Review of Second Half of Course

Dr. Gerhard Roth

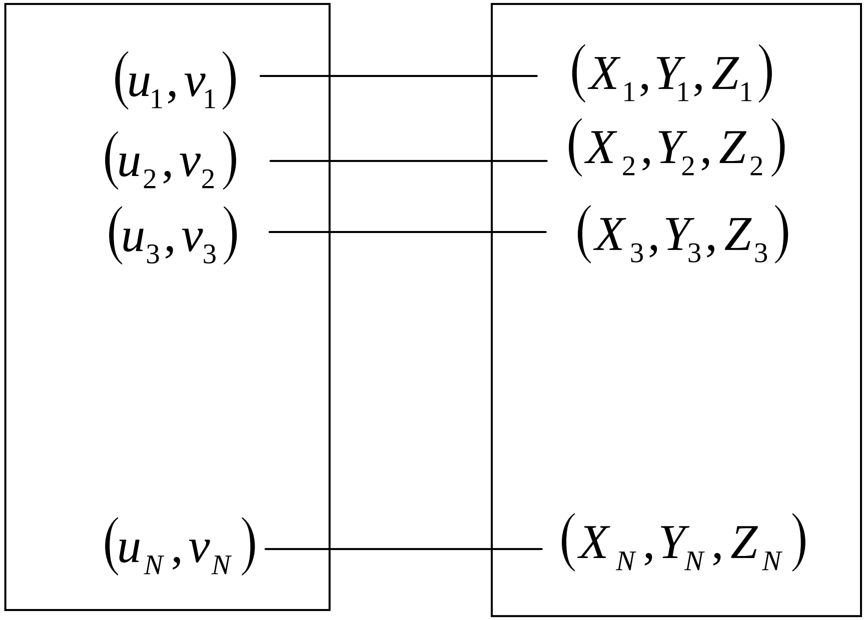
Problem statement

- Write projection equations linking known co-ordinates of a set of 3-D points and their projections and solve for camera parameters
- Use a calibration pattern with known 3d geometry (usually a box)
- Given a set of one or more images of the calibration pattern estimate
    - Intrinsic camera parameters
        - (depend only on camera characteristics)
    - Extrinsic camera parameters
        - (depend only on position camera)

Estimating camera parameters

- Projection matrix

Calibration pattern

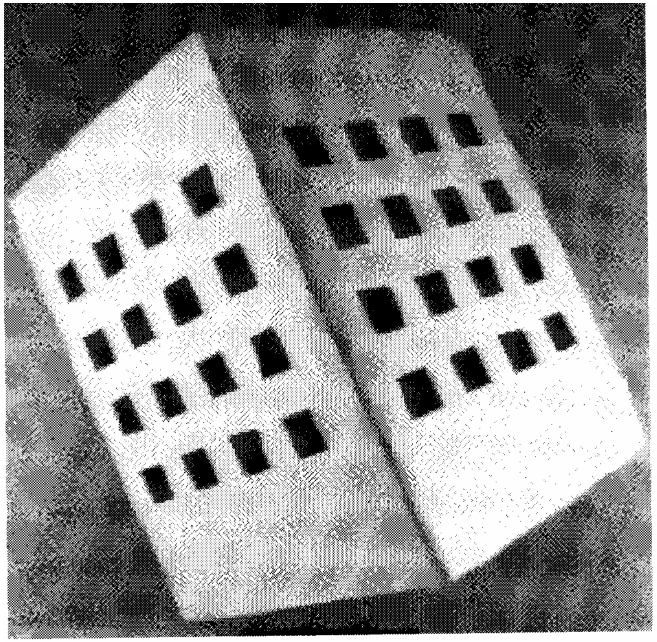


(u_1, v_1) — (X_1, Y_1, Z_1)

(u_2, v_2) — (X_2, Y_2, Z_2)

(u_3, v_3) — (X_3, Y_3, Z_3)

(u_N, v_N) — (X_N, Y_N, Z_N)



Camera parameters

- Intrinsic parameters (K matrix)
  - There are 5 intrinsic parameters
  - Focal length f
  - Pixel size in x and y directions, sx and sy
  - Principal point ox, oy
- But they are not independent
  - Focal length fx = f / sx and fy = f / sy
  - Principal point ox, oy
  - This makes four intrinsic parameters
- Extrinsic parameters [R| T]
  - Rotation matrix and translation vector of camera
  - Relations camera position to a known frame
  - [R|T] are the intrinsic parameters
- Projection matrix
  - 3 by 4 matrix P = [R | T] K is called projection matrix

Projection Equations

Projective Space

- Add fourth coordinate
  - $P_w = (X_w, Y_w, Z_w, 1)^T$
- Define $(u,v,w)^T$ such that
  - $u/w = x_{im}$, $v/w = y_{im}$

$$\begin{pmatrix} x_{im} \\ y_{im} \end{pmatrix} = \begin{pmatrix} u/w \\ v/w \end{pmatrix} \Leftarrow \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{M}_{\mathbf{int}}\mathbf{M}_{\mathbf{ext}} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

3x4 Matrix E_{ext}

- Only extrinsic parameters
- World to camera

$$\mathbf{M}_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^T & T_x \\ \mathbf{R}_2^T & T_y \\ \mathbf{R}_3^T & T_z \end{bmatrix}$$

3x3 Matrix E_{int}

- Only intrinsic parameters
- Camera to frame

$$\mathbf{M}_{\text{int}} = \begin{bmatrix} -f_x & 0 & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Simple Matrix Product! Projective Matrix

$M = M_{int} M_{ext}$

- $(Xw, Yw, Zw)^T$ -> $(xim, yim)^T$
- Linear Transform from projective space to projective plane
- M defined up to a scale factor – 11 independent entries

Rotation Matrices

- Are special 3 by matrices R

$$RR^T = R^T R = I \qquad \det(R) = 1$$

- Rows and columns of R are mutually orthogonal unit vectors
- Each row and column has unit norm
- Dot product of each row with others is zero
- Dot product of each column with others is zero

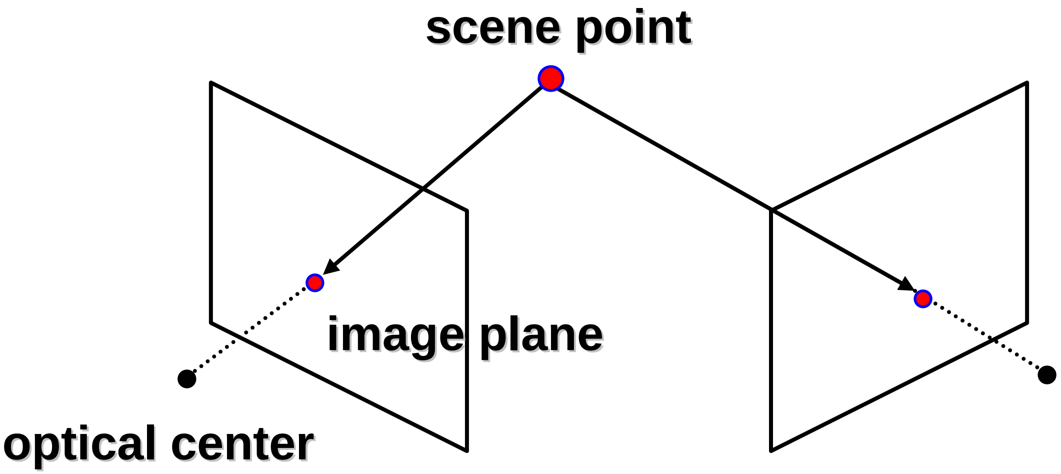
Summary of Projection matrix approach

- Compute the projection matrix
  - Then use characteristics of rotation matrix to find the other parameters
  - Simpler mathematically than the direct approach
- There are other calibration methods
  - Zhang approach uses flat plane
  - Improved by Chang Shu and Mark Fiala
- But all
  - Have some known targets with known 3D geometry
  - Take a number of images of these targets
  - Do some calculations from the measurements
  - Output the camera calibration parameters

Stereo

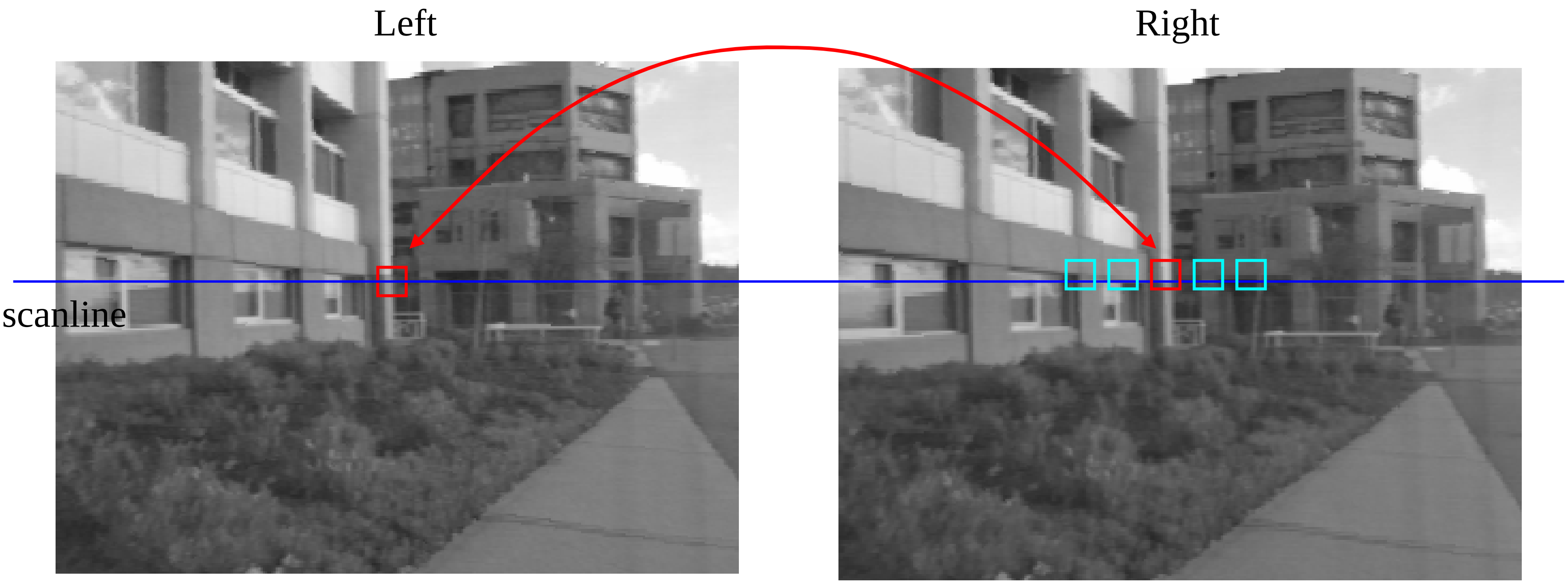
- Stereo
  - Ability to infer information on the 3-D structure and distance of a scene from two or more images taken from different viewpoints
  - Humans use only two eyes/images (try thumb trick)
- Two important problems in stereo
  - Correspondence and reconstruction
- Correspondence
  - What parts of left and right images are parts of same object?
- Reconstruction
  - Given correspondences in left and right images, and possibly information on stereo geometry, compute the 3D location and structure of the observed objects

Stereo

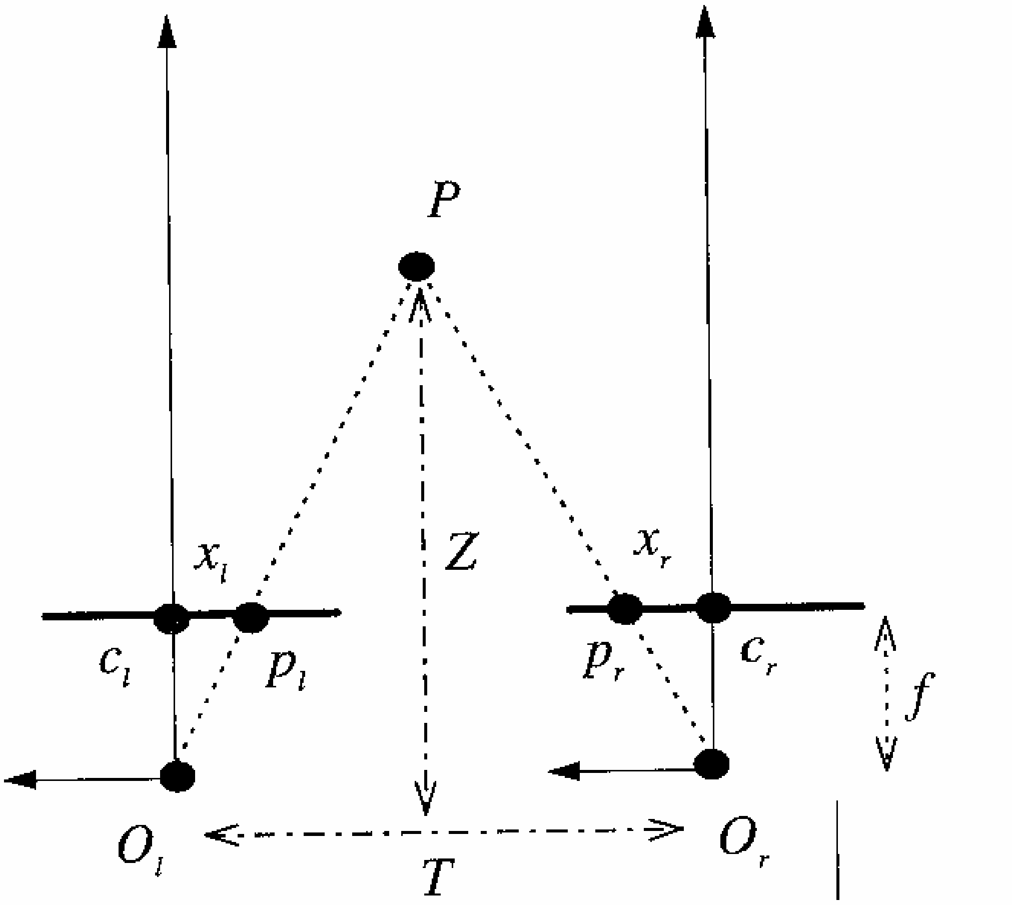


Simple Stereo System

- Left and right image planes are coplanar
  - Represented by I_L and I_R
- So this means that all matching features are on the same horizontal line
  - So we can think of this as a 2D situation

