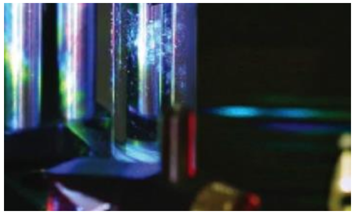
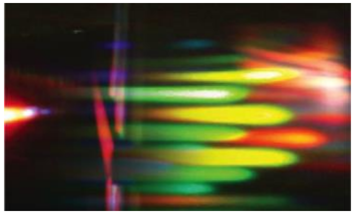
Imperial College
London





COMP70058 Computer Vision

Lecture 15 – Computational Stereo (Part 2)

Stamatia (Matina) Giannarou, PhD
The Hamlyn Centre for Robotic Surgery

stamatia.giannarou@imperial.ac.uk

The Hamlyn Centre
for Robotic Surgery

Contents

- Computation of Fundamental Matrix
- Constraints for the Fundamental Matrix
- Stereo Constraints and Priors
- Stereo Image Rectification and Homography
- Application Examples

A Quick Summary of What's Covered So Far

- The fundamental matrix satisfies the condition that for any pair of corresponding points **x**↔**x'** in the two images

$$\mathbf{x}'^T \mathbf{F}\mathbf{x} = 0$$

$$\begin{aligned}\mathbf{F} &= \mathbf{e}' \times (\mathbf{P}'\mathbf{P}^+) \\ &= [\mathbf{P}'\mathbf{C}]_\times \mathbf{P}'\mathbf{P}^+ \\ &= [\mathbf{K}'\mathbf{t}]_\times \mathbf{K}'\mathbf{R}\mathbf{K}^{-1}\end{aligned}$$

- **Properties of F include:**
  - **F** has 7 degrees of freedom (3x3 – 1 (homogeneous) -1(rank 2))
  - If **F** is the fundamental matrix for camera pair (**P, P'**), then $\mathbf{F}^T$ is for (**P', P**), and $\mathbf{l}=\mathbf{F}^T\mathbf{x}'$ represents the epipolar line for **x'** in the second image

$$\mathbf{F}^T\mathbf{e}' = 0, \text{ and } \mathbf{F}\mathbf{e} = 0$$

Estimate the Fundamental Matrix

- Given corresponding points in image coordinates (they can be found using feature matching), and the unknown fundamental matrix can be written as

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0, \quad \mathbf{x} = [\begin{array}{ccc} x & y & 1 \end{array}]^T \quad \mathbf{x}' = [\begin{array}{ccc} x' & y' & 1 \end{array}]^T$$

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0$$

Estimate the Fundamental Matrix

- For each point-pair, we have one equation with 9 unknowns, or more precisely 8 with un-recoverable scale, so we need at least 8 point-pairs to recover **F**
- By stacking all the elements of **F** together as a long vector, we have

$$\begin{bmatrix} x_1 x_1' & y_1 x_1' & x_1' & x_1 y_1' & y_1 y_1' & y_1' & x_1 & y_1 & 1 \\ x_2 x_2' & y_2 x_2' & x_2' & x_2 y_2' & y_2 y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n x_n' & y_n x_n' & x_n' & x_n y_n' & y_n y_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

$$\mathbf{A}\,\Psi = 0$$

Estimate the Fundamental Matrix

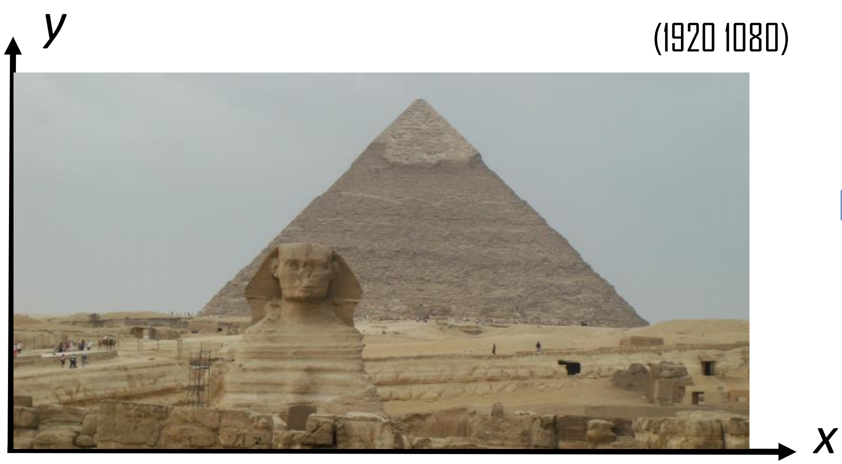
- To solve this, a number of different techniques can be used, most commonly via least squares technique as well as non-linear methods
- Before applying any of these techniques, however, it is necessary to perform normalisation of matrix A, as you can see some of the columns can have very **large values** and some **relatively small**, so far example, by using least-squares approach can result in poor numerical results

$$\begin{bmatrix} x_1 x_1' & y_1 x_1' & x_1' & x_1 y_1' & y_1 y_1' & y_1' & x_1 & y_1 & 1 \\ x_2 x_2' & y_2 x_2' & x_2' & x_2 y_2' & y_2 y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n x_n' & y_n x_n' & x_n' & x_n y_n' & y_n y_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

