

立体视觉与三维建模

申抒含
中国科学院自动化研究所
模式识别国家重点实验室

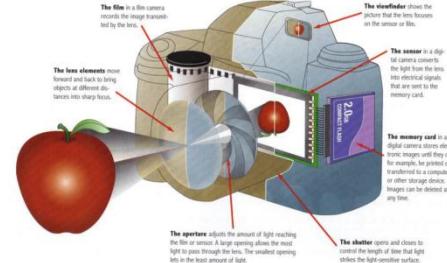


Robot Vision Group

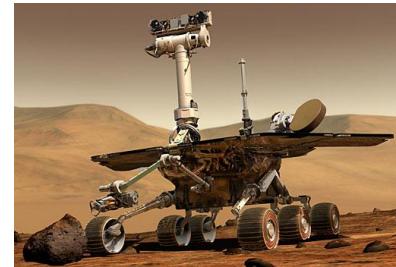
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences

三维计算机视觉

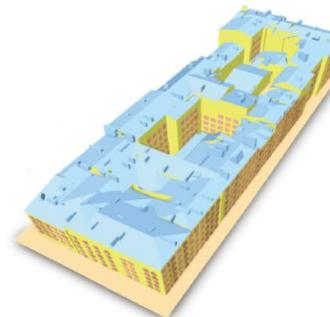
6、相机模型与多视几何



7、相机标定与稀疏重建



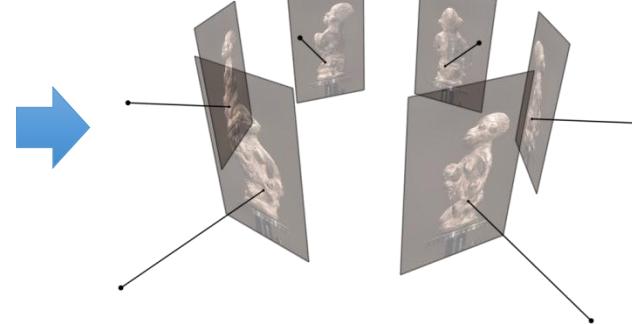
8、立体视觉与三维建模



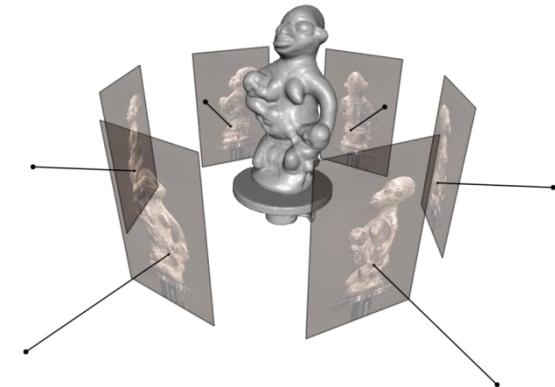
立体视觉与三维建模



多视角图像



相机位姿



三维模型

图像的获取方式

- 室内环境
 - 光照可控
 - 相机稳定
- 室外环境
 - 自然光线
 - 手持拍摄
- 网络图片
 - 不可控



图像的获取方式—室内环境



图像的获取方式—室外环境



图像的获取方式—网络图片



colosseum

Page 3 of about 1,290,000 results (0.05 seconds)

Search

SafeSearch moderate ▾

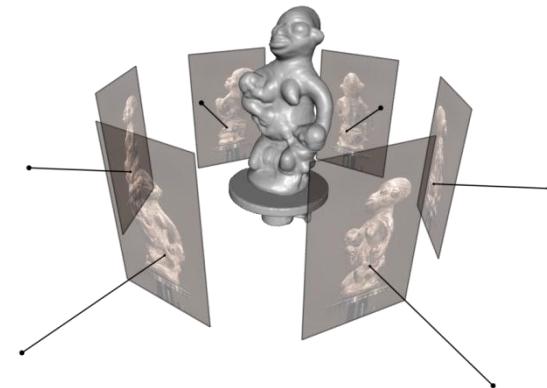
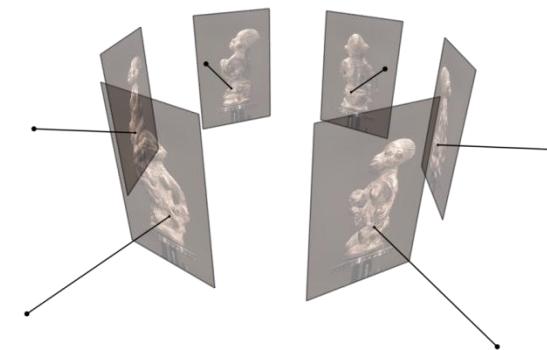
Advanced search

Google search results for "colosseum". The results are displayed in a grid format, showing various images of the Colosseum from different angles and sources.

The results include:

- The Colosseum, Rome (405 x 261 - 34k - jpg, traveltheroads.com, Find similar images)
- Colosseum (700 x 466 - 192k - jpg, faqs.org, Find similar images)
- Roman Colosseum (498 x 360 - 46k - jpg, mariamilani.com, Find similar images)
- The Colosseum (437 x 300 - 59k - jpg, agirsnotebook.page.ph, Find similar images)
- Roman Colosseum (375 x 500 - 58k - jpg, goitally.about.com, Find similar images)
- Colosseum (640 x 435 - 92k - jpg, students.washington.edu, Find similar images)
- the-colosseum-rome-1600 x 1200 - 304k - jpg, 1photoblog.wordpress.com, Find similar images)
- the colosseum rome (1600 x 1200 - 616k - jpg, manywallpapers.com, Find similar images)
- Italy Rome Colosseum (720 x 480 - 20k - jpg, solarnavigator.net, Find similar images)
- colosseum.jpg (500 x 320 - 50k - jpg, salem.k12.va.us, Find similar images)
- The Colosseum was (444 x 275 - 113k - gif, library.thinkquest.org, Find similar images)
- The Colosseum is a (337 x 450 - 46k - jpg, tripadvisor.co.uk, Find similar images)
- the colosseum rome (400 x 300 - 53k - jpg, picsdigger.com, Find similar images)
- Rome monuments: (1000 x 665 - 116k - jpg, hotelrome.net, Find similar images)
- Image of Colosseum (500 x 375 - 41k - jpg, traveladventures.org, Find similar images)
- Free Ware: (350 x 500 - 40k - jpg, freewarezps2.blogspot.com, Find similar images)
- Colosseum An (576 x 449 - 191k - jpg, learnlangs.com, Find similar images)
- Battle Arena (500 x 375 - 38k - jpg, thehexcitingtraveldesti..., Find similar images)
- Roman Colosseum (450 x 360 - 68k - jpg, issa-italy.org, Find similar images)
- Colosseum (416 x 300 - 35k - jpg, colorado.edu, Find similar images)
- ancient rome (567 x 475 - 136k - jpg, picsdigger.com, Find similar images)
- Colosseum. (470 x 340 - 54k - jpg, colosseum.net, Find similar images)
- Recreating The (550 x 382 - 74k - jpg, antsmagazine.com, Find similar images)
- Colosseum (1500 x 880 - 435k - jpg, news.satimagingcorp.com, Find similar images)
- The Colosseum and (667 x 500 - 128k - jpg, nomadicmatt.com, Find similar images)
- The mighty (800 x 600 - 378k - jpg, travelerfolio.com, Find similar images)
- Pizza Colosseum (468 x 334 - 175k - jpg, webecoist.com, Find similar images)
- Obligatory (1024 x 771 - 309k - jpg, cs.unm.edu, Find similar images)

相机位姿的获取方式



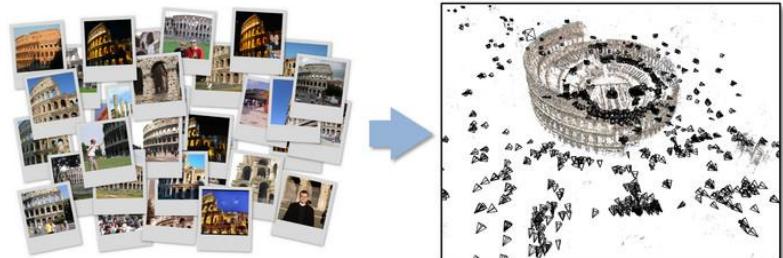
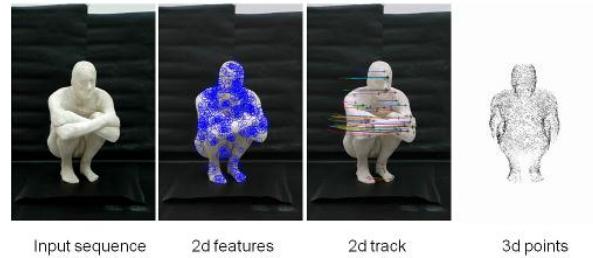
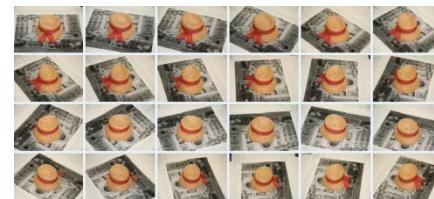
多视角图像

相机位姿

三维模型

相机位姿的获取方式

- 机械臂
- 人工标志物
- Global Structure-from-Motion
- Incremental Structure-from-Motion



相机位姿的获取方式

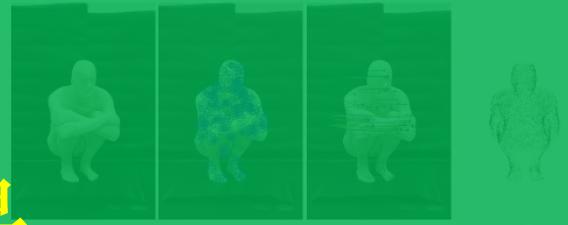
- 机械臂
- 人工标志物

小场景



- Global Structure-from-Motion

大场景



- Incremental Structure-from-Motion

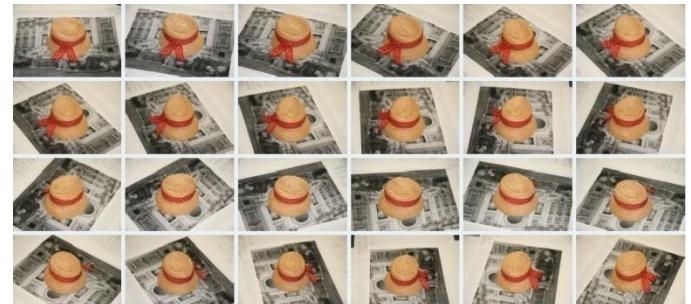
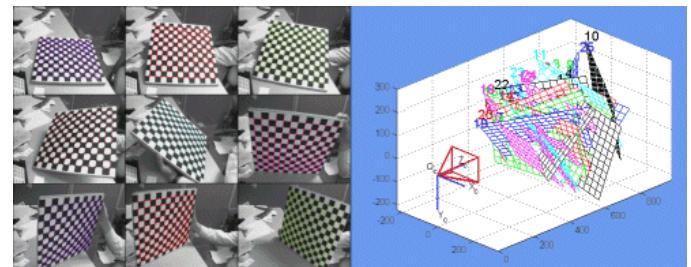
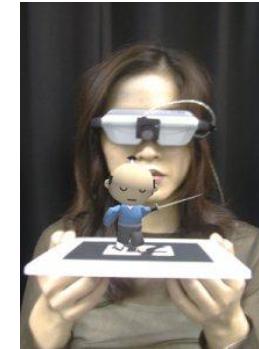


相机位姿的获取方式—机械臂

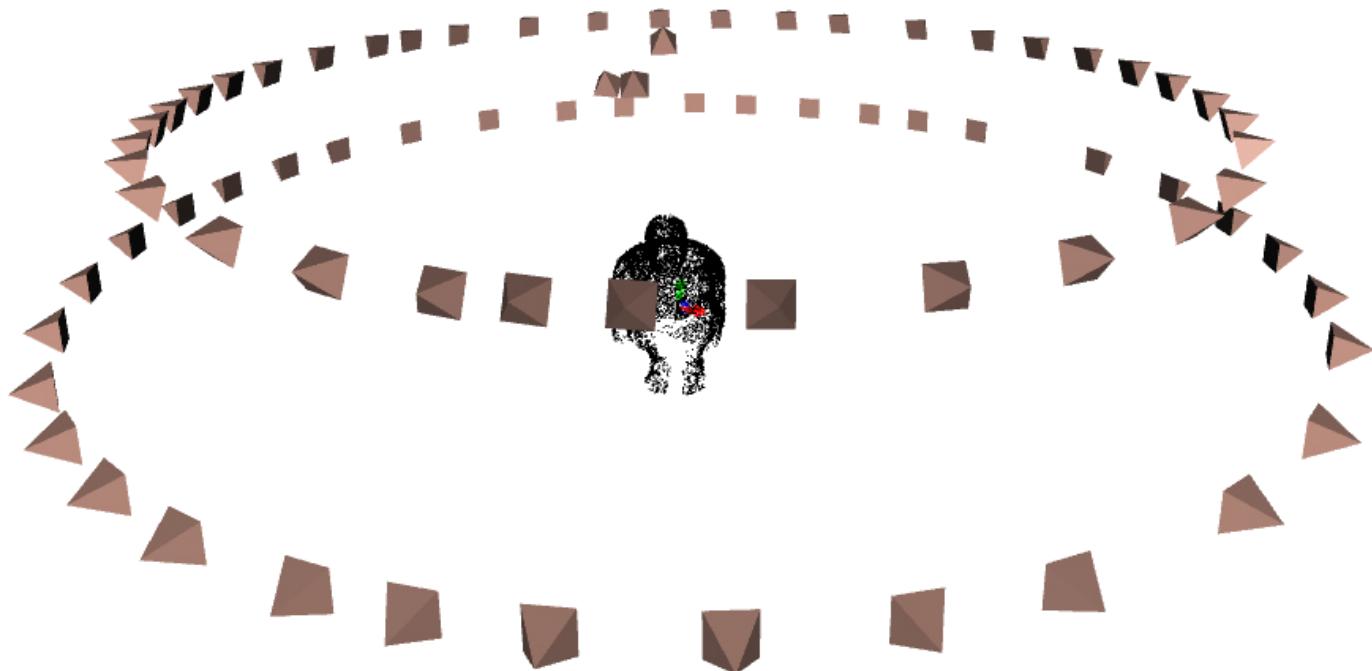


相机位姿的获取方式—人工标志物

- ARToolkit
- MATLAB Toolbox
- Robust planar patterns



相机位姿的获取方式—Global SfM



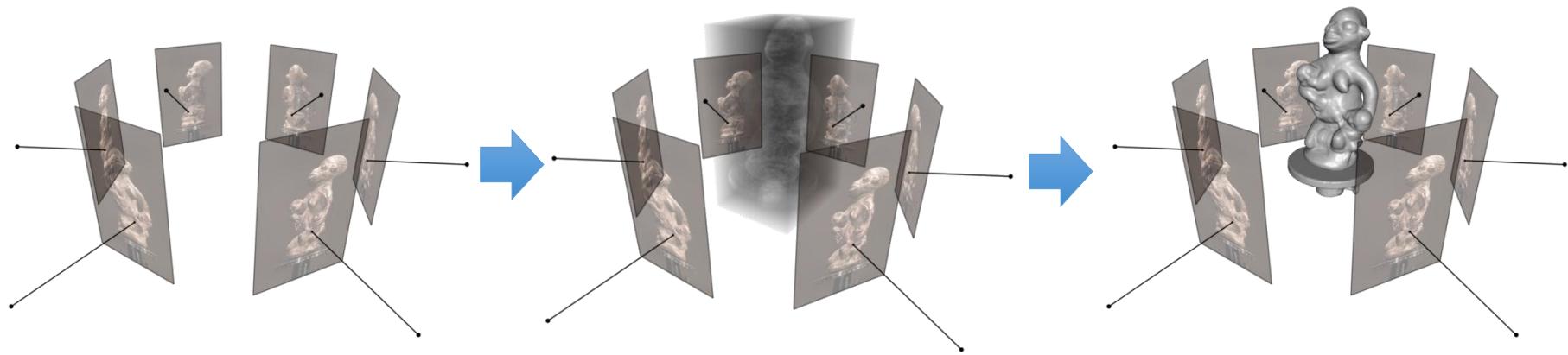
相机位姿的获取方式—Incremental SfM

- Two-view SfM
- Adding images
- Bundle-adjustment

phototour.cs.washington.edu/bundler



三维建模基本流程

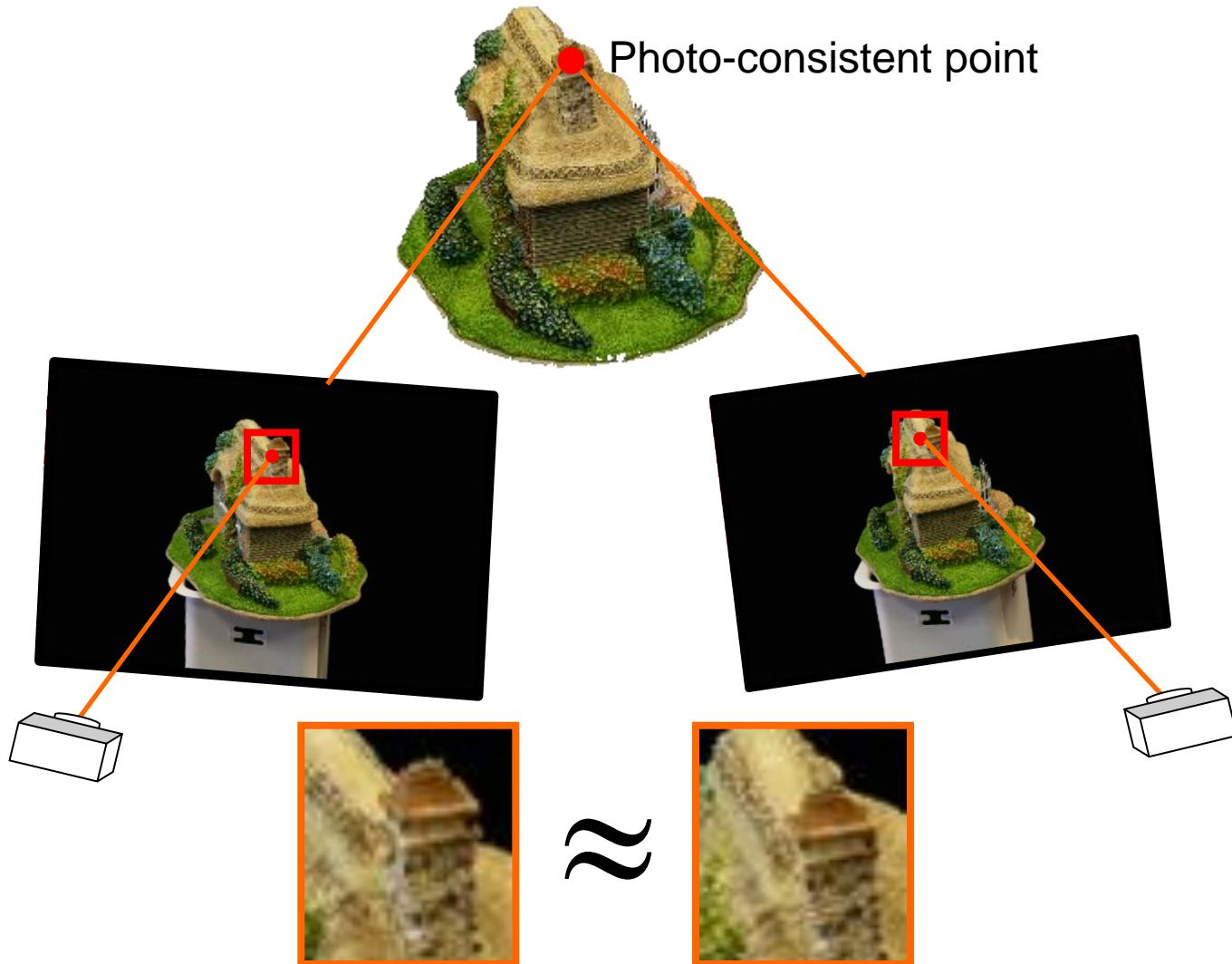


相机位姿

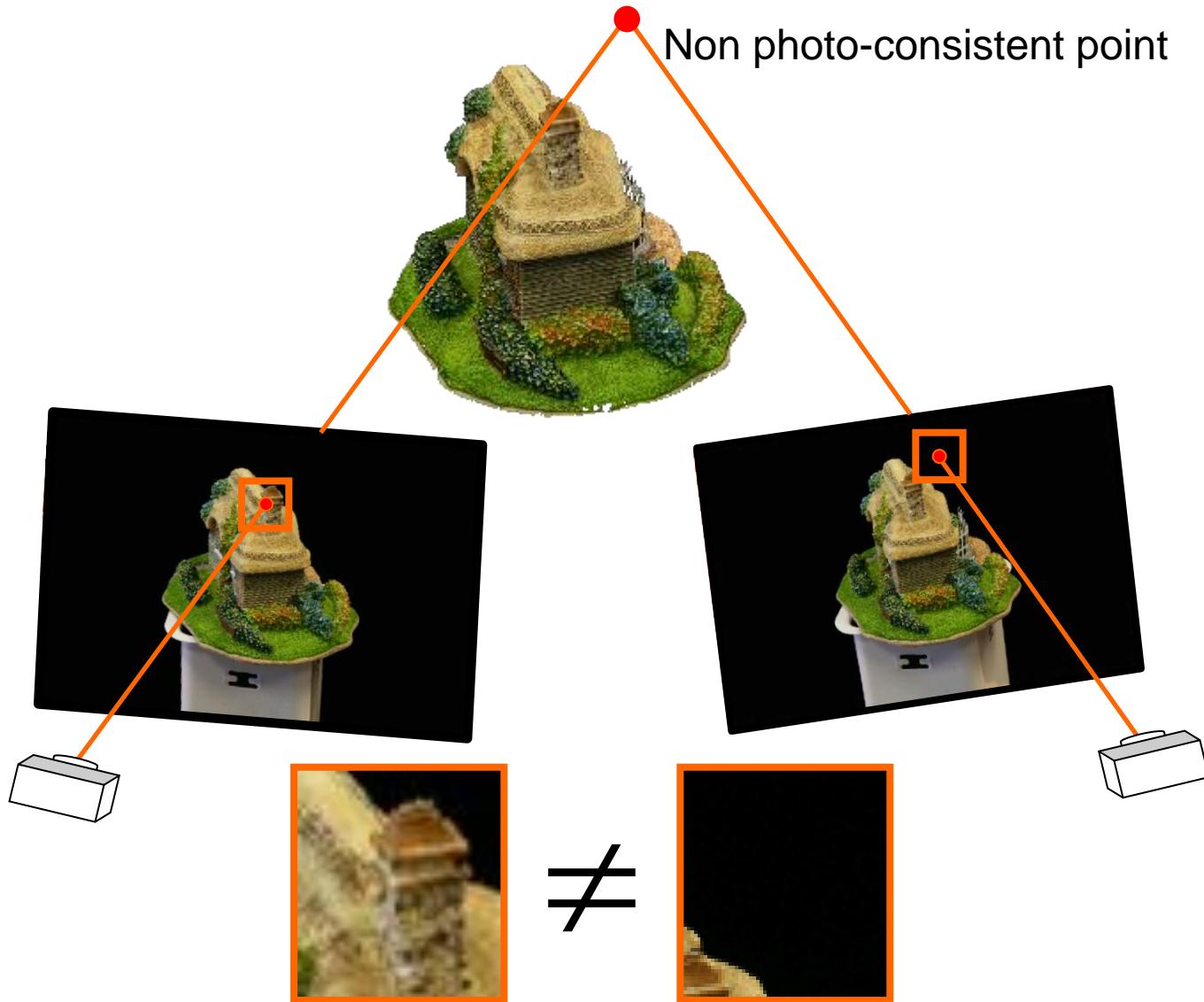
图像一致性

三维模型

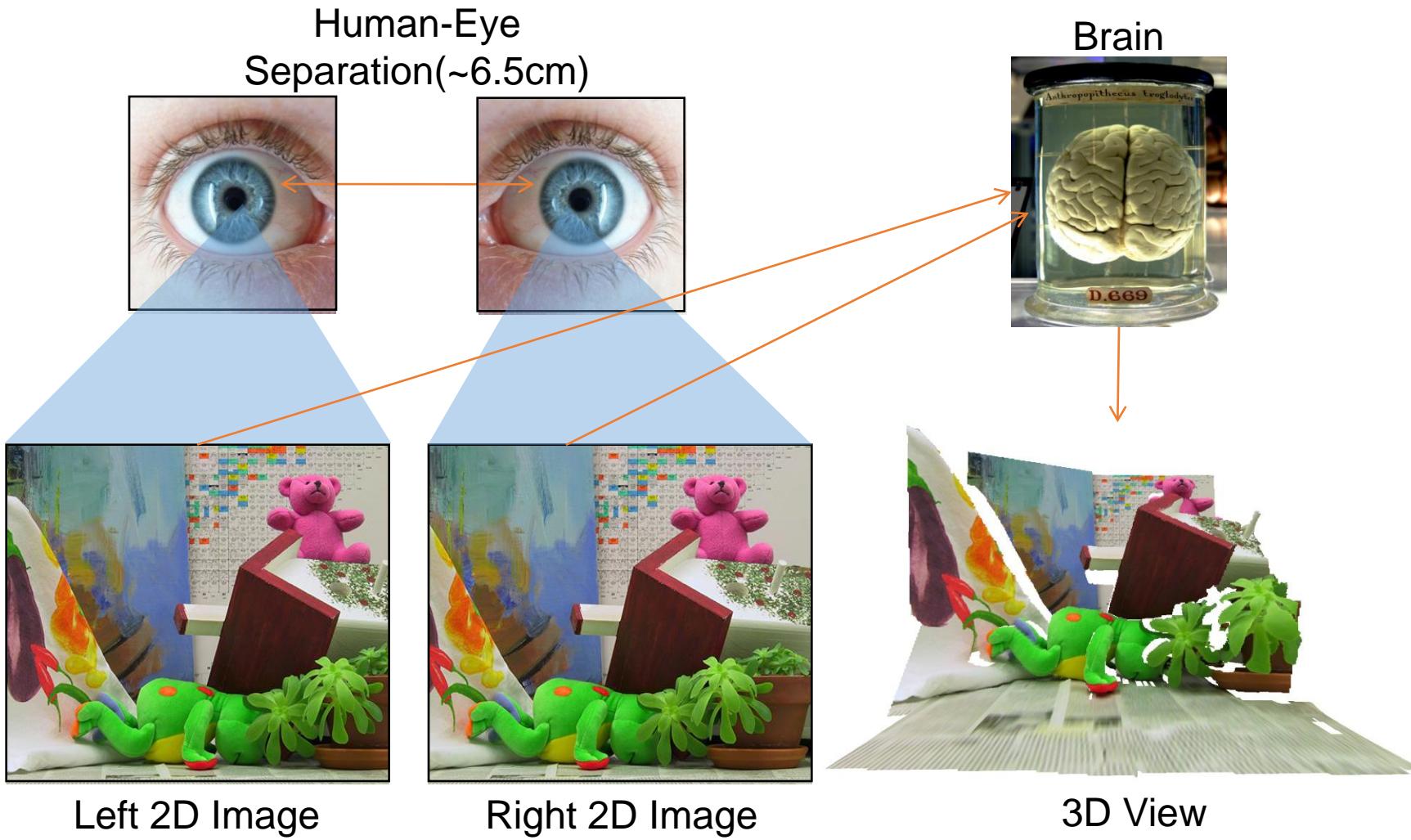
图像一致性 (Photo consistency)



