立体匹配算法原理与应用

奥比研究院 徐玉华

2020年3月

让 所 有 终 端 都 能 看 懂 世 界

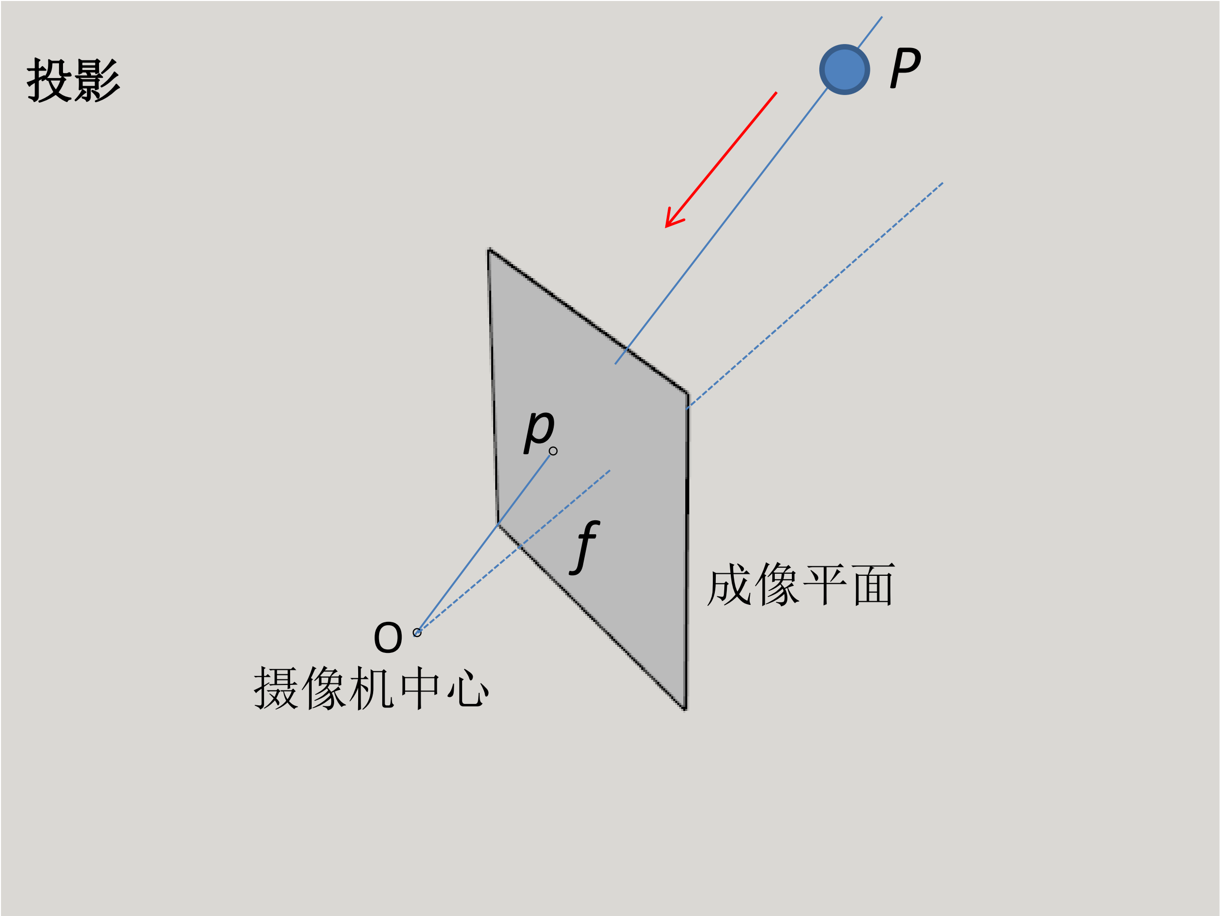
提 纲

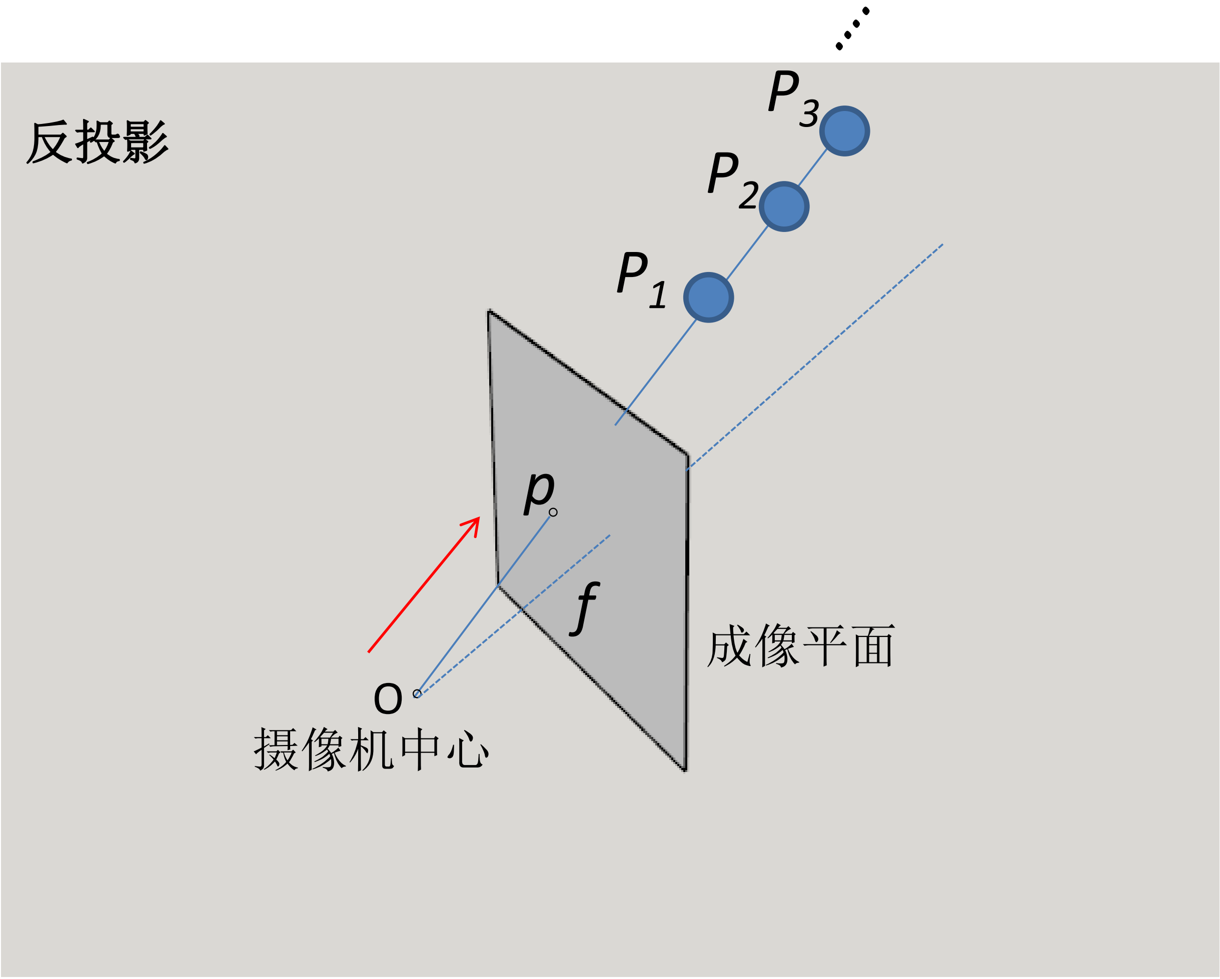
- 双目视觉基础
- 立体匹配算法

  代价函数计算，代价聚合，视差计算，视差优化/后处理，深度学习的方法
- 立体匹配算法评测
- 立体匹配算法应用(Intel D435)