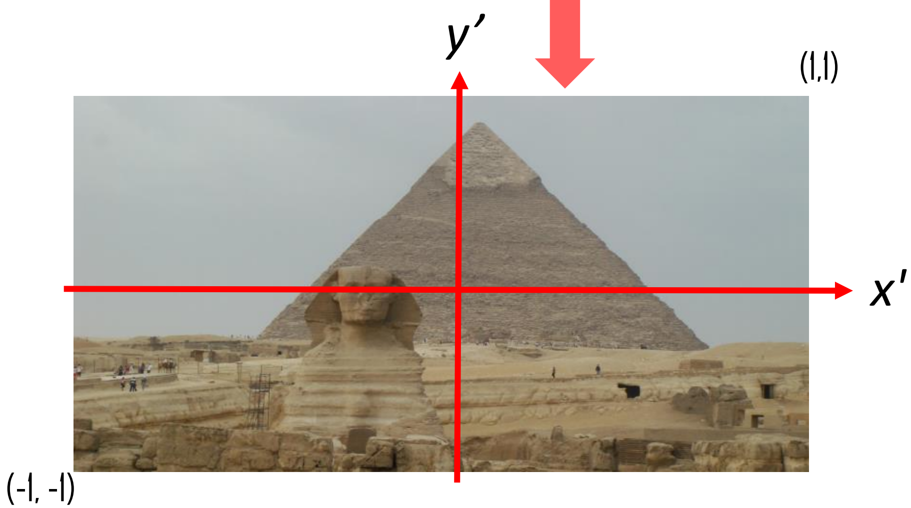
$$\mathbf{A}\,\Psi = 0$$

Estimate the Fundamental Matrix

- Normalisation can be done by mapping the image coordinates to [-1,1] for both dimensions, so for 1080p video frames (1920x1080), we have the following (the original idea was proposed and justified by Hartley, PAMI 1997 to demonstrate how this can yield better least squares results)



$$\begin{bmatrix} \frac{2}{1920} & 0 & -1 \\ 0 & \frac{2}{1080} & -1 \\ 0 & 0 & 1 \end{bmatrix}$$



Constraints for the Fundamental Matrix

- Another problem with direct least-squares approach is that the recovered matrix **F** may not satisfy the constraint of the fundamental matrix, recall for **F**, it is rank 2 and

$$\mathbf{F}^T\mathbf{e}' = 0, \text{ and } \mathbf{F}\mathbf{e} = 0 \text{ and } \det\mathbf{F}=0$$

- This can be done by using the SVD (Singular Value Decomposition) approach to derive **F** that is rank 2

$$F = U\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \underbrace{\sigma_3}_{\text{Set to Zero}} \end{bmatrix} V^T = U_1\sigma_1V_1^T + U_2\sigma_2V_2^T + \underbrace{\boxed{U_3\sigma_3V_3^T}}_{\text{Discard}}$$

- The result is effectively to find an approximate **F'** that is as close as possible to the recovered **F** and satisfies the singularity constraints as stated above

Computational Steps for Estimating F

- A naïve algorithm for calculating **F** can be formulated to include the following main steps
  - Feature extraction, ensure features are salient and stable
  - Calculate a set of potential matches (to begin with you need at least 8)
  - **Do**
    - Select matched feature-pairs
    - Calculate an initial estimation for **F** from AF=0 using SVD
    - Re-projection of feature points using the estimated **F**
    - Determine how many points are re-projected with large errors
    - Remove those outliers and look for additional matches
  - **Until** (the number of outliers is less than a pre-set %)
  - Refine F based on all correct matches

Matlab:

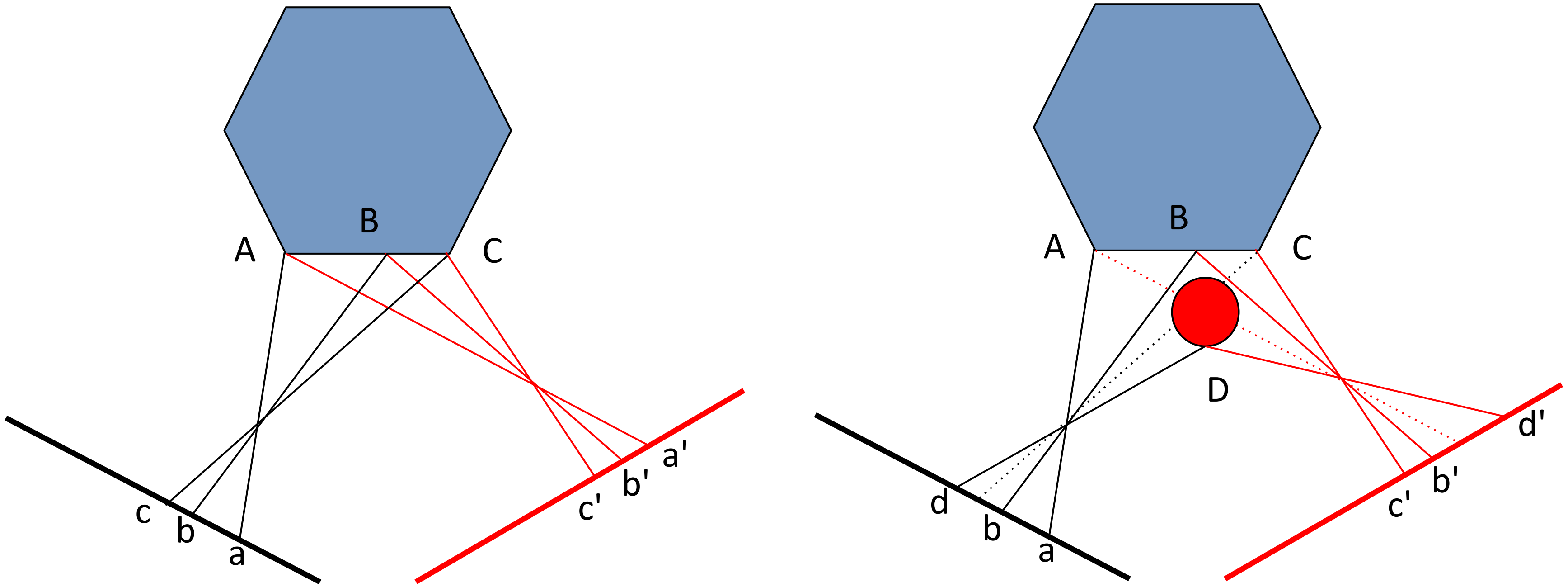
```
[U, S, V] = svd(A);
f = V(:, end);
F = reshape(f, [3 3])';
```