图像一致性 (Photo consistency)



两视图三维建模—立体视觉

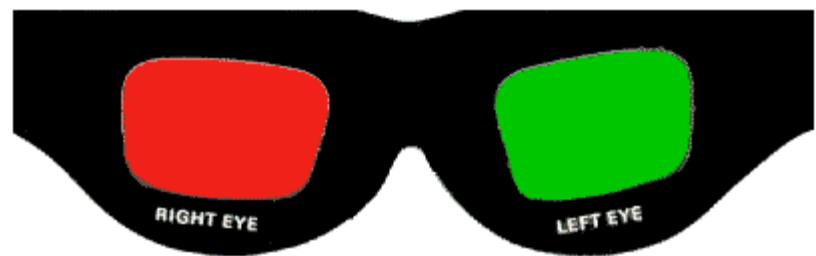


立体视觉—3D感知

- 红绿图像叠加生成3D视频
- 需要佩戴红绿眼睛
- 红色镜片过滤掉红色图像、
绿色镜片过滤掉绿色图像
- 双眼获取不同图像，产生
3D感



(“红/绿”图)



(“红/绿”眼镜)

立体视觉—3D感知

- 图像以帧序列的格式在左右眼交替出现
- 当左眼快门打开时右眼快门关闭；当右眼快门打开时左眼快门关闭
- 需要快门有足够高的刷新率（120Hz）



(快门式3D眼镜，120 Hz)



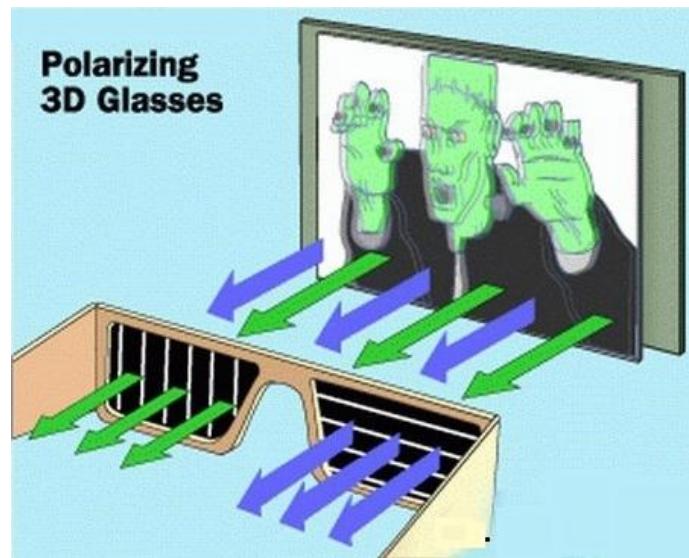
(Nvidia Artwork)

立体视觉—3D感知

- 显示屏幕上加偏光板，向观看者输送偏振方向不同的两幅画面
- 偏振式眼镜的每只镜片只能接受一个偏振方向的画面
- 眼镜成本低，但需要专业放映设备



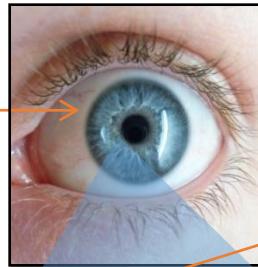
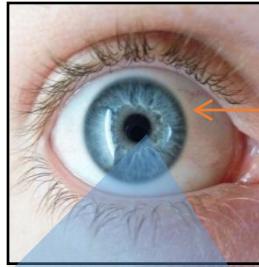
(偏振式3D眼镜)



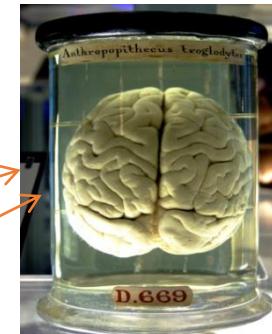
(偏振式3D)

立体视觉

Human-Eye
Separation(~6.5cm)