双目视觉基础

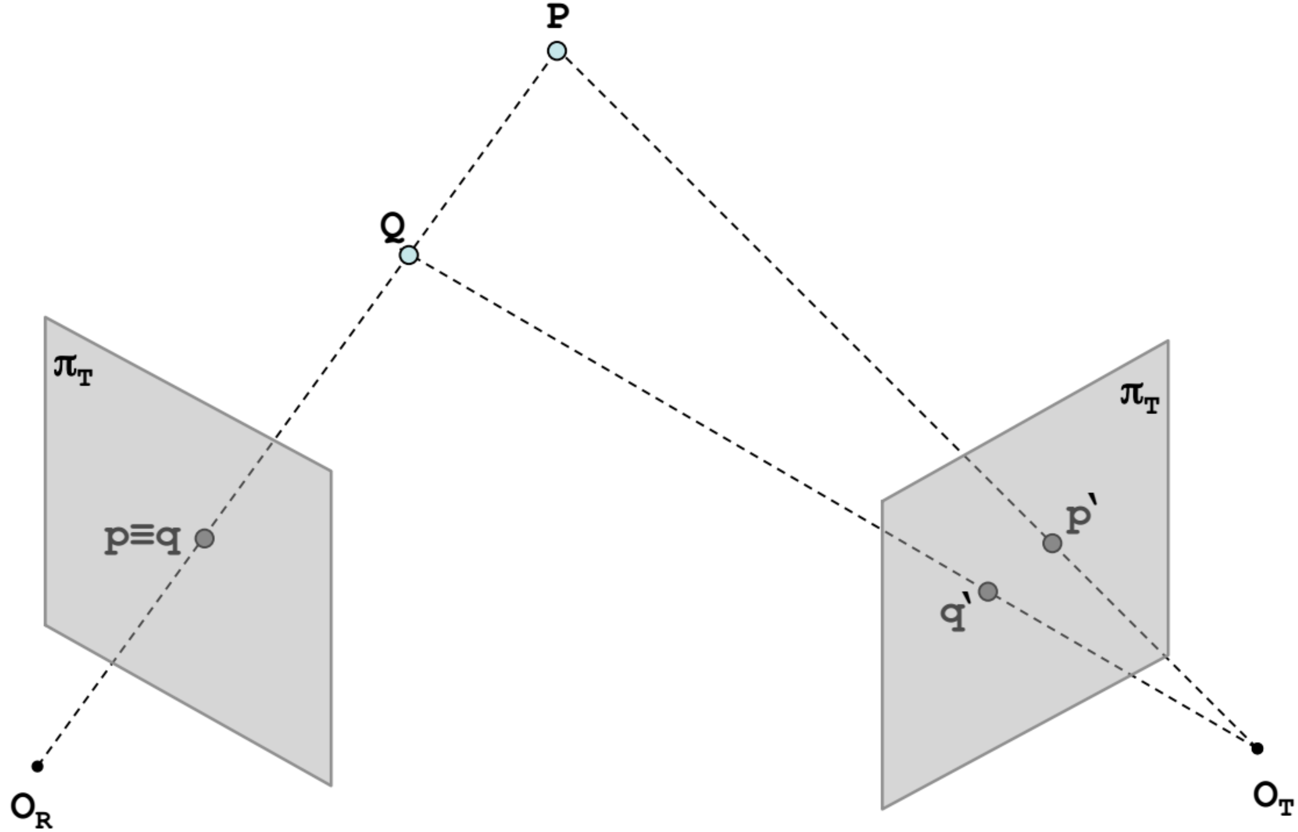
针孔摄像机模型





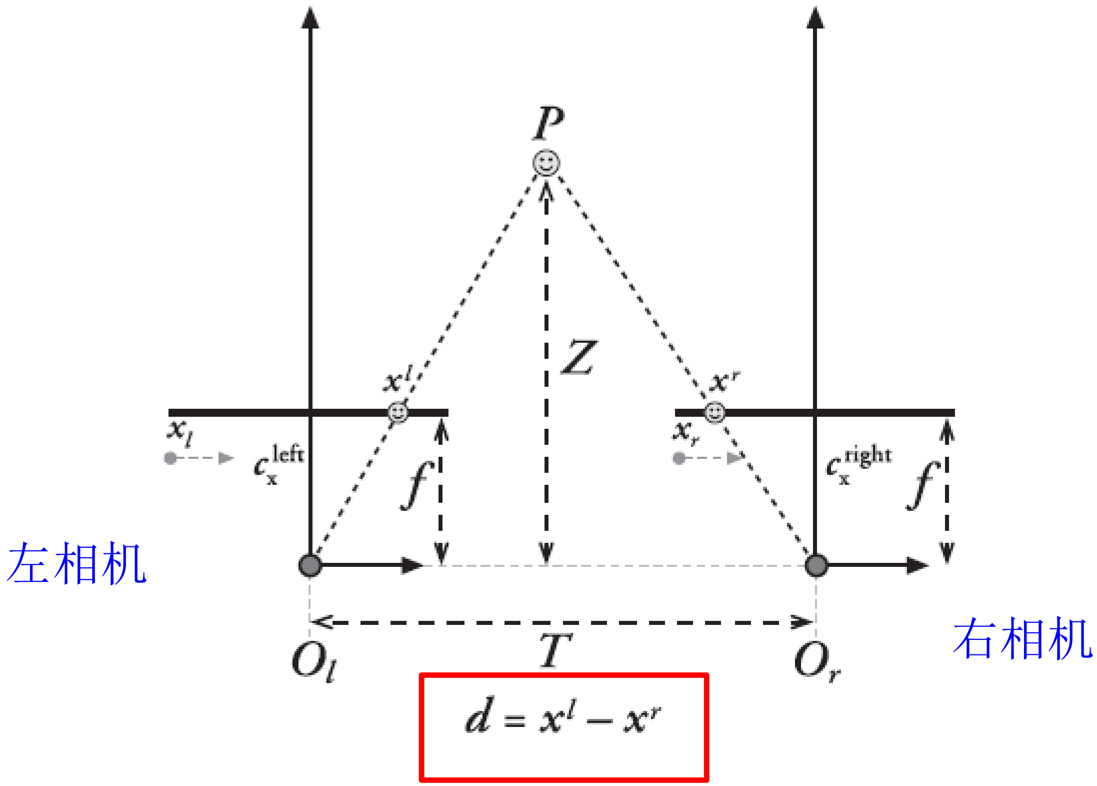
根据一幅图像中的2D像素坐标，只能确定一条射线

双目交会



- 根据2幅图像中的一对同名点，可以确定2条射线。
- 由它们的交点，可确定目标点的三维坐标。

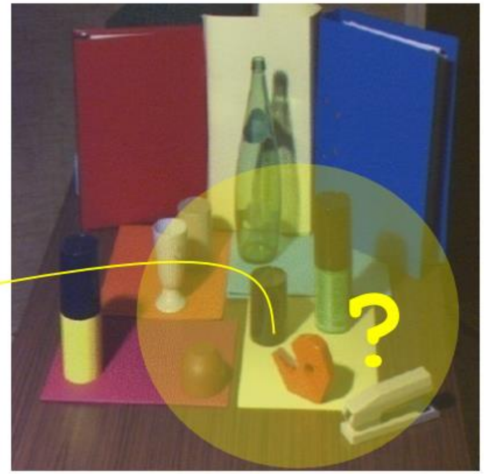
立体测量的基本原理：三角化



$$\frac{T-(x^l-x^r)}{Z-f}=\frac{T}{Z} \quad \Rightarrow \quad Z=\frac{fT}{x^l-x^r}$$

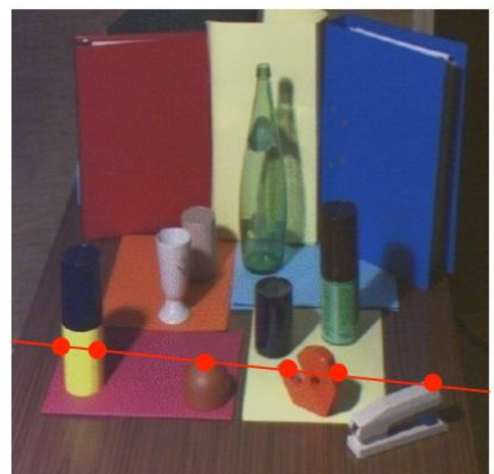
极线约束

Reference (R)



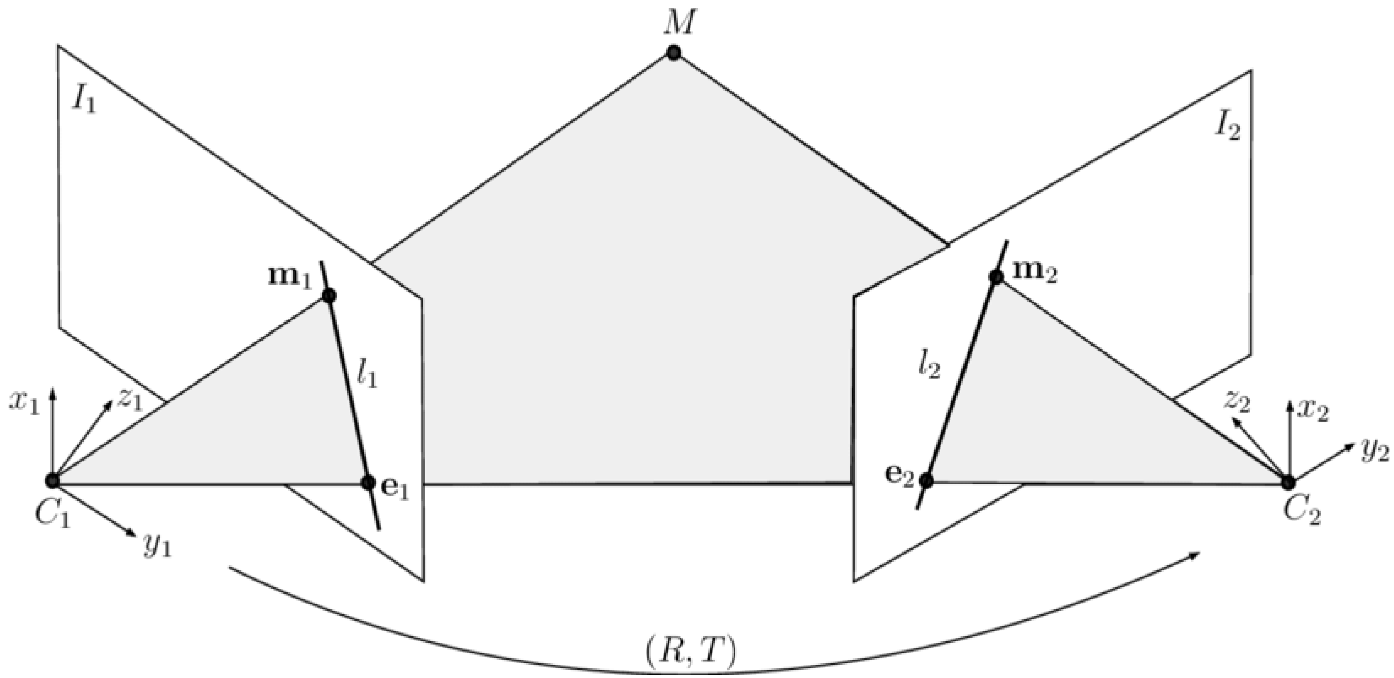
Target (T)

Reference (R)



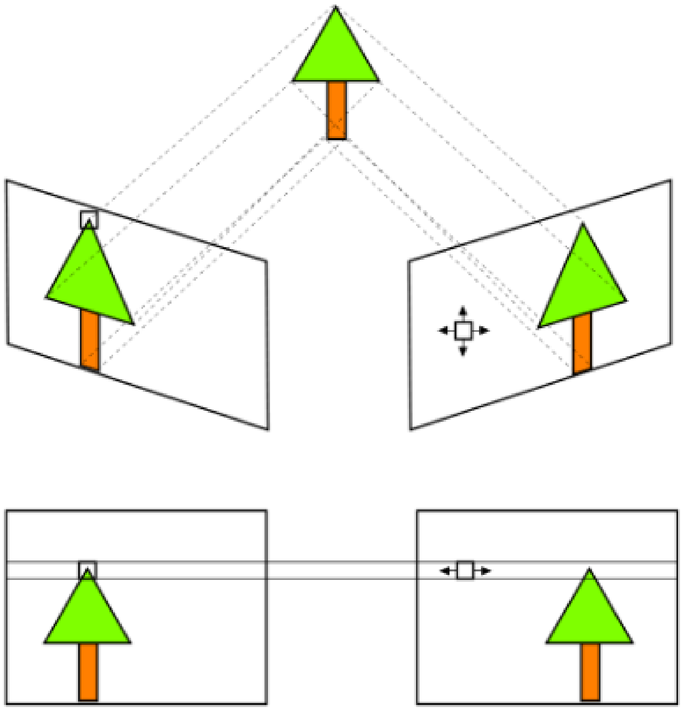
Target (T)