Stereo Constraints and Priors

- Even with the use of epipolar constraints after successful recovery of the fundamental matrix, stereo reconstruction based on feature matching can still be error prone
- This can be due to the absence of reliable features or structures with a lot repeated structure points, and therefore makes robust matching difficult
- **Uniqueness Constraint** – for any point in one image, there should be at most one matching point in the other image
- **Ordering Constraint** – the corresponding points should be in the same order in both views after projection
- **Smoothness Constraint** – disparity values change slowly for most part of the image

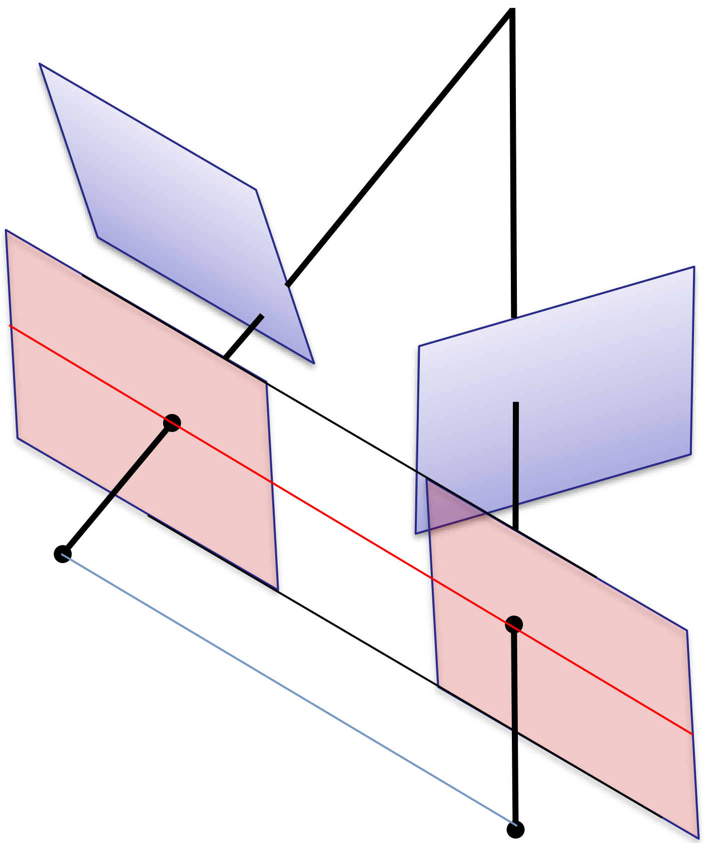
Ordering Constraint for Stereo

- **Ordering Constraint** – the corresponding points should be in the same order in both views after projection



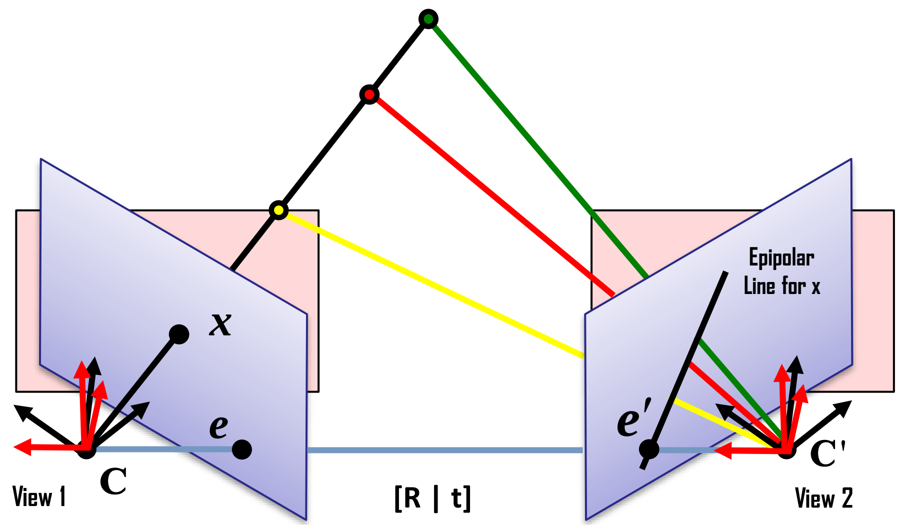
Stereo Image Rectification

- Computationally searching along epipolar lines can be expensive as most of them are obliquely oriented
- Numerical error can also impose problems
- Stereo image rectification is the process of re-projecting image planes onto a common one that is parallel to the base line linking camera centres
- After rectification, all epipolar lines will then become horizontal lines, which makes the search algorithm particularly simple
- The rectification is effectively a simple image warping process, which is easily done by graphics card, particularly modern GPUs
- What's the fundamental matrix for the image plane after rectification?



