C18 Computer Vision

David Murray

david.murray@eng.ox.ac.uk
www.robots.ox.ac.uk/~dwm/Courses/4CV

Michaelmas 2015

UNIVERSITY OF OXFORD

Computer Vision: This time ...

1. Introduction; imaging geometry; camera calibration
2. Salient feature detection – edges, line and corners
3. Recovering 3D from two images I: epipolar geometry.
4. **Recovering 3D from two images II: stereo correspondence algorithms; triangulation.**

UNIVERSITY OF OXFORD

Recovering 3D from two images II

4.1 Introduction

4.2 The Correspondence Problem

4.3 Triangulation for Depth Recovery

UNIVERSITY OF OXFORD

4.1 Introduction

We have seen that because the projection into a second camera of the backprojected ray from the first is a straight line, the search for potential matches must lie along straight epipolar lines

Search occurs in 1D, which must be less demanding than 2D search throughout the image

We have seen how knowledge of the camera matrices allows us to recover the fundamental matrix, which defines this epipolar geometry

UNIVERSITY OF OXFORD

Introduction

The key tasks now are:

1. Determining which points in the images are from the same scene location — **the correspondence problem**.

   We will discuss a correlation-based approach suited to dense stereo.

   Will also consider a dynamic programming approach to incorporate further constraints on matching

2. Determining the 3D structure by back-projecting rays, or **triangulation**.

   Although triangulation seems straighforward, it turns out to be a non-trivial estimation problem to which there are several approaches.

3. We will consider briefly how correspondence between sparse features edges and corners fits into our framework

UNIVERSITY OF OXFORD

4.2 Establishing Correspondence

With knowledge of the fundamental matrix, we are able to reduce the search for correspondence down to a search along the epipolar line $\mathbf{l}' = \boldsymbol{F}\mathbf{x}$.

Correspondence algorithms can be either:

- Sparse: correspondence is sought for individual corner or edge features; or
- Dense: correspondences are sought for all the pixels in an image.

The methods can be either:

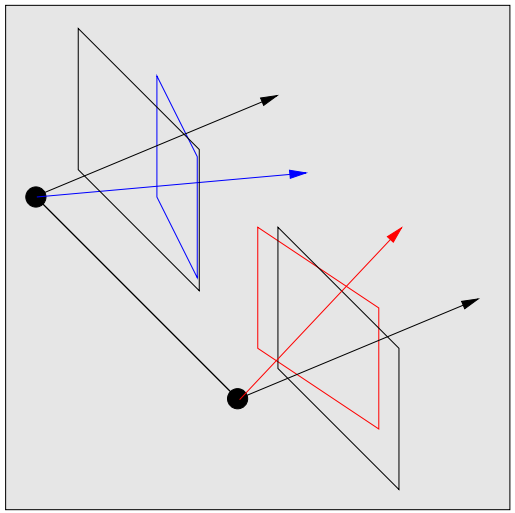
- Top-down, model driven: eg, segment entities and match them
- Bottom-up, data driven: without knowledge of higher entities

We'll explore a dense, data-driven approach.

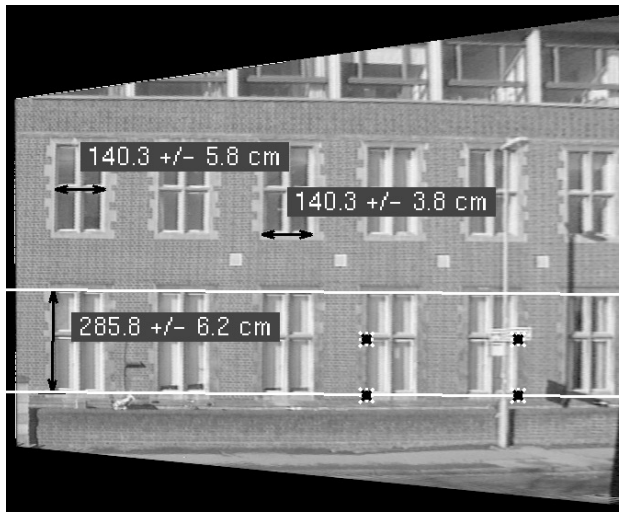
But first we will explore how to adjust the epipolar geometry for convenience — a process called image rectification

Adjusting epipolar geometry: Rectification

New optical axes are chosen to be coplanar and perpendicular to the base line, and the new image planes set with the same intrinsic matrix $\boldsymbol{K}_{rect}$



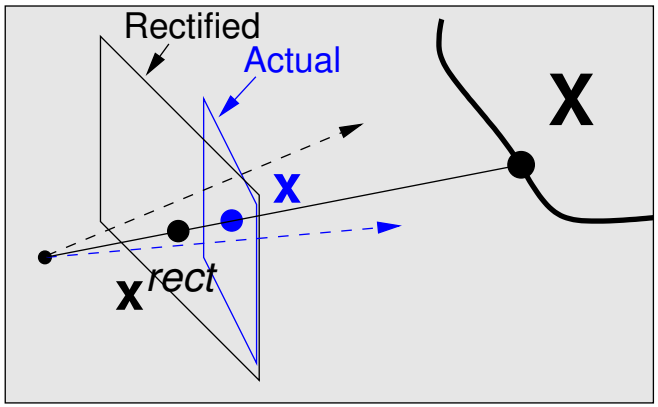
Actual Left image



Rectified

Adjusting epipolar geometry: Rectification

The actual and rectified frames differ by a rotation $\boldsymbol{R}$ only. Obviously this must be rotation about the optic centre.



The scene point $\mathbf{X}$ is projected to $\mathbf{x}$ in the actual image, and $\mathbf{x}^{rect}$ in the rectified image.

What is the relationship between them?

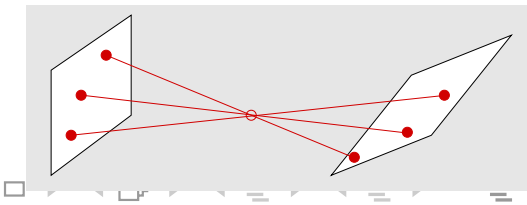
Allow $\mathbf{X}$ to be defined in the actual camera frame. The projections are:

$$\mathbf{x} \overset{P}{=} \boldsymbol{K}\,[\boldsymbol{I}|\mathbf{0}]\,\mathbf{X} = \boldsymbol{K}\mathbf{X}_{3\times 1} \quad \text{and} \quad \mathbf{x}^{rect} \overset{P}{=} \boldsymbol{K}^{rect}\,[\boldsymbol{R}|\mathbf{0}]\,\mathbf{X} = \boldsymbol{K}^{rect}\boldsymbol{R}\mathbf{X}_{3\times 1}$$

Eliminate $\mathbf{X}_{3\times 1}$ using $\mathbf{X}_{3\times 1} \overset{P}{=} \boldsymbol{K}^{-1}\mathbf{x}$ (so $\mathbf{X}$'s frame is actually irrelevant!)

$$\mathbf{x}^{rect} \overset{P}{=} \boldsymbol{K}^{rect}\boldsymbol{R}\boldsymbol{K}^{-1}\mathbf{x}$$

The 3×3 transformation is an homography $\boldsymbol{H} = \boldsymbol{K}^{rect}\boldsymbol{R}\boldsymbol{K}^{-1}$ — a plane-to-plane mapping through a projection centre (Lec1).