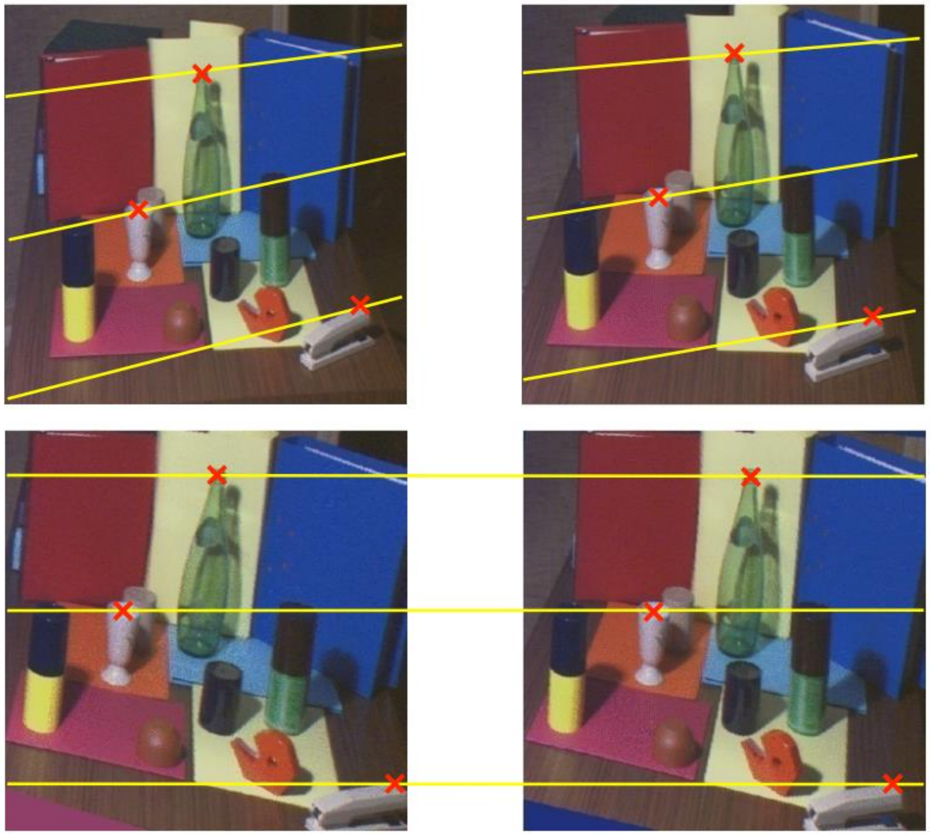
极线约束



M
I_1
I_2
$\mathbf{m}_1$
$\mathbf{m}_2$
l_1
l_2
x_1
z_1
$\mathbf{e}_1$
$\mathbf{e}_2$
z_2
x_2
y_2
C_1
y_1
C_2
(R, T)

极线校正/立体校正





- 使左右相机的X轴与基线平行
- 相机光轴与基线垂直
- 使左右相机具有相同的焦距

A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” Mach. Vis. Appl. 12(1), 16–22 (2000).

立体匹配技术难点

[Mattoccia 2013]

颜色/亮度差异和噪声

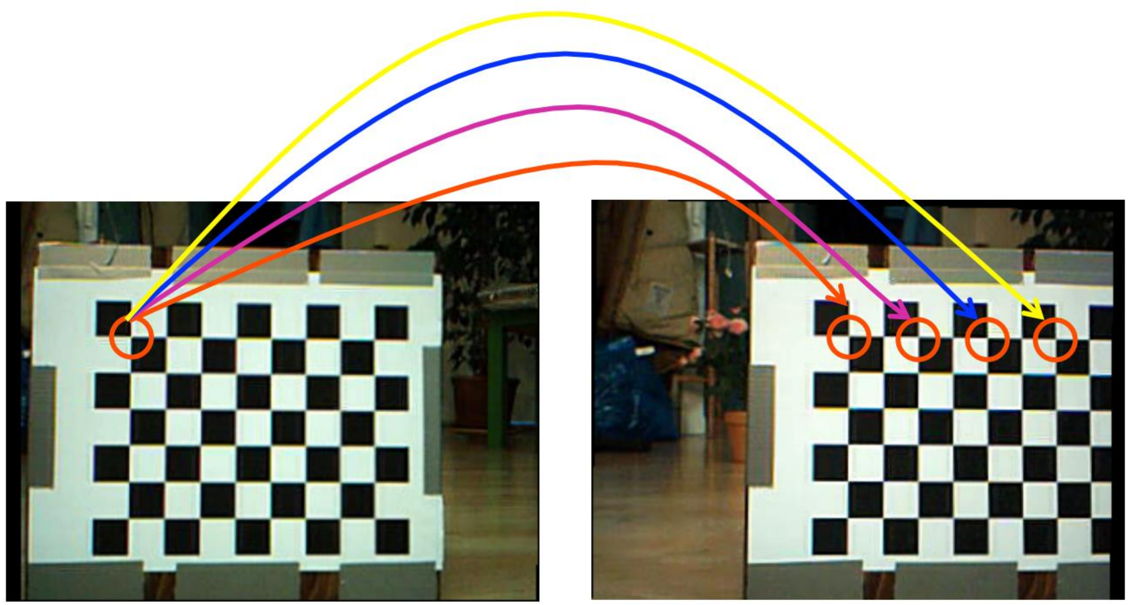
反光区域

倾斜面

透视变形

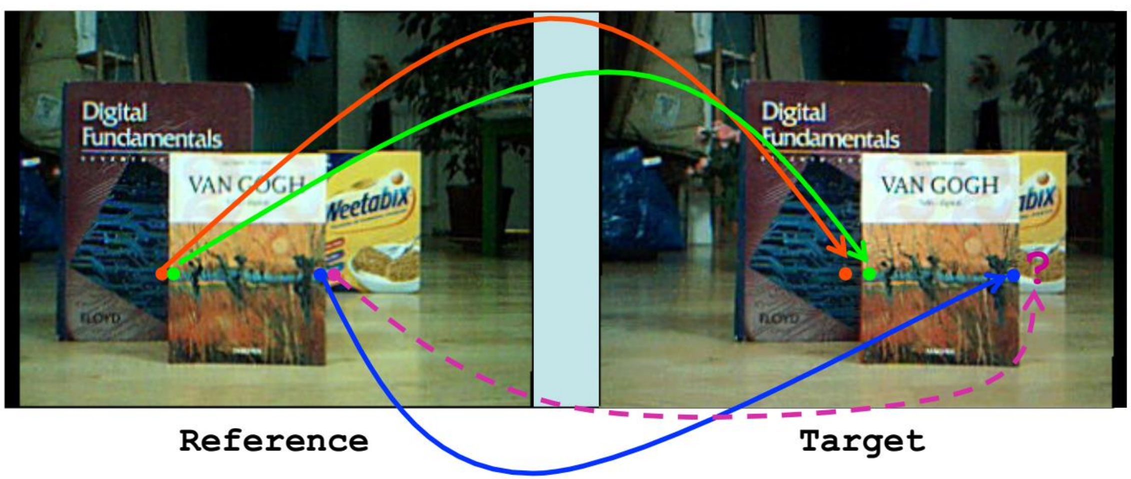
弱纹理区域

重复纹理



透明物体

遮挡和深度不连续



立体匹配方法分类

局部方法

全局方法

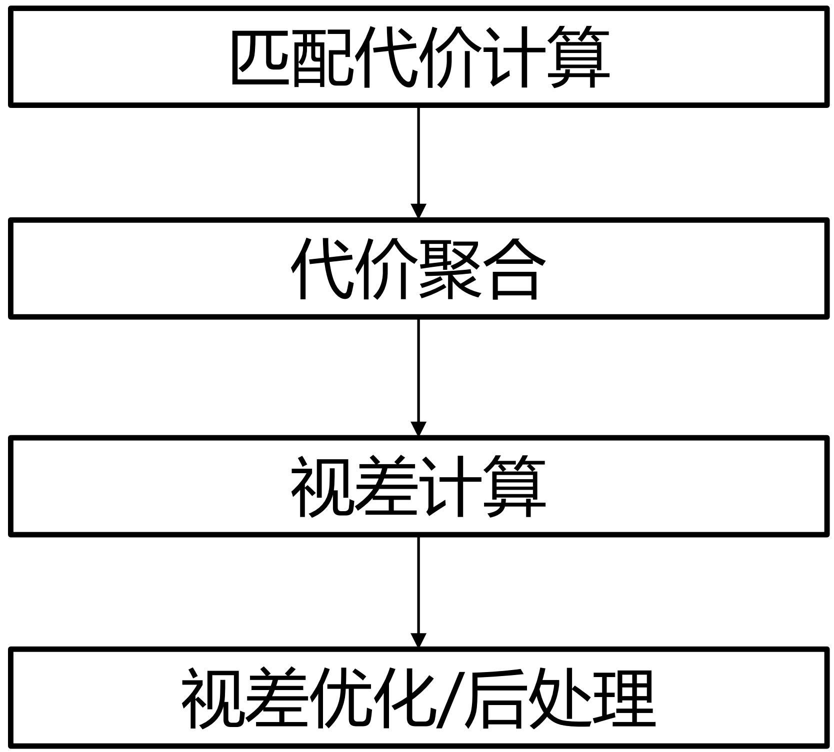
半全局方法

直接的块匹配

left　　　　right

立体匹配流程

四个步骤



匹配代价计算

$I_L(x, y)$ $I_R(x\text{-}d, y)$

代价函数用于计算左、右图中两个像素之间的匹配代价（cost）。

cost越大，表示这两个像素为对应点的可能性越低。

代价函数

- AD/BT
- AD+Gradient
- Census transform
- SAD/SSD
- NCC
- AD+Census
- CNN

代价函数

- AD/BT
- AD+Gradient
- Census
- NCC
- AD+Census
- CNN

Absolute difference

$$C_{AD}(x,y,d) = \left| I_L(x,y) - I_R(x-d,y) \right|$$

BT cost

Birchfield S, Tomasi C. A pixel dissimilarity measure that is insensitive to image sampling[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(4): 401-406.

