Modelling Camera Residual Terms

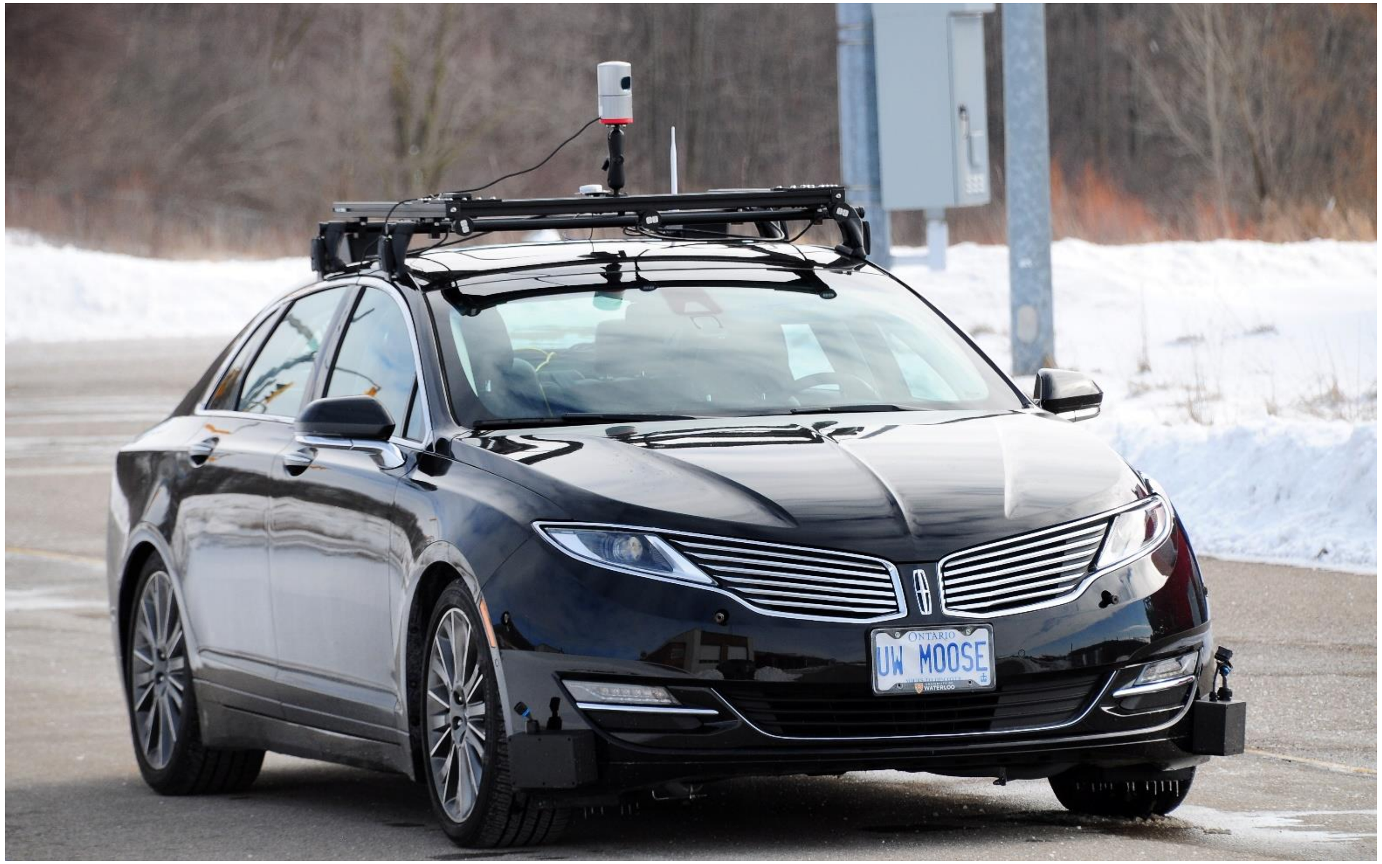
Using Reprojection Error and Photometric Error

June 26 2017

Pranav Ganti

OUTLINE

- Overview
- Geometry
- Cameras – Part 1
- Multi-view Geometry
- Reprojection Error
- Cameras – Part 2
- Photometric Error
- Application (SVO)



Overview

- Residual
  - Difference between the observed value and *estimated* value
- Cost/ Loss function is the function to be minimized
  - Generally a *function* of the residual
- Camera residuals
  - Formulation depends on indirect vs direct methods
  - A value to be minimized, which can **estimate the camera pose**.

- Euclidean Space ($\boldsymbol{R}^3$)

- Euclidean geometry describes:
  - Lines
  - Circles
  - Angles

- **Issue:** $\boxed{\infty}$
  - How do we represent points at infinity?

GEOMETRY | PROJECTIVE SPACE

- Euclidean Space + *ideal points*

- Ideal points: **points at infinity.**
  - Now, 2 lines always meet in a point!

- Projective space is derived from Euclidean space by adding line @ infinity

- Points at infinity can be described using homogeneous coordinates

- Homogeneous coordinates in $\boldsymbol{R}^n$ written as an $n+1$ vector
  - $R^2: (x,\ y,\ 1)^T,\ \ R^3: (x,\ y,\ z,\ 1)^T$
  - Ideal points: $(\boldsymbol{x}, \boldsymbol{y}, \dots, \boldsymbol{0})^{\boldsymbol{T}}$

- What about scaled points?
  - $(kx,\ ky,\ k)^T$ is an equivalence class of $\left(\frac{x}{k},\ \frac{y}{k},\ 1\right)^T$ - (we'll revisit why later!)

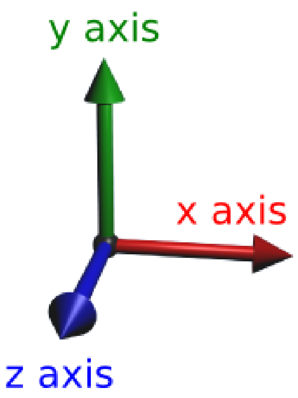
- Euclidean space can be extended to projective space using homogeneous vectors

- Euclidean Transform: **Rotation** + **Translation**

- Affine transform: **Rotation** + **Translation** + **Stretching** (linear scaling)

- For both Euclidean and Affine transforms, points at infinity remain at infinity

- What about a projective transform?



- What properties of an object are preserved?
  - Shape?
  - Angles?
  - Lengths?
  - Distances?
  - Straightness?

- Projective transformation is any mapping that preserves straight lines.

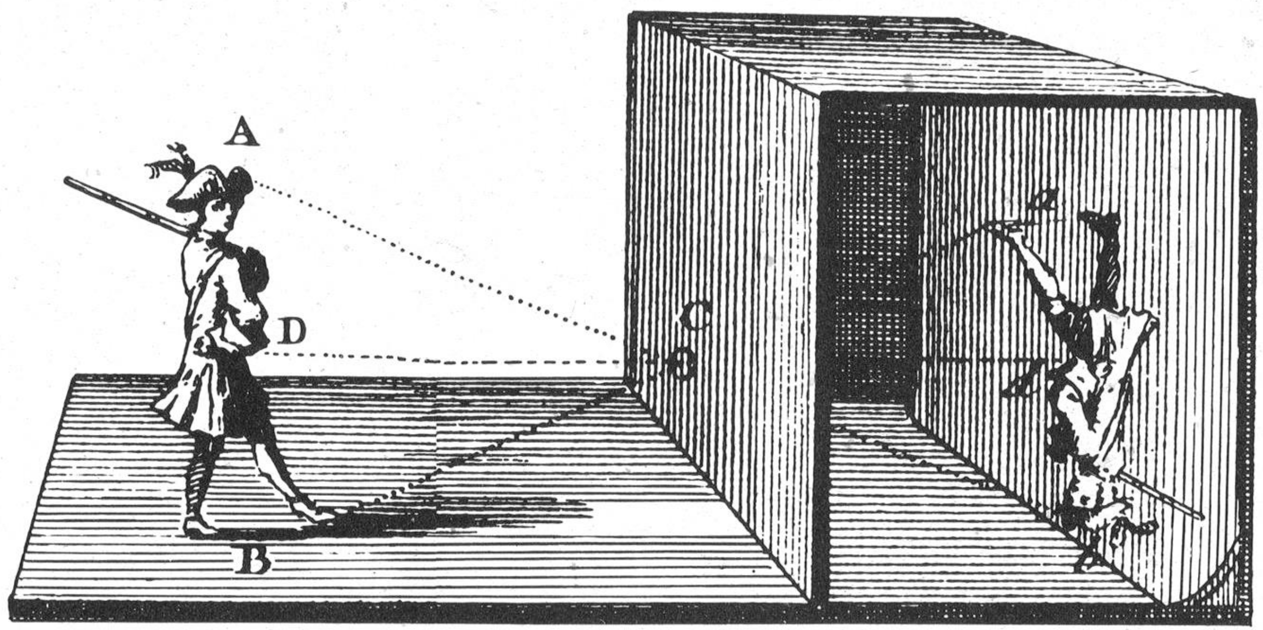
UNIVERSITY OF WATERLOO

- Projective transformation is a mapping of the homogeneous coordinates

- Ideal points are **not** preserved
  - Points at infinity are mapped to arbitrary points

- For computer vision, the projective space is convenient
  - Treat 3D space as P^3 instead of R^3
  - Images as P^2
  - Useful for practical applications – even though we know points at ∞ are our own construct

- Also known as “camera obscura”
- First type of camera
- Light passes through an opening
    - Image is reflected on the other side



- Cameras are a map between the 3D world and 2D image
  - Projection: lose 1 dimension
- Can be mapped via **central projection**
  - Ray from 3D point passes through camera center of projection (COP)
  - Intersects image plane
- If 3D structure is planar, then there is no drop in dimension

