Stereo!

- We can infer the 3D-structure from 2 or more images taken from different viewpoints.
- The position of ~~each image depend~~ an object in each image depends on its depth.

  $\frac{1}{\text{depth}} \propto$ disparity (positional difference).

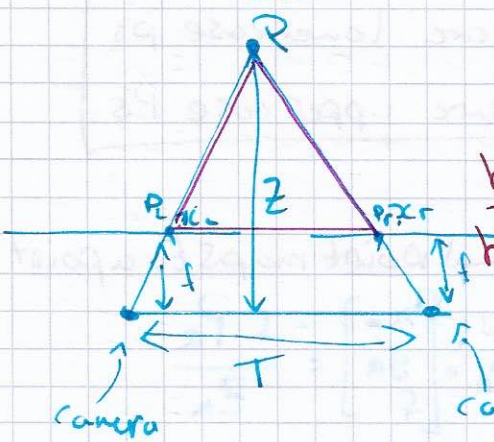
  The difference between the two images.
- If we have disparity and viewpoints then we can infer the 3D-scene structure.

There are 3 key Problems with Stereo.

1. Calibration - we need to determine relative position and orientation of cameras.
2. Correspondence - we need to determine matching points in the stereo view.
3. Reconstruction - we need to determine the 3D location in a scene of matched points via triangulation.

Triangulation

Simple Stereo:



By Similar Triangles:

$$\frac{\text{base}}{\text{height}} = \frac{T}{Z} = \frac{T - x_L + x_R}{Z - f}$$

$$Z = \frac{fT}{x_L - x_R} = \frac{fT}{d}$$

d = disparity. difference between location of P in left camera (x_L) and P in right camera (x_R).

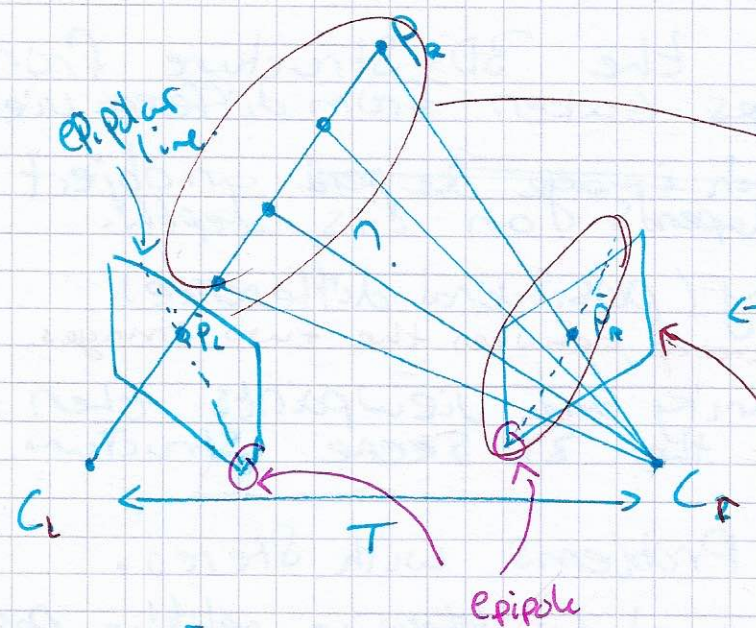
$$\frac{T}{Z} = \frac{T - x_1 + x_r}{Z - f}$$

$$TZ - Tf = TZ - Zx_1 + Zx_2$$

$$Z(x_1 - x_2) = fT$$

$$Z = \frac{fT}{x_1 - x_2}$$

Epipolar Geometry.



Epipolar lines

C_L sees the point P but from the image space cannot infer the depth so it could lie anywhere on a line connecting C_L and the actual point P.

this camera has R^T applied.

C_r ~~sees~~ can view from a different angle. It can 'see' this line. This is the epipolar line.

P ~~pco~~ from C_r's perspective must lie on ~~this~~ the epipolar line.

$$P_L = R^T P_R + T$$

R^T: Rotational difference between C_L and C_r

T: translational difference between C_L and C_r.

Epipolar Geometry defines a relationship between two stereo views

- For known viewpoints it constrains matches to lie ~~on~~ along epipolar lines.
- For unknown viewpoints, given the matching viewpoints it is able to estimate the viewpoint.

$\bar{P}_L$ and $\bar{P}_R$ are lowercase p's
P_L and P_R are uppercase P's.

$$P_L = R^T R_R + T$$

$$P_L - T = R^T P_R$$

$$P_R = R(P_L - T)$$

using previous knowledge of how an actual point maps to a point in the image space.

$$P_{RL} = \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \quad \bar{P}_L = \begin{bmatrix} x_L \\ y_L \\ f \end{bmatrix} = \frac{f P_L}{Z_L} \quad \bar{P}_R = \begin{bmatrix} x_R \\ y_R \\ f \end{bmatrix} = \frac{f P_R}{Z_R}$$

with respect to the left camera

$$P_R = P_L - T$$

all vectors P_L, T and $P_L - T$ all lie in the epipolar plane.

$$(P_L - T)^T (T \otimes P_L) = 0$$

← hence $(P_L - T)$ and $(T \otimes P_L)$ are perpendicular!