代价函数

- AD/BT
- AD+Gradient
- Census
- NCC
- AD+Census
- CNN

截断阈值

截断阈值

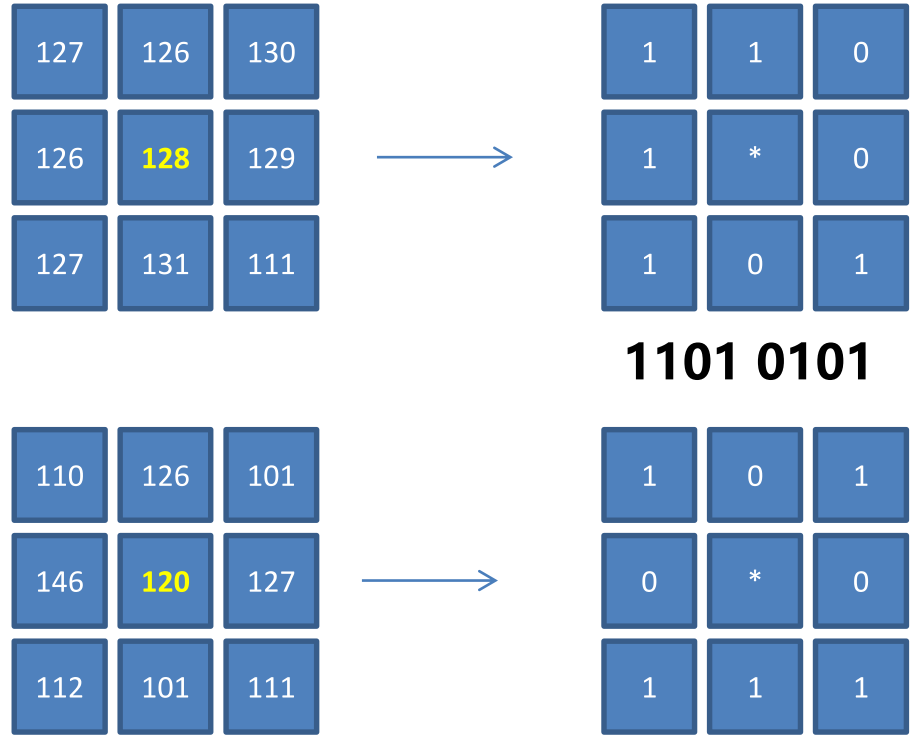
$$\rho(q,q') = (1-\alpha)\cdot\min(||I_q - I_{q'}||,\tau_{col}) + \alpha\cdot\min(||\nabla I_q - \nabla I_{q'}||,\tau_{grad})$$

权重

Bleyer M, Rhemann C, Rother C. PatchMatch Stereo-Stereo Matching with Slanted Support Windows. BMVC. 2011.

代价函数

- AD/BT
- AD+Gradient
- Census
- NCC
- AD+Census
- CNN



1101 0101

1010 0111

1101 0101
1010 0111

异或-> **0111 0010**

Hamming distance: 4

代价函数

- AD/BT
- AD+Gradient
- Census
- SAD/SSD
- NCC
- AD+Census
- CNN

$$C_{NCC}(x',y',d)=\frac{\sum_{(x,y)\in W(x',y')}\left(I_L(x,y)-\bar{u}_L\right)\left(I_R(x-d,y)-\bar{u}_R\right)}{\sqrt{\sum_{(x,y)\in W(x',y')}\left(I_L(x,y)-\bar{u}_L\right)^2}\sqrt{\sum_{(x,y)\in W(x',y')}\left(I_R(x-d,y)-\bar{u}_R\right)^2}}$$

- 特性：对图像亮度的线性变化具有不变性
- 物理意义: 两个向量的夹角的余弦值

代价函数

[Mei 2011]

- AD/BT
- AD+Gradient
- Census
- SAD/SSD
- NCC
- AD+Census
- CNN

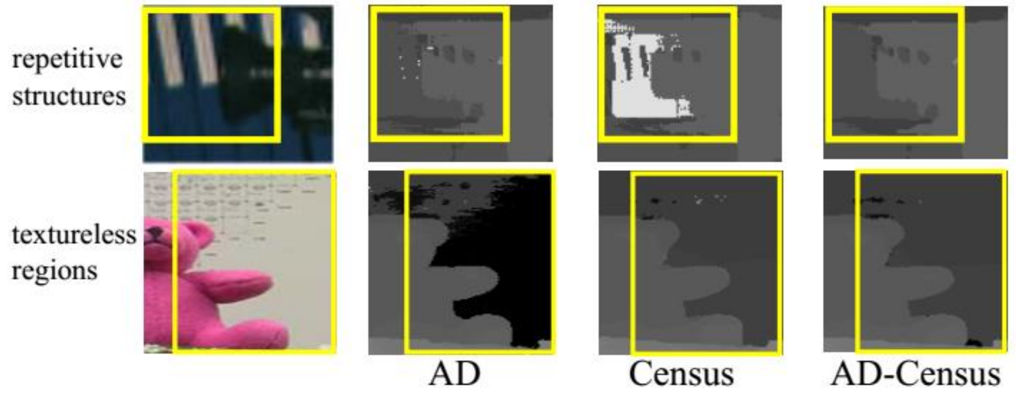


Figure 1. Some close-up disparity results on Tsukuba and Teddy image pair, which are computed with AD, Census, AD-Census cost measures and cross-based aggregation. AD-Census measure produces proper disparity results for both repetitive structures and textureless regions.

AD代价函数容易实现，但是它容易受辐射差异的影响。
而在Census变换中，不要求像对之间的颜色一致性。因此，它对于辐射差异更加鲁棒。

$$C_{AD}(\boldsymbol{p}, d) = \frac{\sum_{i=R,G,B} |I_i^{left}(\boldsymbol{p}) - I_i^{right}(\boldsymbol{p} - (d, 0))|}{3}$$

$$C_I(\boldsymbol{p}, d) = 1 - exp(-\frac{C_{AD}(\boldsymbol{p}, d)}{\lambda_{AD}}) + 1 - exp(-\frac{C_{census}(\boldsymbol{p}, d)}{\lambda_{Census}})$$

代价函数

- AD/BT
- AD+Gradient
- Census
- SAD/SSD
- NCC
- AD+Census
- CNN

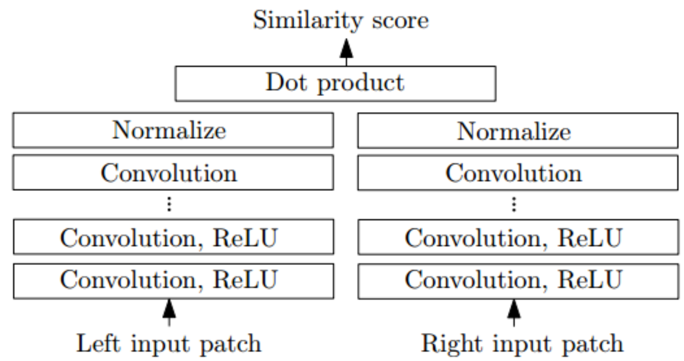


Figure 2: The fast architecture is a siamese network. The two sub-networks consist of a number of convolutional layers followed by rectified linear units (abbreviated "ReLU"). The similarity score is obtained by extracting a vector from each of the two input patches and computing the cosine similarity between them. In this diagram, as well as in our implementation, the cosine similarity computation is split in two steps: normalization and dot product. This reduces the running time because the normalization needs to be performed only once per position (see Section 3.3).

网络结构举例

4层网络，每层3x3卷积核，32个通道
权重的数量：
1*32*3*3 +
32*32*3*3 +
32*32*3*3 +
32*32*3*3 = 27936

计算每一层feature map的乘法计算量：

$$\#FLOPs = ch_{in} * ch_{out} * k^2 * input_w * input_h$$

(32 * 32 * 3 * 3 * 1280 * 720)

1. Zbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research, 2016.
2. Park H, Lee K M. Look wider to match image patches with convolutional neural networks. IEEE Signal Processing Letters, 2017.

代价函数

- AD/BT
- AD+Gradient
- Census
- SAD/SSD
- NCC
- AD+Census
- CNN

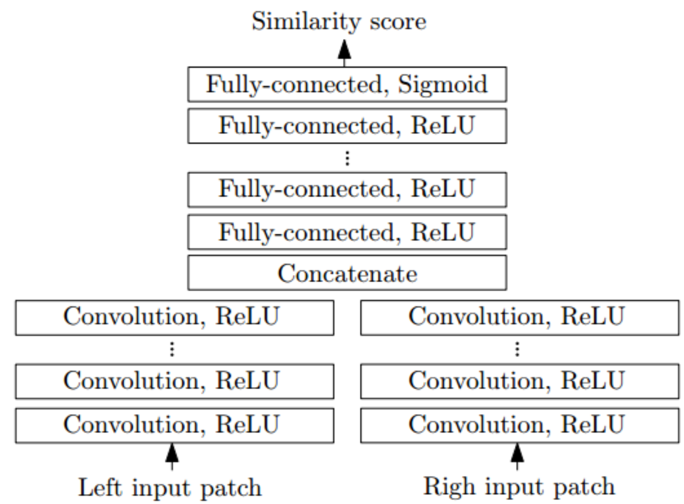


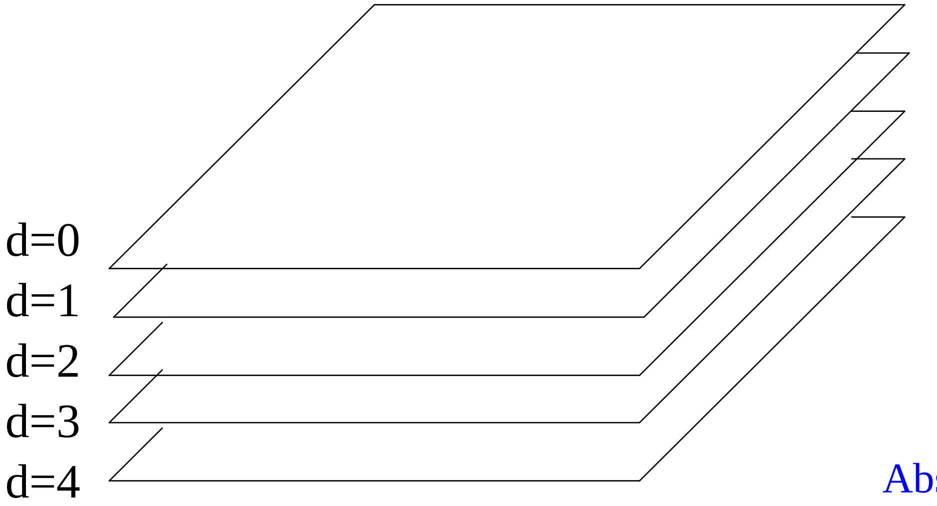
Figure 3: The accurate architecture begins with two convolutional feature extractors. The extracted feature vectors are concatenated and compared by a number of fully-connected layers. The inputs are two image patches and the output is a single real number between 0 and 1, which we interpret as a measure of similarity between the input images.