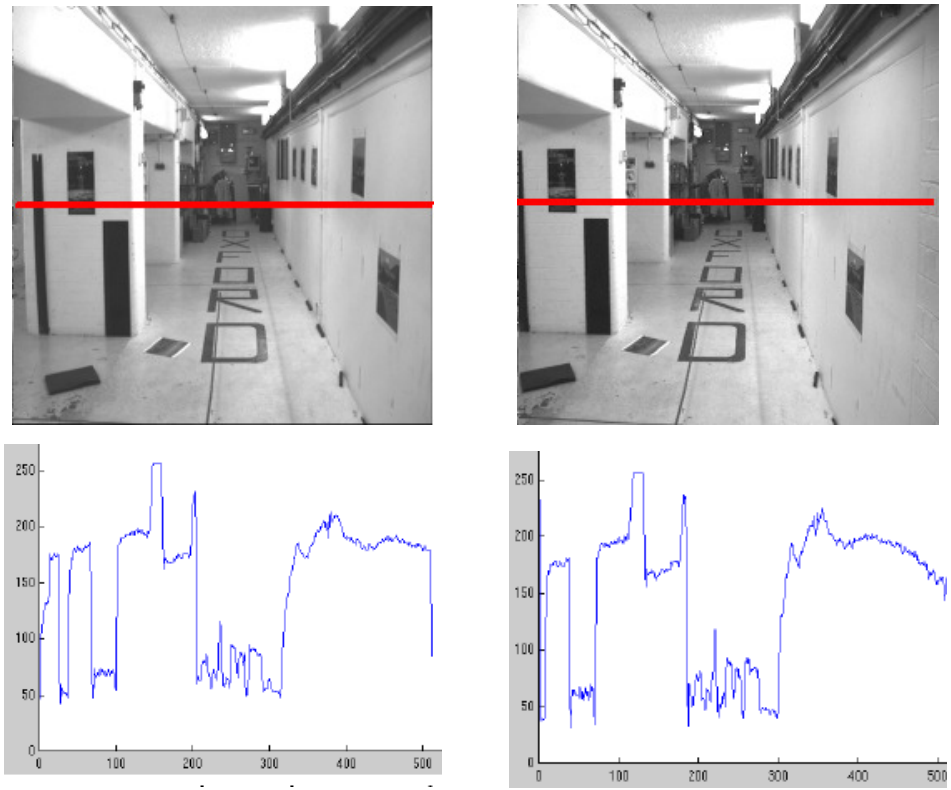


UNIVERSITY OF OXFORD

A correlation-based approach

The basic assumption in "short baseline" correspondence algorithms is that the image bands around the two epipolar lines generated by an epipolar plane are similar.

The example, for || cameras, shows that this is broadly the case, though there are regions of noise, ambiguity, and occlusion.

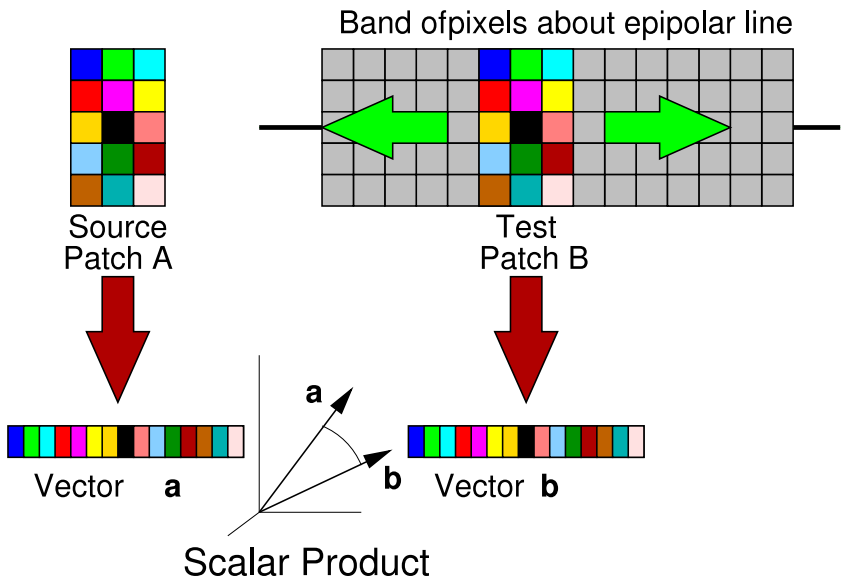


In Lecture 2, we saw that auto-correlation provides a method of quantifying the *self*-similarity of image patches. Here we use cross-correlation.

Zero-Normalized cross-correlation

Images are from different cameras — may be global gains and offsets
Model by $I' = \alpha I + \beta$ and use zero-normalized cross-correlation.

Step 1. Set source patch A around $\mathbf{x}$



Step 2. Subtract the patch mean $A_{ij} \leftarrow A_{ij} - \mu_A$

Step 3. For each $\mathbf{x}'$ on the epipolar line

3a. Generate patch B, and $B_{ij} \leftarrow B_{ij} - \mu_B$.