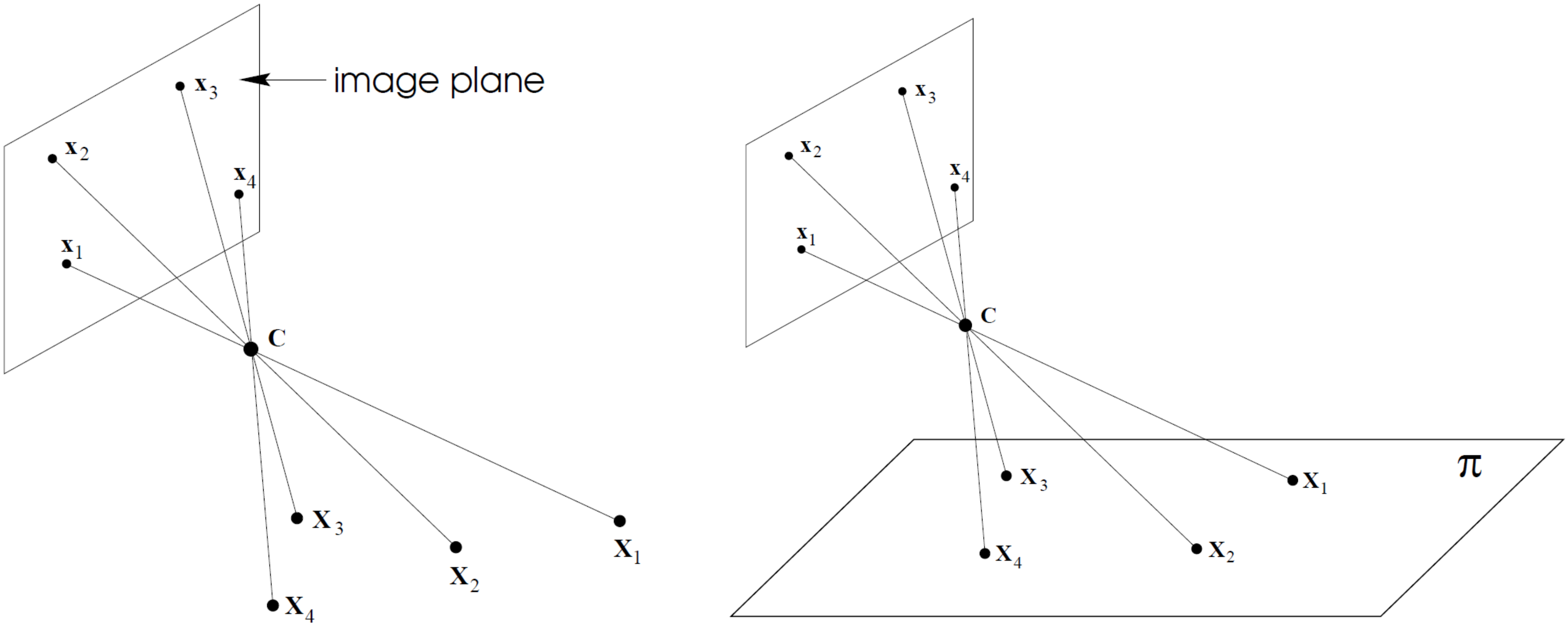
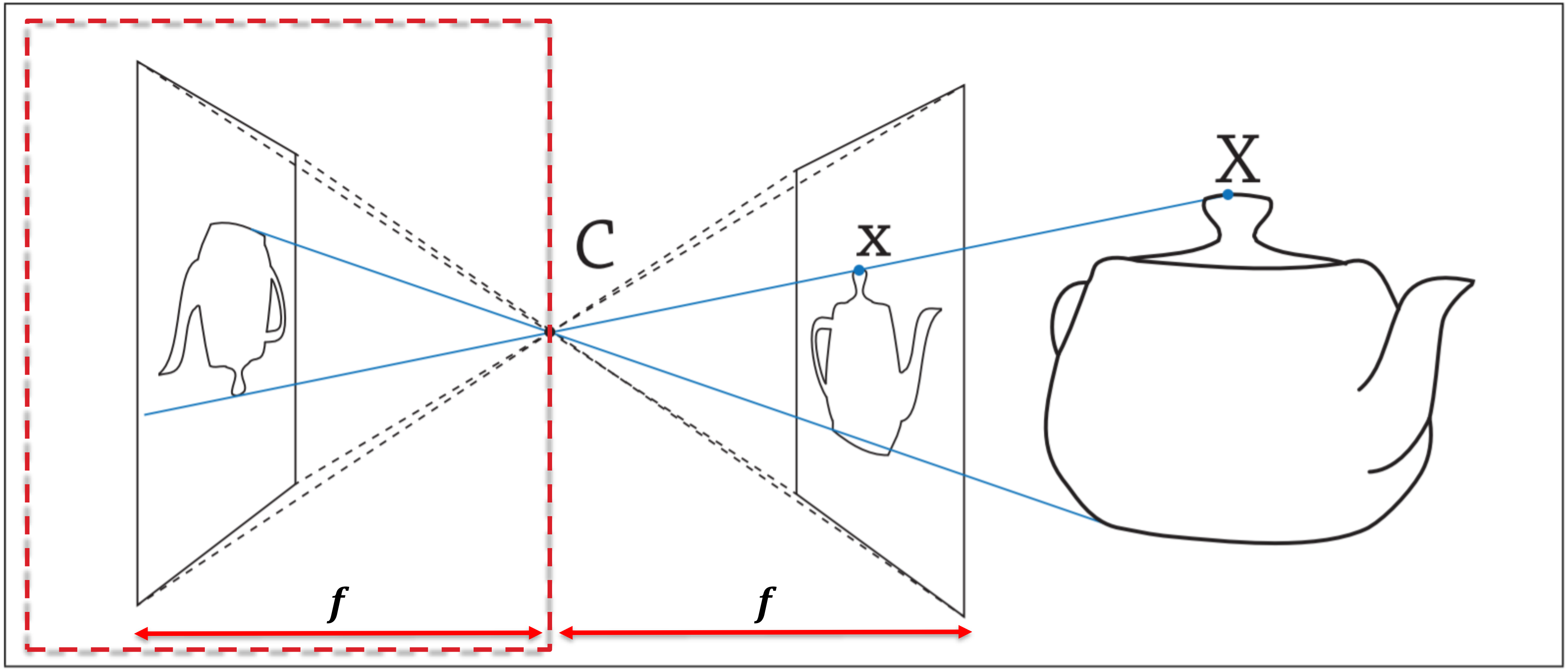


image plane
x3
x2
x4
x1
C
X3
X2
X1
X4
x3
x2
x4
x1
C
π
X3
X1
X4
X2

UNIVERSITY OF WATERLOO

- For convenience: can place image plane **in front** of COP



- In essence, central projection is just mapping $P^3 \to P^2$
  - The camera matrix P is a 3x4 matrix of rank 3

$$\boxed{(x, y, w)^T = P(X, Y, Z, T)^T}$$

- $(x, y, w)^T$ are homogeneous coordinates of image space (P^2)
- $(X, Y, Z, T)^T$ are homogeneous coordinates of 3D world (P^3)

- Ray passing through COP is a projected point in the image.
  - Therefore, **all** points on ray can be considered equal.
  - Rays are image points, and we can represent rays as homogeneous coordinates

- Need calibration to express relative Euclidean geometry between image and world.
  - With a calibrated camera, can back-project 2 points in an image
  - Can then determine angle between two rays

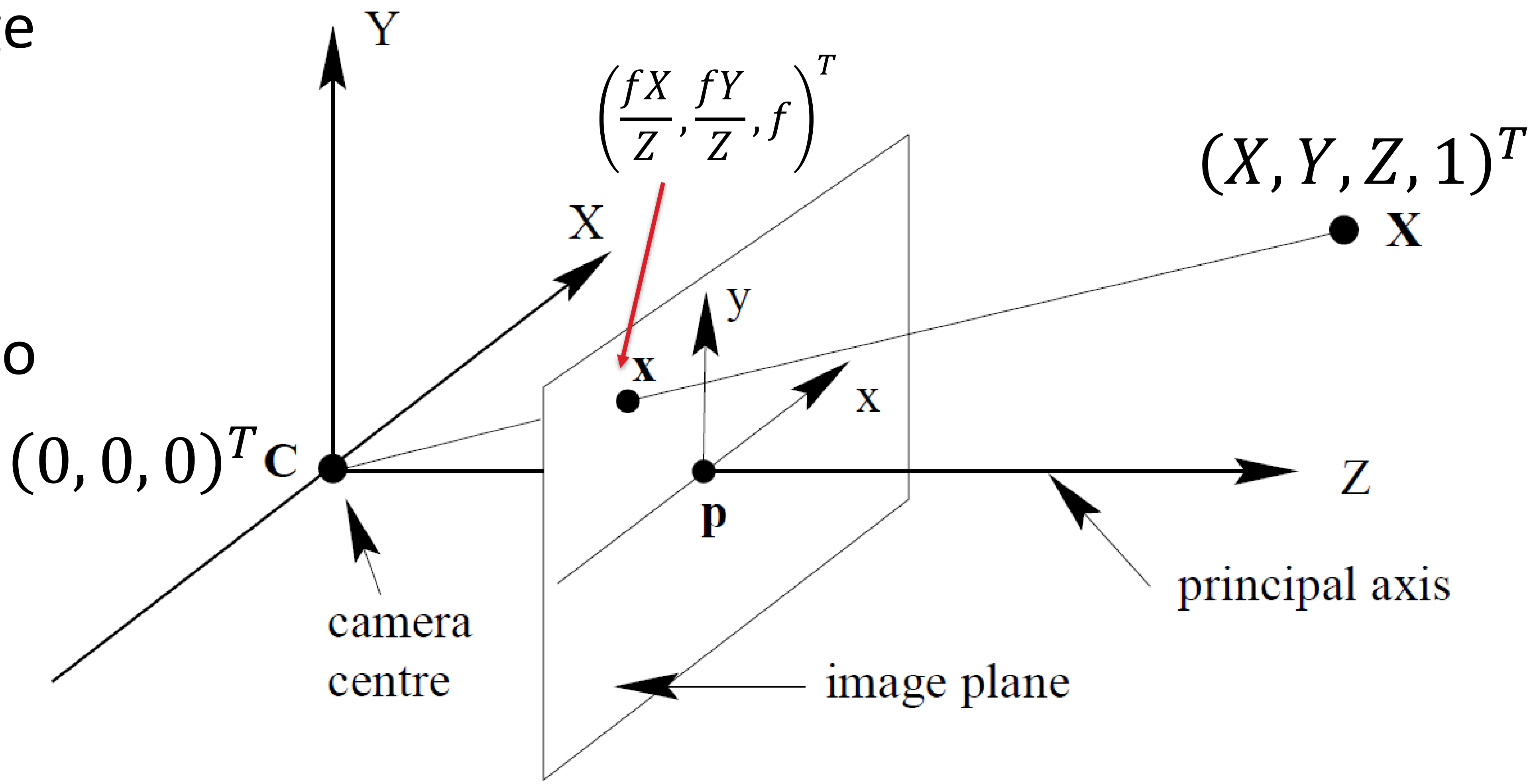
- Let's derive the camera matrix.