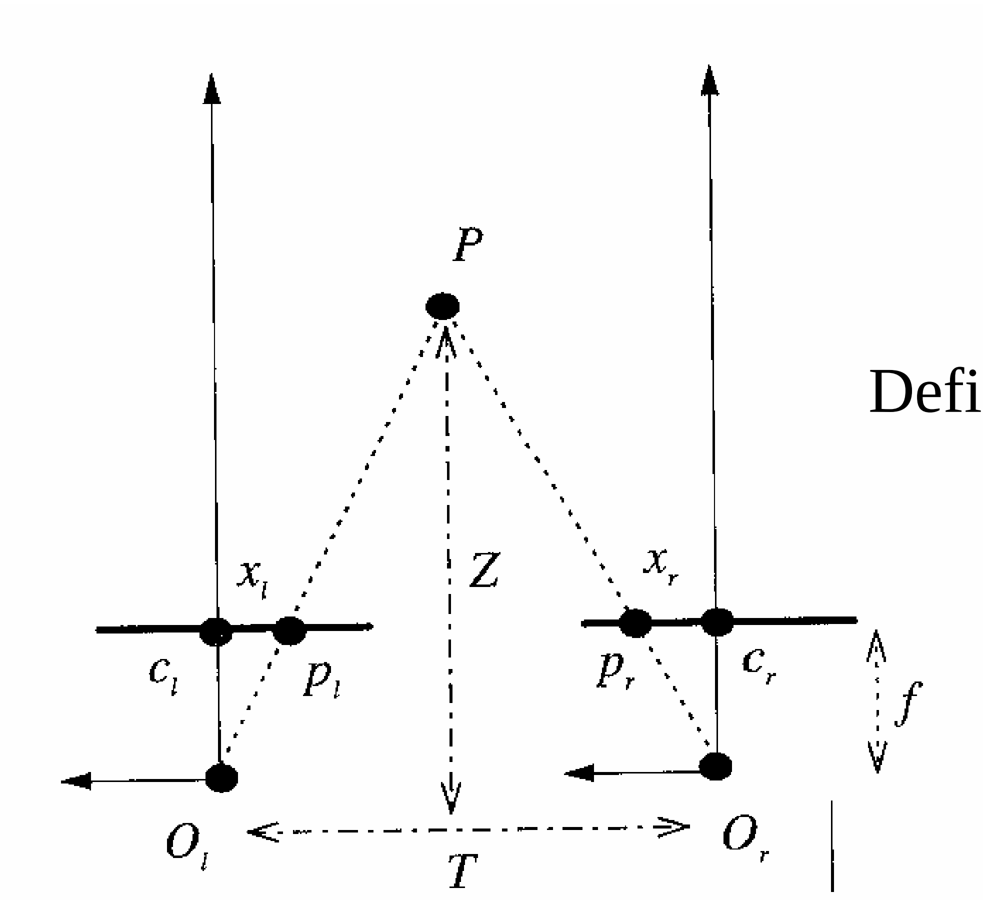


Basic Stereo Derivations



Derive expression for Z as a function of x_1, x_2, f and B

Basic Stereo Derivations



Similar triangles
(P_L,P,P_R) and (O_L,P,O_R)

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

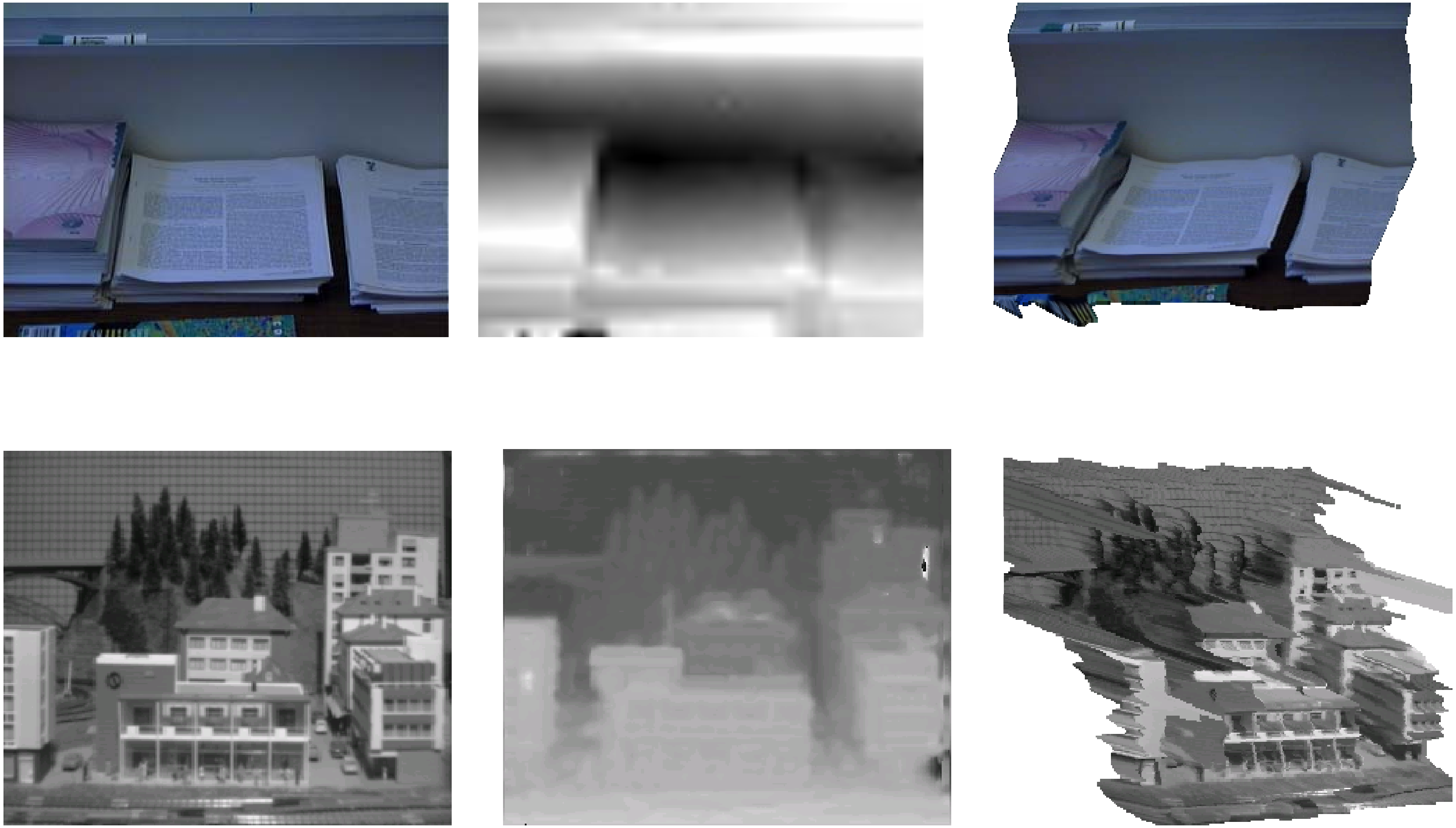
Define the disparity: $d = x_1 - x_2$

$$Z = f\,\frac{T}{d}$$

Disparity Map

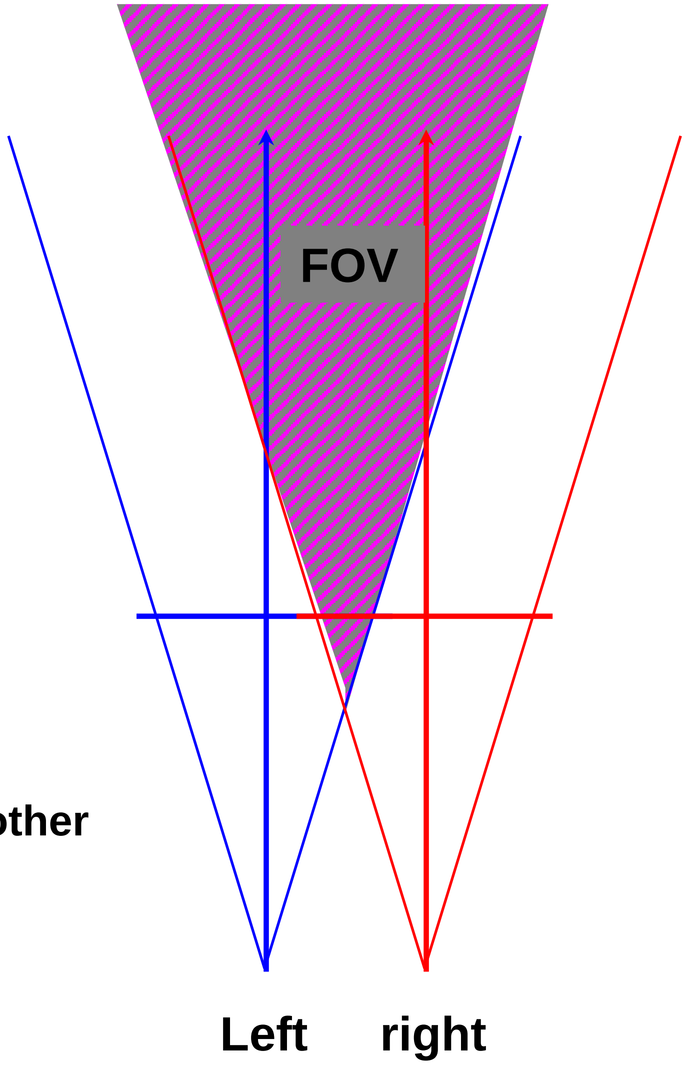
- D = $||x_1 - x_2||$ measures the distance between corresponding points in two images
  - Normally disparity is stated as number of pixels
  - Clearly a particular simple stereo configuration has a maximum and minimum possible disparity
- Depth is inversely proportional to disparity
- If we compute the disparity for the entire images then we have a disparity map
- Display it as an image
  - Bright points have highest disparity (closest)
  - Dark points have lowest disparity (farthest)
- Disparity map is a 3D image

Disparity Map



Characteristics of Simple Stereo

- FOV is field of view of cameras
  - Overlap of the two cameras
- Baseline is a system parameter
  - It is also a tradeoff
- If B is the Baseline
  - Depth Error $\propto$ 1/B
- PROS of Longer baseline
  - **better depth estimation**
- CONS
  - **smaller common FOV**
  - **Correspondence harder due to increased chance of occlusion**
  - **Occlusion means that a feature is visible in one image but not in another because something occludes it**



