上海交通大学

计算机视觉

教师：赵旭

班级：AI4701

2024 春

7. 立体视觉深度估计

主要内容

- 立体视觉介绍
- 立体视觉深度估计
- 深度摄像机

什么视觉线索可以帮助我们感知三维形状和深度?

- 光照与阴影
  - 光度测量学立体视觉: 从多张阴影图像中恢复三维信息

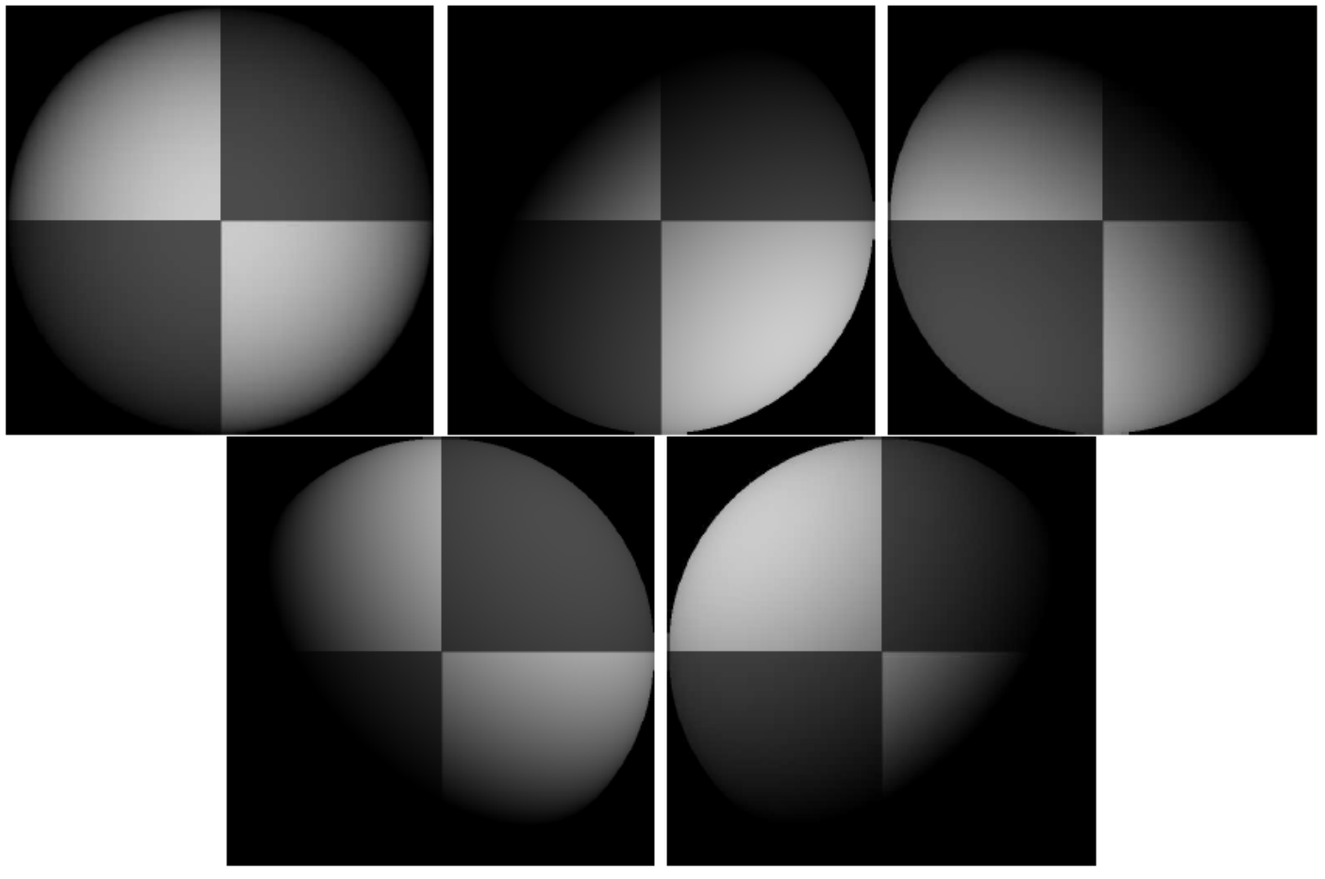
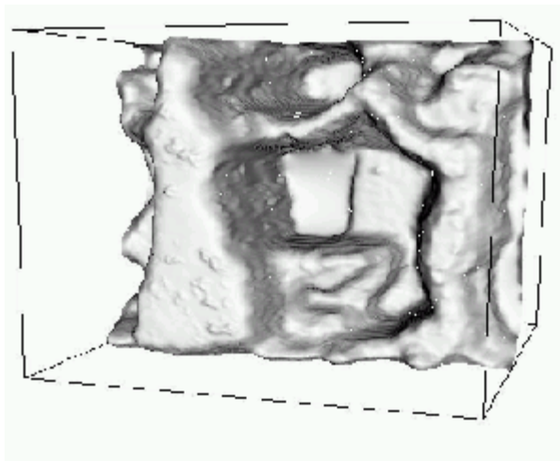
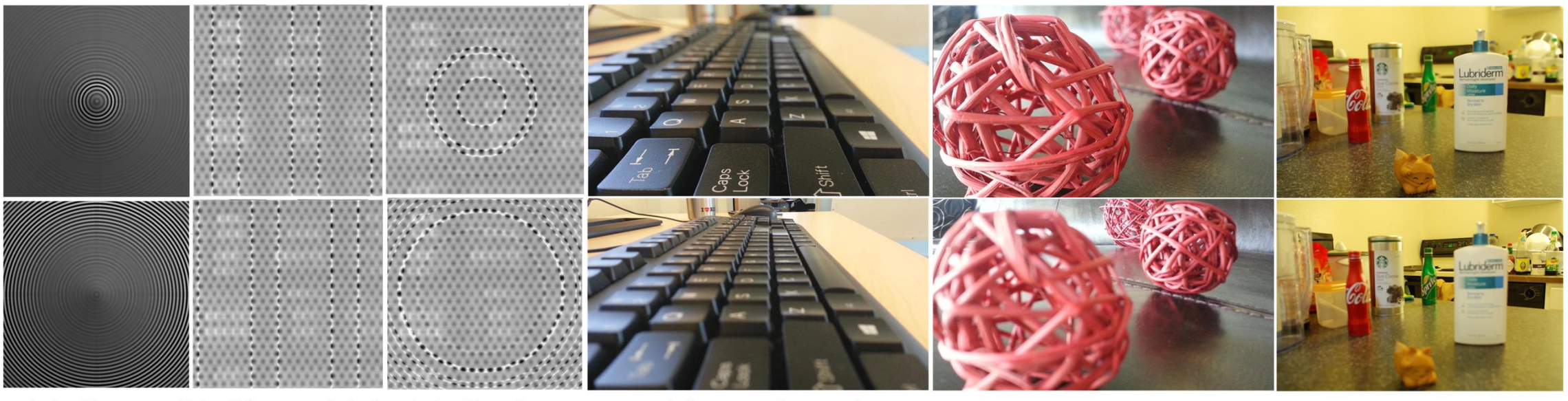


FIGURE 2.11: Five synthetic images of a sphere, all obtained in an orthographic view from the same viewing position. These images are shaded using a local shading model and a distant point source. This is a convex object, so the only view where there is no visible shadow occurs when the source direction is parallel to the viewing direction. The variations in brightness occuring under different sources code the shape of the surface.

什么视觉线索可以帮助我们感知三维形状和深度?

- 从焦点的变化中感知三维信息

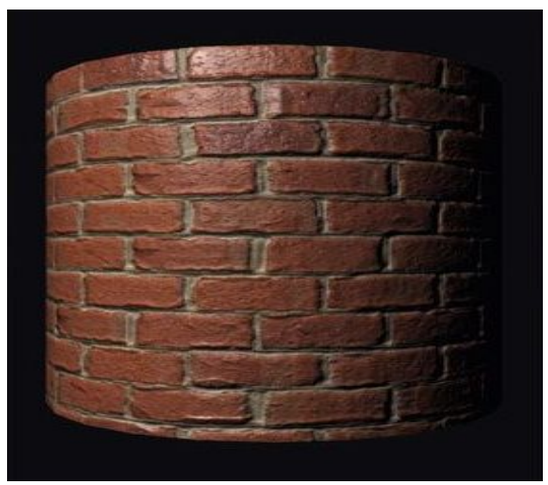


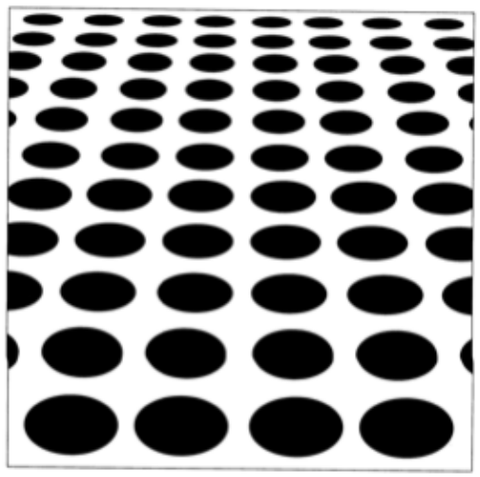


(a) Cone (b) Sinusoidal (c) Cosine (d) Keyboard (e) Balls (f) Kitchen

什么视觉线索可以帮助我们感知三维形状和深度?

- 从纹理结构中恢复三维信息
  - 图像线索：形状的变化、大小的变化、纹理基元密度的变化
  - 生成：表面形状和方向





什么视觉线索可以帮助我们感知三维形状和深度?

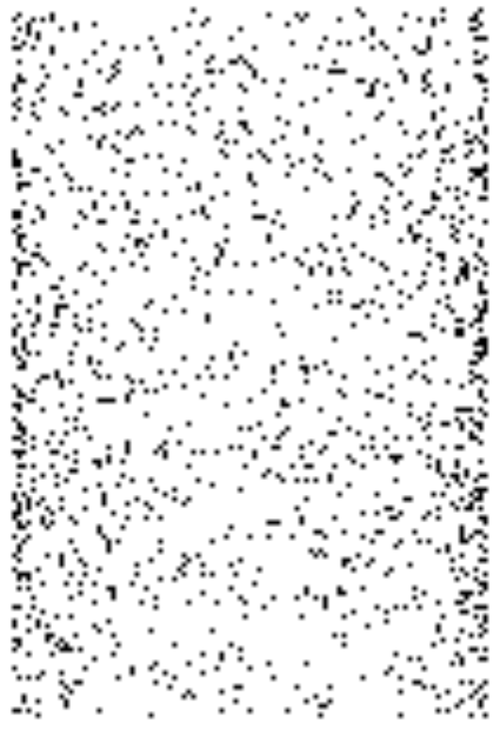
- 透视投影效应

Image credit: S. Seitz

什么视觉线索可以帮助我们感知三维形状和深度?