代价函数

- AD/BT
- AD+Gradient
- Census
- SAD/SSD
- NCC
- AD+Census
- CNN

MC-CNN流程

代价空间 Cost Volume



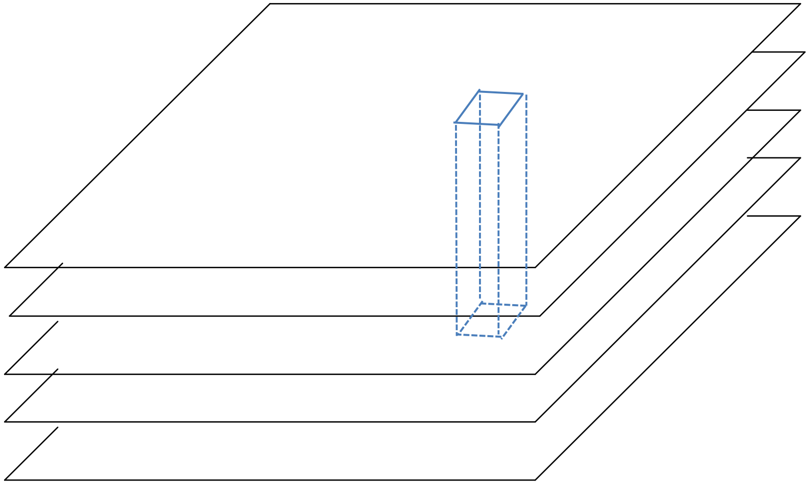
Absolute difference

$$C_{AD}(x,y,d) = \left| I_L(x,y) - I_R(x-d,y) \right|$$

C(x, y, d)

代价空间和Sliding-window之间关系

left　　　　right



d=0

d=1

d=2

d=3

d=4

...

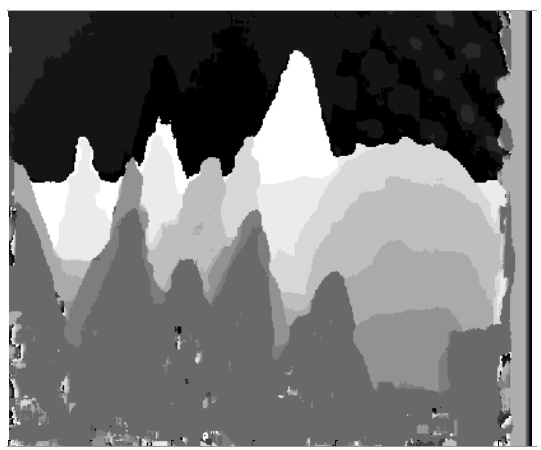
代价聚合

Box Filtering

$$C_d^A(p) = \frac{1}{N}\sum_q C_d(q)$$

优点　计算速度快

缺点　不具备保持边缘的特性



Weight map

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Disparity assumption

d	d	d	d	d
d	d	d	d	d
d	d	d	d	d
d	d	d	d	d
d	d	d	d	d

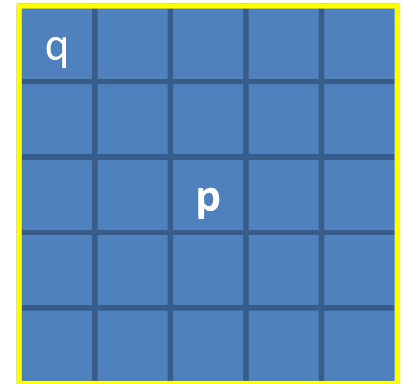
隐含的假设：窗口中的每个点的视差值都相同

(Fronto-parallel windows)

Bilateral filter

自适应权重：颜色+空间距离。影响广泛，Google引用1200+次。

$$C_d^A(p)=\frac{\sum_q \overbrace{exp(-\frac{|p-q|}{\sigma_S})}^{\text{空间距离项}}\overbrace{exp(-\frac{|I(p)-I(q)|}{\sigma_R})}^{\text{颜色距离项}}C_d(q)}{\sum_q exp(-\frac{|p-q|}{\sigma_S})exp(-\frac{|I(p)-I(q)|}{\sigma_R})}$$



q is a pixel within the user-specified support region.

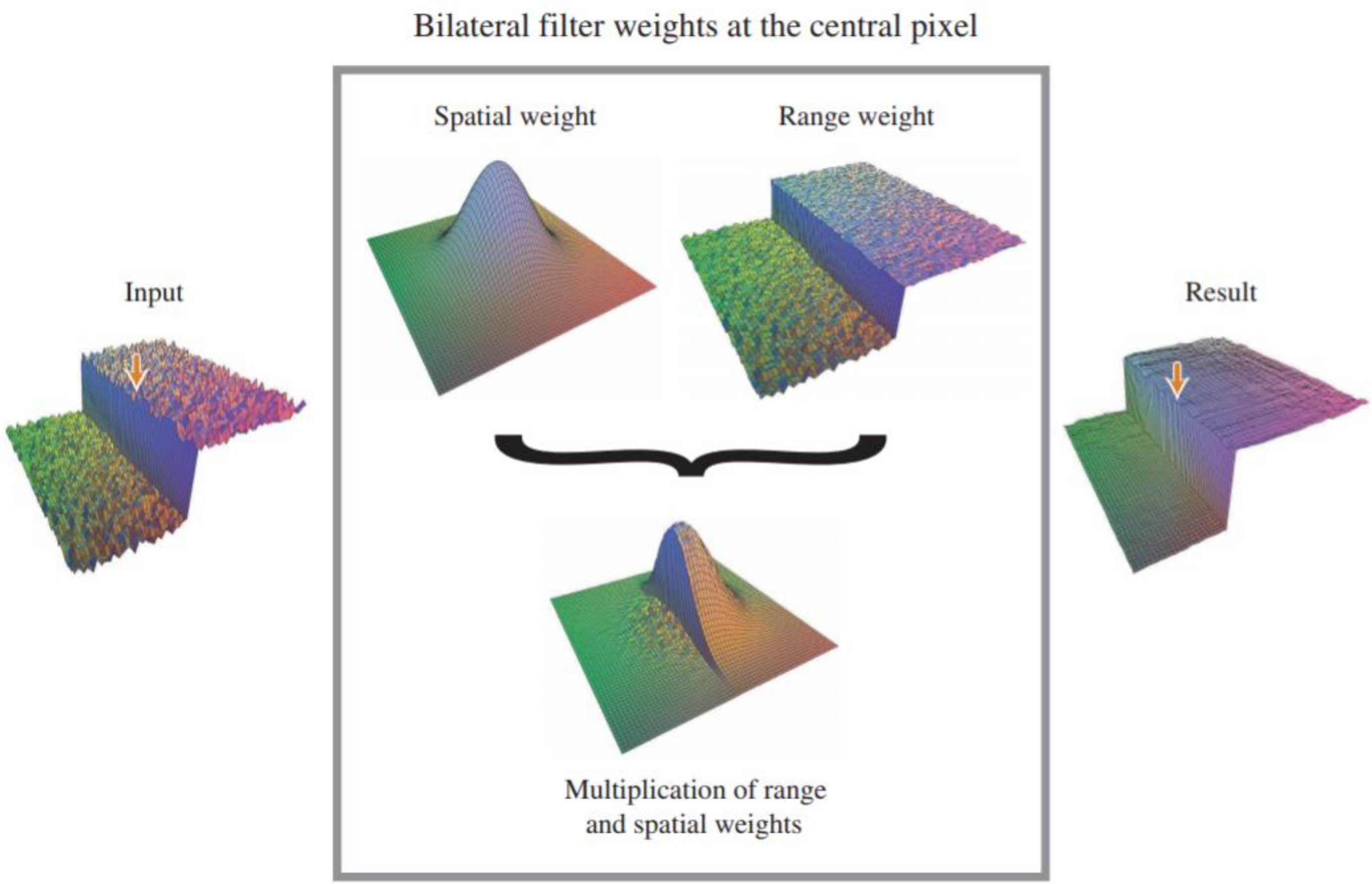
[归一化项可以省略掉]

$$C_d^A(p)=\sum_q exp(-\frac{|p-q|}{\sigma_S})exp(-\frac{|I(p)-I(q)|}{\sigma_R})C_d(q)$$

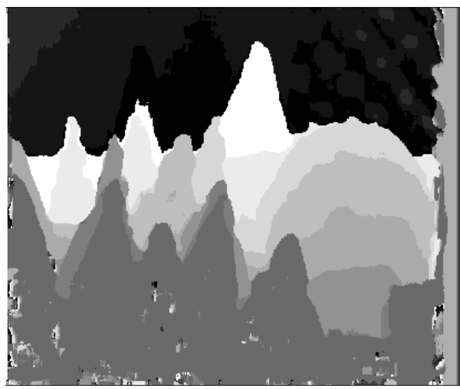
Weight map

Yoon K J, Kweon I S. Adaptive support-weight approach for correspondence search. IEEE TPAMI, 2006.

