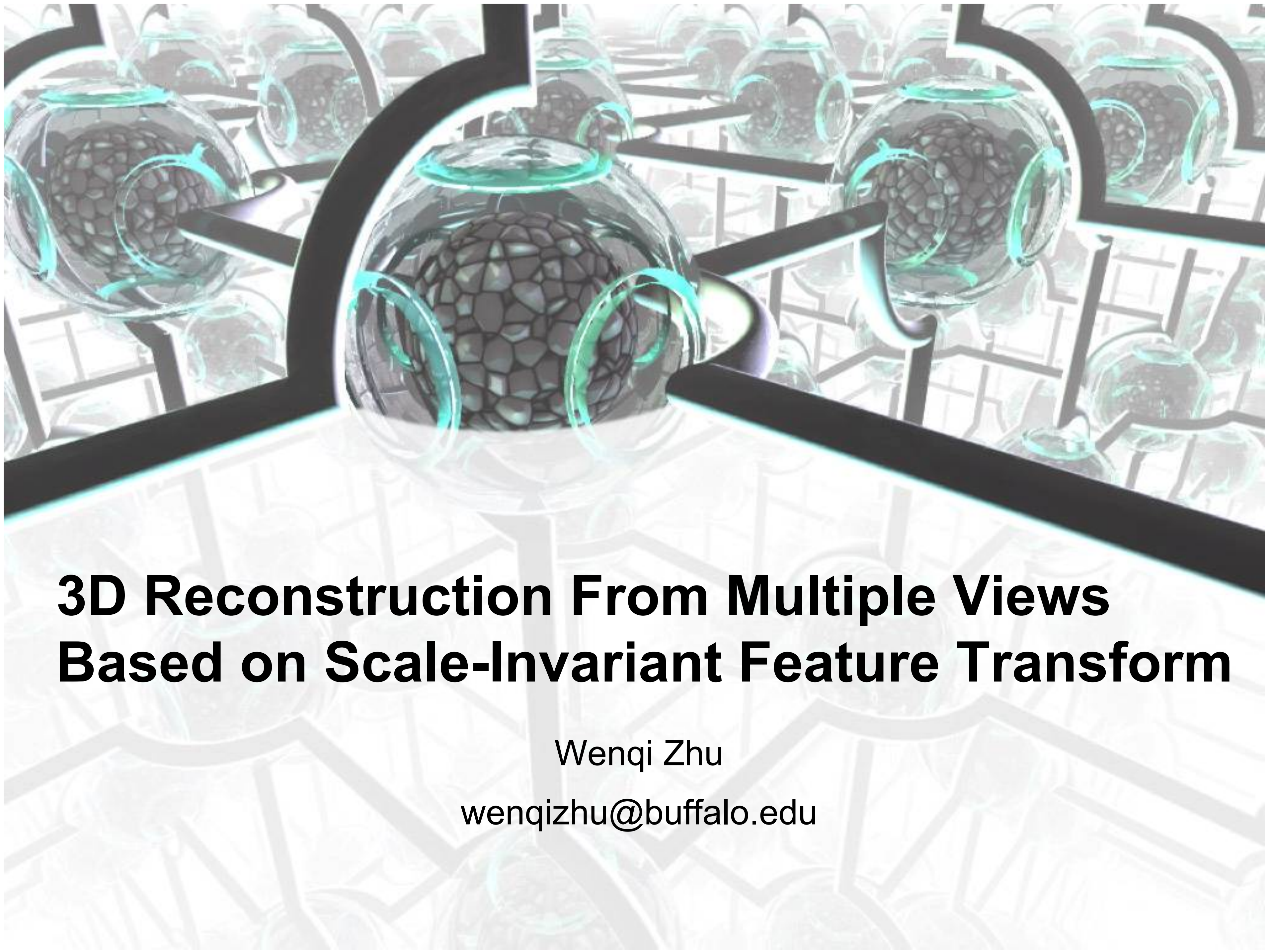


3D Reconstruction From Multiple Views Based on Scale-Invariant Feature Transform

Wenqi Zhu

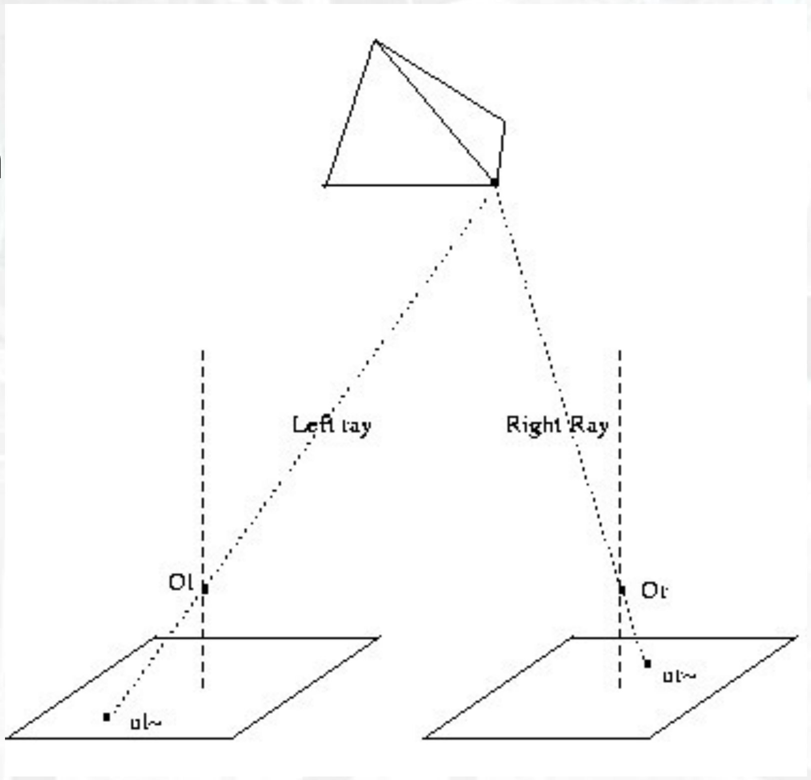
wenqizhu@buffalo.edu

Problem Statement

- 3D reconstruction

  3D reconstruction is a problem of recovering depth information from images.

  Physical point in space is projected onto different locations on images if the viewpoint for capturing the images is changed.The depth information is inferred from the difference in the projected locations.



Problem Statement

- Given an image point x in the first image, how to find the corresponding point x' in the other image ? How does this constrain the position of x'?

Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, i=1,…,n, how to calibrate the cameras C,C'?

Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras C, C', what is the position of X in space?

Overview

Feature extraction & Feature matching

To Solve the corresponding problem.
Based on the SIFT algorithm designed by Lowe.

Camera Calibraion

Suppose that we have already calibrated the cameras in this project.

3D reconstruction

To compute the 3D points in space.