- Assumptions:
  - Center of projection is origin (R^3)

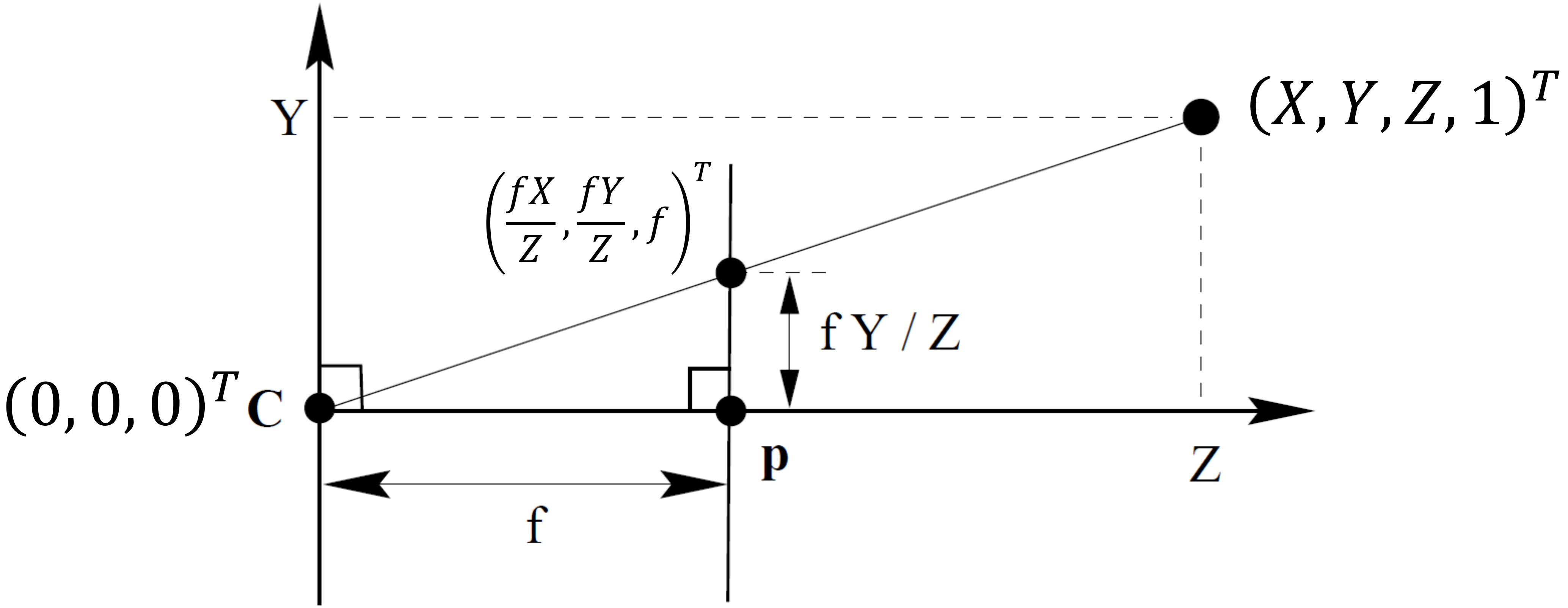
- Using pinhole camera model:
  - By similar triangles:

$$(X, Y, Z)^T \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}, f\right)$$

- Recall: image plane is located at a distance equivalent to the focal length



- Mapping from P^3 to P^2 using similar triangles



- With the Euclidean origin @ the COP:
  - Central projection just becomes a linear map b/w homogenous coordinates

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

- Can be written as: $\mathbf{x} = \mathrm{P}\mathbf{X}$

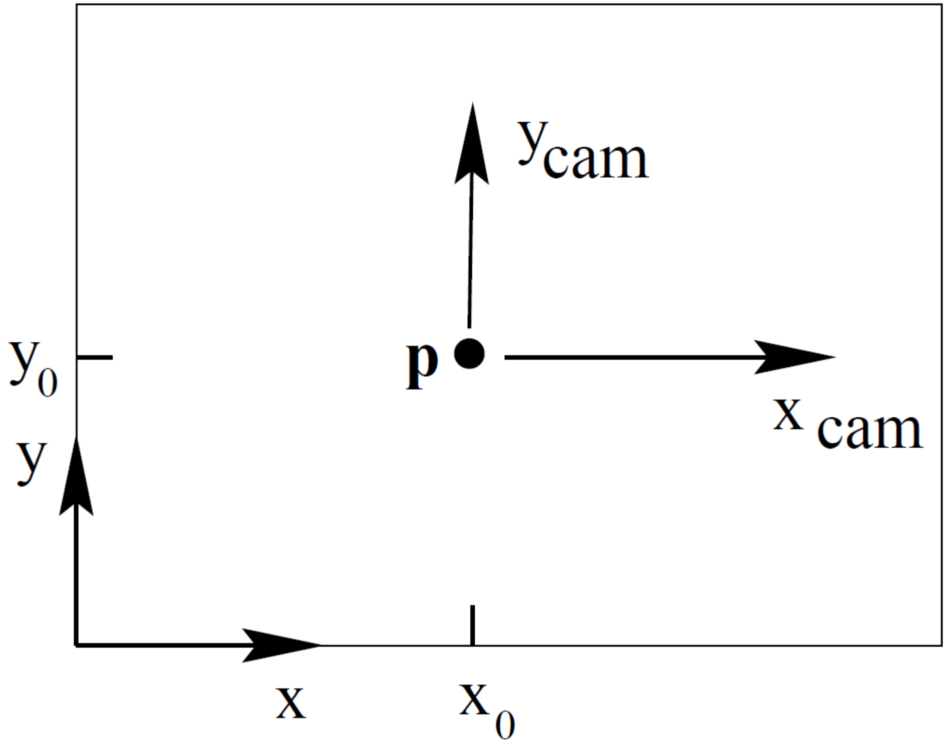
$$\mathrm{P} = \mathrm{diag}(f, f, 1)\,[\mathrm{I} \mid \mathbf{0}]$$

- The previous equation assumes image coordinates at the principal point.

- A more generic mapping is:

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^{\mathsf{T}} \mapsto (f\mathrm{X}/\mathrm{Z} + p_x, f\mathrm{Y}/\mathrm{Z} + p_y)^{\mathsf{T}}$$

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} + \mathrm{Z}p_x \\ f\mathrm{Y} + \mathrm{Z}p_y \\ \mathrm{Z} \end{pmatrix} = \boxed{\begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix}} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}$$



- K is the camera calibration matrix
  - Can also add a *skew* parameter

$$\mathrm{K} = \begin{bmatrix} f & s & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$

