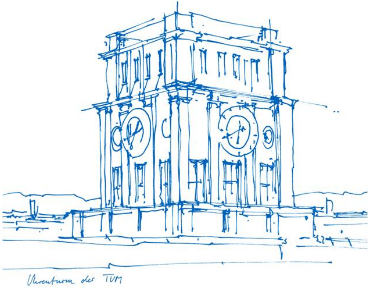
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 06 2D-2D Geometry
(Part 4 Dense Correspondence Search and Homography)

Dr. Haoang Li

15 June 2023 11:00-11:45



Explanation for Pose Definition in Stereo Rectification

➢ Notation

- In our society, $\boldsymbol{R}_{wc}$ and $\boldsymbol{T}_{wc}$ sometimes denote the rotation and translation **from the camera frame to the world frame**, but sometime denote the rotation and translation **from the world frame to the camera frame**.

- In the future classes and final exam, we will use $\boldsymbol{R}_{c\text{->}w}$ and $\boldsymbol{T}_{c\text{->}w}$ to denote the rotation and translation **from the camera frame to the world frame;** We will use $\boldsymbol{R}_{w\text{->}c}$and $\boldsymbol{T}_{w\text{->}c}$ to denote the rotation and translation **from the world frame to the camera frame.**

Explanation for Pose Definition in Stereo Rectification

➢ Equation Validation

- Equations introduced in our last class

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \left(R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \right) \longrightarrow \boxed{\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K R^{-1} \left(\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - T \right)}$$

From world to camera

From camera to world

$$\Rightarrow$$

$$\boxed{K R_{C \to W}^{-1} \left(\begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} - T_{C \to W} \right)}$$

$$= K\left(R_{C \to W}^{-1} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} - R_{C \to W}^{-1} T_{C \to W} \right)$$

$$= K\left(R_{W \to C} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} + T_{W \to C} \right)$$

- Conversion introduced before

$$R_{C \to W}^{-1} = R_{W \to C}$$

$$-R_{C \to W}^{-1} T_{C \to W} = T_{W \to C}$$

Today’s Outline

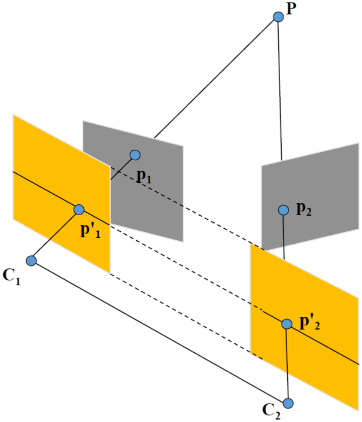
- Dense Correspondence Search
- Homography

Dense Correspondence Establishment

➢ Recap on Stereo Rectification

✓ Image planes are coplanar
✓ Epipolar lines are collinear and horizontal

We can conduct 1D correspondence search!



Dense Correspondence Establishment

一旦左、右图像被矫正，就可以沿着相同的扫描线进行对应搜索。

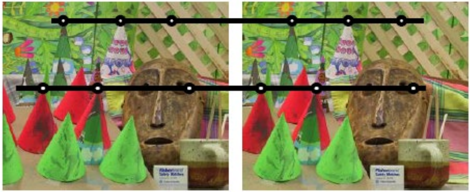
➢ Overview

- 一个直接的策略是计算像素间的相似度。一对与最高相似度（如最小的强度差）相关的像素构成一个点对应。

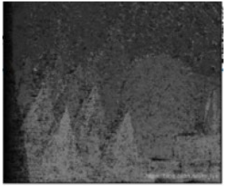
- 一个更可靠的策略是计算块状的相似性。

✓ Once left and right images are rectified, correspondence search can be done **along the same scanlines.**

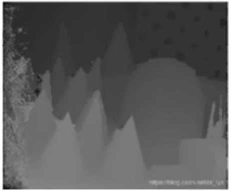
- A straightforward strategy is to compute the **pixel-wise similarity**. A pair of pixels associated with the highest similarity (e.g., smallest intensity difference) constitute a point correspondence.
- A more reliable strategy is to compute the **block-wise similarity**.