Brain



Left 2D Image

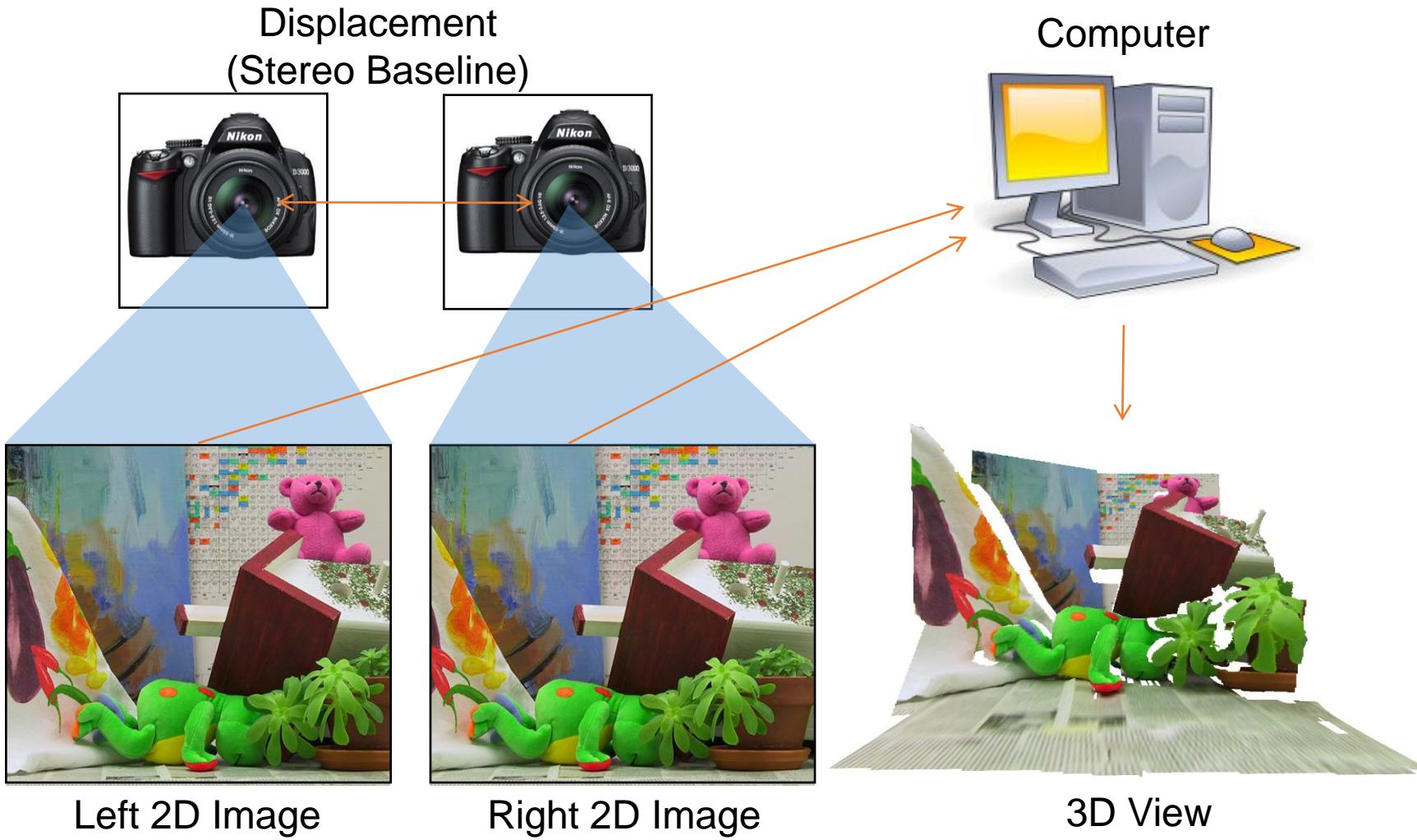


Right 2D Image

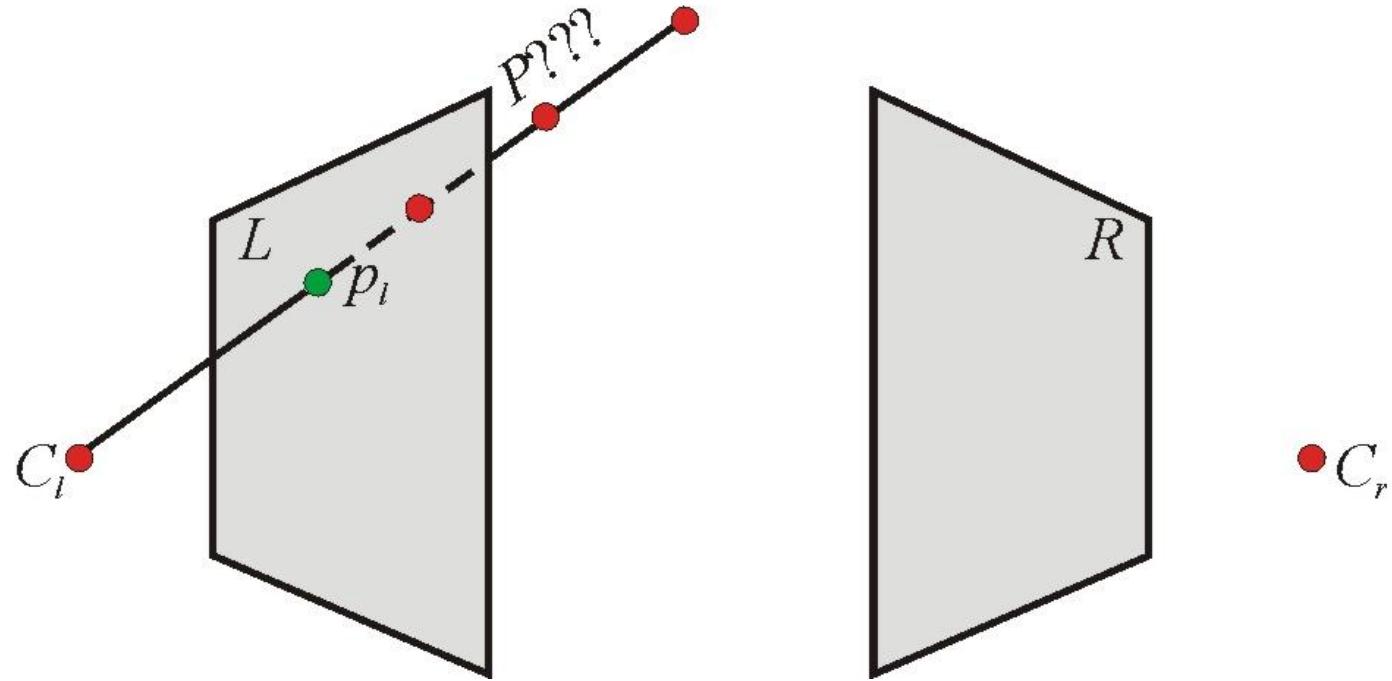


3D View

立体视觉

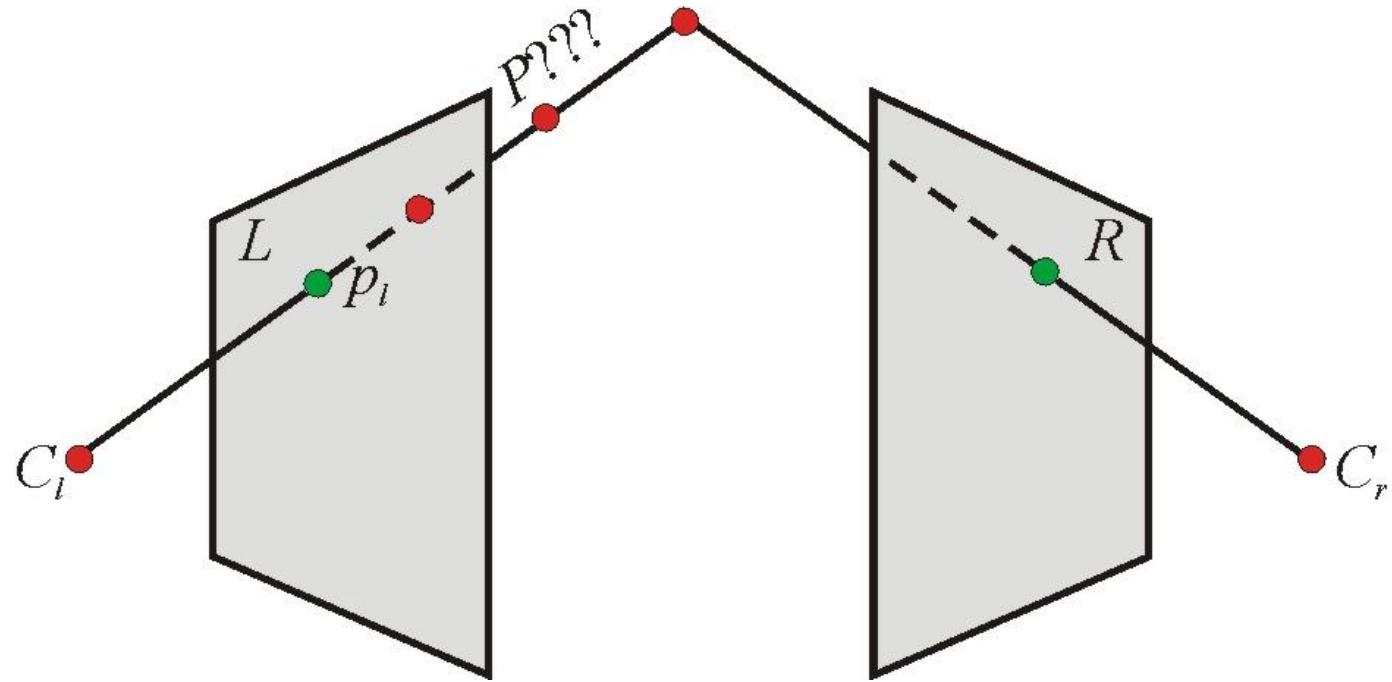


立体视觉—极几何



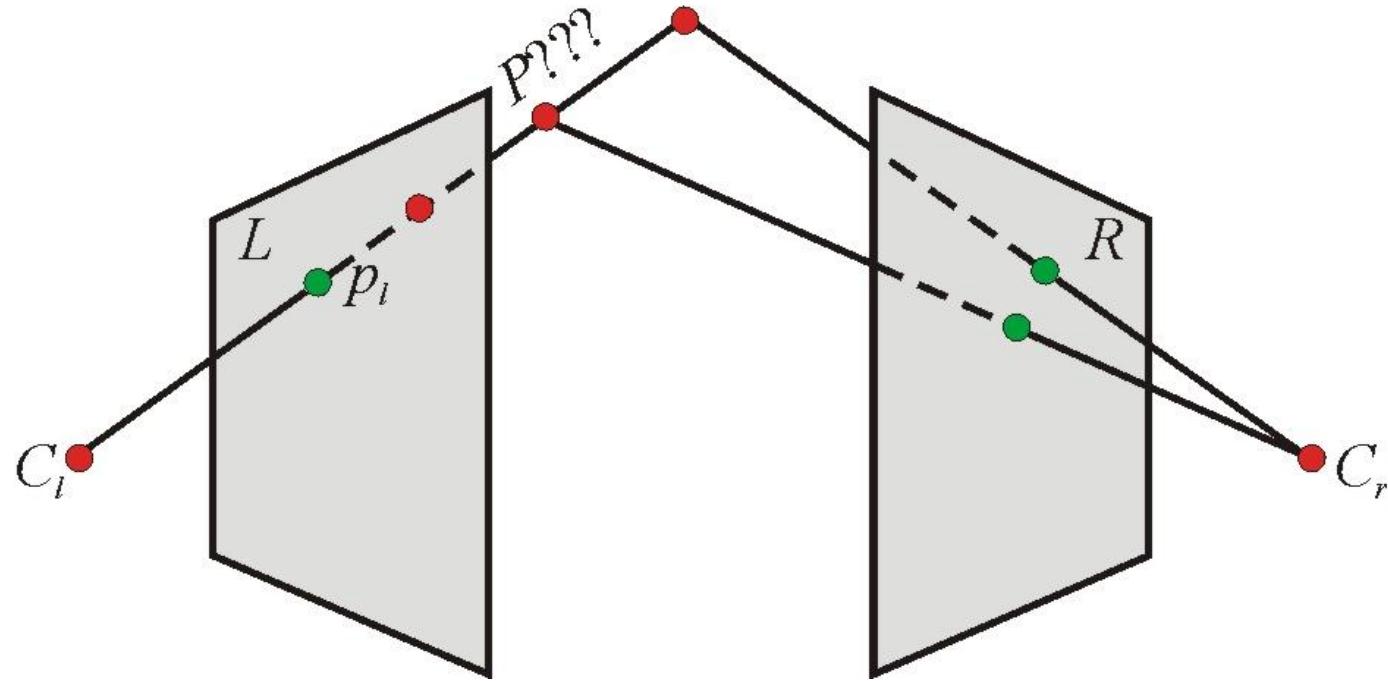
两视图的极几何约束

立体视觉—极几何



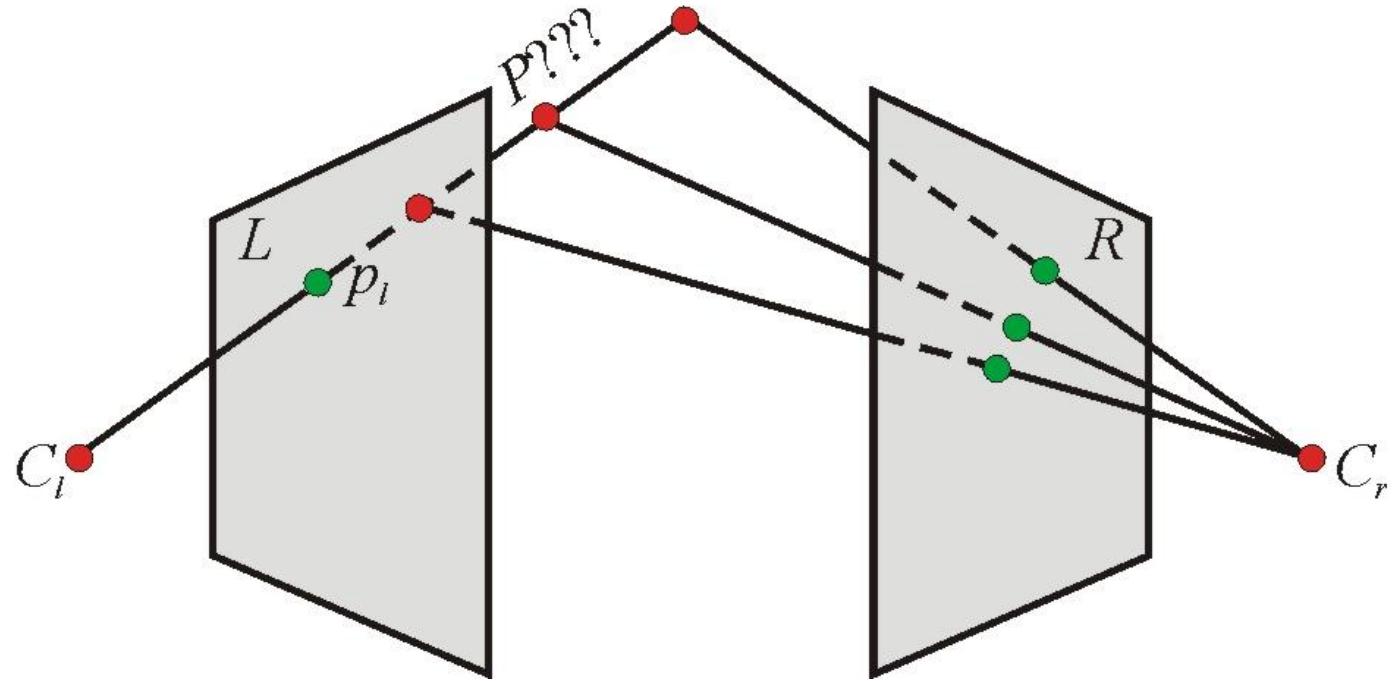
两视图的极几何约束

立体视觉—极几何



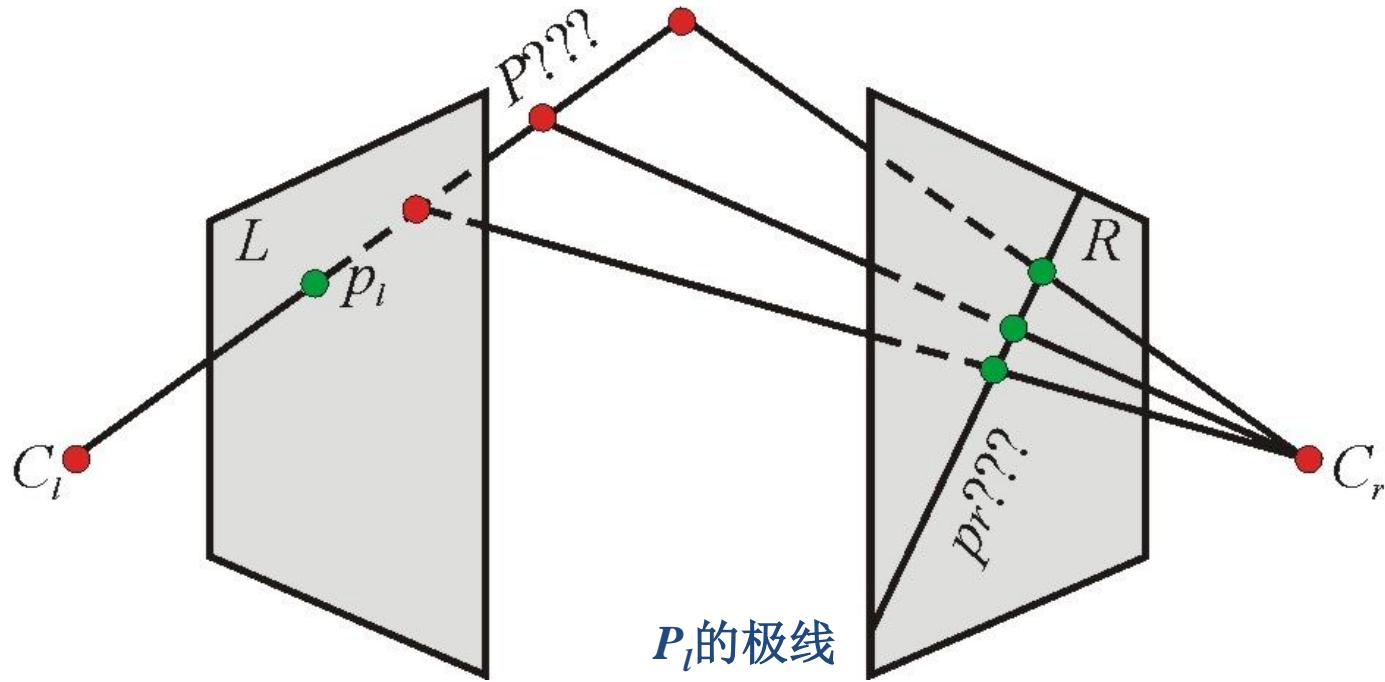
两视图的极几何约束

立体视觉—极几何



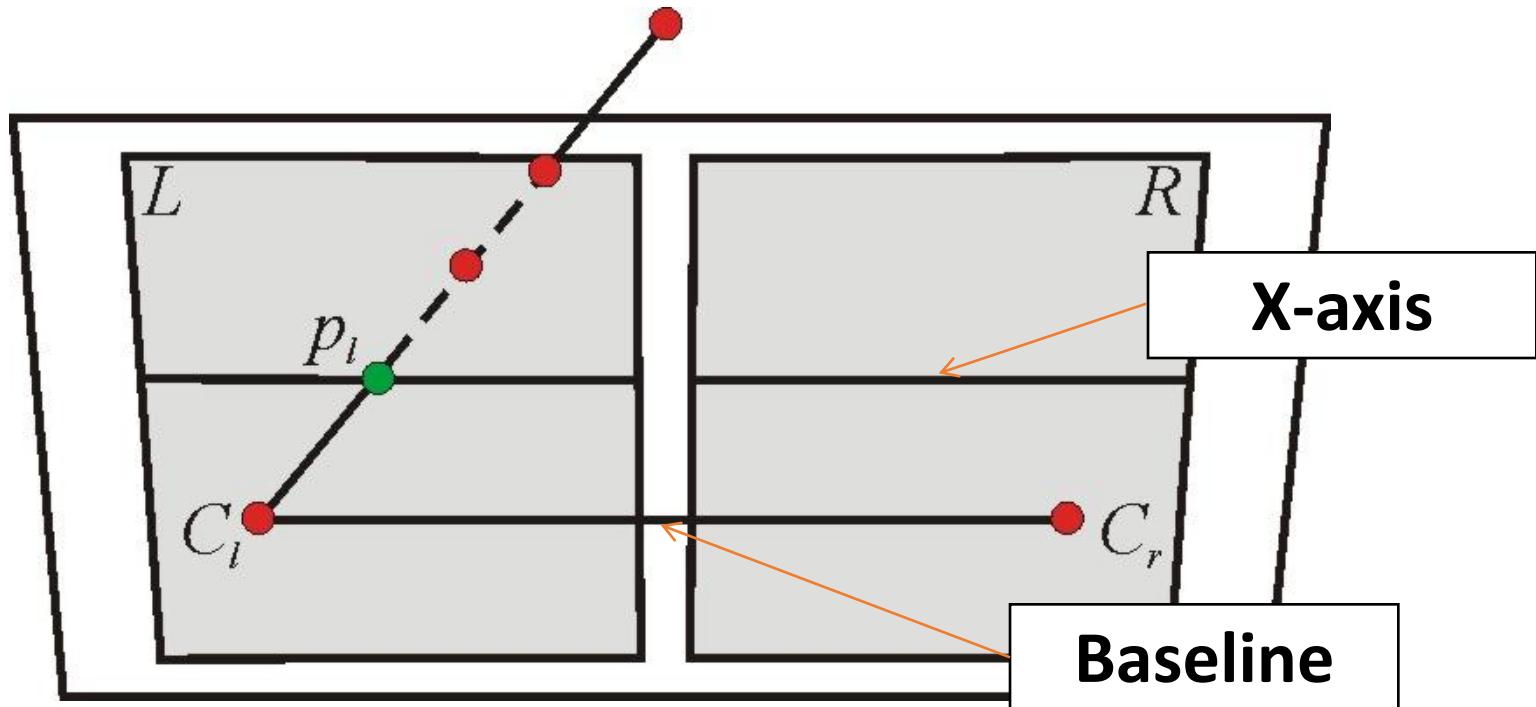
两视图的极几何约束

立体视觉—极几何



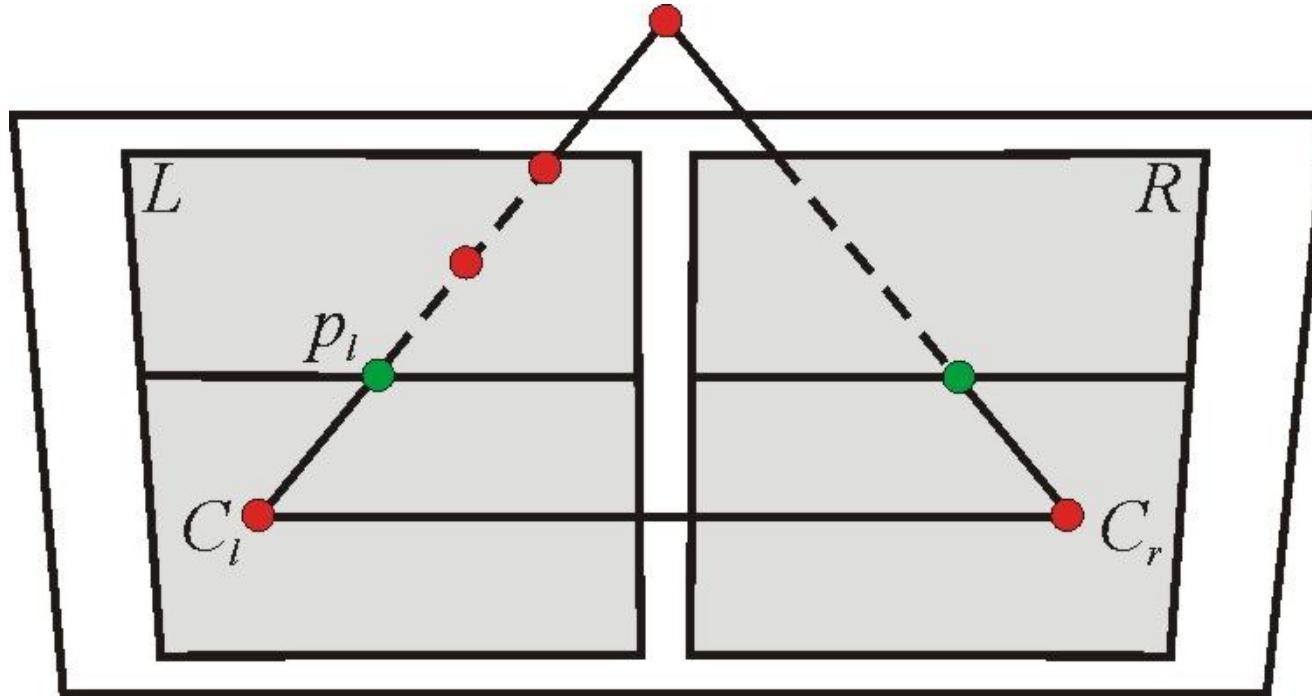
- P_l 在右图像上的匹配点在 P_l 的极线上
- 搜索 P_l 匹配点的过程是一维搜索

立体视觉—极线校正



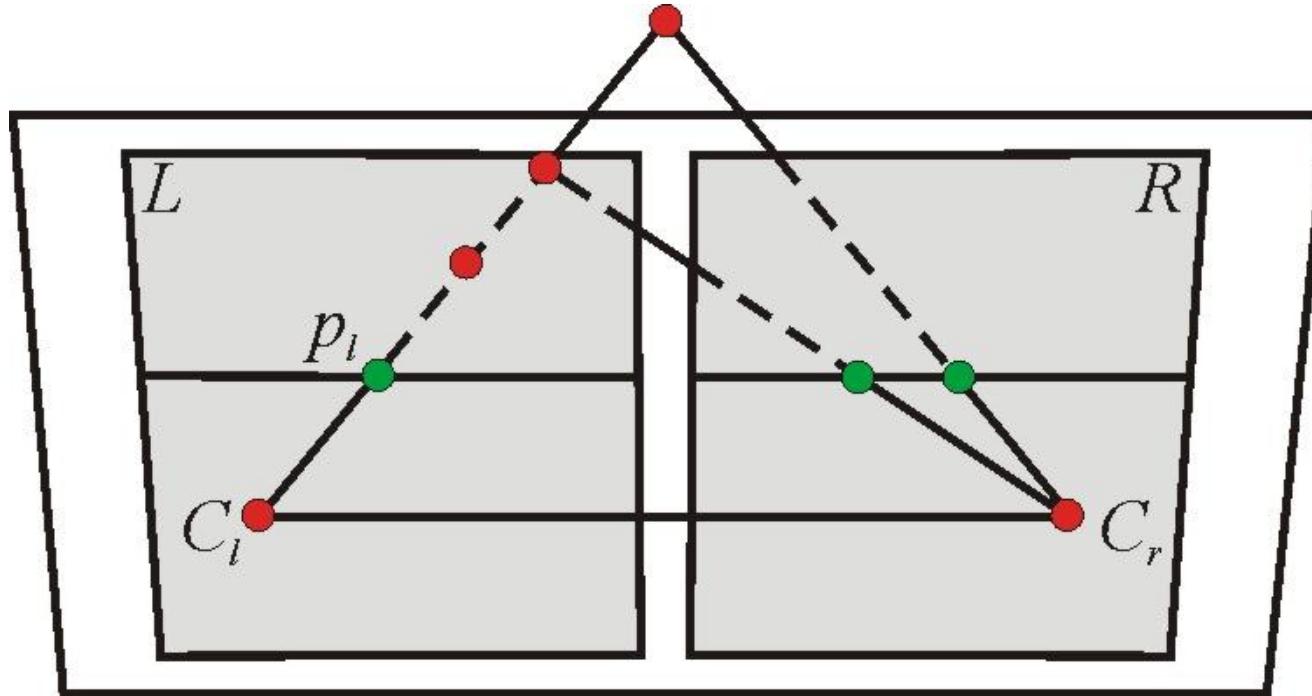
- 左图像L和右图像R位于同一平面
- L和R的x轴与基线平行，对应点具有相同的y坐标

立体视觉—极线校正



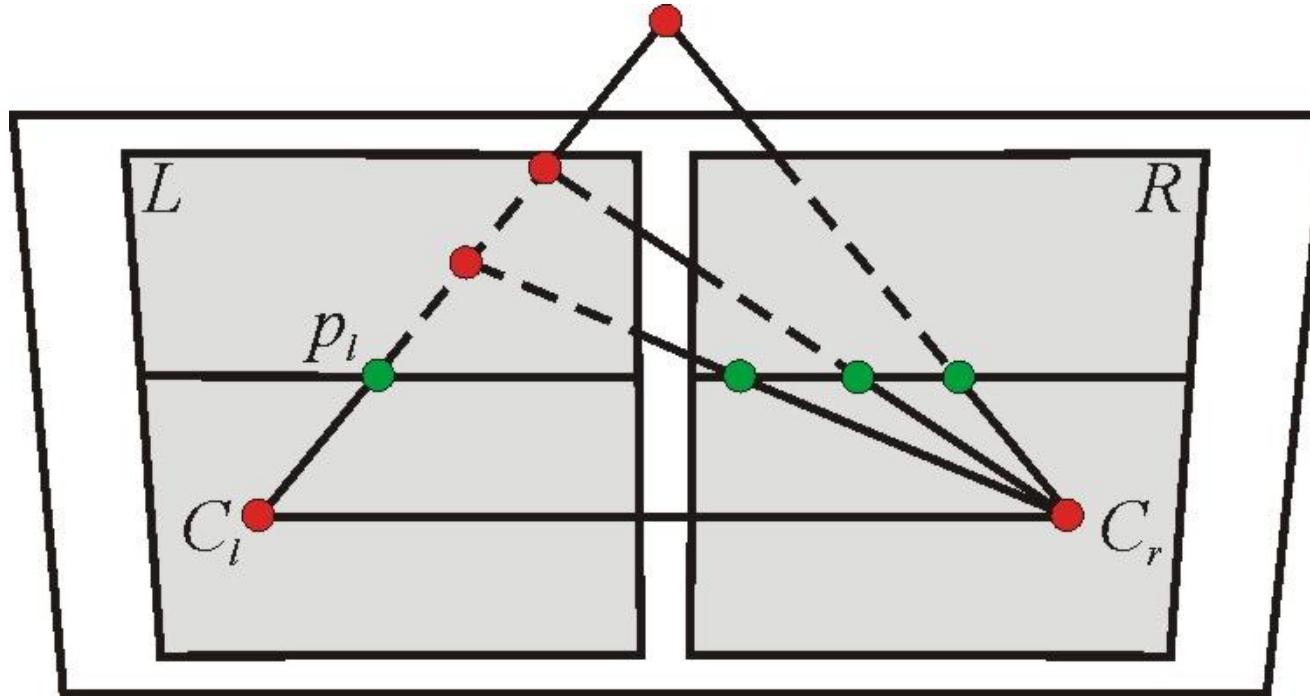
- 左图像L和右图像R位于同一平面
- L和R的x轴与基线平行，对应点具有相同的y坐标

立体视觉—极线校正



- 左图像L和右图像R位于同一平面
- L和R的x轴与基线平行，对应点具有相同的y坐标

立体视觉—极线校正



- P_l 在右图像上的极线平行于x轴
- 搜索 P_l 匹配点的过程只需在水平扫描线上进行

立体视觉—极线校正

- 可以通过设置虚拟摄像机位置进行极线校正
- 这一变换过程通过图像单应变换 $\mathbf{H}_{3 \times 3}$ 完成
- 单应 $\mathbf{H} = \mathbf{K}_2 \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{K}_1^{-1}$
- 在立体视觉中通常都假设极线已经校正



原始图像



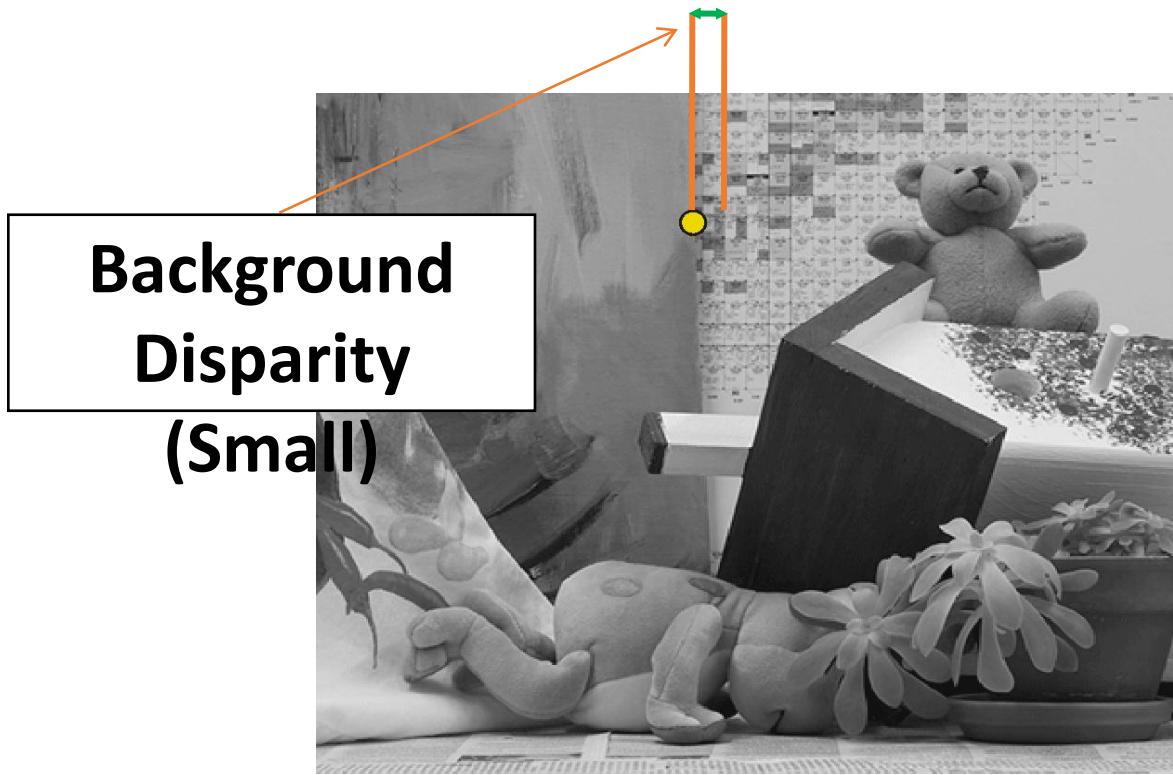
极线矫正后的图像

立体视觉—视差



- 极线校正后，左右图像的一对匹配点在x轴上坐标的差异称为**视差 (Disparity)**
- 视差的大小与点距离相机距离的远近成反比

立体视觉—视差



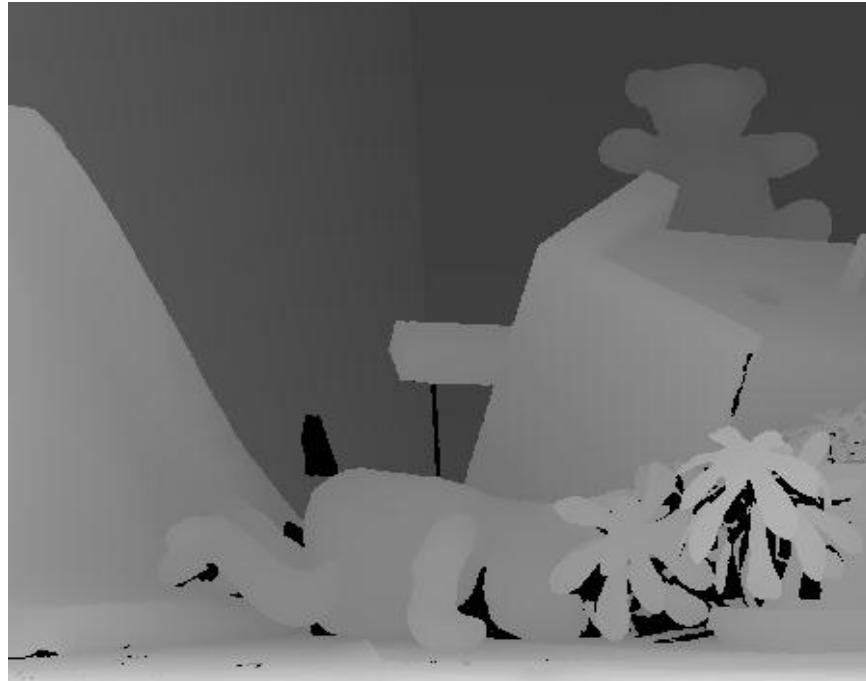
- 极线校正后，左右图像的一对匹配点在x轴上坐标的差异称为**视差 (Disparity)**
- 视差的大小与点距离相机距离的远近成反比

立体视觉—视差



- 极线校正后，左右图像的一对匹配点在x轴上坐标的差异称为**视差 (Disparity)**
- 视差的大小与点距离相机距离的远近成反比

立体视觉—视差图



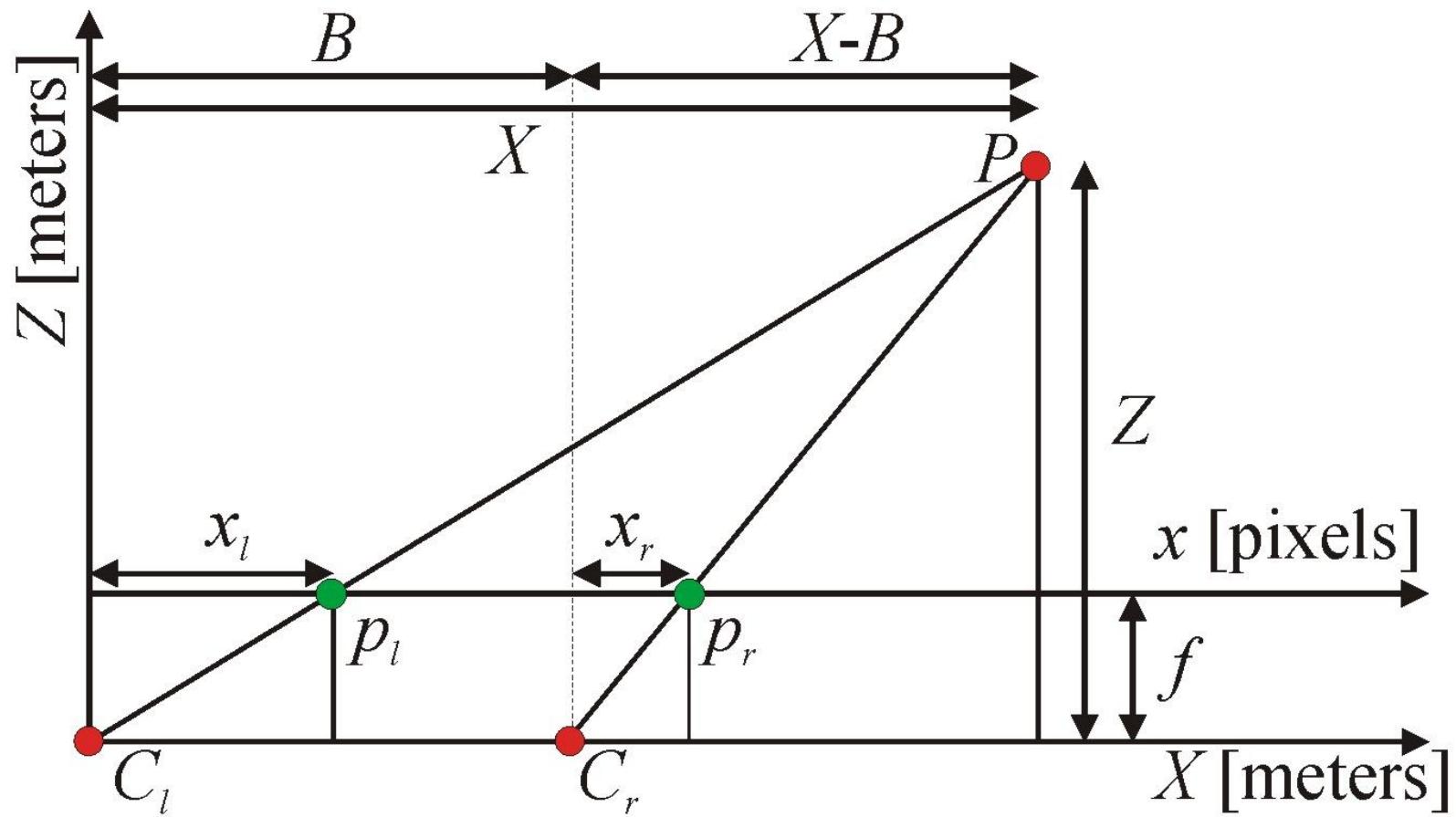
- 立体视觉的目的就是通过左右图像计算（稠密）视差图
- 视差图是一幅灰度图像，像素点的值表示这一点的视差大小，灰度值越高表示视差越大（距离越近）