Reference - Andrea Fusiello et al, A compact algorithm for rectification of stereo pairs, Machine Vision and Applications (2000) 12: 16–22

Stereo Image Rectification



- Rotate the left camera such that its image plane is parallel to the baseline of the system (i.e. the epipole goes to infinity along the x-axis)
  - Make the new x axis along the direction of the baseline (i.e. **(C-C')/|C-C'|**), keep the centre of projection C unchanged
  - Make the new y axis to be orthogonal to the new x and the old z axis, which is along the old optical axis (use cross product)
  - Make the new z axis orthogonal to the baseline and the new y axis (use cross product again)
  - This defines a rotation matrix for the left camera, $\mathbf{R}_{\mathbf{rect}}$
- Rotation of the right camera also to be parallel by keeping its centre of projection, the projection matrix is simply $\mathbf{R}_{\mathbf{rect}}\mathbf{R}^{\mathbf{T}}$

Homography for Rotation

- When camera is only rotated but not translated, there is a simple map between the rotated coordinates and the original one
- Original (home) position

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

- After rotation R, we have

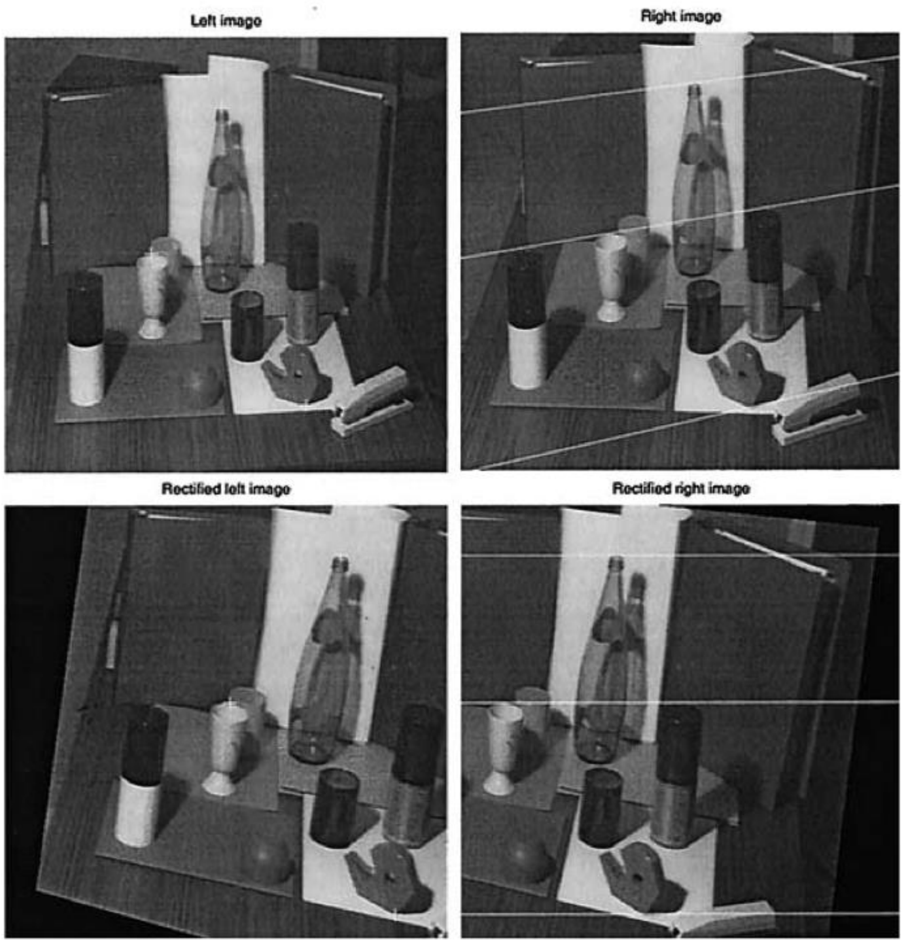
$$\mathbf{x}' = \mathbf{K}\left[\mathbf{R} \mid \mathbf{0}\right]\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{KRX}$$

- Therefore, we have

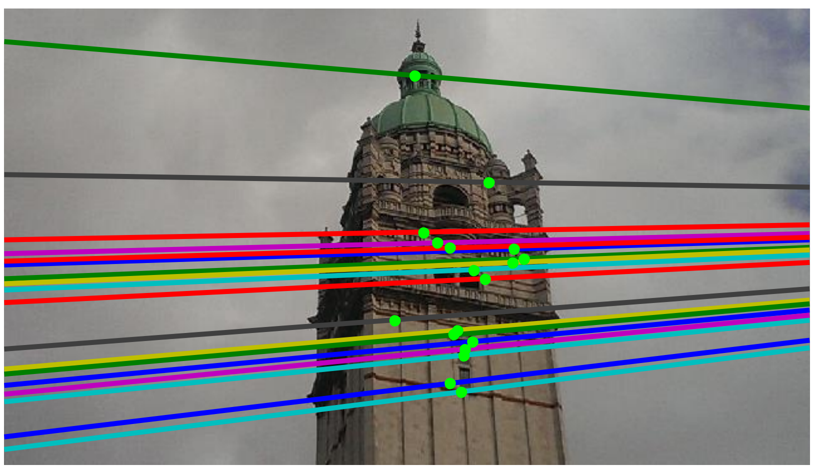
$$\mathbf{x}' = \underbrace{\left(\mathbf{KRK}^{-1}\right)}_{\text{homography}}\mathbf{x}$$