Introduction of SIFT Algorithm

- **Main Idea:**

  SIFT algorithm is a local feature extracion algorithms, in the scale space looking for extrema points, extract the location, scale and rotation invariant.

  http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

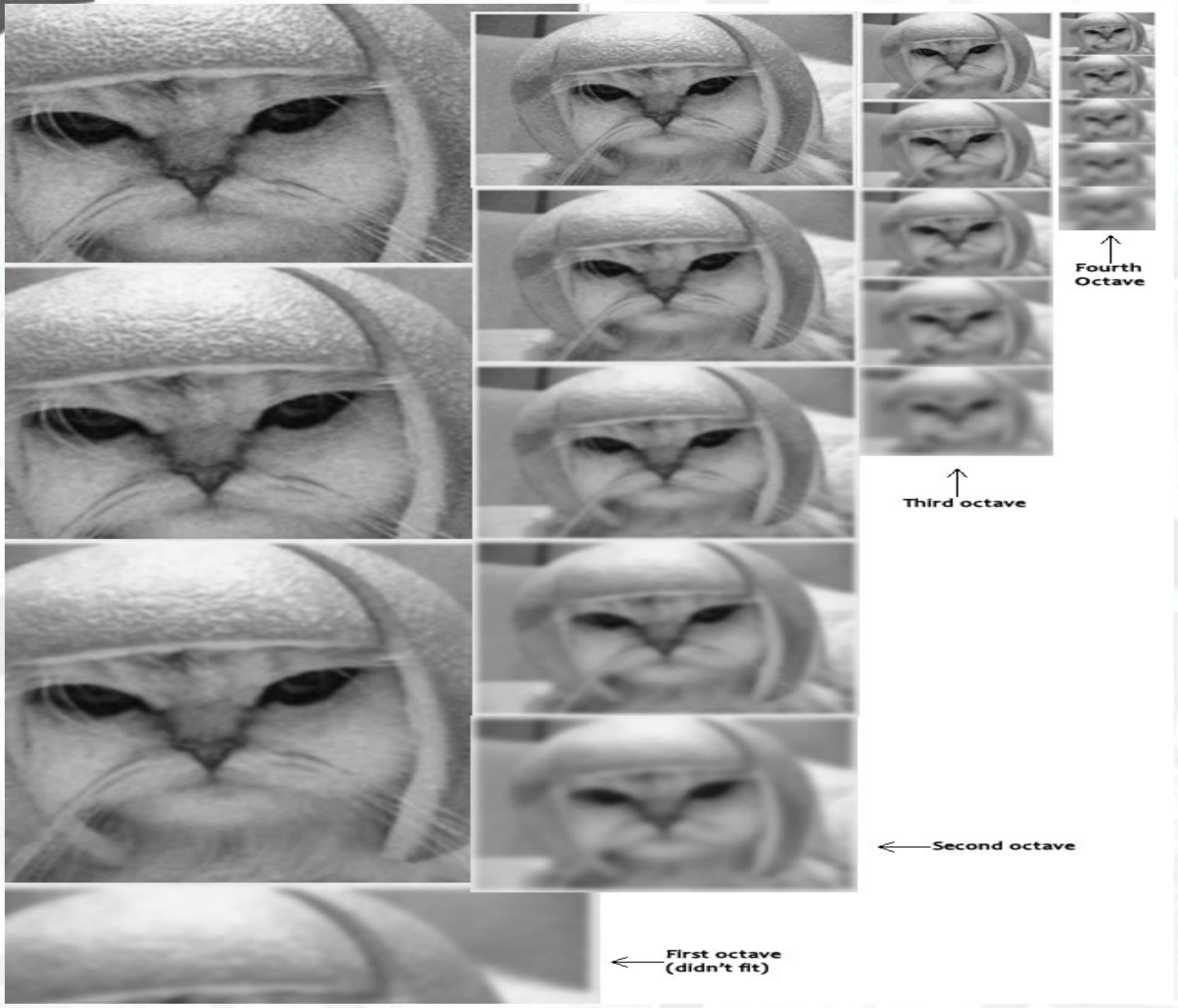
There are 4 basic steps taken by SIFT:

1.Scale-space extrema detection

2.Keypoint(feature matching point) localization

3.Orientation assignment

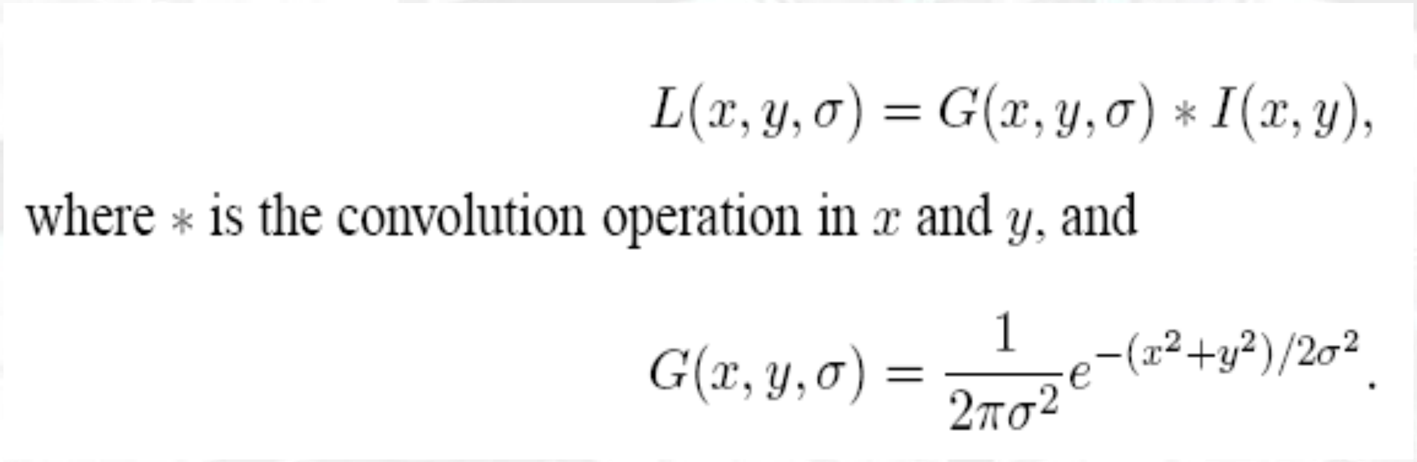
4.Local key point descriptor generation



http://www.aishack.in/2010/05/sift-step-1-constructing-a-scale-space/

Step 1:Detection of scale-space extrema

- The only possible scale-space kernel is the Gaussian function. Therefore, the scale space of an image is defined as a function, L(x; y; σ), that is produced from the convolution of a variable-scale Gaussian, G(x; y;σ), with an input image, I(x; y):



$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where $*$ is the convolution operation in x and y, and