- Can then express

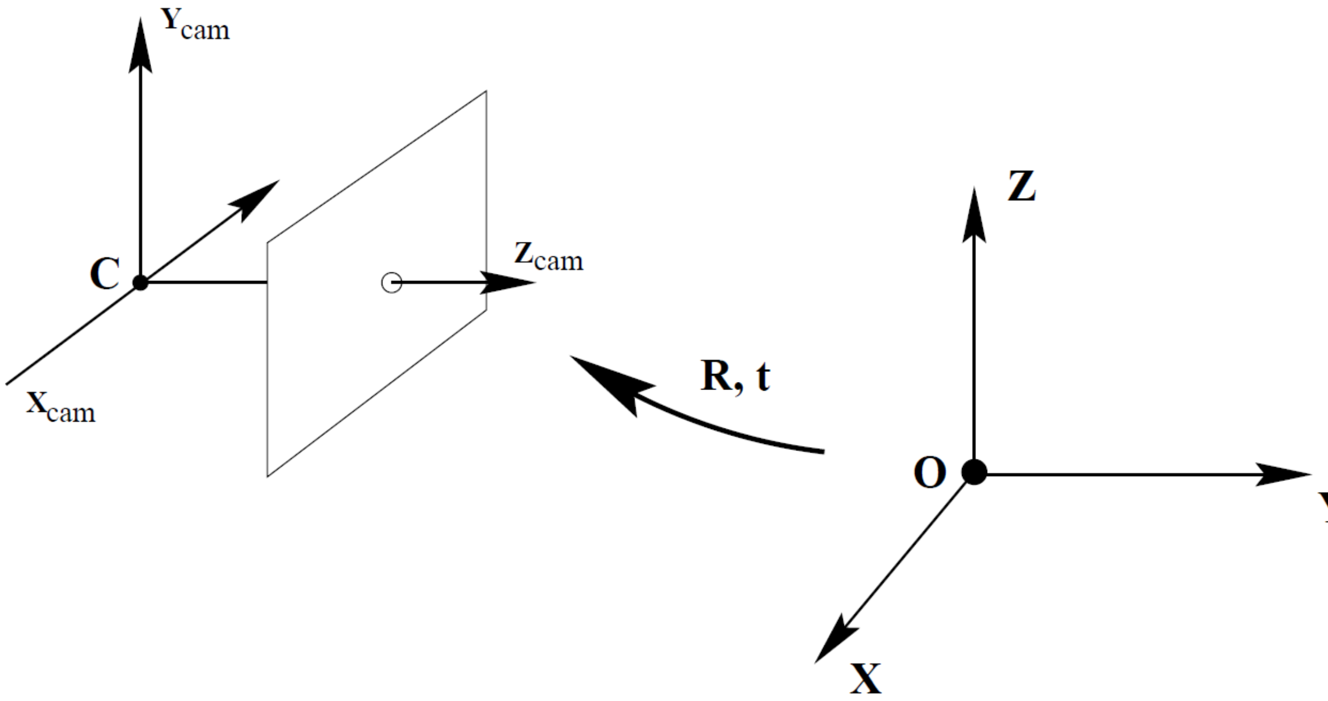
$$\mathbf{x} = \mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathbf{X}_{\mathrm{cam}}$$

where $\boldsymbol{x}_{cam}$ is $(X, Y, Z, 1)^T$, expressed in a coordinate frame at the COP.

- The world coordinate frame is not always expressed at COP.
  - Example: a moving camera!

- Coordinate frames related through a rotation and translation

$$\widetilde{\mathrm{X}}_{\mathrm{cam}} = \mathrm{R}(\widetilde{\mathbf{X}} - \widetilde{\mathrm{C}})$$



- The equation can now be expressed as:

$$\mathbf{X}_{\text{cam}} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\widetilde{\mathrm{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\widetilde{\mathrm{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}$$

$$\mathbf{x} = \mathrm{KR}[\mathrm{I} \mid -\widetilde{\mathbf{C}}]\mathbf{X}$$

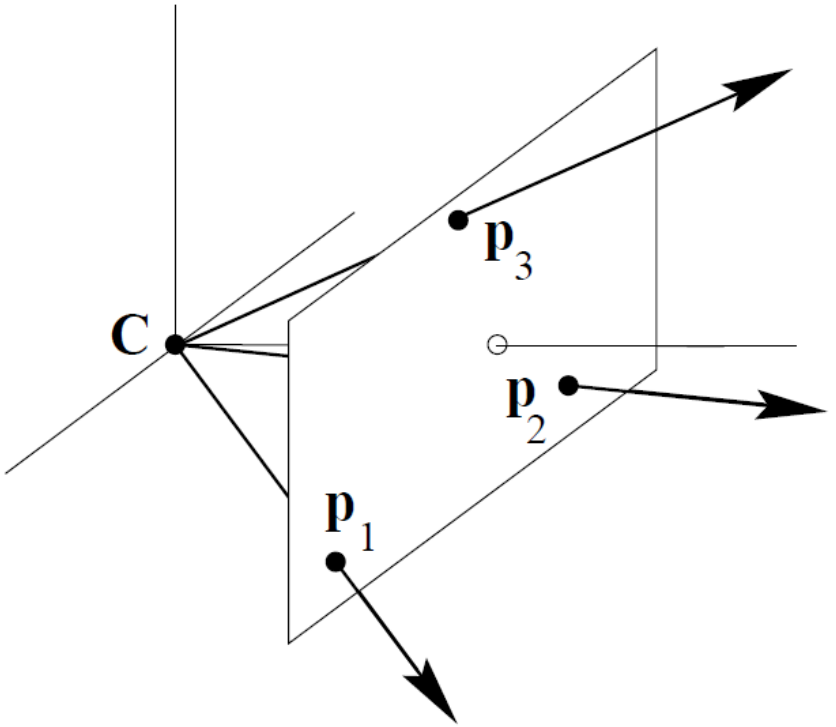
$$\boxed{\mathrm{P} = \mathrm{K}[\mathrm{R} \mid \mathbf{t}]} \qquad \mathbf{t} = -\mathrm{R}\widetilde{\mathrm{C}}$$

- Forward projection: maps a point in 3D space to an image point
  - $x = PX$

- Back projection: from a point x in an image, we can determine the set of points that map to this point.
  - Ray in space passing through the space

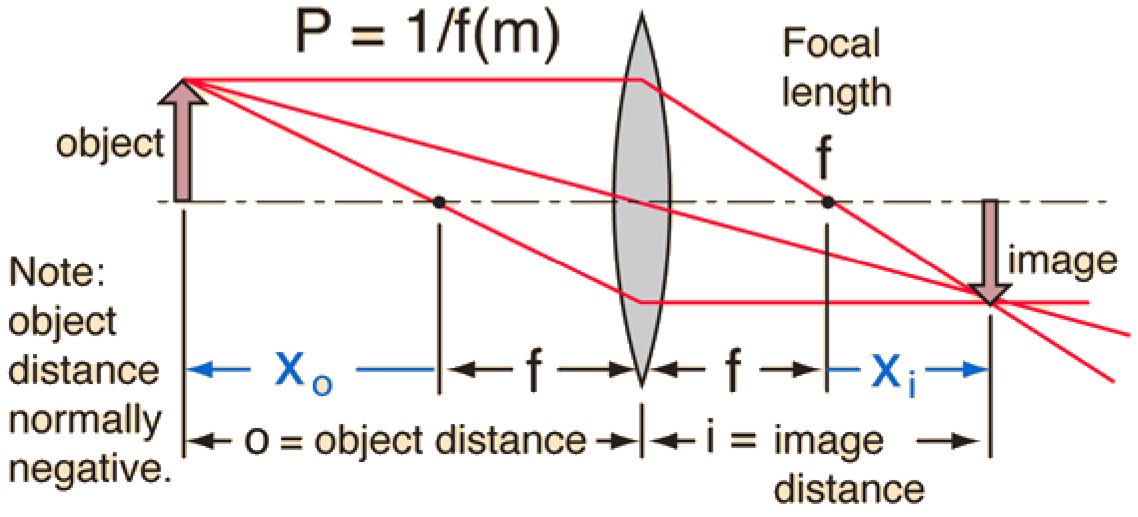
- How can we obtain the back projection?

- Null space of C is the camera center
- We know 2 points on each ray:
  - COP ($PC = 0$)
  - Image point (P^+x), $P^+ = P^T(PP^T)^{-1}$
- Why is P^+x the second point?
  - It projects to x!
  - $P(P^+x) = Ix = x$

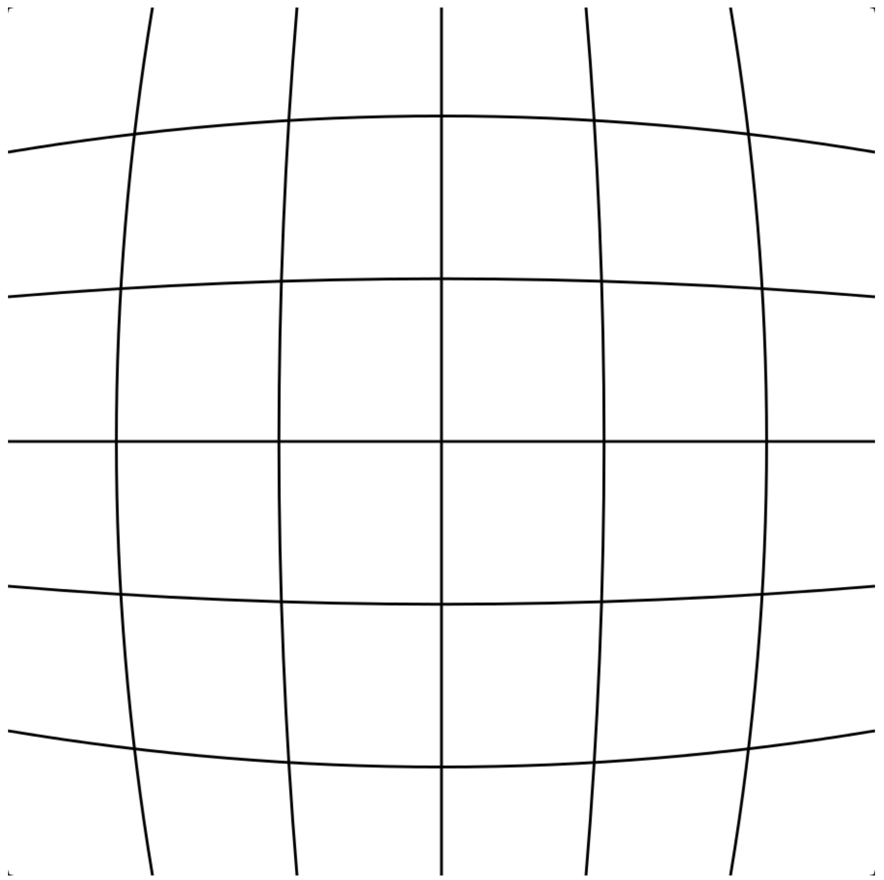


- The ray is then the line connecting these two points.