- 运动

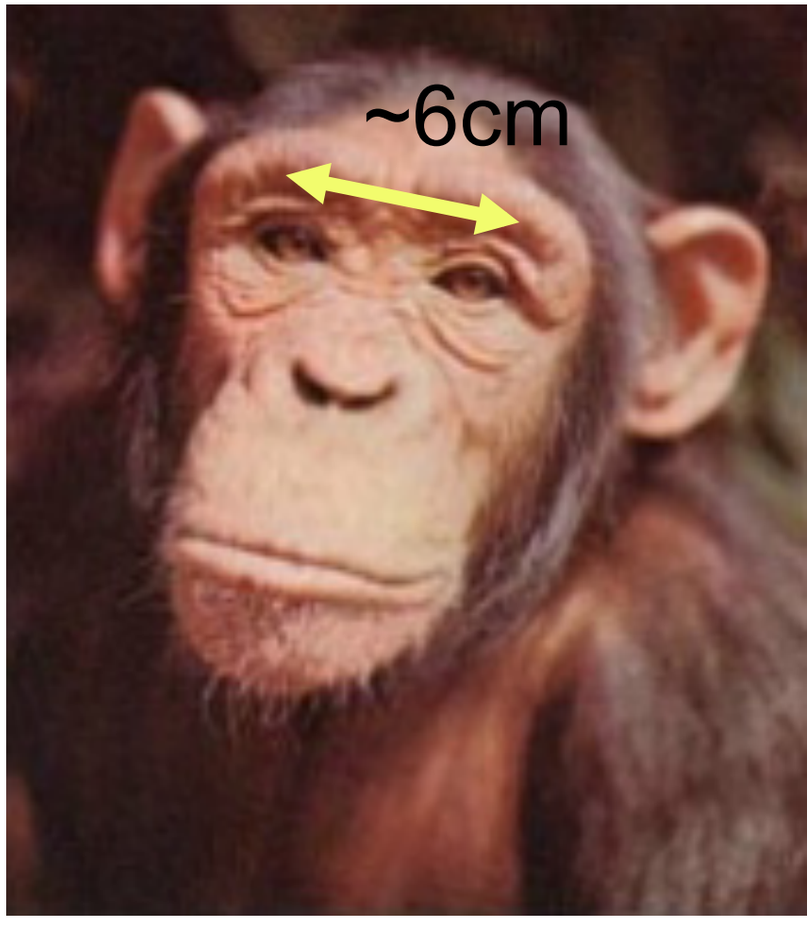


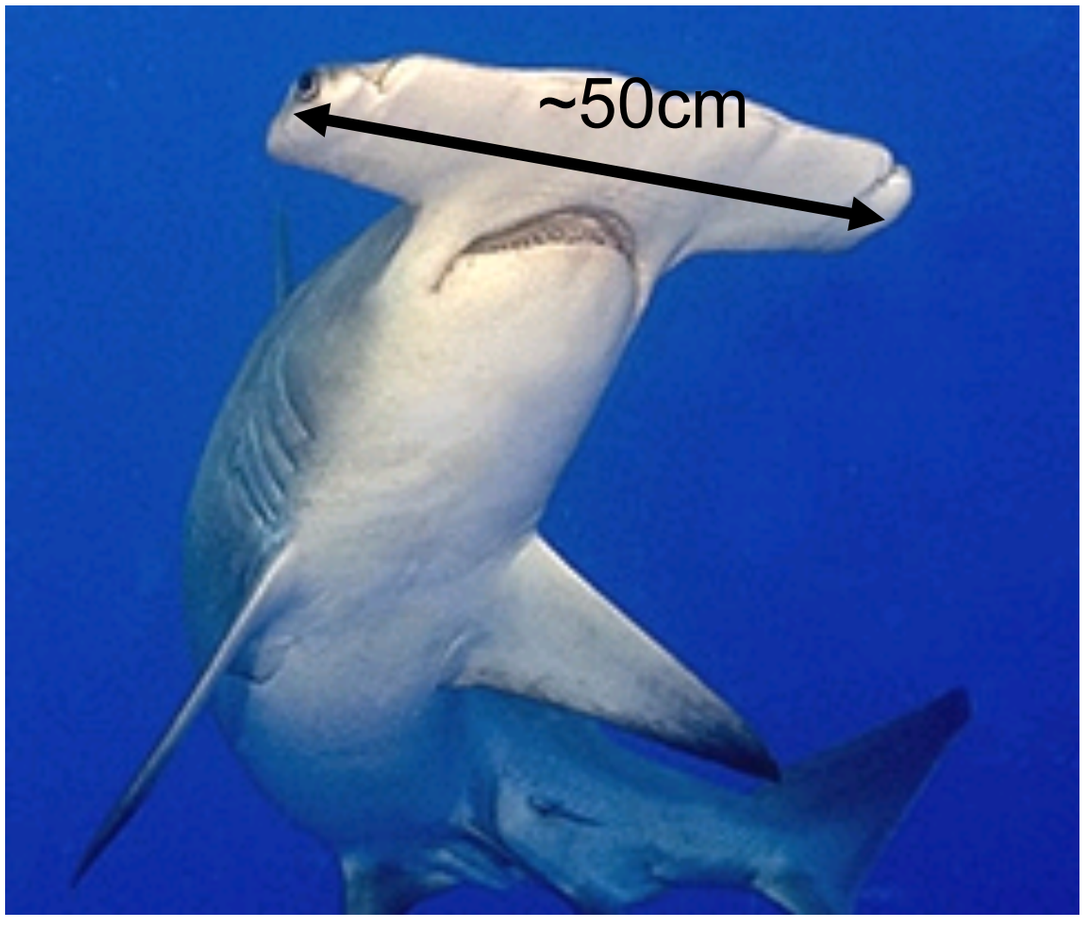



Figures from L. Zhang

http://www.brainconnection.com/teasers/?main=illusion/motion-shape

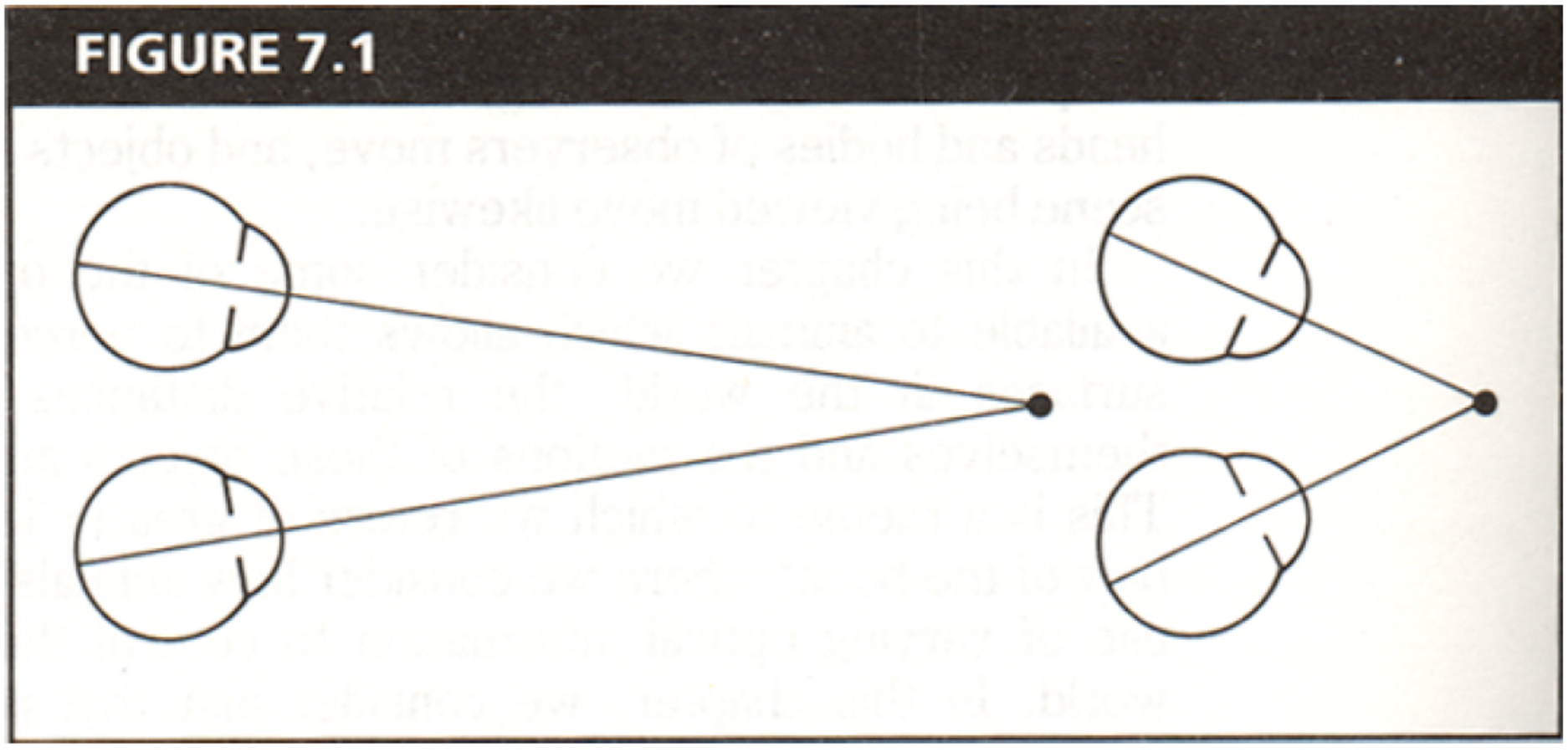
立体视觉





人眼立体视觉

- 当人眼注视空间中一点时：眼球转动使双眼成像在中央凹



From Bruce and Green, Visual Perception,
Physiology, Psychology and Ecology

人眼立体视觉

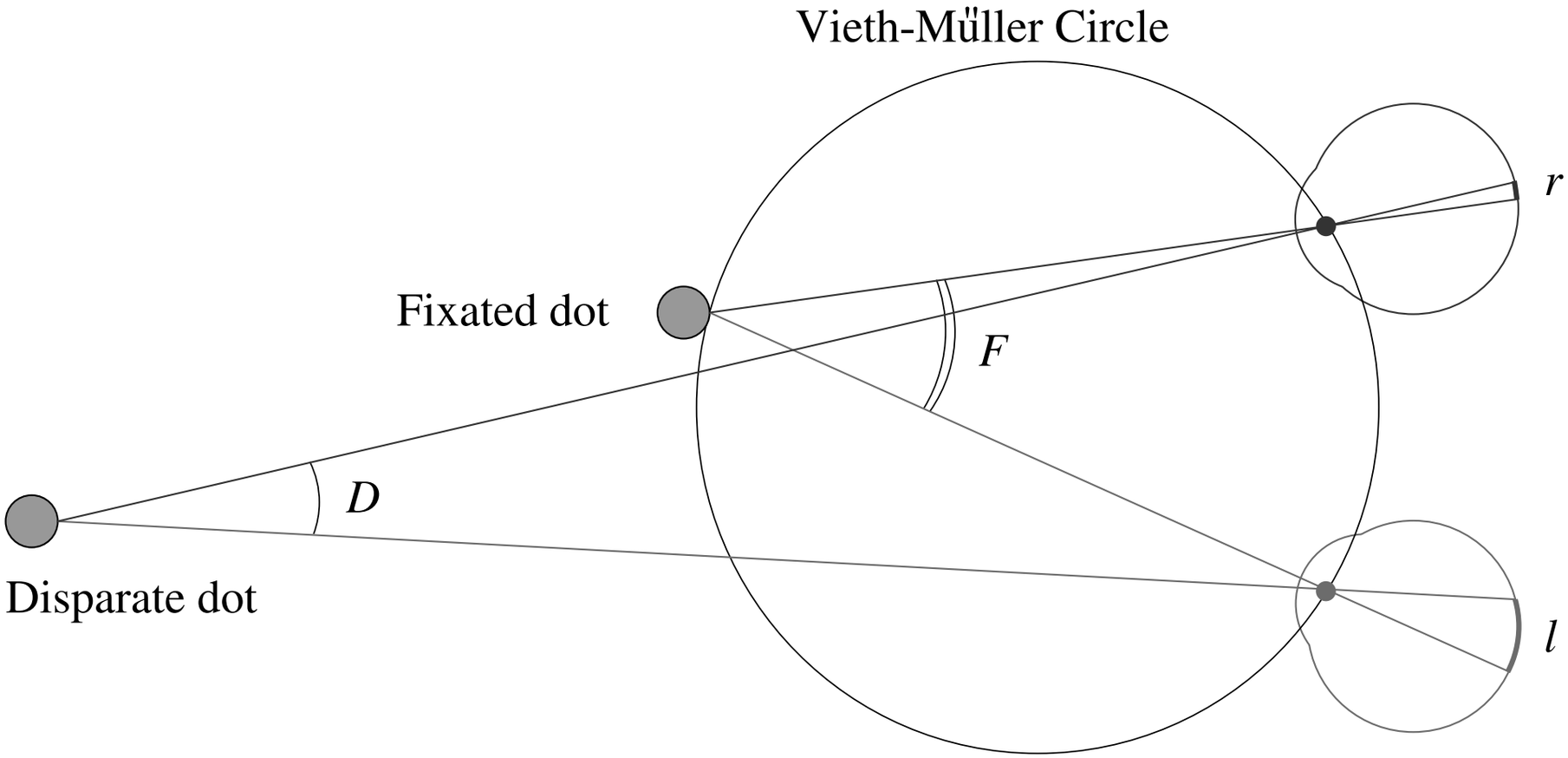
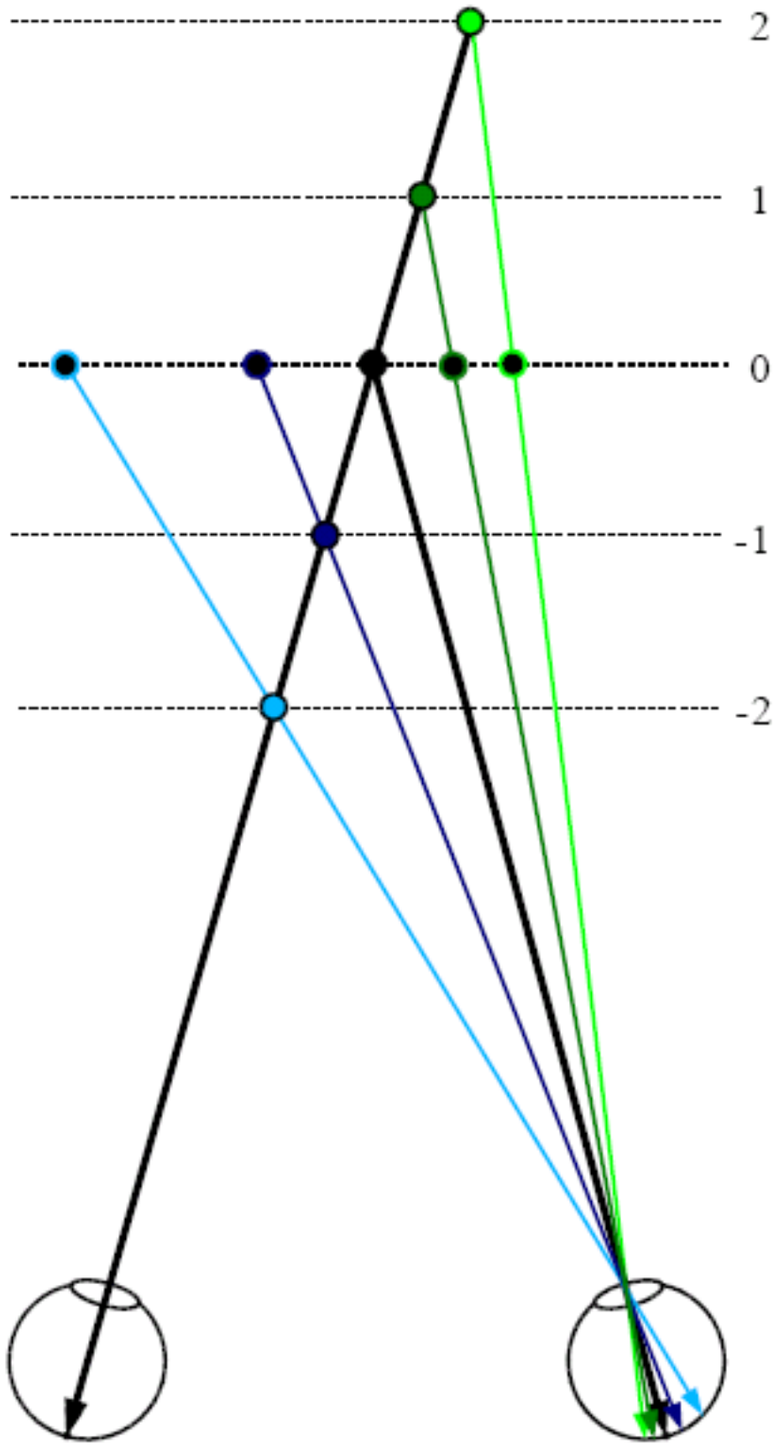
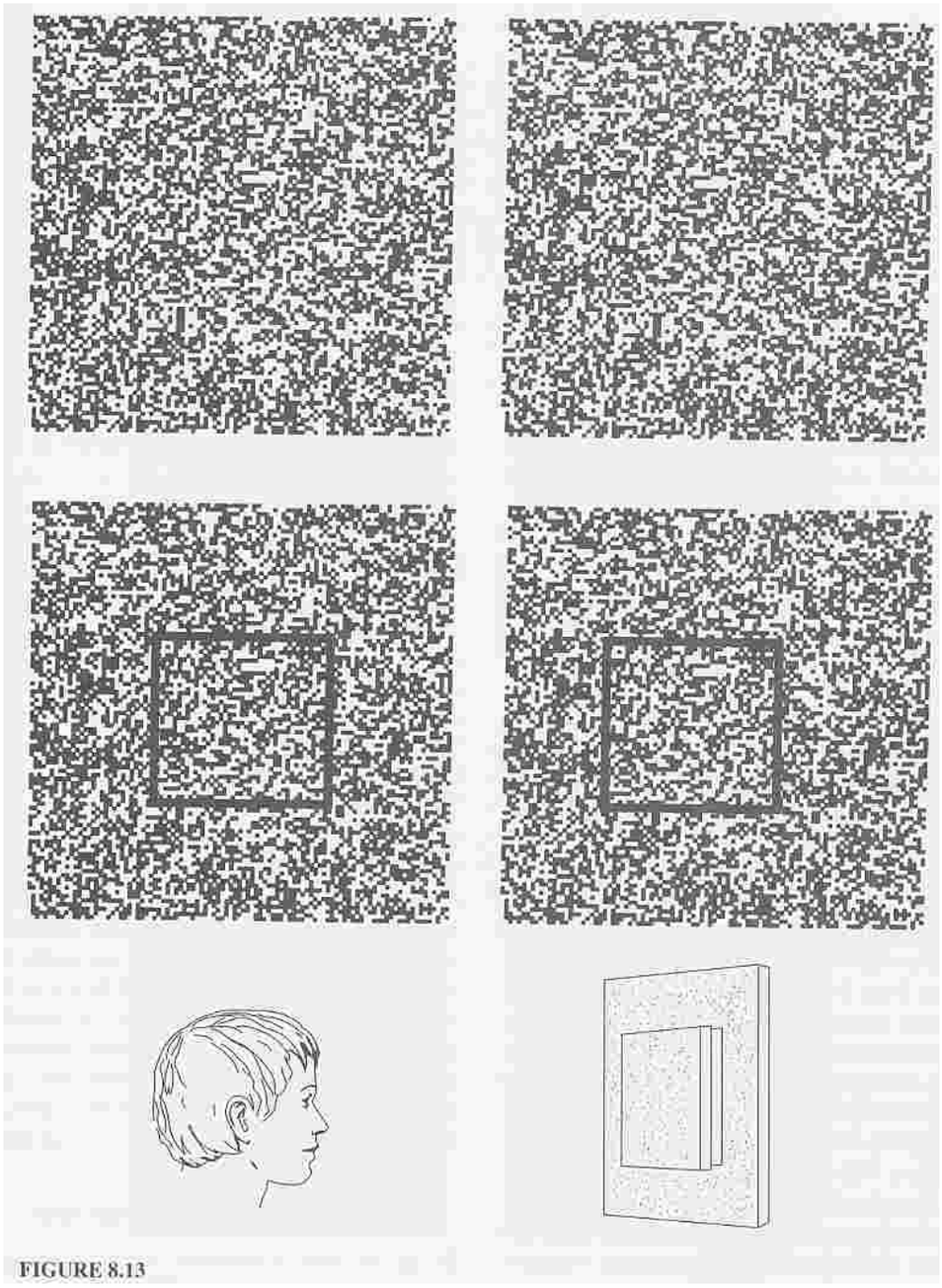
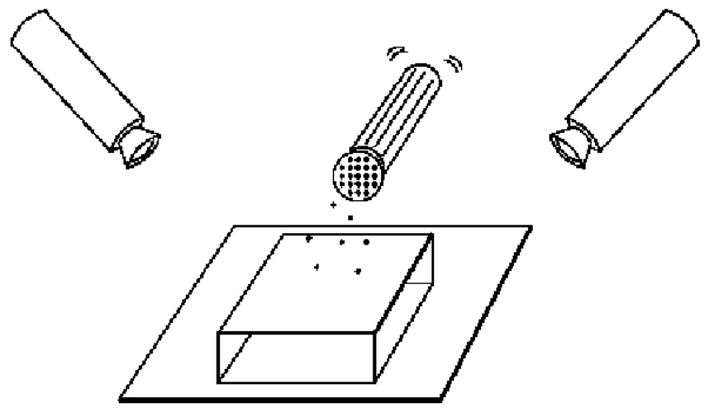


FIGURE 7.7: In this diagram, the close-by dot is fixated by the eyes, and it projects onto the center of their foveas with no disparity. The two images of the far dot deviate from this central position by different amounts, indicating a different depth.


2
1
0
-1
-2

随机点立体图（Bela Julesz）





“When viewed monocularly, the images appear completely random. But when viewed stereoscopically, the image pair gives the impression of a square markedly in front of (or behind) the surround.”

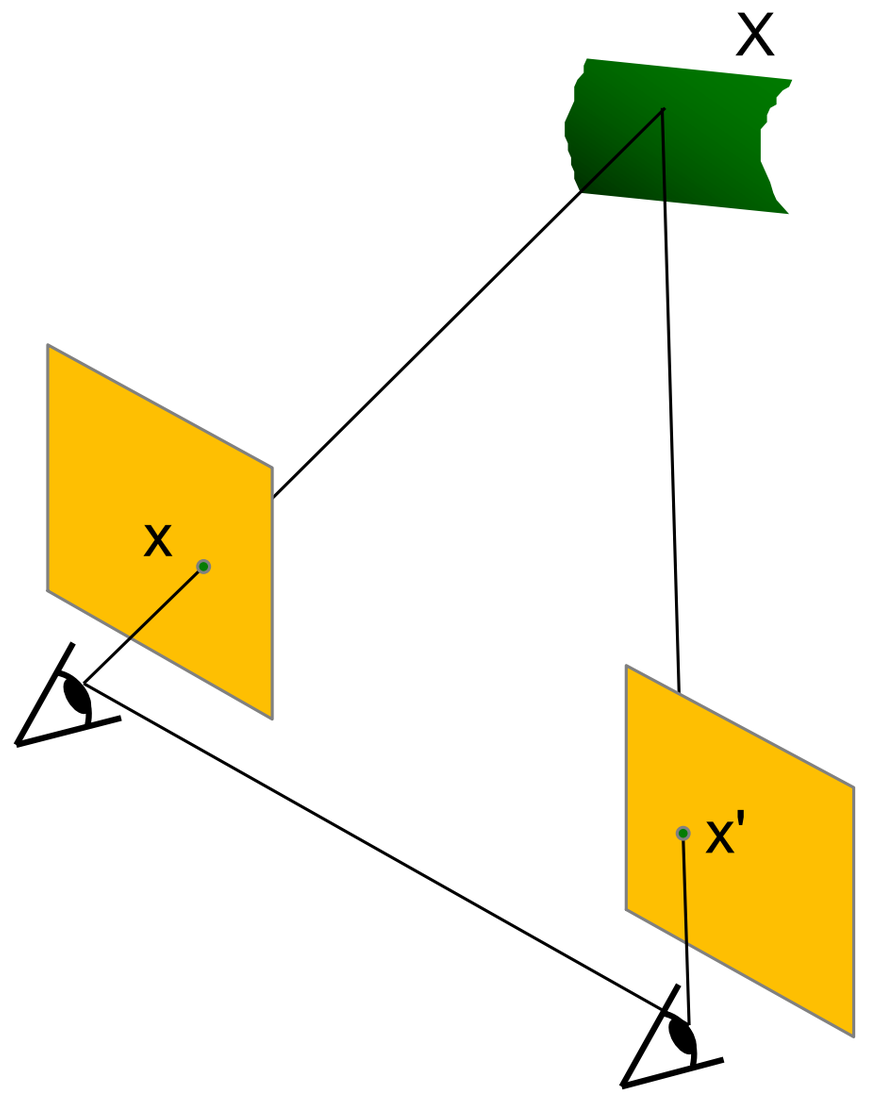
立体视觉深度估计

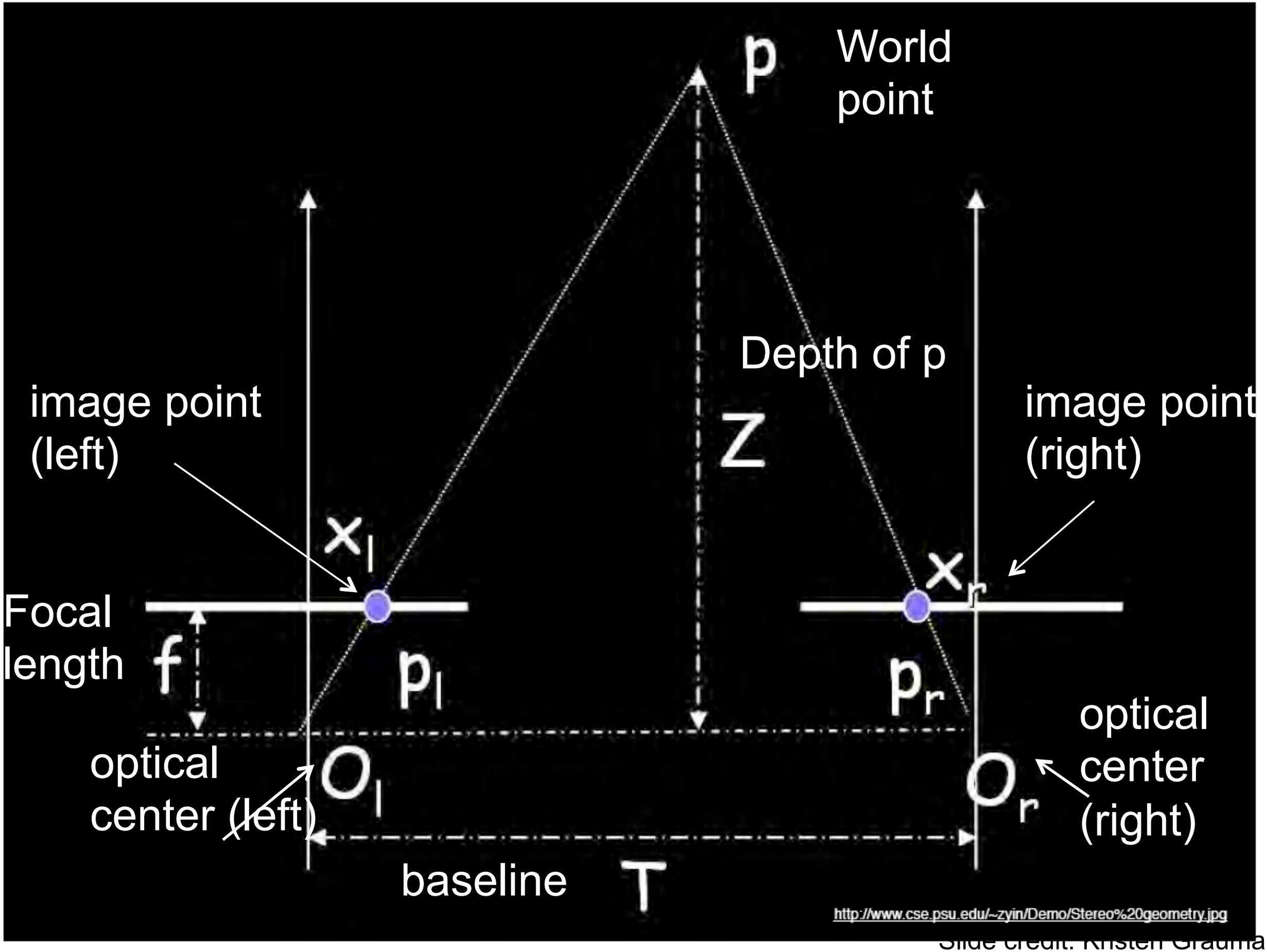
- **深度**: 摄像机中心到场景点的距离，或者三维场景点的Z坐标分量，是从三维场景的图像投影中理解三维场景本身的重要信息.
- **立体视觉（Stereo）**：通过三维点在两幅图像中的位置差异计算场景深度的视觉技术（***shape from disparities between two views.***）
- 需要考虑：
  - 像机的姿态（标定问题）
  - 图像点对应

FIGURE 7.1: **Left:** The Stanford cart sports a single camera moving in discrete increments along a straight line and providing multiple snapshots of outdoor scenes. **Center:** The INRIA mobile robot uses three cameras to map its environment. **Right:** The NYU mobile robot uses two stereo cameras, each capable of delivering an image pair. As shown by these examples, although two eyes are sufficient for stereo fusion, mobile robots are sometimes equipped with three (or more) cameras. The bulk of this chapter is concerned with binocular perception but stereo algorithms using multiple cameras are discussed in Section 7.6. *Photos courtesy of Hans Moravec, Olivier Faugeras, and Yann LeCun.*

一个简单立体视觉系统的几何结构

- 假设:
  - 光轴平行
  - 已知像机参数
- 目标:
  - 通过图像坐标点的对应恢复三维点的深度

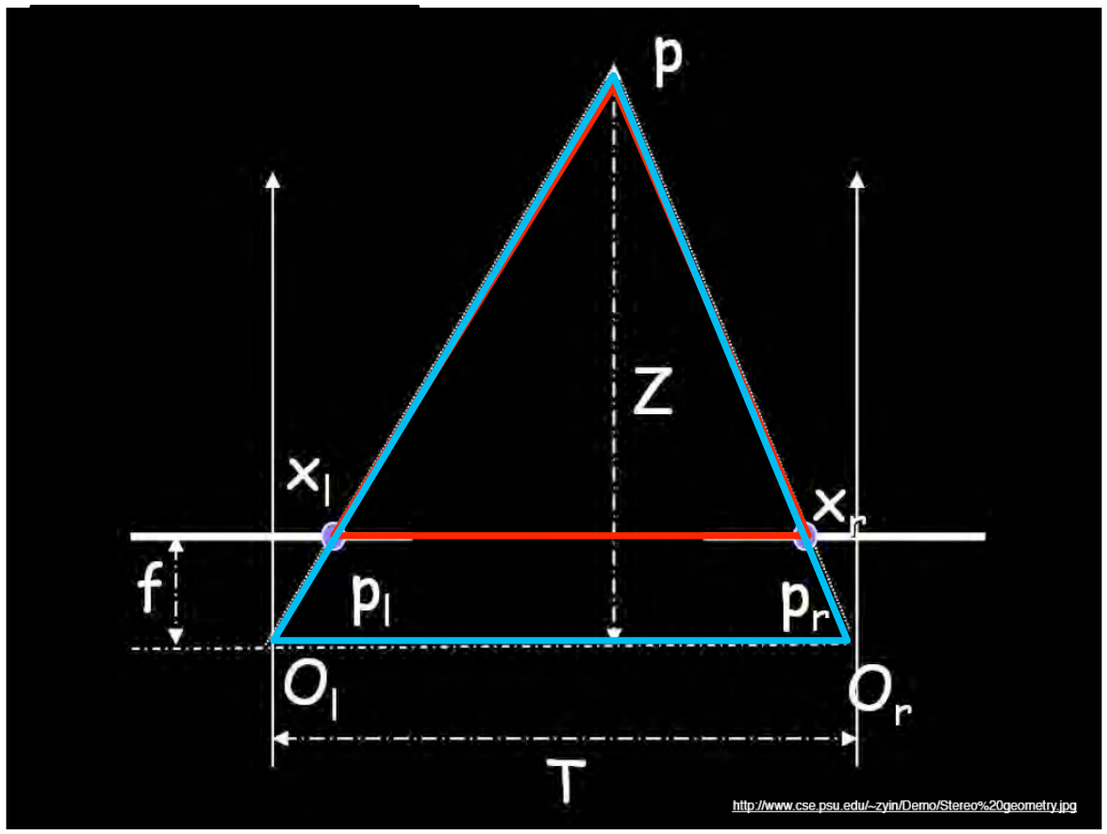




Slide credit: Kristen Grauman

一个简单立体视觉系统的几何结构

- 假设平行光轴、已知像机参数
- 相似三角形 (p_l, P, p_r) and (O_l, P, O_r):