Bilateral filter



Bilateral filter

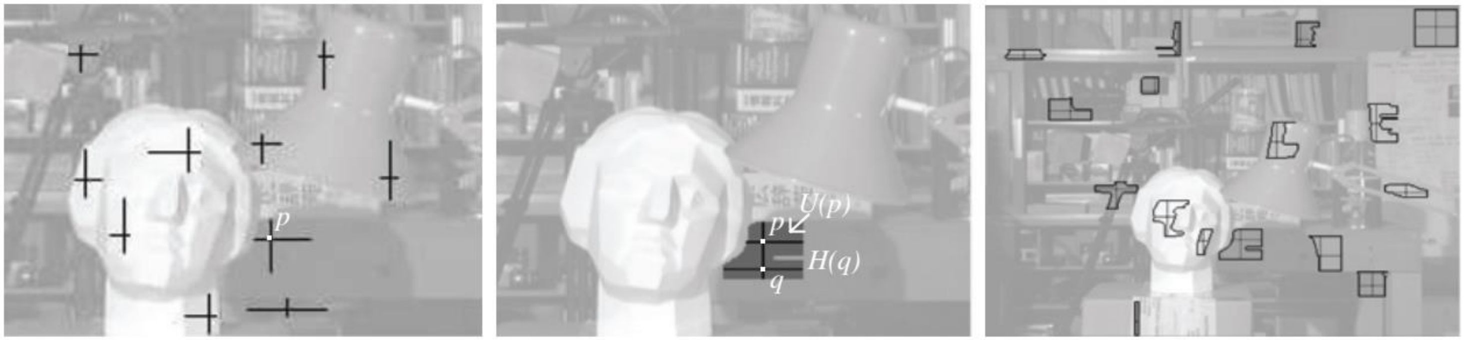


Box Filter

Bilateral Filter

GT

Cross-based local stereo matching 自适应形状



(a) (b) (c)

Fig. 1. Cross-based local support region representation and construction on the *Tsukuba* image [13]. (a) A pixelwise adaptive cross defines a local support skeleton for the anchor pixel, e.g., p. (b) A shape-adaptive full support region $U(p)$ is dynamically constructed for the pixel p, integrating multiple horizontal line segments $H(q)$ of neighboring crosses. (c) Sample shape-adaptive local support regions, approximating local image structures appropriately.

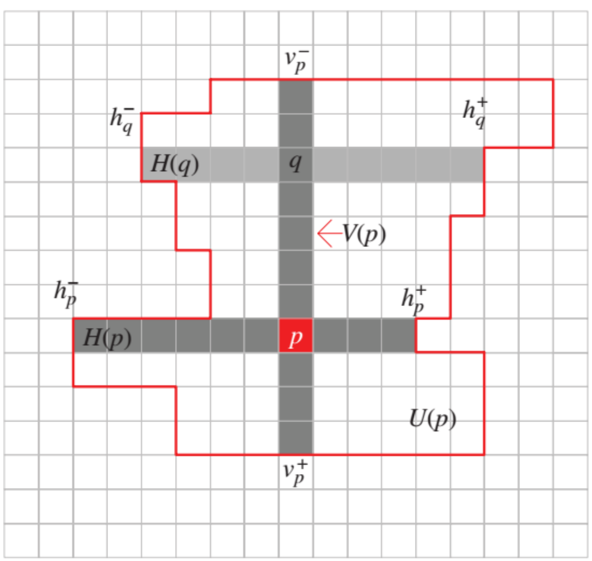


Fig. 2. Configuration of a local upright cross $H(p) \cup V(p)$ for the anchor pixel p, and the constructed full support region $U(p)$. The quadruple $\{h_p^-, h_p^+, v_p^-, v_p^+\}$ defines the left, right, up, and bottom arm length of the cross, respectively. $q \in V(p)$ is a pixel on the vertical segment $V(p)$ in (2).

- 左右两幅图上都计算support region
- 用1D积分图加速（先横向，再纵向）

需要强调的是：这个不规则形状的聚合方式，可以采用高效的积分图方式进行计算

步骤：

(1) 水平方向计算积分图HI;
(2) 用积分图HI计算每个像素点横臂的聚合代价;
(3) 计算纵向积分图VI;
(4) 对于某个十字臂区域，用积分图VI计算

公式(4)中的$U_d(p)$的形状，在各个视差平面中，是不一样的（还与右图相关）

[Zhang 2009] Zhang K, Lu J, Lafruit G. Cross-based local stereo matching using orthogonal integral images. IEEE TCSVT, 2009

Semi-Global Matching

引用次数 3500+

能量函数

$$E(D) = \sum_{\mathbf{p}} \Big(C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \Big)$$

优化方法

动态规划（线扫描优化）

Hirschmuller H. Stereo processing by semiglobal matching and mutual information. IEEE TPAMI, 2008

Semi-Global Matching

优化步骤

(1) 计算代价空间；(AD, BT, Census，MI, ….)

(2) 代价聚合

方向**r**上的路径代价

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) = & C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ & L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\ & L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\ & \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k) \end{aligned} \tag{13}$$

各个方向的总聚合代价

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) \tag{14}$$

(3) WTA

(4) 视差后处理

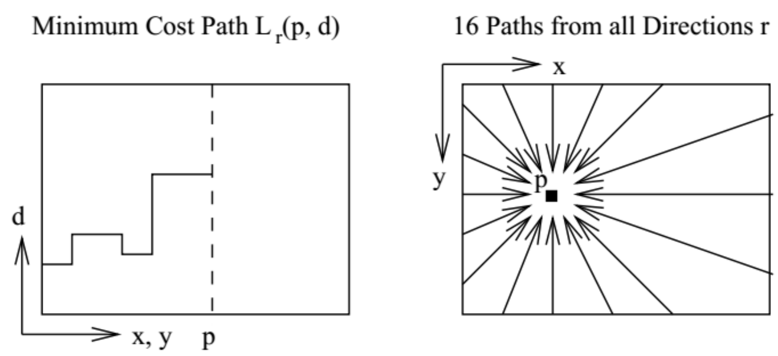
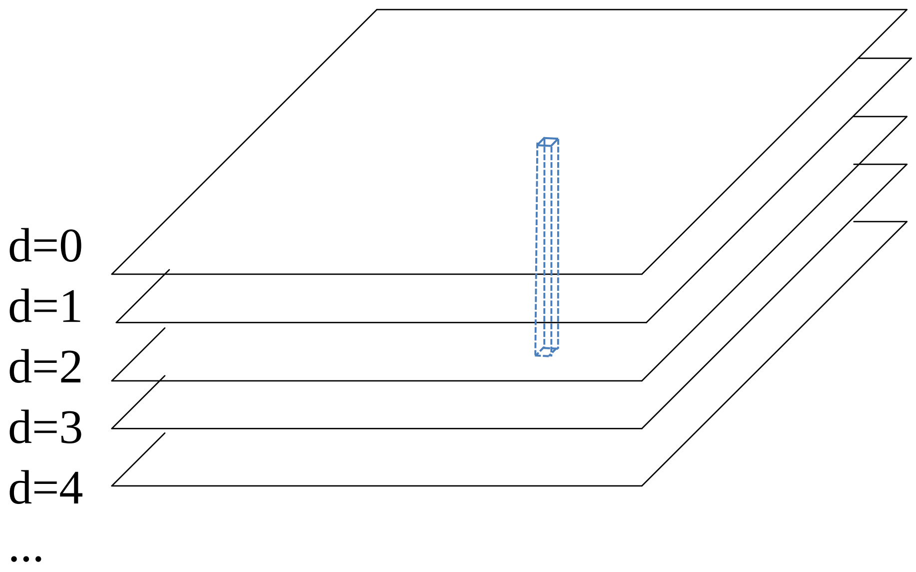


Fig. 2. Aggregation of costs in disparity space.

视差计算

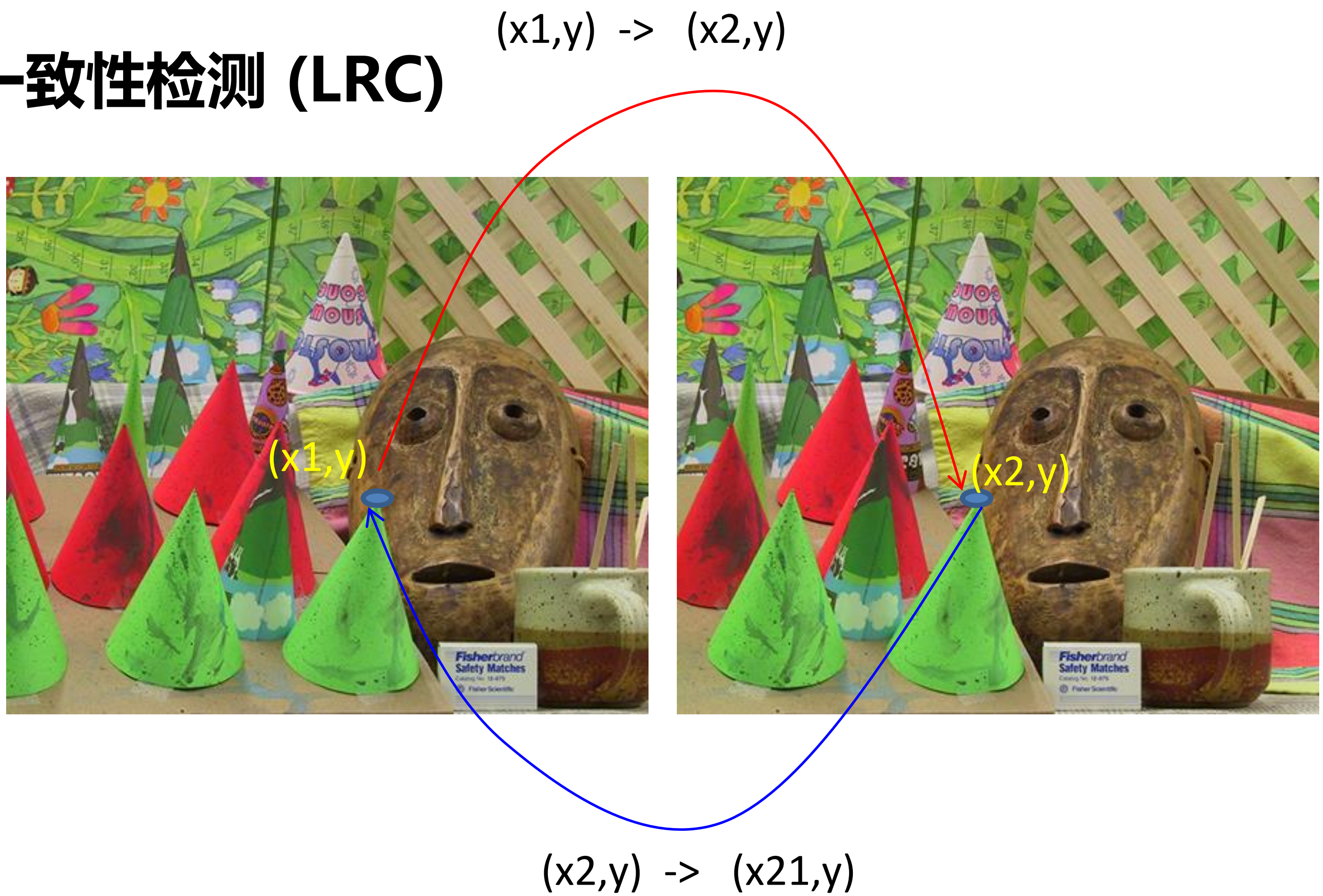
- Winner-Take-All (WTA)
- Disparity propagation (PatchMatch)