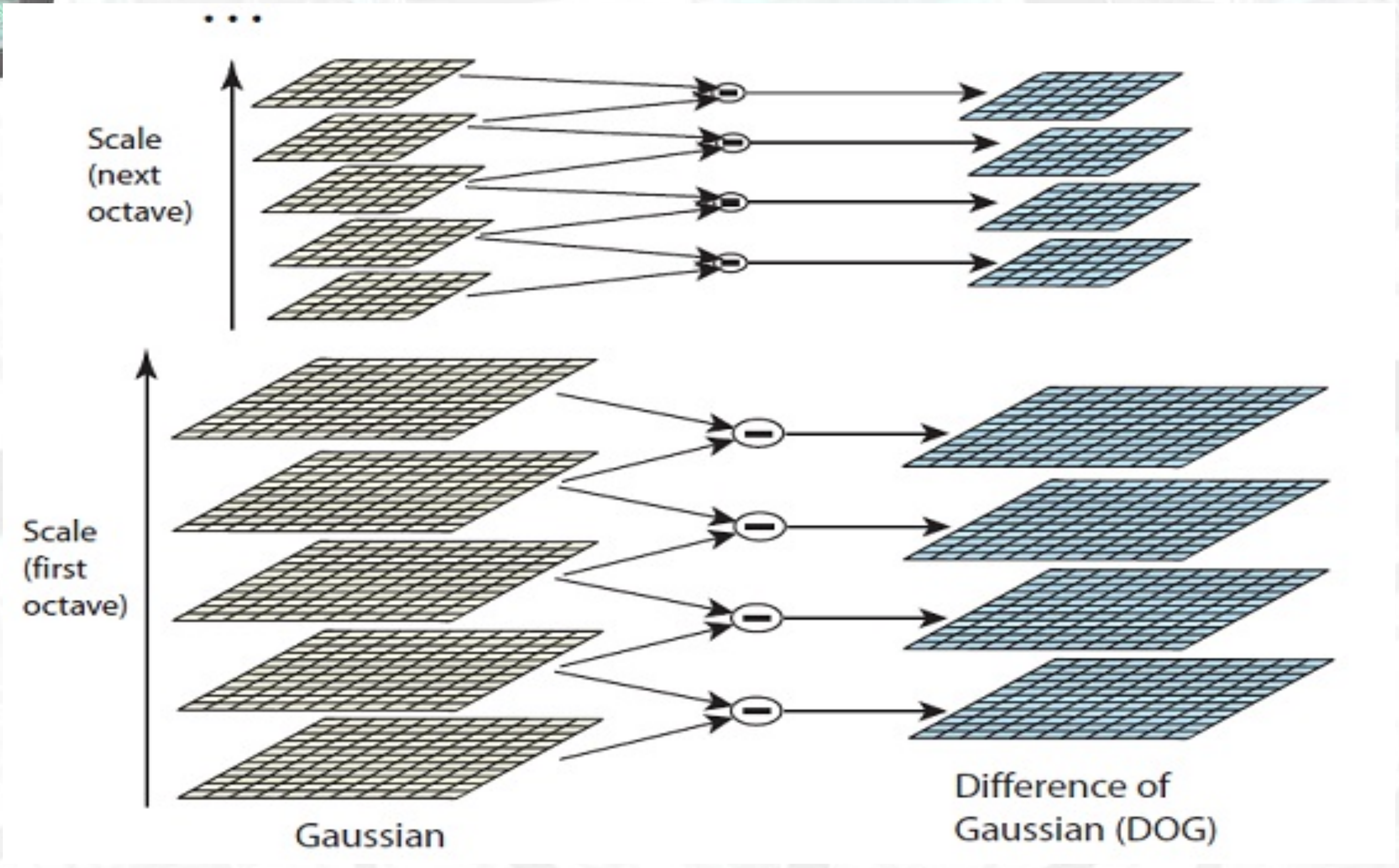
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

Step 1:Detection of scale-space extrema

- To efficiently detect stable keypoint locations in scale space, Lowe proposed using scale-space extrema in the difference-of-Gaussian function convolved with the image, Dog(x; y; σ), which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:
-

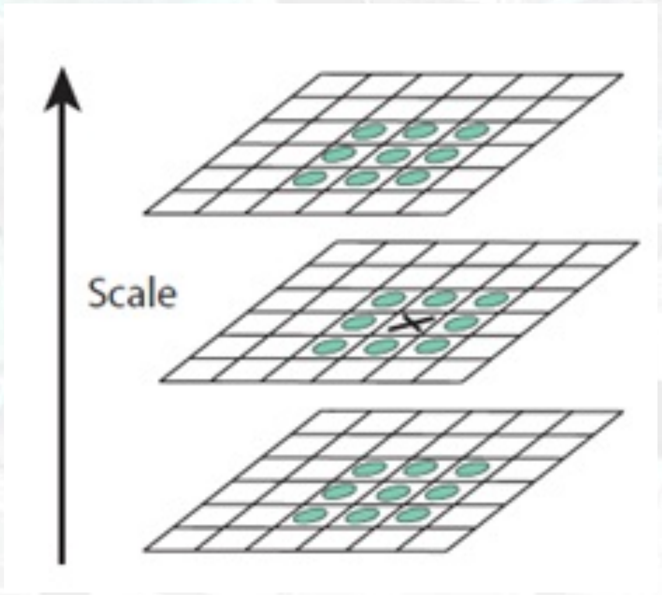
$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma). \end{aligned}$$

Step 1:Detection of scale-space extrema



Step 1:Detection of scale-space extrema

- In order to detect the local maxima and minima of Dog (x; y;σ), each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below.
- It is selected only if it is larger than all of these neighbors or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks.



Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Step 2: Accurate keypoint localization

- Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures.

- This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

Step 3: Orientation assignment

- For each image sample, L(x, y), at thisscale, the gradient magnitude, m(x, y), and orientation, ϴ (x, y)

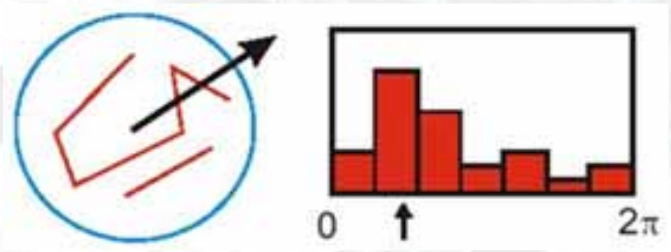
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

- The orientation histogram has 36 bins covering the 360 degree range of orientations.