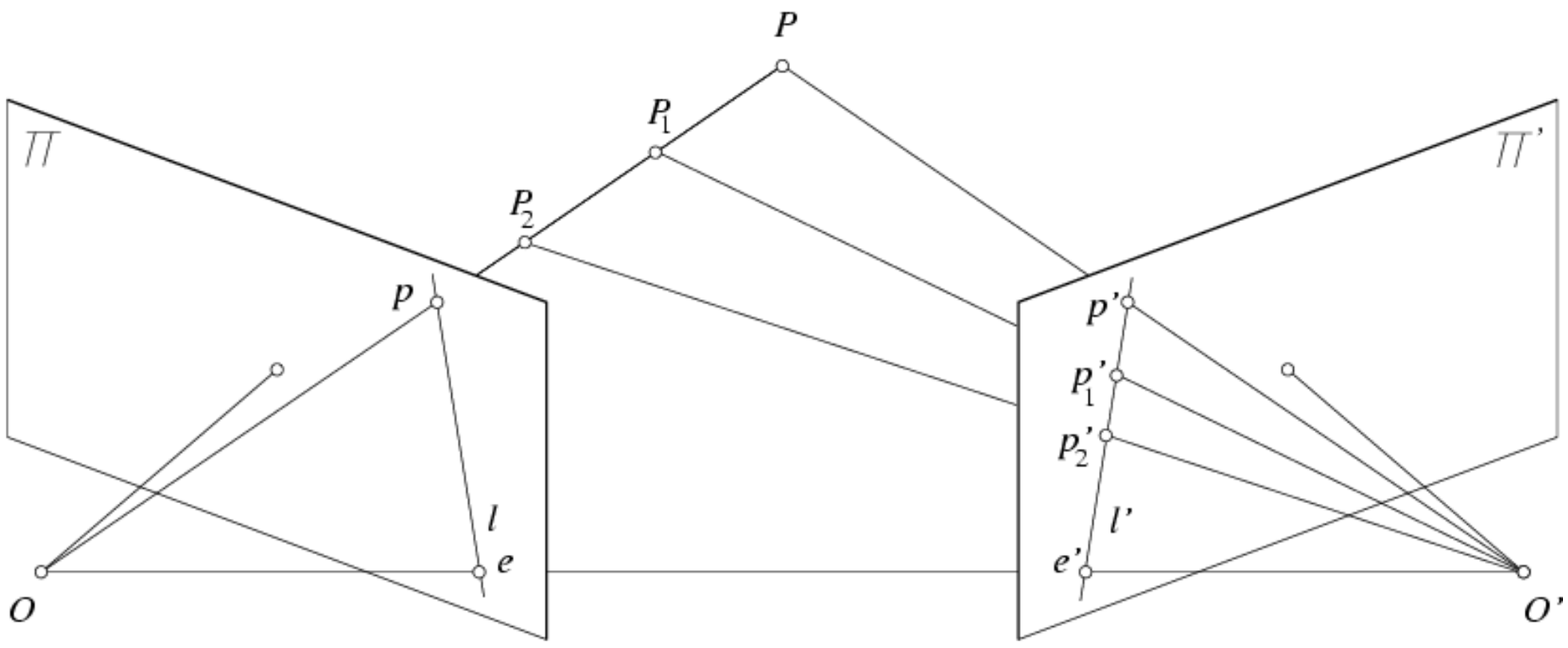
$$\frac{T + x_r - x_l}{Z - f} = \frac{T}{Z}$$

$$Z = f\frac{T}{x_l - x_r}$$

disparity

对极（Epipolar）约束

- 双目的几何结构，约束了左图像中的一点 p在右图像中的位置: 一定在由三维点P、两个光心一起确定的平面与右图像的交线上，反之亦然。



对极约束的作用

- 缩小点对应搜索空间

Image from Andrew Zisserman

如果光轴不平行，应该如何处理?

image I(x,y)

Disparity map D(x,y)

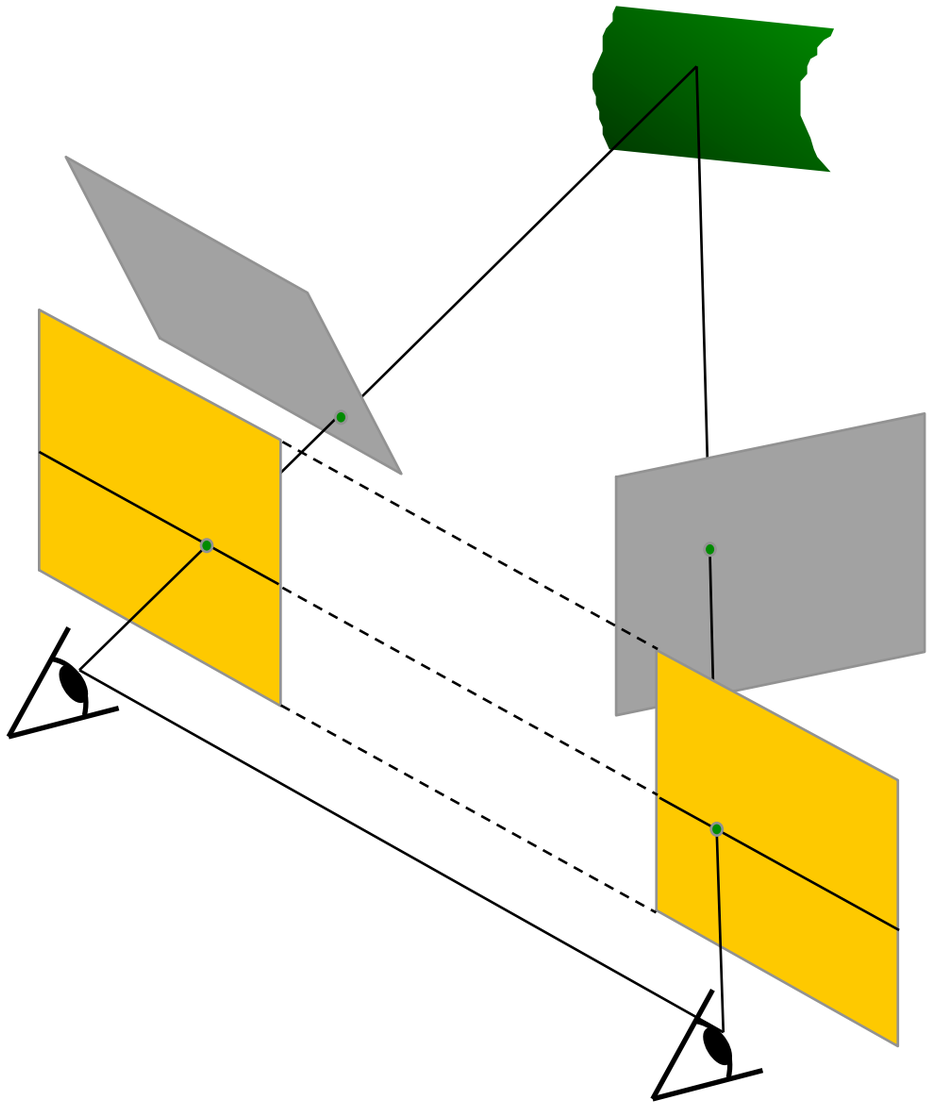
image I´(x´,y´)

$(x', y') = (x + D(x,y), y)$

Slide credit: Kristen Grauman

立体视觉图像矫正

- 实际情况中，如果图像的像素行就是对极线，会带来极大的便利



- 把图像平面重投影到一个平行于基线的平面
- 变换后，像素的运动为水平方向
- 需要确定两个单应变换 (3x3)

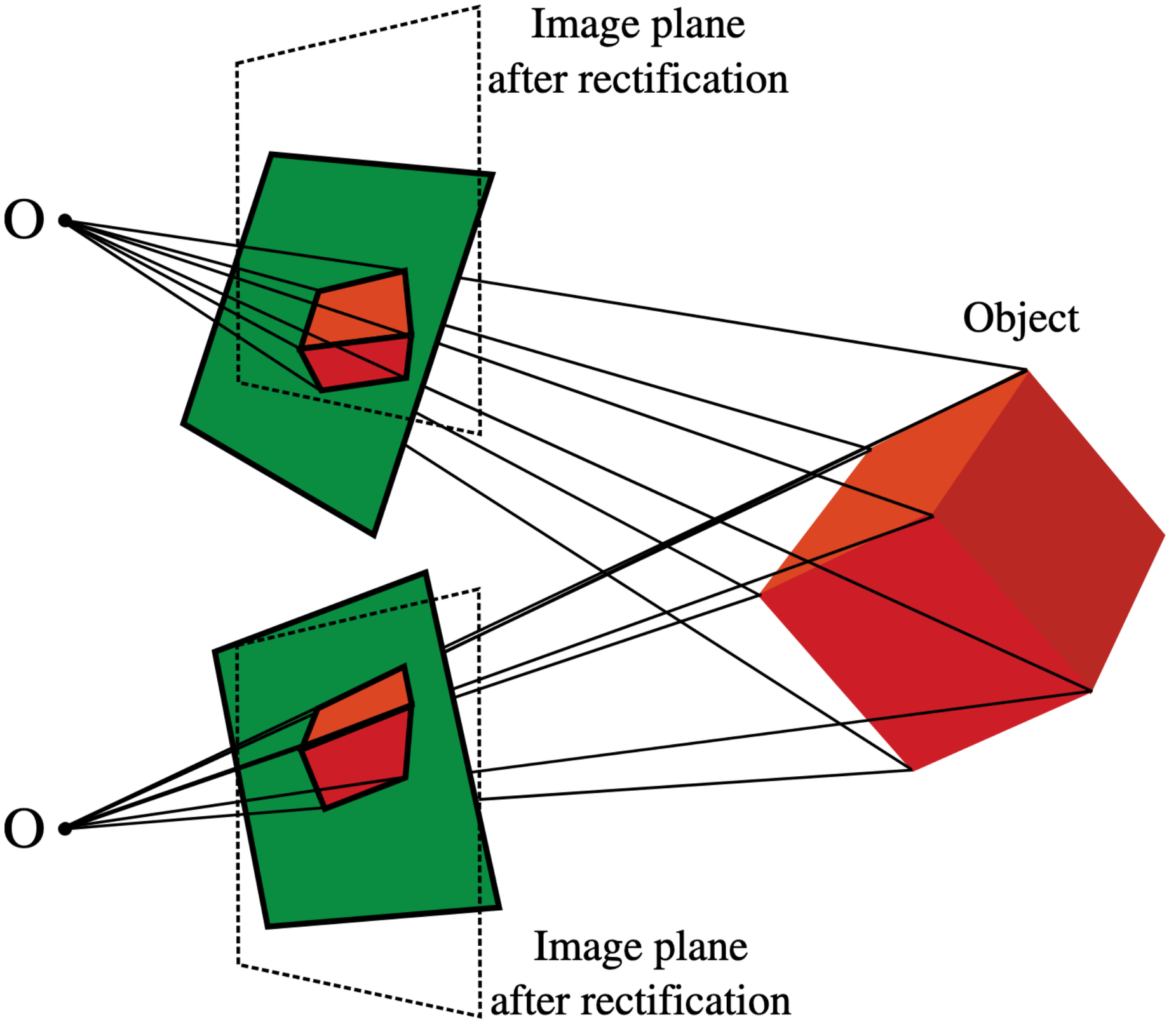
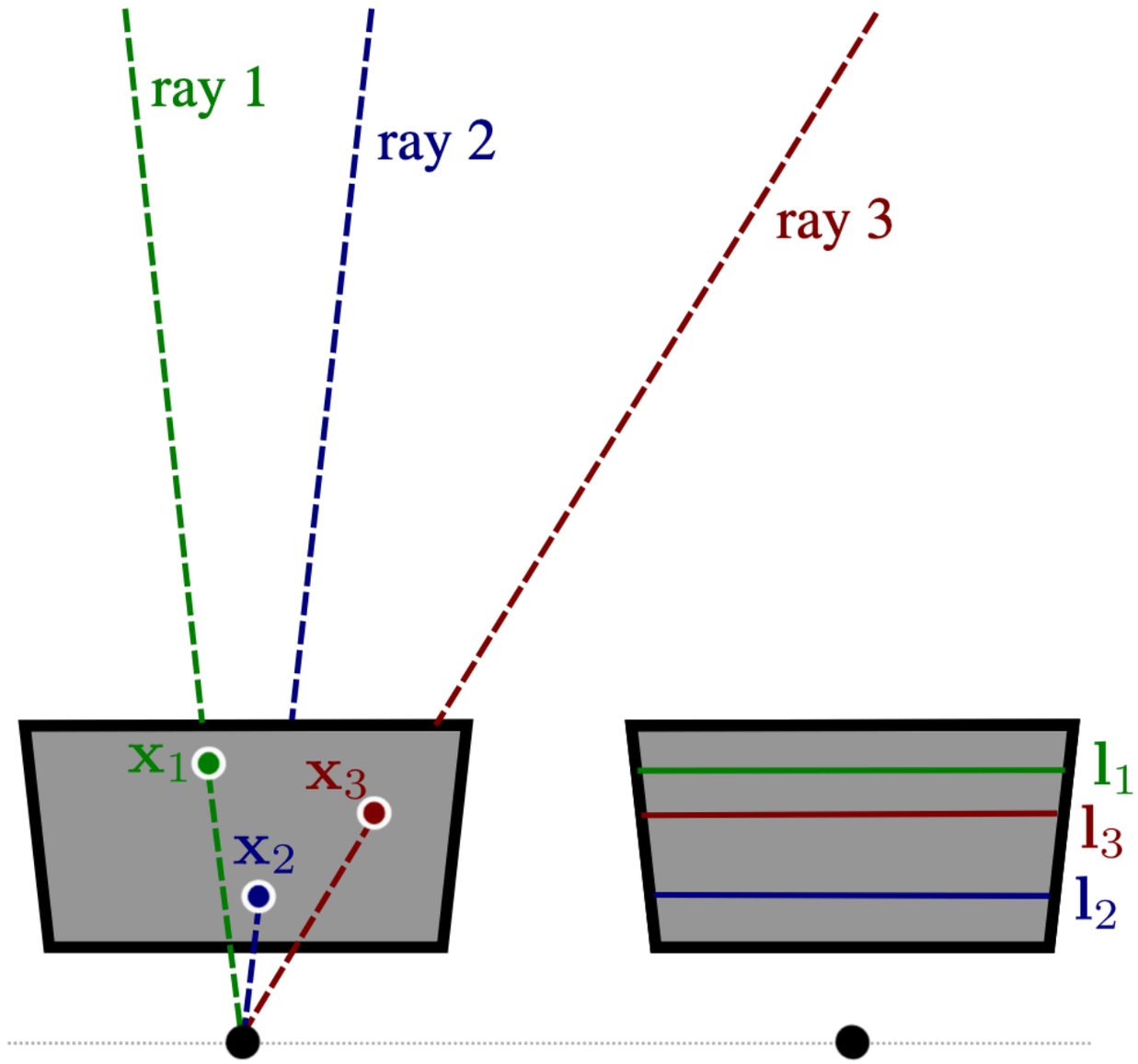


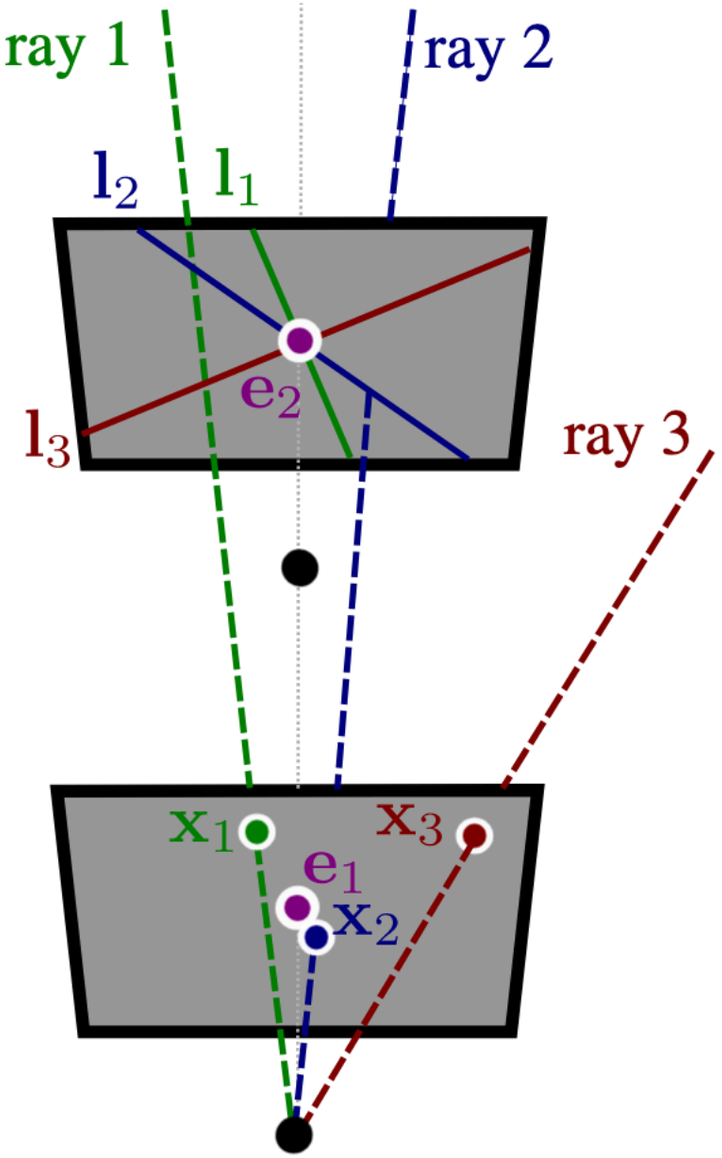
Image plane
after rectification
O
Object
O
Image plane
after rectification

a)



相机沿着垂直于光轴的方向运动

b)



相机沿着光轴的方向运动

立体视觉图像矫正

- **如何得到矫正需要的2个单应变换?**

1. **将图像-2的对极点映射到水平方向的无穷远点：**

$$\mathbf{H}' = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$$

2. **得到图像-1的相应匹配变换：H，使其最小化：**

$$\sum_i d(\mathbf{H}\mathbf{x}_i, \mathbf{H}'\mathbf{x}'_i)^2$$

立体视觉图像矫正

- **如何得到矫正需要的2个单应变换?**

1. **将图像-2的对极点映射到水平方向的无穷远点：** $\mathbf{H}' = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$

（1）将坐标系中心化到主点

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & -\delta_x \\ 0 & 1 & -\delta_y \\ 0 & 0 & 1 \end{bmatrix}$$

（2）关于坐标中心旋转图像，使对极点落到x轴

$$\mathbf{T}_2 = \begin{bmatrix} \cos[-\theta] & -\sin[-\theta] & 0 \\ \sin[-\theta] & \cos[-\theta] & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\theta = \text{atan2}[e_y, e_x]$$

（3）将对极点变换到无穷远处

$$\mathbf{T}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/e_x & 0 & 1 \end{bmatrix}$$

立体视觉图像矫正

- **如何得到矫正需要的2个单应变换?**

1. **将图像-2的对极点映射到水平方向的无穷远点：$\mathbf{H}' = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$**

2. **得到图像-1的相应匹配变换：H，使其最小化：**

$$\sum_i d(\mathrm{H}\mathbf{x}_i, \mathrm{H}'\mathbf{x}'_i)^2$$

（1）若两个矩阵$\mathbf{H}$和$\mathbf{H}'$为匹配变换，则有：

$$\mathrm{H} = (\mathrm{I} + \mathrm{H}'\mathbf{e}'\mathbf{a}^{\mathsf{T}})\mathrm{H}'\mathrm{M}$$

