中国科学技术大学
University of Science and Technology of China

数学建模
Mathematical Modeling

陈仁杰
中国科学技术大学

数码相机定位

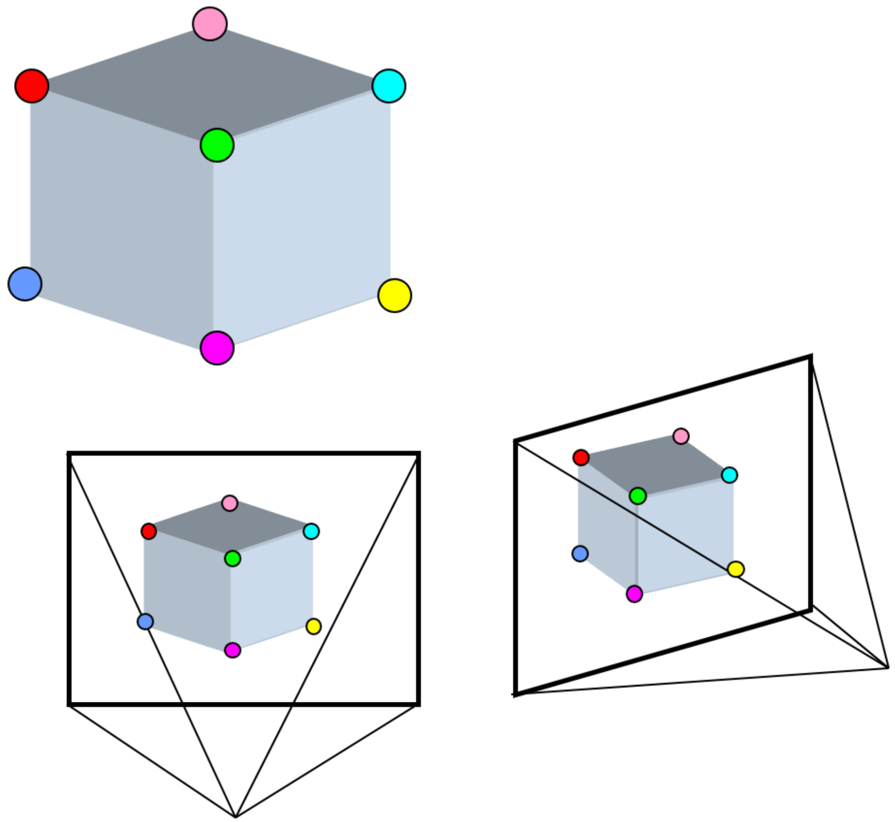
（课本第四章）

机理建模

- 根据问题的实际机理，比如物理规律、几何规律等，进行数学模型
- 机理：
  - 物理规律：牛顿运动学定律、材料力学、流体力学、光学、热力学…
  - 运动方程
  - 偏微分方程
  - 图理论
  - …

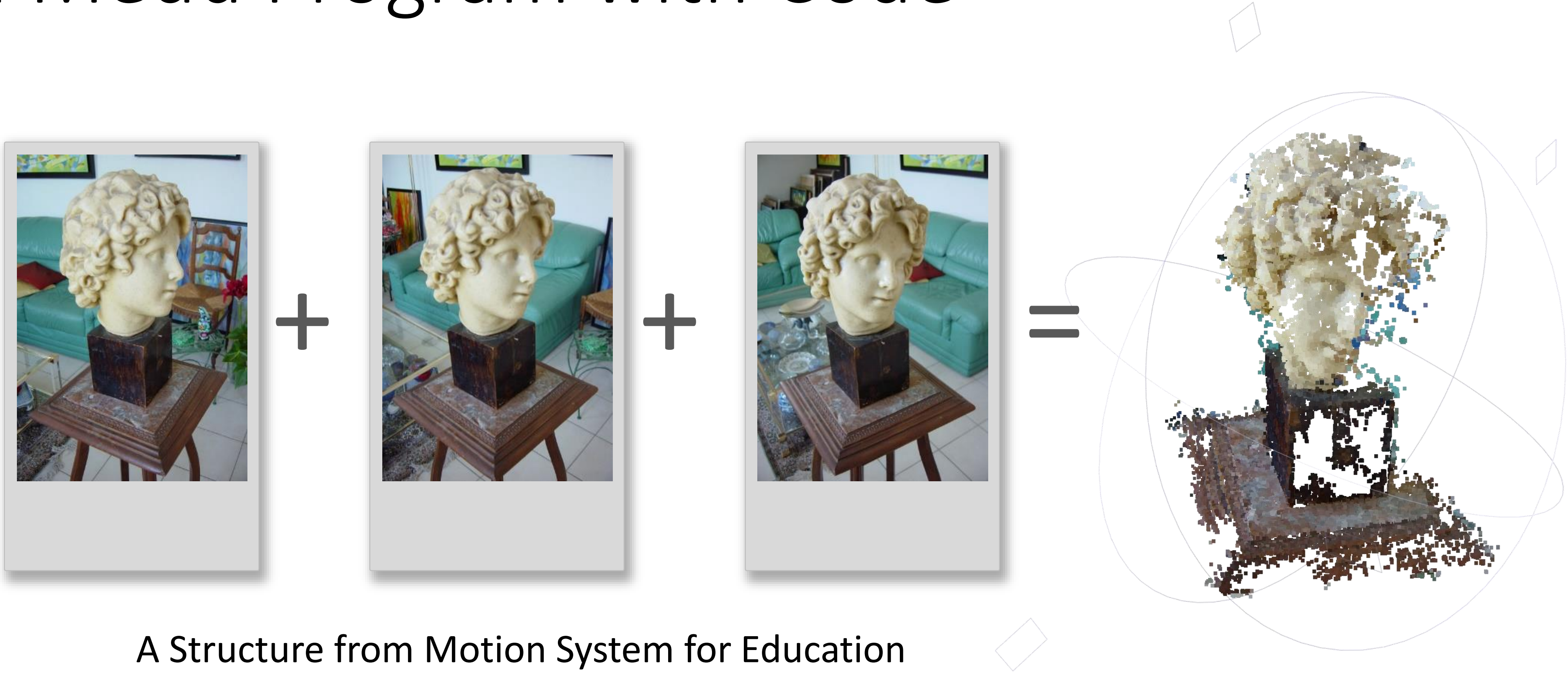
相机定位问题

- 两部相机拍摄同一物体，得到两幅图像
- 根据两幅图像的公共特征，可以求得两部相机的相对位置关系
- 也可以求得对应公共特征点的三维坐标



- 机理：
  - 几何光学
  - 针孔成像原理

SFMedu Program with Code

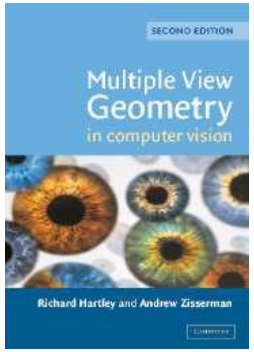


A Structure from Motion System for Education

Download from: http://3dvision.princeton.edu/courses/SFMedu/

How: a complete new way of learning

Standard Way



Reading: the MVG bible
(need ~2 years)

5-6 years

Coding

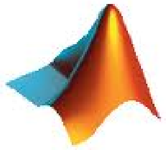
Crying: Not Working At All

PhD ?

Yes

Our Way

1 hour



Run a program

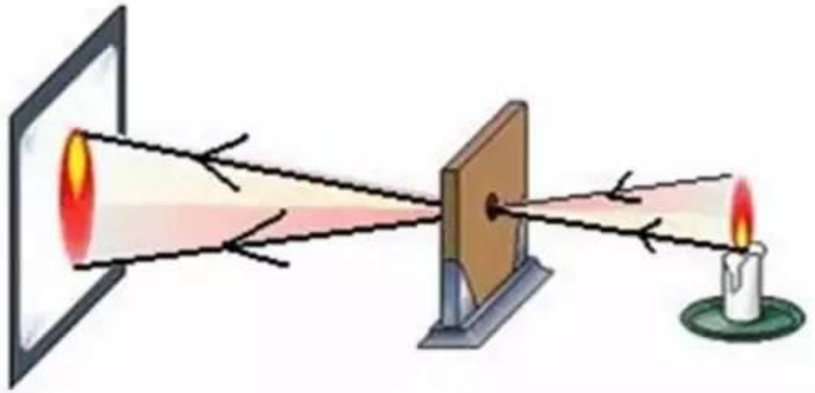
See pictures

Listen stories

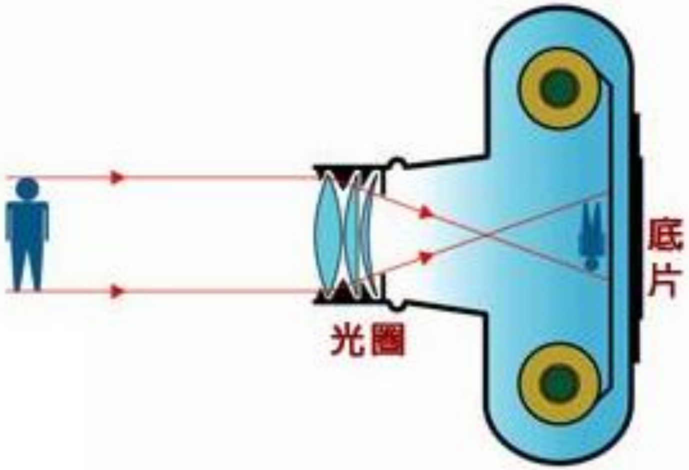
$\mathbf{x} = \mathbf{PX}$ Play with math