Pixel-wise similarity measurement



Disparity result based on **pixel-wise** similarity



Disparity result based on **block-wise** similarity

Dense Correspondence Establishment

➢ Descriptor Similarity Measurement

✓ Scale and viewpoint do not change significantly for a stereo camera.

✓ To average effects of noise or mis-calibration, we can use a **window around the point of interest.**

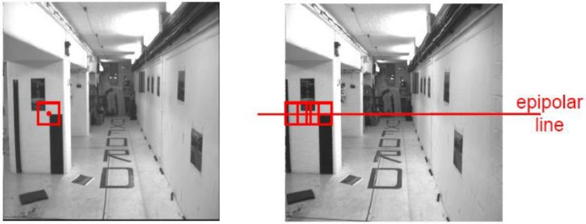
✓ Find a correspondence that minimizes (Z)NCC, (Z)SSD, (Z)SAD, etc.

[?] 对于立体相机来说，比例和视角没有明显变化。

[?] 为了平均噪音或错误校准的影响，我们可以在兴趣点周围使用一个窗口。

[?] 找到一个对应关系，使(Z)NCC、(Z)SSD、(Z)SAD等最小化。

want to find this pixel



along this line find a patch

Dense Correspondence Establishment

➢ General vs. Simplified Cases

✓ General case (sparse correspondences)

- Descriptor

- Descriptors of keypoints may be subject to significant scale change and view point change.

- 描述符
--关键点的描述符可能会有明显的比例变化和视点变化。

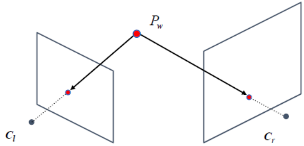
- 位置
- 关键点可以位于图像中的任意位置。
- 我们必须使用二维搜索策略来建立对应关系。

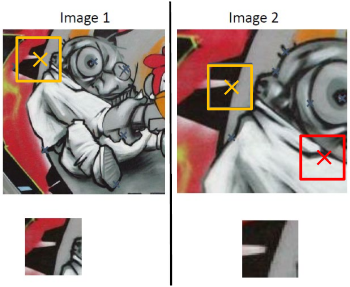
- Position

- Keypoints can lie at arbitrary positions within the image.
- We have to use 2D search strategy to establish correspondences.

General case

(non identical cameras and not aligned)





Dense Correspondence Establishment

➢ General vs. Simplified Cases

✓ Simplified case (dense correspondences)

- Descriptor

- The baseline of a stereo camera is limited.
- Descriptors of keypoints do not have significant scale change and view point change.

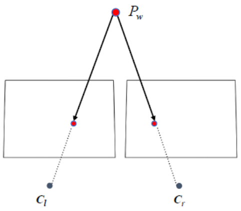
- 描述词
- 立体摄像机的基线是有限的。
- 关键点的描述符没有明显的比例变化和视点变化。

- Position

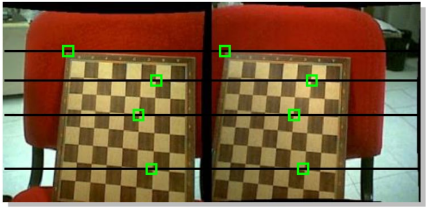
- Based on stereo rectification, a pair of associated points lie on the aligned and horizontal epipolar lines.
- We can use 1D search strategy to establish correspondences.

- 位置
- 基于立体矫正，一对相关的点位于对齐的和水平的双极线上。
- 我们可以使用1D搜索策略来建立对应关系。

Simplified case
(identical cameras and aligned)