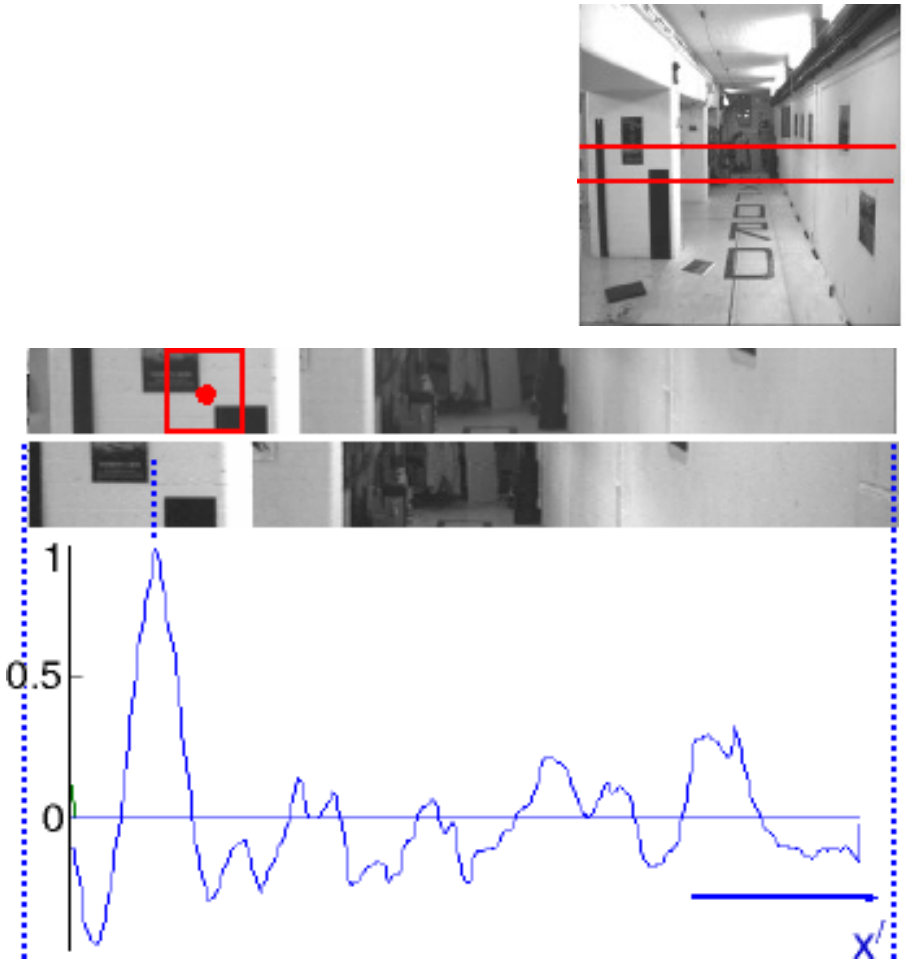
3b. Compute

$$NCC(\mathbf{x}, \mathbf{x}') = \frac{\sum_i \sum_j A_{ij} B_{ij}}{\sqrt{\sum_i \sum_j A_{ij}^2} \sqrt{\sum_i \sum_j B_{ij}^2}}$$

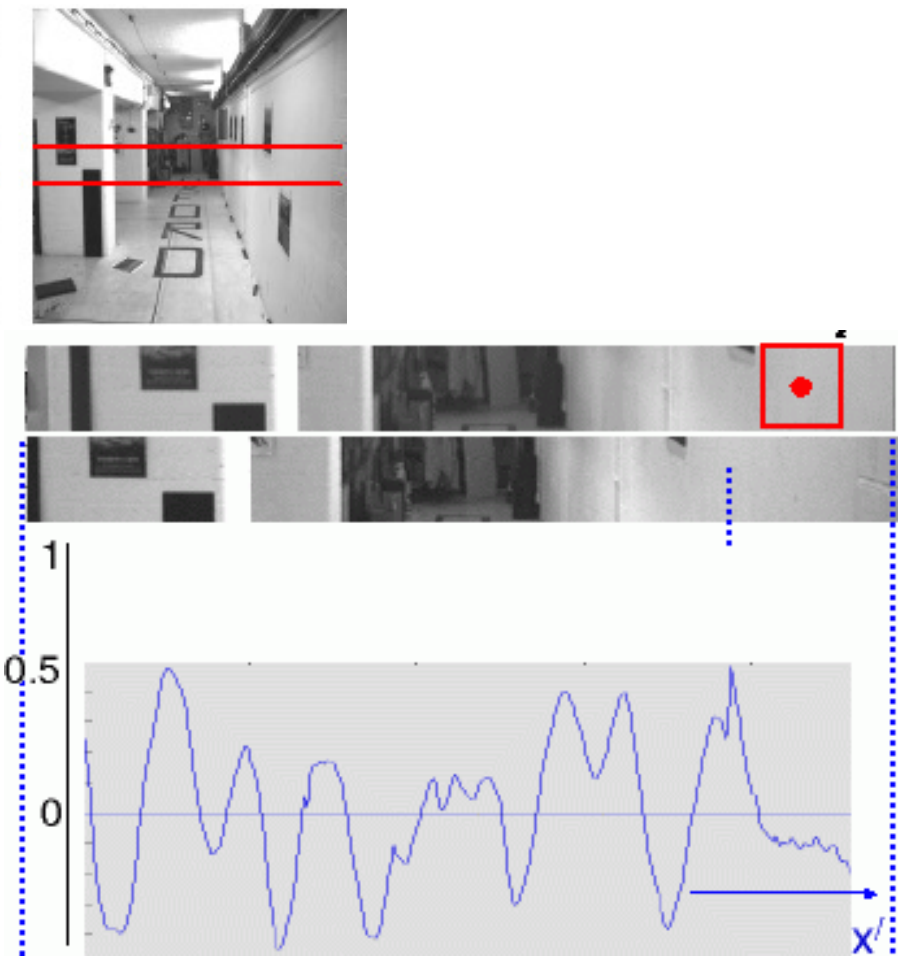
Useful analogy: Imagine the regions as vectors $A \rightarrow \mathbf{a}$ $B \rightarrow \mathbf{b}$.
$NCC \equiv \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$, a scalar product of UNIT vectors. So $-1 \leqslant NCC \leqslant 1$

UNIVERSITY OF OXFORD

Examples



Example 1: Strong correlation

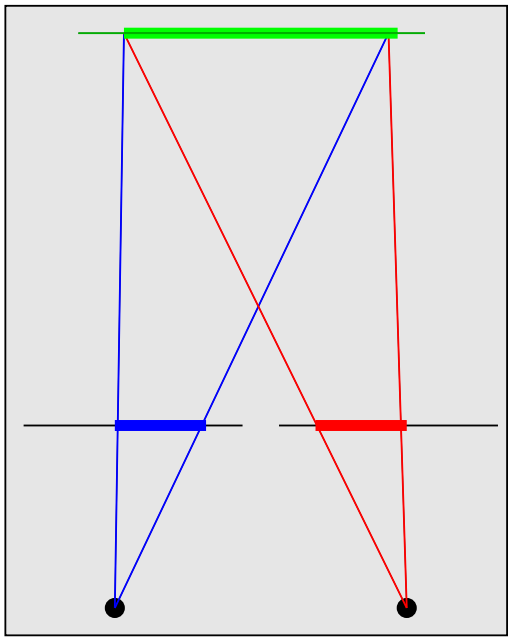


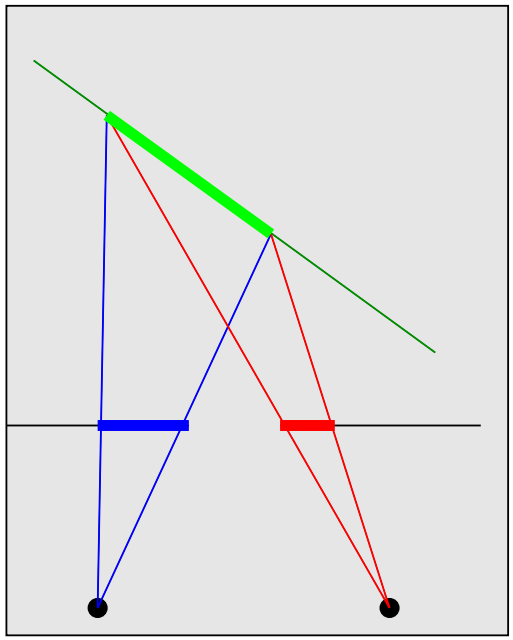
Example 2: Weak correlation

Examples /ctd

Why is cross-correlation a poor measure in Example 2?