Enjoy&improve

小孔成像



小孔成像



针孔成像的几何模型

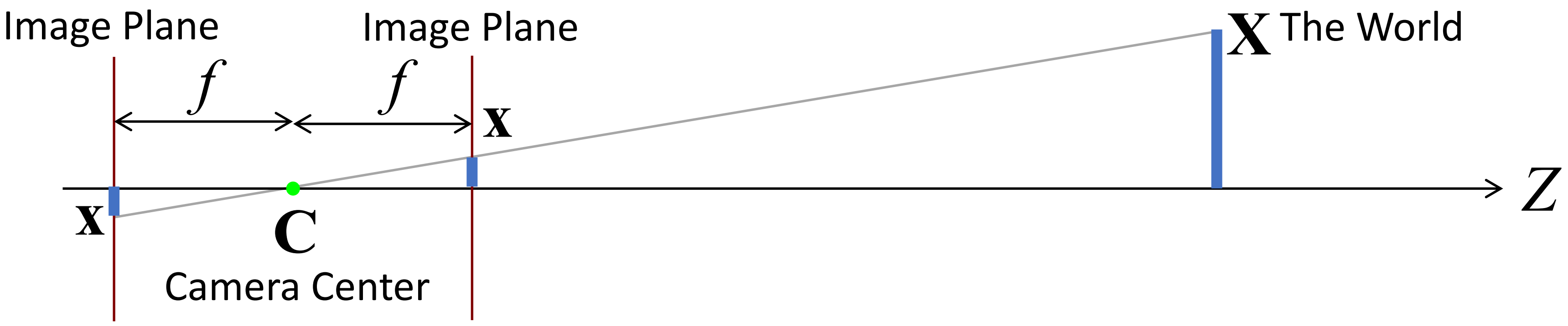
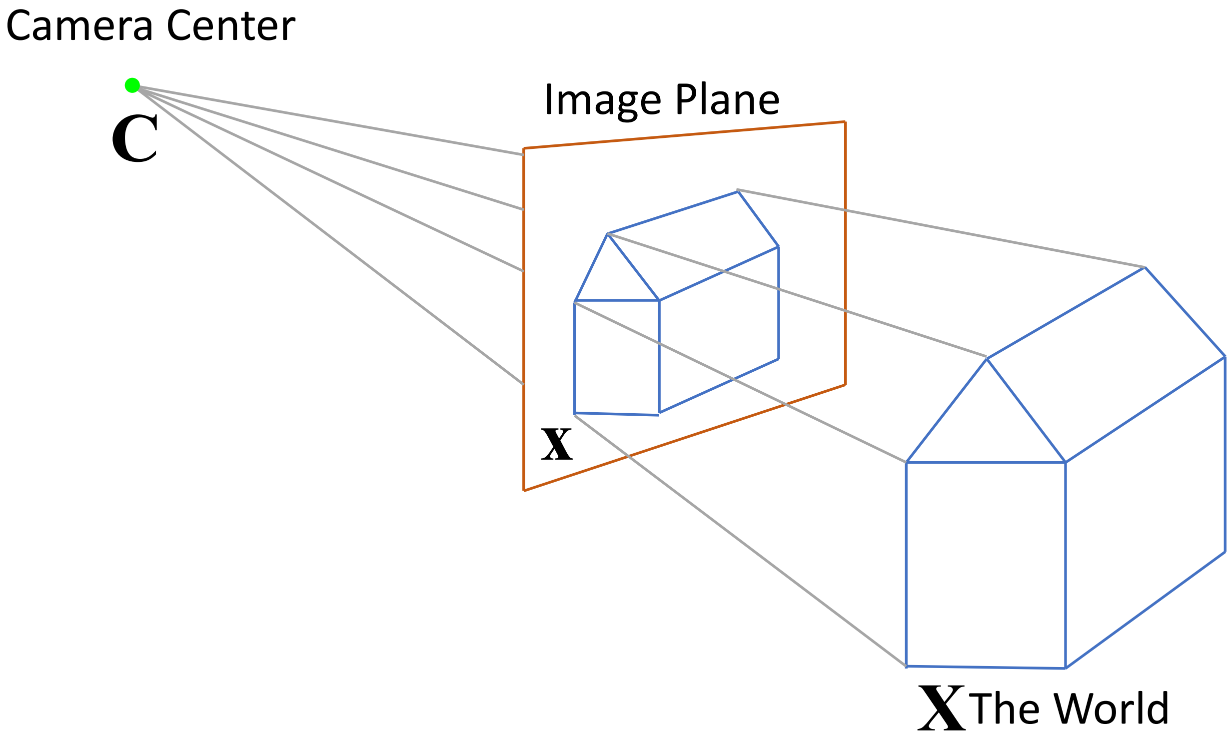
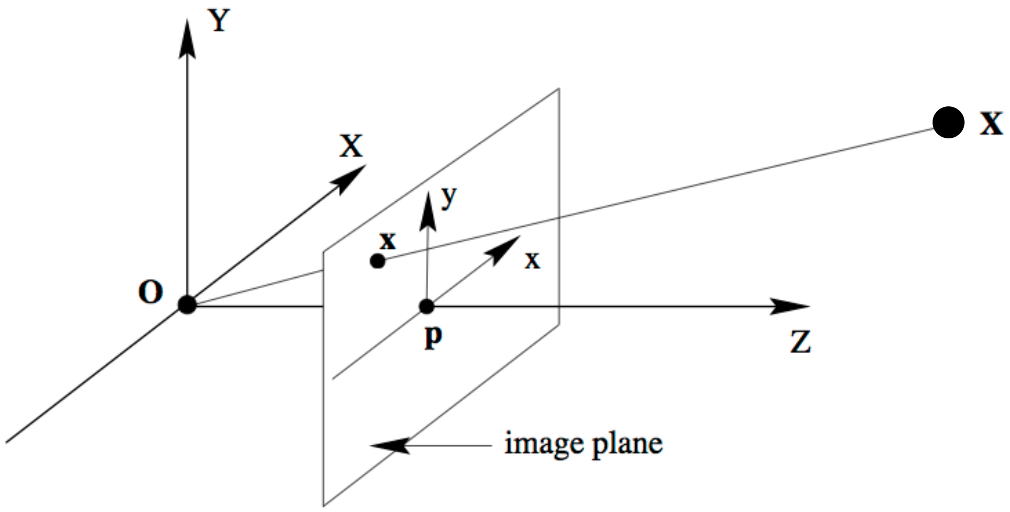


Image Plane
Image Plane
X The World
f
f
x
x
C
Camera Center
Z


Camera Center
C
Image Plane
x
X The World

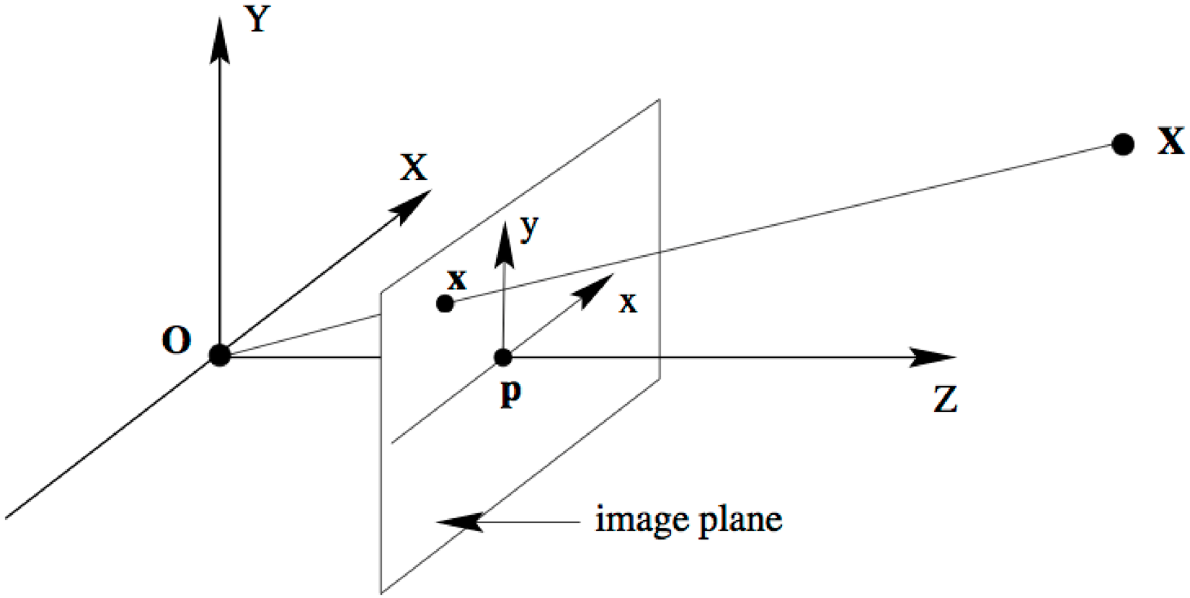
针孔成像的几何模型



$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

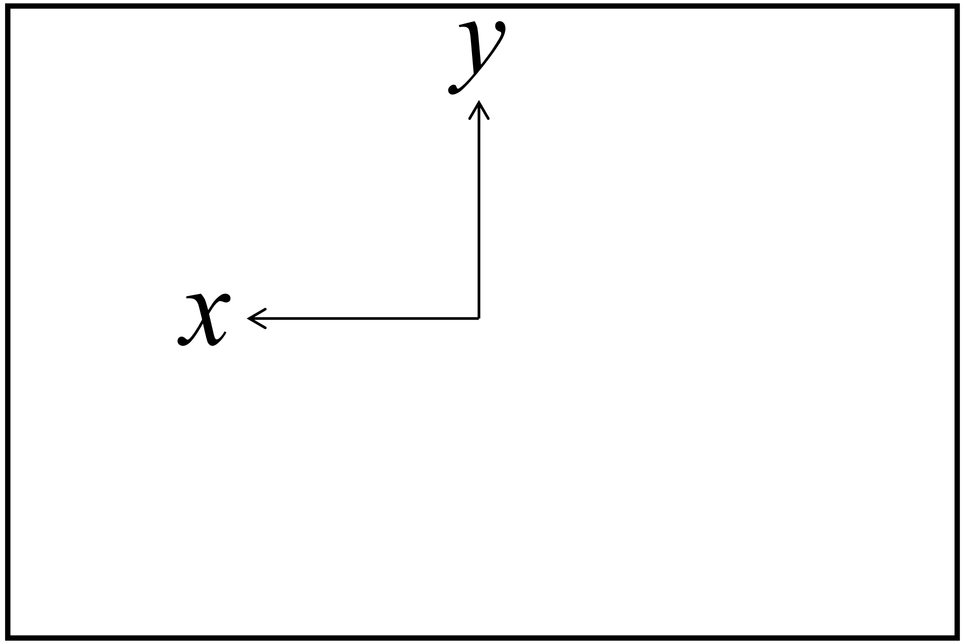
针孔成像的几何模型



$$\lambda \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Longleftrightarrow \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