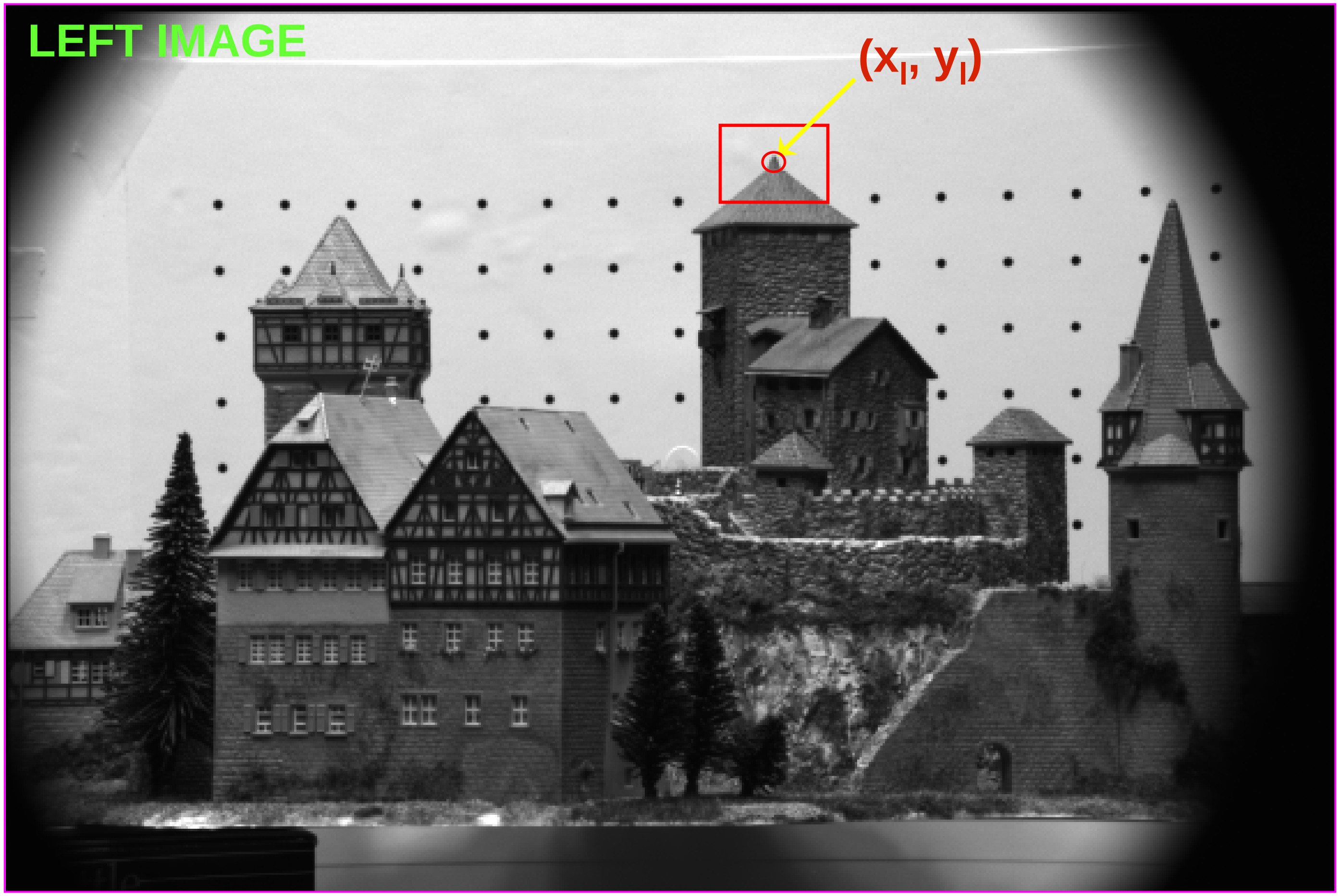
Problem Definition

- Correspondence problem
  - What parts of left and right image are projections of the same point in the 3D scene
- Simple stereo configuration
  - Corresponding points are on same horizontal line
- Assumptions
  - Most scene points are visible from both regions
  - Corresponding image regions are similar
- Search problem
  - Given scene element on left image search for
  - What parts of left and right images are parts of same object?
- Two decisions
  - Which element to match
  - Which similarity measure to adopt

Correspondence and Feature Methods

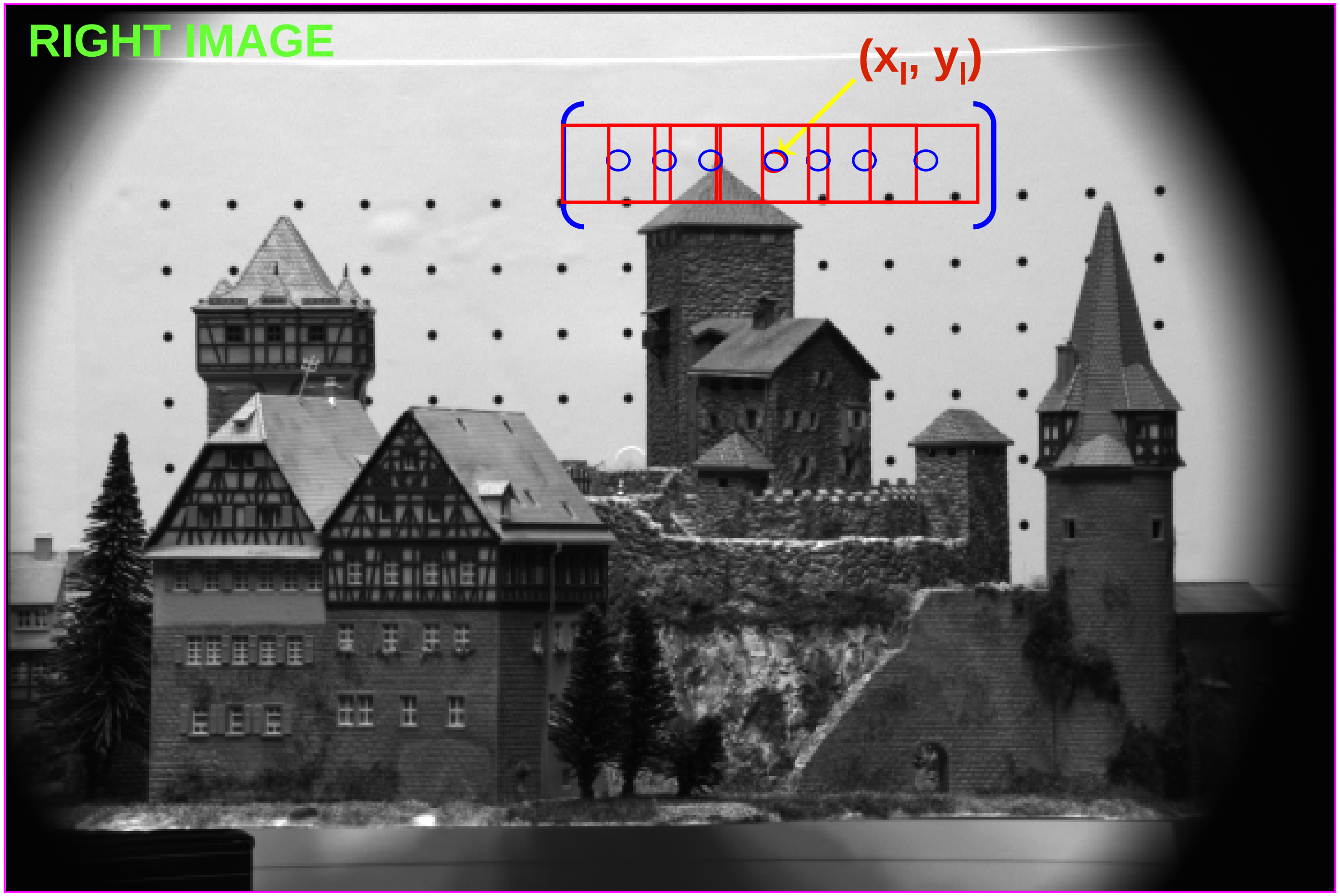
- Two basic approaches
- Correlation methods
  - Apply to all image points
  - Elements are image windows of fixed size
  - Similarity measure is correlation between two windows in the left and right images
  - Corresponding element is window that maximizes similarity criterion within a search window
- Feature methods
  - Apply only to a sparse set of feature points
  - Narrows down feasible matches by using constraints
  - Geometric constraints
  - Analytic constraints – uniqueness and continuity

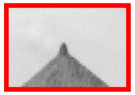
Correlation Approach



For Each point (x_l, y_l) in the left image, define a window centered at the point

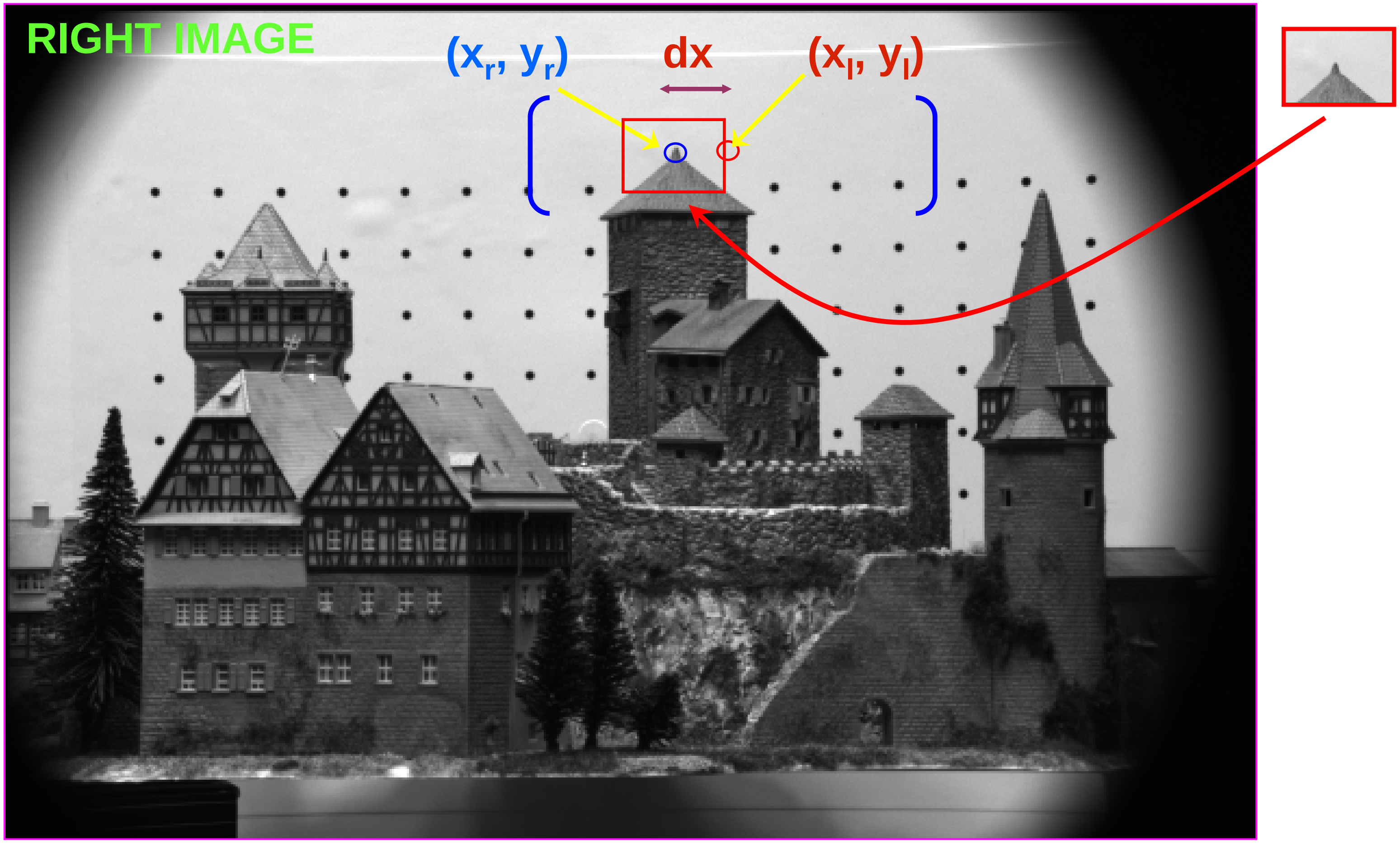
Correlation Approach





… search its corresponding point within a search region in the right image

Correlation Approach



… the disparity (dx, dy) is the displacement when the correlation is maximum

Correlation Approach

Elements to be matched

- Image window of fixed size centered at each pixel in the left image

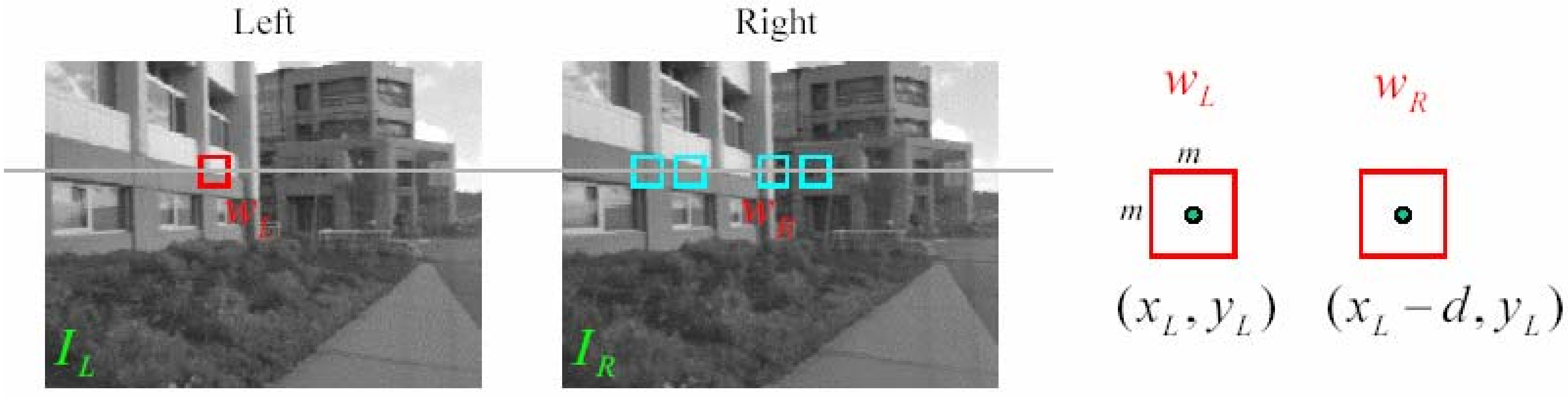
Similarity criterion

- A measure of similarity between windows in the two images
- The corresponding element is given by window that maximizes the similarity criterion within a search region

Search regions

- Theoretically, search region can be reduced to a 1-D segment, along the horizontal line (in future we will use term epipolar line), and within the disparity range.
- In practice, search a slightly larger region due to errors in calibration

Sum of Squared Differences (SSD)



w_L and w_R are corresponding m by m windows of pixels.