Step 3: Orientation assignment

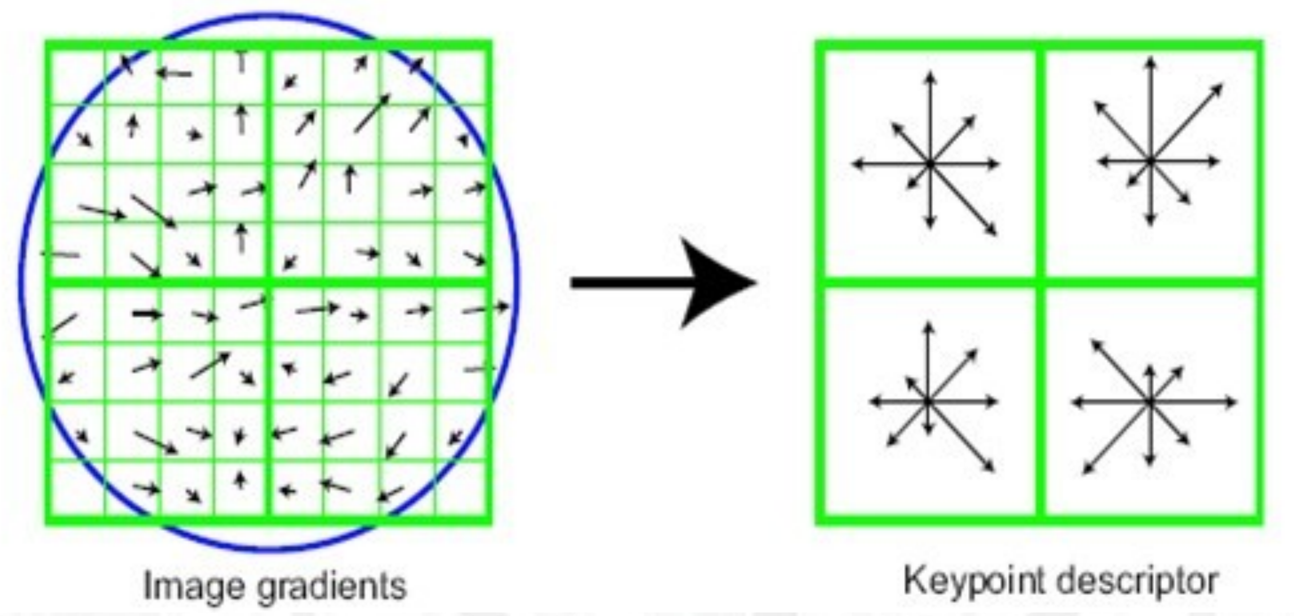
- Peaks are defined by dominant directions of local gradients. The highest peak in the histogram is detected, which the local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation.



- Finally, SIFT Feature is described by location, scale, and orientation

Step 4: Local keypoint descriptor generation

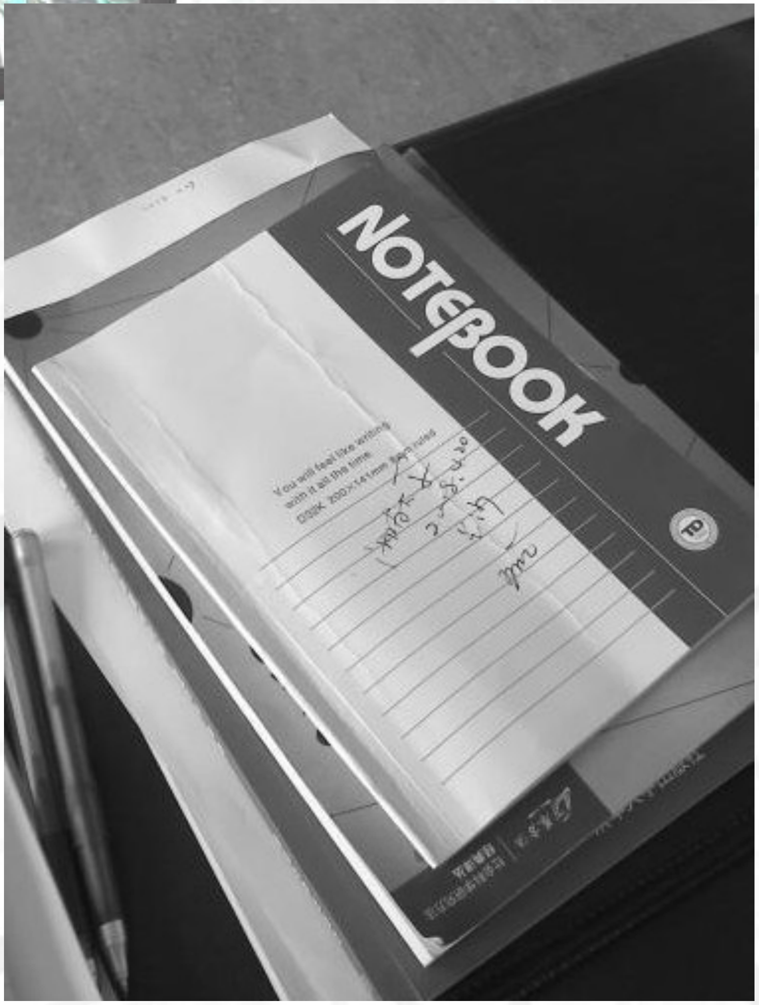
In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation.

