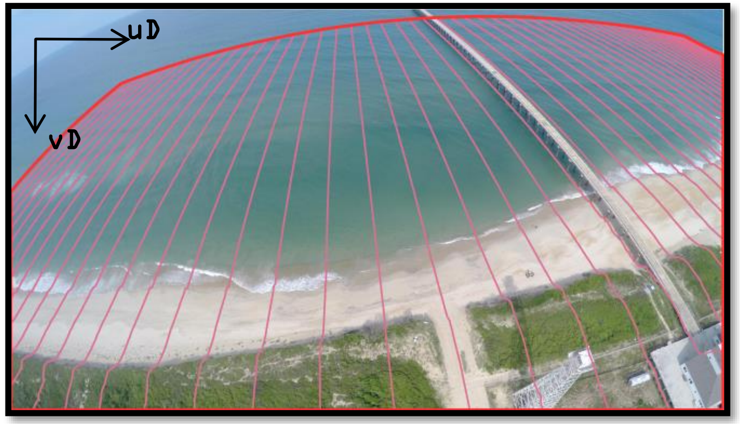
Final step is to distort the UV undistorted coordinates to the actual “distorted” coordinates that result because of the shape of lens.

LEARN MORE:
Lens Distortion Lesson

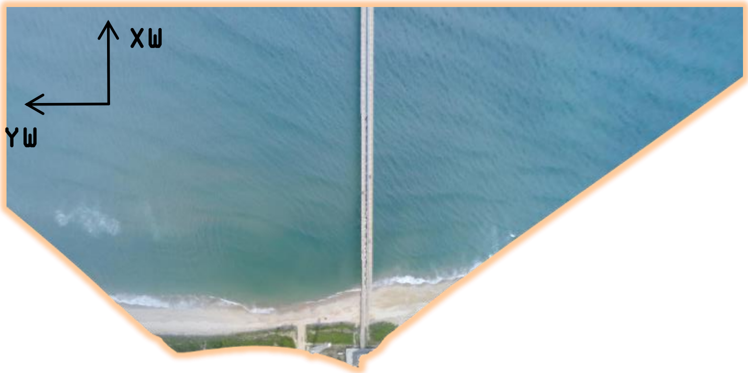
FINAL STEP: GENERATE GEO-RECTIFIED IMAGE

Extract Image intensities (RGB) from UV_D coordinates

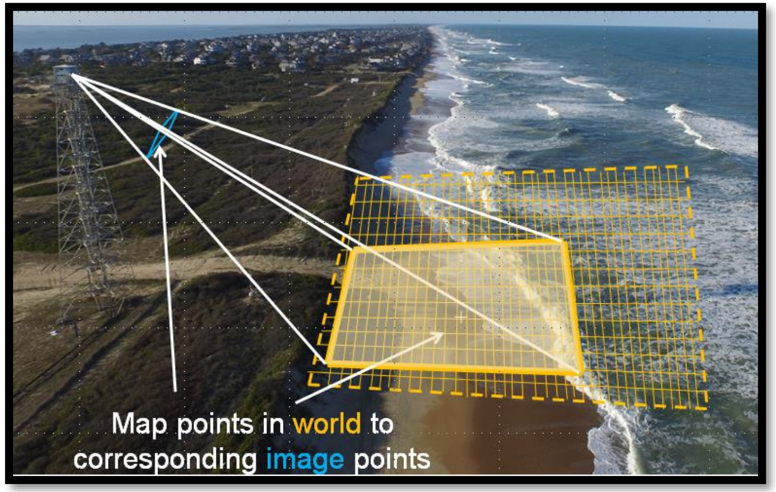
UV_D

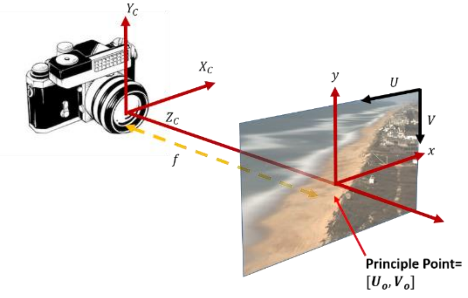


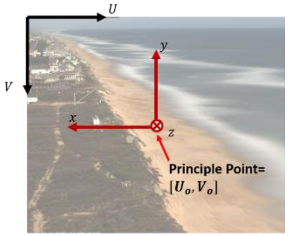
Reference back to (XYZ_w) you originally specified and translated from, plot RGB with (XYZ_w)