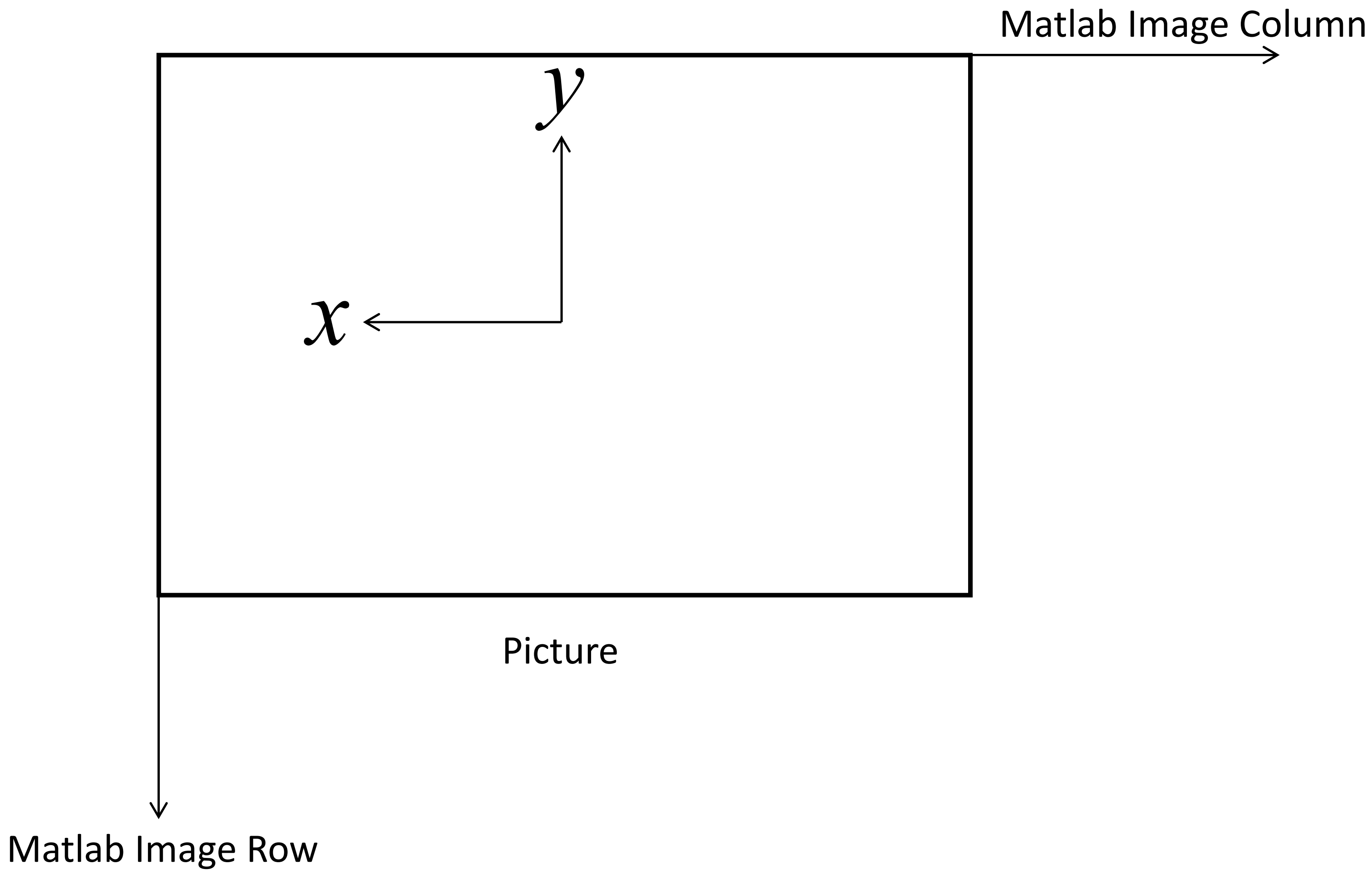
Up to a scale

When they take a picture:

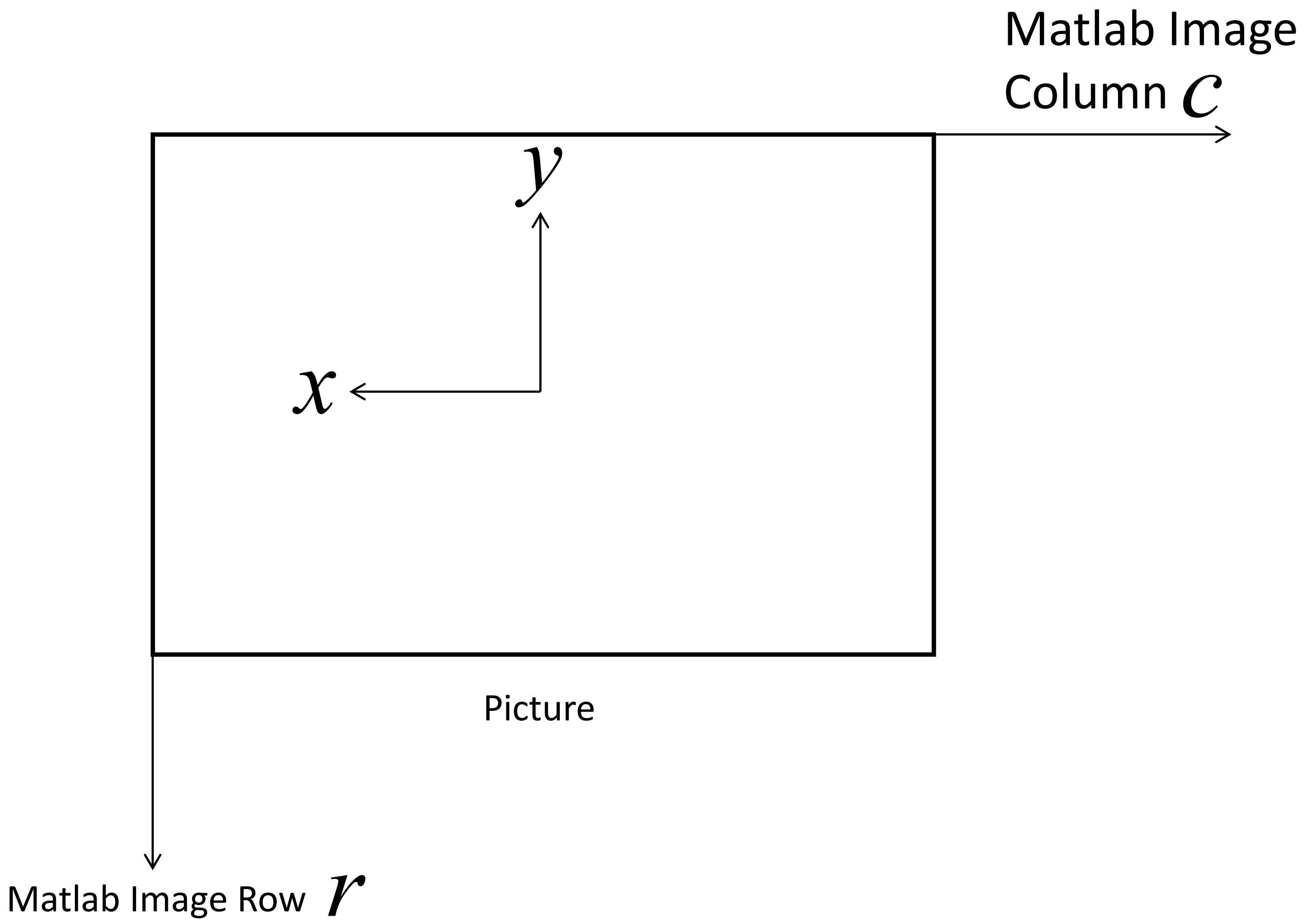


Picture

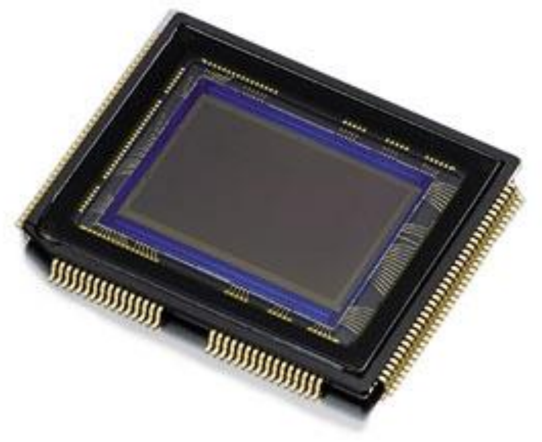
When they take a picture:



When they take a picture:



When they take a picture:



World Unit: e.g. Meters

We let f to take care of this as well.
Unit of f is pixel/world unit.

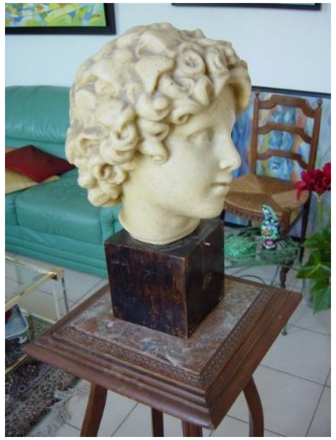
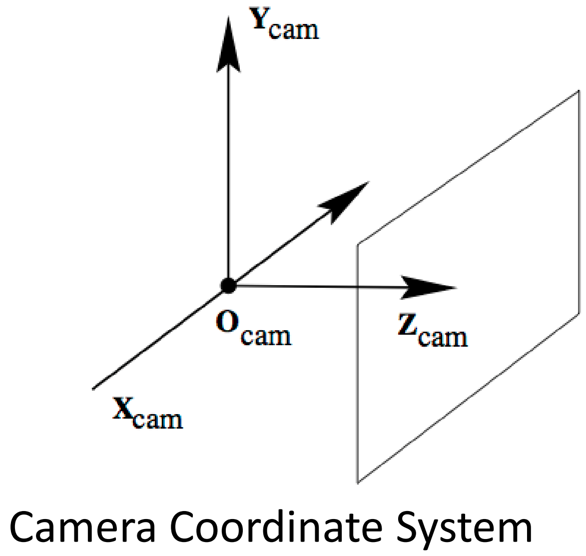
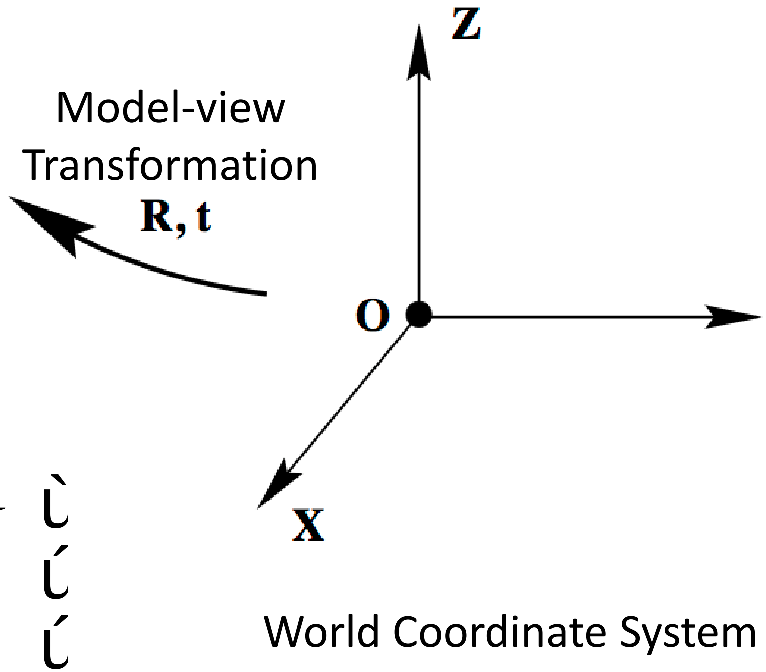


Image Unit: Pixels

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

相机位姿：相对于世界坐标系的刚体变换（旋转+平移）





$$\begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

世界坐标系到相机坐标系

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{K}\begin{bmatrix} \mathbf{R} \mid \mathbf{t} \end{bmatrix}\mathbf{X}$$

Camera Parameter
Camera Projection Matrix

$$\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{R} \mid \mathbf{t} \end{bmatrix}$$