视差优化/后处理

- 左右一致性检测 (LRC)
- the minimum / the second minimum cost
- Speckle Filter
- 亚像素插值
- 中值滤波
- 空洞填充

左右一致性检测 (LRC)



如果$|x21-x1|>T$, 则表示视差$d=x1-x2$没有通过左右一致性检测。

Speckle Filter

为了移除噪声点，对视差图做一个连通区域提取（如果某相邻的2个像素的视差值之差小于某个预先设定的阈值，就可以认为这两个像素属于同一个区域）。

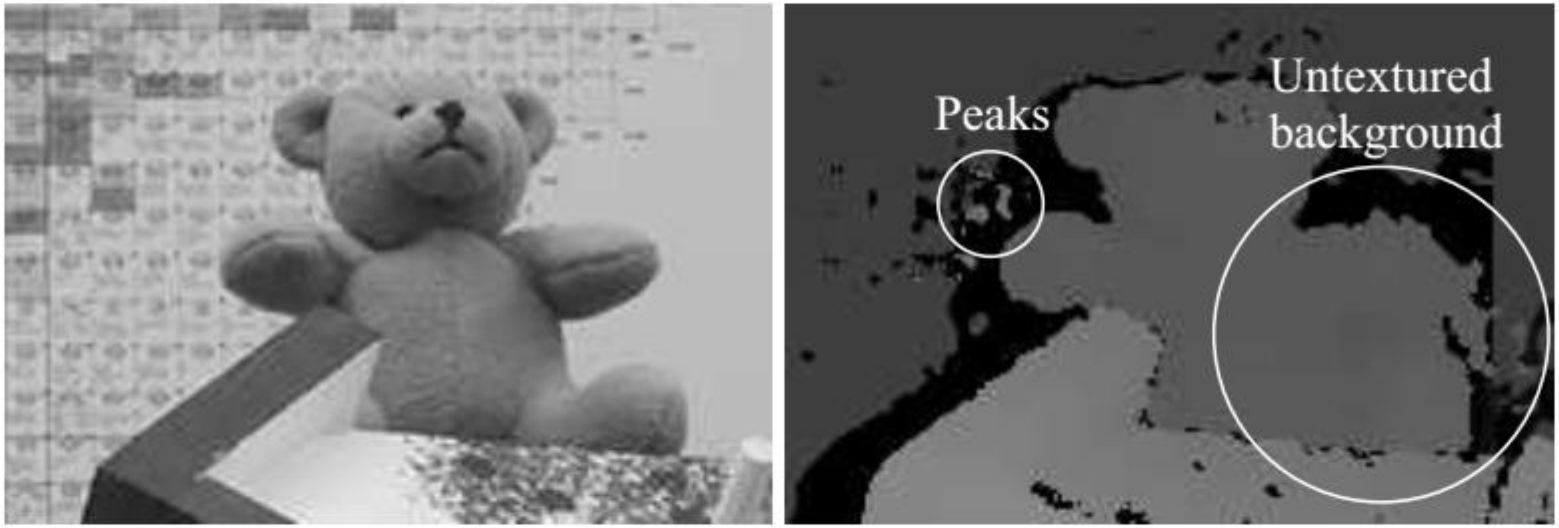
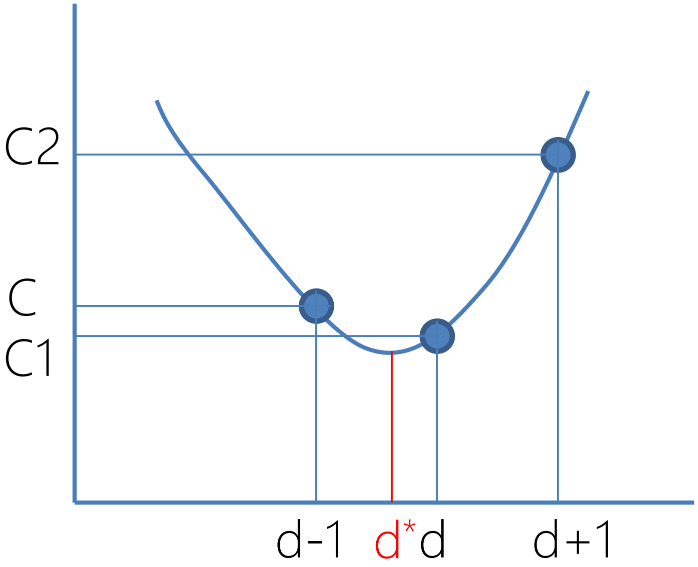


Fig. 4. Possible errors in disparity images (black is invalid).

[Hirschmuller 2008]

亚像素插值

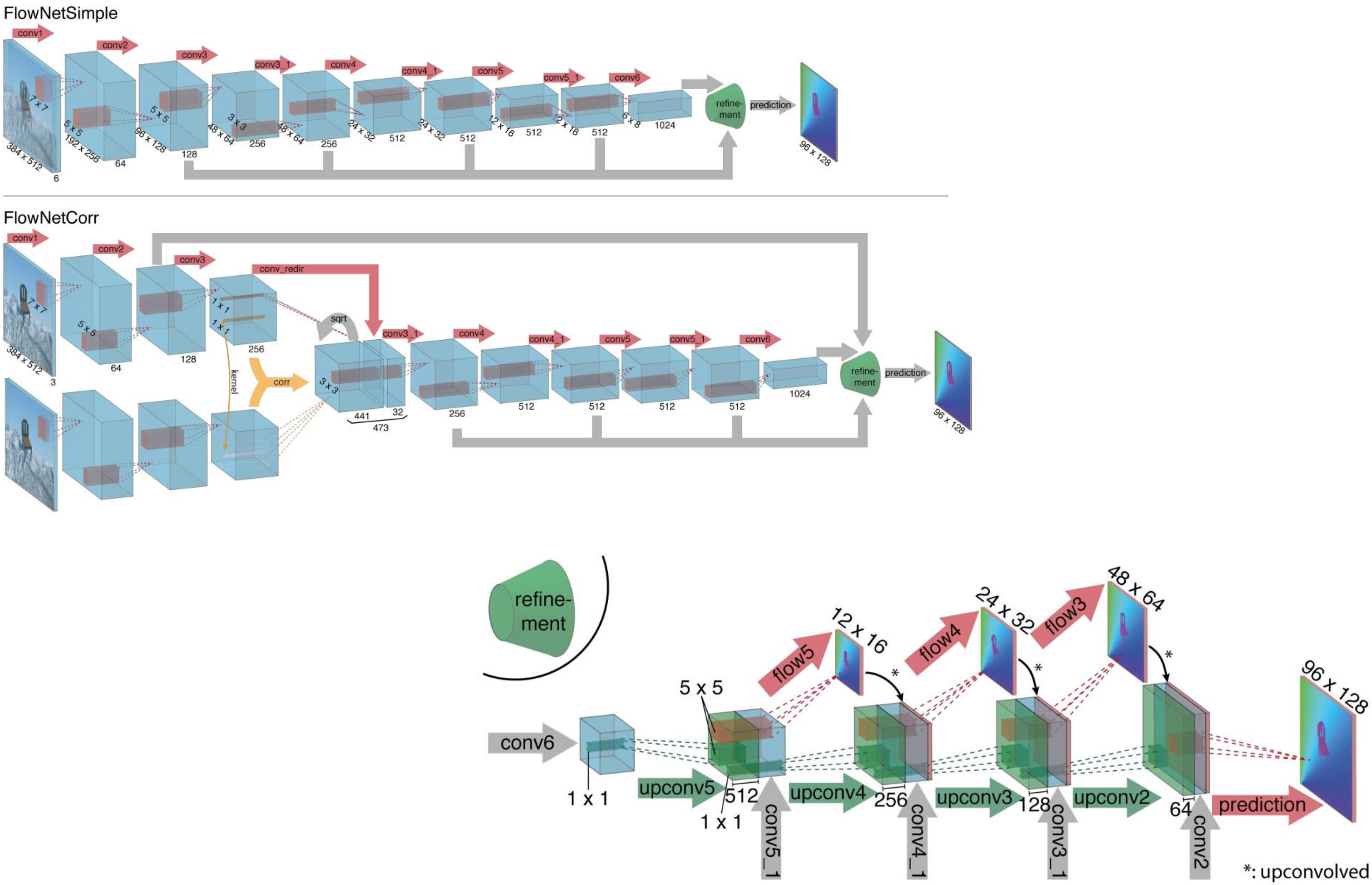


- 抛物线插值
- 线性插值

端到端视差计算网络

- Disp-Net (2016)
- GC-Net (2017)
- iRestNet (2018)
- PSM-Net (2018)
- Stereo-Net (2018)
- GA-Net (2019)
- EdgeStereo (2020)

Disp-Net (2016)



FlowNet: Learning Optical Flow with Convolutional Networks. ICCV 2015

Disp-Net (2016)