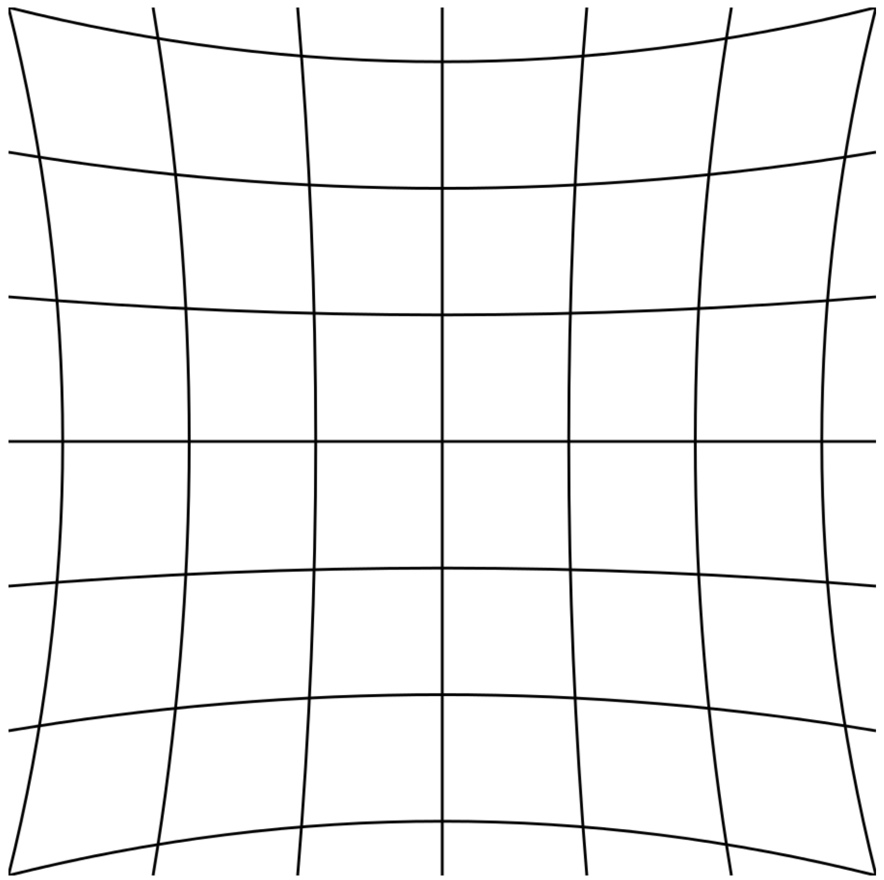
- Pinhole camera is ideal
    - Not a true representation of a camera
- Need to correct for distortions
    - Want images *as if* we were using a pinhole camera
- Distortion can be radial or tangential



Barrel Distortion



Pincushion Distortion



UNIVERSITY OF WATERLOO

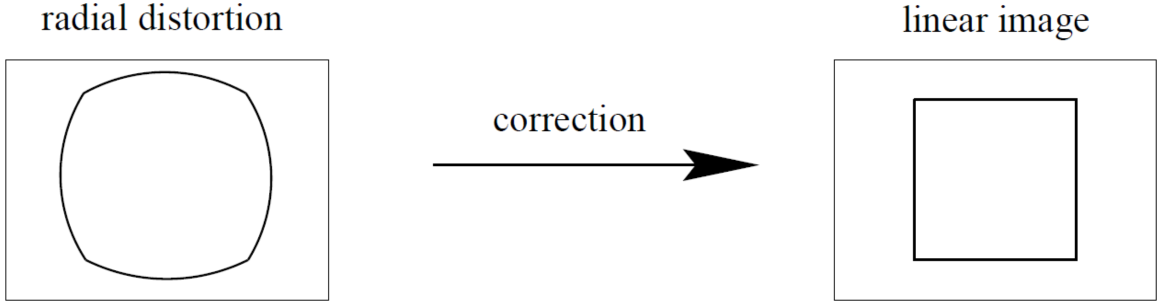
- Lens distortion occurs during initial projection onto image plane

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$$

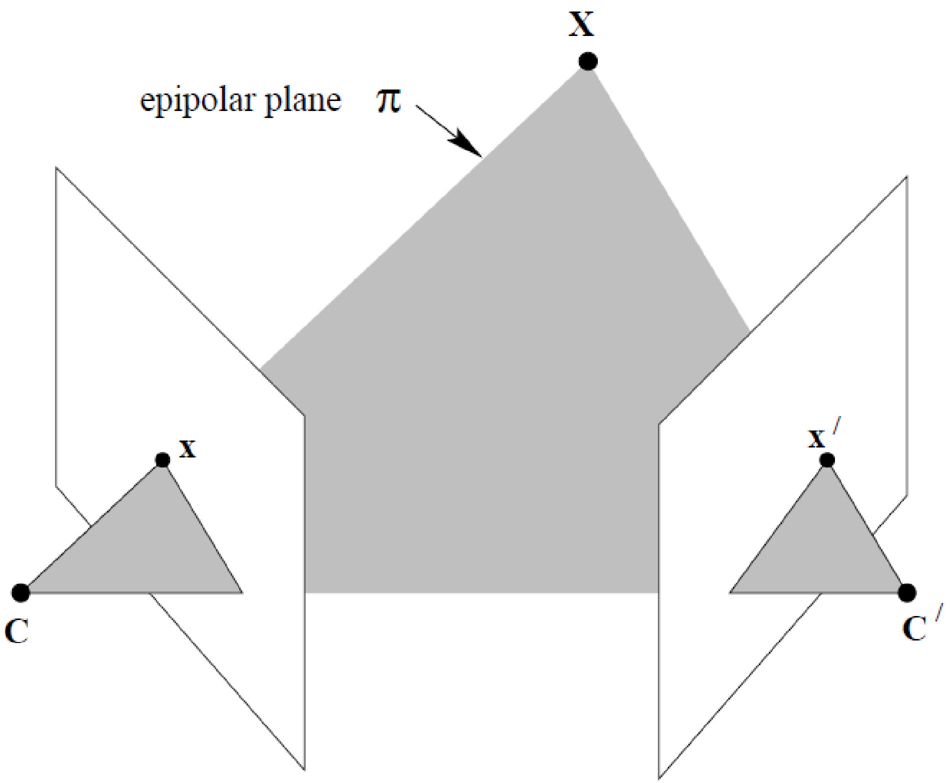
- $\tilde{x}, \ \tilde{y}$ are ideal, x_d, y_d are actual
- $\tilde{r}$ is Euclidean distance
- $L(\tilde{r})$ is the distortion factor.
  - Can be solved for through calibration