$(P_L - T)$: in plane

$(T \otimes P_L)$: cross product finds a vector perpendicular to ~~plane~~ the two vectors T and P_L since they are both in the plane

$$T \otimes P_L = S P_L$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

wrt Right camera.

$$P_R = R(P_L - T)$$

$$R^T = R^{-1}$$

$$R^T P_R = P_L - T$$

$$P_R^T R = (P_L - T)^T$$

the resulting vector will be perpendicular to the plane.

$$P_R^T R S P_L = 0$$

we call $RS = E$ the Essential matrix

$$P_R^T E P_L = 0.$$

$$\dot{P}_L = \frac{f P_L}{Z_L} \qquad \dot{P}_R = \frac{f P_R}{Z_R} \Rightarrow \frac{Z_R}{f} \dot{P}_R^T E \frac{Z_L}{f} \dot{P}_L = 0$$

$$\bar{P}_R^T E \bar{P}_L = 0.$$

now let $u_L = E \bar{P}_L = \begin{bmatrix} u_{L1} \\ u_{L2} \\ u_{L3} \end{bmatrix}$

we can also do ~~the same~~ a similar method to get the equation of the epipolar line in the ~~right image~~ left image.

then,

$$\bar{P}_R^T E \bar{P}_L = P_R^T u_L = x_R u_{L1} + y_R u_{L2} + f u_{L3} = 0.$$

This is the equation of the epipolar line in the right image.

Pixel values represent light intensity within a small region of the image plane with size $S_x \times S_y$.

Each pixel has coordinates $(\hat{x}, \hat{y})$

The image coordinates are $x = S_x(\hat{x} - \hat{O}_x)$

$y = S_y(\hat{y} - \hat{O}_y)$

So $\bar{p}_L = \begin{bmatrix} x_L \\ y_L \\ f \end{bmatrix} = M_L \begin{bmatrix} \hat{x}_L \\ \hat{y}_L \\ f \end{bmatrix} = M_L \hat{p}_L$

we also have

$\bar{p}_R^T = \hat{p}_R^T \begin{bmatrix} \hat{x}_R \\ \hat{y}_R \\ f \end{bmatrix}^T M_R^T = \hat{p}_R^T M_R^T$.

So $\bar{p}_R^T E \bar{p}_L = 0$

$\bar{p}_R^T E M_L \hat{p}_L = 0$

$\hat{p}_R^T M_R^T E M_L \hat{p}_L = 0$

$\hat{p}_R^T F \hat{p}_L = 0$

$F = M_R^T E M_L$

F is the fundamental matrix

We can also estimate F if we are give a set of ~~point~~ correspondences

ie $\hat{p}_R^T(i) F \hat{p}_L(i) = 0$ for $i = 1 \ldots N$

This is $Af = 0$

A: $N \times 9$ matrix defined by correspondence vectors $\hat{p}_R(i), \hat{p}_L(i)$.

f: components of F.

We can solve for f using Singular Value Decomposition ?

3D - Reconstruction.

Where does this come from?

Find a, b, c s.t. $a\hat{p}_L - bR^T P_R - T - c(p_L \otimes R^T p_R) = 0$.

Given corresponding points we know p_L, p_R.
Given calibrated views we know, R, T.

$$a\underbrace{\begin{bmatrix} \\ P_L \\ \\ \end{bmatrix}}_{3\times1} - b\underbrace{\begin{bmatrix} \\ R^T P_R \\ \\ \end{bmatrix}}_{3\times1} - c\underbrace{\begin{bmatrix} \\ P_L \otimes R^T P_R \\ \\ \end{bmatrix}}_{3\times1} = \underbrace{T}_{3\times1}$$

H is a 3x3 matrix

$$H = \begin{bmatrix} \vdots & \vdots & \vdots \\ P_L & R^T P_R & P_L \otimes R^T P_R \\ \vdots & \vdots & \vdots \end{bmatrix} \qquad H \begin{bmatrix} a \\ b \\ c \end{bmatrix} = T \implies \begin{bmatrix} a \\ b \\ c \end{bmatrix} = H^{-1}T.$$

$$\hat{p} = ((a p_L) + (b R^T p_R + T))/2 \; ?$$

($a p_L$: left camera's perspective; $b R^T p_R + T$: Right camera's perspective. with R^T and T being added to map the point to the same space as the left camera.)

Stereo Correspondence

- How do we match up points from each camera.?