We define the window function :

$$W_m(x,y) = \{u,v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x,y,d) = \sum_{(u,v)\in W_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2$$

Correlation Approach

PROS

- Easy to implement
- Produces dense disparity map
- Usually is slow

CONS

- Needs textured images to work well
- Inadequate for matching image pairs from very different viewpoints due to illumination changes
- Window may cover points with quite different disparities
- Inaccurate disparities on the occluding boundaries

Feature-based Approach

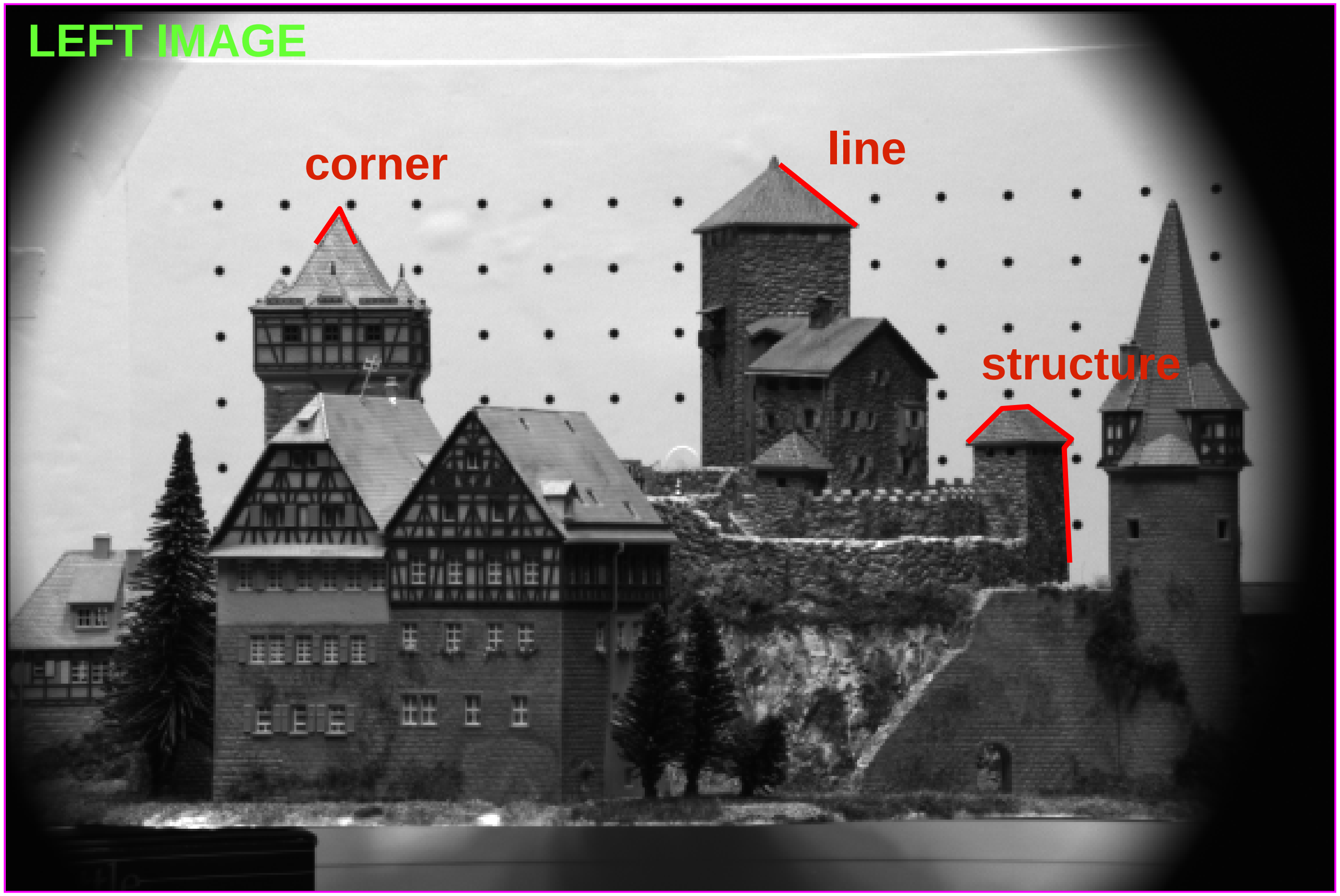
Features

- Edge points
- Lines (length, orientation, average contrast)
- Corners

Matching algorithm

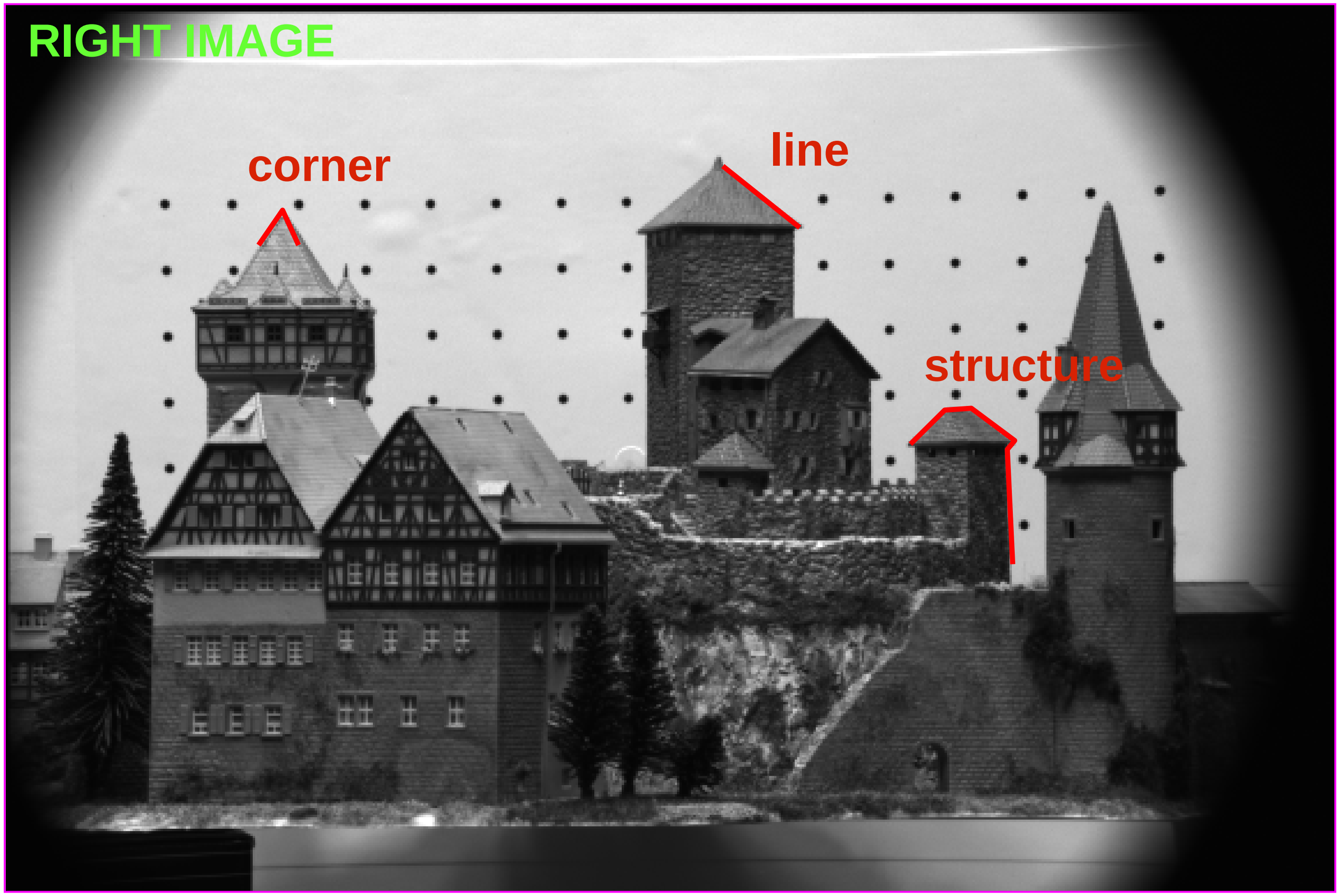
- Extract features in the stereo pair
- Define a suitable similarity measure for these features
- Use constraints to reduce number of matches
- Geometric constraints
  - Need only match features on same horizontal line
- Analytic constraints
  - Uniqueness – each feature has at most one match
  - Continuity – disparity varies continuously almost everywhere across this image

Feature-based Approach



For each feature in the left image…

Feature-based Approach



Search in the right image… the disparity (dx, dy) is the displacement when the similarity measure is maximum

Feature-based Approach

PROS

- Relatively insensitive to illumination changes
- Good for man-made scenes with strong lines but weak texture or textureless surfaces
- Work well on the occluding boundaries (edges)
- Could be faster than the correlation approach

CONS

- Only sparse depth map
- Feature extraction may be tricky for some features
  - Often uses corners as the features to match
  - Lines (Edges) might be partially extracted in one image
  - How to measure the similarity between two lines?

Problem Definition

- Simple stereo configuration
  - Corresponding points are on same horizontal line
  - This makes correspondence search a 1D search
  - Need only look for matches on same horizontal line
- if two cameras are in an arbitrary location is there a similar constraint to make search 1D?
  - Yes, called epipolar constraint
  - Based on epipolar geometry
  - We will derive this constraint
  - Consider two cameras that can see a single point P
  - They are in an arbitrary positions and orientation
    - One camera is rotated and translated relative to the other camera

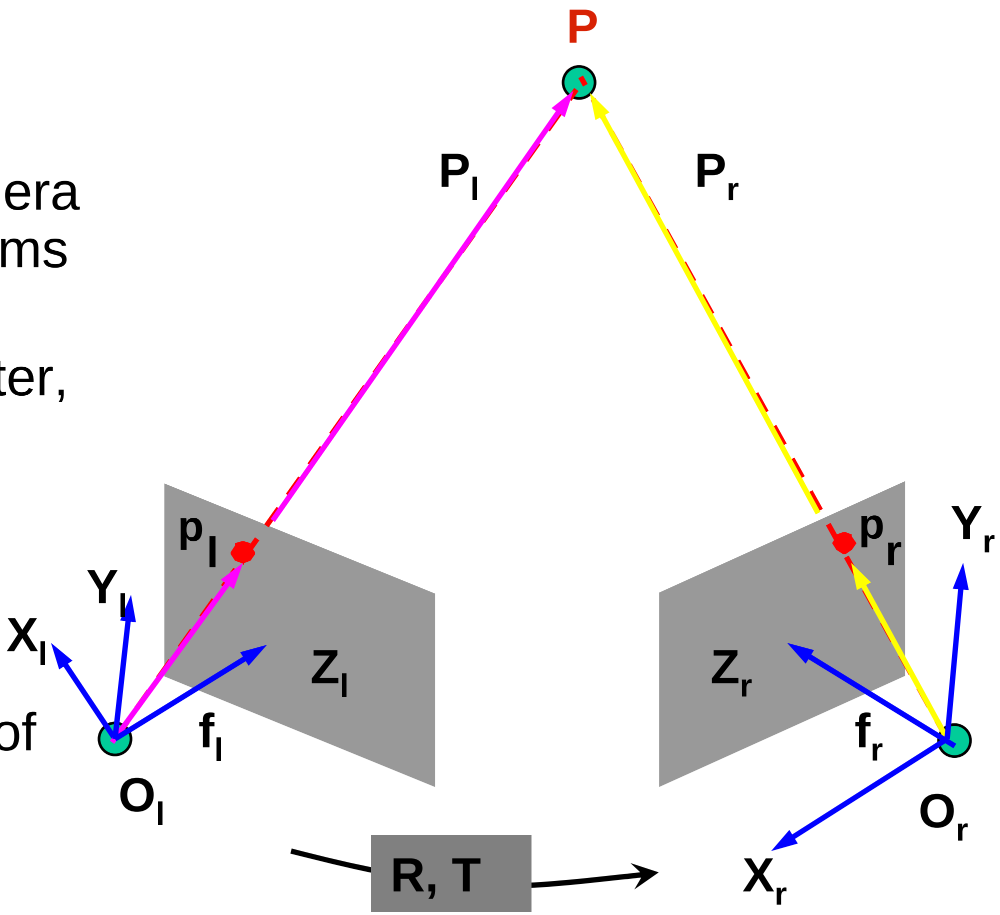
Parameters of a Stereo System

Intrinsic Parameters

- Characterize the transformation from camera to pixel coordinate systems of each camera
- Focal length, image center, aspect ratio