立体视觉—视差图

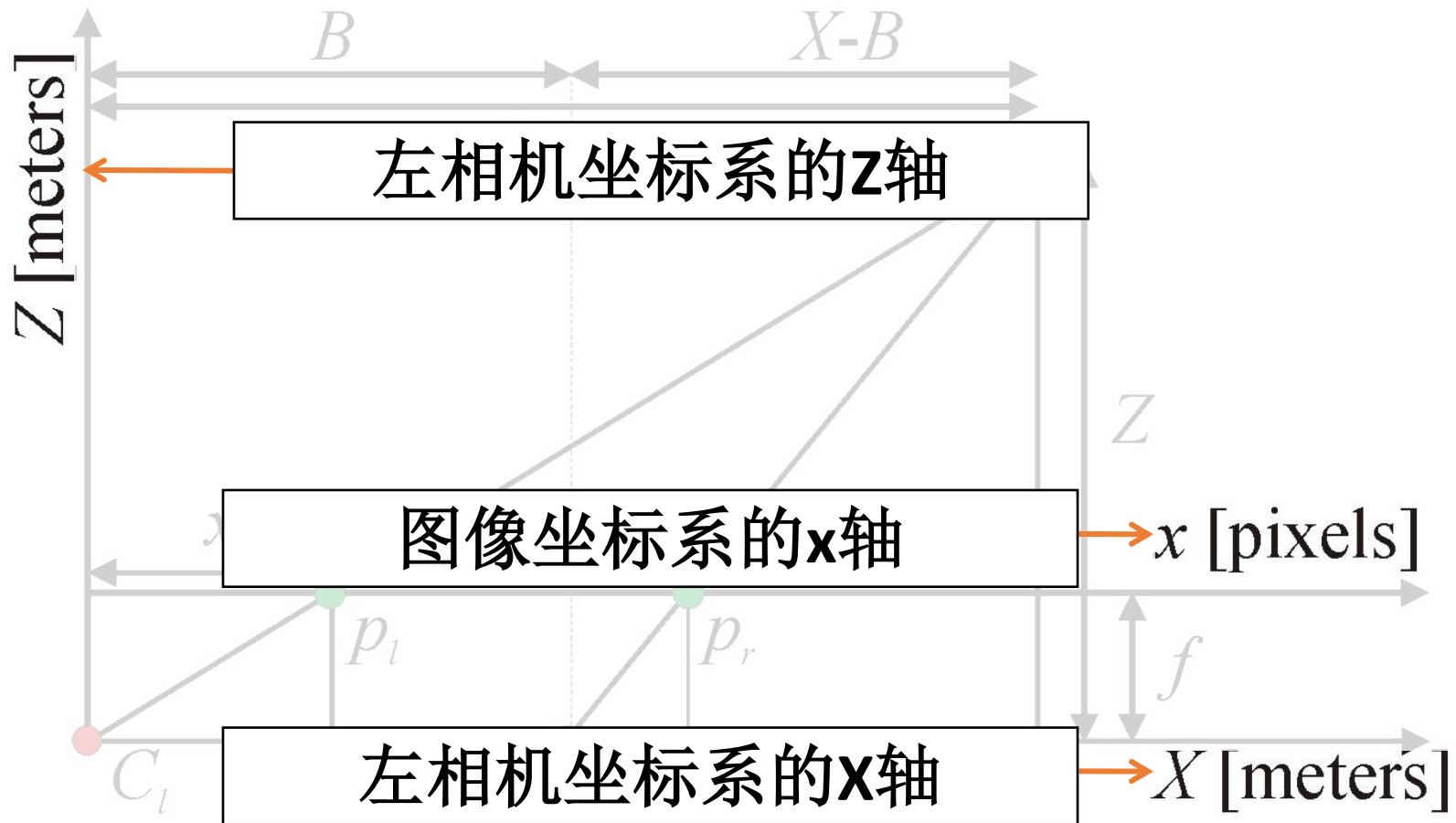


- 通过视差图可以得到单视点下的三维模型（稠密点云）

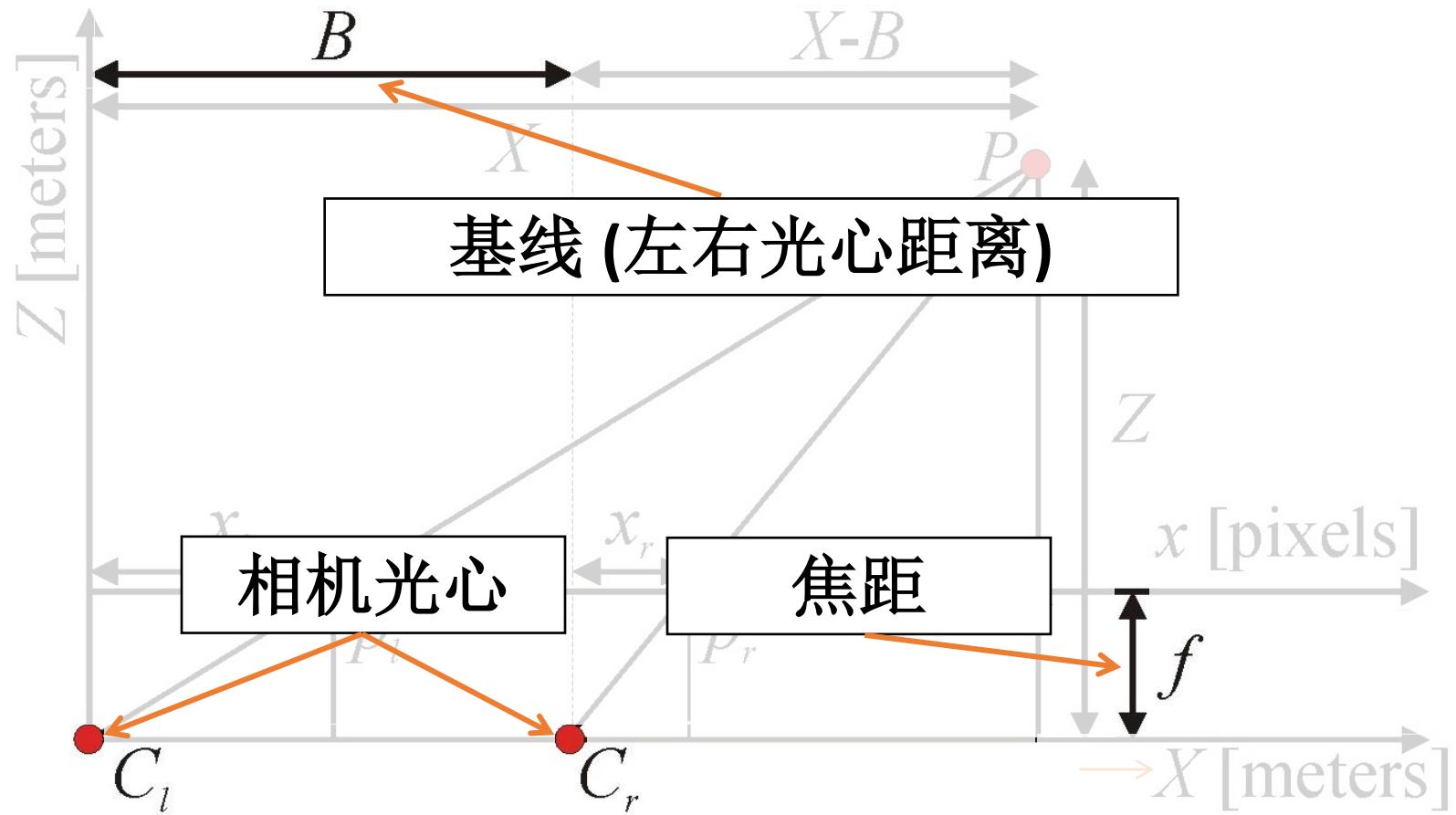
立体视觉—视差与深度



立体视觉—视差与深度

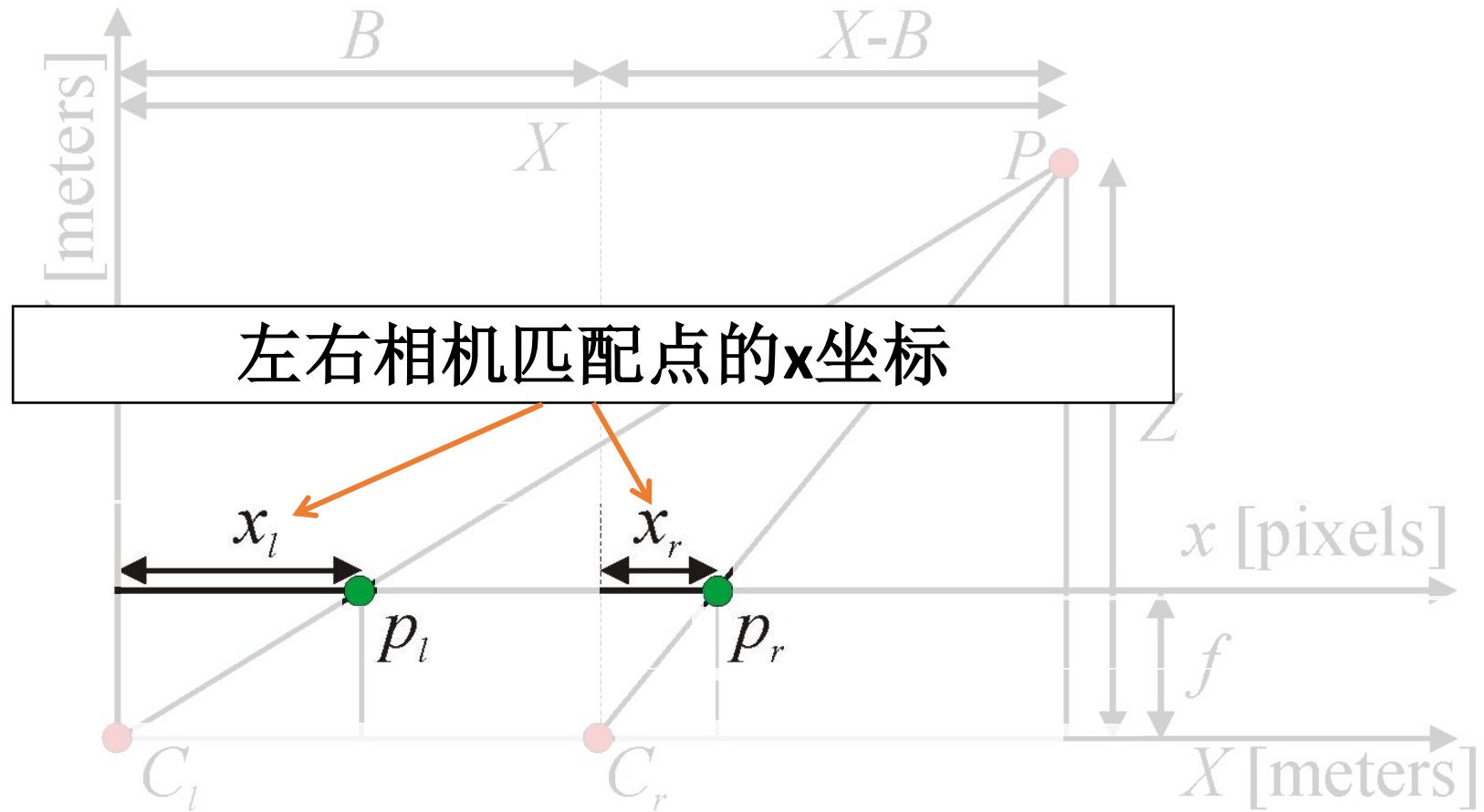


立体视觉—视差与深度

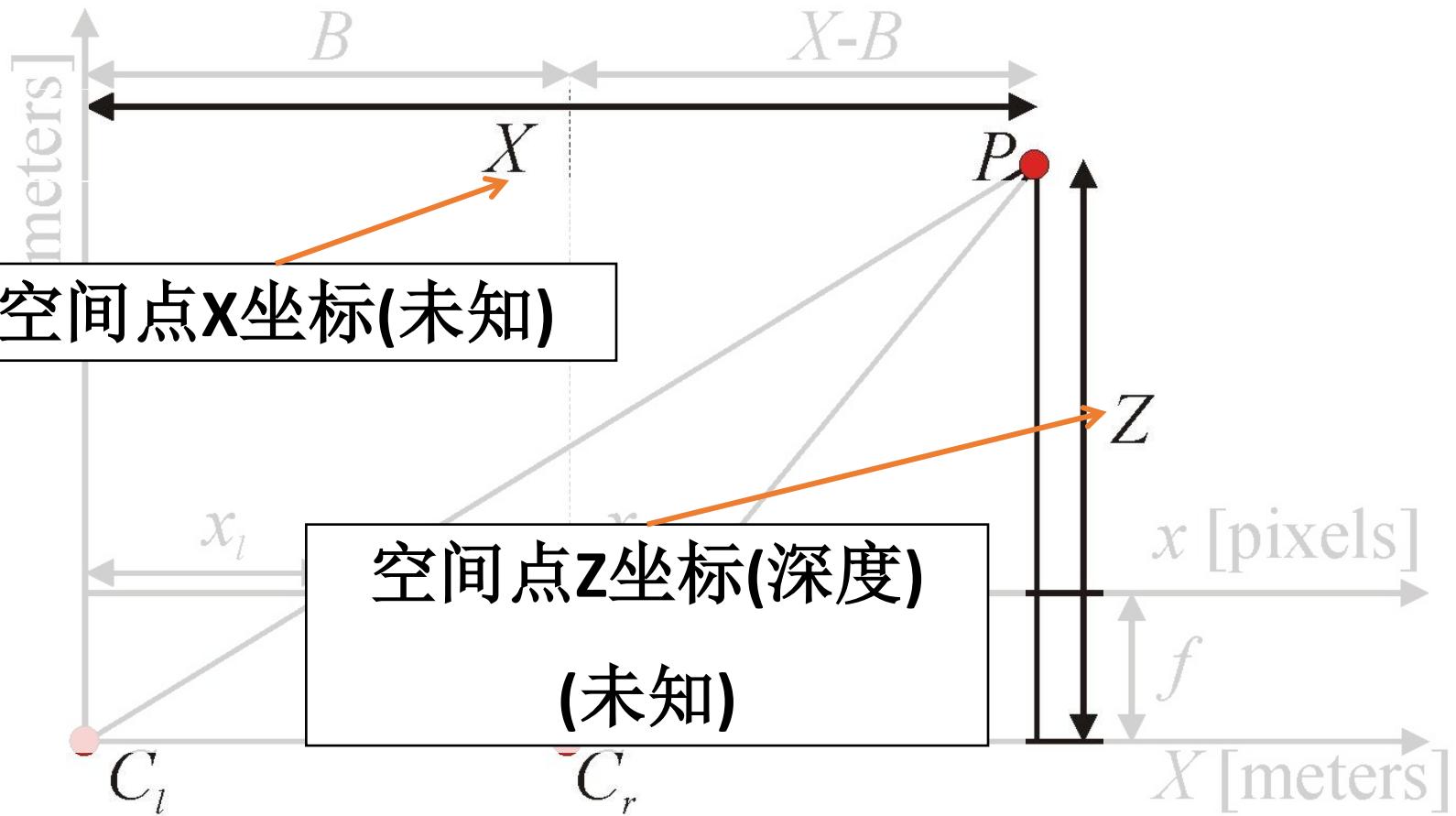


- 基线距离、相机光心位置、焦距均为已知量（相机标定）

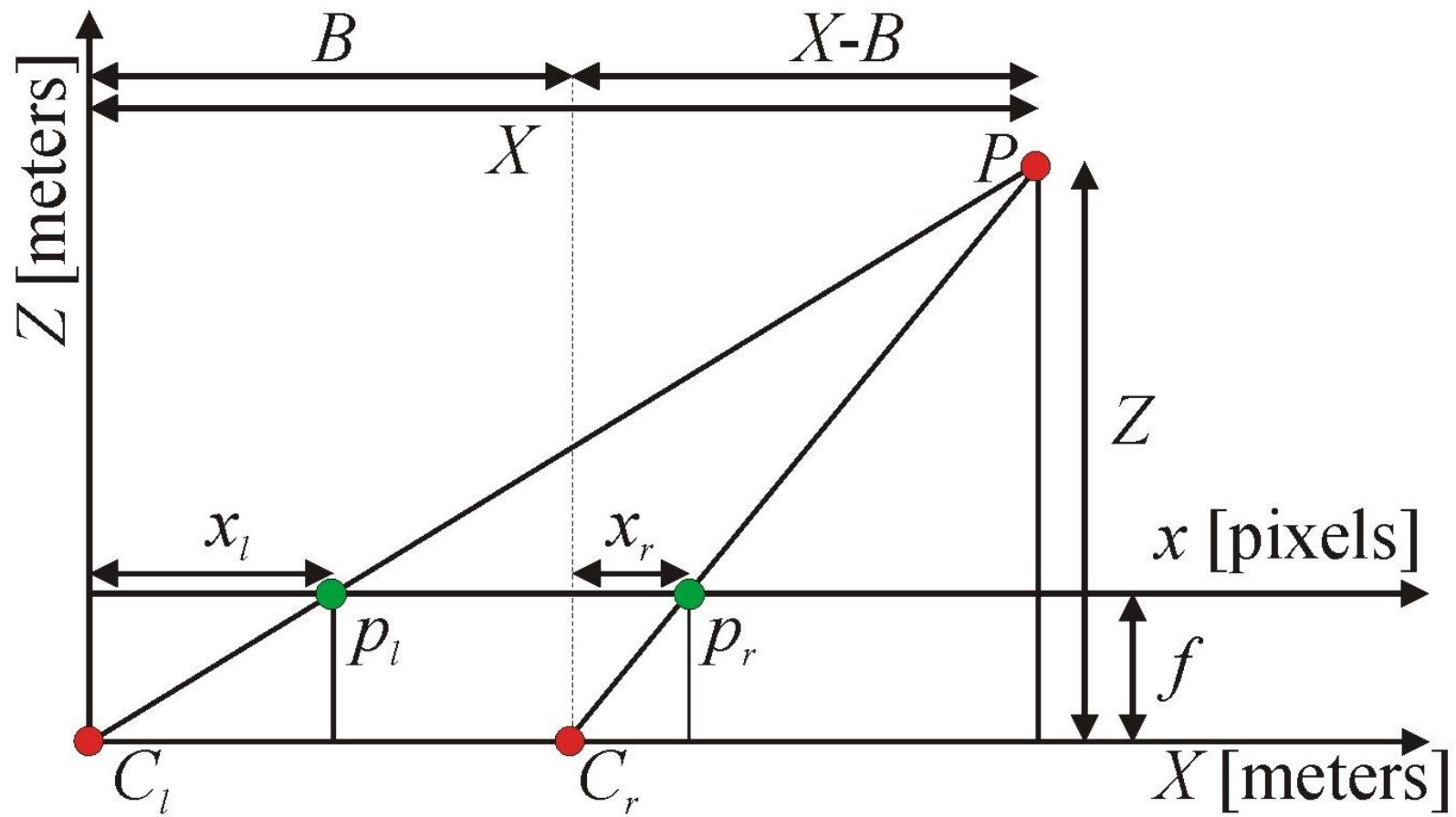
立体视觉—视差与深度



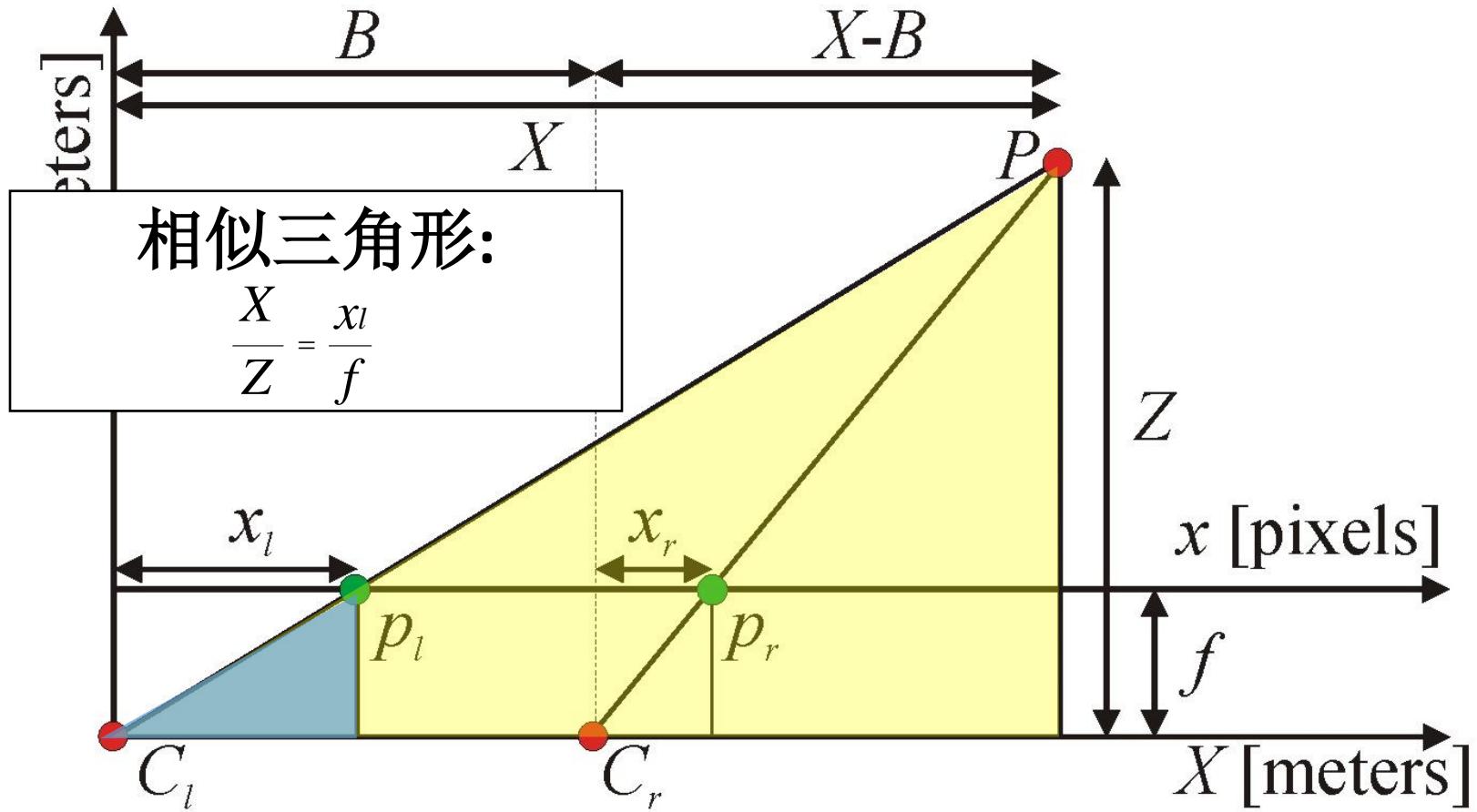
立体视觉—视差与深度



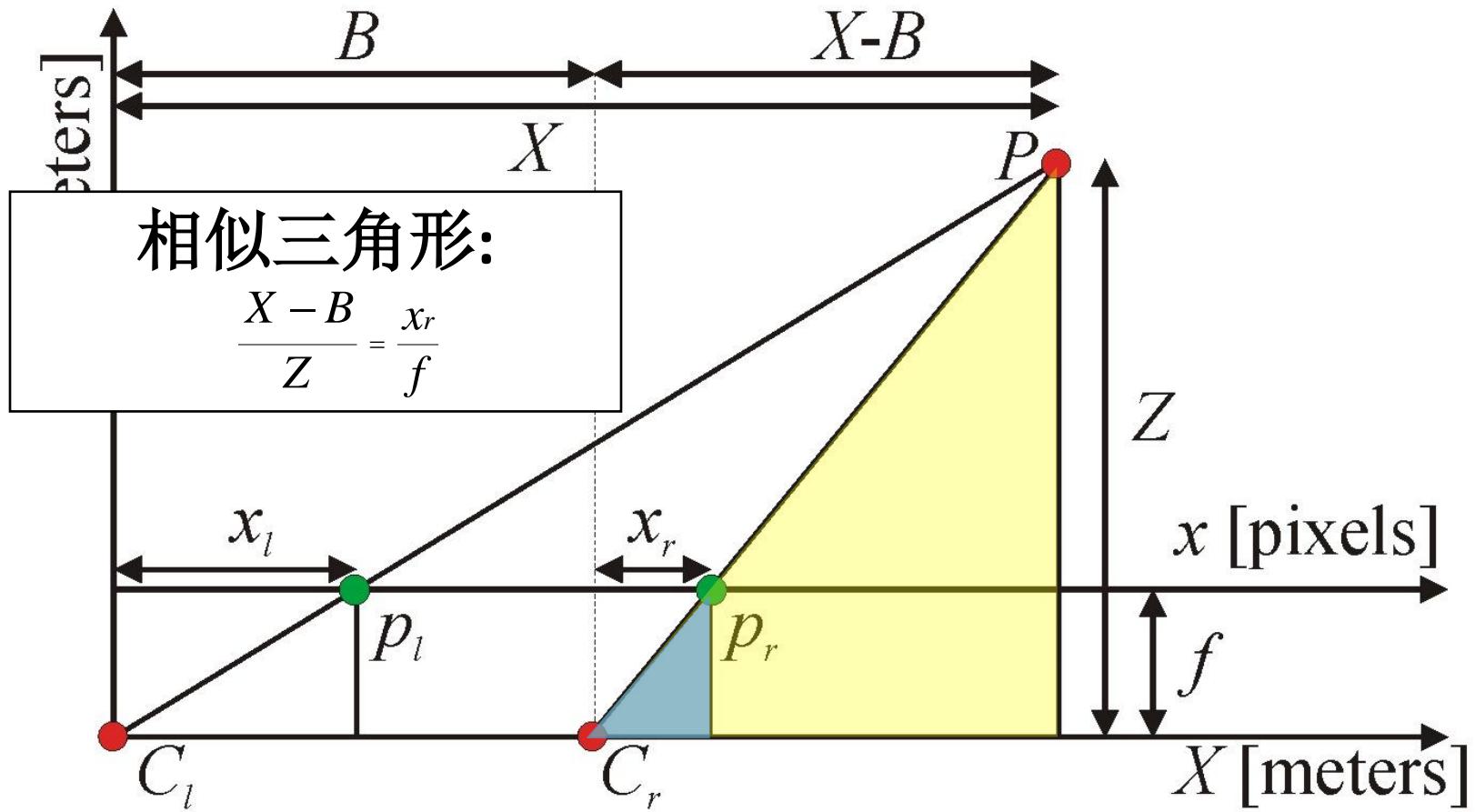
立体视觉—视差与深度



立体视觉—视差与深度



立体视觉—视差与深度



立体视觉—视差与深度

- 根据两个据相似三角形:

$$\frac{X}{Z} = \frac{x_l}{f}$$

$$\frac{X - B}{Z} = \frac{x_r}{f}$$

- X可以表达为:

$$X = \frac{Z \cdot x_l}{f}$$

$$X = \frac{Z \cdot x_r}{f} + B$$

- 将上述两个等式合并:

$$\frac{Z \cdot x_l}{f} = \frac{Z \cdot x_r}{f} + B$$

$$Z \cdot x_l = Z \cdot x_r + B \cdot f$$

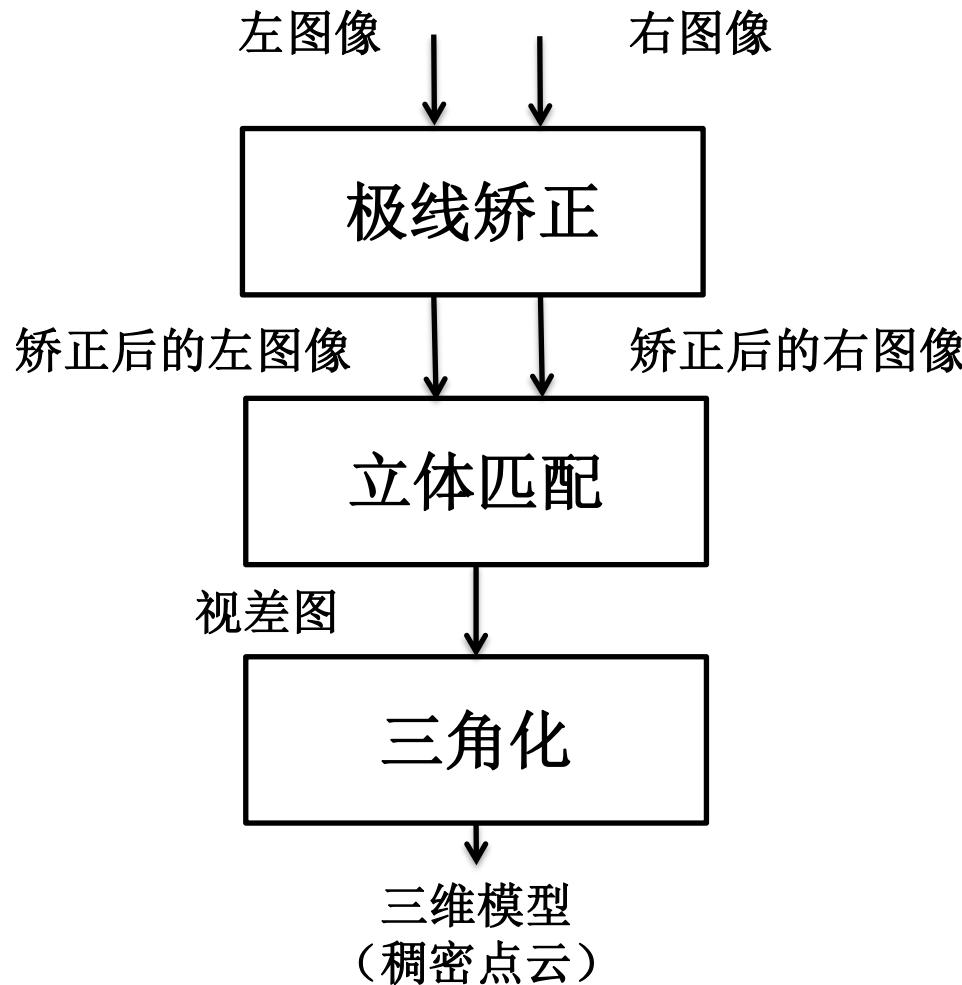
$$Z \cdot (x_l - x_r) = B \cdot f$$

- 深度Z可以表达为:

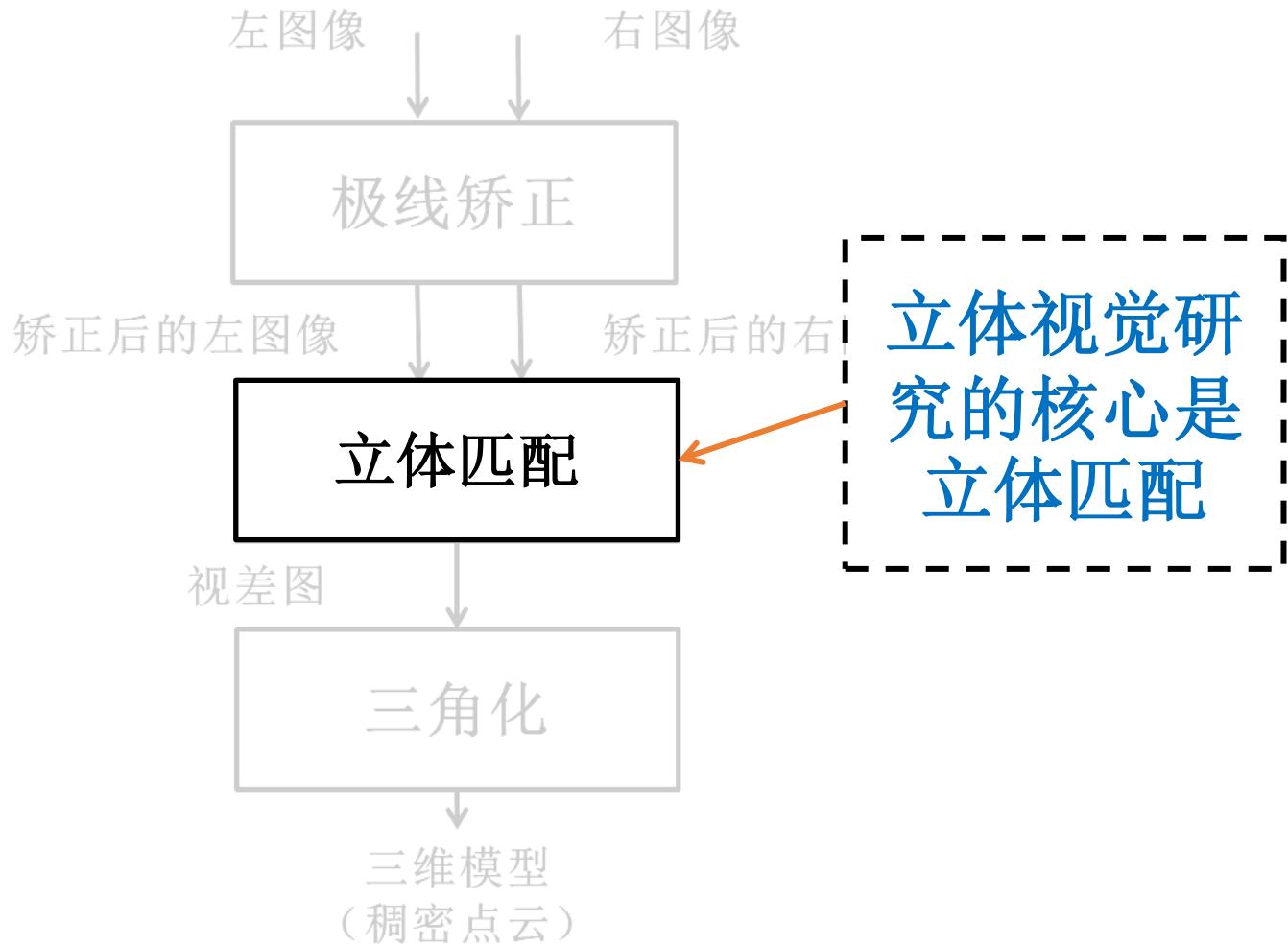
$$Z = \frac{B \cdot f}{x_l - x_r} = \frac{B \cdot f}{d}$$

视差

立体视觉—立体视觉三维建模流程



立体视觉—立体视觉三维建模流程



立体匹配—最简单的算法

- 对于左图像上每一点 p , 在右图极线上寻找对应点
- 极线上颜色最相似的点作为对应点:

$$d_p = \arg \min_{0 \leq d \leq d_{\max}} c(p, p - d)$$



(a) 左图像



(b) 右图像

立体匹配—最简单的算法



左图像



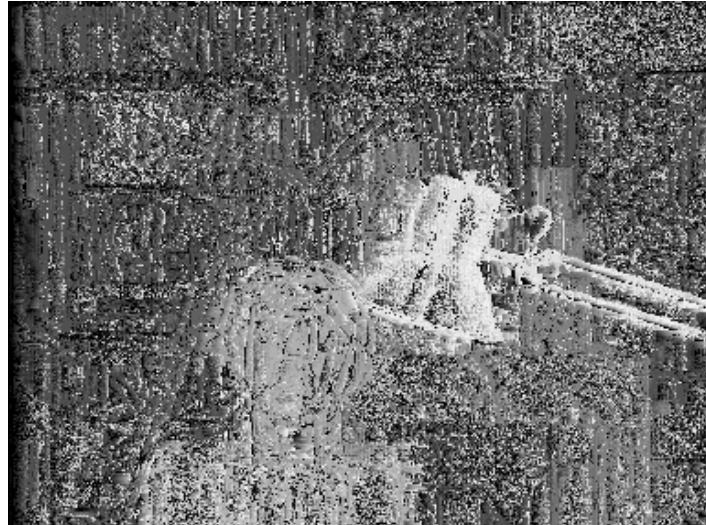
视差图

- 对于左图像上点 p , 右图极线上有很多颜色类似的点
- 仅靠单点颜色比较容易出现错误

立体匹配—平滑性约束



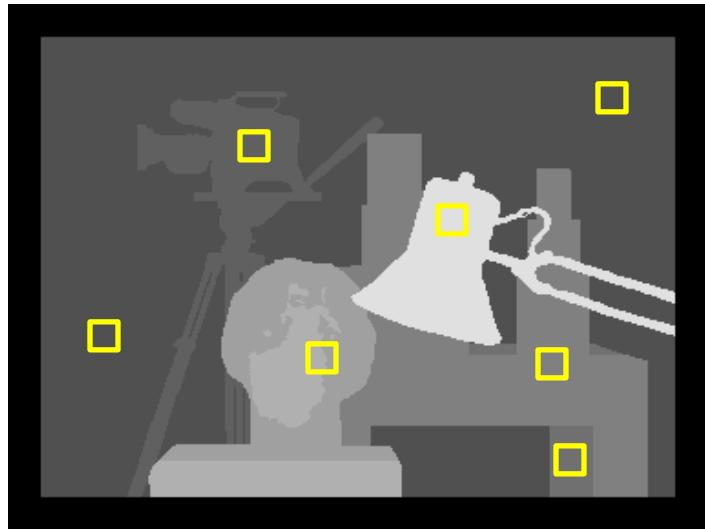
视差图真值



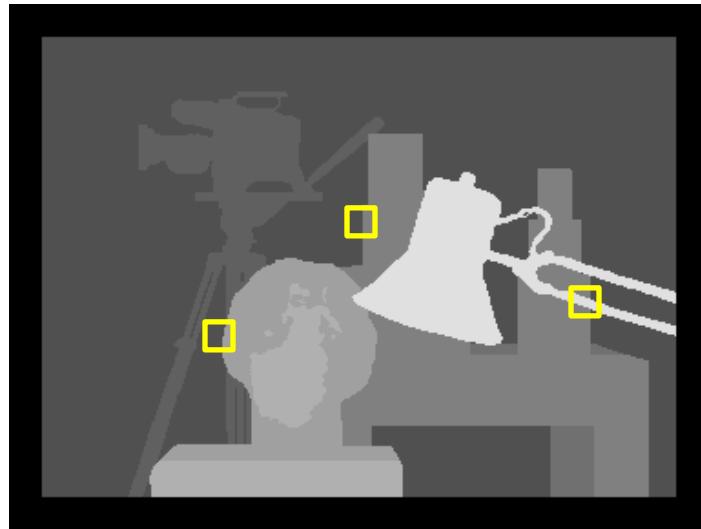
视差图

- 真值视差图中大多数区域是平滑的
- 需要在立体匹配中考虑平滑性约束

立体匹配—平滑性约束



满足平滑约束的区域



不满足平滑约束的区域

- 大多数物体表面都满足平滑性约束
- 在物体的边缘通常不满足平滑性约束

立体匹配—窗口匹配法

- 对于左图像上每一点 p , 在右图极线上寻找对应点
- 通过图像上一个正方形窗口区域衡量匹配程度



(a) 左图像



(b) 右图像

立体匹配—窗口匹配法

- 基于窗口相似度的视差计算：

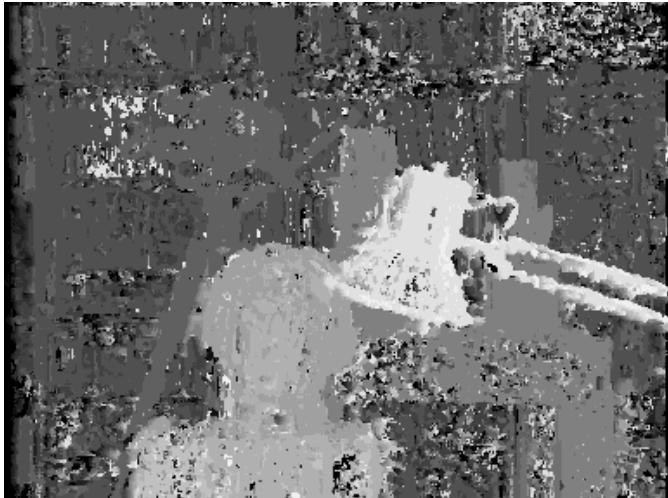
对于左图像上每一点 p , 其视差 d_p 计算如下

$$d_p = \arg \min_{0 \leq d \leq d_{\max}} \sum_{q \in Wp} c(q, q - d)$$

其中：

- d_{\max} 表示最大视差搜索范围；
- Wp 表示以 p 为中心的方窗内所有像素点的集合；
- $c(p,q)$ 是像素点 p 和 q 的相似度函数，通常定义为灰度差的绝对值或RGB差的绝对值和。

立体匹配—窗口匹配法

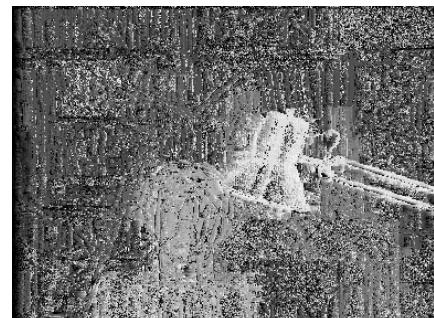


窗口尺寸= **3x3 pixels**



窗口尺寸= **21x21 pixels**

- 窗口尺寸直接影响视差图计算结果
- 窗口匹配法比单点比较法有改进
- 但结果仍然不理想



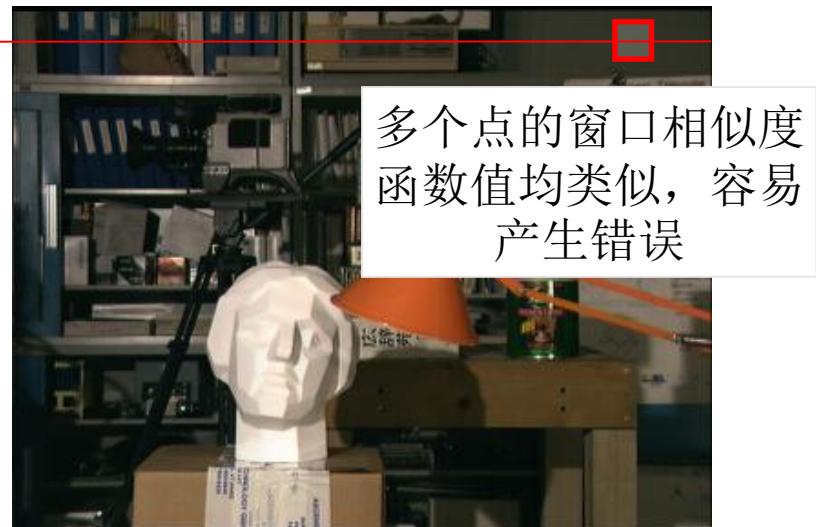
单点比较法

立体匹配—窗口匹配法

- 影响窗口匹配结果的因素一：弱纹理



(a) 左图像



(b) 右图像

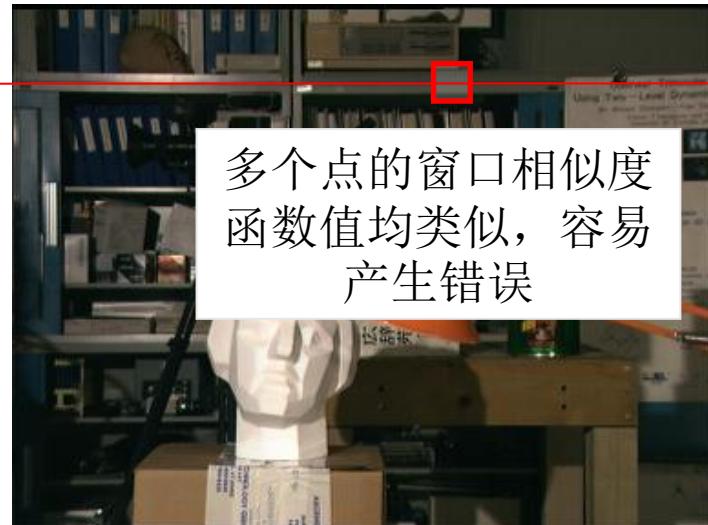
- 弱纹理区域窗口内像素值类似，相邻区域的相似度函数值差异不显著

立体匹配—窗口匹配法

- 影响窗口匹配结果的因素二：孔径问题（Aperture）



(a) 左图像

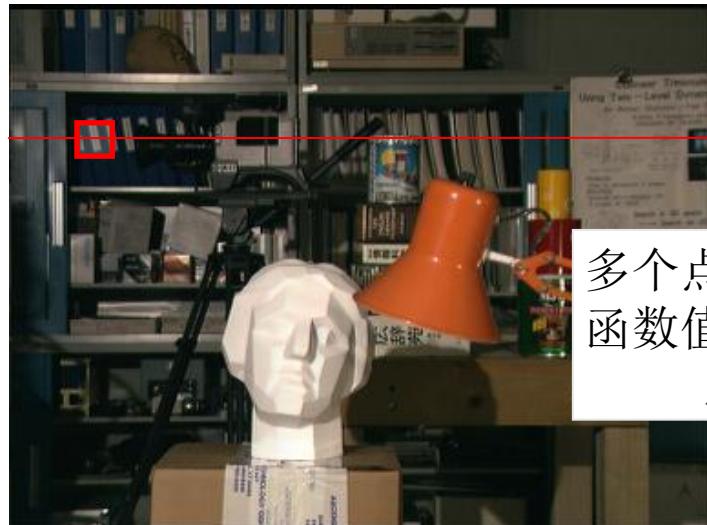


(b) 右图像

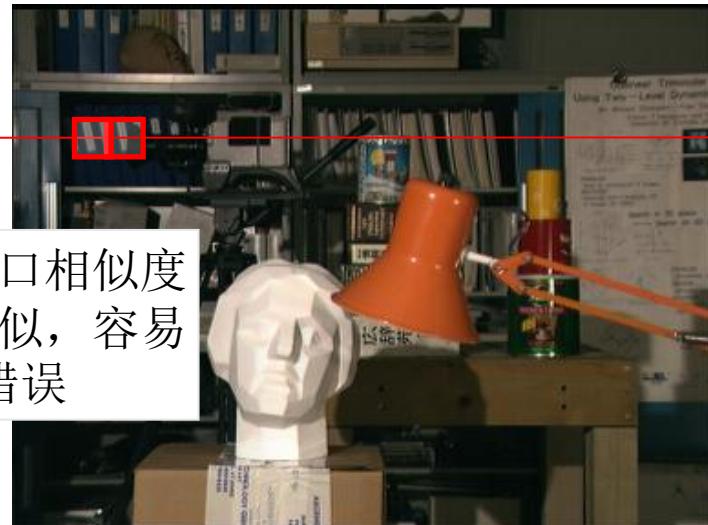
- 窗口包含的纹理区域太小，区分度不足

立体匹配—窗口匹配法

- 影响窗口匹配结果的因素三：重复纹理



(a) 左图像



(b) 右图像

多个点的窗口相似度
函数值均类似，容易
产生错误

- 重复纹理区域窗口内像素值类似，容易产生匹配误差

立体匹配—窗口匹配法

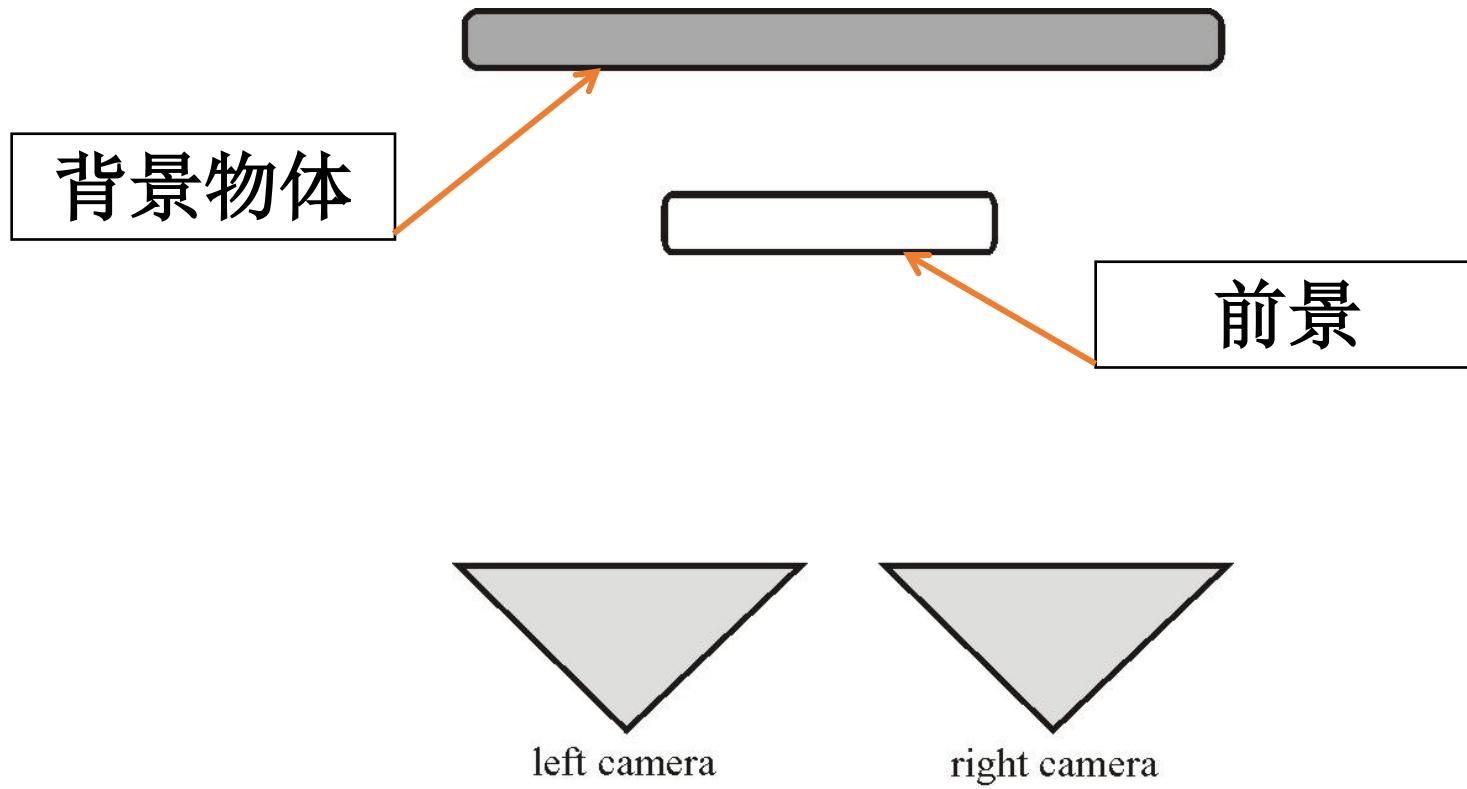
- 影响窗口匹配结果的因素



窗口尺寸= 3×3 pixels

立体匹配—窗口匹配法

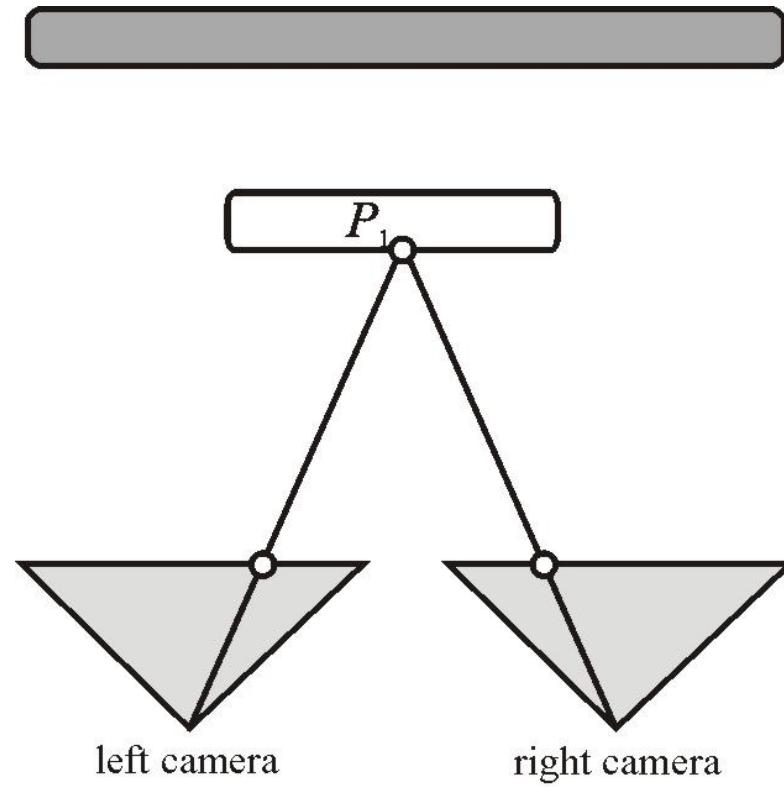
- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



立体匹配中存在遮挡问题

立体匹配—窗口匹配法

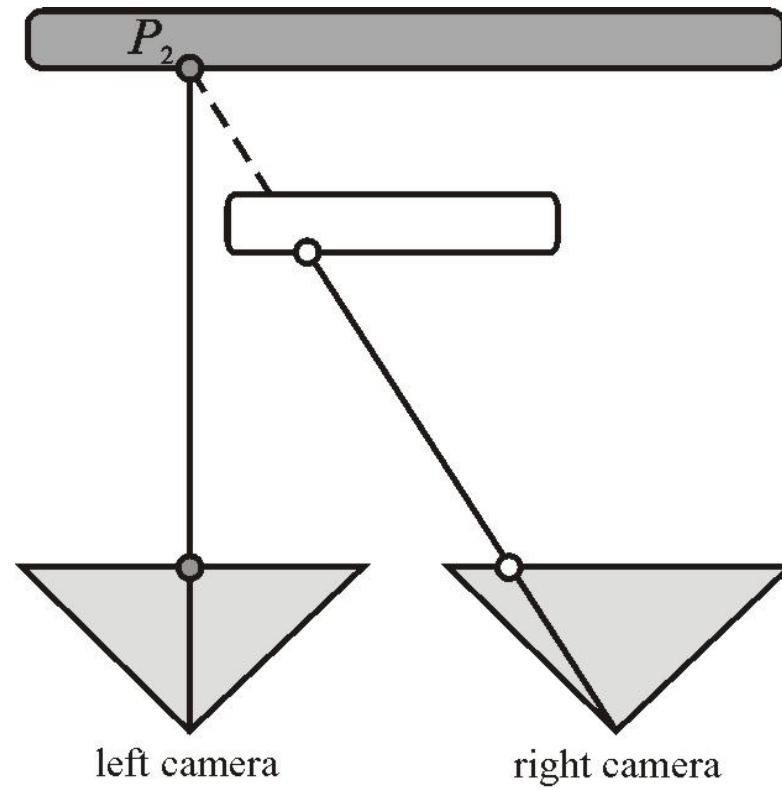
- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



立体匹配中存在遮挡问题

立体匹配—窗口匹配法

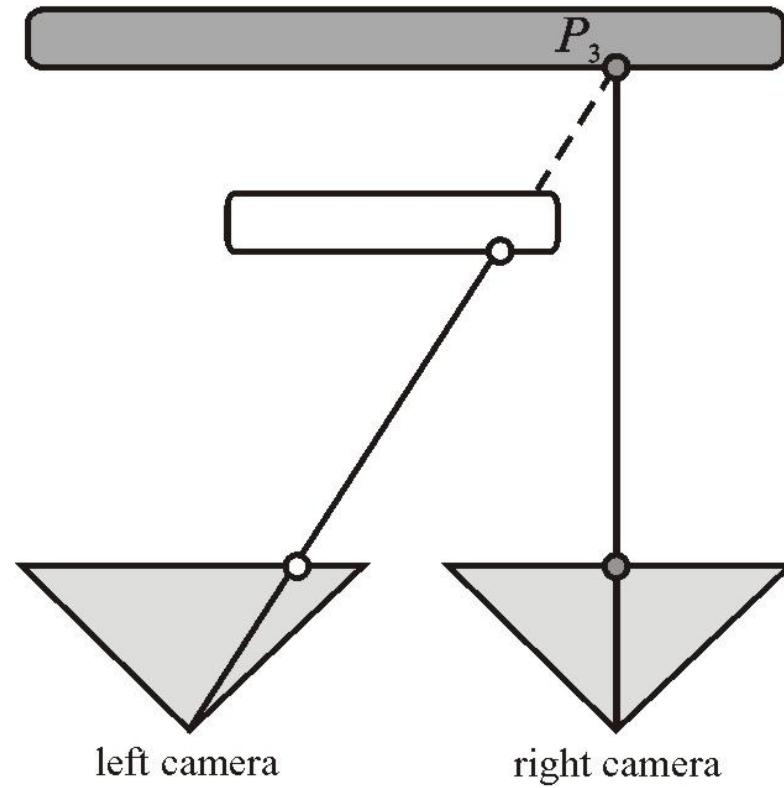
- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



立体匹配中存在遮挡问题

立体匹配—窗口匹配法

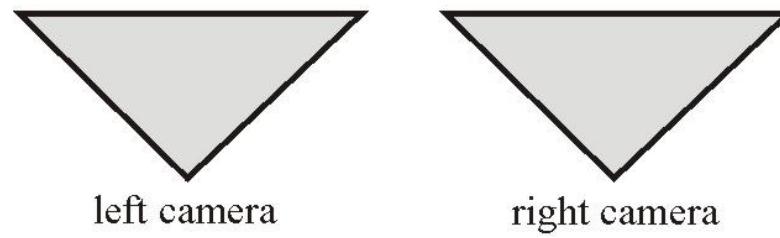
- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