（2）$\mathbf{H}'$将对极点映射到无穷远点$(1,0,0)^T$，有

$$\mathrm{I} + \mathrm{H}'\mathbf{e}'\mathbf{a}^{\mathsf{T}} = \mathrm{I} + (1,0,0)^{\mathsf{T}}\mathbf{a}^{\mathsf{T}}$$

$$\sum_i d(\mathrm{H_A}\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i)^2 \qquad \mathrm{H} = \mathrm{H_A}\mathrm{H_0} \qquad \hat{\mathbf{x}}'_i = \mathrm{H}'\mathbf{x}'_i$$

$$\mathrm{H_0} = \mathrm{H}'\mathrm{M} \qquad \hat{\mathbf{x}}_i = \mathrm{H_0}\mathbf{x}_i$$

（3）线性最小二乘法求解$\mathbf{a} = (a,b,c)^T$

$$\mathrm{H_A} = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

立体视觉图像矫正

- **算法：**

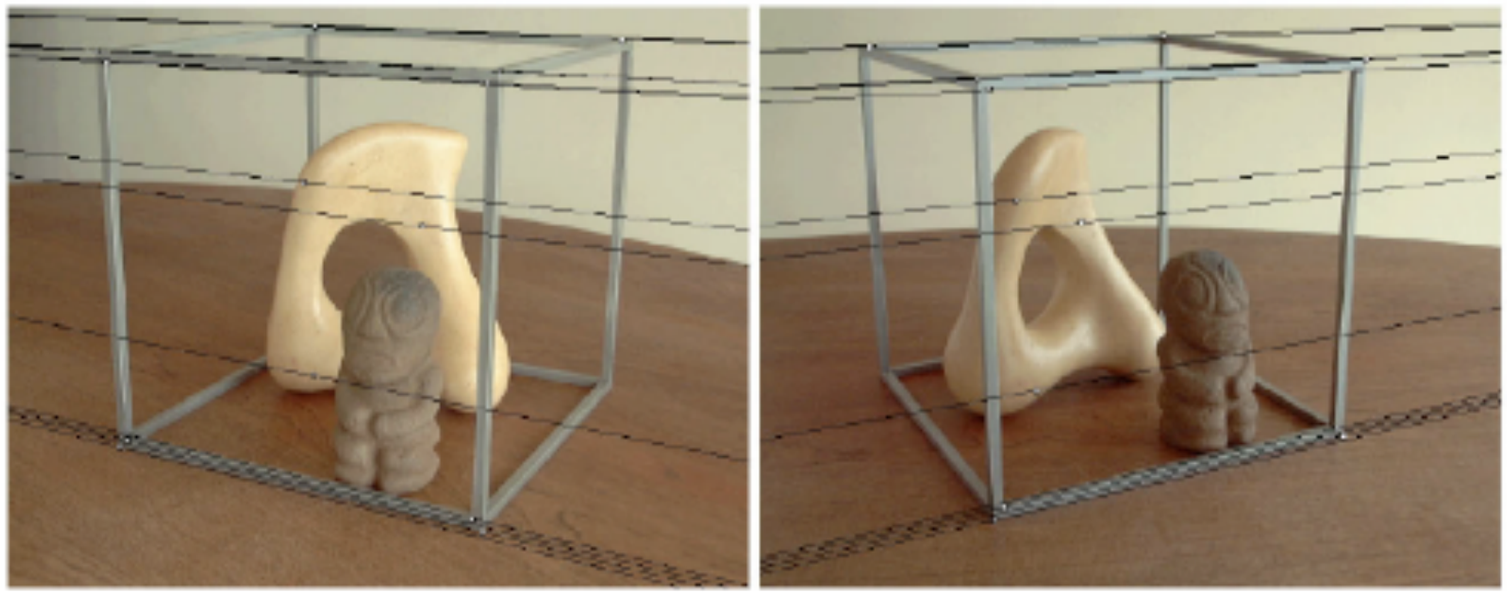
① **得到2张视图的对应特征点**

② **求解基本矩阵F和2张视图的对极点e, e'**

③ **选择一个射影变换$\mathbf{H}'$，将e'映射到无穷点$(1,0,0)^T$**

④ **最小化视差：** $\sum_i d(\mathrm{H}\mathbf{x}_i, \mathrm{H}'\mathbf{x}'_i)$

⑤ **用2个变换矩阵重采样2张图像**



立体视觉匹配

- 对应问题

- 多个匹配均满足对极约束，哪个是对的?

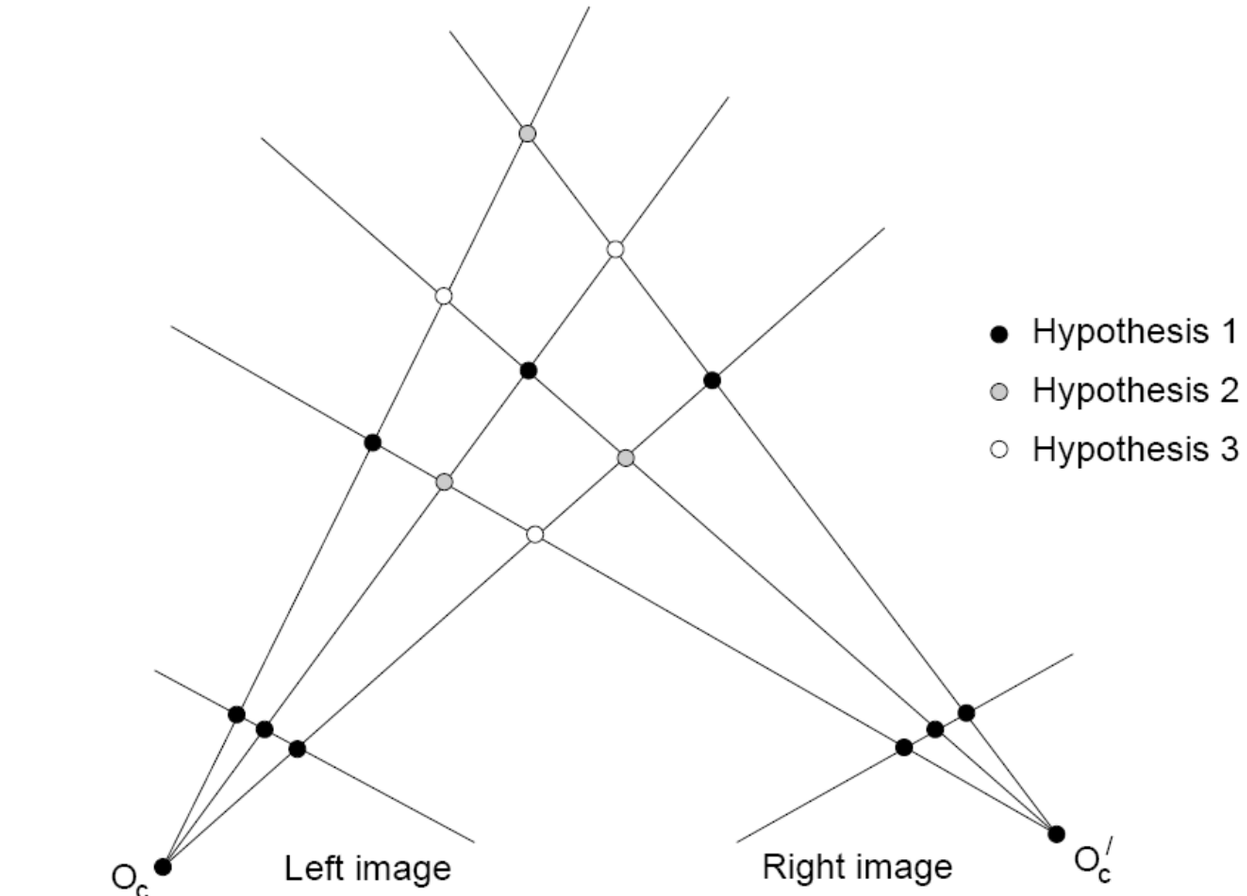
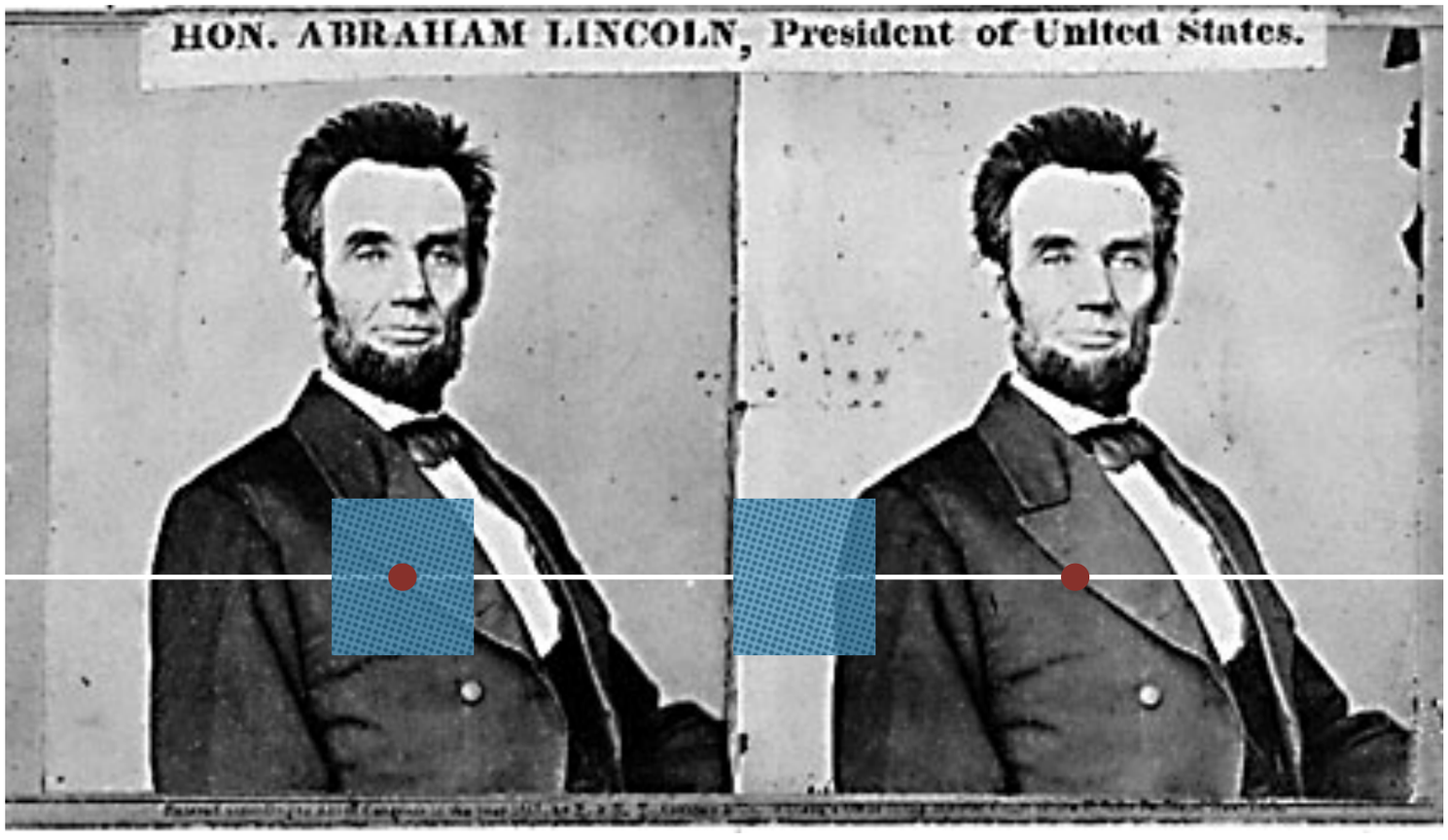


Figure from Gee & Cipolla 1999

立体视觉匹配

- 除了对极约束这个硬约束之外，还有“软约束”
  - 相似度
  - 唯一性
  - 顺序
  - 视差梯度
- 假设
  - 大部分景物点在两个视图中都可见
  - 匹配点图像区域在表观上是相似的

局部法：密集对应搜索



- 对于每一条对极线
  - 对于左图像的每一个像素（窗口）：
    - 在右图的相同对极线上进行逐项素（窗口）比较
    - 选出最小匹配代价的位置 (e.g., SSD, normalized correlation)

归一化相关

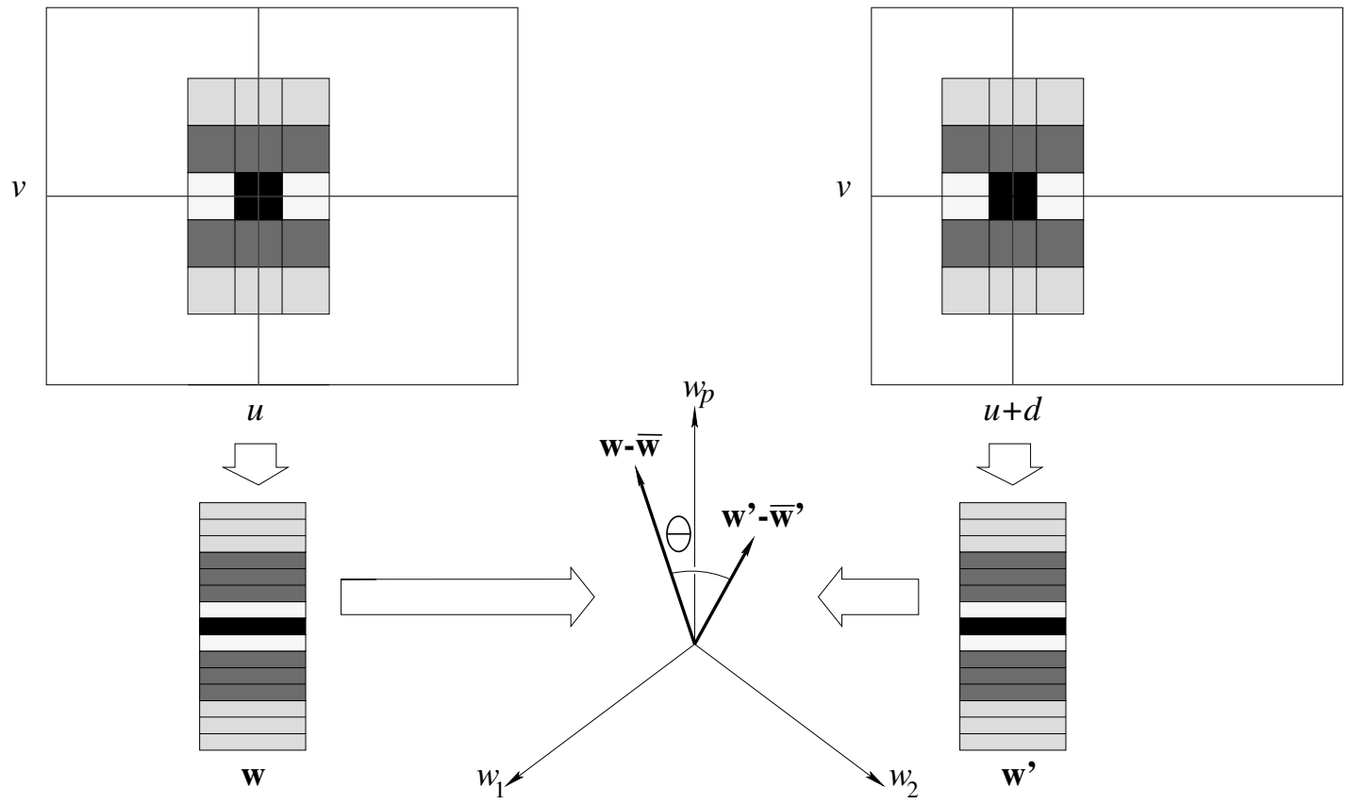
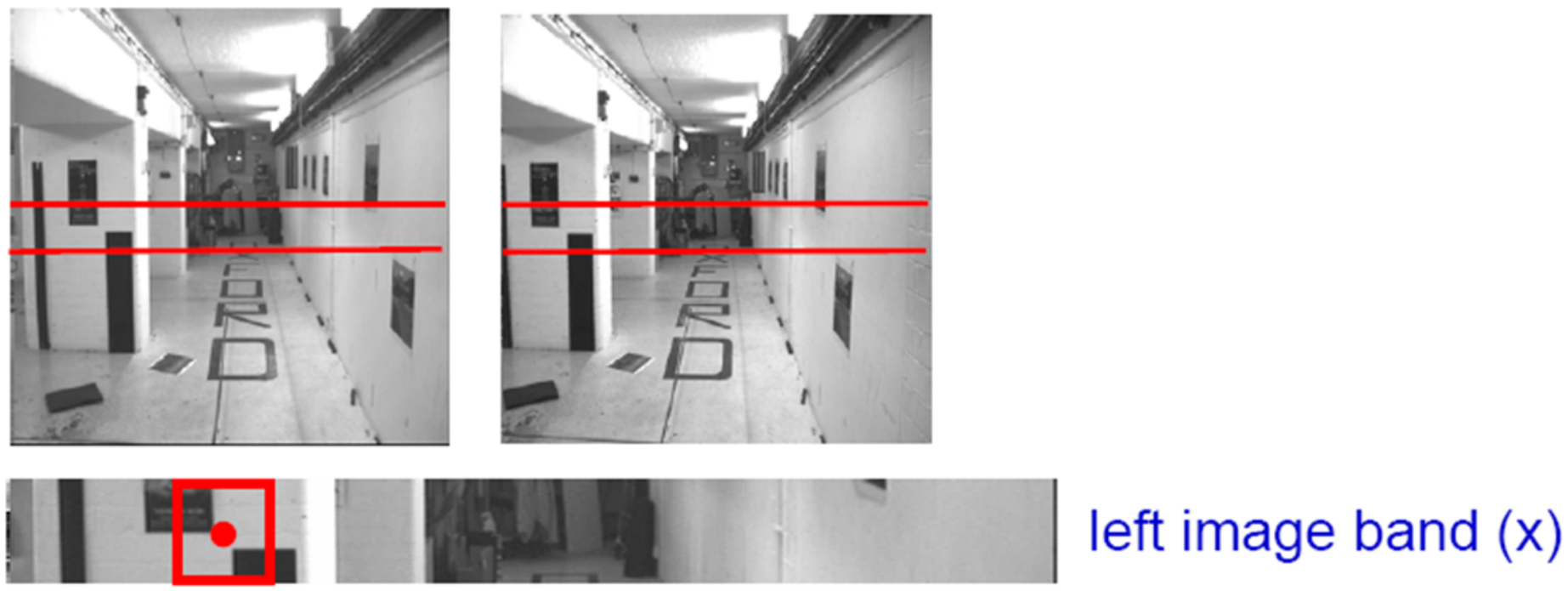


FIGURE 7.9: Correlation of two 3×5 windows along corresponding epipolar lines. The second window position is separated from the first one by an offset d. The two windows are encoded by vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ in $\mathbb{R}^{15}$, and the correlation function measures the cosine of the angle θ between the vectors $\boldsymbol{w} - \bar{\boldsymbol{w}}$ and $\boldsymbol{w}' - \bar{\boldsymbol{w}}'$ obtained by subtracting from the components of $\boldsymbol{w}$ and $\boldsymbol{w}'$ the average intensity in the corresponding windows.