1. The neighbourhood region does not have a distinctive spatial intensity. Weak auto-correlation ⇒ambiguous cross-correlation
2. Foreshortening effects — perspective distortion.
   Using Correlation assumes minimal appearance change
   which favours fronto-parallel surfaces.





Outline of a dense correspondence algorithm

Algorithm

1. Rectify images in $\mathcal{C}$, $\mathcal{C}'$
2. For each pixel $\mathbf{x}$ in image $\mathcal{C}$:
   - 2.1 Compute NCC for each pixel $\mathbf{x}'$ along epipolar line $\mathbf{l}'$ in image $\mathcal{C}'$
   - 2.1 Choose the match $\mathbf{x}'$ with the highest NCC

Parameters to adjust

- size of neighbourhood patches
- limit on maximum disparity $(\mathbf{x}' - \mathbf{x})$

Constraints to apply

- uniqueness of match
- match ordering
- smoothness of disparity field (disparity gradient limit)
- figural continuity.

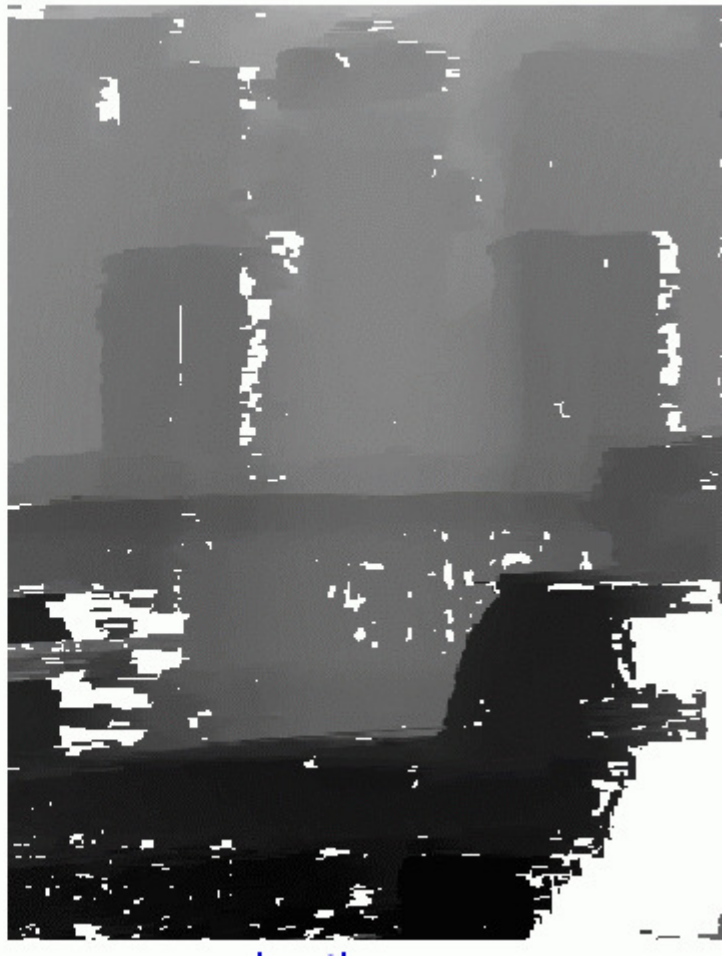
Limitations

- scene must be textured, and
- largely fronto-parallel (related to d.g. limit)

Example: Right and Left Images

Cross-eye fusable

Results: Left Image and 3D Range Map



Results: 3D Reconstruction

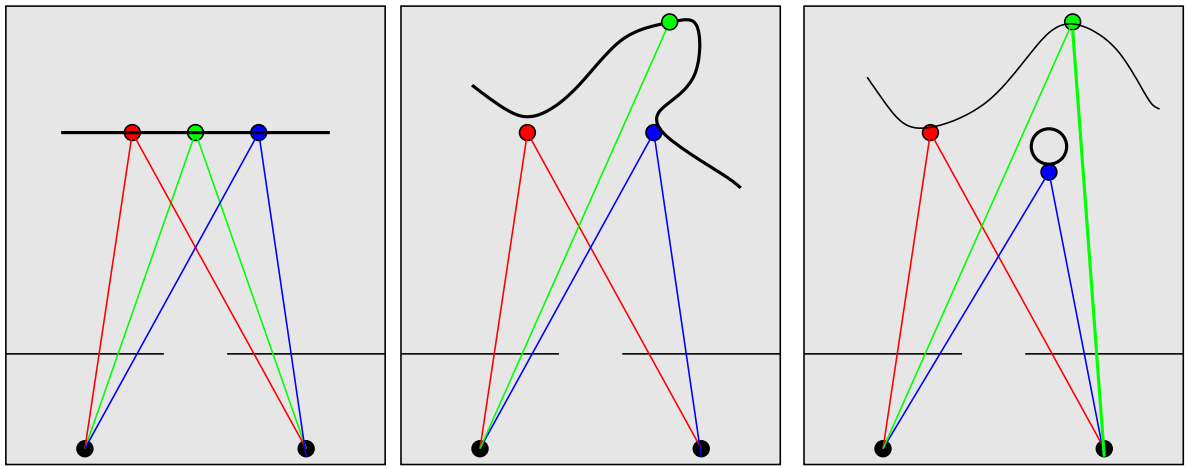
The "Other Constraints" on correspondence

Uniqueness of match: Promote 1-to-1 matches, but there can be 0-to-1 and 1-to-0 because of occlusion.

Ordering: From a continuous opaque surface it is not possible to change the match order.

Disparity smoothness: This is stronger than the previous constraint and favours smooth surfaces with no sudden changes in depth.

Figural continuity: The disparity field along an epipolar line should not be much different from that along a neighbouring epipolar line.