- What do we match up between cameras?

Pixels, Regions, Feature Points, edges and boundaries, Objects.

Pixels - dense but not distinctive ← could have lots of similar pixes.

Regions - dense but more distinctive

Feature Points - Sparse but distinctive

Edges - Sparse but ambiguous ~~lots of~~ few edges, but edges are hard to matchup.

Objects - Sparse, distinctive but hard to identify and match.

(could detect a flower in a flowerbed but how sure are you it's the same flower.)

Where to search?

- Uncalibrated views: Correct matches could lie anywere in other views.
- known fundamental matrix - Matches constrained to lie along the epipolar lines.

How to compare Elements?

This depends on what you are trying to match.

You could use:

- Pixel intensities ~~(RGB).~~
- Local colour variation.
- Local Structure Variation.
- Position and angle of edges
- Shape and pose.

There are several challenges when comparing edges.

- Lighting and noise effects.
- Perspective transformations.
- Occlusions. - things may cover an object from another perspective.
- Repetitive Structure. (chess board) how do you know which area we are in?.

Region Based Methods

Compare pixel values in image regions

- For a region in the left image compute similarity with regions of same size in the right image.
- The corresponding region is the centre of the most similar region.

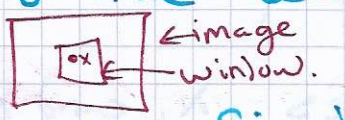
Stereo image pair $I_L(x,y)$ and $I_R(x,y)$.

For each pixel find the disparity $d=(d_x,d_y)$ which minimises/maximises the cost function.

(find the offset which maximises the similarity between pixels).

$$C(x,y,d) = \sum_{(i,j)\in W(x,j)} Sim(I(x+i, y+j), I(x+i+d_x, y+j+d_y))$$

$W(x,y)$ is the window of pixels around (x,y).



There are many similarity measures we could use:

- Sum of squared differences.

$$Sim(u,v) = (u-v)^2$$

- Similar pixel count

$$Sim(u,v) = \begin{cases} 1 & \text{if } |u-v| < T \\ 0 & \text{else.} \end{cases}$$

(T = threshold ~~similarity~~ difference.)

- Normalised cross correlation:

$$Sim(u,v) = (u-\bar{u})(v-\bar{v}) / N_u N_v$$

$$\bar{u} = \frac{1}{|W(i,j)|} \sum_{k,l\in W(i,j)} u(k,l)$$

↑ mean.

$$N_u = \sqrt{\sum_{k,l\in W(i,j)} (u-\bar{u})^2}$$ ← variance.

Feature Based Methods

- This restricts the search to a much sparser set of features.
- We can find distinct points in each view and then match these points by comparing pixels or image descriptors of local regions about each point.

For example, Harris Corner detector, RANSAC for uncalibrated matching, matching using the Scaled-invariant feature transform.

Harris Corner detector

This uses the covariance of spatial gradient vectors within the region W.

Saliency = how important

$$A = \sum_{x,y \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

The eigen values λ_1 and λ_2 ~~eod~~ of A indicate the spread of gradients in the region

for example 2 high values indicate a busy region.

The Saliency metric $Sal = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$

If both λ_1 and λ_2 are large then Sal is large

If λ_1 or λ_2 is small then Sal is small.

can threshold Saliency to get key points.

Calibrated Feature Matching

- we determine the Salient points (key!) in each image.

For each point in on image Find the best matching point in the other image.

we can determine the best matching point by using region based matching ~~along the~~ around the Salient points.

we also only Search around the ~~Sets~~ epipolar lines.

↑ because they tell us where the point should lie based on the other camera's perspective

Uncalibrated Matching

- When we have calibrated cameras we are able to use the epipolar lines (reduces our search space.)
- However when we do not have calibrated cameras we cannot use the epipolar lines, so we need another approach. (Or we will have to do a lot of comparisons. BAD!)

One such approach is to use RANSAC. - RANdom SAmple Consensus.

1. Find potential correspondences
2. Select a subset of matches at random (minimum 7)
3. Compute F from subset.
4. Assess how good F is for all other correspondences.
5. Repeat until you find the best F.

Basically k-fold cross validation. k - number of times an F is calculated.

Scale-Invariant Feature Transform

There are 2 main components: (key)

1. Scale invariant detection of salient points.
2. Match using highly distinct local descriptors.

1. key (Salient) point detection.
   - extrema max/min in the difference of Gaussian blurred versions of image
     (Generate images with different levels of blur. calculate pairwise differences)
     D.o.G tree. ← Difference of Gaussians
   - Points imaged at different resolutions appear at different levels of D.o.G tree.
     Scale invariance

2. Spatial Gradient Descriptors:

- Built from histograms of spatial gradients in local neighbourhood.
- Good rotational and perspective wrap properties.