Extrinsic parameters

- Describe the relative position and orientation of the two cameras
- Rotation matrix R and translation vector T



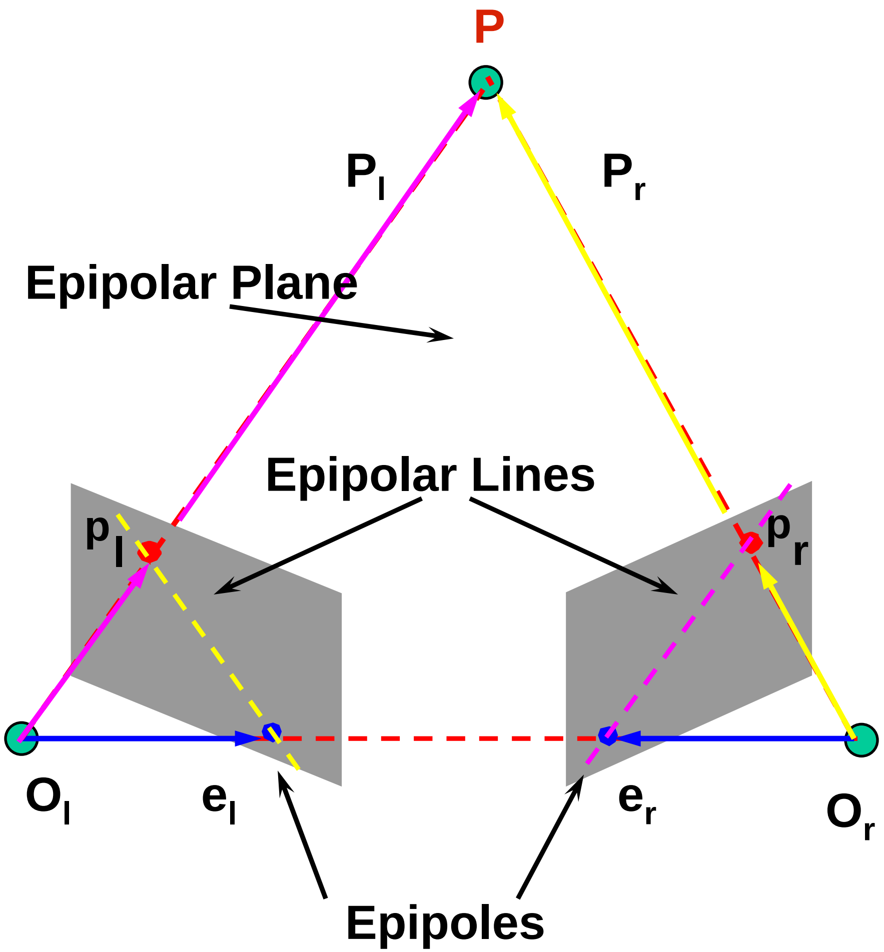
Epipolar Geometry

Motivation: where to search correspondences?

- Epipolar Plane
  - A plane going through point P and the centers of projection (COPs) of the two cameras
- Epipolar Lines
  - Lines where epipolar plane intersects the image planes
- Epipoles
  - The image in one camera of the COP of the other

Epipolar Constraint

- Corresponding points must lie on epipolar lines
- True for EVERY camera configuration!



Epipolar Geometry

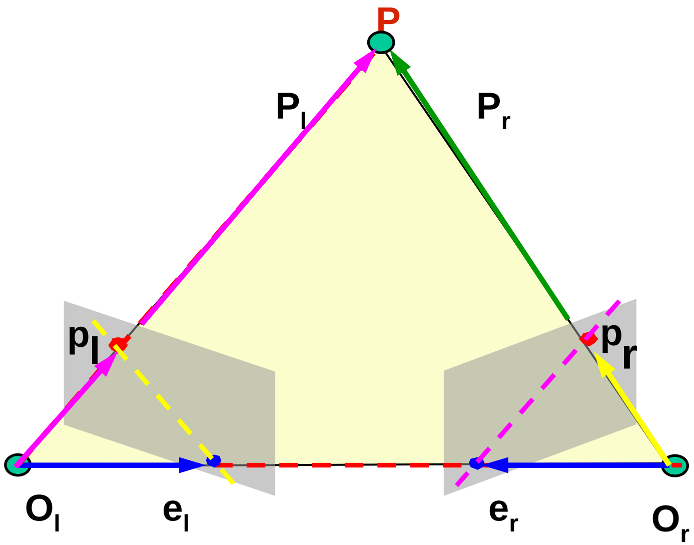
Epipolar plane: plane going through point P and the centers of projection (COPs) of the two cameras

Epipolar lines: where this epipolar plane intersects the two image planes

Epipoles: The image in one camera of the COP of the other

Epipolar Constraint: Corresponding points between the two images must lie on epipolar lines

Essential Matrix



$$P_r^T E P_l = 0 \qquad \Rightarrow \quad p_r^T E p_l = 0$$

$$\mathbf{P_r}^{\mathbf{T}}\mathbf{E}\mathbf{P_l} = 0$$

$$\mathbf{p}_l = \frac{f_l}{Z_l}\mathbf{P_l} \qquad \mathbf{p}_r = \frac{f_r}{Z_r}\mathbf{P}_r$$

$$\downarrow$$

$$\mathbf{p_r}^{\mathbf{T}}\mathbf{E}\mathbf{p_l} = 0$$

Essential Matrix

Essential Matrix E = RS

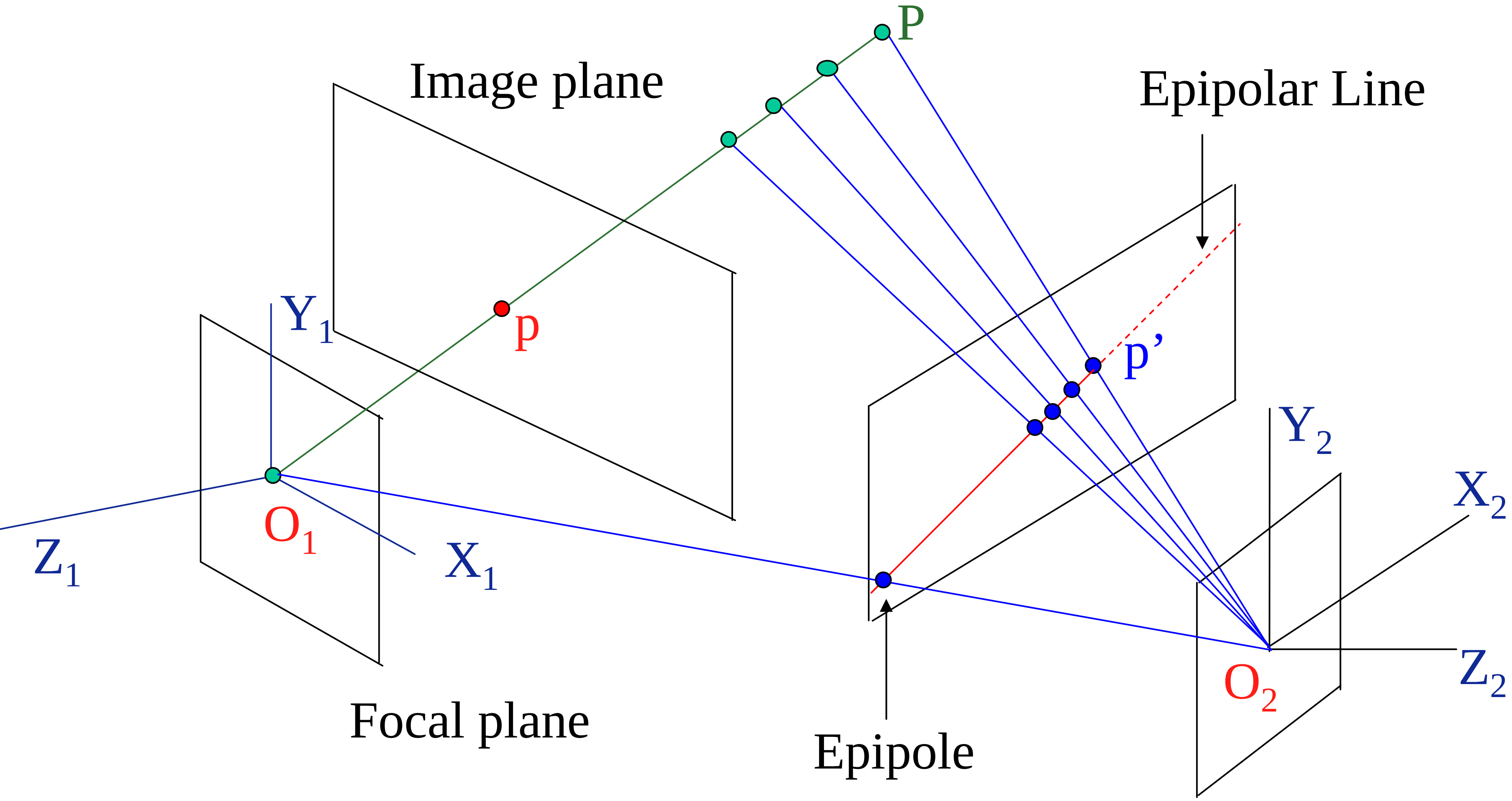
$$\mathbf{p_r}^{\mathbf{T}}\mathbf{E}\mathbf{p_l} = 0$$

- A natural link between the stereo point pair and the extrinsic parameters of the stereo system
  - One correspondence -> a linear equation of 9 entries
  - Given 8 pairs of (pl, pr) -> E
- Mapping between points and epipolar lines we are looking for
  - Given p_l, E -> p_r on the projective line in the right plane
  - Equation represents the epipolar line of either pr (or pl) in the right (or left) image

Note:

- pl, pr are in the camera coordinate system, not pixel coordinates that we can measure

What does Essential Matrix Mean?



Projective Line

- If we have a point in one image then this means the 3D point P is on the line from the origin through that point in the image plane
- So in the other image the corresponding point must be on the epipolar line
- What is the meaning of Ep_l ?
    - the line in the right plane that goes through p_r and the epipole e_r
- Therefore the essential matrix is a mapping between points and epipolar lines
- Ep_l defines the equation of the epipolar line in the right image for point p_r in the right image
- $E^T p_r$ defines the equation of the epipolar line in the right image for point p_l in the left image

Fundamental Matrix

Mapping between points and epipolar lines in the pixel coordinate systems

- With no prior knowledge of the stereo system parameters

From Camera to Pixels: Matrices of intrinsic parameters

$$\mathbf{M}_{\text{int}} = \begin{bmatrix} -f_x & 0 & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Rank ($\mathbf{M}_{\text{int}}$) =3

$$\begin{tikzcd}
\mathbf{p_r}^{\mathbf{T}}\mathbf{E}\mathbf{p_l} = 0 \arrow[d, "\mathbf{p_l} = \mathbf{M_l^{-1}}\overline{\mathbf{p}}_\mathbf{l}"', "\mathbf{p}_r = \mathbf{M}_r^{-1}\overline{\mathbf{p}}_r"] \\
\overline{\mathbf{p}}_\mathbf{r}^{\mathbf{T}}\mathbf{F}\overline{\mathbf{p}}_\mathbf{l} = 0
\end{tikzcd}$$

$$\mathbf{F} = \mathbf{M}_r^{-\mathbf{T}}\mathbf{E}\mathbf{M}_l^{-1}$$

Essential/Fundamental Matrix

- Essential and fundamental matrix differ
- Relate different quantities
  - Essential matrix is defined in terms of camera co-ordinates
  - Fundamental matrix defined in terms of pixel co-ordinates
- Need different things to calculate them
  - Essential matrix requires camera calibration and knowledge of correspondences
    - known intrinsic parameters, unknown extrinsic parameters
  - Fundamental matrix does not require any camera calibration, just knowledge of correspondences
    - Unknown intrinsic and unknown extrinsic
- Essential and fundamental matrix are related by the camera calibration parameters