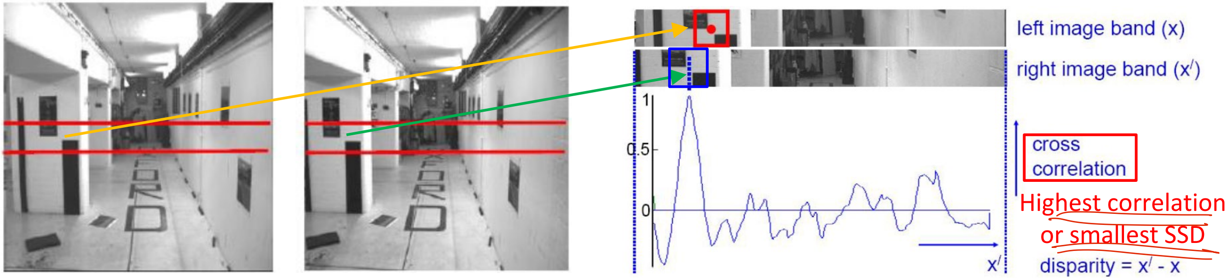
Left Right



Dense Correspondence Establishment

➢ Descriptor Similarity Measurement

✓ Example of optimal matched blocks



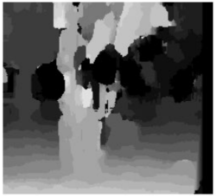
Dense Correspondence Establishment

➢ Effects of window size (W) on the disparity map

✓ Smaller window
- more detail
- but more noise

✓ Larger window
- smoother disparity maps
- but less detail

Smaller window



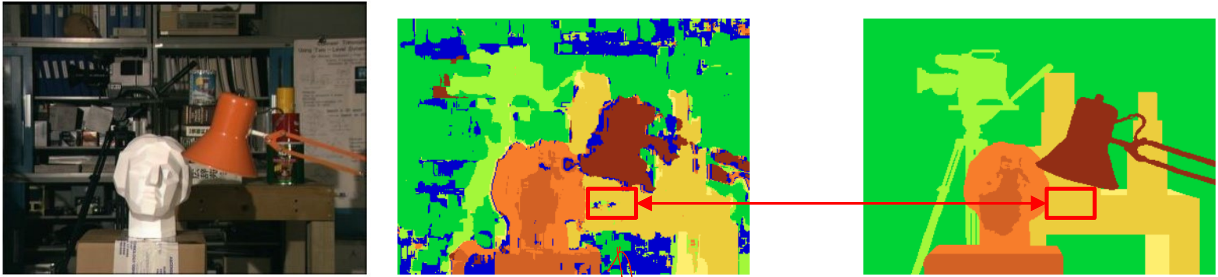
Large window

Dense Correspondence Establishment

➢ Problem of Initial Accuracy

Block matching result is not smooth enough.

Data | Block matching | Ground truth



smooth

Dense Correspondence Establishment

➢ How Can We Improve Window/block-based Matching?

✓ Beyond the epipolar constraint, there are **"soft" constraints** to help identify corresponding points.

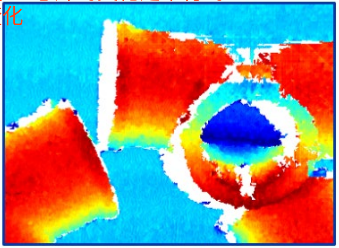
✓ A representative constraint based on **disparity gradient**: Disparity changes **smoothly** between points that lie on the same surface

? 除了极点约束外，还有一些 "软 "约束来帮助识别相应的点。

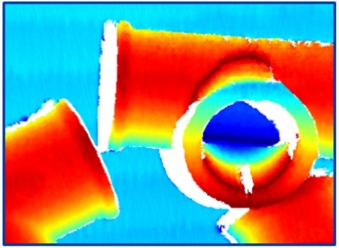
? 一个基于悬殊梯度的代表性约束：悬殊的变化是平稳的

位于同一表面上的各点之间平稳变化

With out smoothness constraint



Without FlexView



FlexView1

Smoothness constraint

Dense Correspondence Establishment

➢ How Can We Improve Window/block-based Matching?

(1)

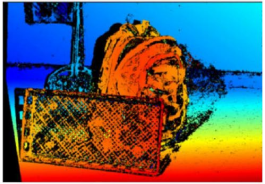
✓ An effective method: **semi-global** matching (SGM)

global without block matching

- Main idea: Perform **block matching** followed by **regularization e.g. smoothing**.
- Another strategy of "global" matching **skips the block matching**. It starts from pixel-wise similarity, followed by **global smoothing** (e.g., graph-cut methods).