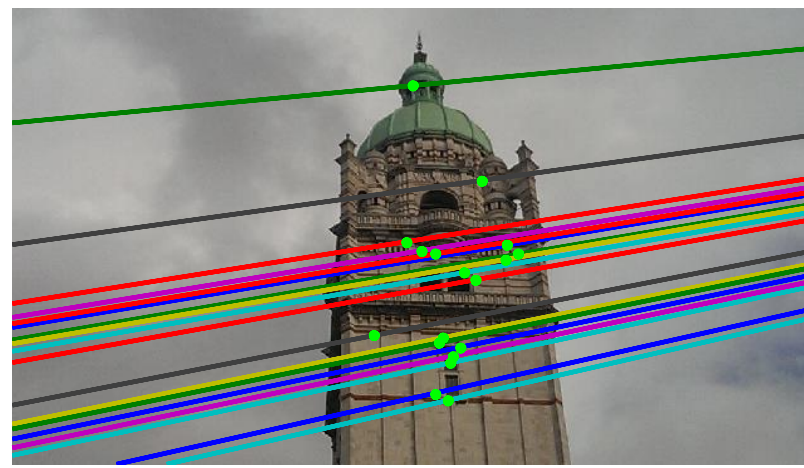
Image Rectification Examples

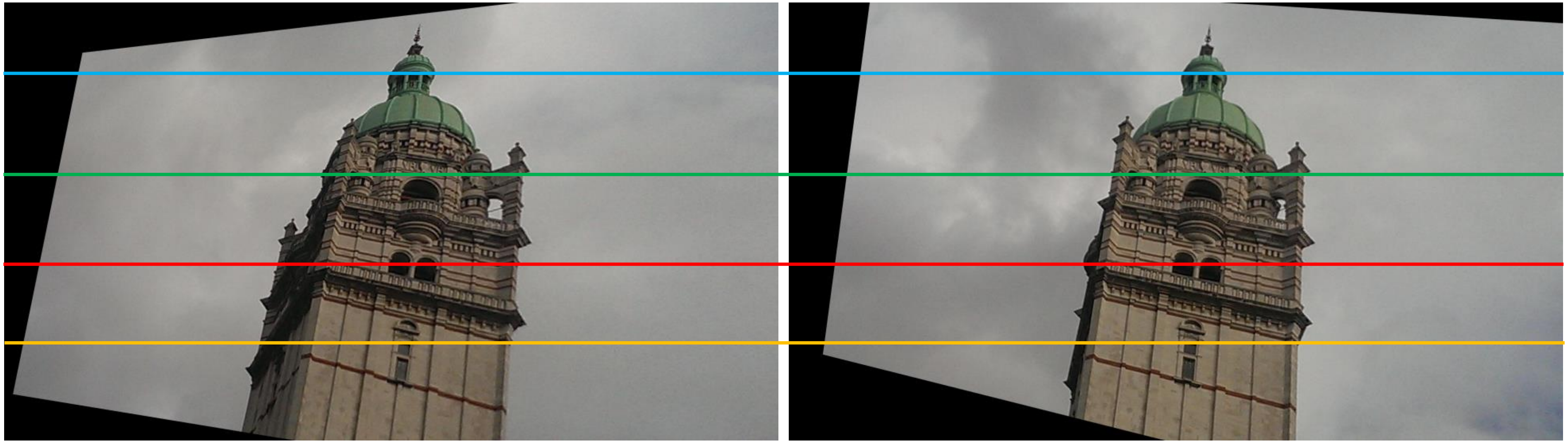
- Original result by **Andrea Fusiello, Emanuele Trucco, and Alessandro Verri**, A compact algorithm for rectification of stereo pairs, Machine Vision and Applications (2000) 12: 16–22