Intrinsic　　Extrinsic

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

空间点在相机平面的成像位置

- 1个相机

$$\mathbf{x} = \mathbf{K}\left[\mathbf{R}\middle|\mathbf{t}\right]\mathbf{X}$$

空间点在相机平面的成像位置

- 2个相机

$$\mathbf{x}_1 = \mathbf{K}\left[\mathbf{R}_1 | \mathbf{t}_1\right]\mathbf{X}$$

$$\mathbf{x}_2 = \mathbf{K}\left[\mathbf{R}_2 | \mathbf{t}_2\right]\mathbf{X}$$

空间点在相机平面的成像位置

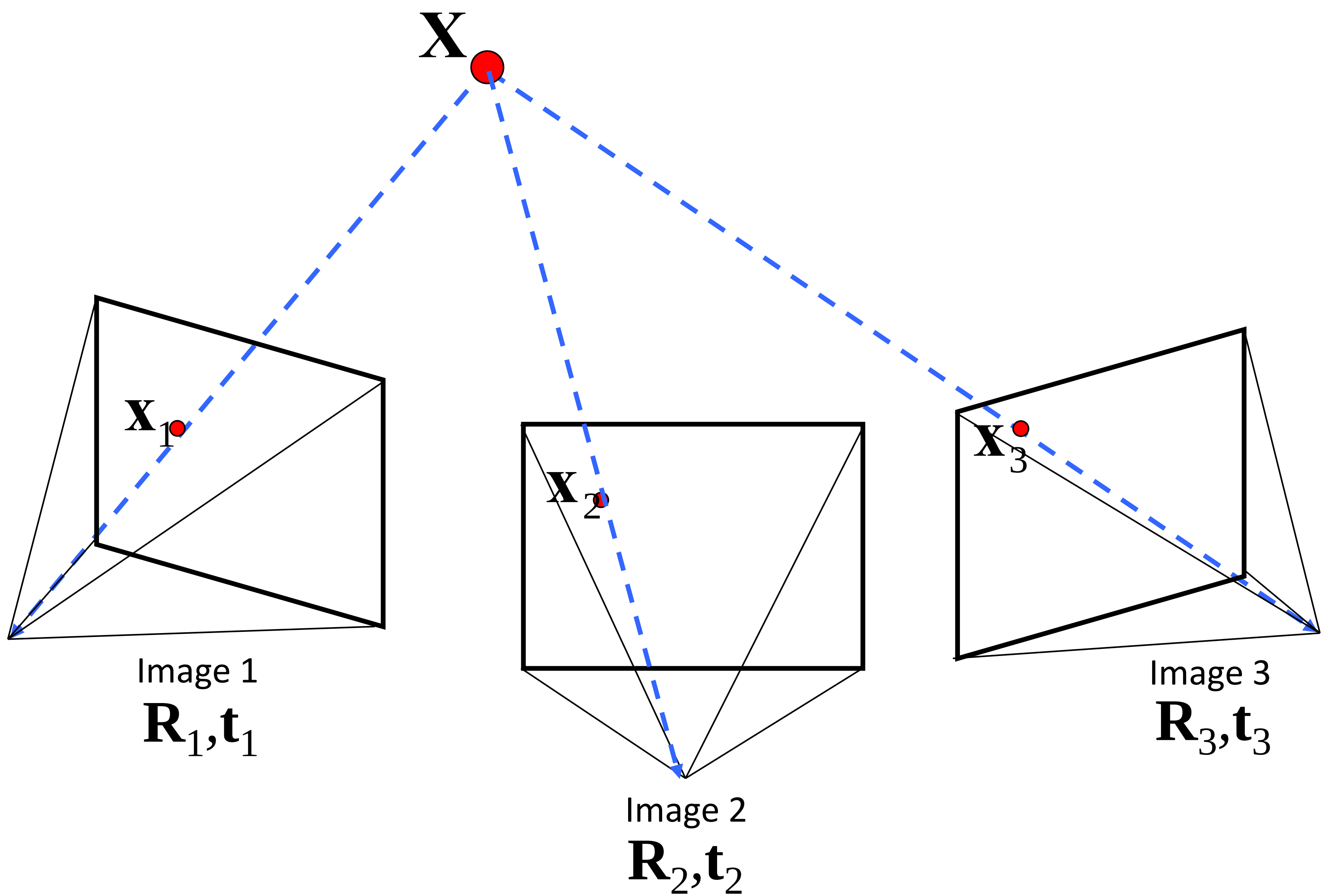
- 3个相机

$$\mathbf{x}_1 = \mathbf{K}\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right]\mathbf{X}$$

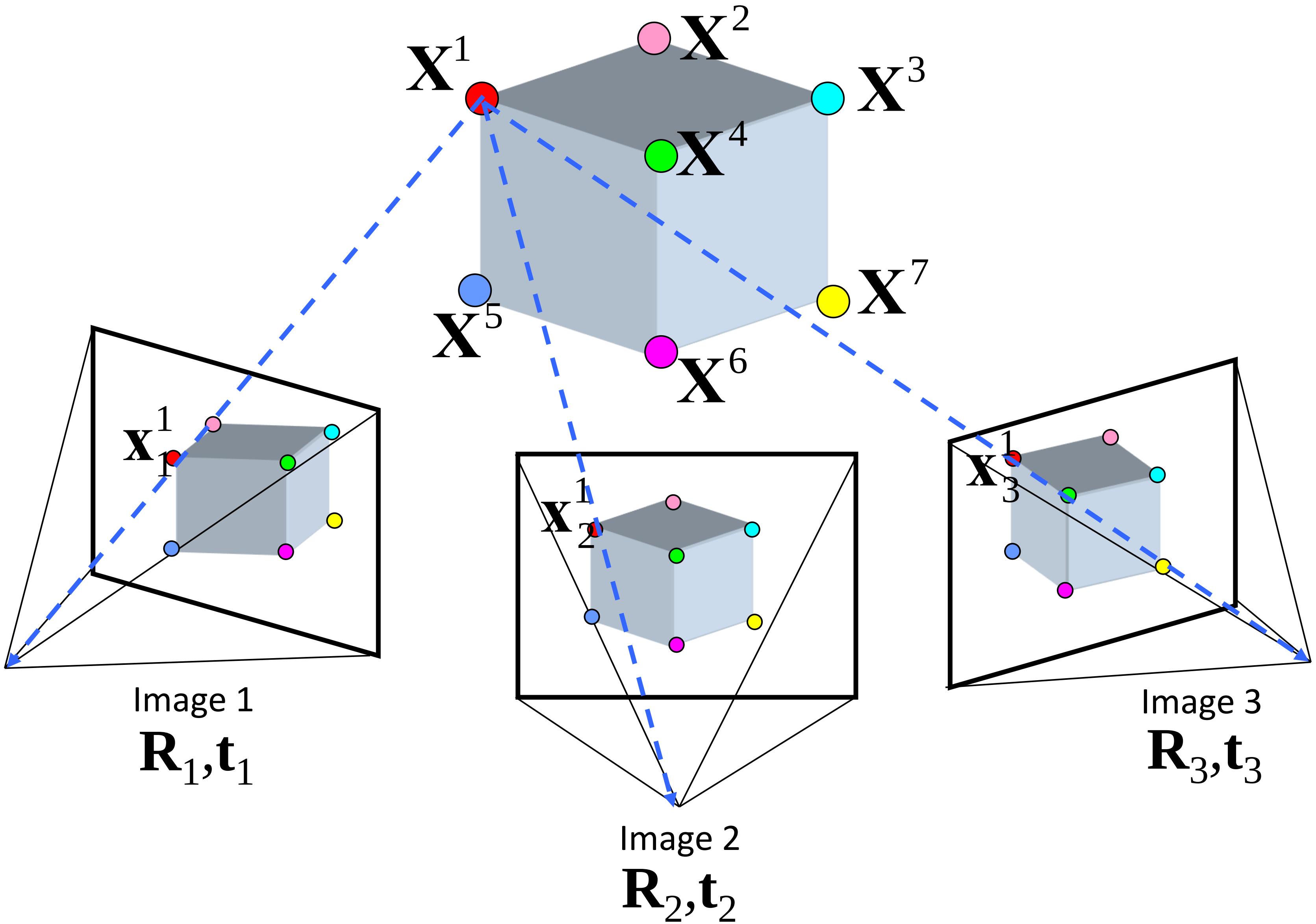
$$\mathbf{x}_2 = \mathbf{K}\left[\mathbf{R}_2 \middle| \mathbf{t}_2\right]\mathbf{X}$$

$$\mathbf{x}_3 = \mathbf{K}\left[\mathbf{R}_3 \middle| \mathbf{t}_3\right]\mathbf{X}$$

图像的公共对应点



图像的公共对应点



$\mathbf{X}^1$
$\mathbf{X}^2$
$\mathbf{X}^3$
$\mathbf{X}^4$
$\mathbf{X}^5$
$\mathbf{X}^6$
$\mathbf{X}^7$
$\mathbf{x}_1^1$
$\mathbf{x}_2^1$
$\mathbf{x}_3^1$
Image 1
$\mathbf{R}_1,\mathbf{t}_1$
Image 2
$\mathbf{R}_2,\mathbf{t}_2$
Image 3
$\mathbf{R}_3,\mathbf{t}_3$

图像的公共对应点

	Point 1	Point 2	Point 3
Image 1	$\mathbf{x}_1^1 = \mathbf{K}\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right]\mathbf{X}^1$	$\mathbf{x}_1^2 = \mathbf{K}\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right]\mathbf{X}^2$	
Image 2	$\mathbf{x}_2^1 = \mathbf{K}\left[\mathbf{R}_2 \middle| \mathbf{t}_2\right]\mathbf{X}^1$	$\mathbf{x}_2^2 = \mathbf{K}\left[\mathbf{R}_2 \middle| \mathbf{t}_2\right]\mathbf{X}^2$	$\mathbf{x}_2^3 = \mathbf{K}\left[\mathbf{R}_2 \middle| \mathbf{t}_2\right]\mathbf{X}^3$
Image 3	$\mathbf{x}_3^1 = \mathbf{K}\left[\mathbf{R}_3 \middle| \mathbf{t}_3\right]\mathbf{X}^1$		$\mathbf{x}_3^3 = \mathbf{K}\left[\mathbf{R}_3 \middle| \mathbf{t}_3\right]\mathbf{X}^3$

Same Camera Same Setting = Same $\mathbf{K}$

标定及重建问题

- Input: Observed 2D image position

$$\tilde{\mathbf{x}}_1^1 \quad \tilde{\mathbf{x}}_1^2$$

$$\tilde{\mathbf{x}}_2^1 \quad \tilde{\mathbf{x}}_2^2 \quad \tilde{\mathbf{x}}_2^3$$

$$\tilde{\mathbf{x}}_3^1 \qquad \quad \tilde{\mathbf{x}}_3^3$$

- Output:

Unknown Camera Parameters (with some guess)

$$\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right], \left[\mathbf{R}_2 \middle| \mathbf{t}_2\right], \left[\mathbf{R}_3 \middle| \mathbf{t}_3\right]$$

Unknown Point 3D coordinate (with some guess)

$$\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \cdots$$

标定及重建问题

A valid solution $[\mathbf{R}_1|\mathbf{t}_1], [\mathbf{R}_2|\mathbf{t}_2], [\mathbf{R}_3|\mathbf{t}_3]$ and $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \cdots$ must let