立体匹配中存在遮挡问题

立体匹配—窗口匹配法

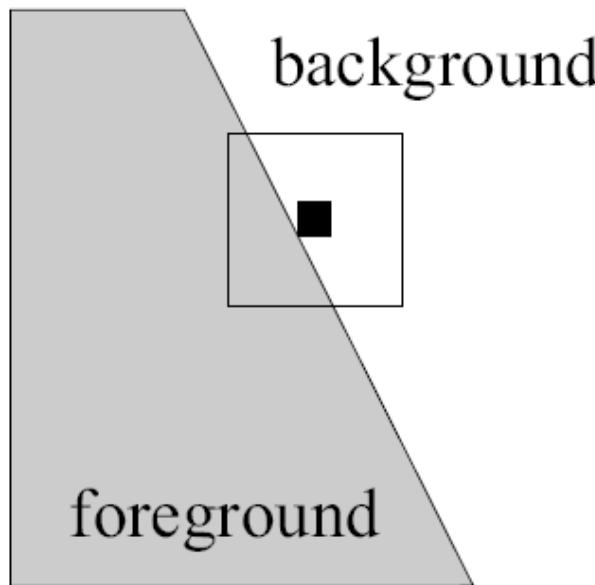
- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



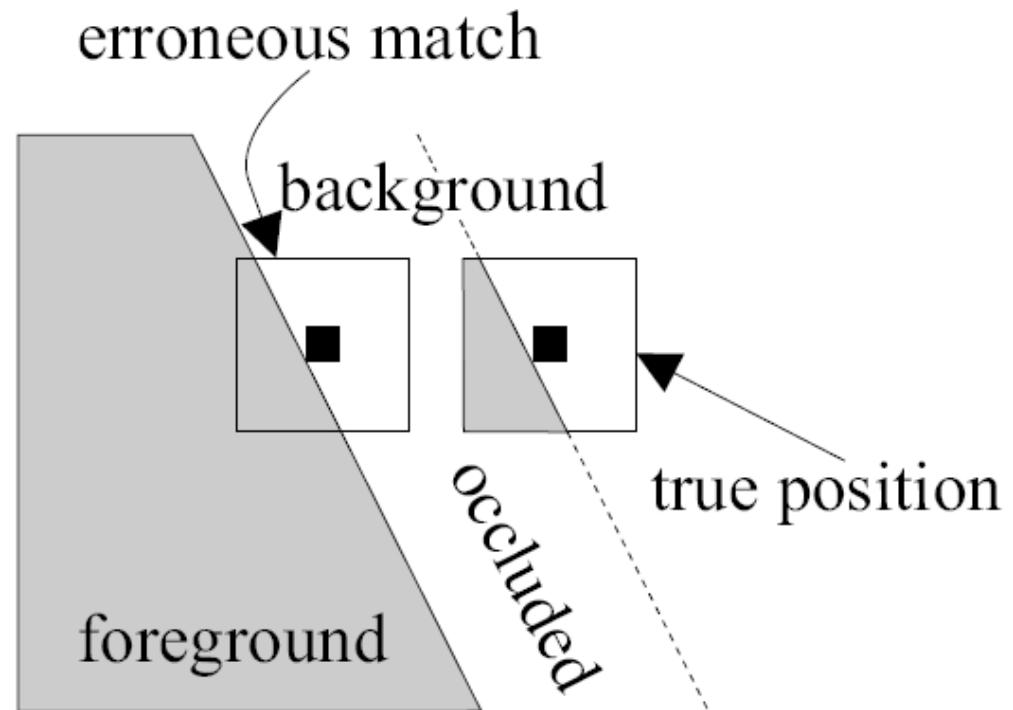
立体匹配中存在遮挡问题

立体匹配—窗口匹配法

- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



(a) 左图像



(b) 右图像

在边缘处背景容易被错分为前景，造成前景放大

立体匹配—窗口匹配法

- 影响窗口匹配结果的因素四：前景放大（Foreground Fattening）



真值视差图



窗口尺寸= **21x21 pixels**

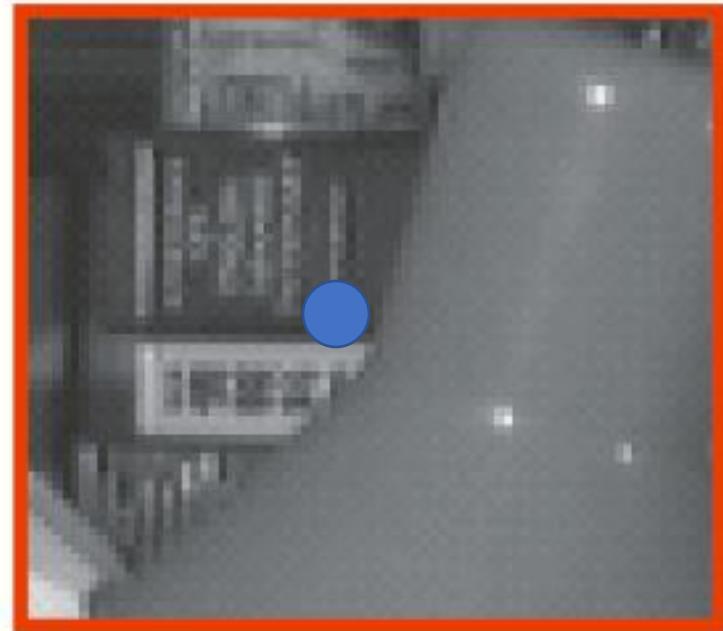
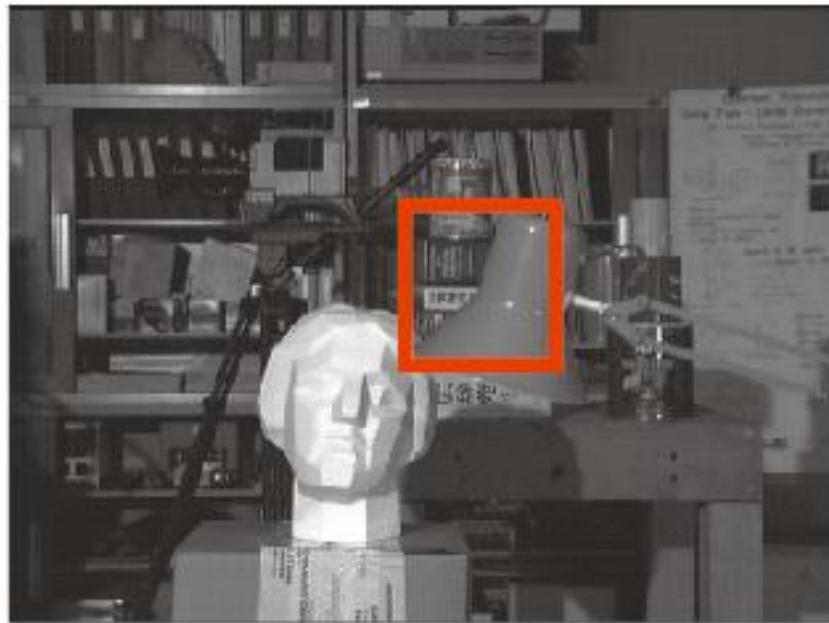
前景放大效应造成视差图中前景物体比实际大

立体匹配—窗口匹配法

- 窗口匹配法中窗口尺寸的选择:
 - 大尺寸窗口有利于解决:
 - 弱纹理;
 - 孔径问题;
 - 重复纹理;
 - 小尺寸窗口有利于解决:
 - 前景放大;
 - 窗口尺寸选择依据:
 - 没有最优窗口尺寸能够解决所有问题。

立体匹配—自适应窗口匹配法

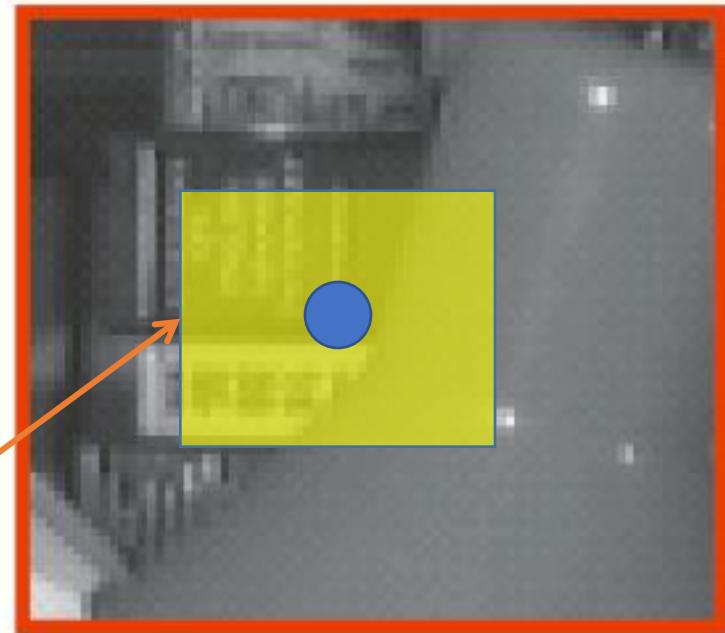
- 融合大窗口和小窗口的各自优势



位于边缘（视差跳变）区域的窗口不利于匹配

立体匹配—自适应窗口匹配法

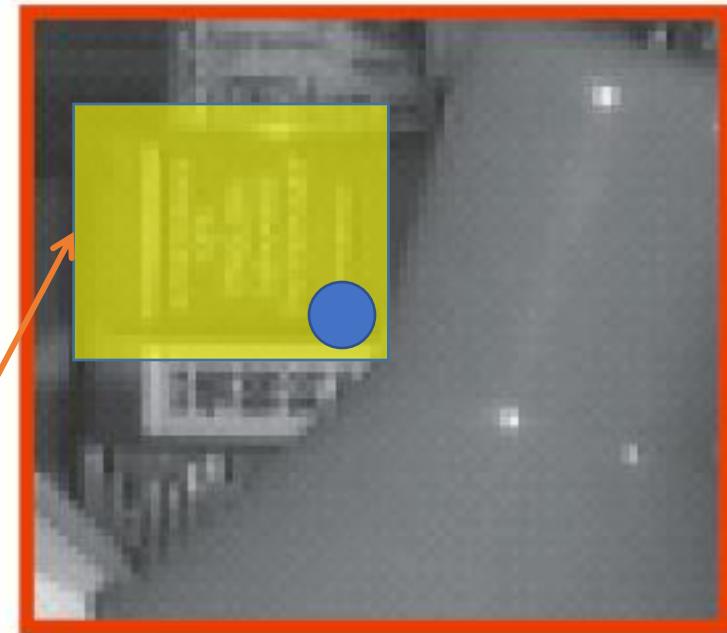
- 融合大窗口和小窗口的各自优势



位于边缘（视差跳变）区域的窗口不利于匹配

立体匹配—自适应窗口匹配法

- 融合大窗口和小窗口的各自优势



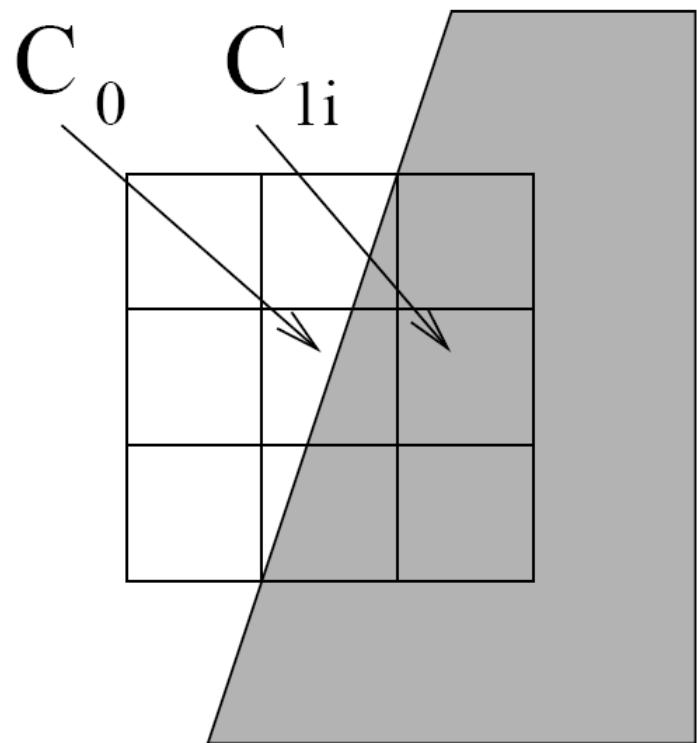
位于边缘（视差跳变）区域的窗口不利于匹配

立体匹配—自适应窗口匹配法

- 融合大窗口和小窗口的各自优势

自适应窗口：

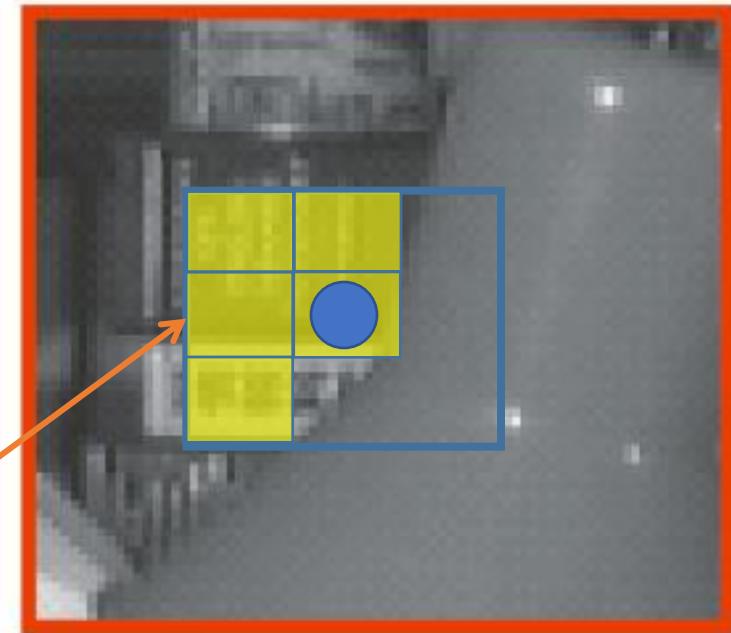
- 将窗口分为9个子窗口；
- 在每个子窗口中分别计算相似度；
- 在相似度最高的5个子窗口上计算相似度度量。



[Hirschmueller, IJCV02]

立体匹配—自适应窗口匹配法

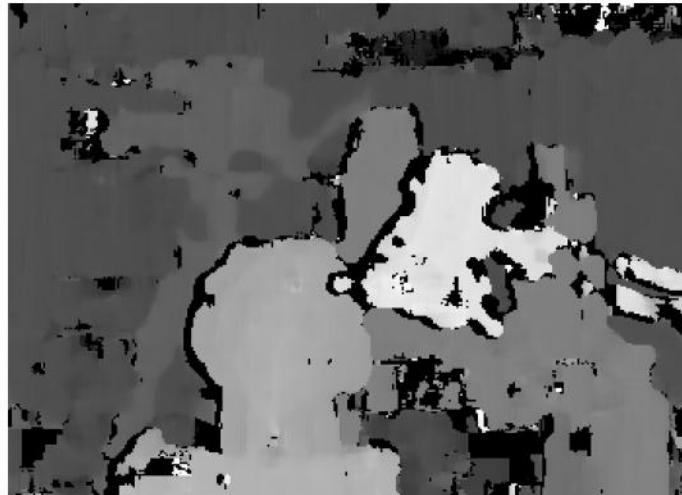
- 融合大窗口和小窗口的各自优势



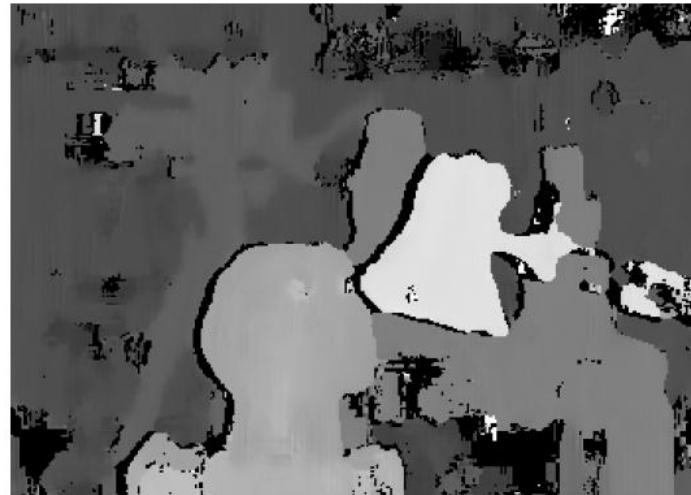
自适应窗口匹配法：在相似度最高的5个子窗口上计算相似度度量

[Hirschmueller, IJCV02]

立体匹配—自适应窗口匹配法



标准窗口匹配



自适应窗口匹配

- 自适应窗口匹配可以有效改进边缘区域的视差
- 但结果仍然不够理想

[Hirschmueller, IJCV02]

立体匹配—自适应权值窗口匹配法

- 对于左图像上每一点 p , 其视差 d_p 计算如下:

$$d_p = \arg \min_{0 \leq d \leq d_{\max}} \sum_{q \in W_p} w(p, q) \cdot c(q, q - d)$$

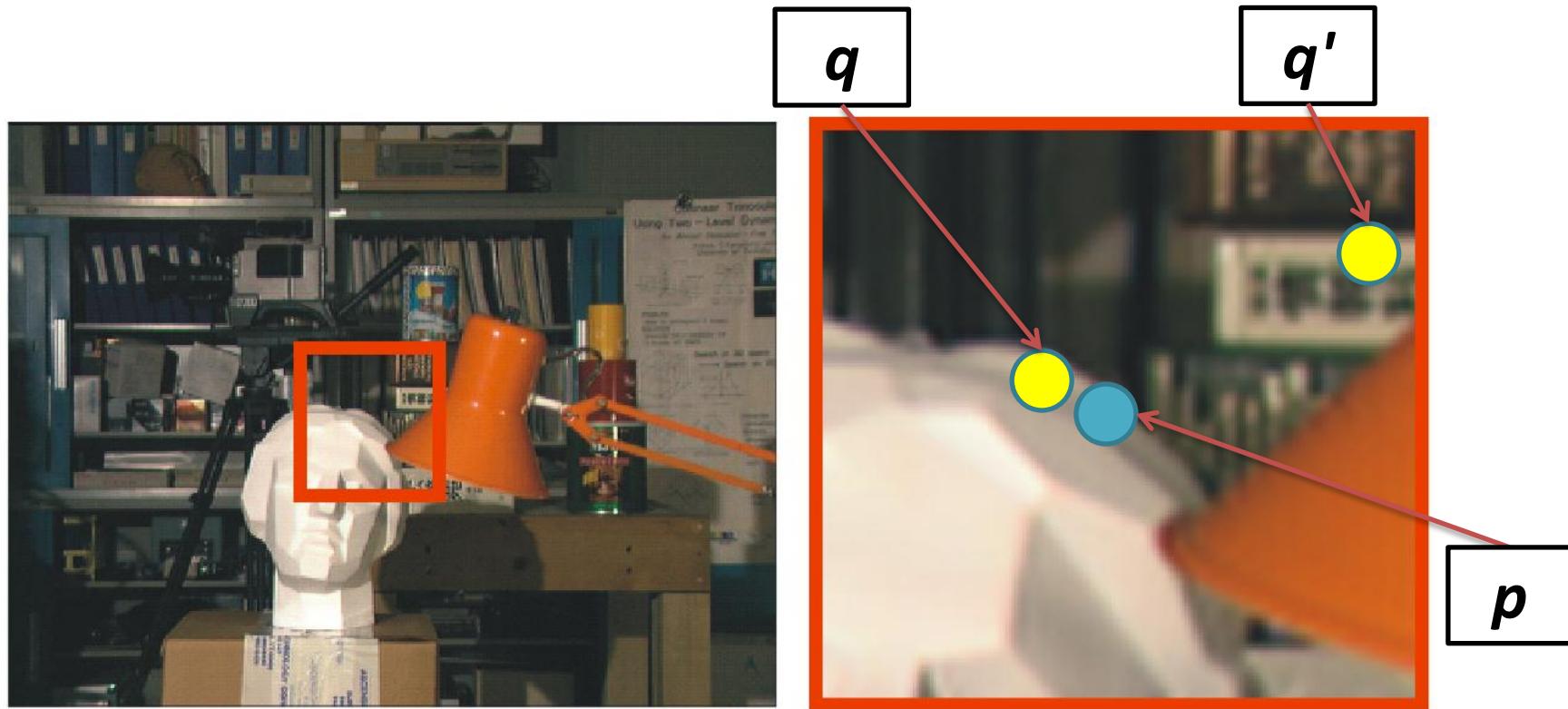
其中:

- $w(p, q)$ 表示点 p 和 q 视差相同的可能性;
- 理想情况下:

$$w(p, q) = \begin{cases} 1 & p \text{ 和 } q \text{ 视差相同} \\ 0 & p \text{ 和 } q \text{ 视差不同} \end{cases}$$

立体匹配—自适应权值窗口匹配法

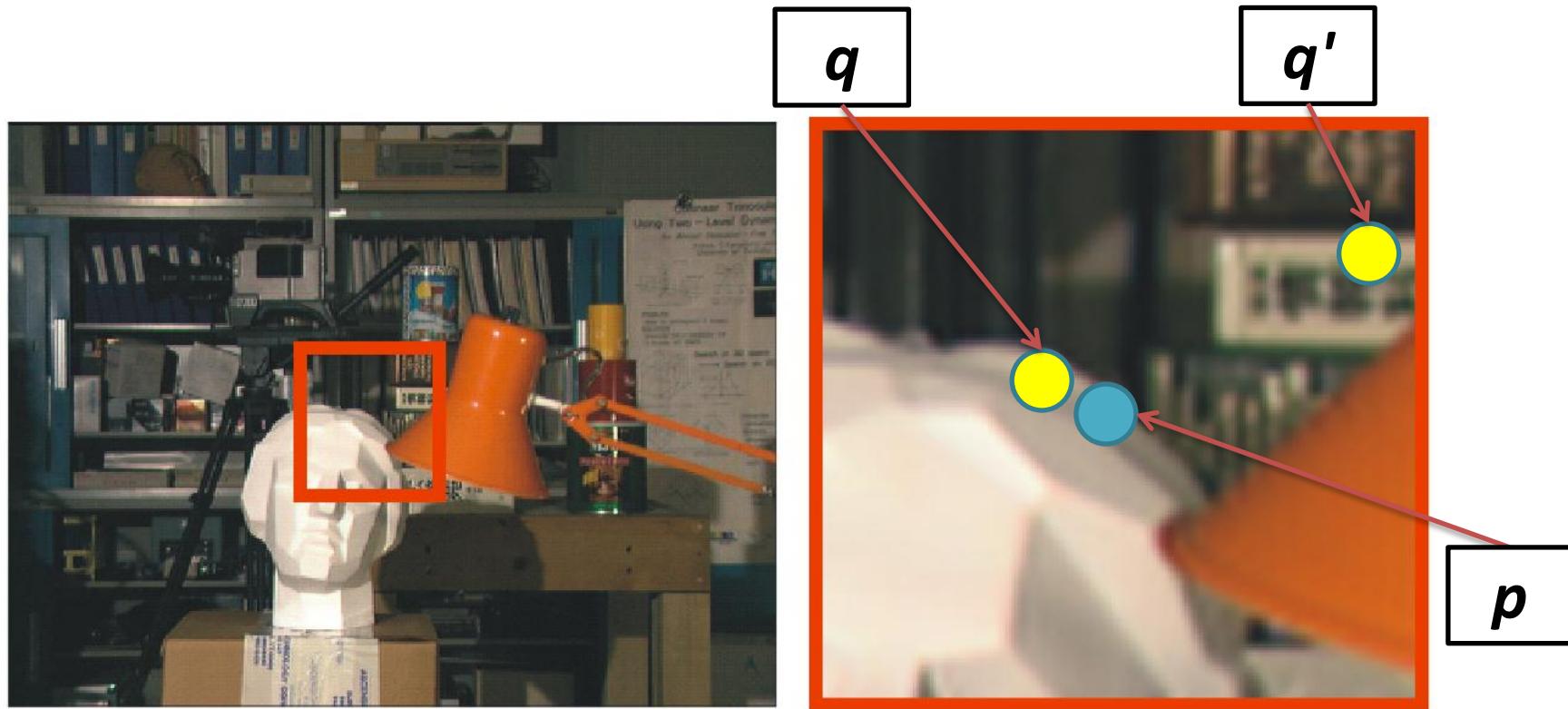
- 如何判断视差相同的可能性（我们只有图像）：



根据颜色：颜色相似的点视差相同的可能性大

立体匹配—自适应权值窗口匹配法

- 如何判断视差相同的可能性（我们只有图像）：



根据距离：在窗口内距离近的点视差相同的可能性大

立体匹配—自适应权值窗口匹配法

- 对于左图像上每一点 p , 其视差 d_p 计算如下:

$$d_p = \arg \min_{0 \leq d \leq d_{\max}} \sum_{q \in W_p} w(p, q) \cdot c(q, q - d)$$

其中:

$$w(p, q) = \exp \left(- \left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_g} \right) \right)$$

- Δc_{pq} 表示点 p 和 q 的颜色差异 (RGB差的绝对值和)
- Δg_{pq} 表示点 p 和 q 的距离差异 (图像欧氏距离)
- γ_c 和 γ_g 为人工指定参数

[Yoon, CVPR05]

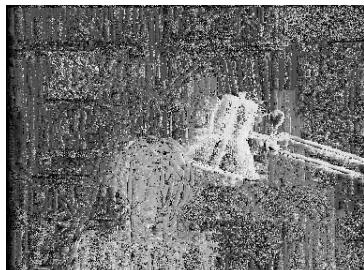
立体匹配—自适应权值窗口匹配法



真值视差图



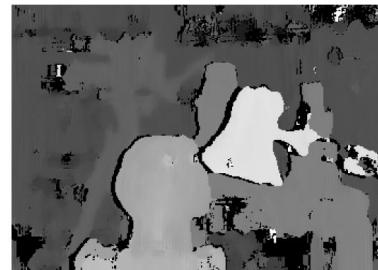
自适应权值窗口匹配



单点比较



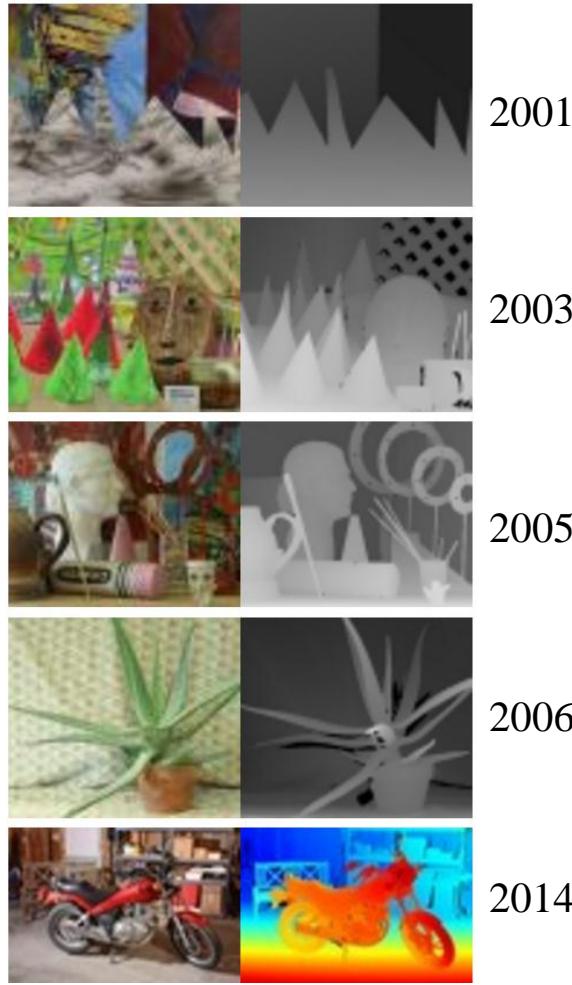
标准窗口



自适应窗口

[Yoon, CVPR05]

立体视觉—标准评测库 (Middlebury)



Middlebury Stereo Evaluation - Version 3

ELAS update

Snapshot of 9/14/2015 before updating ELAS results. The old ELAS results were worse because of a bug in the conversion to grayscale images (using ImageMagick's "convert" under linux). The conversion is now done via the code in the SDK 1.6.