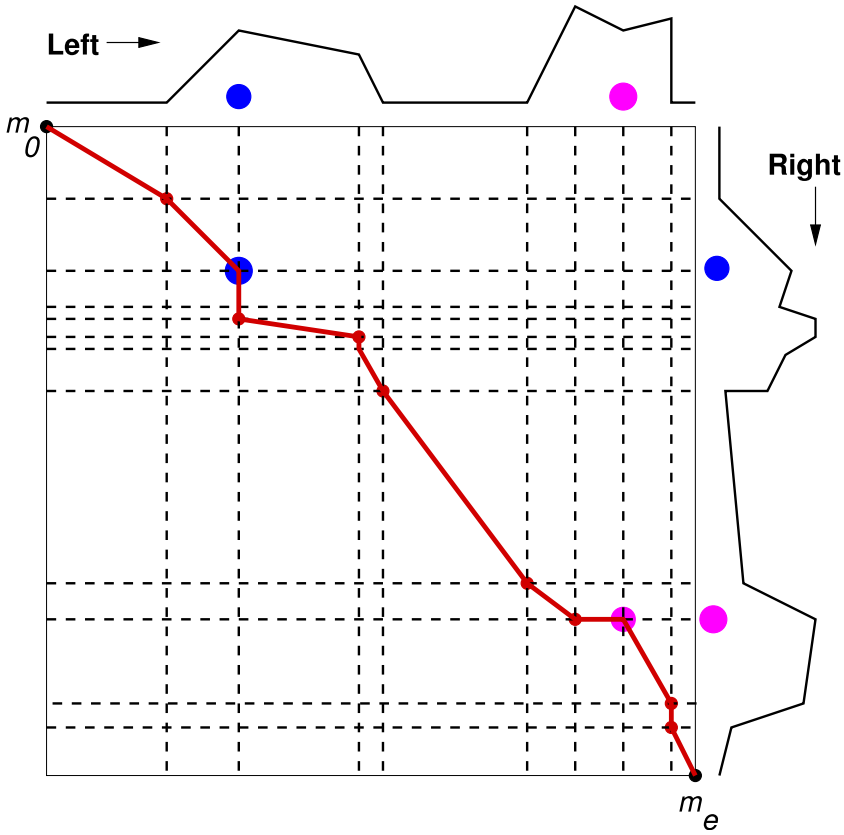
UNIVERSITY OF OXFORD

Correspondence using Dynamic programming /ctd

These constraints can be imposed using costs and optimization —

eg, using Dynamic Programming (Ohta and Kanade 1985, Polleyfeys *et al* 2000).

1. Imagine the raster intensities from the plotted out on two sides of a grid.
2. Significant features (edges, corners, or pixels) used as key potential matches
3. Crossing dashed lines represent potential matches at nodes.



Correspondence using Dynamic programming /ctd

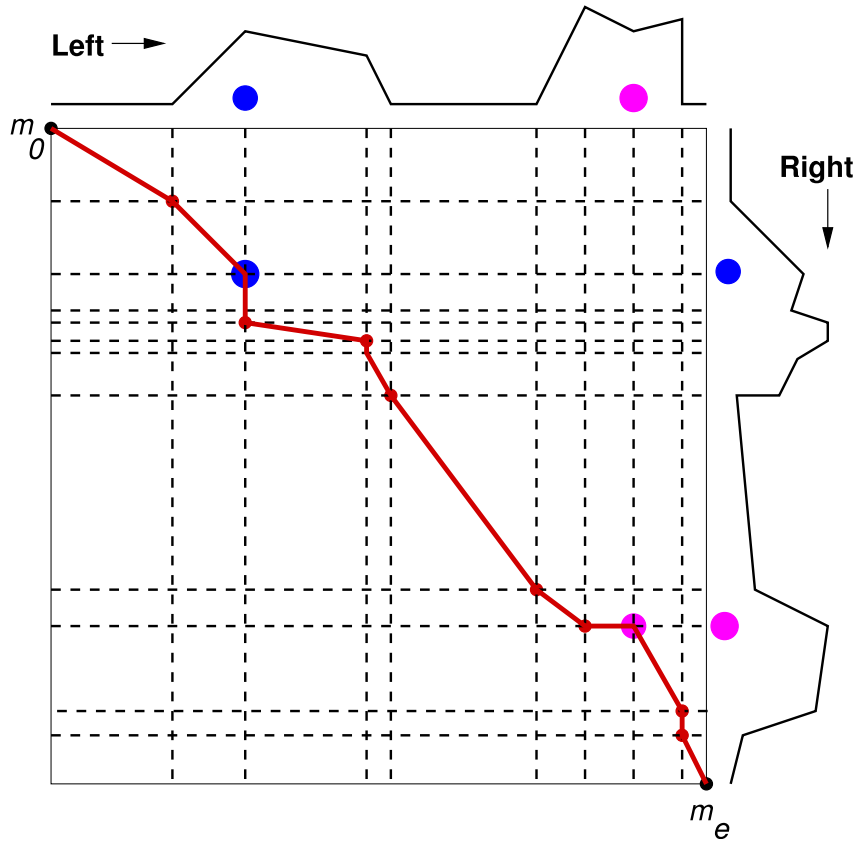
Aim: Construct a minimum cost path from top-left to bottom-right of the grid, which:

Ordering: has negative gradient in South-East quadrant

Uniqueness: rarely goes across or down

Completeness: visits as many potential nodes as possible

Figural Cont: is similar to paths constructed for neighbouring rasters.



Correspondence using Dynamic programming /ctd

Costs: Some costs associated with nodes, others with links.

Nodes: **Eg Similarity**

Features should be similar in images $\mathcal{C}$,$\mathcal{C}'$.

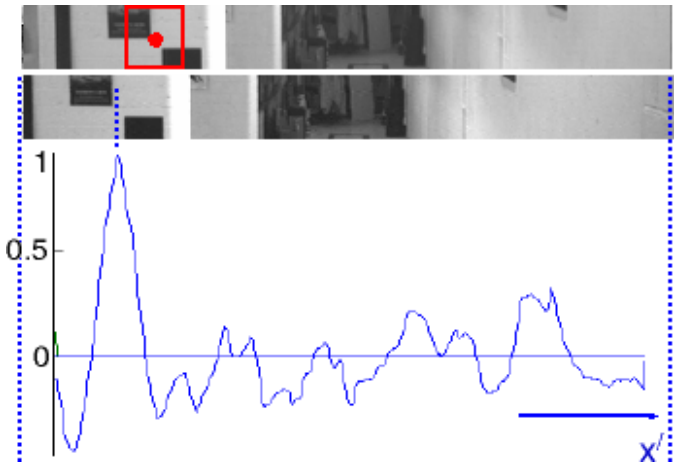
Eg, for NCC between patches

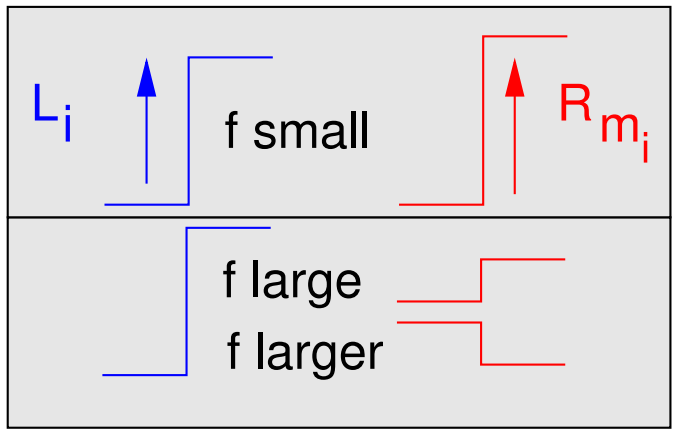
$$f_i(m_i) = \alpha(1 - NCC)^2$$

If you were using edges as features, you could use the constrast similarity

$$f_i(m_i) = \alpha(\mathcal{C}_i - \mathcal{C}'_{m_i})^2$$

α is a factor chosen empirically to trade-off the different costs.