Re-projection

$$
\begin{cases}
\mathbf{x}_1^1 = \mathbf{K}[\mathbf{R}_1|\mathbf{t}_1]\mathbf{X}^1 & \mathbf{x}_1^2 = \mathbf{K}[\mathbf{R}_1|\mathbf{t}_1]\mathbf{X}^2 & \\
\mathbf{x}_2^1 = \mathbf{K}[\mathbf{R}_2|\mathbf{t}_2]\mathbf{X}^1 & \mathbf{x}_2^2 = \mathbf{K}[\mathbf{R}_2|\mathbf{t}_2]\mathbf{X}^2 & \mathbf{x}_2^3 = \mathbf{K}[\mathbf{R}_2|\mathbf{t}_2]\mathbf{X}^3 \\
\mathbf{x}_3^1 = \mathbf{K}[\mathbf{R}_3|\mathbf{t}_3]\mathbf{X}^1 & & \mathbf{x}_3^3 = \mathbf{K}[\mathbf{R}_3|\mathbf{t}_3]\mathbf{X}^3
\end{cases}
$$

=

Observation

$$
\begin{cases}
\tilde{\mathbf{x}}_1^1 & \tilde{\mathbf{x}}_1^2 & \\
\tilde{\mathbf{x}}_2^1 & \tilde{\mathbf{x}}_2^2 & \tilde{\mathbf{x}}_2^3 \\
\tilde{\mathbf{x}}_3^1 & & \tilde{\mathbf{x}}_3^3
\end{cases}
$$

标定及重建问题：非线性最小二乘优化

A valid solution $\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right], \left[\mathbf{R}_2 \middle| \mathbf{t}_2\right], \left[\mathbf{R}_3 \middle| \mathbf{t}_3\right]$ and $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \cdots$ must let the Re-projection close to the Observation, i.e. to minimize the reprojection error

$$\min \sum_i \sum_j \left(\tilde{\mathbf{x}}_i^j - \mathbf{K}\left[\mathbf{R}_i \middle| \mathbf{t}_i\right] \mathbf{X}^j \right)^2$$

标定及重建问题：非线性最小二乘优化

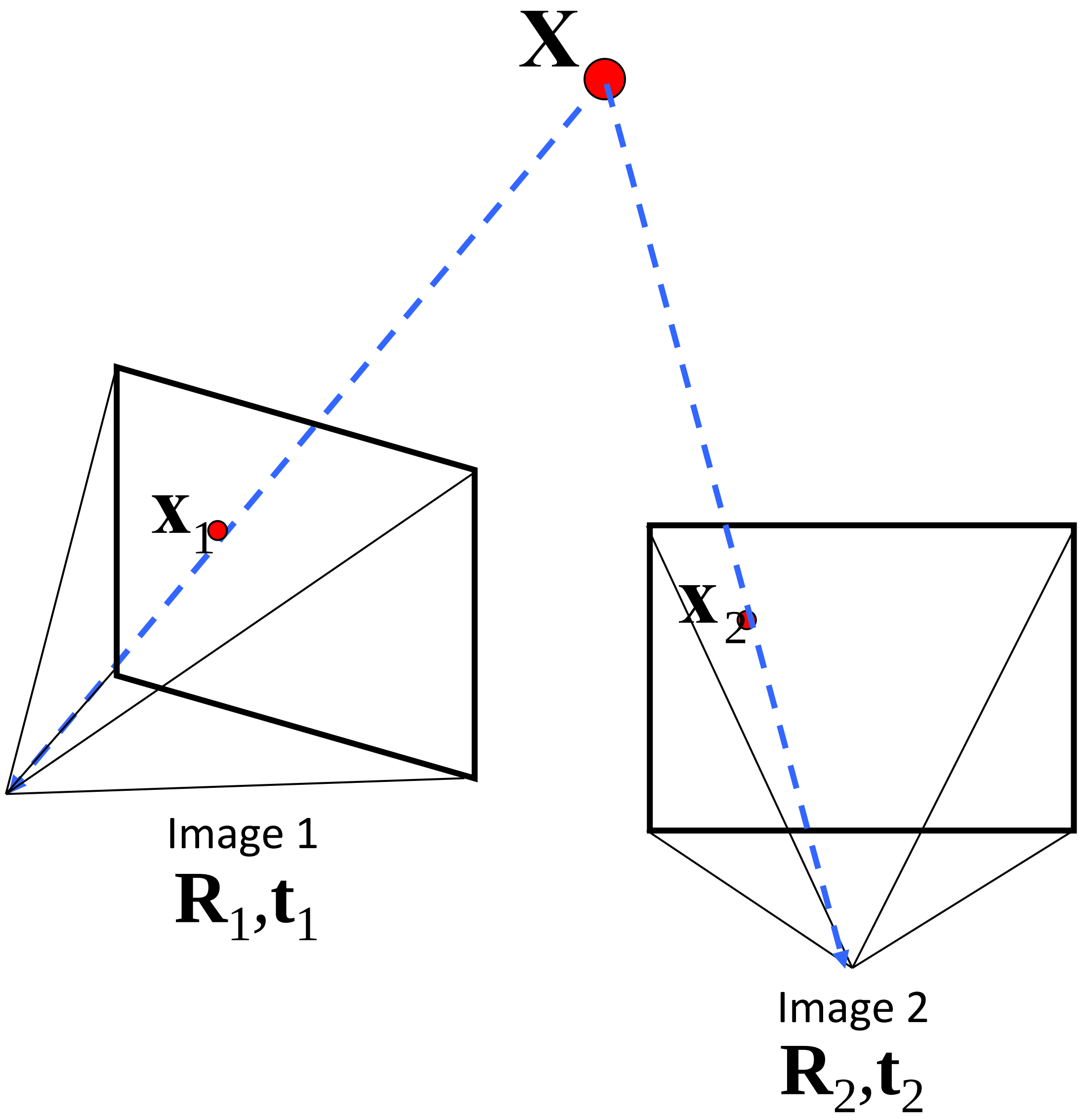
A valid solution $\begin{bmatrix}\mathbf{R}_1 | \mathbf{t}_1\end{bmatrix}, \begin{bmatrix}\mathbf{R}_2 | \mathbf{t}_2\end{bmatrix}, \begin{bmatrix}\mathbf{R}_3 | \mathbf{t}_3\end{bmatrix}$ and $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \cdots$

must let the Re-projection close to the Observation, i.e. to minimize the reprojection error

$$\min \sum_i \sum_j \left(\tilde{\mathbf{x}}_i^j - \mathbf{K}\begin{bmatrix}\mathbf{R}_i | \mathbf{t}_i\end{bmatrix}\mathbf{X}^j \right)^2$$

Question: What is the unit of this objective function?

Fundamental Matrix



$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$$

Estimating Fundamental Matrix

- Given a correspondence

$$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$$

- The basic incidence relation is

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0 \quad \Leftrightarrow \quad [x_1x_2, x_1y_2, x_1, y_1x_2, y_1y_2, y_1, x_2, y_2, 1] \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Need 8 points

Estimating Fundamental Matrix

$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$ for 8 point correspondences:

$$\mathbf{x}_1^1 \leftrightarrow \mathbf{x}_2^1, \mathbf{x}_1^2 \leftrightarrow \mathbf{x}_2^2, \mathbf{x}_1^3 \leftrightarrow \mathbf{x}_2^3, \mathbf{x}_1^4 \leftrightarrow \mathbf{x}_2^4, \mathbf{x}_1^5 \leftrightarrow \mathbf{x}_2^5, \mathbf{x}_1^6 \leftrightarrow \mathbf{x}_2^6, \mathbf{x}_1^7 \leftrightarrow \mathbf{x}_2^7, \mathbf{x}_1^8 \leftrightarrow \mathbf{x}_2^8$$

$$\begin{bmatrix} x_1^1 x_2^1 & x_1^1 y_2^1 & x_1^1 & y_1^1 x_2^1 & y_1^1 y_2^1 & y_1^1 & x_2^1 & y_2^1 & 1 \\ x_1^2 x_2^2 & x_1^2 y_2^2 & x_1^2 & y_1^2 x_2^2 & y_1^2 y_2^2 & y_1^2 & x_2^2 & y_2^2 & 1 \\ x_1^3 x_2^3 & x_1^3 y_2^3 & x_1^3 & y_1^3 x_2^3 & y_1^3 y_2^3 & y_1^3 & x_2^3 & y_2^3 & 1 \\ x_1^4 x_2^4 & x_1^4 y_2^4 & x_1^4 & y_1^4 x_2^4 & y_1^4 y_2^4 & y_1^4 & x_2^4 & y_2^4 & 1 \\ x_1^5 x_2^5 & x_1^5 y_2^5 & x_1^5 & y_1^5 x_2^5 & y_1^5 y_2^5 & y_1^5 & x_2^5 & y_2^5 & 1 \\ x_1^6 x_2^6 & x_1^6 y_2^6 & x_1^6 & y_1^6 x_2^6 & y_1^6 y_2^6 & y_1^6 & x_2^6 & y_2^6 & 1 \\ x_1^7 x_2^7 & x_1^7 y_2^7 & x_1^7 & y_1^7 x_2^7 & y_1^7 y_2^7 & y_1^7 & x_2^7 & y_2^7 & 1 \\ x_1^8 x_2^8 & x_1^8 y_2^8 & x_1^8 & y_1^8 x_2^8 & y_1^8 y_2^8 & y_1^8 & x_2^8 & y_2^8 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{A}\mathbf{f} = \mathbf{0}$$

Direct Linear Transformation (DLT)

Algebraic Error vs. Geometric Error

- Algebraic Error

$$\min \left\| \mathbf{A}\mathbf{f} \right\|$$

- Geometric Error (better)

Unit: pixel

$$\min \sum_{j} d\left(\mathbf{x}_1^j, \mathbf{F}\mathbf{x}_2^j\right)^2 + d\left(\mathbf{x}_2^j, \mathbf{F}^T\mathbf{x}_1^j\right)^2$$

Solved by (non-linear) least square solver (e.g. Ceres)

Solving This Optimization Problem

- Theory:

  The Levenberg–Marquardt algorithm

  http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm

- Practice:

  The Ceres-Solver from Google

  http://code.google.com/p/ceres-solver/

Ceres-solver: A Nonlinear Least Squares Minimizer

Toy problem to solve $\min(10 - x)^2$

```
class SimpleCostFunction
  : public ceres::SizedCostFunction<1 /* number of residuals */,
                                    1 /* size of first parameter */> {
 public:
  virtual ~SimpleCostFunction() {}
  virtual bool Evaluate(double const* const* parameters,
                        double* residuals,
                        double** jacobians) const {
    const double x = parameters[0][0];
    residuals[0] = 10 - x;  // f(x) = 10 - x
    // Compute the Jacobian if asked for.
    if (jacobians != NULL && jacobians[0] != NULL) {
      jacobians[0][0] = -1;
    }
    return true;
  }
};
```