Left Image

Right Image



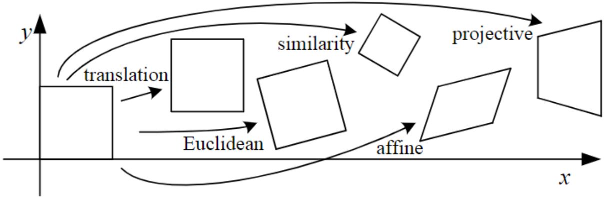
Estimated Disparity

This knowledge will not be asked in the exam.

Homography

- Overview
- ✓ Homography is a transformation of point correspondences (typically, we talk about 2D-2D correspondences).
- ✓ It is derived based on perspective projection (more general than Affine transformation).
- ✓ It encodes the **co-planarity** information.

? Homography是一种点对应关系的转换（通常，我们谈论的是二维-二维对应关系）。
? 它是在透视投影的基础上得出的（比Affine变换更普遍）。
? 它编码了共面性信息。



Homography

➢ A Common Definition (2D-2D)

normal + distance

✓ 3D plane expression

3D point in the **left camera frame**

$$n^T P + d = 0$$

✓ Projective geometry

$$p_2 = K(RP + t) = K\left(RP + t\boxed{\frac{n^T P}{-d}}\right)$$

1

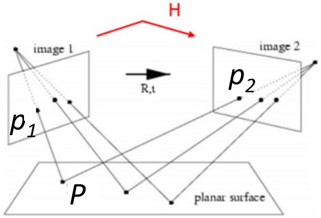
Homogeneous coordinates

Perspective projection in the **right** camera frame

$$p_2 = K\left(R + t\frac{n^T}{-d}\right)\boxed{P} = K\left(R + t\frac{n^T}{-d}\right)\boxed{K^{-1}p_1}$$

Distributive law

Normalized image coordinates

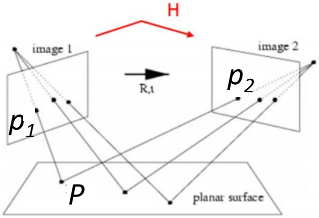


3D point P is projected to both left and right image planes

Homography

➢ A More Common Definition (2D-2D)

✓ Conclusion



We define homography matrix *H* as

$$p_2 = \boxed{K\left(R + t\frac{n^T}{-d}\right)K^{-1}} p_1 \quad \Rightarrow \quad \begin{aligned} p_2 &= \boxed{H} p_1 \\ H &= K\left(R + t\frac{n^T}{-d}\right)K^{-1} \end{aligned}$$

Homography encodes the relative camera pose information.

Homography

- Computation (2D-2D)

✓ A pair of points in homogeneous coordinates satisfy the homograhpy

$$q_2 \propto \mathbf{H} q_1$$

✓ We expand the above constraint

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \propto \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} \\ \mathbf{H}_{31} & \mathbf{H}_{32} & \boxed{\mathbf{H}_{33}} \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

Homography is up to scale and thus has 8 degrees of freedom

Homography

➢ Computation (2D-2D)

✓ Without loss of generality, we fix the last element of Homography to 1:

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \propto \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} \\ \mathbf{H}_{31} & \mathbf{H}_{32} & \mathbf{H}_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \propto \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & \boxed{1} \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

✓ We re-write the matrix form into two constraints

$$\begin{cases} u_2 = \dfrac{H_{11}u_1 + H_{12}v_1 + H_{13}}{H_{31}u_1 + H_{32}v_1 + 1} \\ v_2 = \dfrac{H_{21}u_1 + H_{22}v_1 + H_{23}}{H_{31}u_1 + H_{32}v_1 + 1} \end{cases}$$

Homography

$$\begin{cases} u_2 = \dfrac{H_{11}u_1 + H_{12}v_1 + H_{13}}{H_{31}u_1 + H_{32}v_1 + 1} \\ v_2 = \dfrac{H_{21}u_1 + H_{22}v_1 + H_{23}}{H_{31}u_1 + H_{32}v_1 + 1} \end{cases}$$

➢ Computation (2D-2D)

✓ Each point correspondence provides **two** linear constraint.