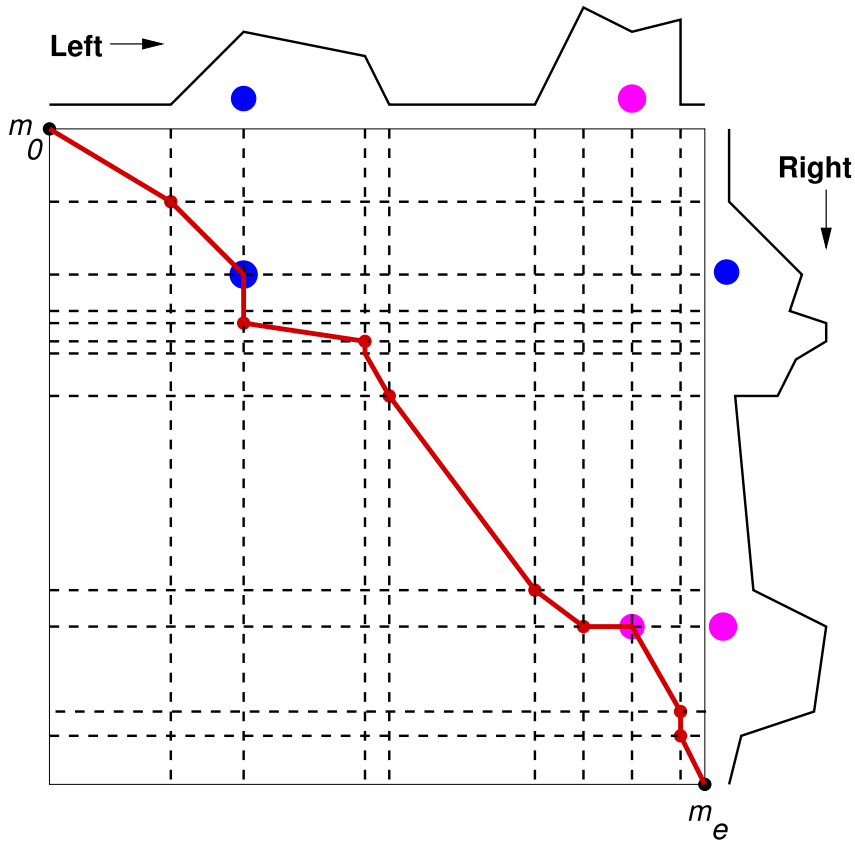




Correspondence using Dynamic programming /ctd

Links: Eg, uniqueness

Devise a link cost $g(m, m')$ between **matches** m and m', where
$(m, m') \equiv (i \rightarrow i', j \rightarrow j')$



Cost is

$$g(m, m') = \begin{cases} 0 & \text{if } i < i' \ \& \ j < j' \quad \text{OK} \\ \beta & \text{if } i = i' \ \& \ j < j' \quad \text{discourage} \\ \beta & \text{if } i < i' \ \& \ j = j' \quad \text{discourage} \\ \infty & \text{otherwise} \quad \text{verboten} \end{cases}$$

Correspondence using Dynamic programming /ctd

Choose m_0 to m_e to **Minimize sum of node and link costs**

$$C(m_0, m_e) = \sum_{i=0}^{e} f(m_i) + \sum_{i=0}^{e-1} g(m_i, m_{i+1})$$

m_0 and m_e almost certainly don't match, so why are they included?
... Because they impose ordering!

The minimum cost $C(m_0, m)$ of a path from m_0 to m is defined recursively

$$C(m_0, m) = \min_{p \in \mathcal{N}(m)} [f(m) + g(p, m) + C(m_0, p)]$$

DP replaces *simultaneous* minimization over all variables by a *separate* minimizations involving just 1 variable (ie, p).

Correspondence using Dynamic programming /ctd

Algorithm involves exploring a graph or net:

For putative matches at level 1:
$C(m_0, m_1) = g(m_0, m_1)$.

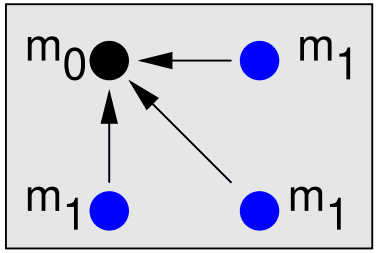
For putative matches at level n:
Find which p in the neighbourhood $\mathcal{N}(m_n)$ of match m_n gives

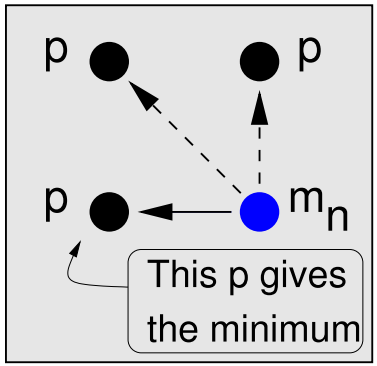
$$C(m_0, m_n) = \min_{p \in \mathcal{N}(m_n)} [f(m_n) + g(p, m_n) + C(m_0, p)]$$

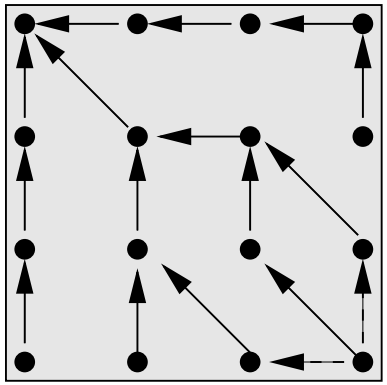
Keep only the minimum route/cost

Finally, **for putative matches at level $MN - 1$:**
Compute $C(m_e)$, then trace back.

Note that at level n all the $C(m_0, p)$'s required are already derived and in place.







4.3 Triangulation: bursting into 3D

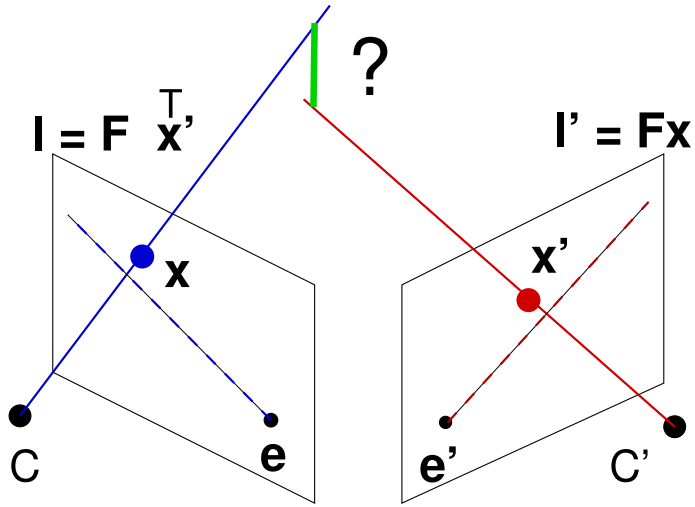
Given

a. exact proj. matrices $\boldsymbol{P}$, $\boldsymbol{P}'$

b. positionally noisy $\mathbf{x} \leftrightarrow \mathbf{x}'$

Compute

the 3D point $\mathbf{X}$.