Ceres-solver: A Nonlinear Least Squares Minimizer

Toy problem to solve $\min(10 - x)^2$

```
int main(int argc, char** argv) {
  double x = 5.0;
  ceres::Problem problem;

  // The problem object takes ownership of the newly allocated
  // SimpleCostFunction and uses it to optimize the value of x.
  problem.AddResidualBlock(new SimpleCostFunction, NULL, &x);

  // Run the solver!
  Solver::Options options;
  options.max_num_iterations = 10;
  options.linear_solver_type = ceres::DENSE_QR;
  options.minimizer_progress_to_stdout = true;
  Solver::Summary summary;
  Solve(options, &problem, &summary);
  std::cout << summary.BriefReport() << "\n";
  std::cout << "x : 5.0 -> " << x << "\n";
  return 0;
}
```

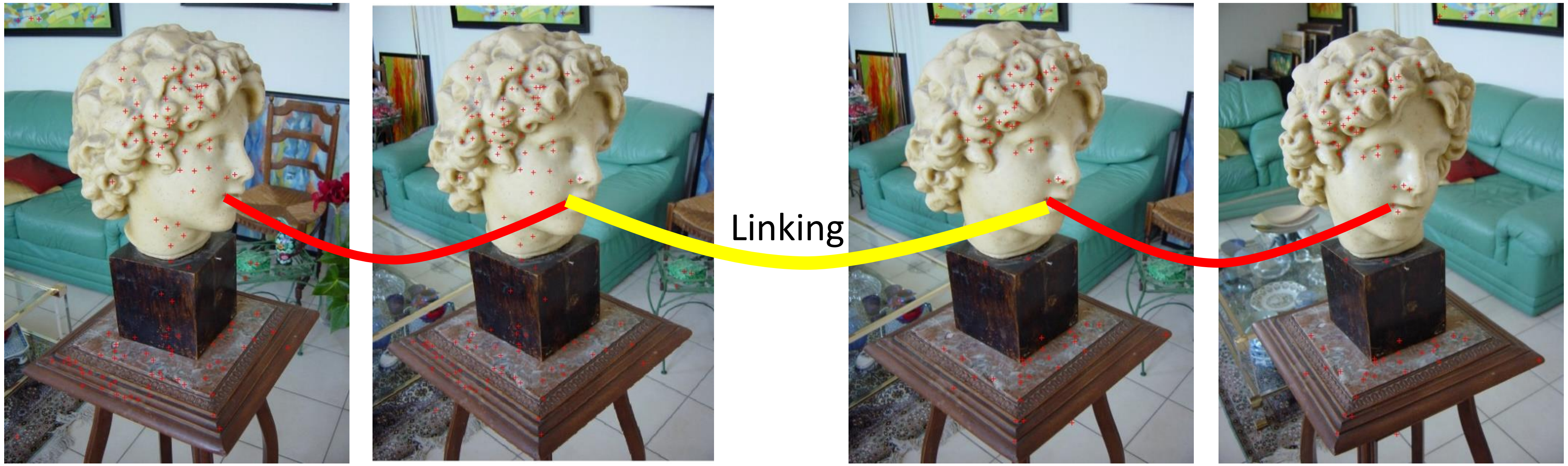
Ceres-solver: A Nonlinear Least Squares Minimizer

Toy problem to solve $$\min(10 - x)^2$$

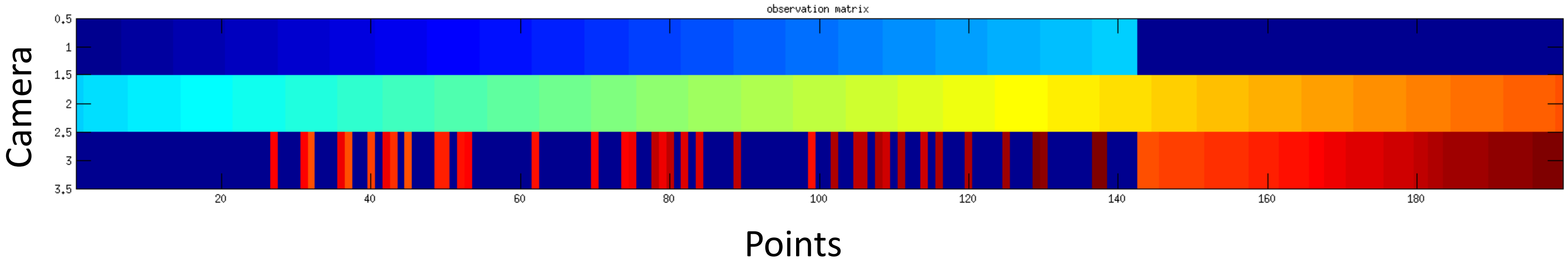
```
0: f: 1.250000e+01 d: 0.00e+00 g: 5.00e+00 h: 0.00e+00 rho: 0.00e+00 mu: 1.00e-04 li:  0
1: f: 1.249750e-07 d: 1.25e+01 g: 5.00e-04 h: 5.00e+00 rho: 1.00e+00 mu: 3.33e-05 li:  1
2: f: 1.388518e-16 d: 1.25e-07 g: 1.67e-08 h: 5.00e-04 rho: 1.00e+00 mu: 1.11e-05 li:  1
Ceres Solver Report: Iterations: 2, Initial cost: 1.250000e+01,  \
Final cost: 1.388518e-16, Termination: PARAMETER_TOLERANCE.
x : 5 -> 10
```