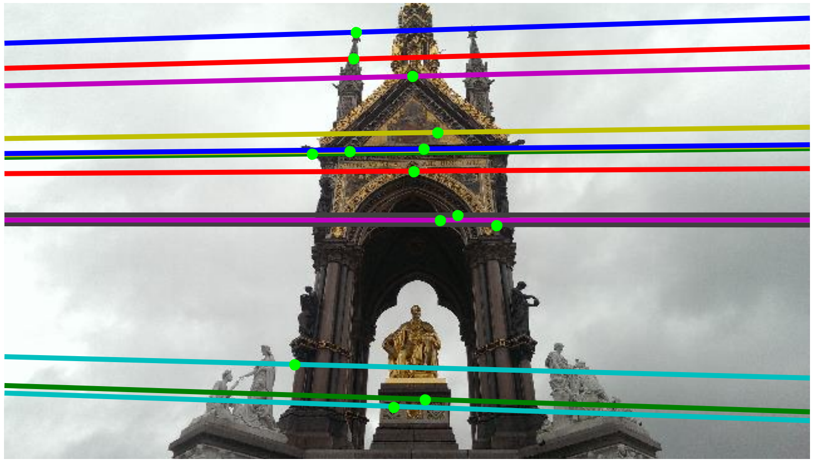

matched points 1
matched points 2

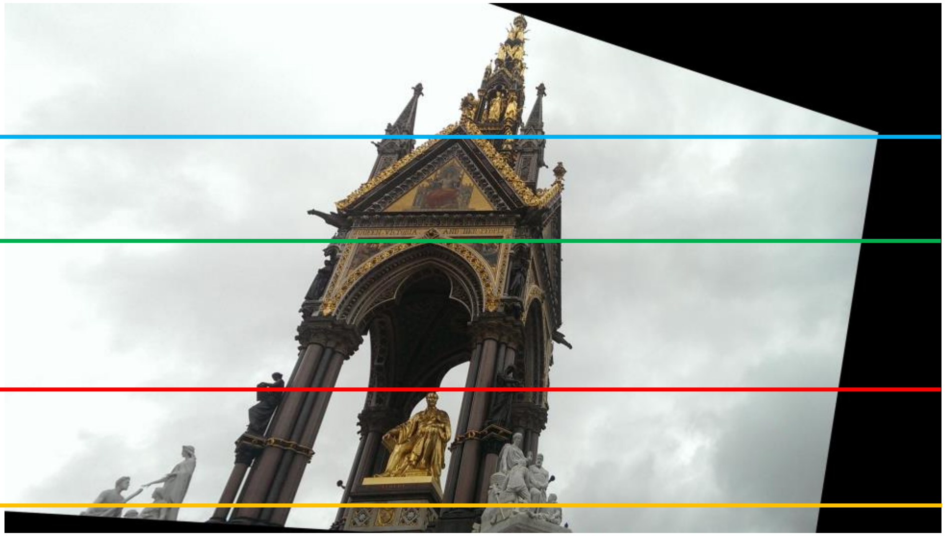







matched points 1
matched points 2



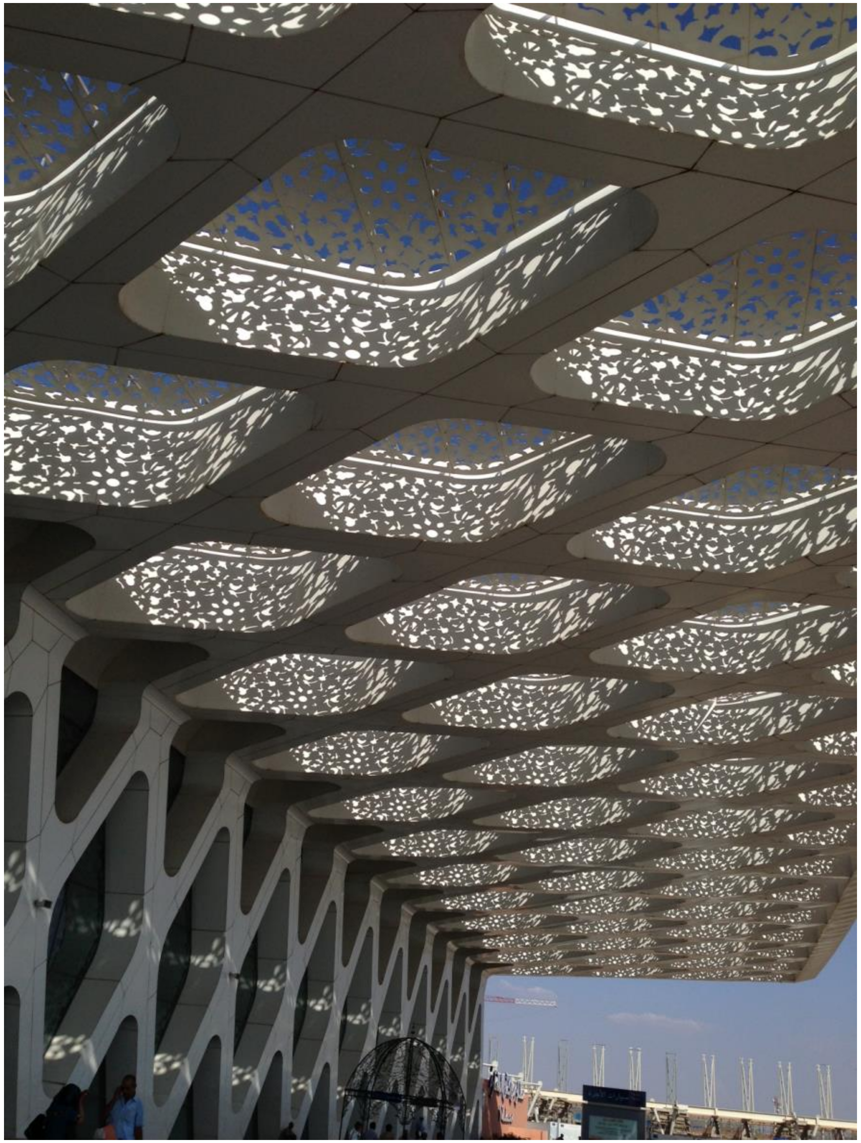
Basic Stereo Matching

- For each detected feature point x in the first image
  - Find corresponding epipolar scanline in the other image
  - Examine all pixels on the epipolar scanline and identify the best match x’
  - Compute disparity x-x’ and set depth of x as:

$$d = \frac{tf}{x - x'}$$

where **t** is the baseline between the two cameras

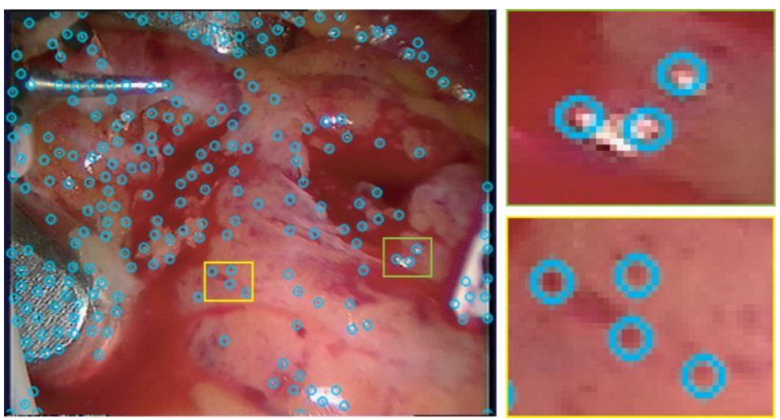
- Issues related to stereo matching
  - Paucity of salient features, uniform surfaces
  - Occlusions and repetitions – texture is good but repetitive textures can impose problems
  - Specular highlights (view dependent)