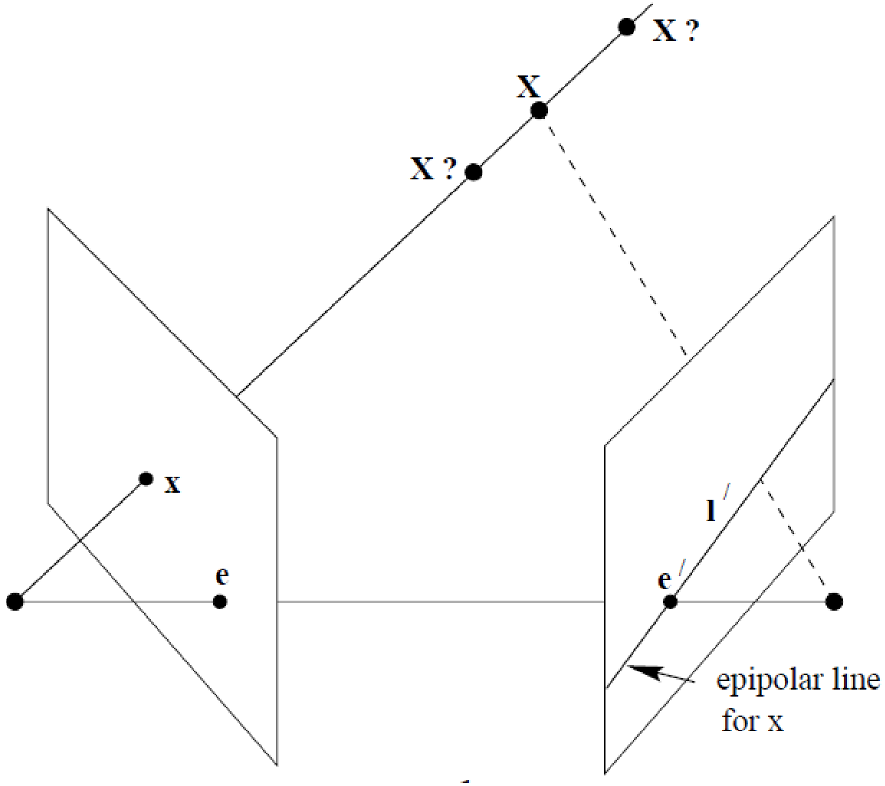
a

b

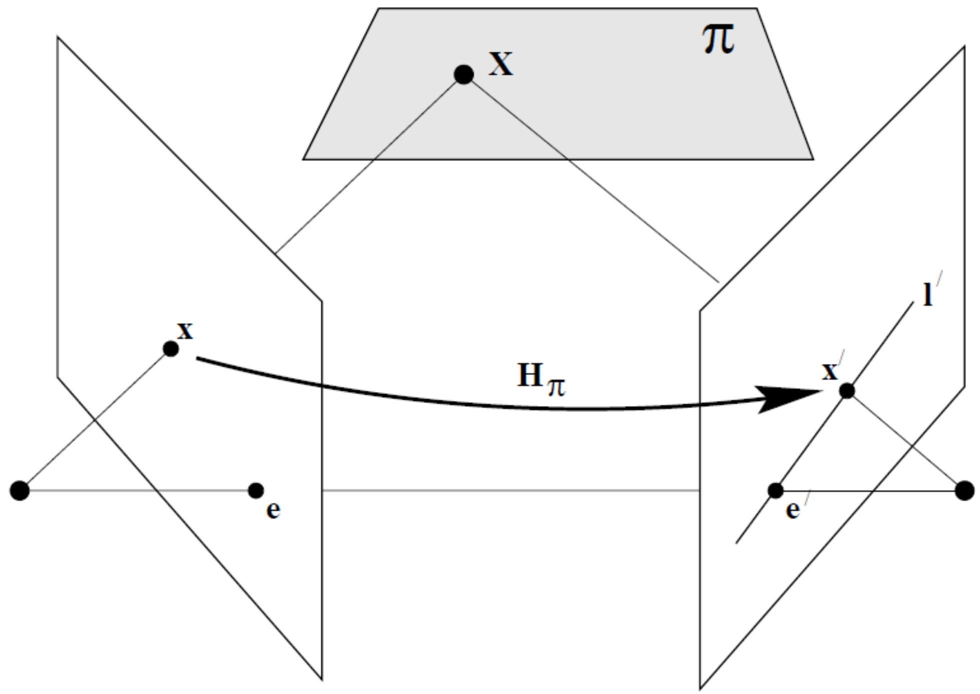


- Motivation: to search for corresponding points in stereo matching
- Baseline: Line joining camera centers
- Epipole: point of intersection b/w baseline and image plane
- Epipolar line: intersection of an epipolar plane with the image plane
- Epipolar plane: plane containing the baseline


X
epipolar plane π
x
x'
C
C'

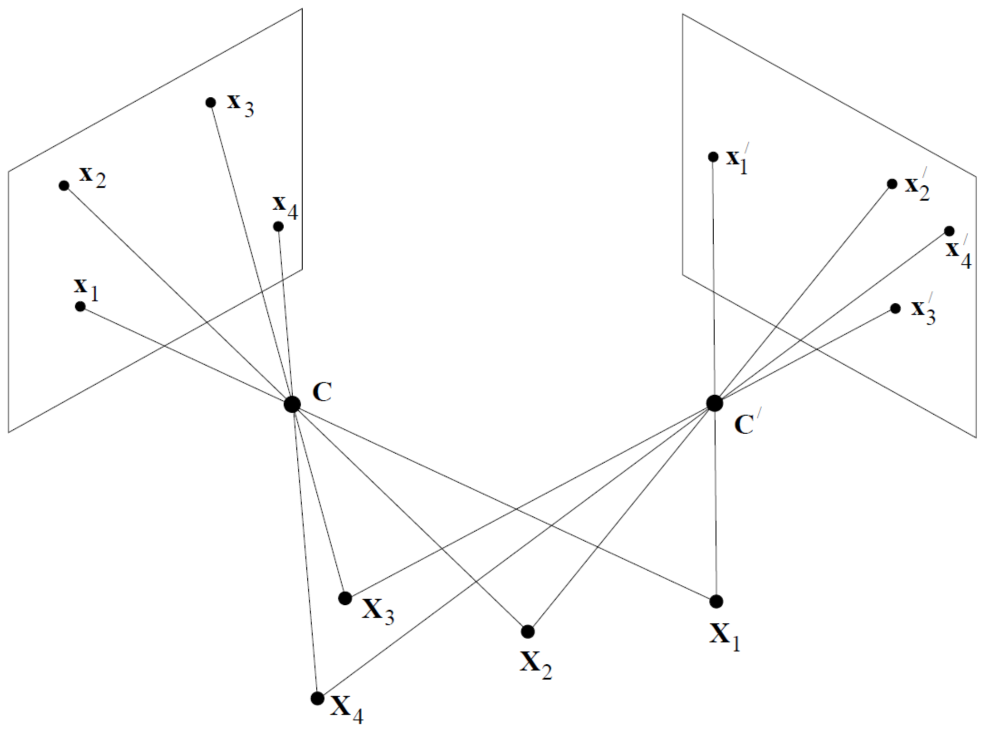

X ?
X
X ?
x
e
l'
e'
epipolar line
for x

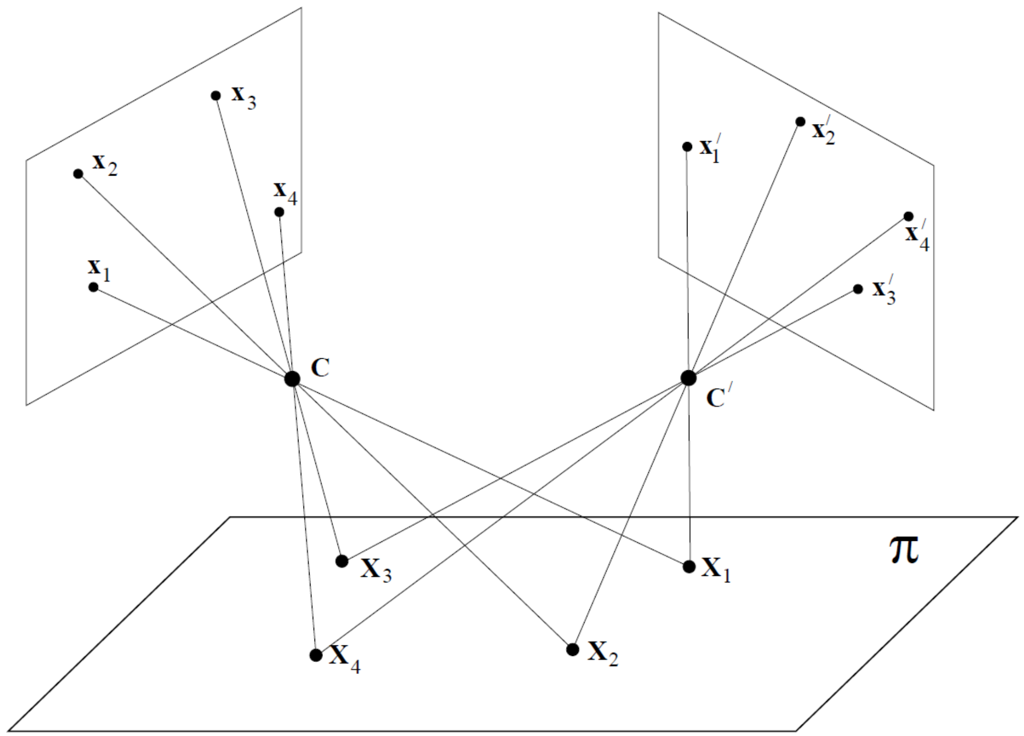
- Algebraic representation of epipolar geometry
- F represents the mapping from $P^2 \rightarrow P$, through the epipolar lines.
- Two steps:
  - Map point x to x'
  - Obtain l' from joining x' to e'



- Properties:
  - Correspondence: $x'^T F x = 0$
  - Transpose: If F is the matrix for camera P, F^T is the corresponding fundamental matrix for camera P'
  - Epipolar lines: $l' = Fx, l = F^T x'$
  - $Fe = 0, e'^T F = 0$

- Methods to solve: 7 point algorithm, 8 point algorithm, RANSAC…

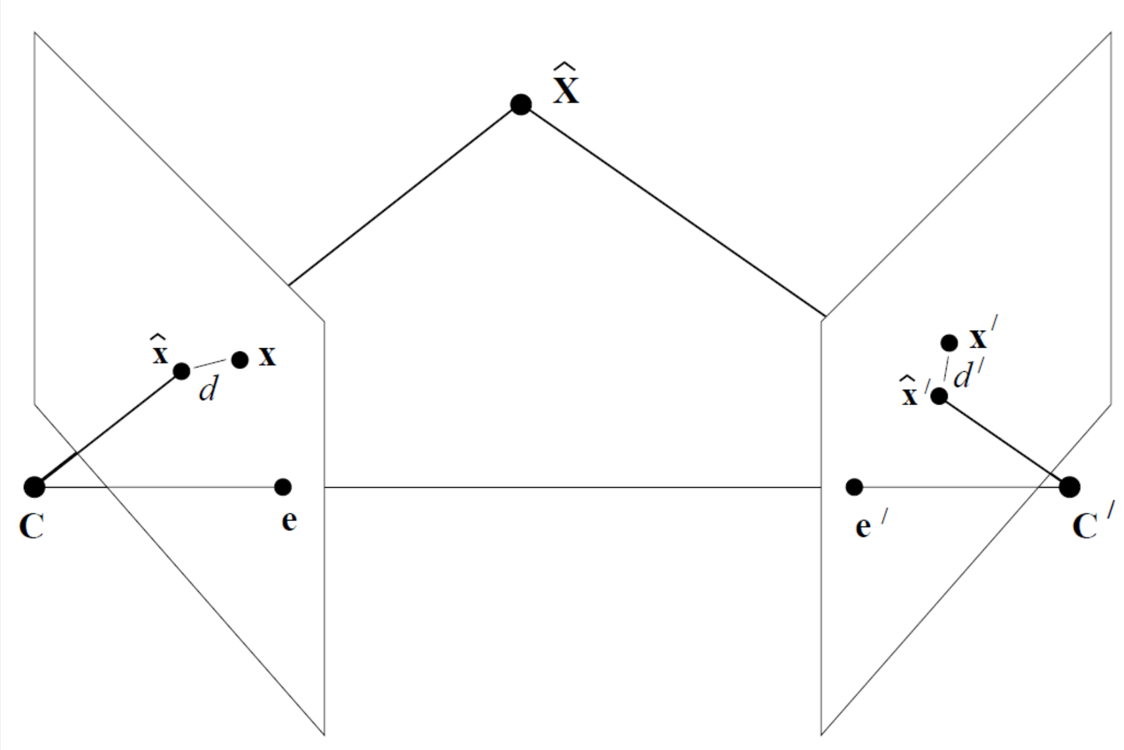

x3
x2
x4
x1
C
x1'
x2'
x4'
x3'
C'
X3
X2
X1
X4


x3
x2
x4
x1
C
x1'
x2'
x4'
x3'
C'
π
X3
X1
X4
X2

Reprojection Error

- Summed squared distance between *projections* of X, and measured image points.
  - Euclidean distance
  - In 2 images

$$\sum_i d(\mathbf{x_i}, \hat{\mathbf{x_i}})^2 + d(\mathbf{x_i}', \hat{\mathbf{x_i}}')^2$$