图像对应点问题
（特征点匹配）

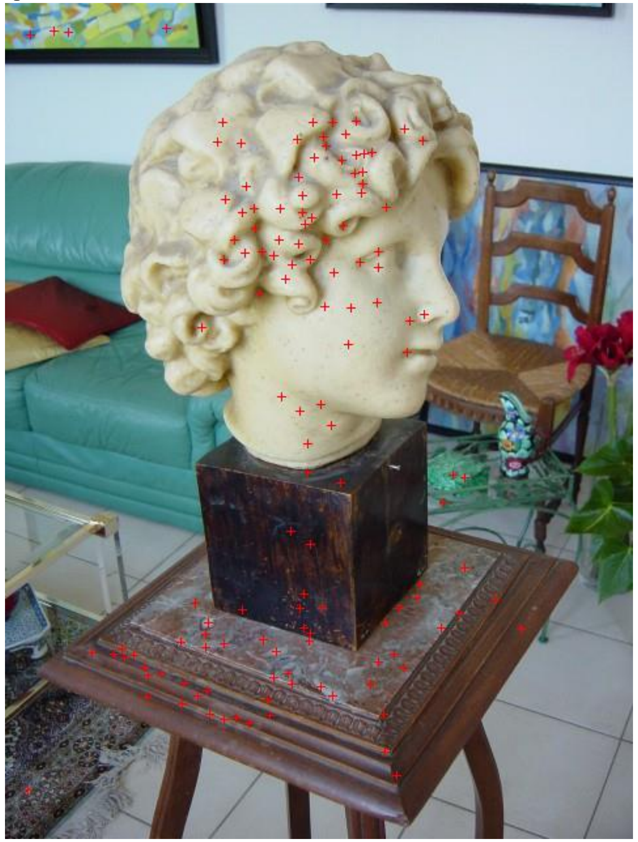
对应点

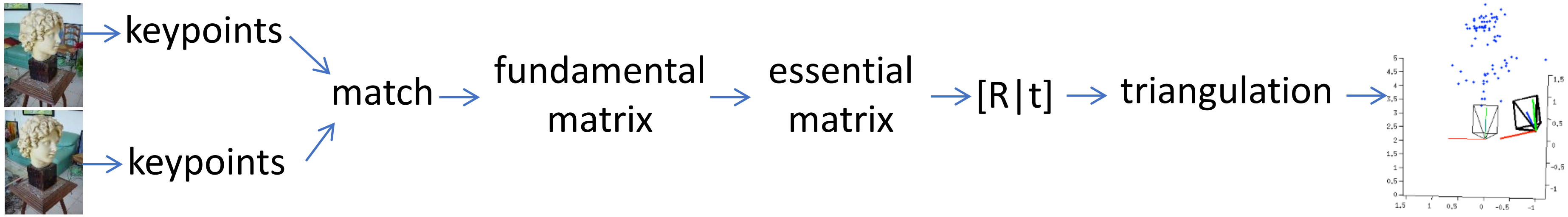


SIFT Matching **SIFT Matching**

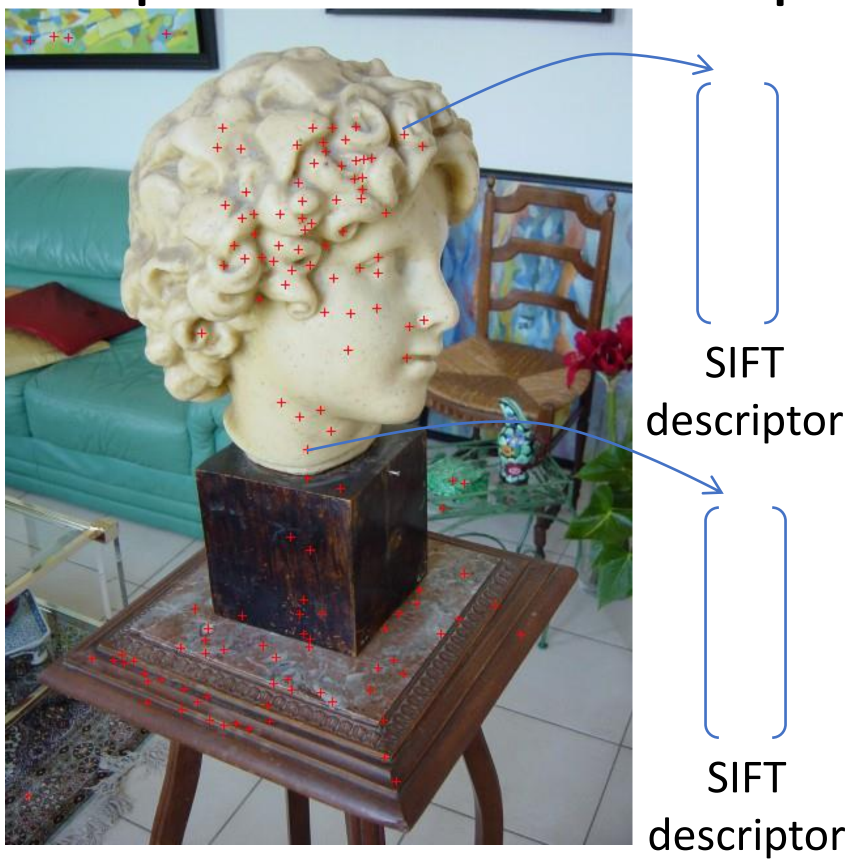


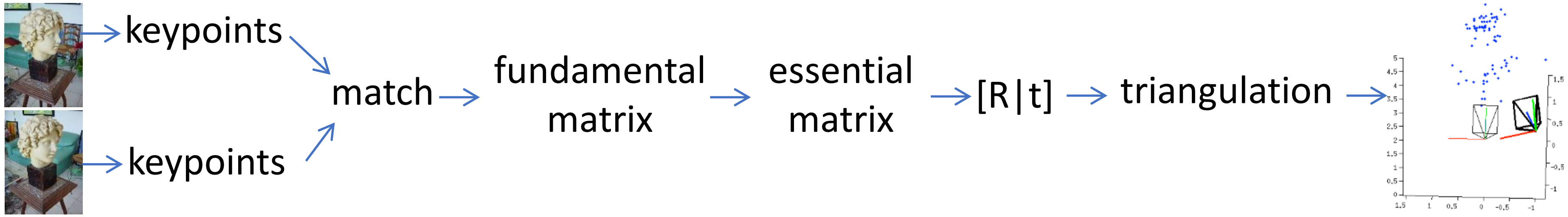
Keypoints Detection



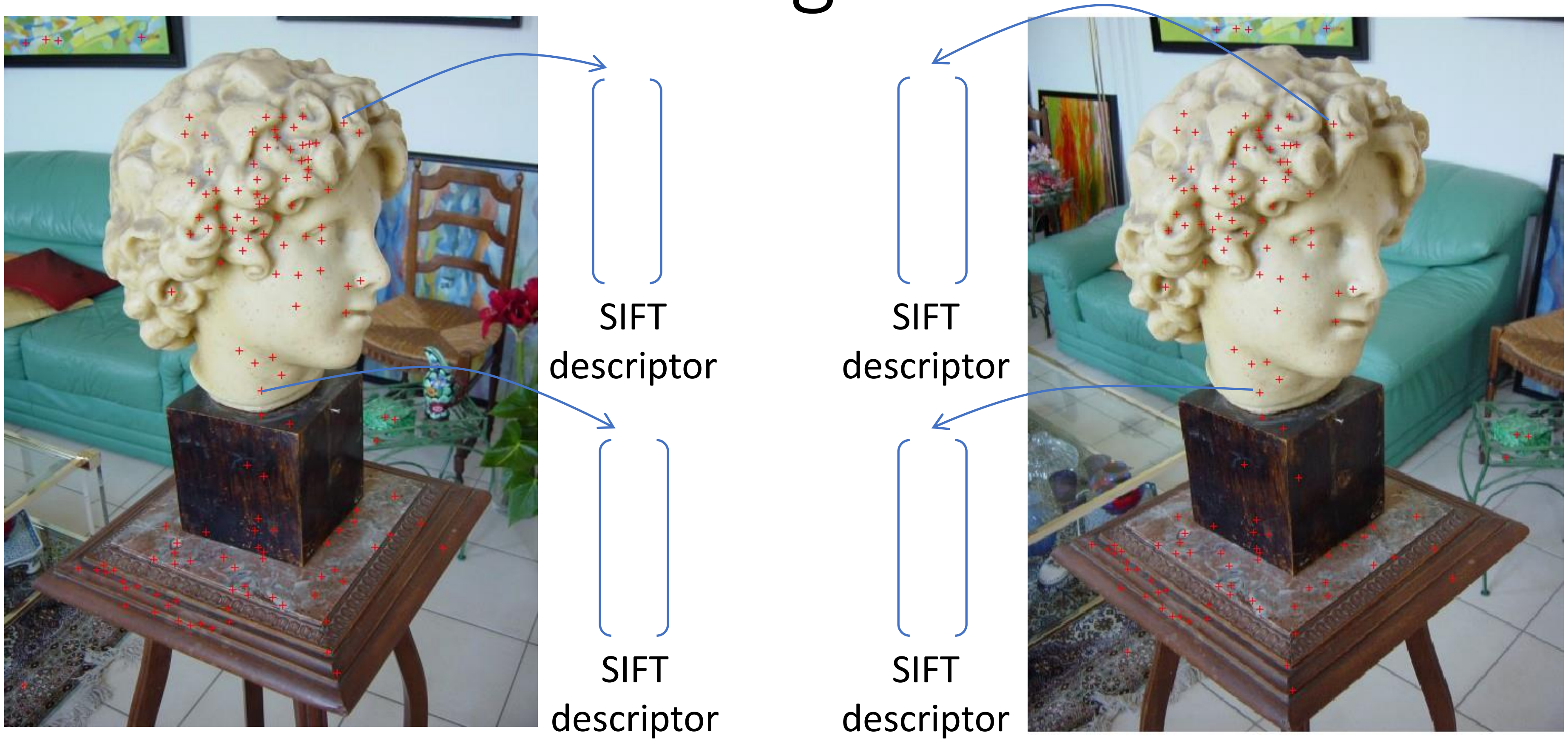


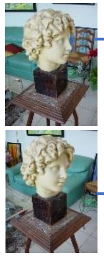
Descriptor for each point



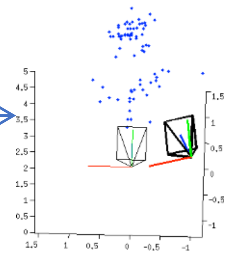


Same for the other images

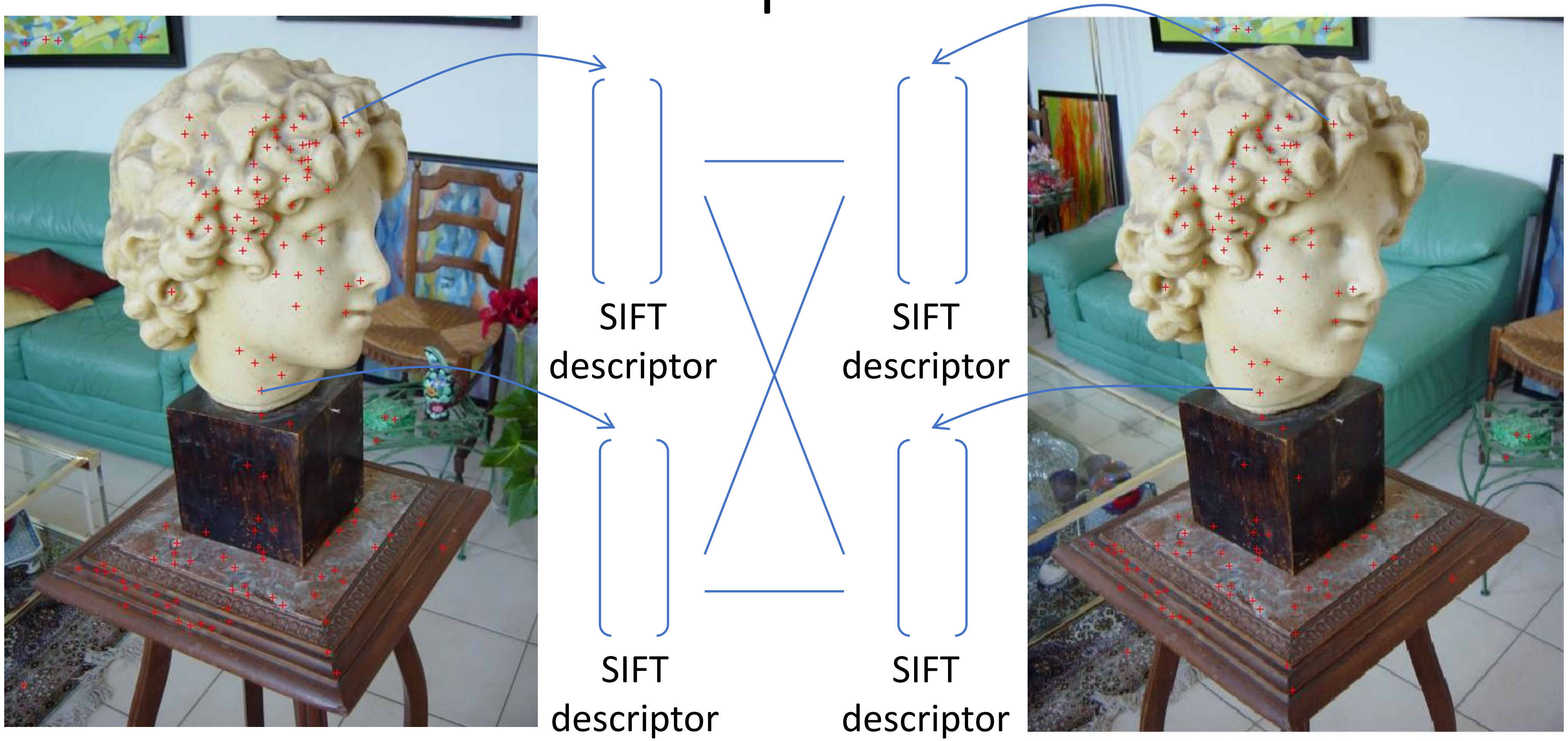


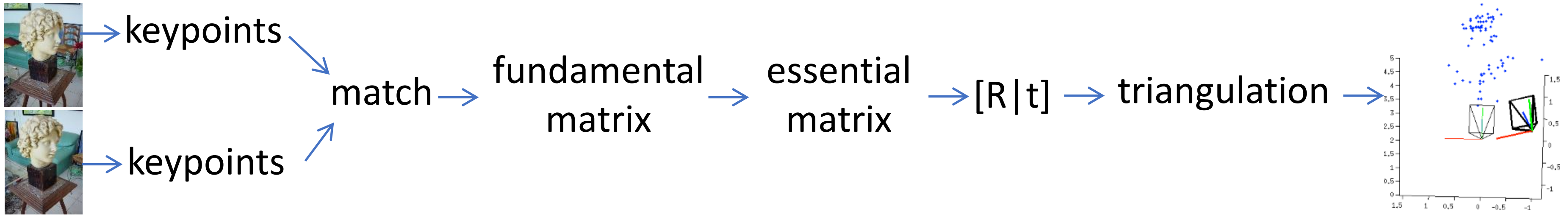


keypoints → match → fundamental matrix → essential matrix → [R|t] → triangulation →
keypoints ↗