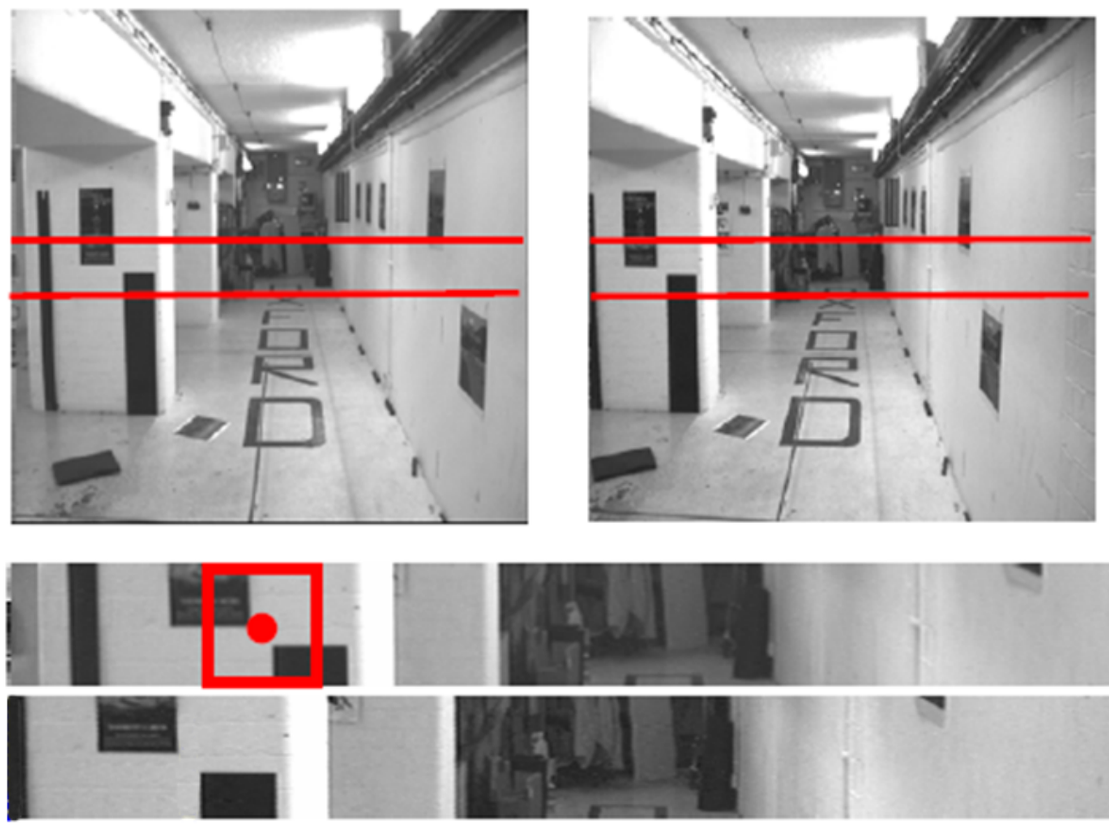
$$C(d) = \frac{1}{||\boldsymbol{w} - \bar{\boldsymbol{w}}||} \frac{1}{||\boldsymbol{w}' - \bar{\boldsymbol{w}}'||} [(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}')],$$

基于相关的窗口匹配



Source: Andrew Zisserman

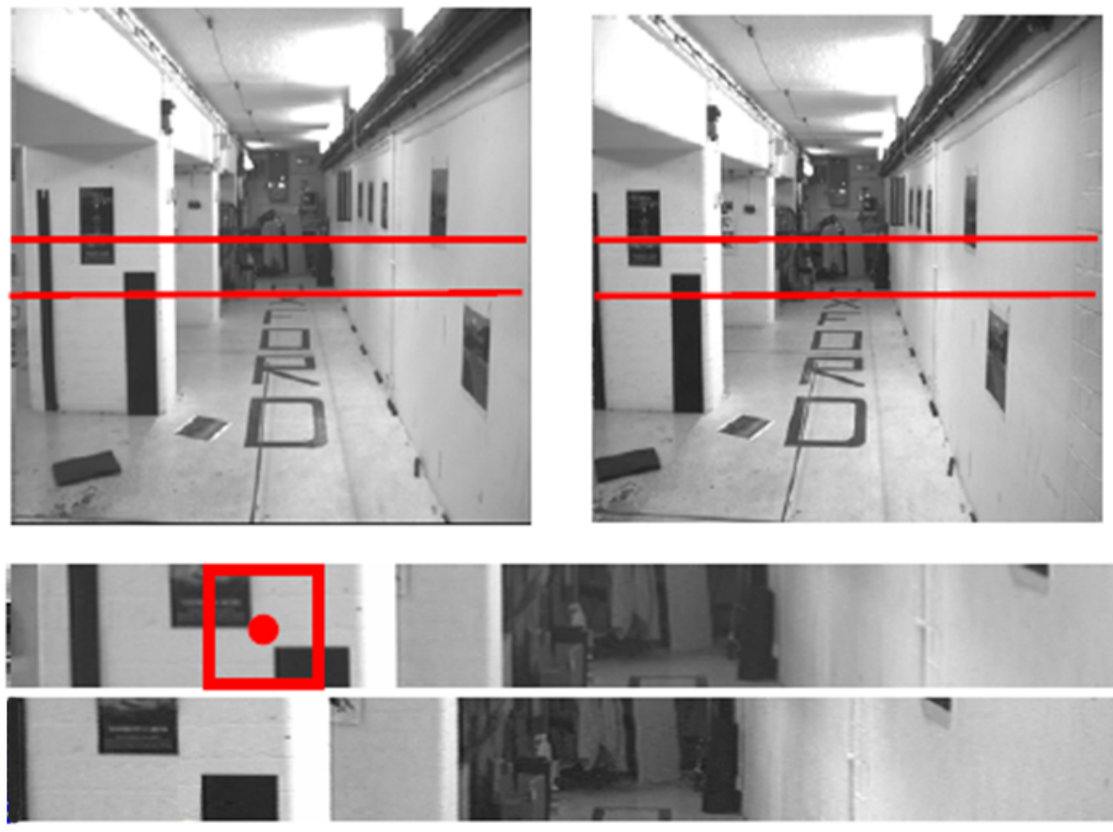
基于相关的窗口匹配



left image band (x)

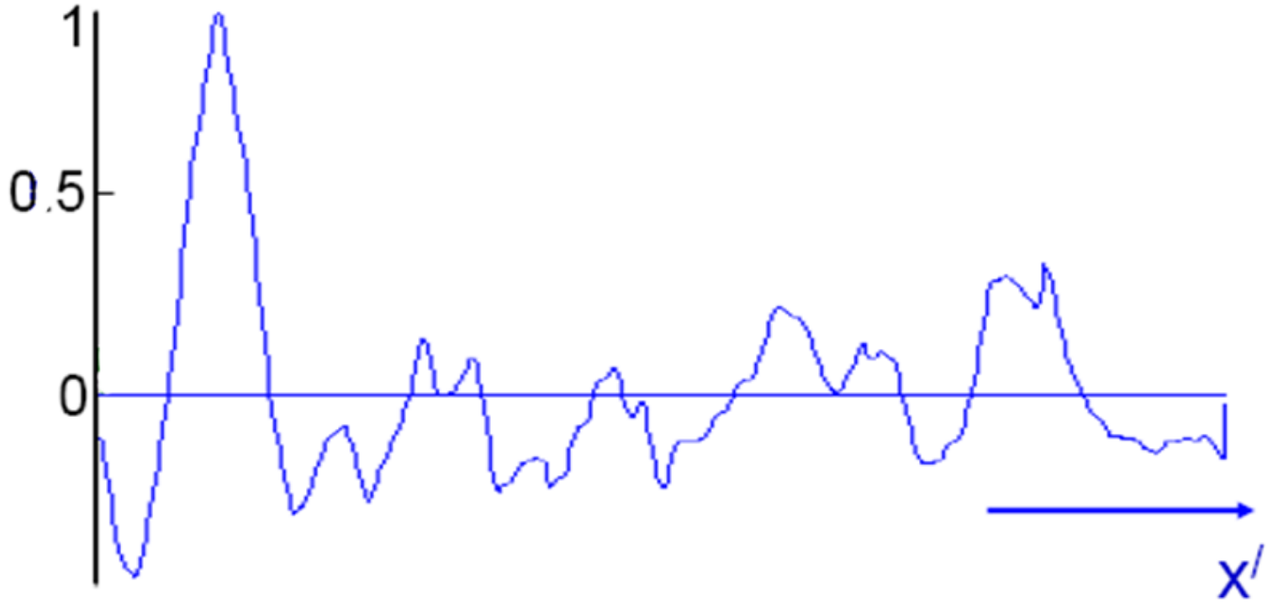
right image band ($x^{/}$)

基于相关的窗口匹配



left image band (x)

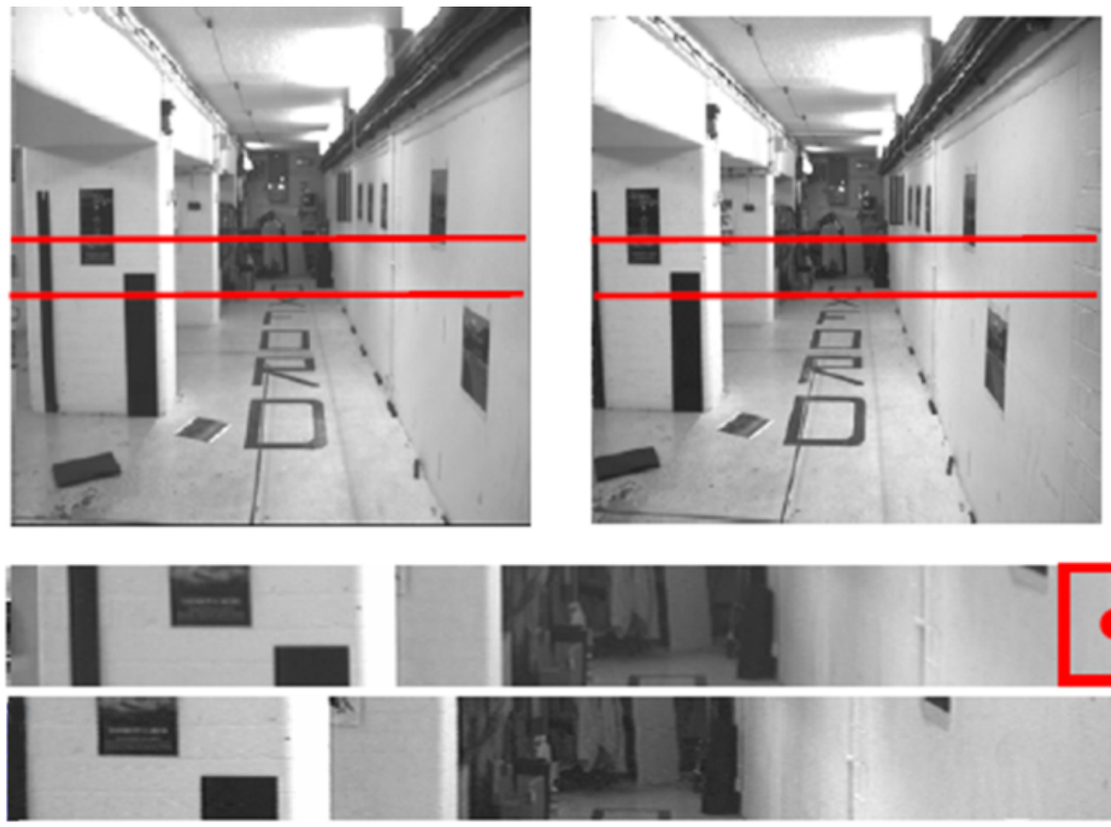
right image band ($x^{/}$)

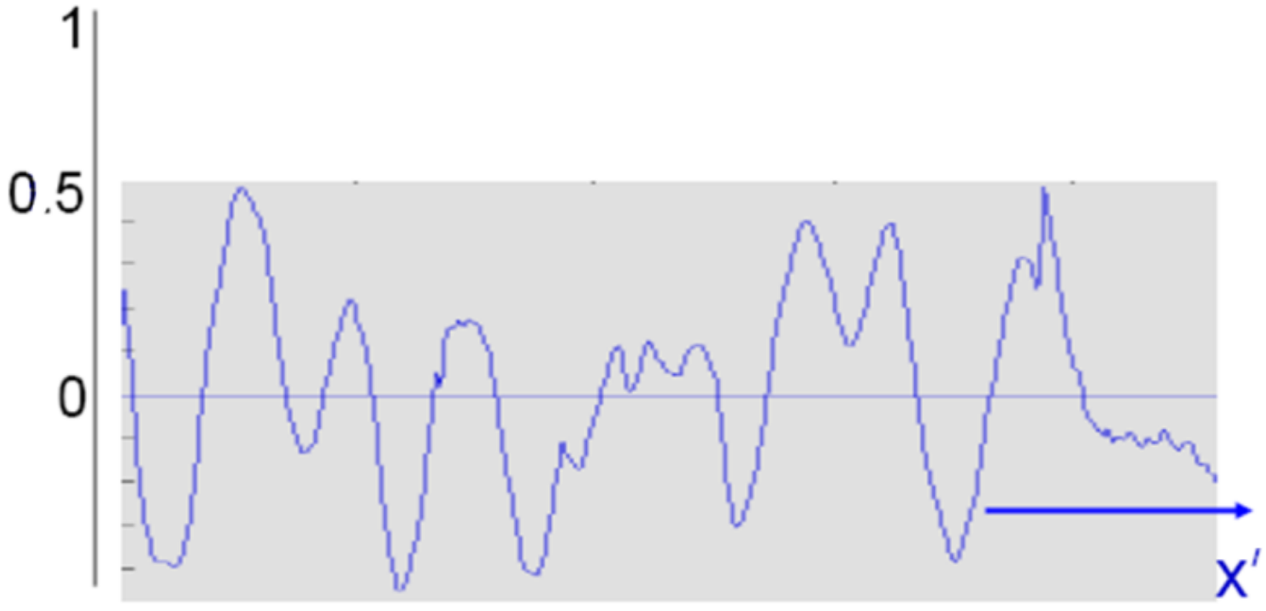


cross correlation

disparity = $x^{/}$ - x

基于相关的窗口匹配

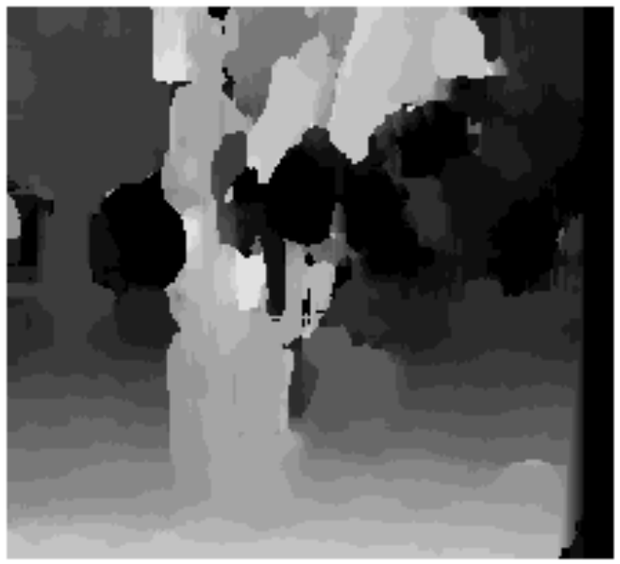




Textureless regions are non-distinct; high ambiguity for matches.

窗口大小的影响

W = 3

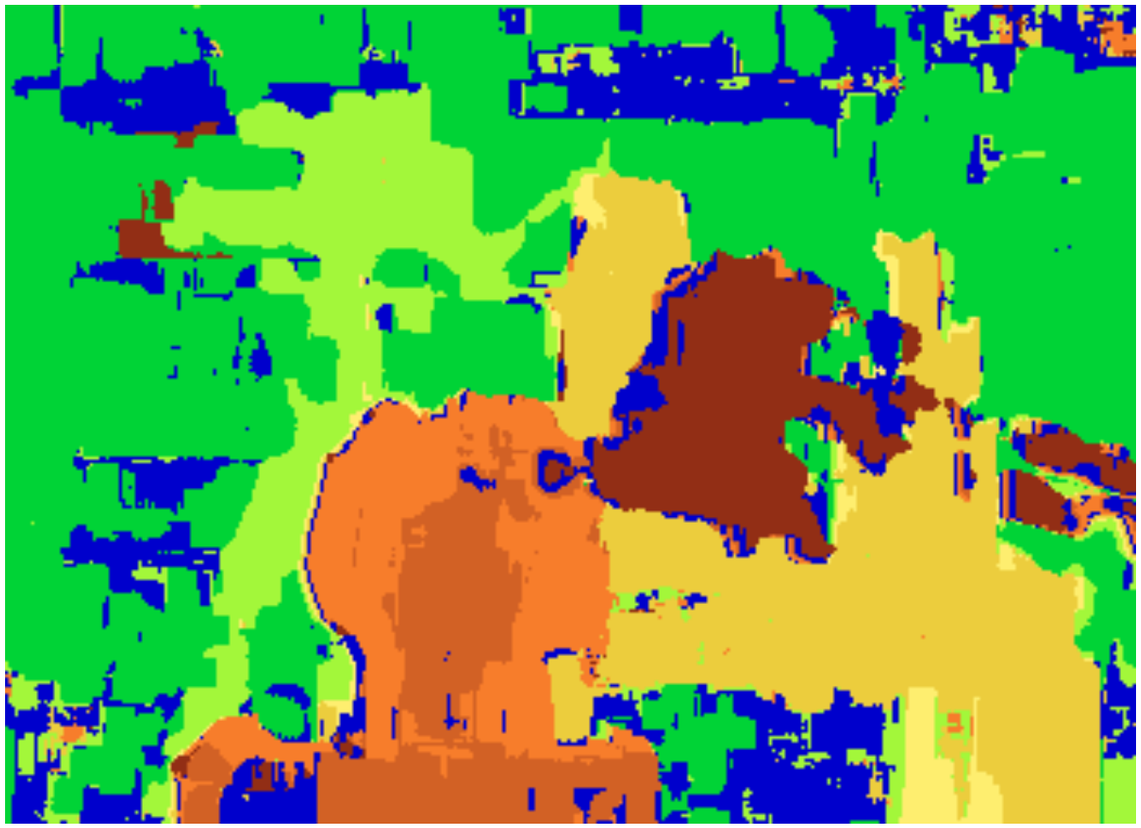


W = 20

窗口大：细节少；噪声少
窗口小：细节多；噪声多
改进：自适应窗口大小

Tsukuba （筑波）立体视觉测试场景

窗口搜索实验结果



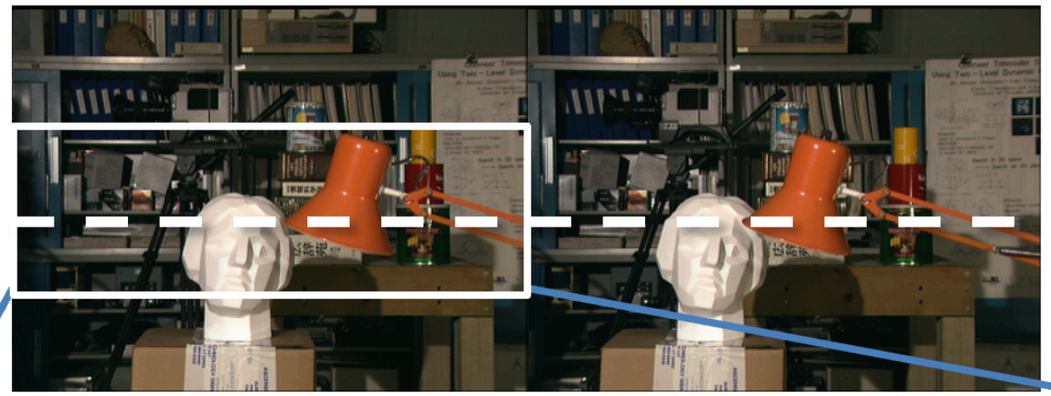
Window-based matching
(best window size)

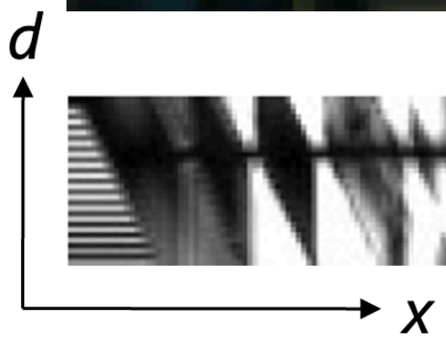
‘Ground truth’

立体视觉匹配：能量最小化

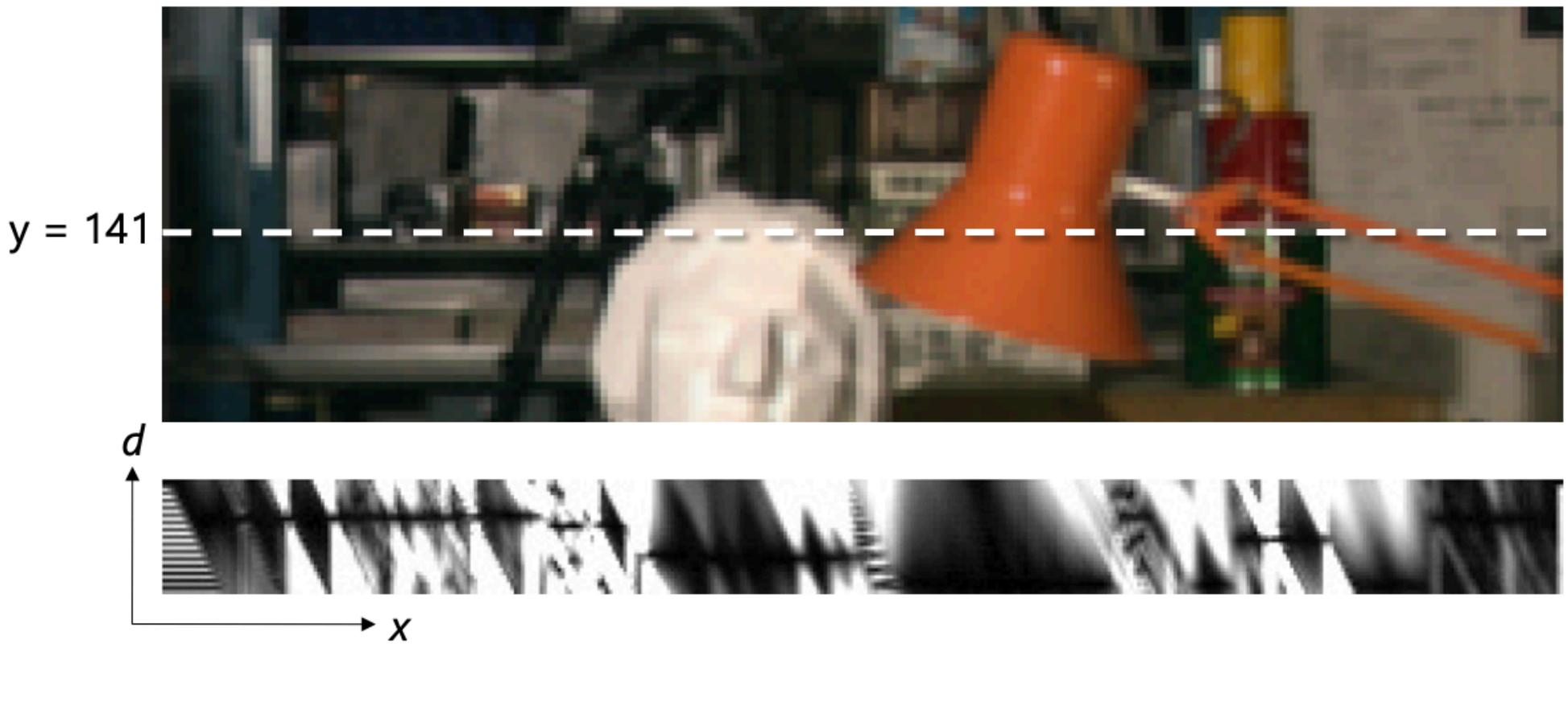
- 影响匹配的关键因素
  - 单像素匹配质量
  - 平滑性：相邻像素的视差接近
- 能量最小化问题：找到视差d，使其可最小化能量函数$E(d)$
  - 单像素窗口匹配

$$E(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$





C(*x*, *y*, *d*); the *disparity space image* (DSI)



$$d(x,y) = \arg\min_{d'} \; C(x,y,d')$$

单像素（窗口）匹配：在DSI的每一列独立选择最小

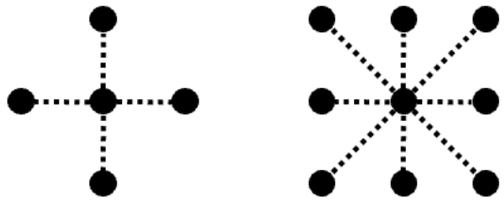
立体视觉匹配：能量最小化

- 能量最小化问题：找到视差d，使其可最小化能量函数$E(d)$
  - 单像素窗口匹配：$E(d)=E_d(d)$
  - 更好的目标函数：$E(d)=E_d(d)+\lambda E_s(d)$