Essential Matrix

Essential Matrix E = RS

- 3x3 matrix constructed from R and T (extrinsic only)
  - Rank (E) = 2, two equal nonzero singular values

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Rank (R) =3 **Rank (S) =2**

- E has five degrees of freedom (3 rotation, 2 translation)
- If we know R and T it is easy to compute E
  - use the camera calibration method of Ch. 6 for two cameras
- But if we just have correspondences between the two stereo cameras then computing E is not simple

Fundamental Matrix

Fundamental Matrix

$$\mathbf{F} = \mathbf{M}_r^{-\mathrm{T}}\mathbf{E}\mathbf{M}_l^{-1}$$

- Rank (F) = 2
- Encodes info on both intrinsic and extrinsic parameters

- Enables full reconstruction of the epipolar geometry
- In pixel coordinate systems without any knowledge of the intrinsic and extrinsic parameters
- Linear equation of the 9 entries of F but only 8 degrees of freedom because of homogeneous nature of equations

$$\overline{\mathbf{p}}_{\mathbf{r}}{}^{\mathbf{T}}\mathbf{F}\overline{\mathbf{p}}_{\mathbf{l}} = 0 \quad \Rightarrow \quad (x_{im}^{(l)} \quad y_{im}^{(l)} \quad 1)\begin{bmatrix} f11 & f12 & f13 \\ f21 & f22 & f23 \\ f31 & f32 & f33 \end{bmatrix}\begin{pmatrix} x_{im}^{(r)} \\ y_{im}^{(r)} \\ 1 \end{pmatrix} = 0$$

Computing E and F

- If we known extrinsic parameters of stereo system we can compute E = R S
- Using E and intrinsic parameters we can compute epipolar lines
- If we do not know intrinsic and extrinsic parameters can we compute F?
- Yes, for at least 8 correspondences
  - Why 8? Nine elements in F, but system is homogeneous
  - Choose 8 correct correspondences manually
- Then compute F matrix
  - Then use F to guide search by computing the epipolar lines
  - This will simply the process of finding correspondences

Problem Definition

- What we have so far
  - Correspondences using either correlation or feature based approaches
  - Epipolar Geometry from at least 8 point correspondences
- Given a set of valid correspondences
  - Want to compute the 3D position of the points in space that are associated with each correspondence
  - There are different approaches depending on what we know about the cameras

Reconstruction possibilities

Three cases of 3D reconstruction depending on the amount of a priori knowledge on the stereo system

- Both intrinsic and extrinsic known - > can solve the reconstruction problem unambiguously by triangulation. By unambiguously we mean computing the true positions in 3D space of the feature points relative to the stereo cameras.
- Only intrinsic known -> recovery structure and extrinsic up to an unknown scaling factor. Since we do not know the position of the stereo cameras this means that we can know the true distance between the cameras only up to a scale.
- Only correspondences -> reconstruction only up to an unknown, global projective transformation. We will not cover this situation, since this is rather theoretical.

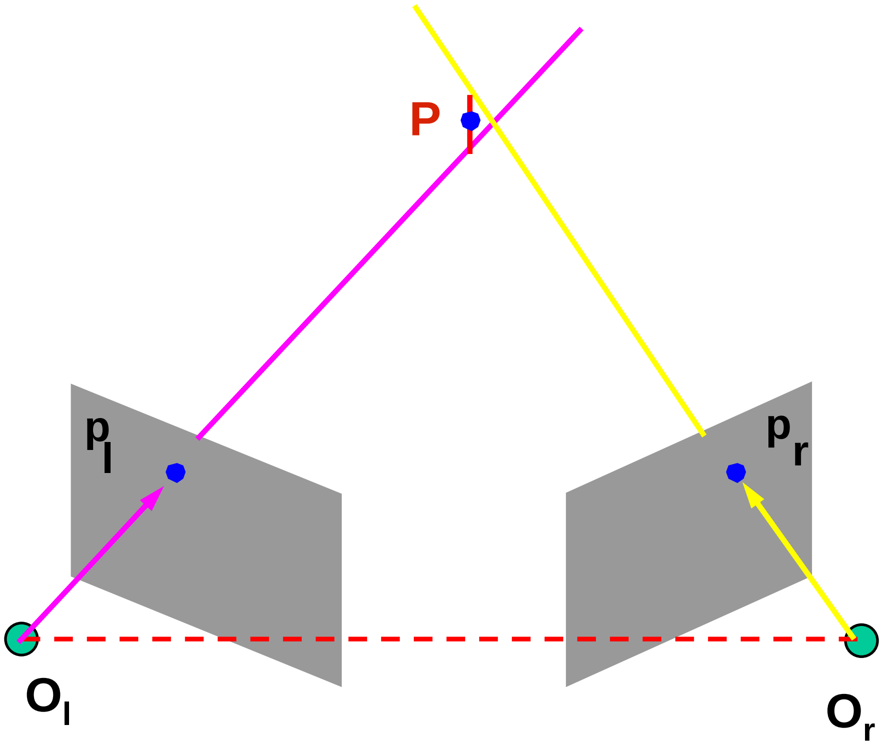
Both intrinsic and extrinsic known

- Occurs when you have calibrated the cameras using standard camera calibration
- Using the intrinsic and extrinsic camera calibration and the pixel co-ordinates of the matching feature points we need to compute the 3D location of that feature point
- Can be done using simple geometric triangulation

Known intrinsic and extrinsic

Solution

- Triangulation: Two rays are known and the intersection can be computed
- Problem: Two rays will not actually intersect in space due to errors in calibration and correspondences, and pixelization
- Solution: find a point in space with minimum distance from both rays



Reconstruction results

- You get camera position and 3D data points for the correspondences
- Can use camera position for doing augmented reality (adding virtual objects)
- Can use 3D data points to make 3D models of objects
- Basic technology used in many other applications

Homography

- Consider a point x = (u,v,1) in one image and x’=(u’,v’,1) in another image
- A homography is a 3 by 3 matrix M
- $$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$
- The homography relates the pixel co-ordinates in the two images if x’ = M x
- When applied to every pixel the new image is a warped version of the original image

Homography conditions

- Two images are related by a homography if and only if
- Both images are viewing the same plane from a different angle (your assignment)
- Both images are taken from the same camera but from a different angle
  - Camera is rotated about its center of projection without any translation
- Note that the homography relationship is independent of the scene structure
  - It does not depend on what the cameras are looking at
  - Relationship holds regardless of what is seen in the images

Homography for Rotation

- If there is no translation $P_r = RP_l$
- Home position – projection equation

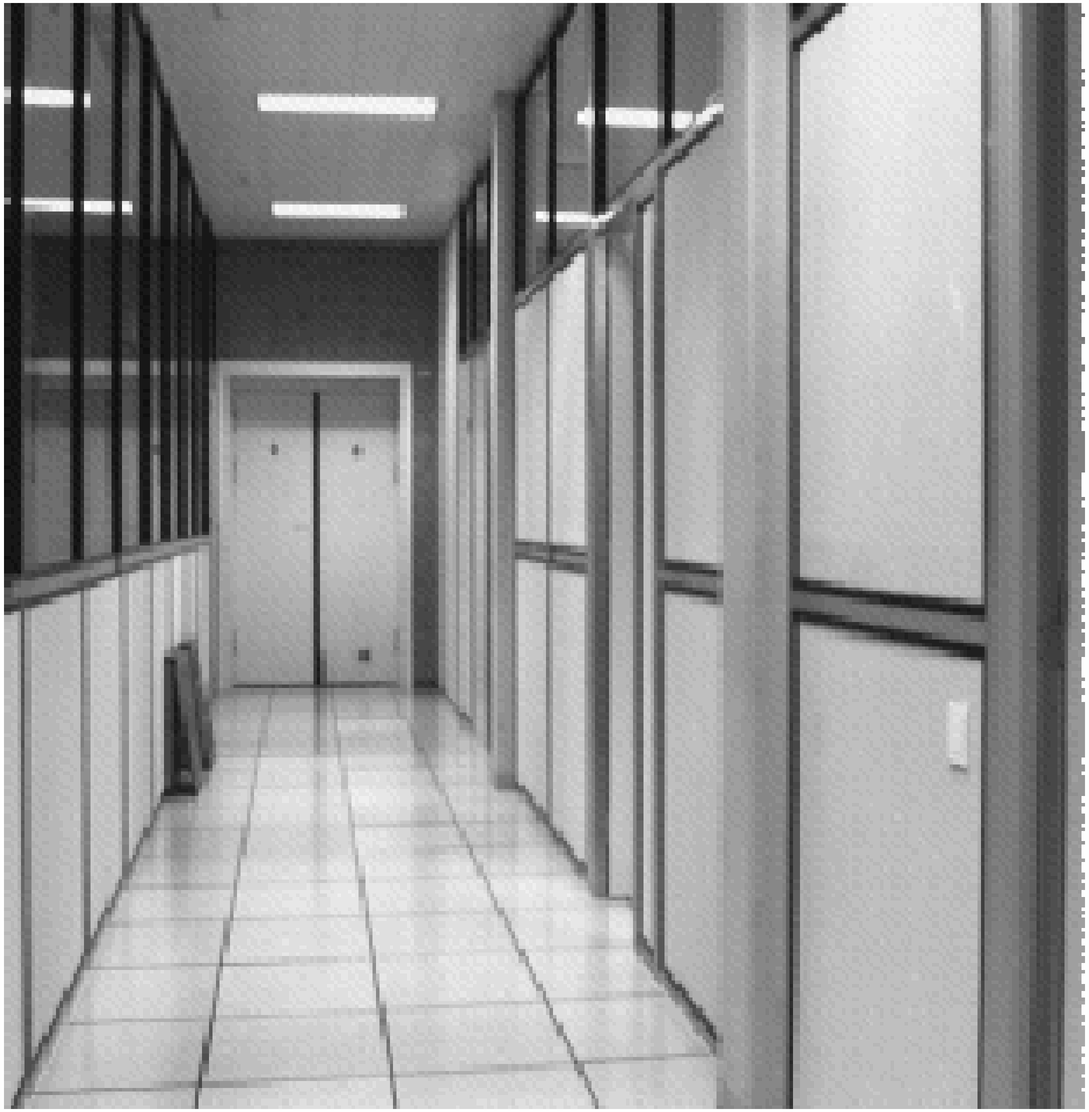
$$x = K[I \mid 0]\begin{bmatrix} X \\ 1 \end{bmatrix} = KX$$

- Rotation by a matrix R – projection equation

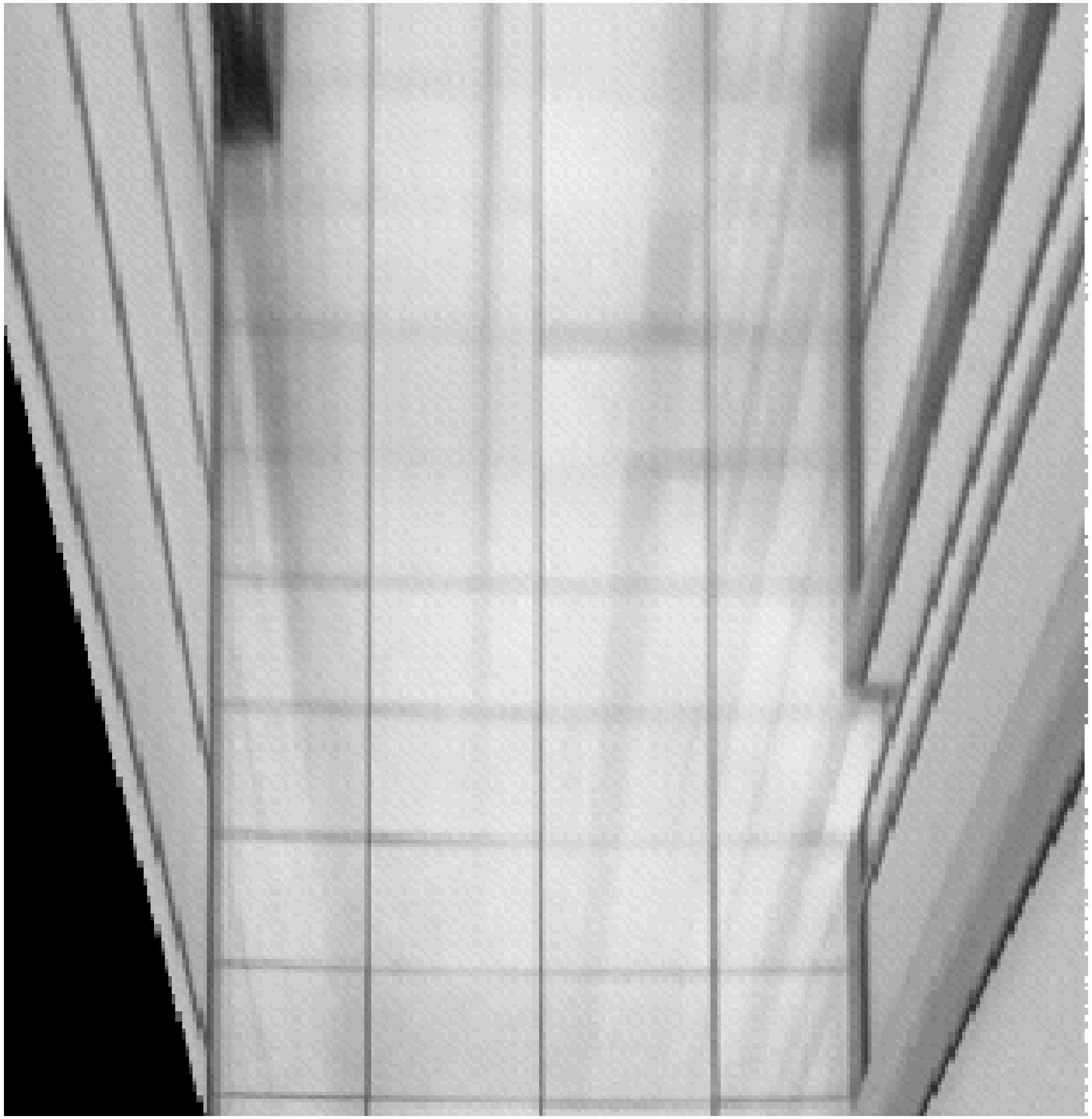
$$x' = K[R \mid 0]\begin{bmatrix} X \\ 1 \end{bmatrix} = KRX$$