Difficulties:

a. Back-projected rays will not meet, because

b. Measured points don't lie properly on epipolar lines

There are several approaches ... Vector Midpoints Linear Triangulation

UNIVERSITY OF OXFORD

Approach I: Vector Midpoint

Assume $\mathbf{a}, \mathbf{b}, \mathbf{c}$ defined in $\mathcal{C}$ and use 3 components.

$\Rightarrow \mathbf{X}_{3\times 1} = (\mathbf{0} + \alpha\mathbf{a} + \mathbf{c} + \beta\mathbf{b})/2$ when

$\alpha\mathbf{a} + \gamma(\mathbf{a} \times \mathbf{b}) = \mathbf{c} + \beta\mathbf{b}$

$\alpha(\mathbf{a}\cdot\mathbf{a}) = (\mathbf{a}\cdot\mathbf{c}) + \beta(\mathbf{a}\cdot\mathbf{b})$ and

$\alpha(\mathbf{a}\cdot\mathbf{b}) = (\mathbf{b}\cdot\mathbf{c}) + \beta(\mathbf{b}\cdot\mathbf{b})$

Hence with $D = (\mathbf{a}\cdot\mathbf{a})(\mathbf{b}\cdot\mathbf{b}) - (\mathbf{a}\cdot\mathbf{b})^2$

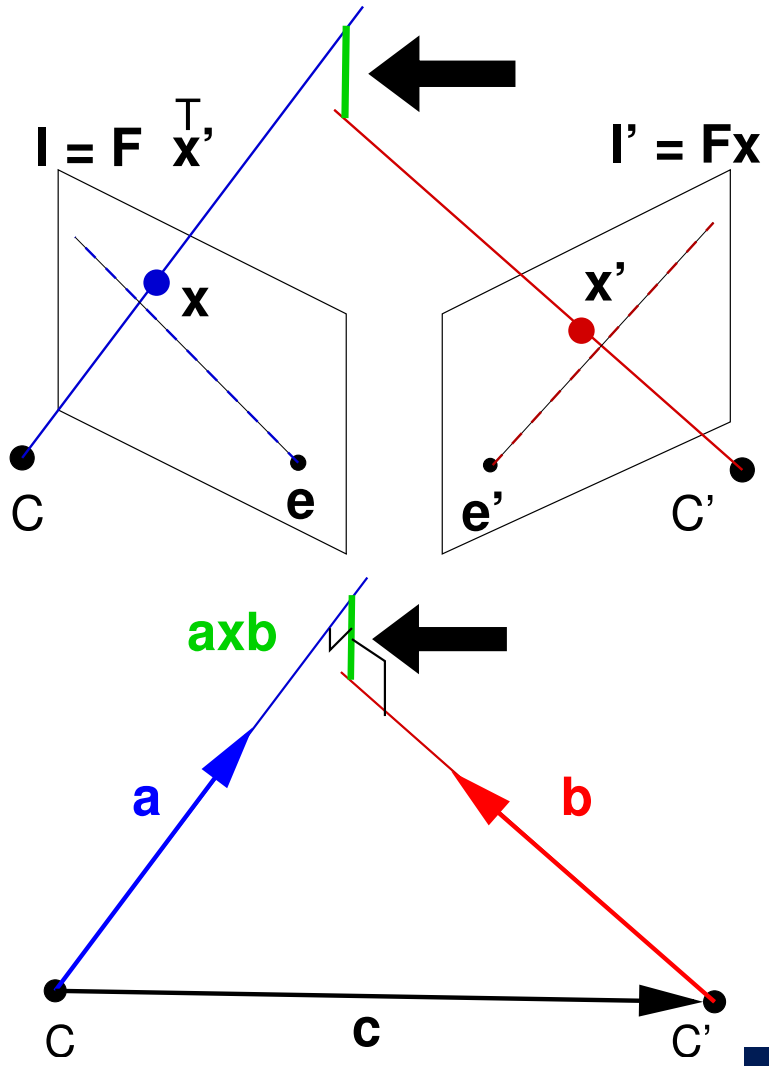
$\alpha = ((\mathbf{b}\cdot\mathbf{b})(\mathbf{a}\cdot\mathbf{c}) - (\mathbf{a}\cdot\mathbf{b})(\mathbf{b}\cdot\mathbf{c}))/D$

$\beta = ((\mathbf{a}\cdot\mathbf{b})(\mathbf{a}\cdot\mathbf{c}) - (\mathbf{a}\cdot\mathbf{a})(\mathbf{b}\cdot\mathbf{c}))/D$

Now specify the vectors in the $\mathcal{C}$ frame.

$$\mathbf{a} = \begin{bmatrix} \boldsymbol{K}^{-1}\mathbf{x} \\ 0 \end{bmatrix}; \mathbf{b} = \begin{bmatrix} \boldsymbol{R}^{-1}\boldsymbol{K}'^{-1}\mathbf{x}' \\ 0 \end{bmatrix};$$

$$\mathbf{c} = \begin{bmatrix} -\boldsymbol{R}^{-1}\mathbf{t} \\ 1 \end{bmatrix}.$$



Assuming all vectors in $\mathcal{C}$.

Approach I: Vector Midpoint /ctd

Why is $\mathbf{c} = \begin{bmatrix} -\boldsymbol{R}^{-1}\mathbf{t} \\ 1 \end{bmatrix}$?

First, recall that $\mathbf{x}' = \boldsymbol{K}'[\boldsymbol{R}|\mathbf{t}]\mathbf{X}$ because

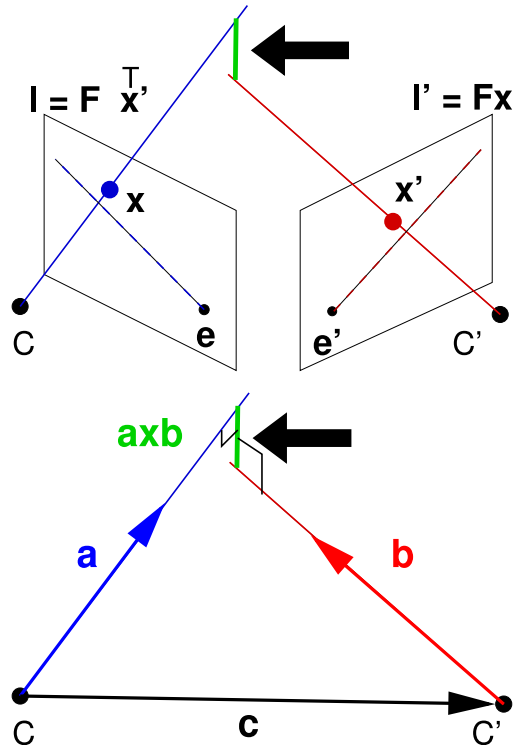
$$\mathbf{X}' = \begin{bmatrix} \boldsymbol{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\Rightarrow \mathbf{X} = \begin{bmatrix} \boldsymbol{R}^{-1} & -\boldsymbol{R}^{-1}\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}'$$

The camera centre of $\mathcal{C}'$ is at $\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$ in $\mathcal{C}'$, and so in $\mathcal{C}$

$$\mathbf{c} = \begin{bmatrix} \boldsymbol{R}^{-1} & -\boldsymbol{R}^{-1}\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} -\boldsymbol{R}^{-1}\mathbf{t} \\ 1 \end{bmatrix}$$

All vectors in $\mathcal{C}$.