See other snapshots [here](#).

Set: [test dense](#) [test sparse](#) [training dense](#) [training sparse](#)

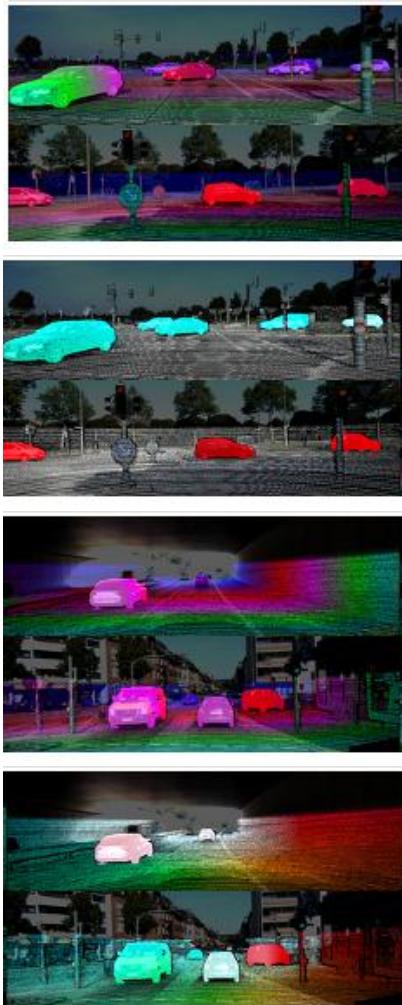
Metric: [bad 0.5](#) [bad 1.0](#) [bad 2.0](#) [bad 4.0](#) [avgerr](#) [rms](#) [A50](#) [A90](#) [A95](#) [A99](#) [time](#) [time/MP](#) [time/GD](#)

Mask: [nonocc](#) [all](#)

plot selected show invalid [Reset sort](#)

Date	Name	Res	Weight	Austr	AustrP	Bicyc2	Class	Compu	Crusa	CrusaP	Djemb	DjembP	Hoops	Livgrm	Nkuba	Plants	Stairs	
				MP: 5.6 nd: 290 im0 im1 GT nonocc	MP: 5.6 nd: 290 im0 im1 GT nonocc	MP: 5.6 nd: 610 im0 im1 GT nonocc	MP: 5.7 nd: 610 im0 im1 GT nonocc	MP: 5.5 nd: 256 im0 im1 GT nonocc	MP: 5.5 nd: 800 im0 im1 GT nonocc	MP: 5.7 nd: 320 im0 im1 GT nonocc	MP: 5.7 nd: 410 im0 im1 GT nonocc	MP: 5.7 nd: 320 im0 im1 GT nonocc	MP: 5.9 nd: 570 im0 im1 GT nonocc	MP: 5.5 nd: 320 im0 im1 GT nonocc	MP: 5.2 nd: 450 im0 im1 GT nonocc			
				↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑		
08/28/15	MC-CNN	H	8.291	5.59 1	4.55 4	5.96 1	2.83 1	11.4 2	8.44 1	8.32 1	8.89 1	2.71 1	16.3 1	14.1 1	13.2 1	13.0 1	6.40 1	11.1 1
04/19/15	MeshStereo	H	13.42	5.90 2	4.88 5	10.8 9	12.9 6	10.6 1	13.6 3	12.2 4	9.01 2	5.39 4	27.4 3	23.5 3	17.7 2	21.0 6	15.4 6	20.9 3
11/12/14	LCU	Q	17.03	24.7 6	7.59 10	11.6 10	11.9 4	27.9 3	14.0 4	19.3 5	15.8 7	8.10 14	36.1 9	29.1 7	21.3 5	18.4 2	14.1 3	23.8 5
04/17/14	TMAP	H	17.14	20.2 5	4.94 6	8.13 5	12.8 5	30.0 4	14.1 6	27.9 10	20.4 11	5.09 2	31.5 7	23.1 2	20.9 4	19.0 3	18.8 10	18.0 2
10/07/14	IDR	H	18.45	37.5 12	4.08 1	7.49 3	23.3 13	40.6 7	15.7 13	24.5 6	11.3 6	5.46 8	33.1 8	26.0 4	21.5 6	21.7 7	15.3 5	21.2 4
07/28/14	SGM	H	18.76	40.3 13	4.54 3	8.03 4	22.9 12	40.5 6	14.6 9	24.7 7	10.1 4	5.40 5	29.6 5	28.5 6	23.9 8	20.0 4	14.2 4	30.9 9
08/25/14	LPS	H	19.47	6.14 3	5.34 7	9.24 6	7.53 2	96.0 21	15.0 11	9.61 2	9.40 3	5.18 3	92.4 23	27.4 5	24.3 9	23.0 10	10.0 2	25.6 7
08/27/14	LPS	F	20.38	6.72 4	6.06 8	9.72 7	9.87 3	94.3 20	14.1 15	11.2 3	11.2 5	5.88 8	89.3 22	36.0 11	20.5 3	23.8 11	16.0 7	25.4 6
07/25/14	SGM	Q	21.29	35.5 11	9.57 14	13.8 16	16.5 10	32.1 5	23.3 20	25.8 6	16.7 9	8.95 16	39.8 13	31.1 8	22.6 7	20.7 5	21.3 14	32.2 10
09/18/14	SNCC	H	22.20	48.6 18	6.98 9	9.79 8	25.7 15	46.0 10	15.2 12	36.8 14	16.6 8	7.25 11	23.1 2	34.2 9	26.7 13	21.8 8	19.9 12	28.4 8
07/28/14	SGBM1	H	24.011	32.9 6	10.8 15	13.6 14	16.2 9	71.2 17	14.7 10	26.6 9	23.0 13	5.83 7	53.8 16	39.2 18	25.6 10	22.8 9	18.8 9	47.4 19
07/28/14	SGM	F	25.3 12	45.1 16	4.33 2	6.87 2	32.2 18	50.0 11	13.0 2	48.1 22	18.3 10	7.66 12	29.6 4	36.1 12	31.2 21	24.2 12	24.5 15	50.2 20
09/10/14	LAMC_DSM	H	26.2 13	55.8 20	11.9 17	14.3 17	18.3 11	44.0 8	21.3 18	39.9 18	29.5 19	6.67 10	31.1 8	34.5 10	28.8 18	26.3 17	30.1 13	35.7 12
07/25/14	SGBM2	Q	26.6 14	27.9 7	12.1 18	17.8 20	13.7 7	74.5 18	16.2 14	30.3 11	26.3 17	11.0 18	64.4 19	37.9 15	25.8 11	25.3 15	29.3 17	43.7 16
07/28/14	Cens5	H	26.9 15	47.1 17	8.74 11	11.9 11	25.6 14	45.3 9	22.6 19	40.6 17	29.0 18	9.93 17	36.5 10	38.6 16	31.0 20	25.0 14	25.6 16	44.6 17
07/25/14	SGBM1	Q	28.1 16	28.3 8	17.2 20	19.0 21	14.5 6	57.9 13	18.2 16	31.8 13	31.4 21	13.2 19	58.6 17	38.6 17	27.0 15	25.9 16	31.4 19	59.7 22
07/25/14	SGBM1	F	28.4 17	43.5 15	9.09 13	13.6 15	25.9 16	82.0 19	14.4 8	43.4 18	30.3 20	5.98 9	59.3 18	45.8 21	28.5 17	24.9 13	20.1 13	45.9 18
07/28/14	ELAS	H	29.3 18	41.8 14	11.1 16	16.6 19	27.9 17	96.0 22	19.8 17	31.6 12	21.4 12	14.1 20	50.8 16	40.2 20	29.2 19	29.7 18	32.3 20	37.3 14
04/08/15	REAF	H	31.4 19	58.3 21	30.9 23	13.1 13	45.3 22	63.8 15	30.9 22	38.7 15	25.3 16	8.60 15	39.3 12	36.8 13	27.0 14	35.5 20	18.2 8	39.7 15
04/09/15	PFS	F	32.2 20	65.1 23	29.4 22	12.1 12	50.0 23	70.8 16	28.2 21	44.6 19	23.1 14	7.85 13	37.0 11	37.7 14	27.9 16	36.0 21	19.8 11	35.7 13

立体视觉—标准评测库 (KITTI)



Additional information used by the methods								
■ Flow: Method uses optical flow (2 temporally adjacent images)								
■ Multiview: Method uses more than 2 temporally adjacent images								
■ Motion stereo: Method uses epipolar geometry for computing optical flow								
■ Additional training data: Use of additional data sources for training (see details)								

Evaluation ground truth									All pixels	Evaluation area	All pixels
Method	Data	Code	D1-bg	D1-fg	D1-all	Density	Time	Environment	Compare		
1 Displets v2		code	3.00 %	5.56 %	3.43 %	100.00 %	265 s	>8 cores @ 3.0 Ghz (Matlab + C/C++)			
F. Guney and A. Geiger: Displets: Resolving Stereo Ambiguities using Object Knowledge , Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
2 MC-CNN-acrf		code	2.89 %	8.88 %	3.89 %	100.00 %	67 s	Nvidia GTX Titan X (CUDA, Lua/Torch7)			
J. Zbontar and Y. LeCun: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches , Submitted to JMLR .											
3 CNN-SPS		code	3.30 %	7.92 %	4.07 %	100.00 %	80 s	GPU @ 2.5 Ghz (C/C++)			
Anonymous submission											
4 PRSM		code	3.02 %	10.52 %	4.27 %	99.99 %	300 s	1 core @ 2.5 Ghz (C/C++)			
C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model , ijcv 2015.											
5 DispNetC		code	4.32 %	4.41 %	4.34 %	100.00 %	0.06 s	Nvidia GTX Titan X (Caffe)			
N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy and T. Brox: A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation , CVPR 2016.											
6 Content-CNN		code	3.73 %	8.58 %	4.54 %	100.00 %	1 s	Nvidia GTX Titan X (Torch)			
W. Luo, A. Schwing and R. Urtasun: Efficient Deep Learning for Stereo Matching , CVPR 2016.											
7 SGM+CNN		code	3.93 %	10.56 %	5.04 %	100.00 %	2 s	Nvidia GTX 970			
Anonymous submission											
8 EEL		code	3.86 %	11.16 %	5.07 %	99.99 %	5 s	1 core @ 2.5 Ghz (C/C++)			
Anonymous submission											
9 SPS-St		code	3.84 %	12.67 %	5.31 %	100.00 %	2 s	1 core @ 3.5 Ghz (C/C++)			
K. Yamaguchi, D. McAllester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation , ECCV 2014.											
10 MDP		code	4.19 %	11.25 %	5.36 %	100.00 %	11.4 s	4 cores @ 3.5 Ghz (Matlab + C/C++)			
Anonymous submission											
11 CNN-MS		code	3.89 %	13.28 %	5.45 %	100.00 %	3 min	GPU @ TITAN X (Lua/Torch)			
Anonymous submission											
12 OSF		code	4.54 %	12.03 %	5.79 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)			
M. Menze and A. Geiger: Object Scene Flow for Autonomous Vehicles , Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
13 SFFG		code	4.57 %	13.04 %	5.98 %	99.99 %	80 s	1 core @ 2.5 Ghz (C/C++)			
Anonymous submission											
14 MBM		code	4.69 %	13.05 %	6.08 %	100.00 %	0.13 s	1 core @ 3.0 Ghz (C/C++)			
N. Einecke and J. Eggert: A Multi-Block-Matching Approach for Stereo , IV 2015.											

<http://www.cvlibs.net/datasets/kitti/>

立体视觉—标准评测

Middlebury Stereo Evaluation - Version 3

ELAS update

Snapshot of 9/14/2015 before updating ELAS results. The old ELAS results were worse because of a bug in the conversion to grayscale images (using ImageMagick's "convert" under linux). The conversion is now done via the code in the SDK 1.6.

See other snapshots [here](#).

Set: [test dense](#) [test sparse](#) [training dense](#) [training sparse](#)

Metric: [bad 0.5](#) [bad 1.0](#) [bad 2.0](#) [bad 4.0](#) [avgerr](#) [rms](#) [A50](#) [A90](#) [A95](#) [A99](#) [time](#) [t](#)

Mask: [none](#) [all](#)

[plot selected](#) [show invalid](#) [Reset sort](#)

Date	Name	Res	Avg	Austr	AustrP	BicvC2	Class	ClassE	Compu	Crusa	Crusap	Diemb
bad 2.0 (%)												

Additional information used by the methods

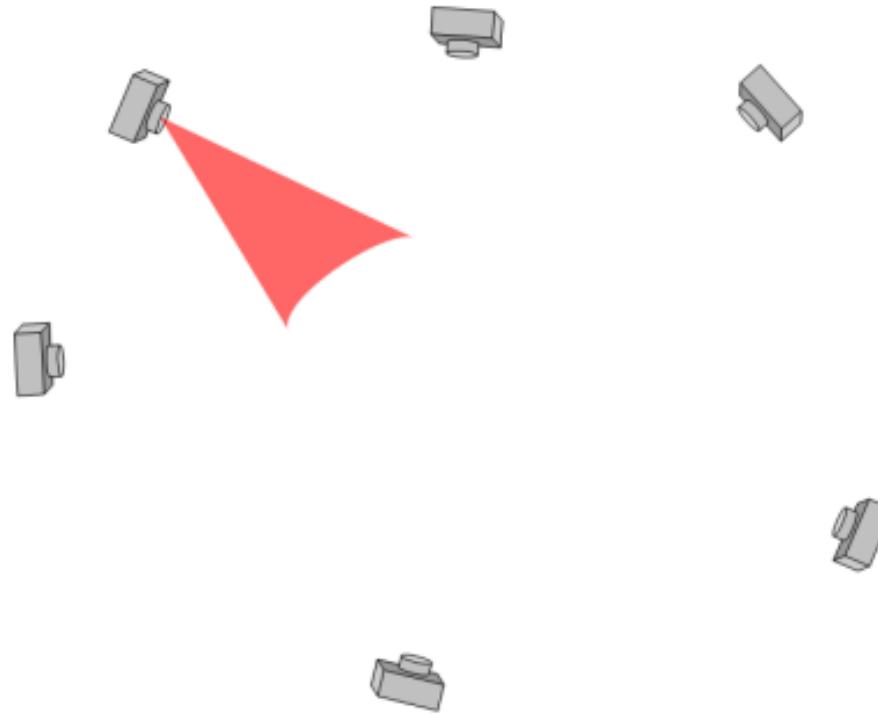
- Flow: Method uses optical flow (2 temporally adjacent images)
- Multiview: Method uses more than 2 temporally adjacent images
- Motion stereo: Method uses epipolar geometry for computing optical flow

Current State-of-the-art: Computing the Stereo Matching Cost with a Convolutional Neural Network, J. Zbontar and Y. LeCun

07/28/14	SGM	F	25.31	45.1	16	4.33	2	6.87	2	32.2	18	50.0	11	13.0	2	48.1	22	18.3	10	7.66	12
09/10/14	LAMC_DSM	H	26.24	55.8	26	11.9	17	14.3	17	18.3	11	44.0	8	21.3	18	39.9	18	29.5	10	6.67	10
07/25/14	SGBM1	G	26.64	27.9	17	1.1	16	17.8	20	13.7	7.5	16	12.4	30	13.1	26.3	17.1	11	1.0	10	
07/26/14	Cens5	H	26.99	47.1	17	8.74	11	11.9	11	25.6	14	45.3	9.9	22.6	16	40.6	17	28.0	18	9.93	17
07/25/14	SGBM1	G	28.11	28.3	8	17.2	20	19.0	21	14.5	8	57.9	13	18.2	10	31.8	13	31.4	21	13.2	19
07/25/14	SGBM1	F	28.41	43.5	12	9.09	13	13.6	15	25.9	10	82.0	18	14.4	8	43.4	18	30.3	20	5.98	9
07/28/14	ELAS	H	29.31	41.8	14	11.1	16	16.8	19	27.9	17	98.0	22	19.8	17	31.6	12	21.4	12	14	20
04/08/15	REAF	H	31.41	58.3	21	30.9	23	13.1	13	45.3	22	63.8	15	30.9	22	38.7	15	25.3	16	8.65	10
04/09/15	PFS	F	32.24	65.1	21	29.4	22	12.1	12	50.3	20	70.8	18	28.2	11	44.6	19	31.1	14	7.85	15

多视图立体重建 (Multiple View Stereo)

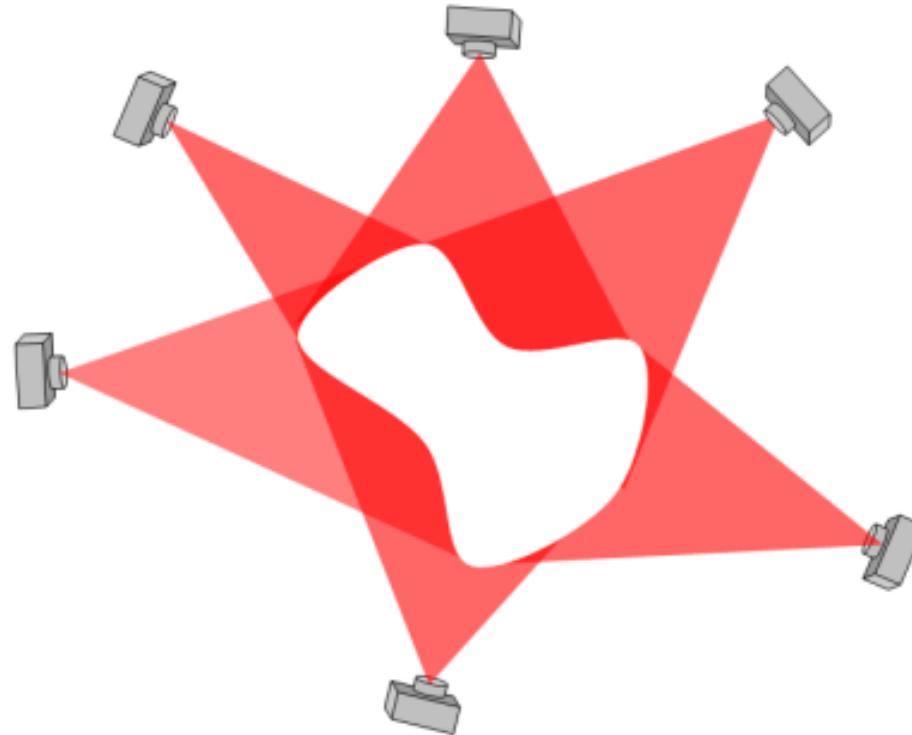
- 从两视图Stereo推广到多视图Stereo



1、对每一幅图像计算深度图

多视图立体重建 (Multiple View Stereo)

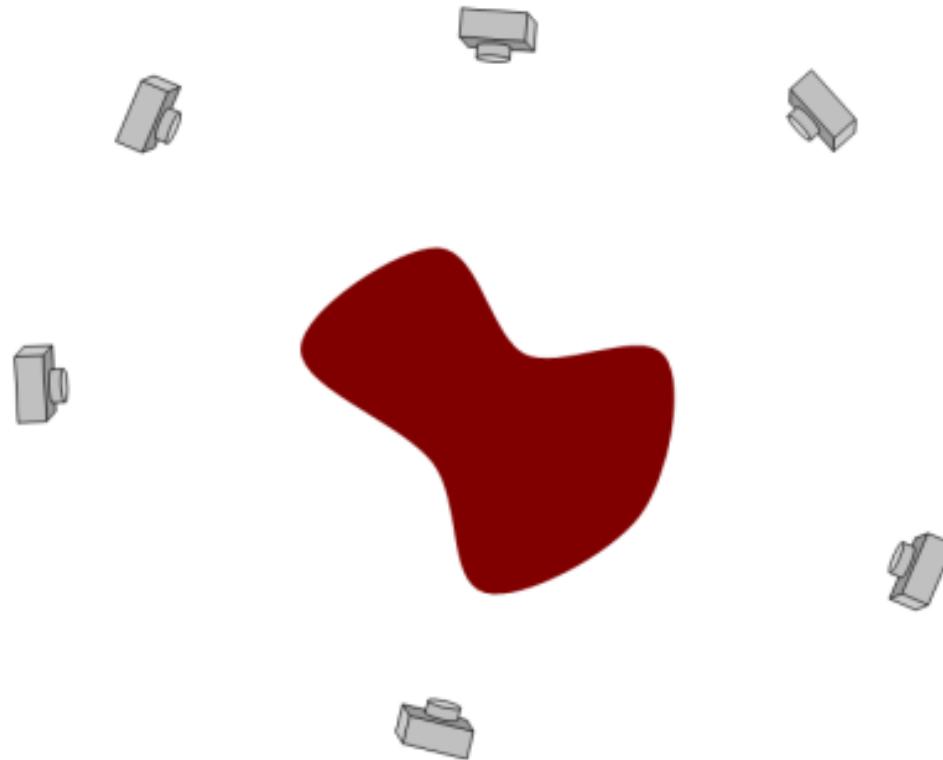
- 从两视图Stereo推广到多视图Stereo



2、深度图融合

多视图立体重建 (Multiple View Stereo)