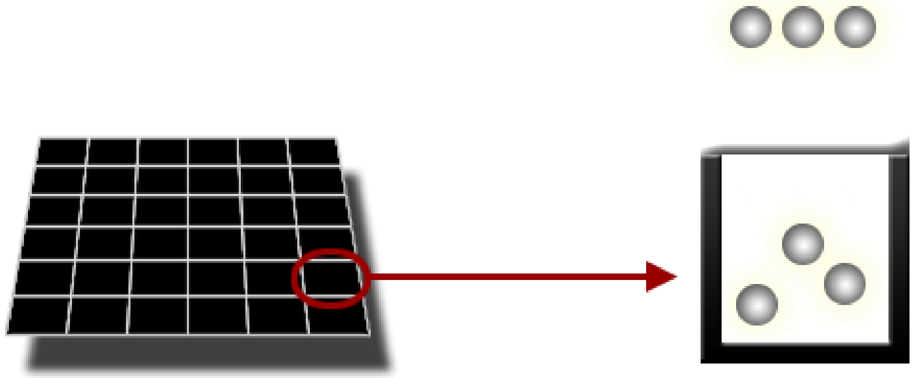


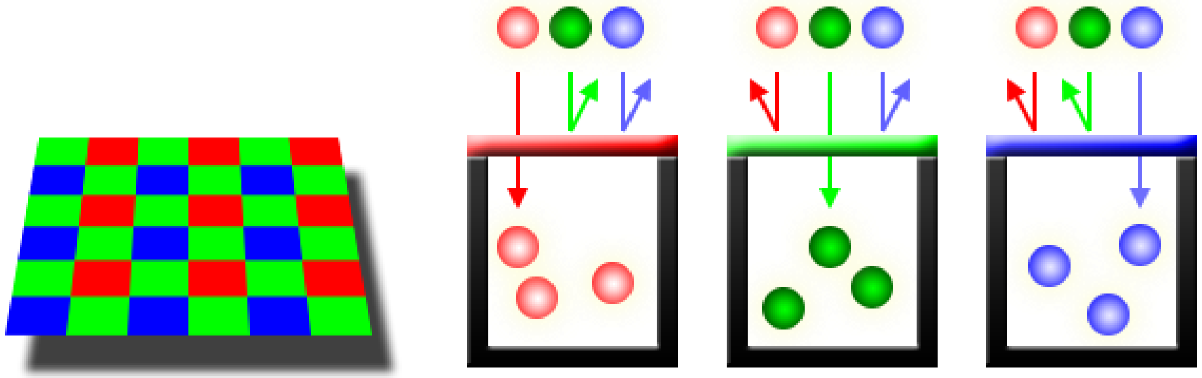
- Fundamental matrix
  - MLE of F (assuming Gaussian noise) minimizes reprojection error
  - $\hat{x}, \widehat{x'}$ are ideal points, and obtained from $\hat{x} = PX$.
    - Both P and X can be modified to minimize this error.
      - Recall: $\mathrm{P} = \mathrm{K}[\mathrm{R} \mid \mathbf{t}]$, and R, t represent the camera pose in the world frame!

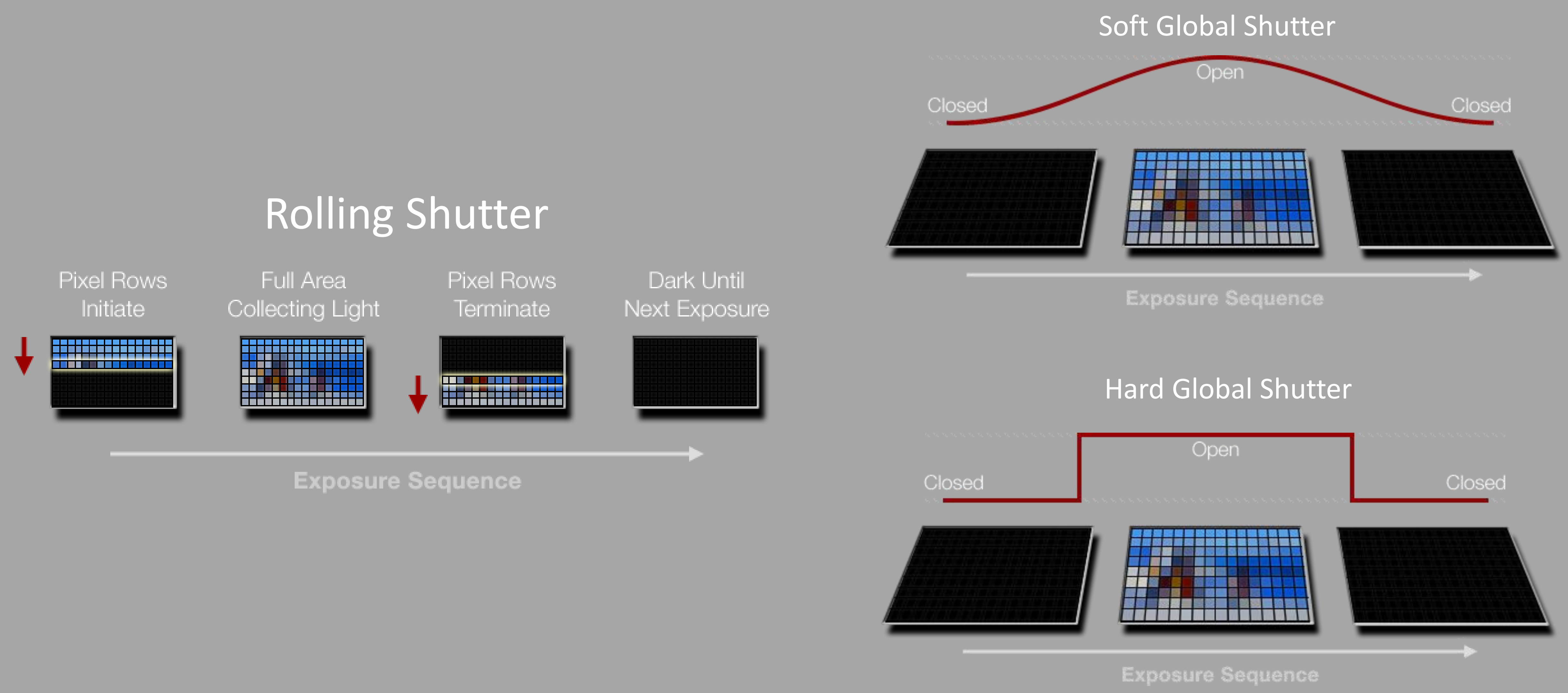
- Bundle adjustment
  - Similar, except the intrinsic parameters can also be modified.

- Camera sensors consist of photosites
  - Quantifies amount of light collected
  - The digitized information is a pixel

- CCD (charge-coupled device), CMOS (complementary metal-oxide semiconductor)






Soft Global Shutter
Open
Closed
Closed
Exposure Sequence
Rolling Shutter
Pixel Rows Initiate
Full Area Collecting Light
Pixel Rows Terminate
Dark Until Next Exposure
Exposure Sequence
Hard Global Shutter
Open
Closed
Closed
Exposure Sequence

- The resulting information from the image capture is an **intensity image.**

- Allows for use of the **entire** image, as opposed to just keypoints.
  - Becomes dense, so some direct methods only use patches of interest

- Intensity image is defined as:
  - Ω is image domain
  - Recall previously, images were $R^3 \rightarrow R^2$

$$\boxed{I_k : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}}$$

- Notation (from SVO)
  - I_{k-1}, I_k: intensity images
  - $T_{k,k-1}$: frame transform
  - u: image coordinate
  - p: 3D point
  - d_u: depth
  - $\pi: R^3 \rightarrow R^2$: camera projection model
    - π^{-1}: inverse
  - k: camera frame of reference, or timestep k
  - ξ: twist coordinates, se(3)