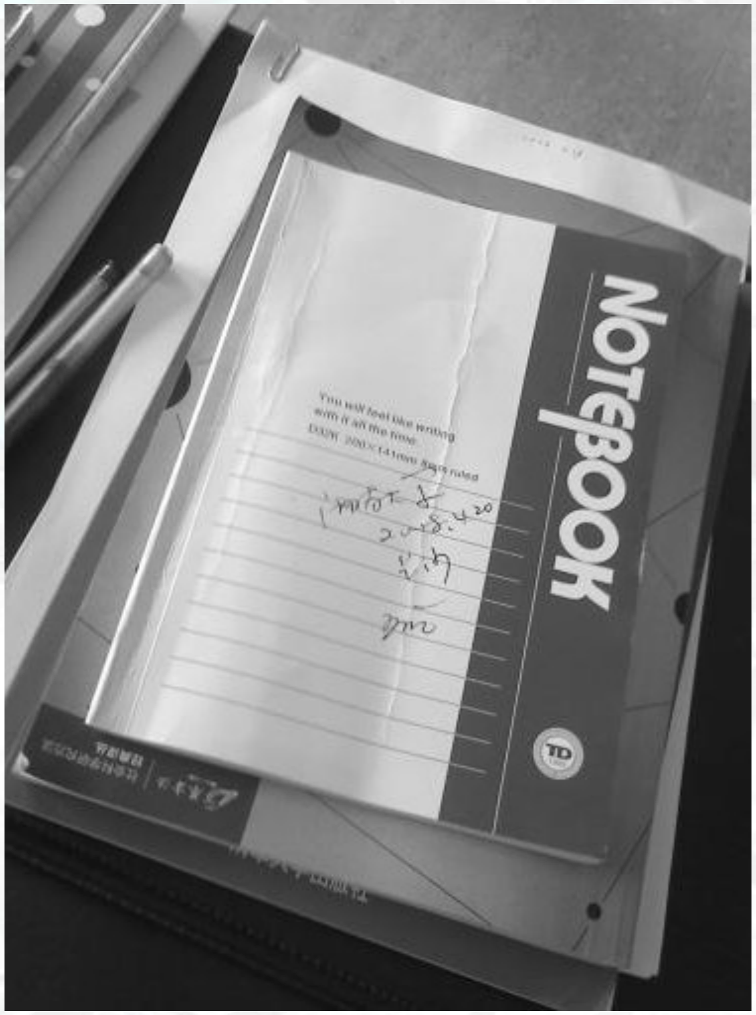


Feature points matching

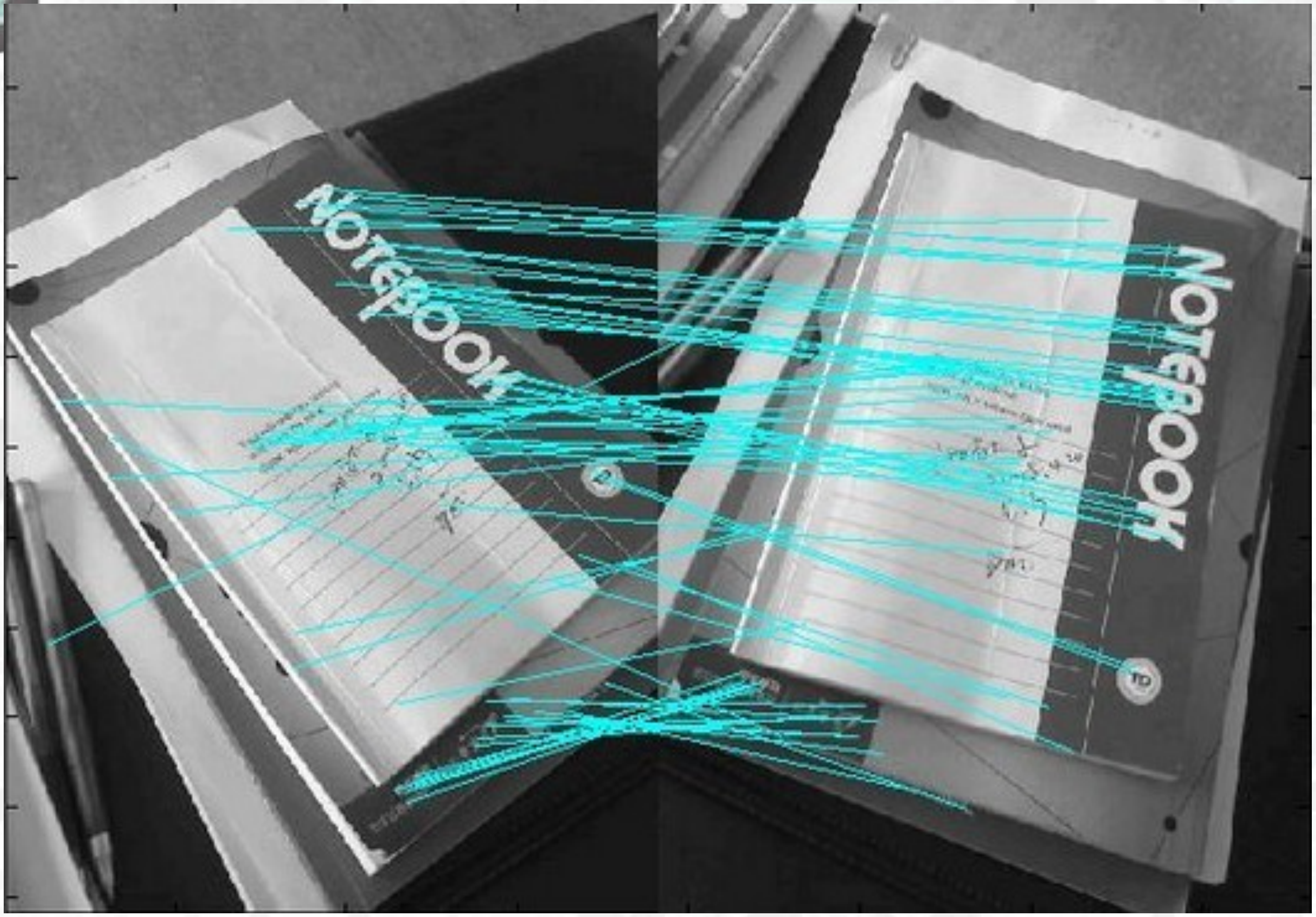
- After the SIFT feature vectors of the key points are created, the Euclidean distances between the feature vectors are exploited to measure the similarity of key points in different digital images.
- A feature point from one image is chosen, and then another two feature points are found by traversing all the feature points in another corresponding image which have the shortest and next-shortest Euclidean distances. Among these two feature points, if the divisor between the shortest and next-shortest Euclidean distances is less than a threshold value, then they are judged to be paired feature points.

Test


NOTEBOOK
You will feel like writing with it all the time.


NOTEBOOK
You will feel like writing with it all the time.

Test



Overview

Feature extration & Feature matching

To Solve the corresponding problem.
Based on the SIFT algorithm designed by Lowe.

Camera Calibraion

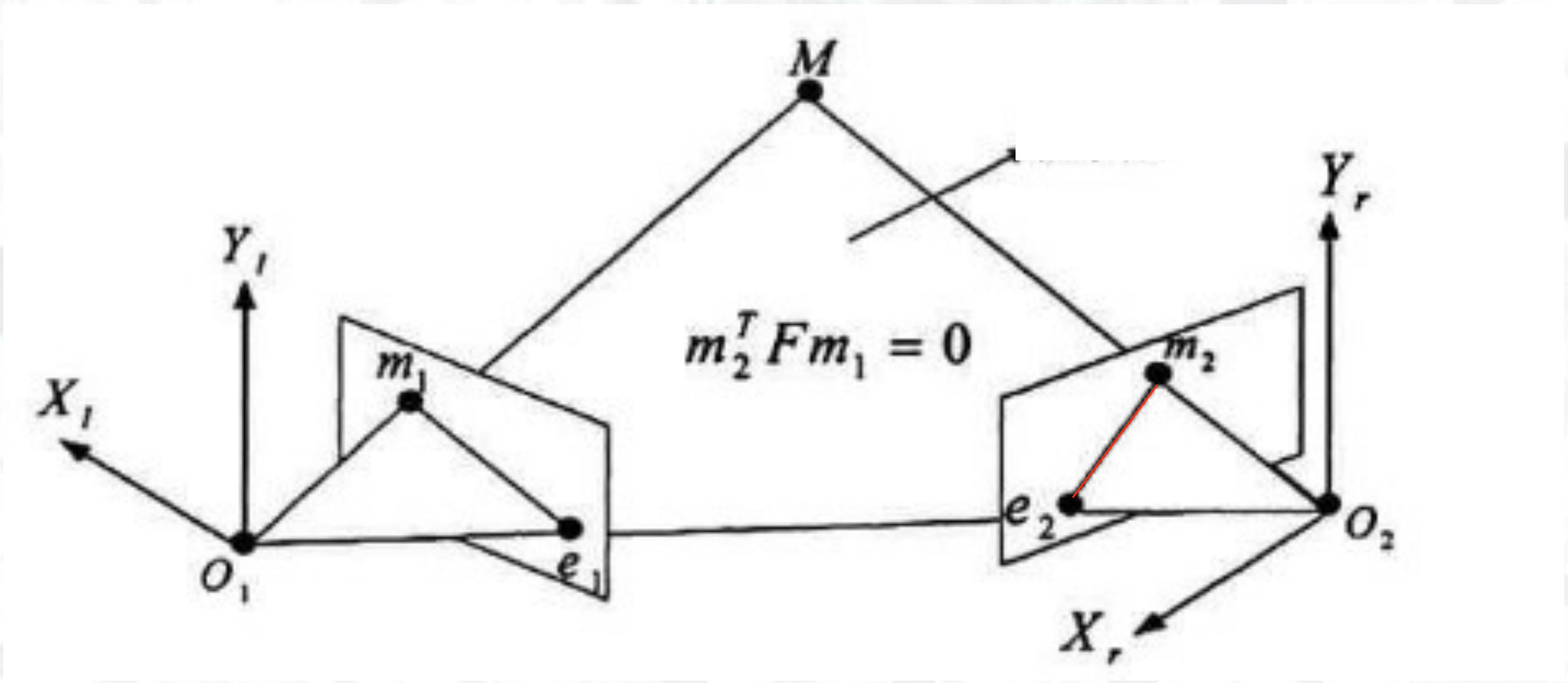
Suppose that we have already calibrated the cameras in this project.