匹配代价　　平滑代价

$$E_d(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

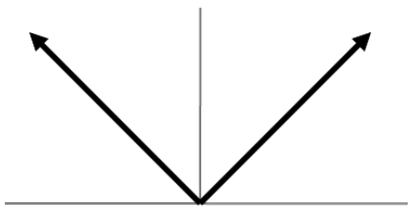
$$E_s(d) = \sum_{(p,q)\in \mathcal{E}} V(d_p, d_q)$$



4-connected neighborhood　　8-connected neighborhood

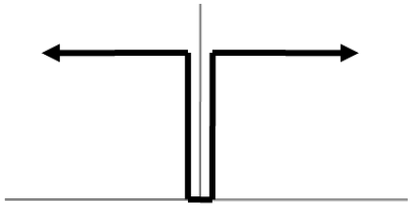
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



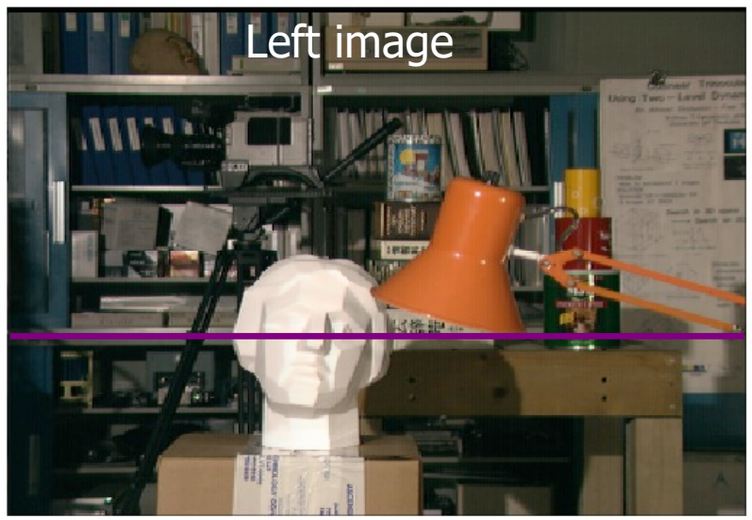
$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

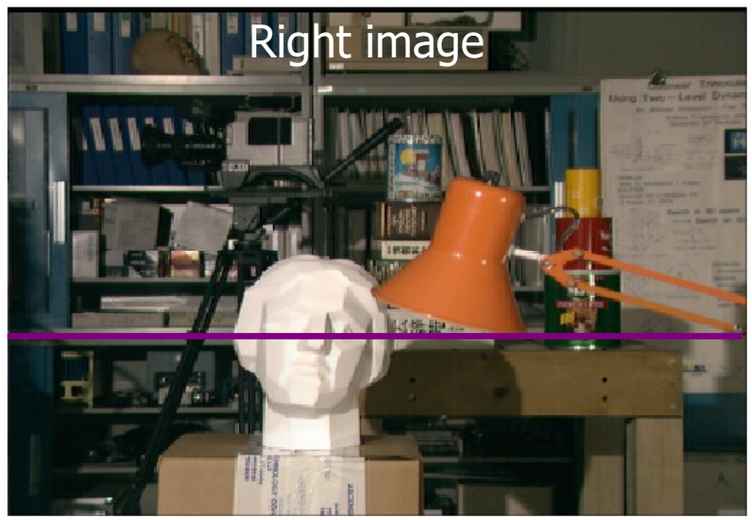
"Potts model"

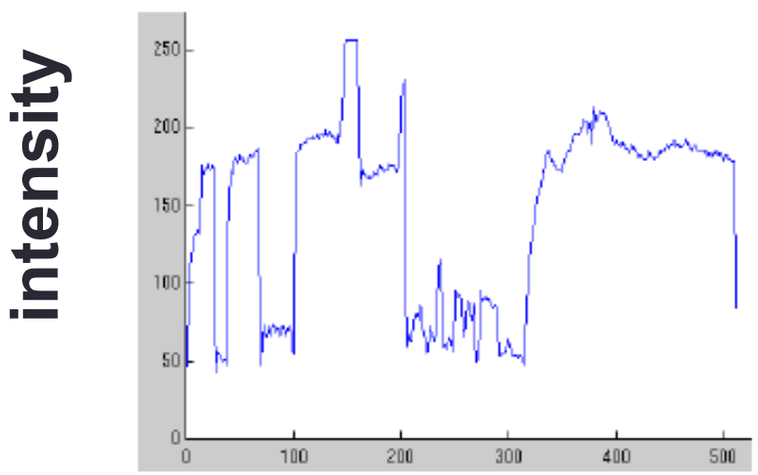


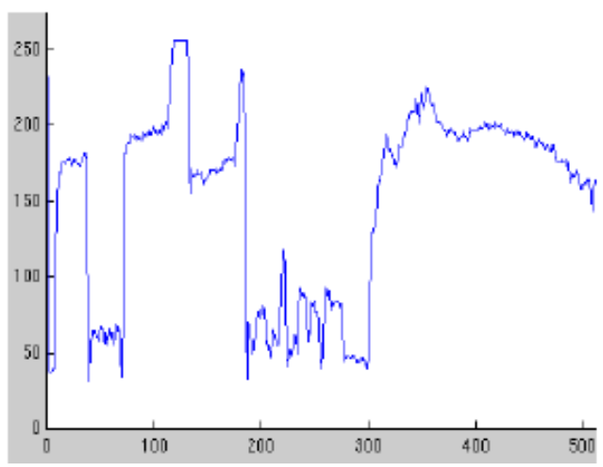
能量最小化：动态规划

- 扫描线立体视觉法
  - 连贯一致地在整条扫描线上进行匹配
  - 不同扫描线彼此独立



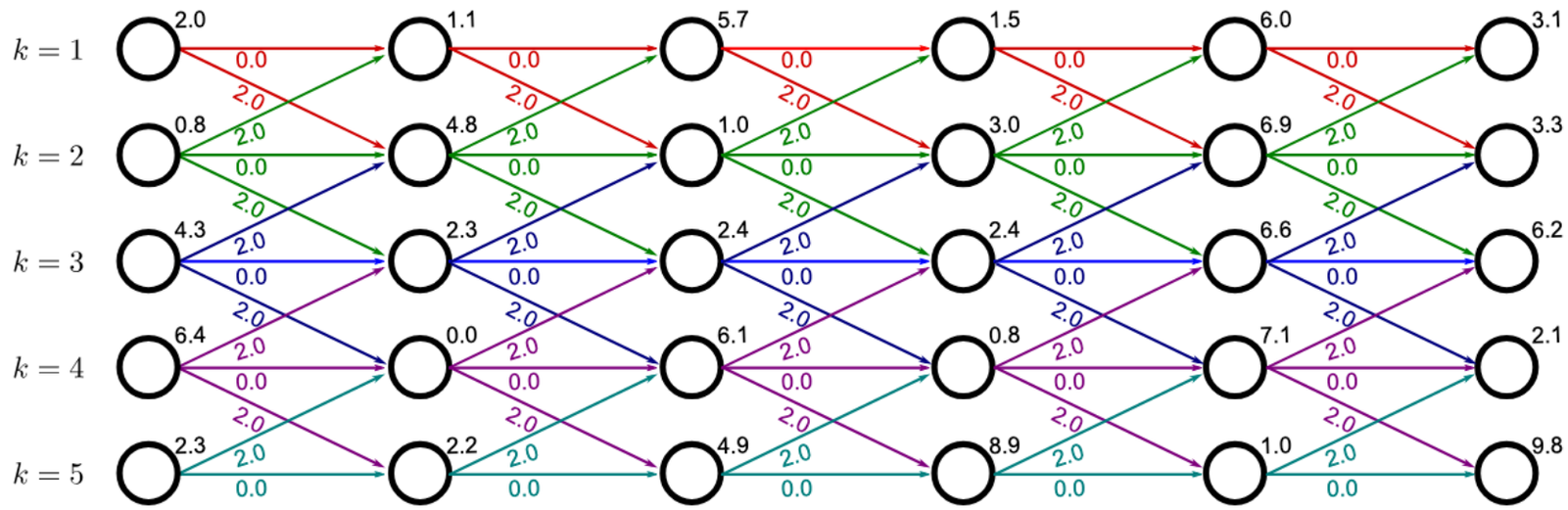




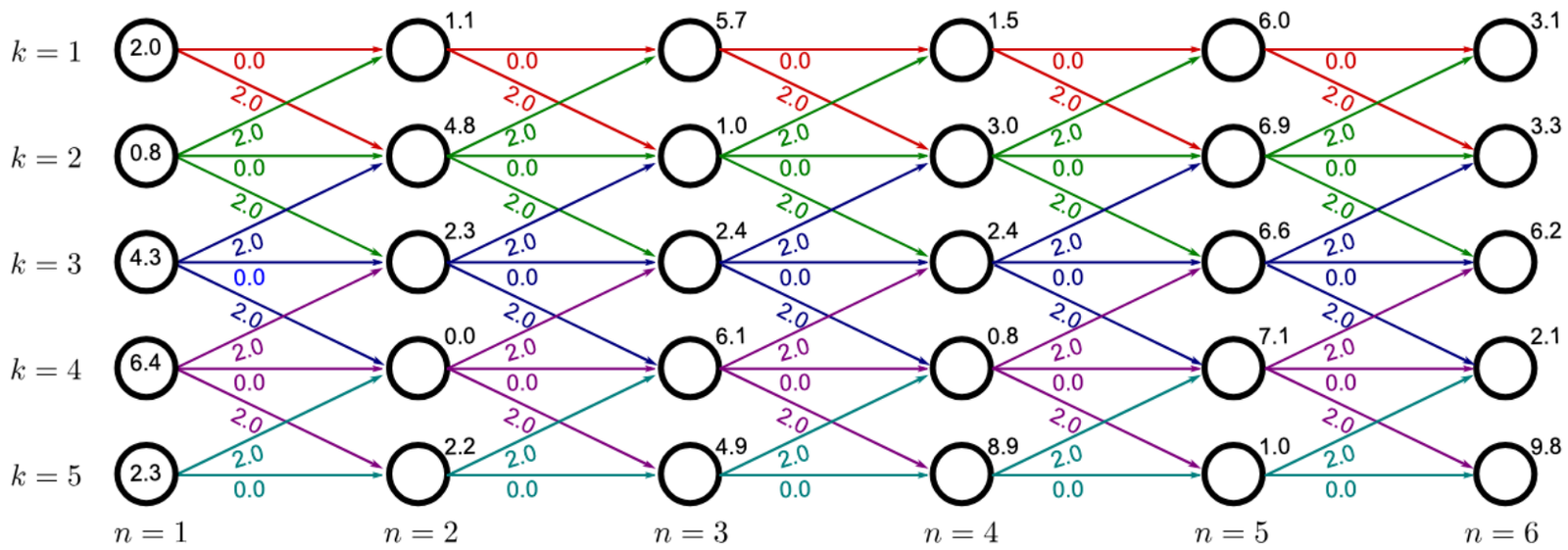


动态规划：Viterbi 算法

a)

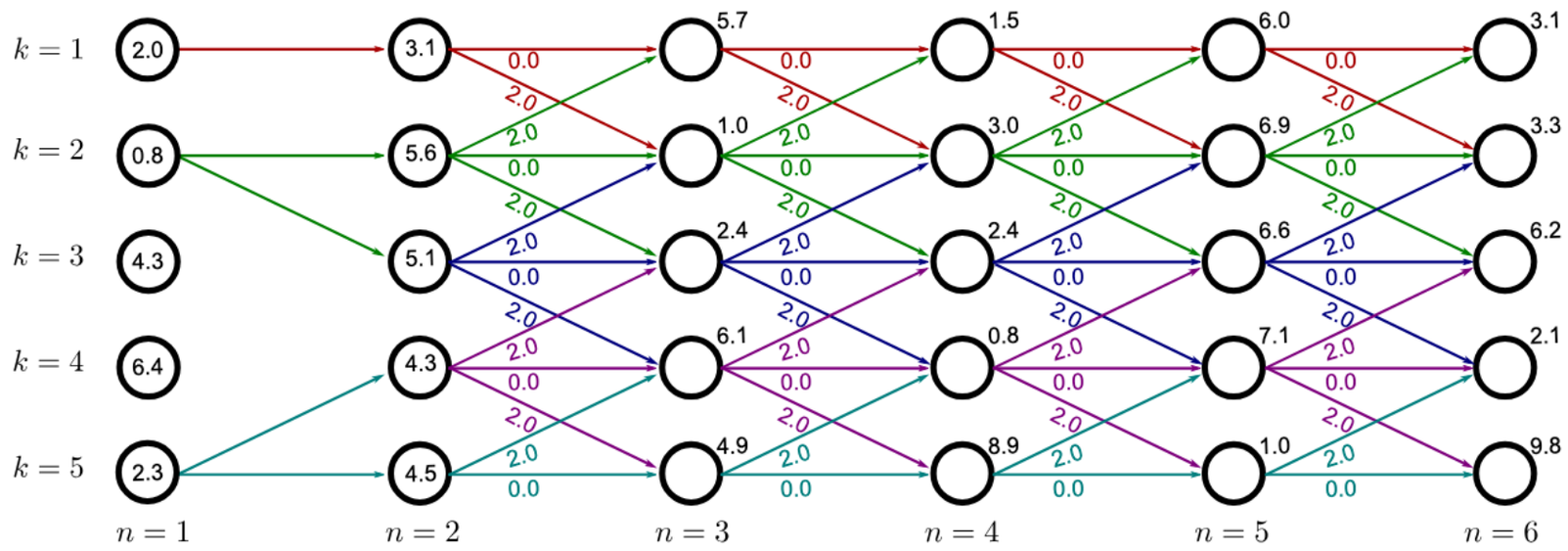


b)

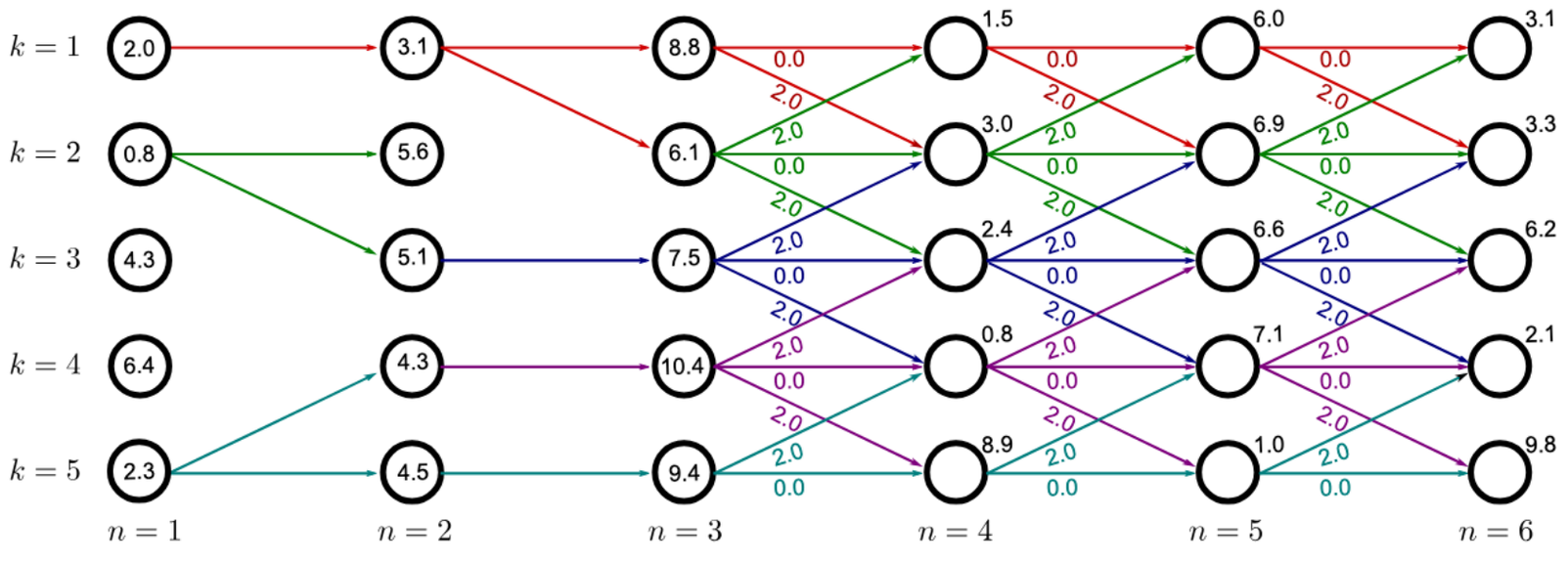


动态规划：Viterbi 算法

c)

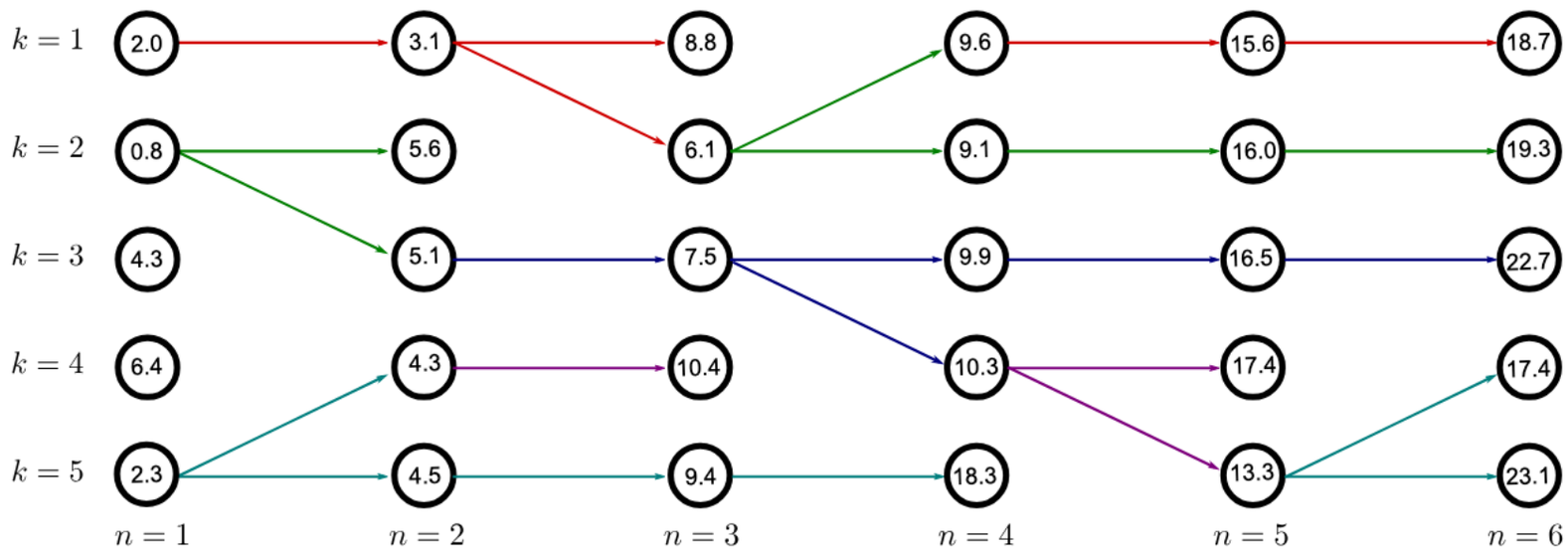


d)