- So $x' = KRK^{-1}x$ where K is calibration matrix
- Where KRK^{-1} is a 3by3 matrix M called a homography

Original camera view



Downward rotation via homography



Homgraphy versus epipolar geometry

- If two cameras are translated then epipolar geometry holds
  - It maps a point in one image to a line in the other image
  - Actual matching point depends on the depth of that point, that is its distance from the camera
- As the translations -> 0 the baseline b of the two cameras also goes to zero
  - Once it is zero epipolar geometry does not hold any longer
- Homgraphy holds when no translation
  - It maps a point in one image to a point in the other image
  - The mapping does not depend on the depth (distance) of point to the camera
  - This makes sense because baseline z zero means ordinary stereo triangulation equations fail

Computing homography

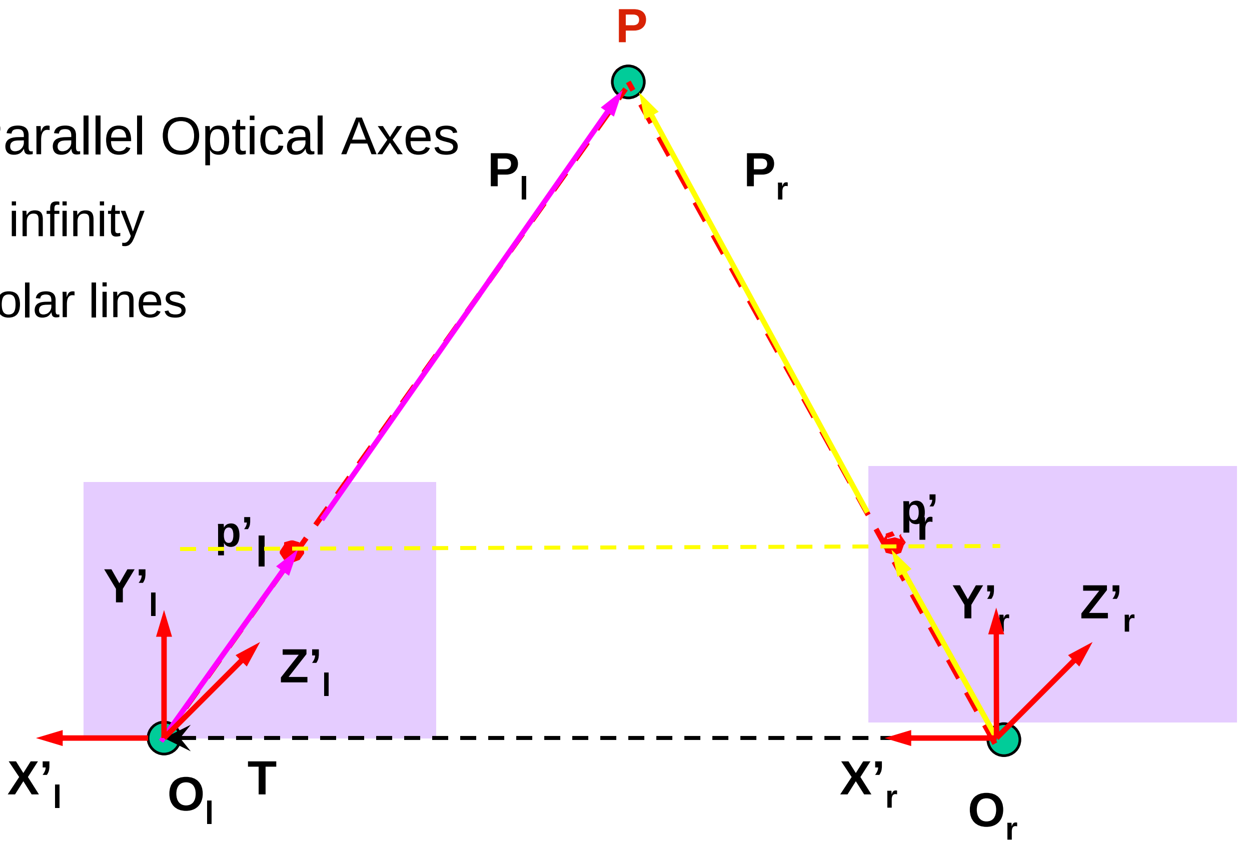
- For your purposes important to understand
  - When translation is null epipolar geometry fails to hold
  - In this case you only have image rotation
  - Can not compute depth for any point for which you have correspondences
  - Can compute the homography matrix from
    - The camera calibration and the know rotation or
    - Correspondences between the two images
  - If you just use correspondences you can make a mosaic from rotating images
  - Actually, can make a mosaic whenever you have a homograpy relationship
    - when you are looking at a plane and when rotating
    - Like having a high resolution camera from a single point of view

Problem Definition

- Given a pair of stereo images, the intrinsic parameters of each camera, and the extrinsic parameters of the system, R, and T, compute the image transformation that makes epipolar lines collinear and parallel to horizontal axis
- Basically convert a general stereo configuration to a simple stereo configuration
    - This speeds up matching because searching horizontal epipolar lines is easier than general epipolar lines
    - So rectification is a good pre-processing step if you want to do correspondence faster
    - Correlation based correspondence algorithms always assume a simple stereo configuration

Stereo Rectification

- Stereo System with Parallel Optical Axes
  - Epipoles are at infinity
  - Horizontal epipolar lines



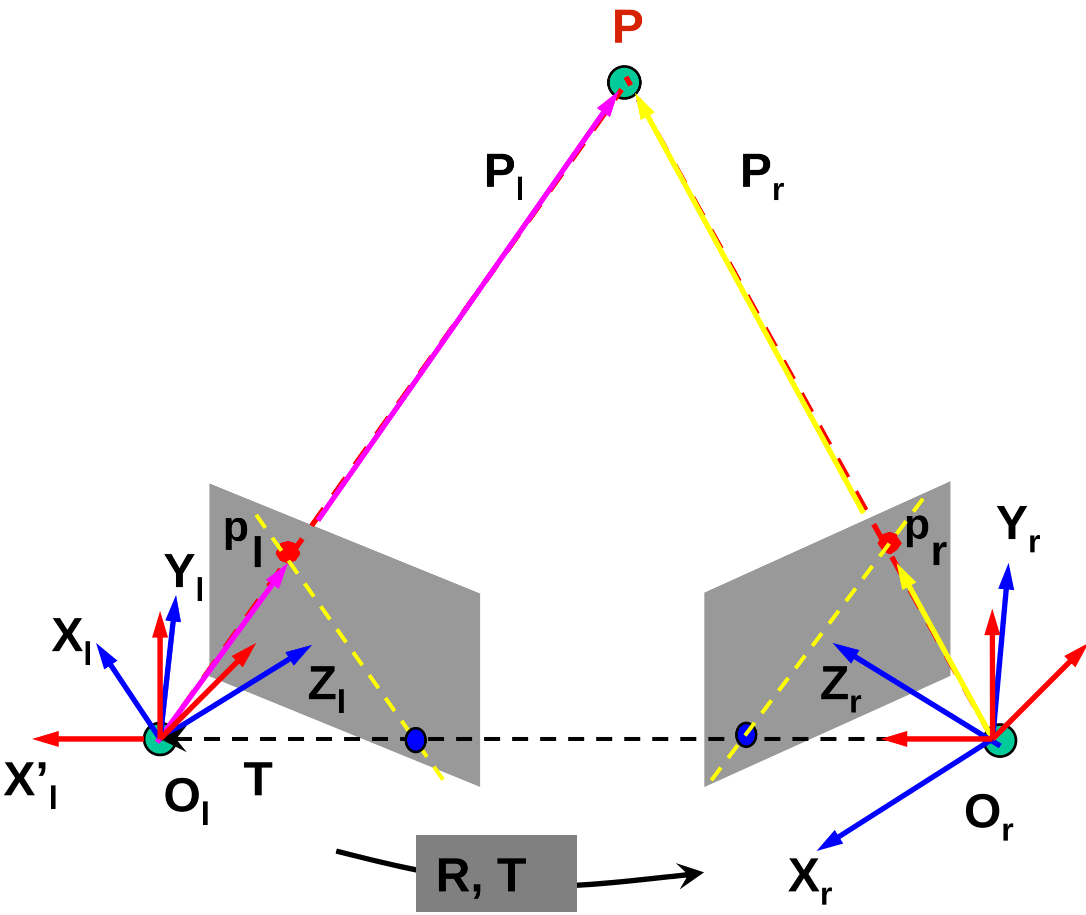
Rectification

- Given a stereo pair, the intrinsic and extrinsic parameters, find the image transformation to achieve a stereo system of horizontal epipolar lines
- A simple algorithm: Assuming calibrated stereo cameras

Stereo Rectification

Algorithm

- Rotate both left and right camera so that they share the same X axis : O_r-O_l = T
- Define a rotation matrix R_{rect} for the left camera
- Rotation Matrix for the right camera is $R_{rect}R^T$
- Rotation can be implemented by image transformation called a homography (see homography.ppt)

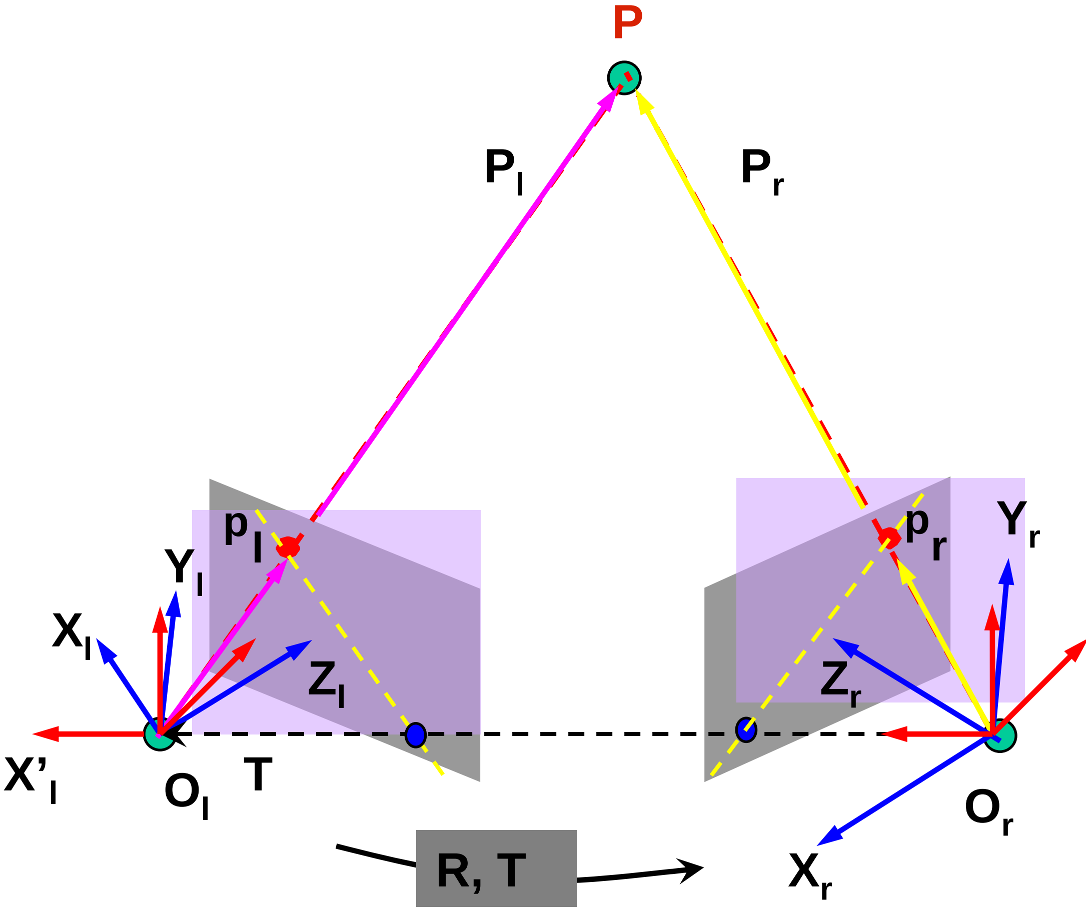


$X_l' = T$, $Y_l' = X_l' x Z_l$, $Z'_l = X_l' x Y_l'$

Stereo Rectification

Algorithm

- Rotate both left and right camera so that they share the same X axis : O_r-O_l = T
- Define a rotation matrix R_{rect} for the left camera
- Rotation Matrix for the right camera is $R_{rect}R^T$
- Rotation can be implemented by image transformation called a homography (see homography.ppt)