SUMMARY:
PIN-HOLE CAMERA FUNDAMENTAL PROCESS







- Make **grid** in World Coordinates (XYZ_w)
- Identify the corresponding Undistorted Image Coordinates (UV) for the grid points using camera projection matrix P = [K]*[R]*[T]
  - Convert World Coordinates (XYZ_w) to Camera Coordinates (XYZ_c) using *R & T (EXTRINSICS) in world units*
  - Convert and Homogenize (Scale, make unitless) Camera Coordinates (XYZ_c) to Undistorted Pixel Coordinates (UV) using K (INTRINSICS)
- Convert Undistorted Pixel Coordinates (UV) to Distorted Pixel Coordinates (UV_D) *(INTRINSICS)*
- Extract Image intensities (RGB) from UV_D coordinates
- Assign to XYZ_w and plot grid as image!

Can be a local or world coordinate system

Distortion

Grid

$$\begin{bmatrix}U_D\\V_D\\1\end{bmatrix} \leftrightarrow \begin{bmatrix}U\\V\\1\end{bmatrix} \overset{*}{=} \overbrace{\underbrace{\begin{bmatrix}-f_x&0&U_o\\0&-f_y&V_o\\0&0&1\end{bmatrix}}_{K}\underbrace{\begin{bmatrix}r_{11}&r_{12}&r_{13}\\r_{21}&r_{22}&r_{23}\\r_{31}&r_{22}&r_{33}\end{bmatrix}}_{R}\underbrace{\begin{bmatrix}1&0&0&-C_{XW}\\0&1&0&-C_{YW}\\0&0&1&-C_{ZW}\end{bmatrix}}_{T}}^{P}\begin{bmatrix}X_W\\Y_W\\Z_W\\1\end{bmatrix}$$

* Homogenization

SUMMARY:
TOOLBOX FUNCTIONS WORLD → IMAGE & IMAGE → WORLD

- From WORLD → DISTORTED IMAGE Steps:
  - Calculate your P Matrix
  - For N points, create a 4 x N, XYZ1 matrix
  - Multiply!
  - Convert to homogenous coordinates (scale your image by dividing by the last element of the output 3xN matrix)
  - OR use:
  - **[UVd,flag] = xyz2DistUV(intrinsics,extrinsics,xyz)**

(flag identifies bad points that are outside of the image plane)

- From DISTORTED IMAGE → WORLD
  - This is harder…too many unknowns (under-determined)
  - Need to know some additional information (e.g. z coordinates of the feature you identify in the image)
  - Solve the system of equations (lots of algebra)
  - OR use:

[xyz] = distUV2XYZ(intrinsics,extrinsics,UVd,knownDim,knownV)

LEARN MORE:
Image Rectification Lesson

SUMMARY

- Photogrammetry allows us to make quantitative measurements from imagery
- We can exploit our knowledge of:
  - where the camera is and where it's pointing (extrinsics),
  - the properties of the camera & lens (intrinsics),
  - geometric principles, and
  - our understanding of how cameras work (e.g. pin-hole camera model)
- We do this using Projective Mapping and the Camera Equation by calculating the Camera Projection Matrix, P

Function **imageRectifier**

solves for complete Process/Equation

Distortion

$$\begin{bmatrix} U_D \\ V_D \\ 1 \end{bmatrix} \leftrightarrow \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = {}^{*}\overbrace{\underbrace{\begin{bmatrix} -f_x & 0 & U_o \\ 0 & -f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{22} & r_{33} \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -C_{XW} \\ 0 & 1 & 0 & -C_{YW} \\ 0 & 0 & 1 & -C_{ZW} \end{bmatrix}}_{T}}^{P} \overbrace{\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}}^{\text{Grid}}$$

* Homogenization