3D reconstruction

To compute the 3D points in space.

Application of Epipolar Geometry in Corresponding Problem

- **According to the epipolar geometry, we know the corresponding point must lie along the epipolar line. It can be used to emliminate the wrong matching points.**



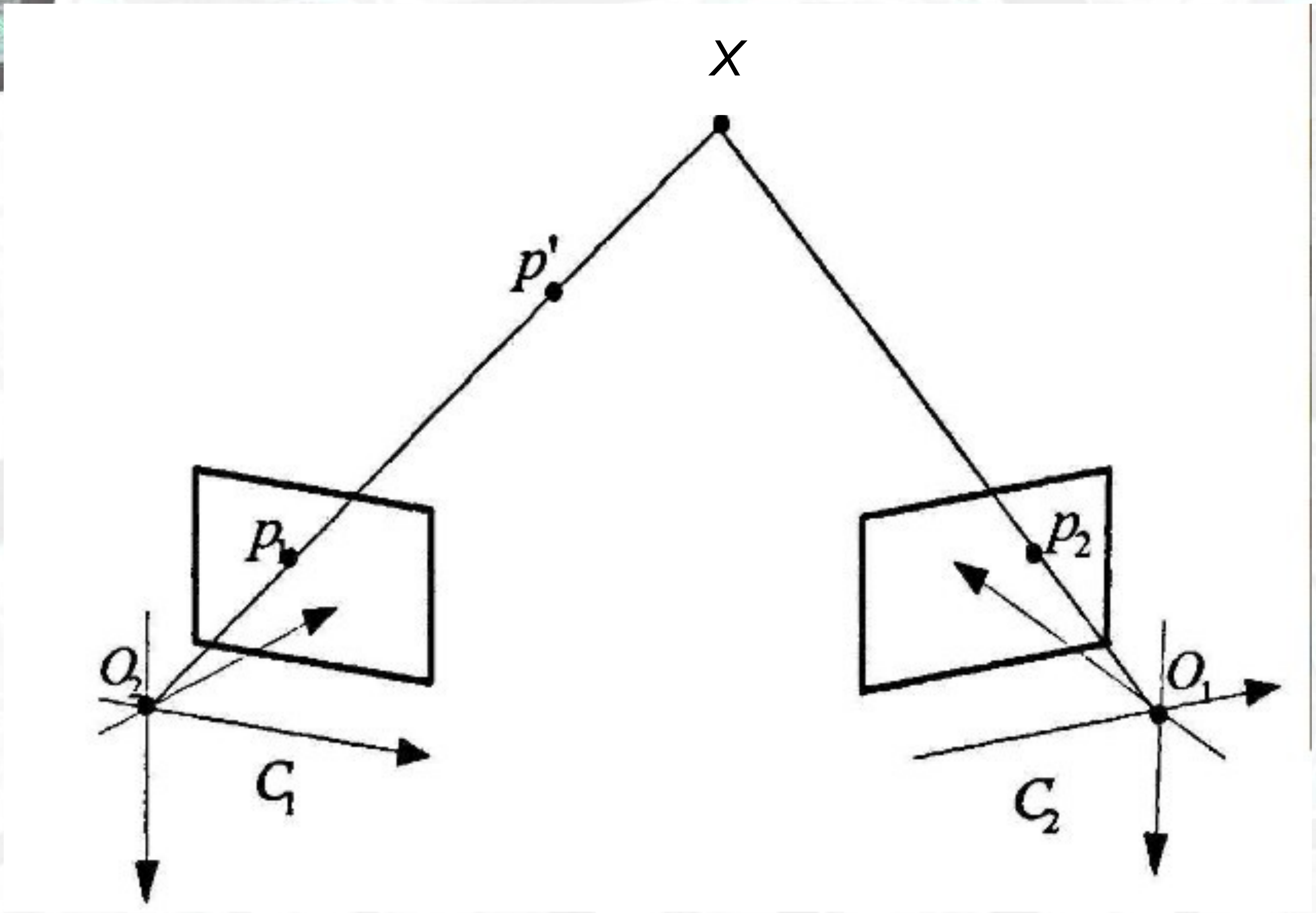
The improved SIFT algorithm

- In SIFT algorithm, a feature point from one image is chosen, and then another two feature points are found by traversing all the feature points in another corresponding image which have the shortest and next-shortest Euclidean distances.

- By epipolar geometry, we only have to traverse those feature points along the epipolar lines.We compute the distance from matching points to the epipolar line. If the value is less than a certain value we supposed, we consider it a matching point. Here is the formula to compute this distance:

$$d = \frac{\left(m_2^T F m_1\right)^2}{\left(Fm_2\right)_1^2 + \left(Fm_2\right)_2^2}$$

3D construction by two views



3D construction by two views

- **Given the image point p1(u1,v1),p2(u2,v2) corresponding to a desired scene point X.**

According to the pinhole camera model PCM:

$$[x_w\ y_w\ z_w]^T \text{ -> } [u\ v]^T$$

$$z_c\ [u1\ v1\ 1]^T = K\ R\ ([x_w\ y_w\ z_w]^T - t)$$

Or using homogeneous notation

$$u\sim = z_c[u\ v\ 1]^T, \qquad X_w\sim = [x_w\ y_w\ z_w\ 1]^T$$

we have the camera model in homogeneous coordinates

$$u\sim = M\ X_w\sim$$

where M is the 3x4 matrix M = [K*R -K*R*t] called the projective matrix.

3D construction by two views

- Suppose that camera C1 and C2 are calibrated, M1 and M2 are their projective matrix. So we have

$$Zc_1\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^1 & m_{12}^1 & m_{13}^1 & m_{14}^1 \\ m_{21}^1 & m_{22}^1 & m_{23}^1 & m_{24}^1 \\ m_{31}^1 & m_{32}^1 & m_{33}^1 & m_{34}^1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$Zc_2\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^2 & m_{12}^2 & m_{13}^2 & m_{14}^2 \\ m_{21}^2 & m_{22}^2 & m_{23}^2 & m_{24}^2 \\ m_{31}^2 & m_{32}^2 & m_{33}^2 & m_{34}^2 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where c1,c2,and m_{ij} can be obtained by camera calibration.