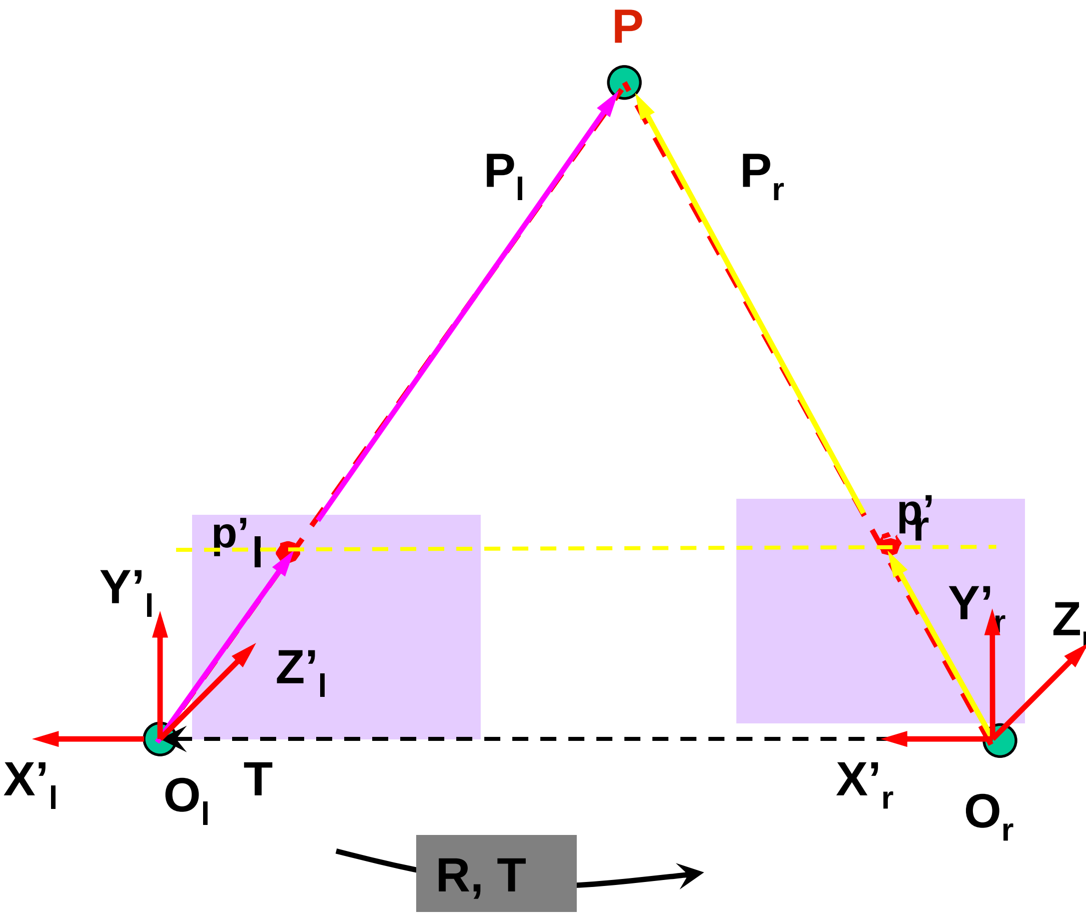


$X_l' = T$, $Y_l' = X_l'xZ_l$, $Z'_l = X_l'xY_l'$

Stereo Rectification

Algorithm

- Rotate both left and right camera so that they share the same X axis : O_r-O_l = T
- Define a rotation matrix R_{rect} for the left camera
- Rotation Matrix for the right camera is $R_{rect}R^T$
- Rotation can be implemented by image transformation (see homography.ppt)

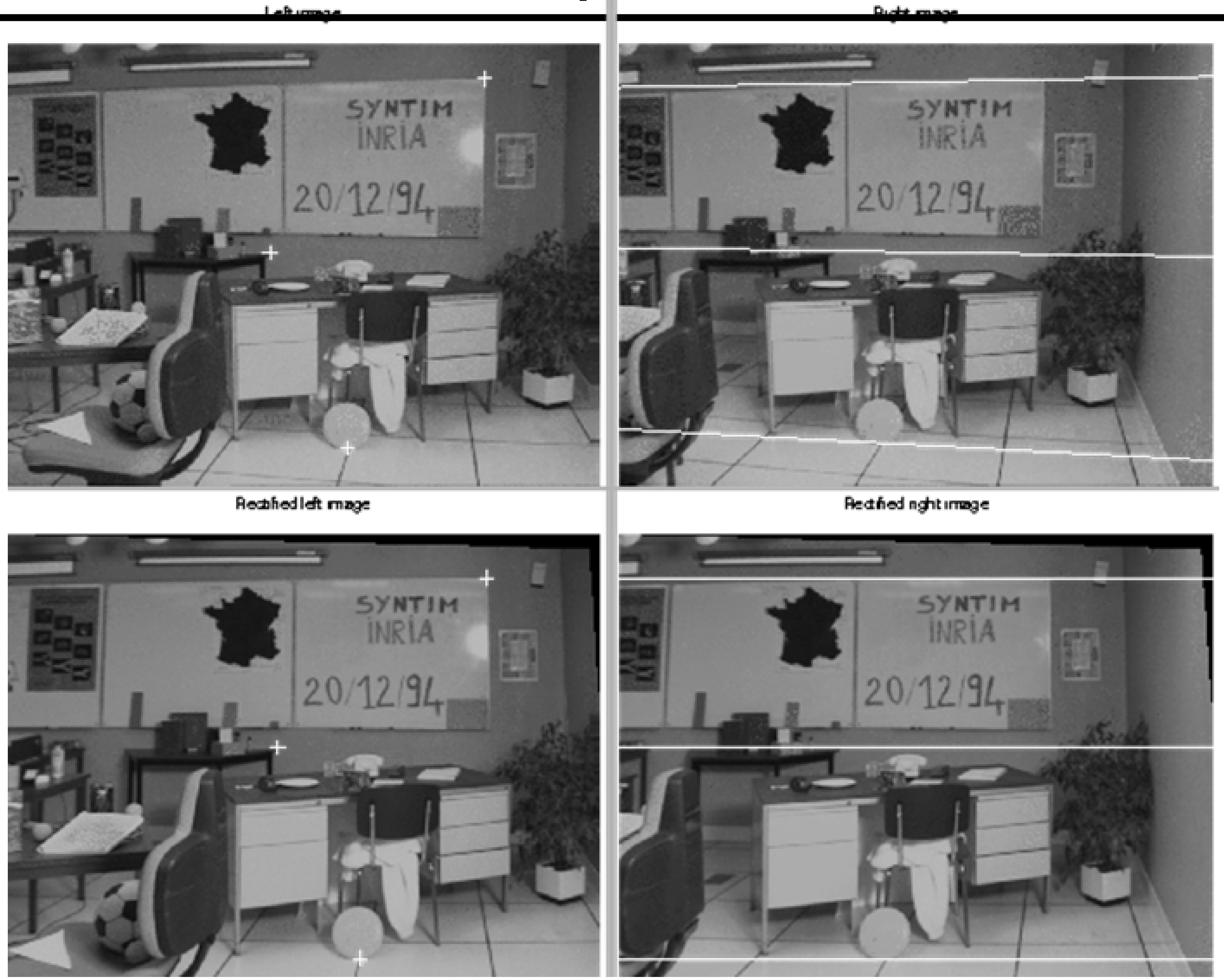


T' = (B, 0, 0), $P'_r = P'_l - T'$

Homography for Rotation

- We know that $x' = KRK^{-1}x$
- This means that for any pixel in the new image where x’ = [u’,v’,1] we can compute the pixel x = [u,v,1] in th old image that is mapped to x under this rotation
    - Go through every pixel x’ in the new image and put in its place the image at location x in the old image
- This make a new image that is a rotated version of the original old image
- A simple way to apply the rotation to these images
- Note that x ‘ = M x and x = M^{-1} x’ where M is the homography that does the rotation
    - So given M (which we can compute) we can make a new image which looks like a rotated version of the old

Rectification Example



Rectification

- Used to transform a general stereo system into a simple stereo system

1. Do the rectification to get two new images
2. Do correspondence in the new simple stereo images
3. Take the 3D points that are found and convert them back into the old image frames (invert rotations)

The result is a set of 3D points, but they are computed more quickly because we do correspondence in simple stereo.

Appearance Based Recognition

Dr. Gerhard Roth

Recognition (Section 10.4)

- Given an image I, containing a single object and a database of images find the image in the database that is most similar to image I
- One possible way to recognize objects
  - Database has views of same object under different conditions
  - Input image is “close” to one of these database views
- Commonly used in face recognition systems
  - Database has number of faces (standard position)
  - Input image is a single face (standard position)

= ?

Assumptions in appearance recognition

1. Each image contains only a single object.
2. The objects are imaged by a fixed camera under weak perspective.
3. The images are normalized in size: that is the image frame is the minimum rectangle enclosing the largest appearance of this object.
4. The energy of the pixel values is normalized to one: $\sum_{i=1}^{N} \sum_{i=1}^{N} I(i, j)^2 = 1$
5. The object is completely visible and unconcluded in all images.

Comparing Images

- First transform 2D image into a 1D vector
  - N x N image X, becomes an N^2 dimensional vector
  - $x = [X_{11}, X_{12}, \ldots, X_{1N}, X_{21}, \ldots, X_{NN}]^T$
- Now given two images X_1 and X_2, and the two vectors x_1 and x_2 how do we compare them?
- One way, is to find distance between them according to some norm (usually L2)
  - $\mathrm{Dist}(x_1, x_2) = || x_1 – x_2 ||$ (just sum of squares of differences)
- If $\mathrm{Dist}(x_1, x_2) = 0$ then x_1 and x_2 are identical
- In image processing language
  - $\mathrm{Dist}(x_1, x_2)$ is called Euclidian distance (L2 norm)

Comparing Images - Euclidian Distance

- Given a single image y and a database of m images labeled x_1, x_2, …, x_m
- Want to find closest image in database to y?
  - Compute Euclidian distance and find smallest result
- How long does this take?
  - Time is proportional to m * N^2, where N^2 is number of pixels in the original image, and m number of images in database
  - Takes a long time since N^2 is large and often so is m

Problems with this approach

- May need a very large database
  - Require a different image for different lighting conditions
  - Not much can be done about this issue in a simple way
    - Just hope that enough memory is available
- Often will require a lot of time
  - As in the previous slide to find the best match we need to do convolution against every image in the database
- Can not do much about size of database
  - This problem is intrinsic to the basic approach
- But we can decrease the execution time
  - By using the eigenspace or Principal Components Method
  - Sometimes called the PCA approach

Idea behind the PCA Approach

- There is redundancy in these images
  - Parts of all the images are very similar
  - The faces are in a standard position and all faces are similar
  - We can exploit this redundancy with the PCA approach
- Normally to compare images we need to compare N^2 numbers, but with the PCA approach this is not true
  - We can represent images by k numbers, where $k << N^2$
  - Then to compare images we need only compare k numbers
  - How big is k? It depends on the redundancy of the images
  - If the images are not similar then k is large (close to N^2)
  - The more similar the images the smaller is k
  - For face comparison k is around 100, which is small!

PCA Approach

- To implement the PCA approach you need to
    - Apply the PCA algorithm to all the images in the database
    - Represent each image as a K element vector instead of an N^2 element vector
    - The value of K depends on the redundancy of the images but usually k << N^2
- To match a new image with the database
    - Convert the new image into this k element vector form
    - Compare the k element vector to each of the m vectors of k elements in the image database
    - Return the closest vector
- This is the image that is the best match in the sense of Euclidian distance