✓ Linear system w.r.t. elements of Homography defined by **four** point correspondences.

$$\begin{pmatrix} u_1^1 & v_1^1 & 1 & 0 & 0 & 0 & -u_1^1u_2^1 & -v_1^1u_2^1 \\ 0 & 0 & 0 & u_1^1 & v_1^1 & 1 & -u_1^1v_2^1 & -v_1^1v_2^1 \\ u_1^2 & v_1^2 & 1 & 0 & 0 & 0 & -u_1^2u_2^2 & -v_1^2u_2^2 \\ 0 & 0 & 0 & u_1^2 & v_1^2 & 1 & -u_1^2v_2^2 & -v_1^2v_2^2 \\ u_1^3 & v_1^3 & 1 & 0 & 0 & 0 & -u_1^3u_2^3 & -v_1^3u_2^3 \\ 0 & 0 & 0 & u_1^3 & v_1^3 & 1 & -u_1^3v_2^3 & -v_1^3v_2^3 \\ u_1^4 & v_1^4 & 1 & 0 & 0 & 0 & -u_1^4u_2^4 & -v_1^4u_2^4 \\ 0 & 0 & 0 & u_1^4 & v_1^4 & 1 & -u_1^4v_2^4 & -v_1^4v_2^4 \end{pmatrix} \begin{pmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{pmatrix} = \begin{pmatrix} u_2^1 \\ v_2^1 \\ u_2^2 \\ v_2^2 \\ u_2^3 \\ v_2^3 \\ u_2^4 \\ v_2^4 \end{pmatrix}$$

Homography

➢ Essential Matrix vs. Homography (2D-2D)

? 相似性
- 它们都是对相对姿态信息的编码。
- 不同的点对应关系可以由同一个矩阵来拟合。

? 差异
- 基本矩阵是由任意的三维点得出的。
- 同位素是由共面的三维点得出的。
- 基本矩阵的计算需要至少5个点的对应关系。
- 同位素计算需要至少4个点的对应关系。

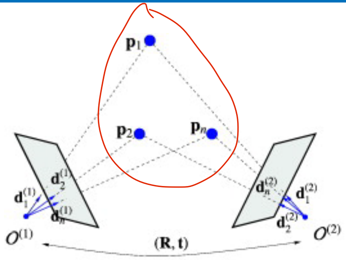
✓ Similarity

- They both encodes the relative pose information.
- Different point correspondences can be fitted by the same matrix.

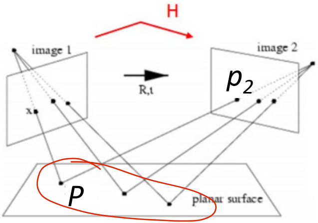
✓ Difference

- Essential matrix is derived from arbitrary 3D points.
- Homography is derived from coplanar 3D points.

- Essential matrix computation needs at least 5 point correspondences.
- Homography computation needs at least 4 point correspondences.



Arbitrary 3D points



3D points lying on the same 3D plane

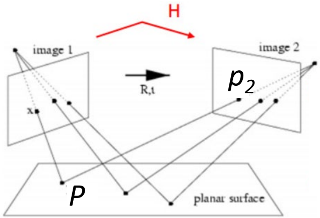
Homography

➢ Recovering Camera Pose from Homography (2D-2D)

✓ Recall that Homography encodes the camera pose information

$$p_2 = \boxed{H} p_1$$

$$H = K\left(R + t\frac{n^T}{-d}\right)K^{-1}$$



✓ Assume that we have computed homograhpy, we aim to recover rotation and translation.

✓ Two representative methods: [1] (popular method) and [2]

[1] Faugeras O D, Lustman F. Motion and structure from motion in a piecewise planar environment. 1988
[2] Ezio Malis, Manuel Vargas, and others. Deeper understanding of the homography decomposition for vision-based control. 2007

Homography decomposition will not be asked in the exam.

Summary

- Dense Correspondence Search
- Homography

Thank you for your listening!

If you have any questions, please come to me :-)