Approach II: Linear Triangulation

Use the equations $\mathbf{x} = \boldsymbol{P}\mathbf{X}$ and $\mathbf{x}' = \boldsymbol{P}'\mathbf{X}$ to solve for $\mathbf{X}$

Write

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{1\top} \\ \mathbf{p}^{2\top} \\ \mathbf{p}^{3\top} \end{bmatrix}$$

Eliminate unknown scale in $\lambda\mathbf{x} = \boldsymbol{P}\mathbf{X}$ by forming a cross product $\mathbf{x} \times (\boldsymbol{P}\mathbf{X}) = \mathbf{0}$. Its three components are

$$\begin{aligned} x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) &= 0\ . \end{aligned}$$

Rearrange just the first two as

$$\begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \end{bmatrix}_{2\times 4} \mathbf{X}_{4\times 1} = \mathbf{0}_{2\times 1}$$

UNIVERSITY OF OXFORD

Approach II: Linear Triangulation /ctd

Write the same for the second camera

$$\left[\begin{array}{c} x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{array}\right]_{2\times 4} \mathbf{X}_{4\times 1} = \mathbf{0}_{2\times 1}$$

Put one above the other to obtain $\boldsymbol{A}_{4\times 4}\mathbf{X}_{4\times 1} = \mathbf{0}_{4\times 1}$ where

$$\boldsymbol{A}_{4\times 4}\mathbf{X}_{4\times 1} = \left[\begin{array}{c} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y'\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{array}\right] \mathbf{X}_{4\times 1} = \mathbf{0}_{4\times 1}$$

This is a null-space problem, from which $\mathbf{X}$ can be recovered up to scale

Advantage: extends to more than two views.

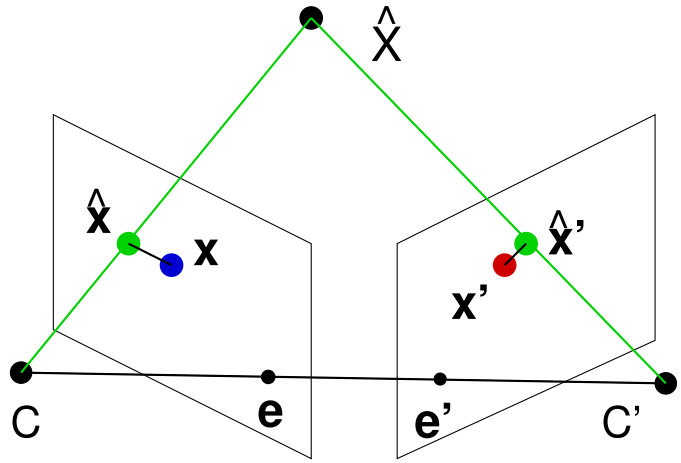
Disadvantage: this is a minimization, but the quantity being minimized is not related to realistic error.

UNIVERSITY OF OXFORD

Approach III: Minimizing a geometrical/statistical error

Project estimated scene position

$$\hat{\mathbf{x}} = \boldsymbol{P}\hat{\mathbf{X}} \quad \hat{\mathbf{x}}' = \boldsymbol{P}'\hat{\mathbf{X}}$$



Measure Euclidean displacements

$$d(\mathbf{x}, \hat{\mathbf{x}}) \quad d(\mathbf{x}', \hat{\mathbf{x}}')$$

Compute the cost

$$C(\hat{\mathbf{X}}) = d^2(\mathbf{x}, \hat{\mathbf{x}}) + d^2(\mathbf{x}', \hat{\mathbf{x}}')$$

Adjust $\hat{\mathbf{X}}$ to minimize the cost.

If the measurement noise is a zero-mean Gaussian $\mathcal{N}(0, \sigma^2)$, then this is the Maximum Likelihood Estimator of $\mathbf{X}$.

It looks like a minimization over 3 parameters ...
... but it can be reduced to a single parameter.

Can you see how?

A note on Triangulation depth error

Triangulation involves intersecting rays in the epipolar plane. Depth uncertainty must depend on the uncertainty in point location along the epipolar lines.

Analyze parallel camera formula

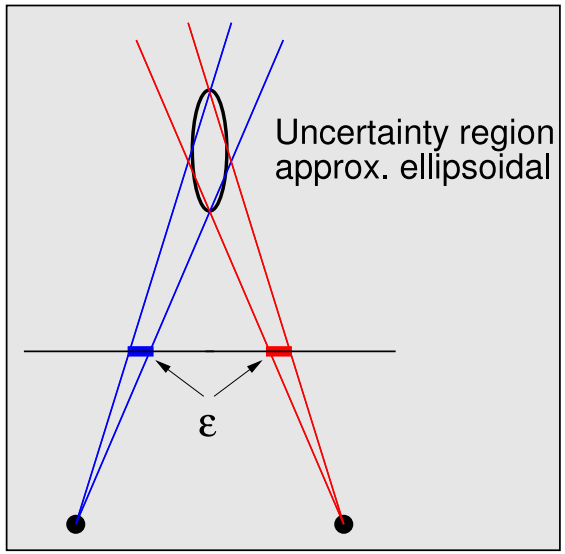
$$\frac{1}{Z} = \frac{1}{ft_x}(x' - x)$$

$$\Rightarrow d\left(\frac{1}{Z}\right) = -\frac{dZ}{Z^2} = \frac{1}{ft_x}(dx' - dx)$$

$$\Rightarrow \left(\frac{\delta Z}{Z^2}\right)^2 \approx \left(\frac{1}{ft_x}\right)^2 \left((\delta x')^2 + (\delta x)^2\right)$$

Conclusion is:
Depth Error: $\delta Z \approx Z^2\sqrt{2}\epsilon/ft_x$
where ϵ is typical image measurement error.

Reminder ...
$f = f(u, v...)$, $\Rightarrow$1st-order Taylor
$\delta f = \frac{\partial f}{\partial u}\delta u + \frac{\partial f}{\partial v}\delta v + \ldots$
Assuming no correlation, we add errors in quadrature:
$\Rightarrow (\delta f)^2 = \left(\frac{\partial f}{\partial u}\delta u\right)^2 + \left(\frac{\partial f}{\partial v}\delta v\right)^2 + \ldots$



Summary

4.1 Introduction

4.2 The Correspondence Problem

Rectification
Zero Normalized cross correlation
Outline of a dense stereo algorithm
Incorporating contraints using dynamic programming.

4.3 Triangulation for Depth Recovery

Vector midpoints
Linear method
MLE, driven by a one parameter variation of the epipolar pencil.

UNIVERSITY OF OXFORD