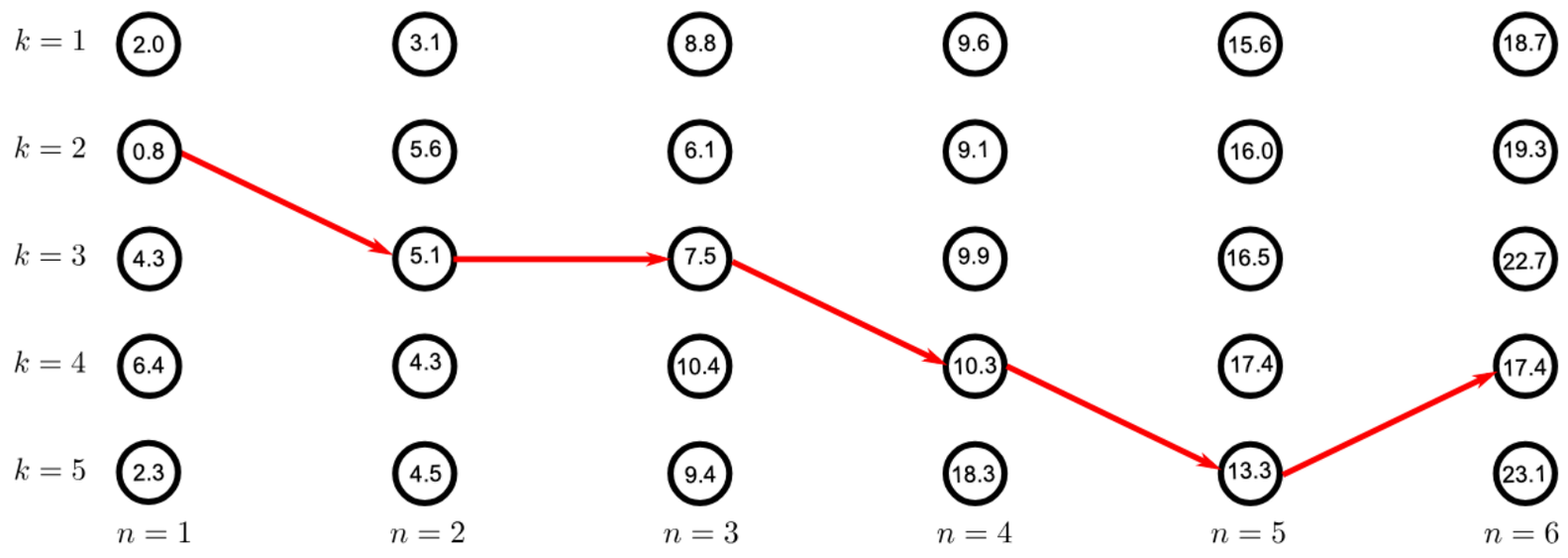


动态规划：Viterbi 算法

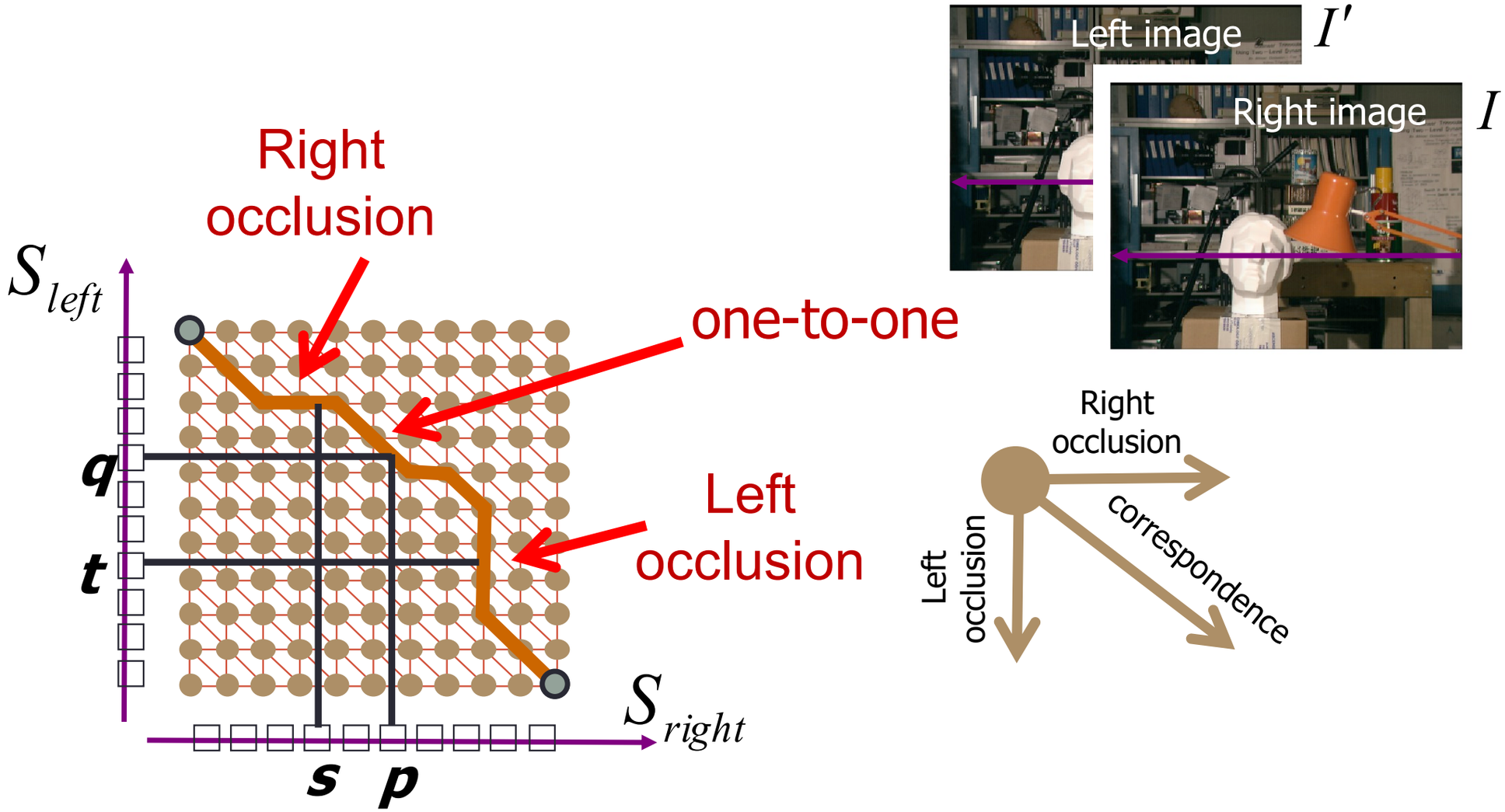
e)



f)



"最短路径"



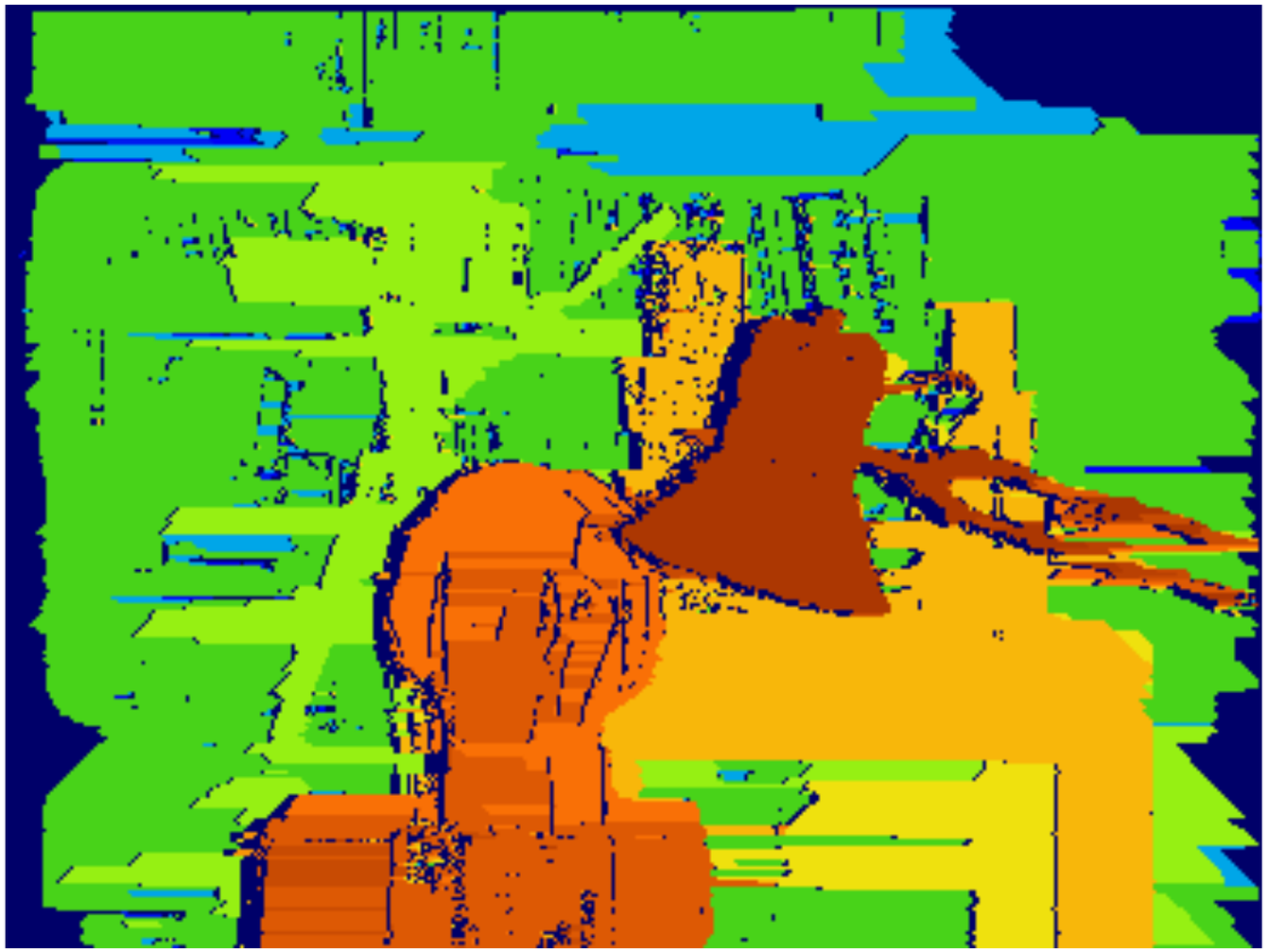
Can be implemented with dynamic programming

Ohta & Kanade '85, Cox et al. '96, Intille & Bobick, '01

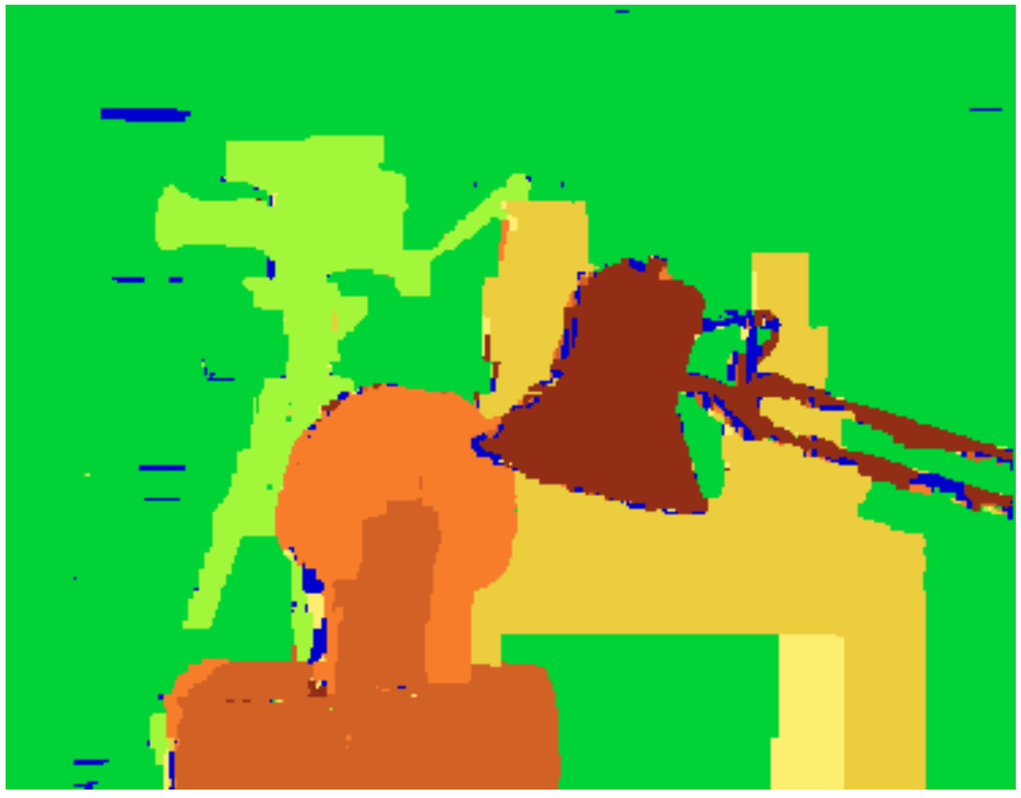
Slide credit: Y. Boykov

扫描线法会生成条状缺损

- 动态规划仅仅在线上找到一致性匹配，而不能在2D网格上



可以得到更好的结果



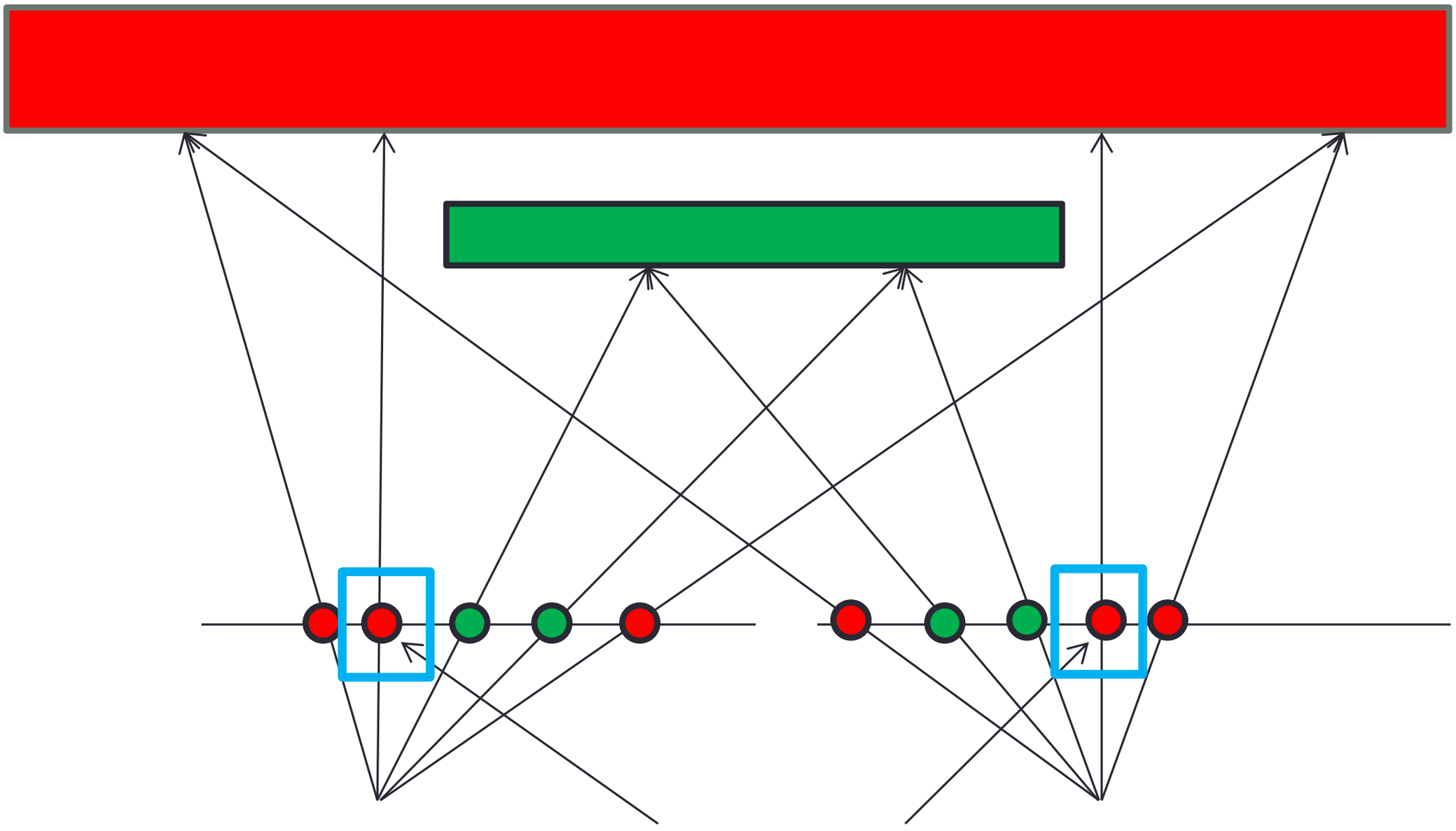
Graph cut method

Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts,
International Conference on Computer Vision, September 1999.