3D construction by two views

- By eliminating Zc1 or Zc2, we get four formulas about X,Y,Z:

$$
\begin{aligned}
(u_1 m_{31}^1 - m_{11}^1)X + (u_1 m_{32}^1 - m_{12}^1)Y + (u_1 m_{33}^1 - m_{13}^1)Z &= m_{14}^1 - u_1 m_{34}^1 \\
(v_1 m_{31}^1 - m_{21}^1)X + (v_1 m_{32}^1 - m_{22}^1)Y + (v_1 m_{33}^1 - m_{23}^1)Z &= m_{24}^1 - v_1 m_{34}^1 \\
(u_2 m_{31}^2 - m_{11}^2)X + (u_2 m_{32}^2 - m_{12}^2)Y + (u_2 m_{33}^2 - m_{13}^2)Z &= m_{14}^2 - u_2 m_{34}^2 \\
(v_2 m_{31}^2 - m_{21}^2)X + (v_2 m_{32}^2 - m_{22}^2)Y + (v_2 m_{33}^2 - m_{23}^2)Z &= m_{24}^2 - v_2 m_{34}^2
\end{aligned}
$$

X,Y,Z can be obtained by any three of above formulas.

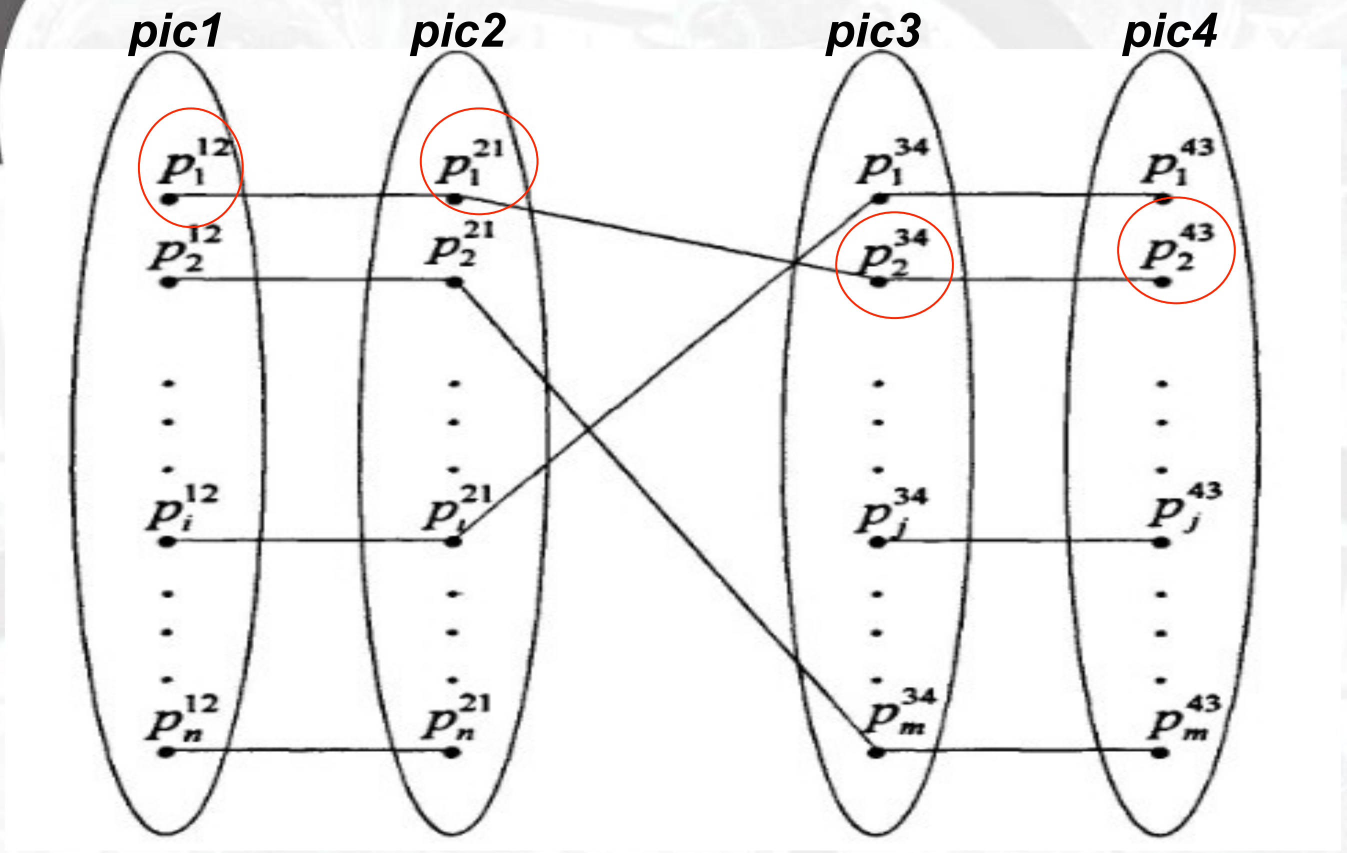
3D construction by multiple views

- In this project, we propose a method to solve the 3D construction by multiple views . It is based on the two-views 3D construction theory we have discussed above.

The most important problems to be solved:

- feature matching among a set of images
- transform the coordinates of a set of 3D points we have got.

Feature matching for multiple views



If p_1^{12} and p_1^{21} are matching points, p_2^{34} and p_2^{43} are matching points, p_1^{21} and p_2^{34} are matching points, we can infer that any two of those four points are matching points mutually.