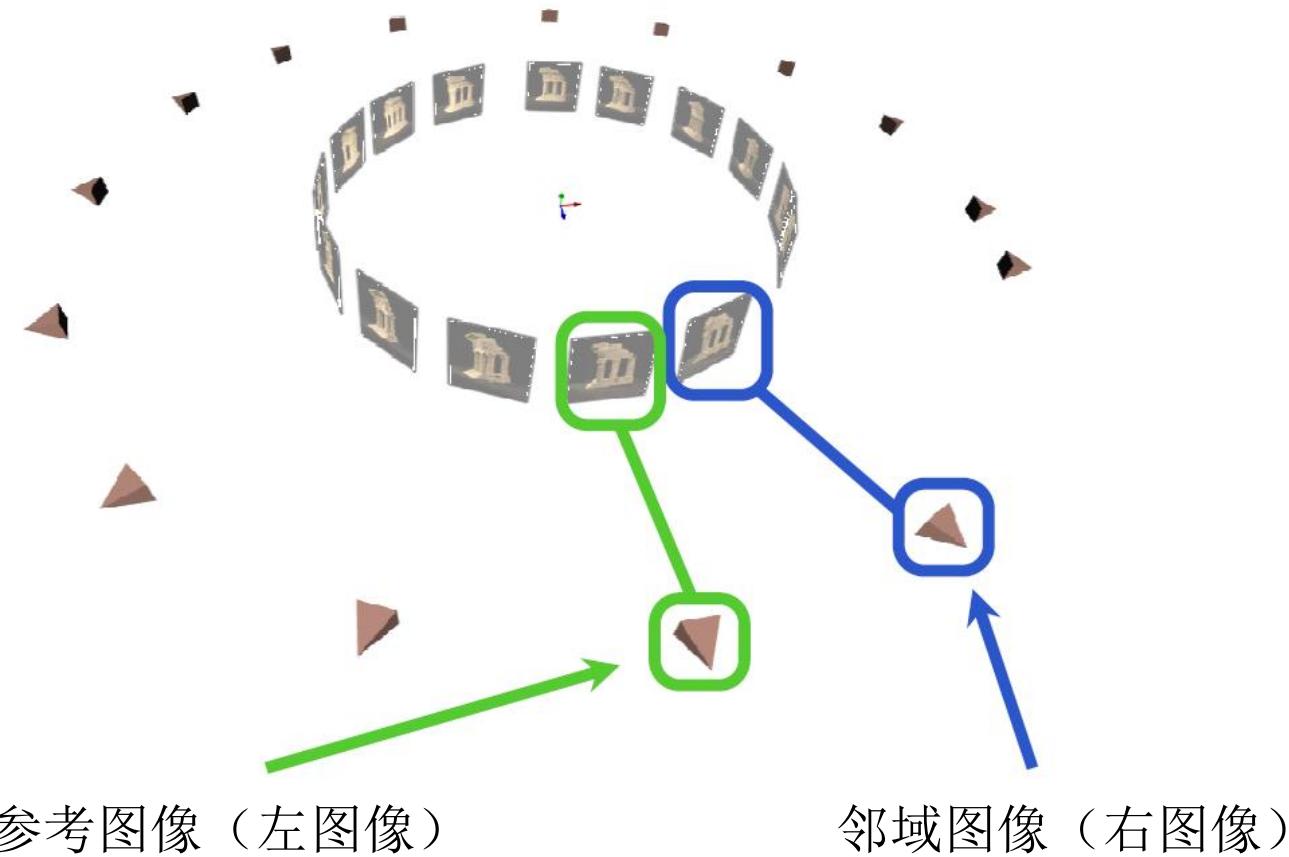
- 从两视图Stereo推广到多视图Stereo



3、抽取物体表面

多视图立体重建—深度图计算

- 为每一幅参考图像选择一幅邻域图像构成立体图像对



多视图立体重建—深度图计算

- 为每一幅参考图像选择一幅邻域图像构成立体图像对

选择原则：

- 基线足够长（保证视差精度）
- 公共可视区域足够大（保证双目覆盖范围）

选出的立体图像对是否需要极线矫正？

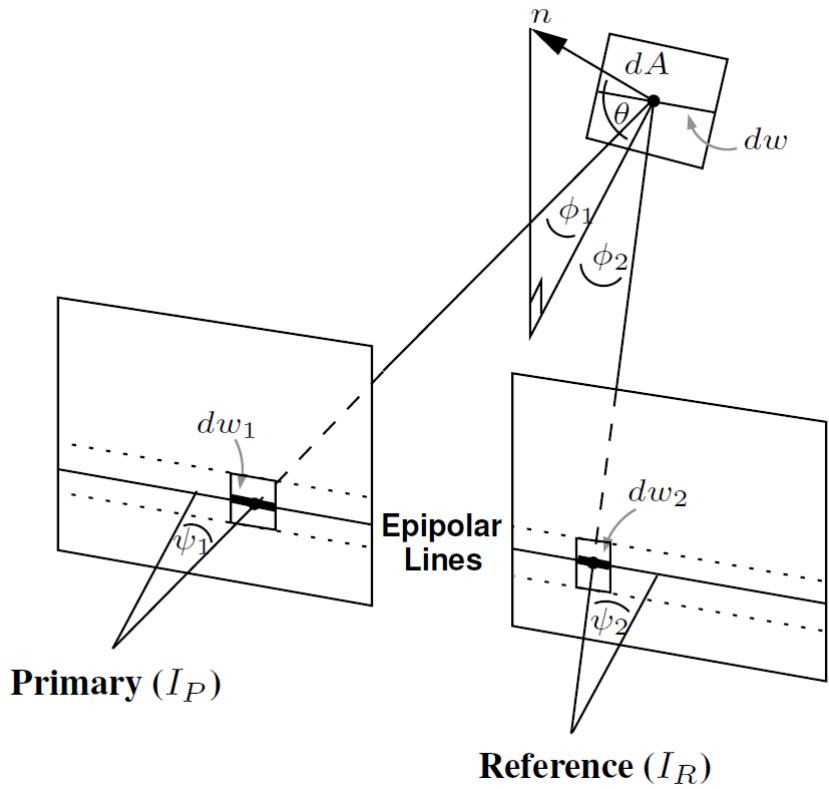
- 两视图视线夹角较小时可以进行极线矫正
- 两视图视线夹角较大时极线矫正会使图像严重拉伸
- 有些算法会选择多幅邻域图像同时参与计算
- 后两种情况通常不做极线矫正
- 如不做极线矫正，则直接在原始极线上寻找对应点

多视图立体重建—深度图计算

- 对于每一幅参考图像，根据选择的邻域图像，使用两视图 Stereo方法计算深度图（视差图）

考虑物体表面法向的窗口匹配：

- 空间面片投影到左右视图的窗口尺寸不同；
- 校正视图情况下体现在窗口宽度不同；
- 固定左图像窗口尺寸，测试右图像多种宽度窗口 ($\frac{1}{\sqrt{2}}, 1, \sqrt{2}$)，并选择相似度最高的匹配值



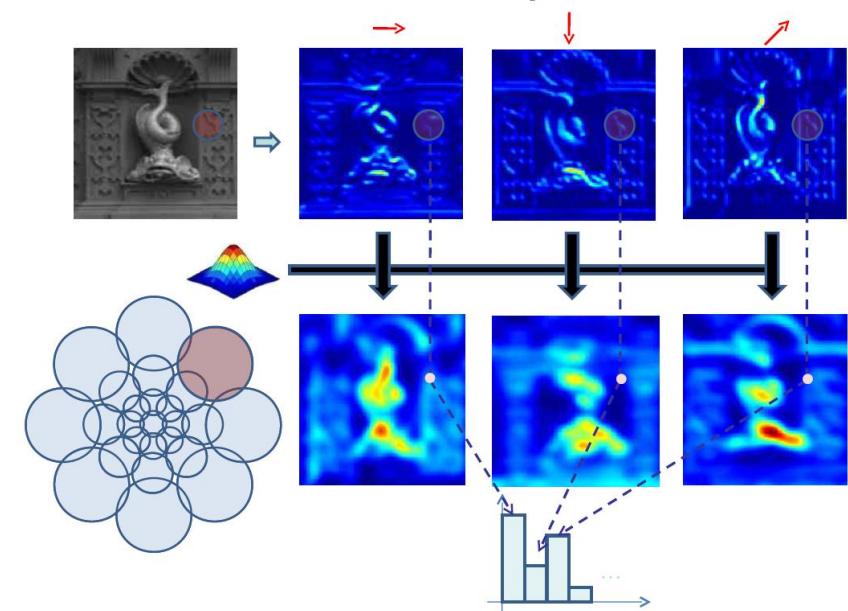
[Bradley, CVPR08]

多视图立体重建—深度图计算

- 对于每一幅参考图像，根据选择的邻域图像，使用两视图 Stereo方法计算深度图（视差图）

DAISY匹配：

- 计算每幅图像中每个像素点的DAISY描述子；



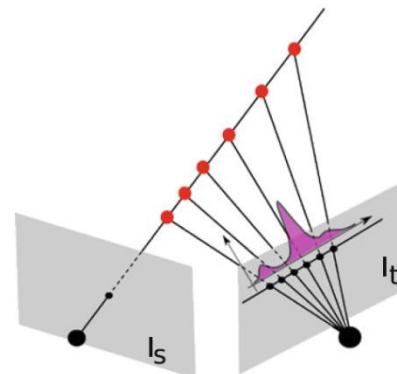
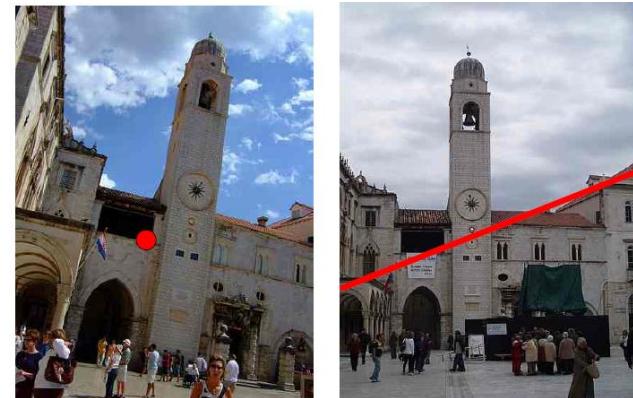
[Tola, MVA11]

多视图立体重建—深度图计算

- 对于每一幅参考图像，根据选择的邻域图像，使用两视图 Stereo方法计算深度图（视差图）

DAISY匹配：

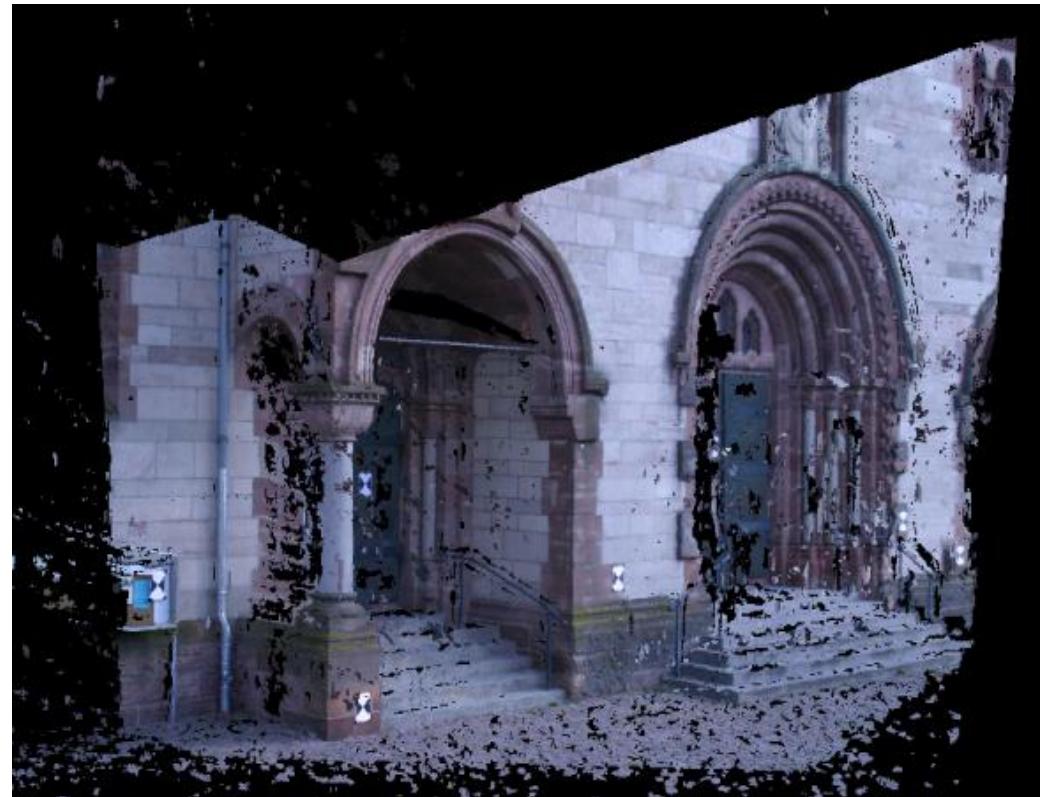
- 计算每幅图像中每个像素点的DAISY描述子；
- 直接在右图像极线上寻找DAISY描述子相似度最高的点作为匹配点；
- 这一方法特别适合于高分辨率图像（2000万像素以上）



[Tola, MVA11]

多视图立体重建—深度图过滤

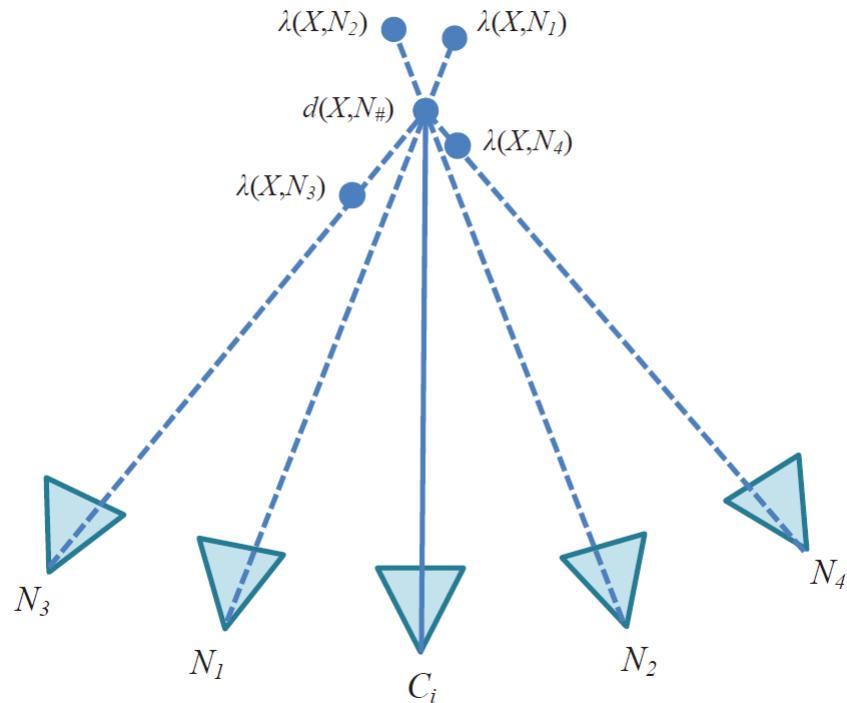
- 两视图Stereo计算的深度图不可避免存在误差和错误，直接将所有图像对应的深度图在空间融合会包含大量外点



[Shen, TIP13]

多视图立体重建—深度图过滤

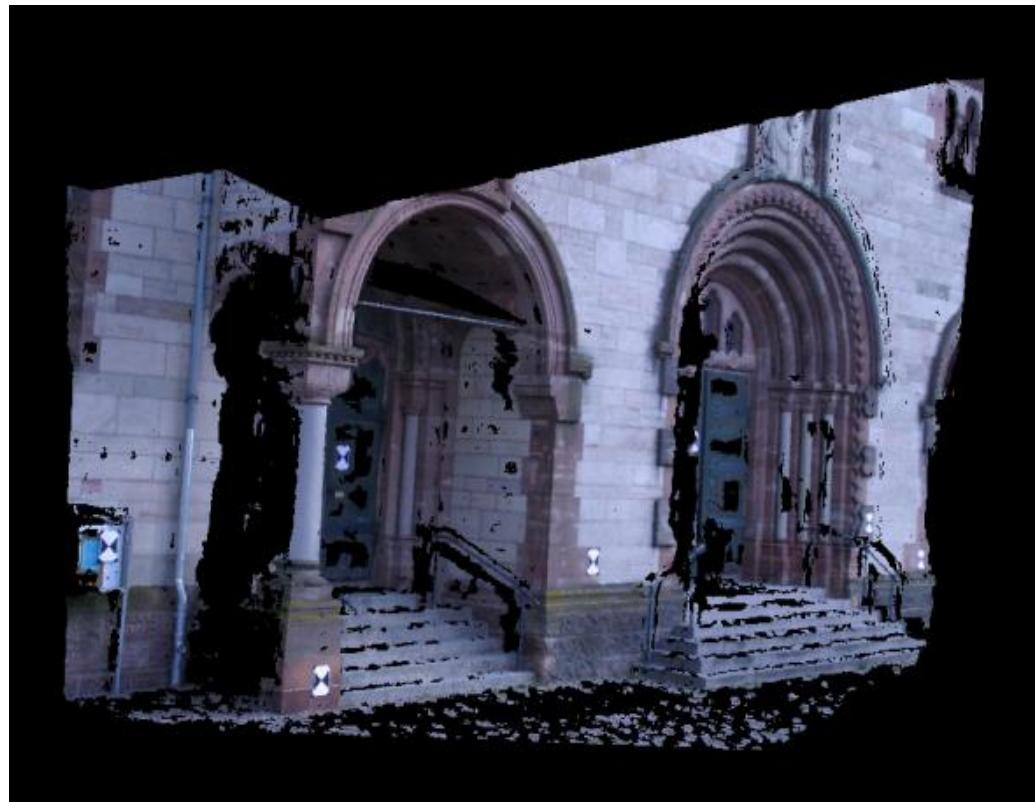
- 通过多视图深度值交叉验证可以显著减少深度图中的外点



[Shen, TIP13]

多视图立体重建—深度图过滤

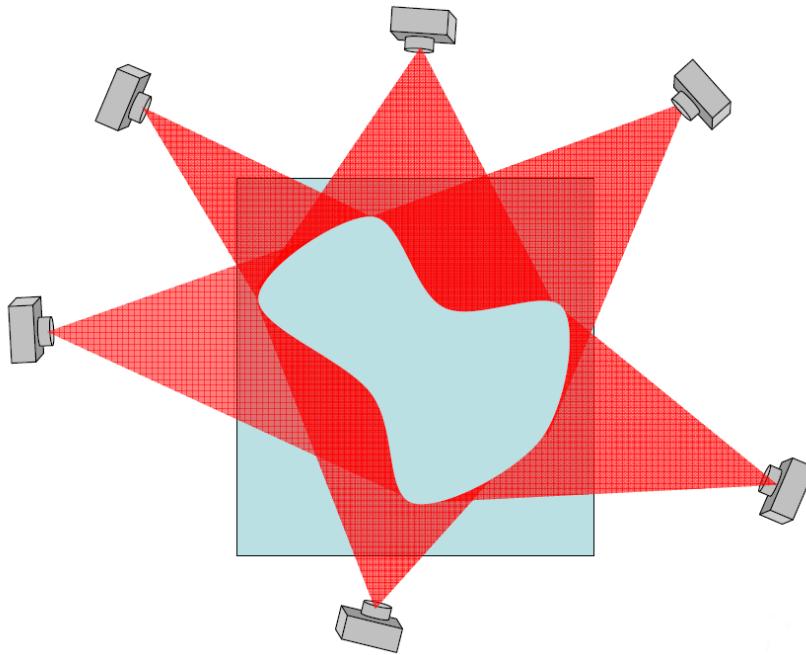
- 通过多视图深度值交叉验证可以显著减少深度图中的外点



[Shen, TIP13]

多视图立体重建—深度图融合

- 单视角深度图只包含可见表面的深度信息，通过融合所有图像上的深度图可获得完整的物体表面点云

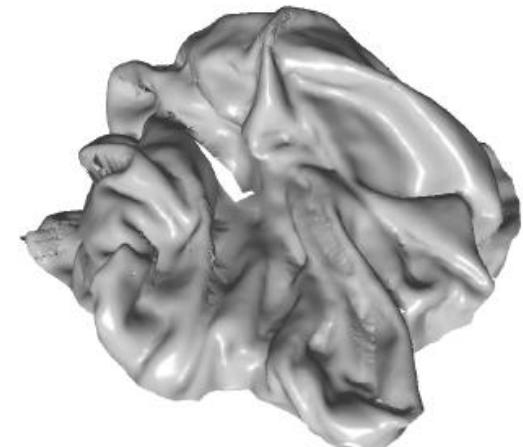
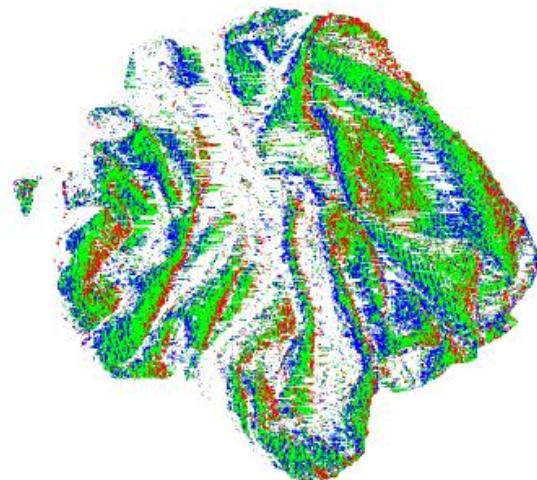


Voxel + Graph Cut + Mesh Refine

[Vogiatzis, PAMI07]

多视图立体重建—深度图融合

- 单视角深度图只包含可见表面的深度信息，通过融合所有图像上的深度图可获得完整的物体表面点云

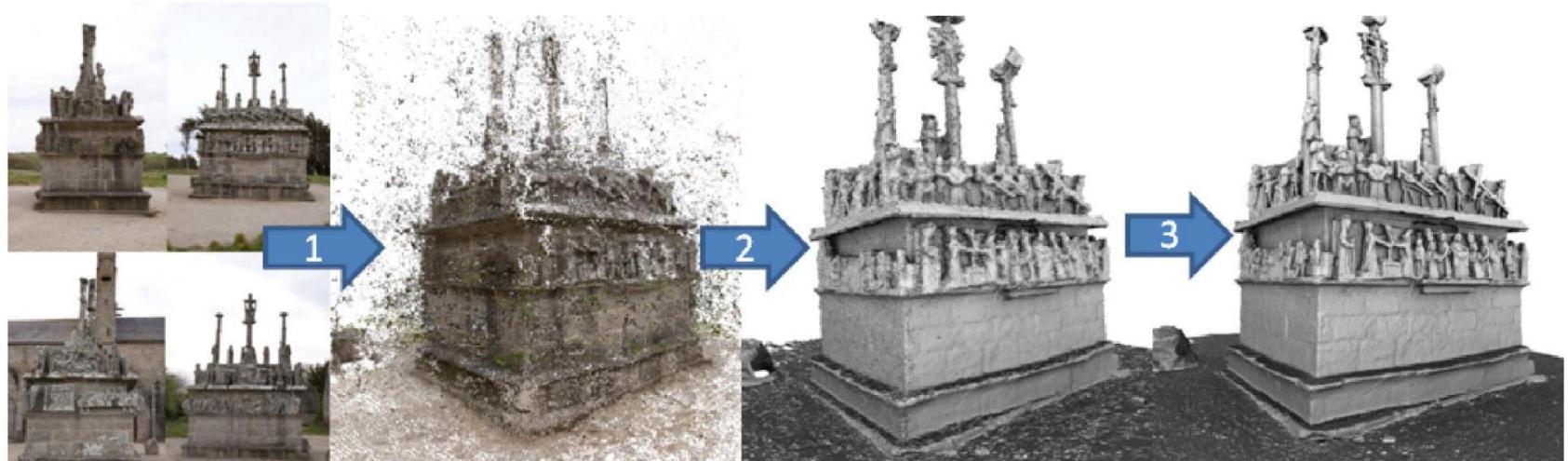


Downsampling + Cleaning + Meshing

[Bradley, CVPR08]

多视图立体重建—其他方法

- 除了基于深度图融合的多视图立体重建，还有其他一些类型的多视图立体重建方法

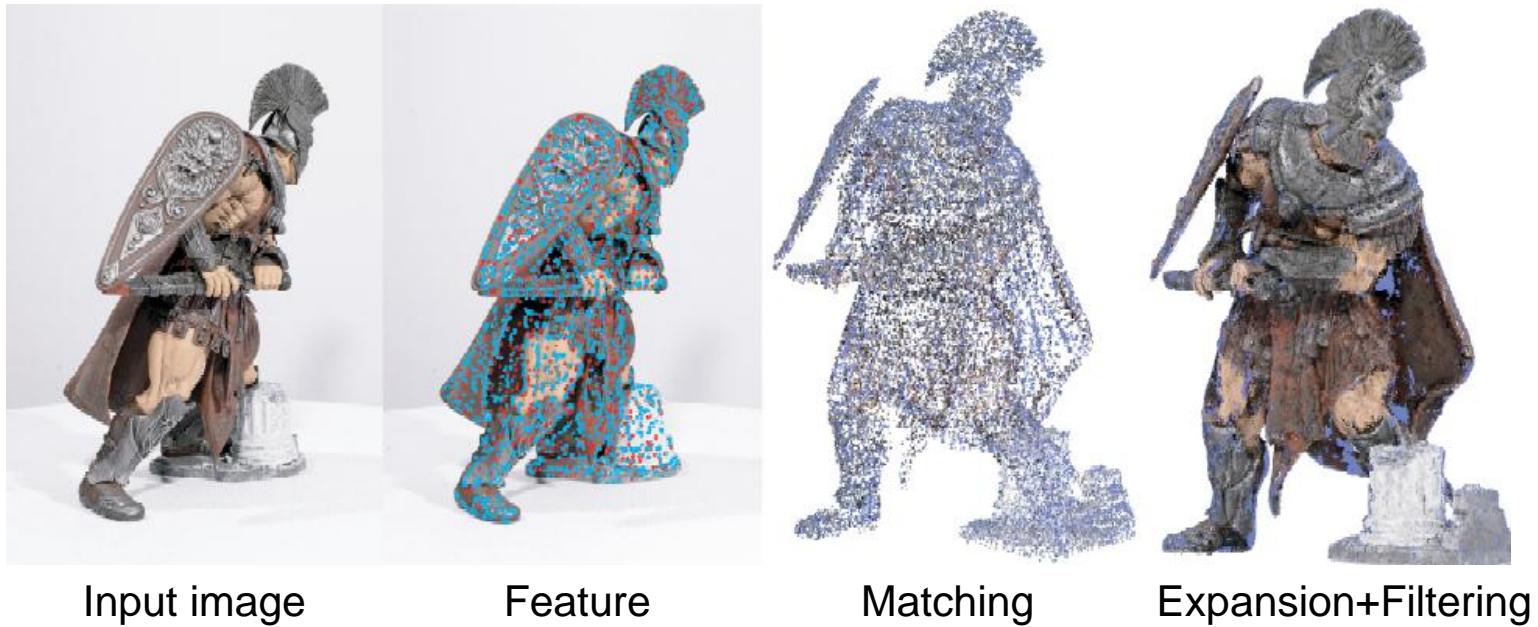


基于表面演化的方法：Delaunay + Graph Cut + Mesh Refine

[Vu, PAMI12]

多视图立体重建—其他方法

- 除了基于深度图融合的多视图立体重建，还有其他一些类型的多视图立体重建方法

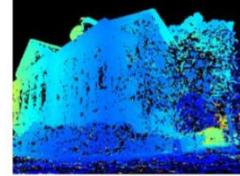
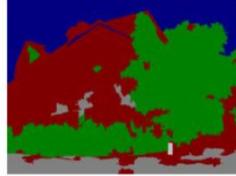


基于特征点扩散的方法： Sparse feature + Expansion + Filtering

[Furukawa, PAMI10]

多视图立体重建—其他方法

- 除了基于深度图融合的多视图立体重建，还有其他一些类型的多视图立体重建方法



基于语义的方法： Semantic segmentation + Depth-map + Graph cut

[Hane, CVPR13]

总结—本节课内容回顾

- 视差 (Disparity) 与深度 (Depth)
- 立体视觉 (Stereo)
- 多视图立体重建 (Multiple View Stereo)

总结—参考文献

- A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms
D. Scharstein and R. Szeliski
International Journal of Computer Vision, 47:7-42, 2002.
- A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms
S M Seitz, B Curless, J Diebel, D Scharstein, R Szeliski
CVPR 2006.