For the latest and greatest: http://www.middlebury.edu/stereo/

问题：遮挡



Occluded pixels

视差梯度约束

- 假定物体表面是分段连续的, 因此视差估计应该是局部平滑的

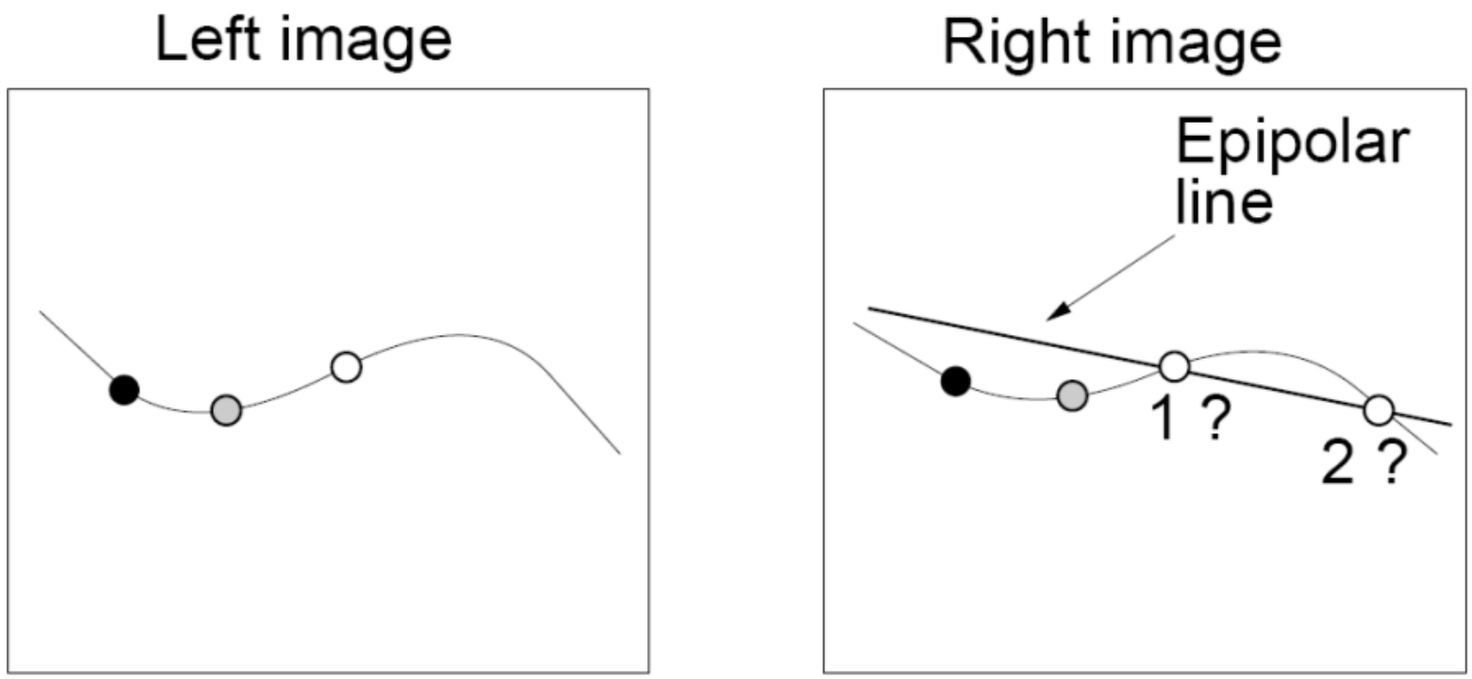


Figure from Gee & Cipolla

顺序约束

- 不透明物体上相同表面上的点在两个视图中出现的顺序相同

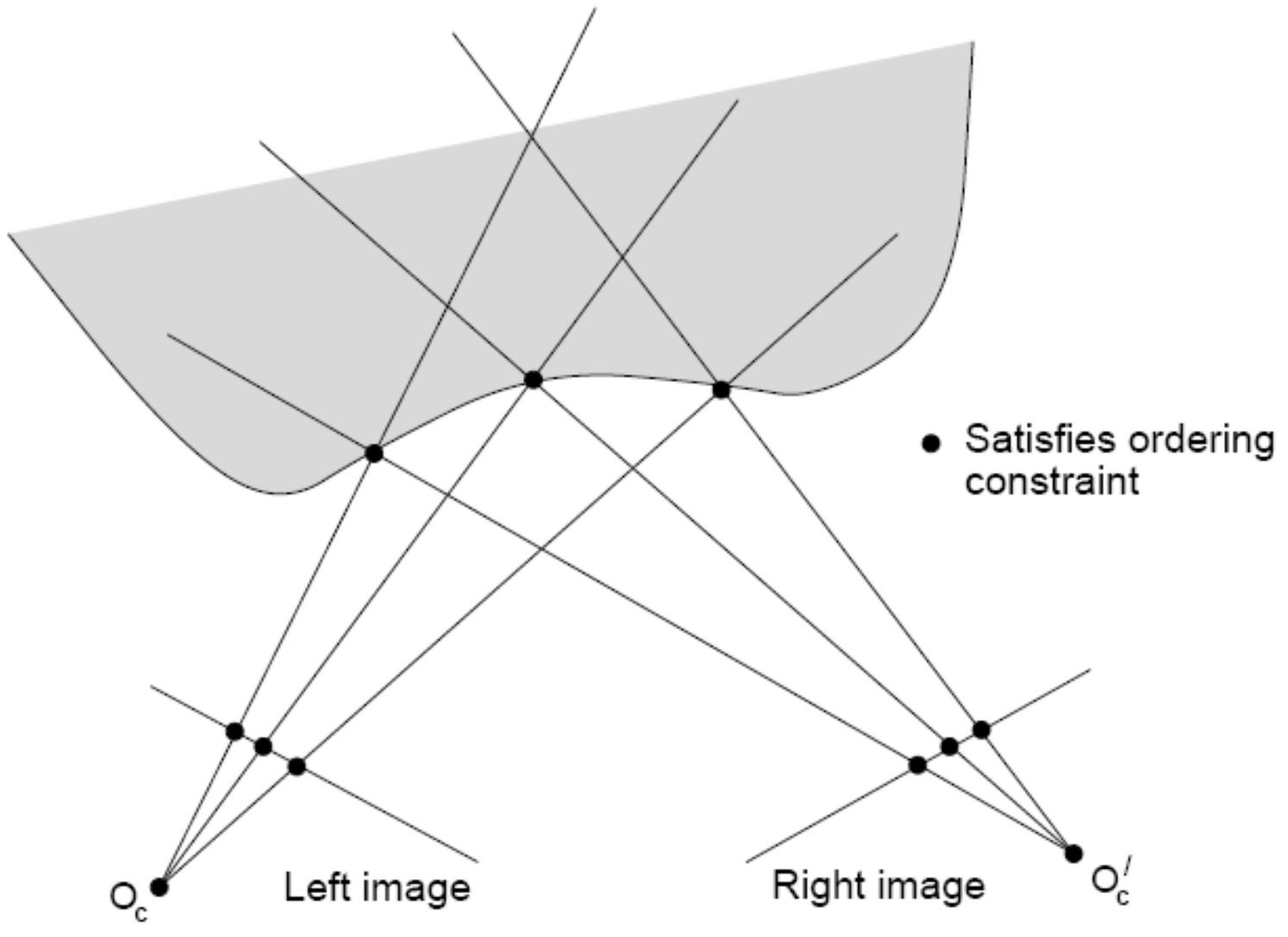
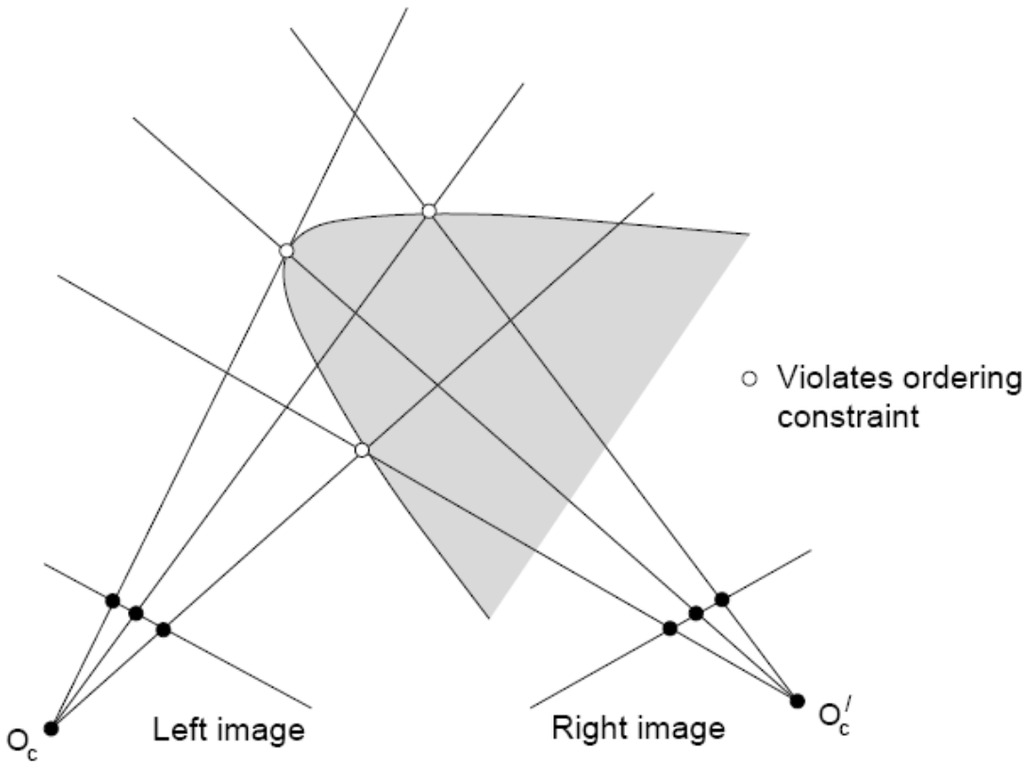
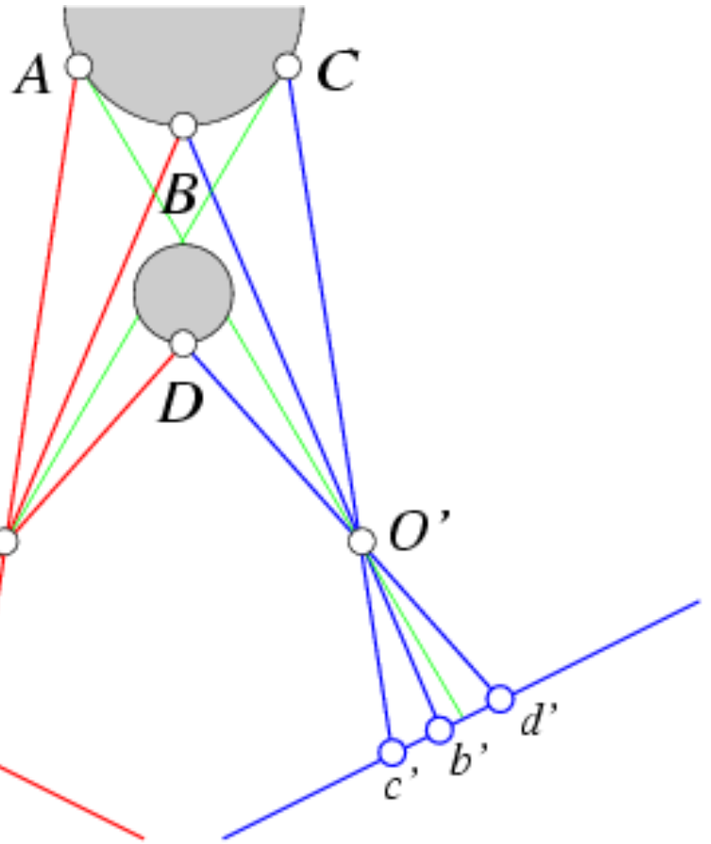


Figure from Gee & Cipolla

顺序约束

- 该约束不一定总是成立，如透明物体或者遮挡表面

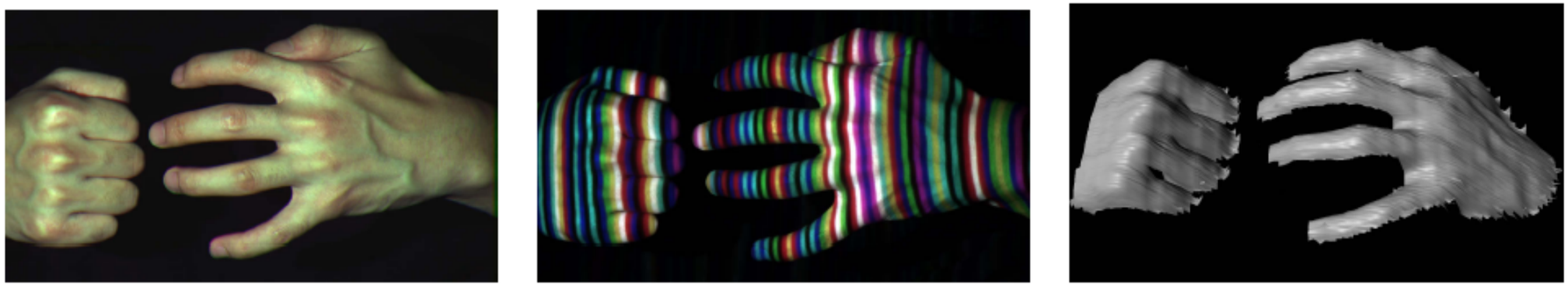




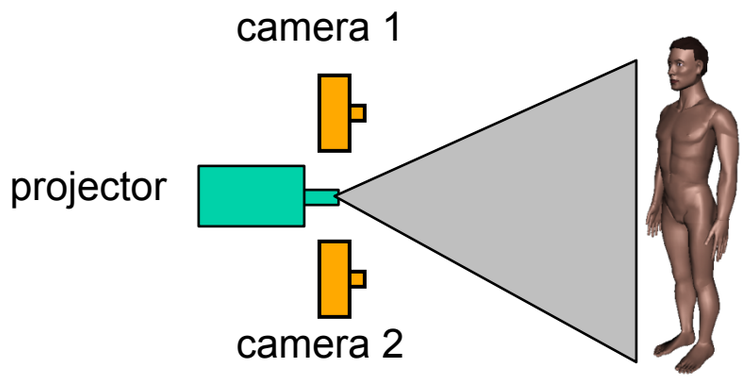
挑战问题

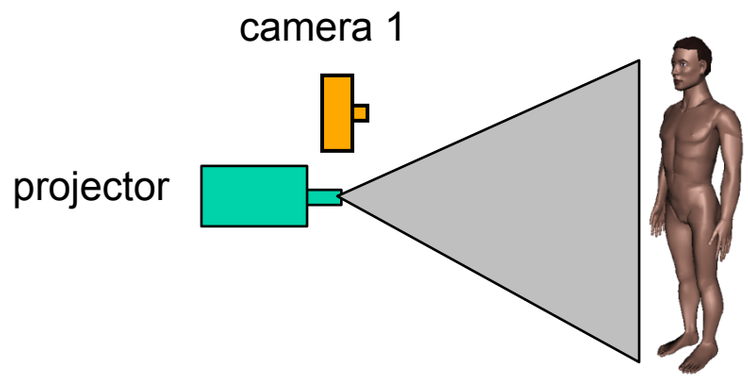
- 低对比度纹理缺乏图像区域
- 遮挡
- 亮度恒常性不满足
- 镜面反射
- 大基线
- 透视表观变形
- 相机标定误差

主动立体视觉: 结构光

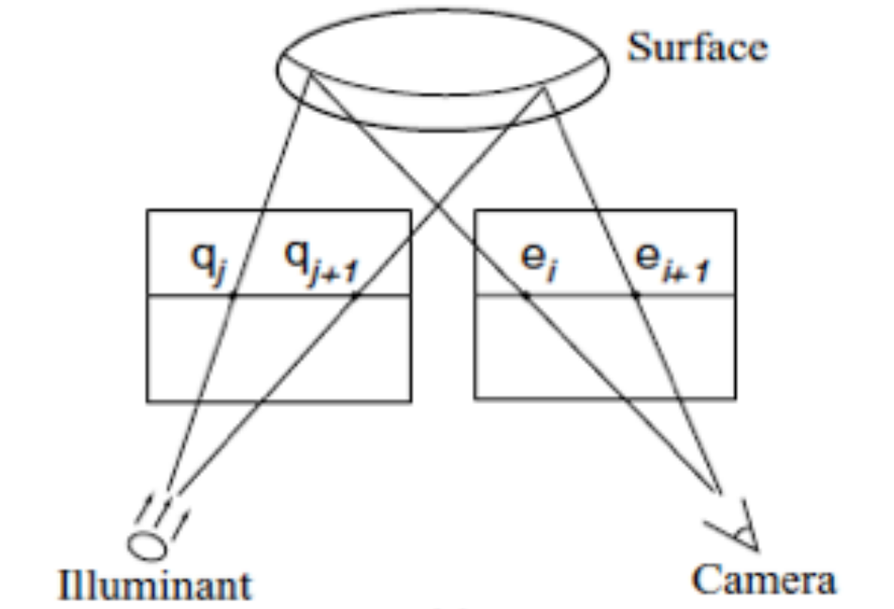


Project “structured” light patterns onto the object simplifies the correspondence problem



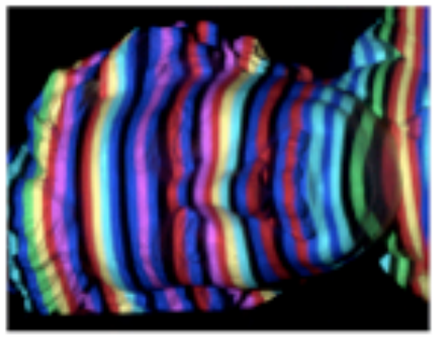


Slide credit: Rick Szeliski

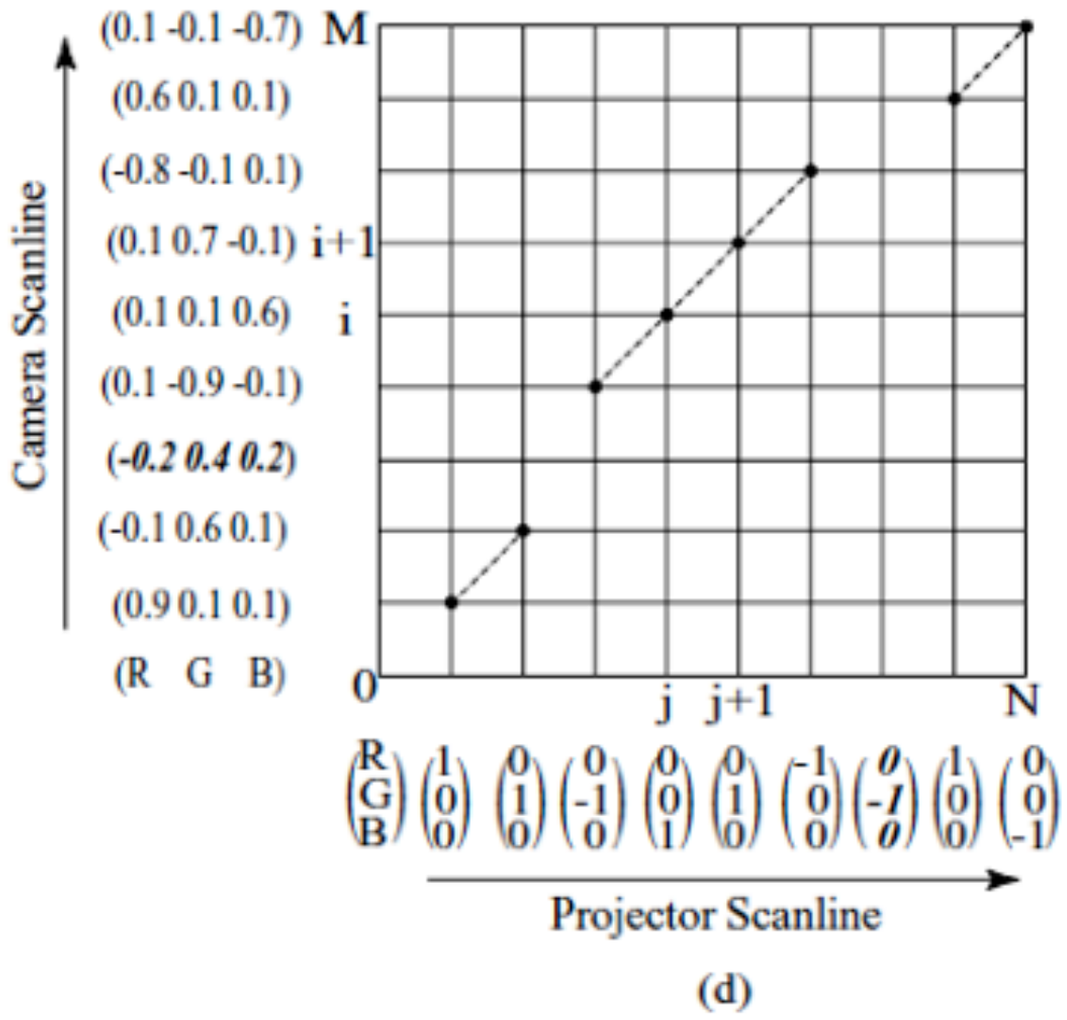


(a)

(b)



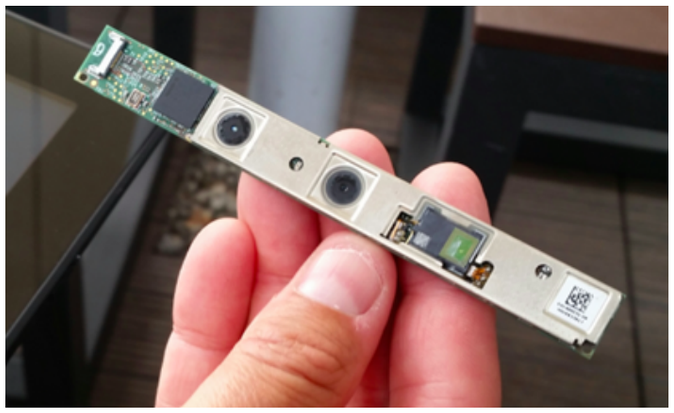
(c)



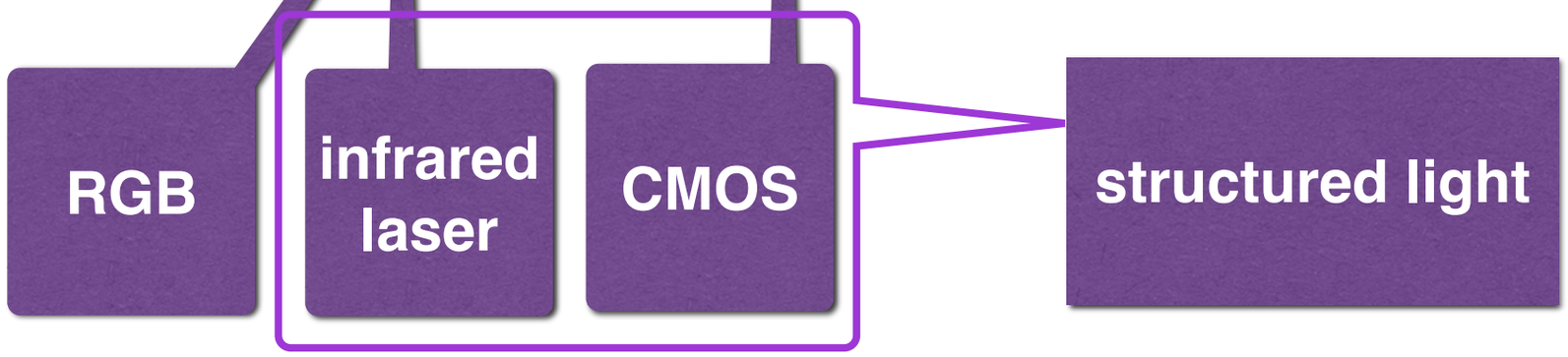
(d)

Figure 2. Summary of the one-shot method. (a) In optical triangulation, an illumination pattern is projected onto an object and the reflected light is captured by a camera. The 3D point is reconstructed from the relative displacement of a point in the pattern and image. If the image planes are rectified as shown, the displacement is purely horizontal (one-dimensional). (b) An example of the projected stripe pattern and (c) an image captured by the camera. (d) The grid used for multi-hypothesis code matching. The horizontal axis represents the projected color transition sequence and the vertical axis represents the detected edge sequence, both taken for one projector and rectified camera scanline pair. A match represents a path from left to right in the grid. Each vertex (j, i) has a score, measuring the consistency of the correspondence between e_i, the color gradient vectors shown by the vertical axis, and q_j, the color transition vectors shown below the horizontal axis. The score for the entire match is the summation of scores along its path. We use dynamic programming to find the optimal path. In the illustration, the camera edge in bold italics corresponds to a false detection, and the projector edge in bold italics is missed due to, e.g., occlusion.

Kinect-v1: structured infrared light



Intel laptop depth camera



Time of flight (Kinect - v2)

- Depth cameras in HoloLens use *time of flight*
  - "SONAR for light"
  - Emit light of a known wavelength, and time how long it takes for it to come back