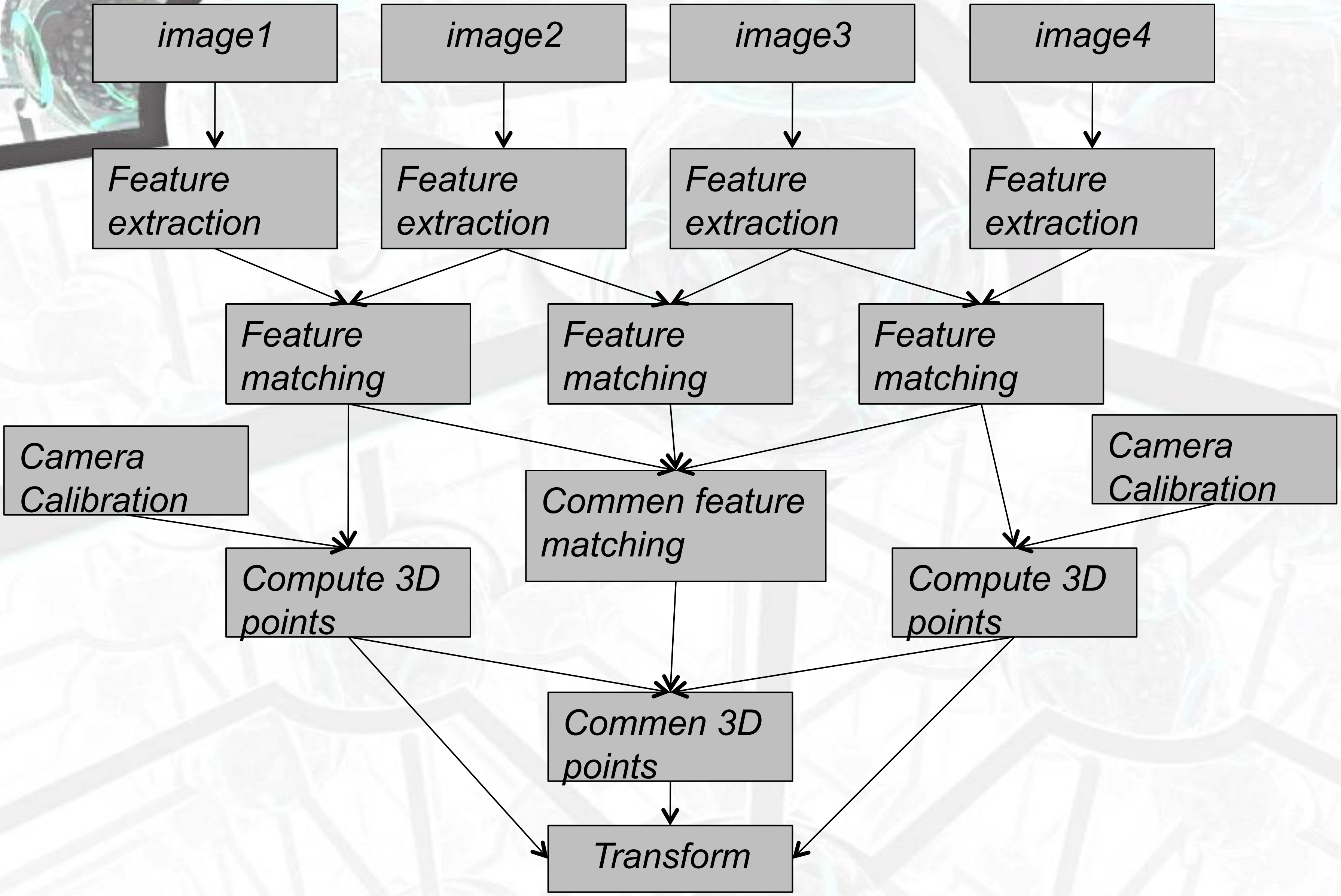


image1
image2
image3
image4
Feature extraction
Feature extraction
Feature extraction
Feature extraction
Feature matching
Feature matching
Feature matching
Camera Calibration
Commen feature matching
Camera Calibration
Compute 3D points
Compute 3D points
Commen 3D points
Transform

Suppose we compute a set of 3D points X={xi} (i=1,2...n) from Pic1 and Pic2,

and another set of 3D points Y={yi}(i=1,2,...n)from Pic3 and Pic4,

now we want to transform them into the same coordinate system.

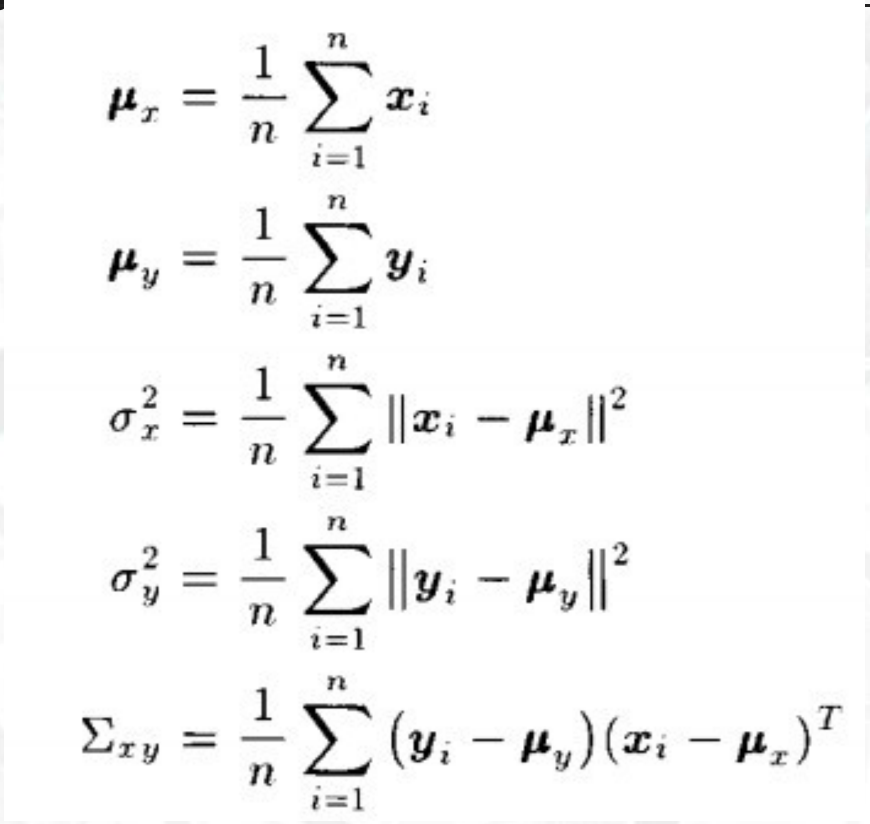
- **We can use this method:**

There is a similarity transformation between the two sets of 3D points.

Y= c*R*X+T

So if we get the scale factor c, the rotation matrix R and the translation matrix T, we can transform those two sets of 3D points.

- **According to “ least-squares estimation of transformation parameters between two point patterns” （http://cis.jhu.edu/software/lddmm-similitude/umeyama.pdf ）, let**



$$\boldsymbol{\mu}_x = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$$

$$\boldsymbol{\mu}_y = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i$$

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu}_x\|^2$$

$$\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{y}_i - \boldsymbol{\mu}_y\|^2$$

$$\Sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\mu}_y)(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^T$$

and let a singular value decomposition of Σ_{xy} be UDV^T ($D = \mathrm{diag}(d_i)$, $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$), and

$$S = \begin{cases} I & \text{if } \det(\Sigma_{xy}) \geq 0 \\ \mathrm{diag}(1, 1, \cdots, 1, -1) & \text{if } \det(\Sigma_{xy}) < 0. \end{cases}$$

Σ_{xy} is a covariance matrix of X and Y, $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$ are mean vectors of X and Y, and σ_x^2 and σ_y^2 are variances around the mean vectors of X and Y, respectively.

When rank $(\Sigma_{xy}) \geq m - 1$, the optimum transformation parameters are determined uniquely as follows:

$$R = USV^T \quad)$$

$$\boldsymbol{t} = \boldsymbol{\mu}_y - cR\boldsymbol{\mu}_x$$

$$c = \frac{1}{\sigma_x^2}\mathrm{tr}(DS)$$

Goals for completing the project

- Use the improved SIFT algorithm to finish the feature matching among images.
- Compute several sets of 3D points from those feature matching points. Programing on Matlab.
  (minimum goal)
- Unified the 3D points coordinate system by “ least-squares estimation of transformation parameters between two point patterns” .   (if there is time after minimum goal)

The End

Thanks for listening ☺