Relationships

$$\mathbf{u} = \pi({}_k\mathbf{p})$$

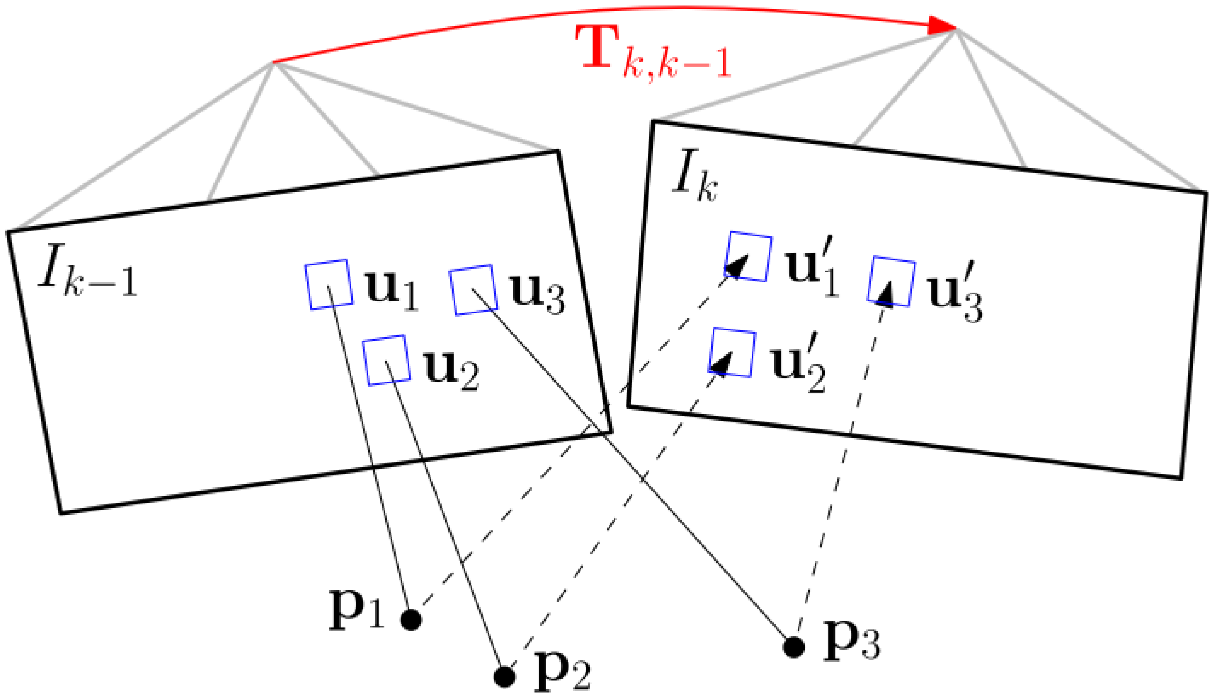
$${}_k\mathbf{p} = \pi^{-1}(\mathbf{u}, d_{\mathbf{u}})$$

$$\mathbf{T}_{k,w} \in SE(3)$$

$$\mathbf{T}_{k,k-1} = \mathbf{T}_{k,w} \cdot \mathbf{T}_{k-1,w}^{-1}$$

$$\mathbf{T}(\xi) = \exp(\hat{\xi})$$

- Photometric error: intensity difference between pixels observing the same point in 2 scenes.



- Intensity residual can be computed by:
  - Back-projecting a 2D point from the previous image.
  - Reprojecting it into the current camera view.

$$\delta I(\mathbf{T},\mathbf{u}) = I_k\Big(\pi\big(\mathbf{T}\cdot\pi^{-1}(\mathbf{u},d_{\mathbf{u}})\big)\Big) - I_{k-1}(\mathbf{u}) \quad \forall\ \mathbf{u}\in\bar{\mathcal{R}}$$

- Looking to minimize negative log-likelihood between camera poses, using intensity residual.

$$\mathbf{T}_{k,k-1} = \arg\min_{\mathbf{T}_{k,k-1}} \frac{1}{2}\sum_{i\in\bar{\mathcal{R}}} \|\ \delta\mathbf{I}(\mathbf{T}_{k,k-1},\mathbf{u}_i)\ \|^2$$

- Intensity residuals are normally distributed

- The equation is nonlinear in $T_{k,k-1}$, can be solved via the Gauss-Newton algorithm
  - Incremental update: $T(\xi)$
  - $\hat{T}_{k,k-1}$ is an estimate of the relative transformation
  - $\xi \in \boldsymbol{se}(3)$

$$\boxed{\delta \mathbf{I}(\xi, \mathbf{u}_i) = \mathbf{I}_k\Big(\pi\big(\hat{\mathbf{T}}_{k,k-1} \cdot \mathbf{p}_i\big)\Big) - \mathbf{I}_{k-1}\Big(\pi\big(\mathbf{T}(\xi) \cdot \mathbf{p}_i\big)\Big)}$$

$$\mathbf{p}_i = \pi^{-1}(\mathbf{u}_i, d_{\mathbf{u}_i})$$

Camera Residual Terms

- Reprojection error:
  - Binary factor between feature and camera pose

$$\sum_i d(\mathbf{x_i}, \hat{\mathbf{x}}_\mathbf{i})^2 + d(\mathbf{x_i}', \hat{\mathbf{x}}_\mathbf{i}')^2$$

- Photometric error:
  - Unary factor (at least in SVO)
    - No feature locations to estimate position of.

$$\mathbf{T}_{k,k-1} = \arg\min_{\mathbf{T}_{k,k-1}} \frac{1}{2} \sum_{i \in \bar{\mathcal{R}}} \| \delta \mathbf{I}(\mathbf{T}_{k,k-1}, \mathbf{u}_i) \|^2$$

- Applications
  - Reprojection Error: Indirect VO/ SLAM
  - Photometric Error: Direct VO/SLAM
- SVO (Semi-direct Visual Odometry) takes advantage of both.
  - Initial pose estimate using direct
  - Further refinement using indirect methods on keyframes

- Indirect methods extract features, match them, and then recover camera pose (+structure) using epipolar geometry and reprojection error
  - Pros: Robust matches even with high inter-image motion
  - Cons: Extraction, matching, correspondence…can be quite costly

- Direct methods estimate camera pose (+structure) directly from intensity values and image gradients.
  - Pros: Can use all information in image. More robust to motion blur, defocus. Can outperform indirect methods.
  - Cons: Can also be costly, due to density.

UNIVERSITY OF WATERLOO

- SVO steps:
    1. Initial pose estimate through minimizing photometric error.
    2. Relaxation through feature alignment.
    3. Further refinement through reprojection error.
- In parallel:
    1. Determine keyframes, extract features
    2. Estimate depth through projection model

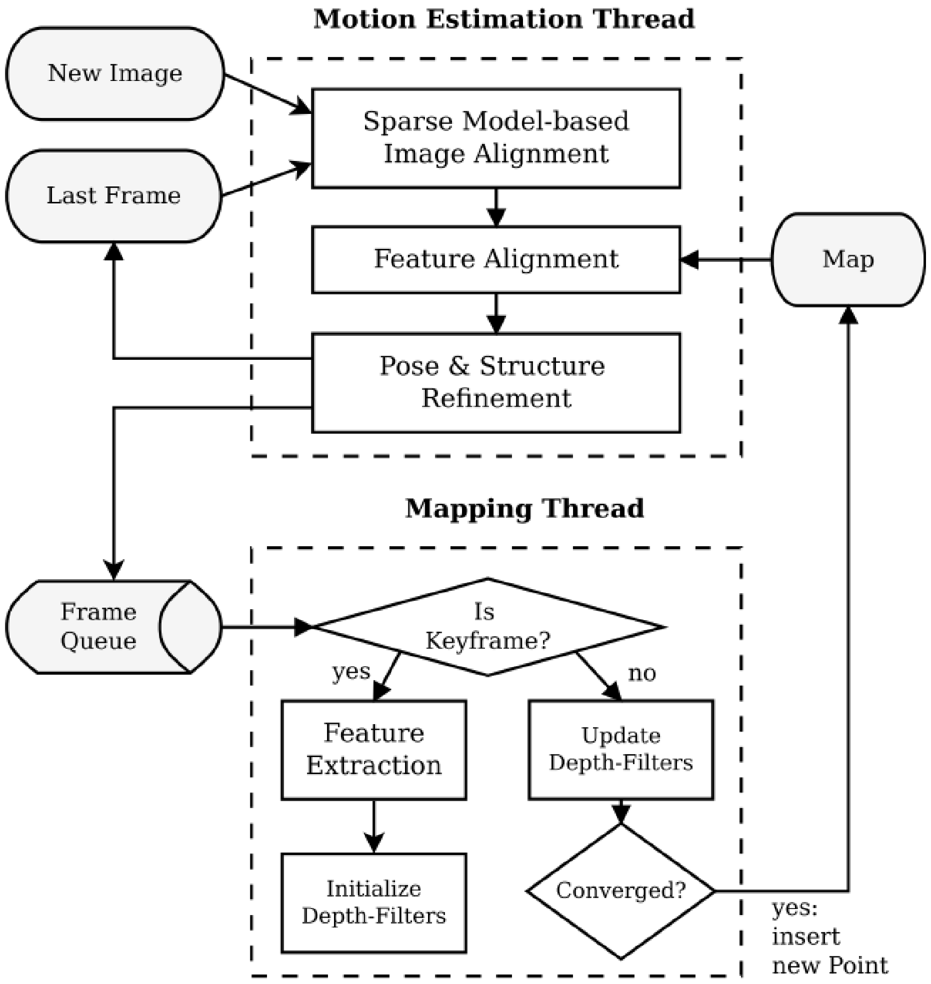


Fig. 1: Tracking and mapping pipeline

APPLICATION | SVO

- Results:

- SVO 2.0:

References

- R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge university press, 2003.
- C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 15–22, IEEE, 2014.
- C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “Svo: Semidirect visual odometry for monocular and multicamera systems,” IEEE Transactions on Robotics, 2016.

Image References

- https://i.stack.imgur.com/SitTF.png Retrieved June 24 2017
- http://www.red.com/learn/red-101/global-rolling-shutter Retrieved June 25 2017
- https://en.wikipedia.org/wiki/Errors_and_residuals
- https://en.wikipedia.org/wiki/Euclidean_space
- https://en.wikipedia.org/wiki/Distortion_(optics)
- http://www.cc.gatech.edu/~afb/classes/CS4495-Fall2013/slides/CS4495-05-CameraModel.pdf
- https://en.wikipedia.org/wiki/Camera_obscura