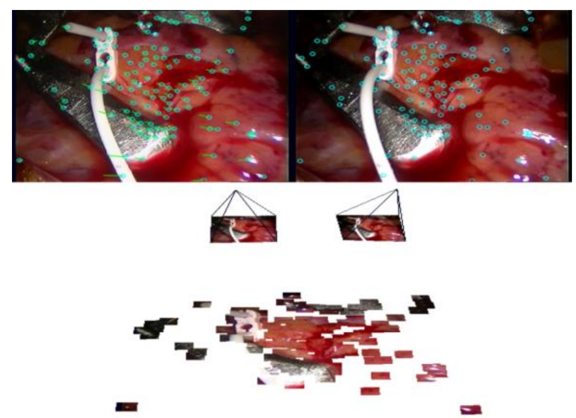
Application Examples

Calibration of Optics and Lights

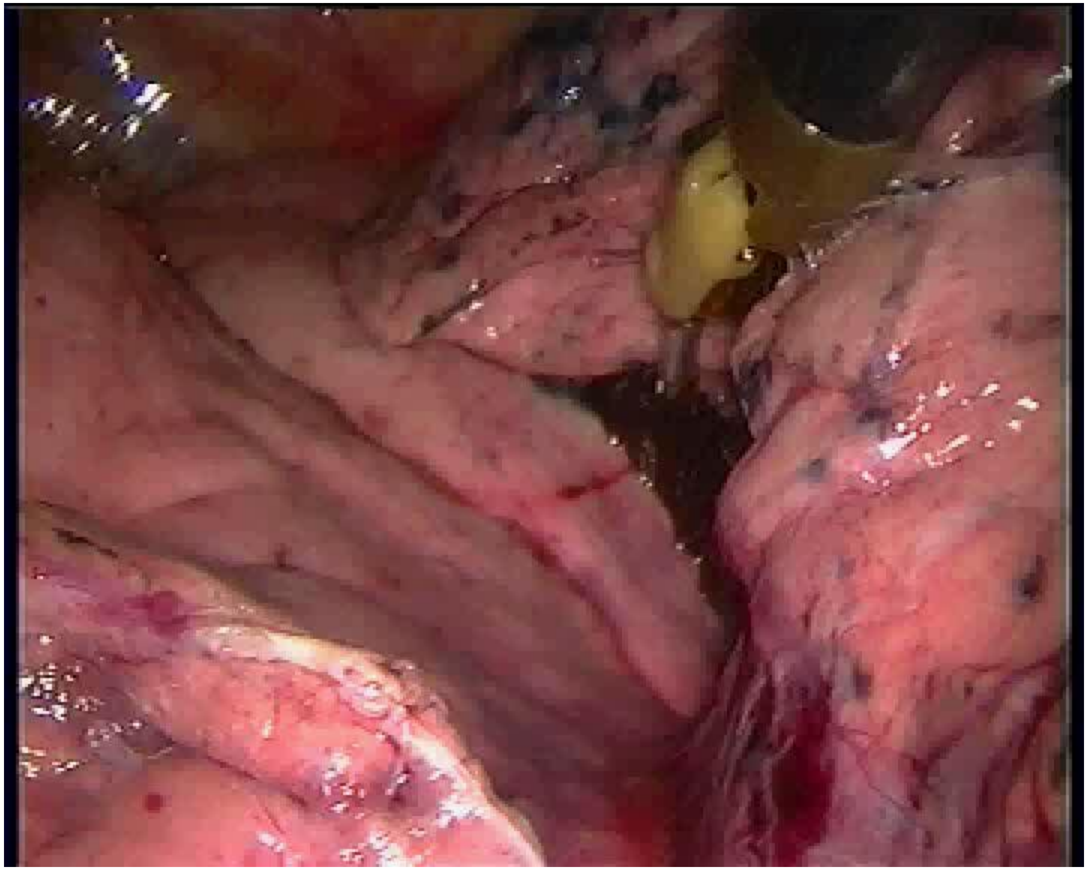
Salient Feature Detection

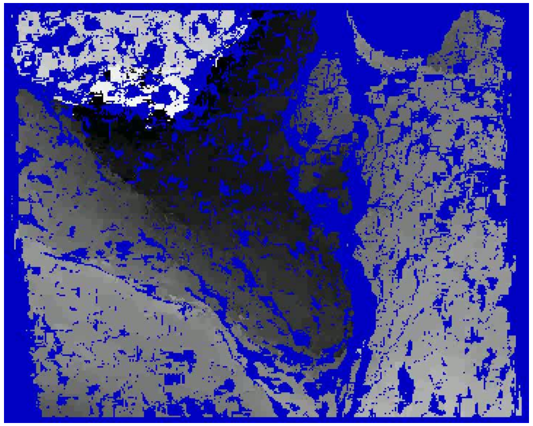


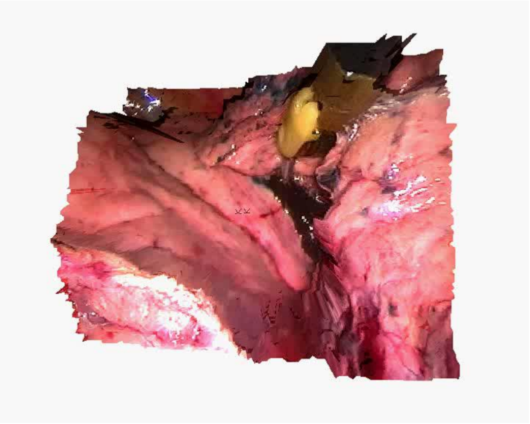
3D Reconstruction

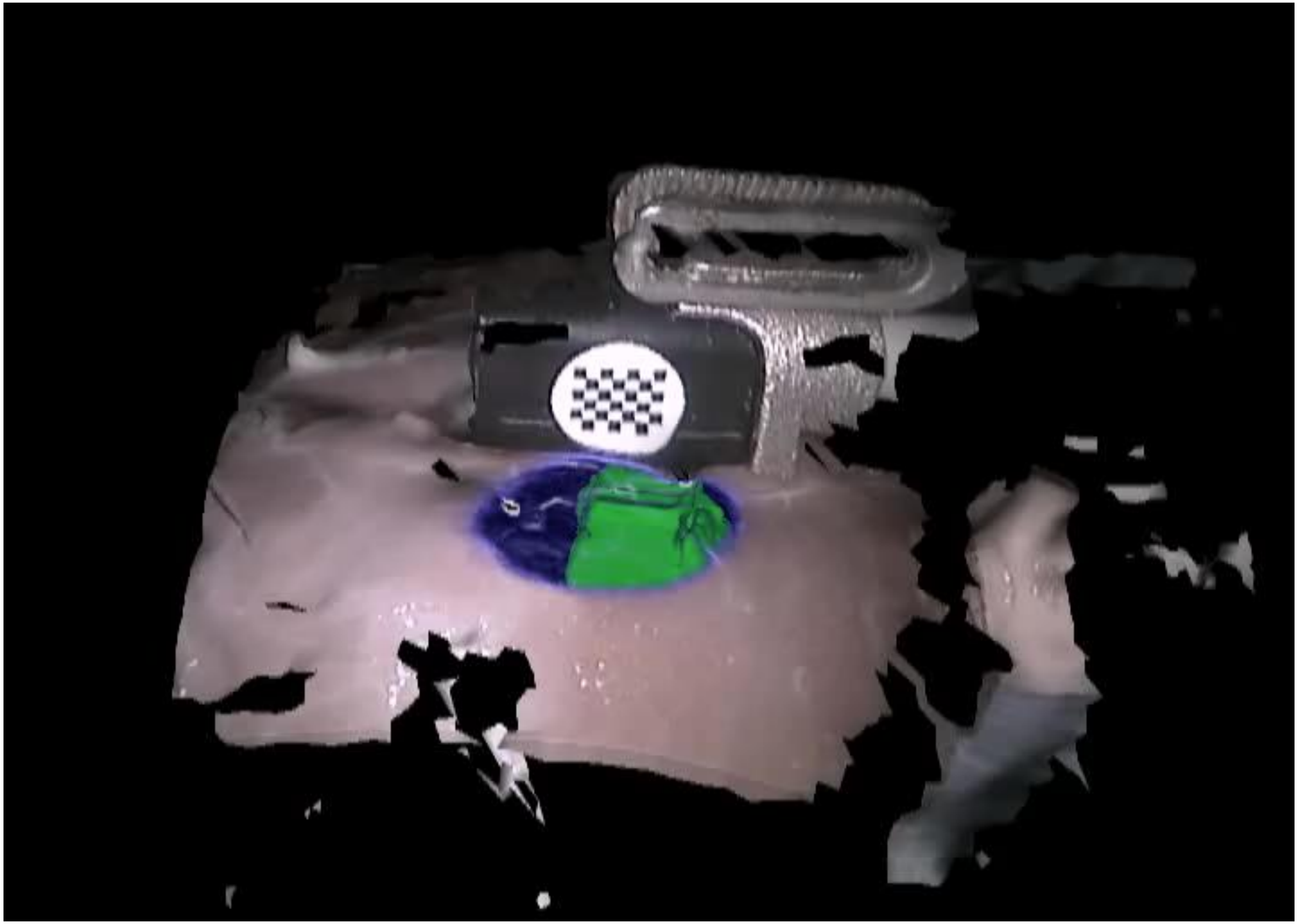


Real-time deformation reconstruction









Conclusions

- Computation of Fundamental Matrix
- Constraints for the Fundamental Matrix
- Stereo Constraints and Priors
- Stereo Image Rectification and Homography
- Application Examples