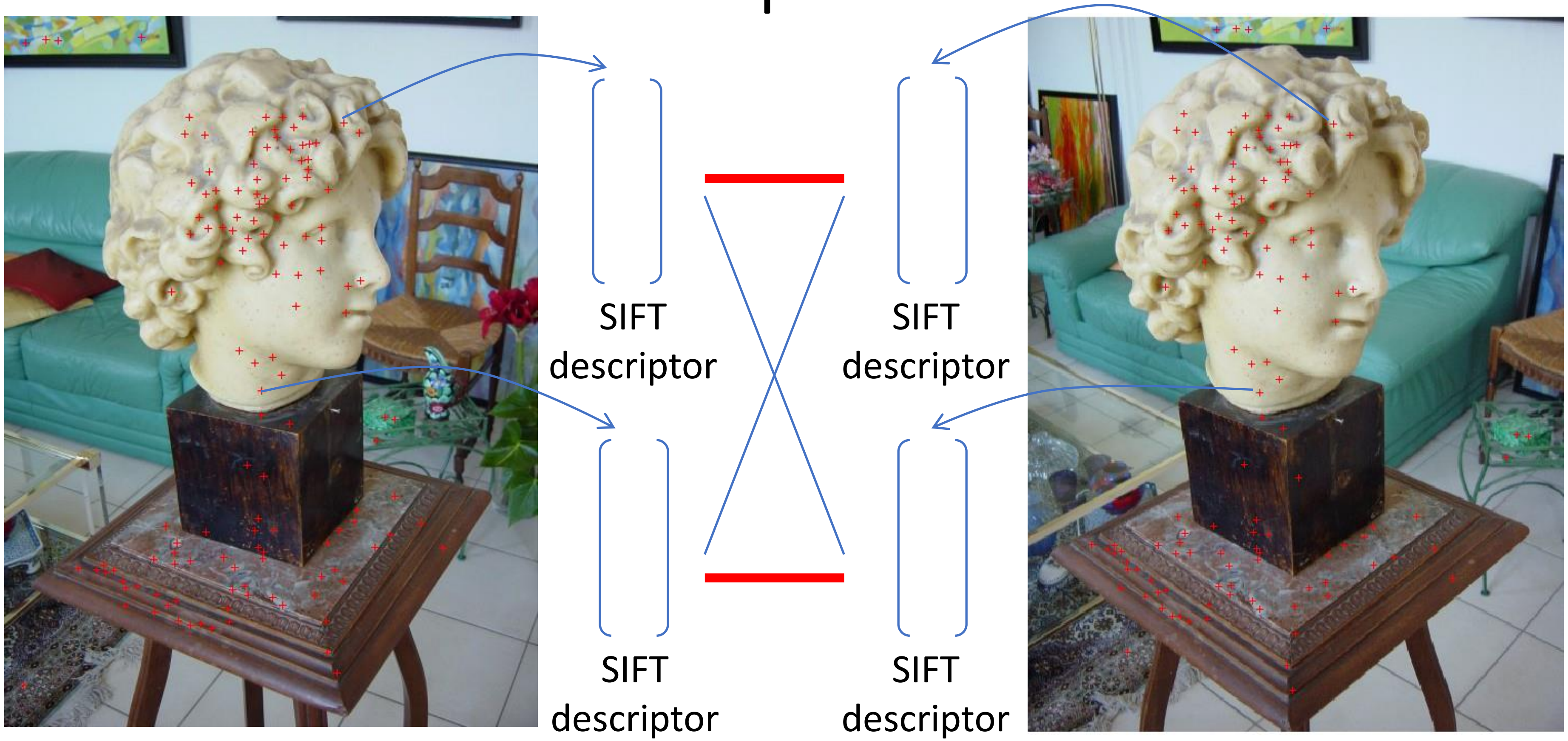


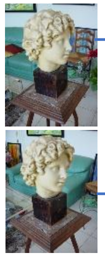
Point Match for correspondences





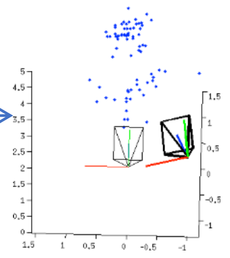
Point Match for correspondences





keypoints → match
keypoints → match

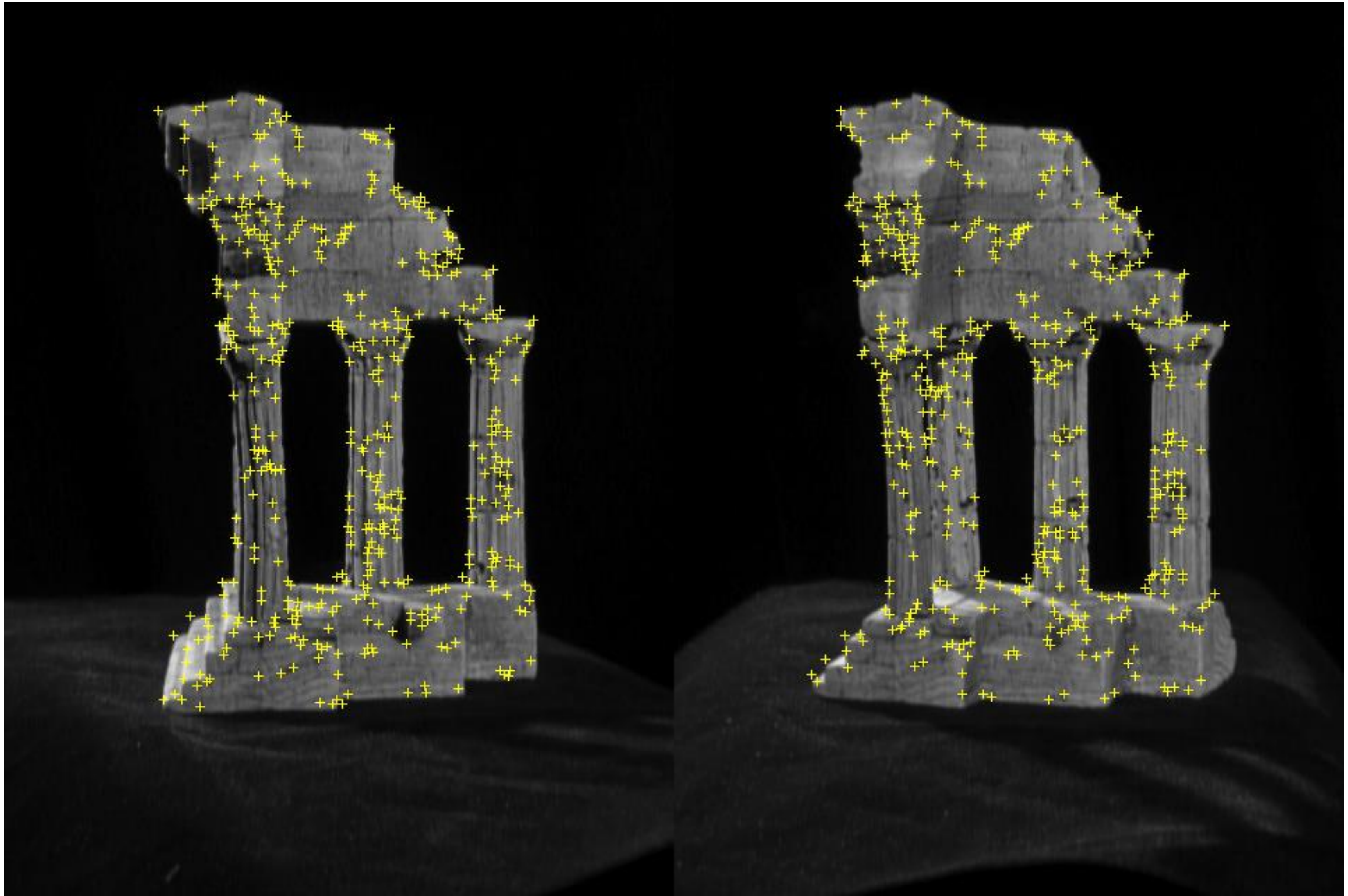
match → fundamental matrix → essential matrix → [R|t] → triangulation →



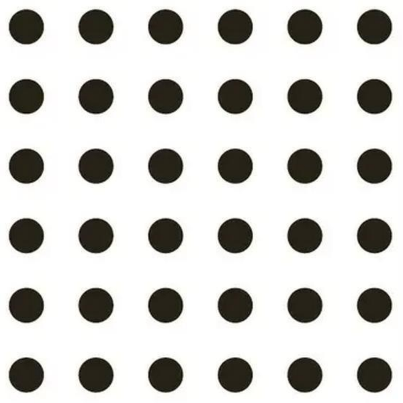
对应点

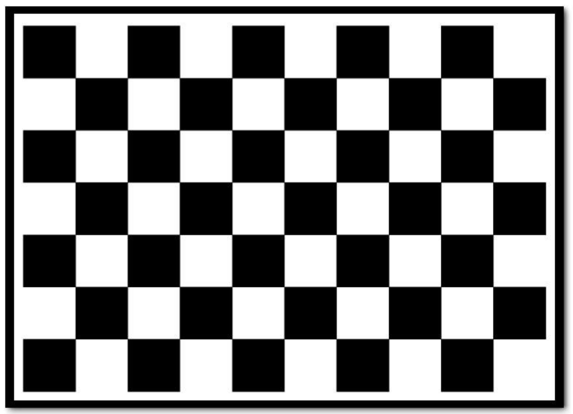
重建点及点颜色

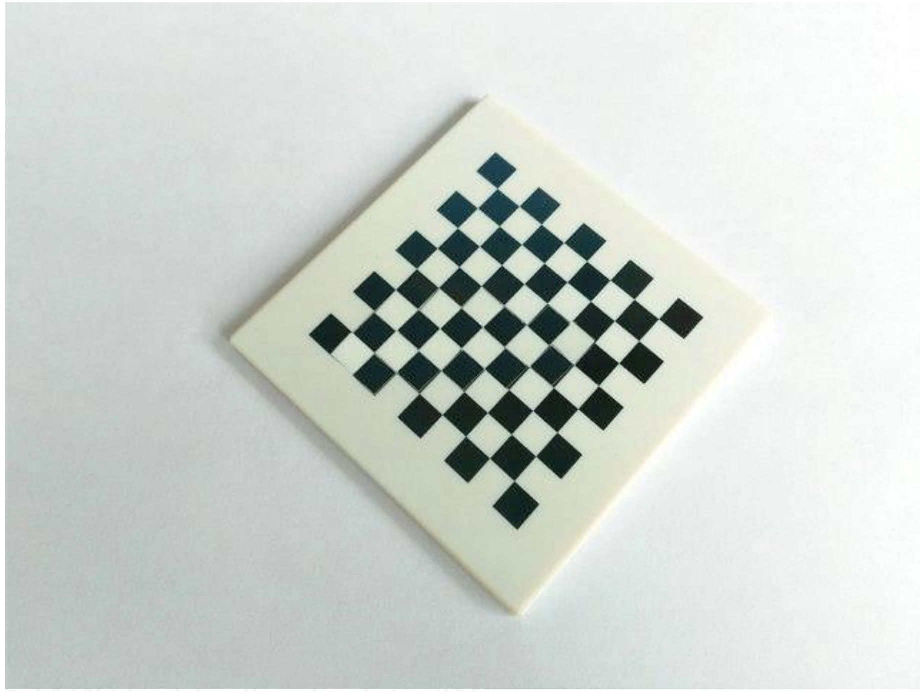
Another Example



简化问题：使用简单的标定板







Other Intrinsic Camera Parameters

- Principle point offset
  - especially when images are cropped (Internet)
- Skew

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{K} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Radial distortion (due to optics of the lens)

$$r^2 = \|\mathbf{x}\|^2 = x^2 + y^2$$

$$\mathbf{x}' = (1 + k_1 r^2 + k_2 r^4)\mathbf{x}$$

Other Camera Models

- Fisheye
- Mirror
- Panorama
- Tilt-Shift Lens
- Biological Eyes

http://www.popgadget.net/2006/07/fisheye_camera.php

http://www.0-360.com/

http://sun360.csail.mit.edu

Canon TS-E 24mm f/3.5L II

Recap: SFMedu Program with Code

Download from: http://3dvision.princeton.edu/courses/SFMedu/

COLMAP：SfM和MVS的开源库
https://github.com/colmap/colmap
https://colmap.github.io/tutorial.html

中国科学技术大学
University of Science and Technology of China

谢　谢　！