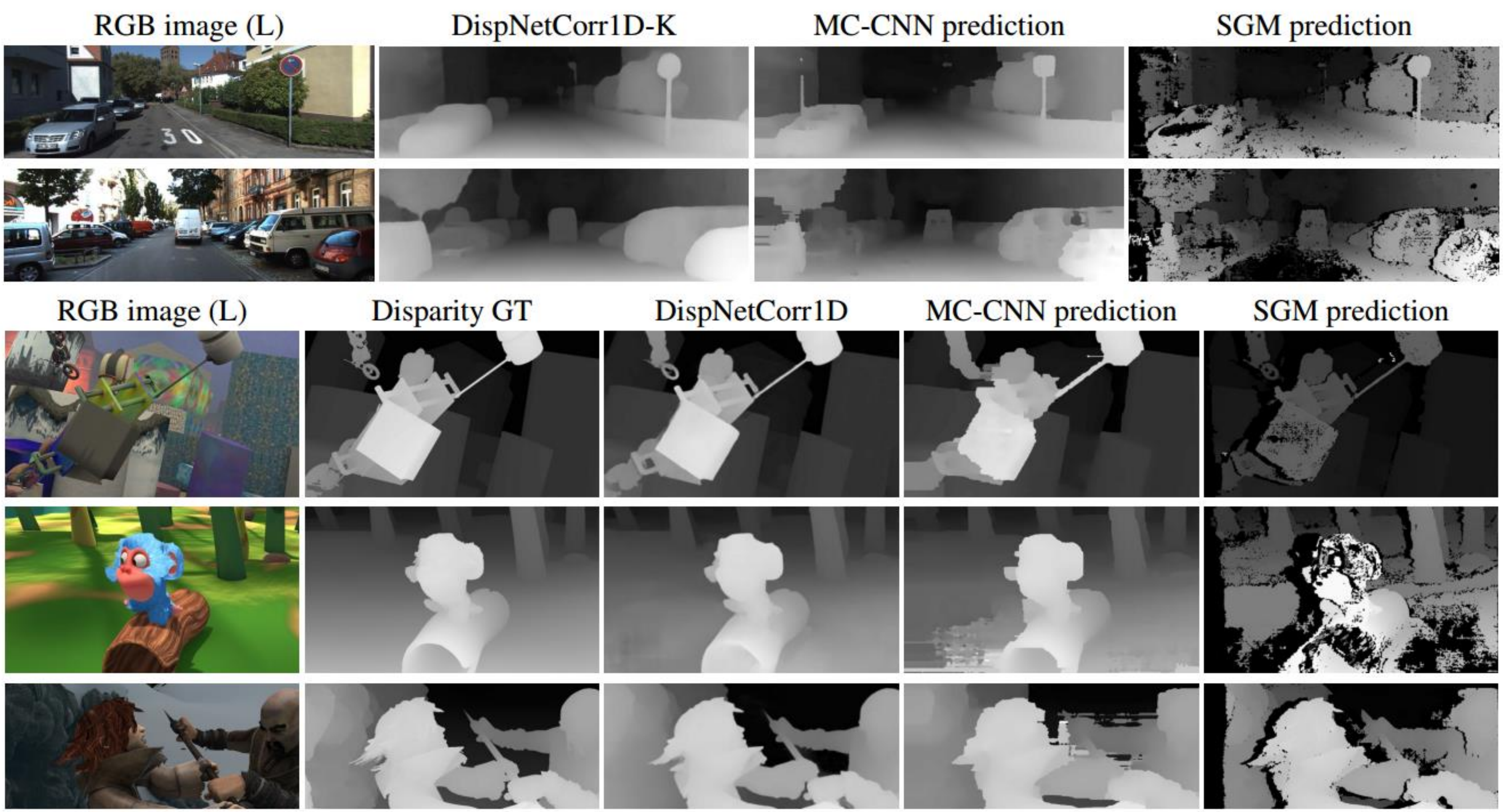


Figure 7. Disparity Results. Rows from top to bottom: KITTI 2012, KITTI 2015, FlyingThings3D, Monkaa, Sintel. Note how the DispNet prediction is basically noise-free.

立体视觉方法评测网站

- Middlebury Stereo 3.0

Hoops

Livingroom

Newkuba

- Kitti Stereo

- ETH3D

- Robust Vision Challenge

Stereo | Evaluation • Datasets • Code • Submit

Middlebury Stereo Evaluation - Version 3

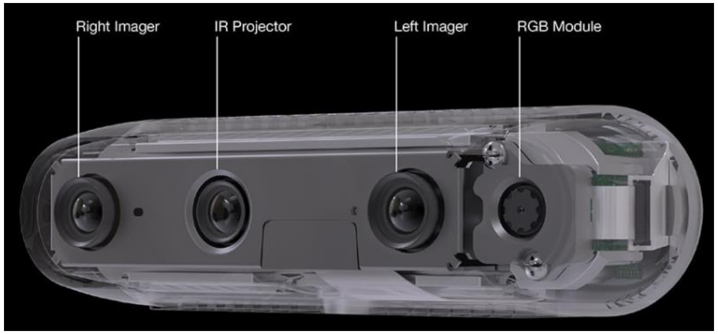
Mouseover the table cells to see the produced disparity map. Clicking a cell will blink the ground truth for comparison. To change the table type, click the links below. For more information, please see the description of new features.

Submit and evaluate your own results. See snapshots of previous results. See the evaluation v.2 (no longer active).

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 Reference list

Date	bad 2.0 (%) Name	Res	Weight Avg	Austr MP: 5.6 nd: 290 im0 im1 GT nonocc	AustrP MP: 5.6 nd: 290 im0 im1 GT nonocc	Bicyc2 MP: 5.6 nd: 250 im0 im1 GT nonocc	Class MP: 5.7 nd: 610 im0 im1 GT nonocc	ClassE MP: 5.7 nd: 610 im0 im1 GT nonocc	Compu MP: 1.5 nd: 256 im0 im1 GT nonocc	Crusa MP: 5.5 nd: 800 im0 im1 GT nonocc	CrusaP MP: 5.5 nd: 800 im0 im1 GT nonocc	Djemb MP: 5.7 nd: 320 im0 im1 GT nonocc	DjembL MP: 5.7 nd: 320 im0 im1 GT nonocc	Hoops MP: 5.7 nd: 410 im0 im1 GT nonocc	Livgrm MP: 5.9 nd: 320 im0 im1 GT nonocc	Nkuba MP: 5.5 nd: 570 im0 im1 GT nonocc	Plants MP: 5.6 nd: 320 im0 im1 GT nonocc	Stairs MP: 5.2 nd: 450 im0 im1 GT nonocc
05/26/18	NOSS_ROB	H	**5.01** 1	3.57 4	2.84 3	3.99 4	**1.93** 1	**5.15** 1	3.34 2	3.32 2	**3.15** 1	2.32 5	8.55 2	**7.45** 1	7.06 3	12.5 3	5.20 3	9.06 5
03/06/18	NOSS	H	5.04 2	3.57 4	2.84 3	3.99 4	**1.93** 1	**5.15** 1	3.34 2	3.32 2	**3.15** 1	2.32 5	8.55 2	**7.45** 1	7.06 3	12.5 3	5.20 3	10.0 11
06/22/17	LocalExp	H	5.43 3	3.65 6	2.87 5	**2.98** 1	1.99 3	5.59 5	3.37 5	3.48 6	3.35 5	2.05 2	10.3 7	9.75 6	8.57 7	14.4 17	5.40 8	9.55 8
03/09/19	3DMST-CM	H	5.47 4	4.10 11	3.37 9	2.99 2	2.95 13	7.63 10	4.55 9	**3.26** 1	3.95 8	2.16 3	10.2 6	8.28 3	**6.37** 1	13.2 7	5.86 12	9.35 6
06/01/18	NaN_ROB	H	5.73 5	**3.41** 1	2.90 6	3.69 3	2.33 4	5.32 4	3.35 4	3.49 7	3.31 4	2.31 4	10.3 8	10.8 8	9.64 10	15.3 23	5.45 9	9.89 9
01/24/17	3DMST	H	5.92 6	3.71 8	2.78 2	4.75 7	2.72 7	7.36 9	4.28 7	3.44 4	3.76 6	2.35 8	12.6 13	11.5 10	8.56 6	14.0 15	5.35 7	8.87 4
03/10/17	MC-CNN+TDSR	F	6.35 7	5.45 19	4.45 23	6.80 26	3.46 19	10.7 20	6.05 16	5.01 15	5.19 16	2.62 12	10.8 9	9.62 5	6.59 2	**11.4** 1	6.01 14	**7.04** 1
05/12/16	PMSC	H	6.71 8	3.46 2	**2.68** 1	6.19 21	2.54 5	6.92 7	4.54 8	3.96 9	4.04 10	2.37 9	13.1 15	12.3 11	12.2 13	16.2 29	5.88 13	10.8 15
02/26/19	FFE	H	6.72 9	3.68 7	4.52 24	5.43 15	2.79 9	11.0 21	3.93 6	3.57 8	4.42 13	2.34 7	21.5 29	12.9 13	8.65 8	13.8 11	5.33 6	9.35 6
10/19/16	LW-CNN	H	7.04 10	4.65 13	3.95 13	5.30 13	2.63 6	11.2 25	5.41 12	4.32 11	4.22 11	2.43 11	12.2 12	13.4 16	13.6 23	14.8 20	**4.72** 1	12.0 22
04/12/16	MeshStereoExt	H	7.08 11	4.41 12	3.98 15	5.40 14	3.17 15	10.0 14	6.23 18	4.62 12	4.77 14	3.49 28	12.7 14	12.4 12	10.4 11	14.5 18	7.80 23	8.85 3
10/12/17	FEN-D2DRR	H	7.23 12	4.68 14	4.11 17	5.03 10	3.03 14	8.42 11	6.05 16	4.90 14	5.32 17	3.20 25	11.5 11	14.1 19	13.4 22	13.9 13	5.06 2	14.3 29
05/28/16	APAP-Stereo	H	7.26 13	5.43 18	4.91 31	5.11 12	5.17 27	21.6 42	6.99 23	4.31 10	4.23 12	3.24 27	14.3 18	9.78 7	7.32 5	13.4 9	6.30 15	8.46 2
03/11/18	SGM-Forest	H	7.37 14	4.71 15	3.69 12	4.93 9	3.18 16	11.1 23	5.37 10	5.57 17	5.81 20	2.65 14	14.5 19	13.2 14	13.1 18	14.8 21	5.63 10	11.2 20
01/19/16	NTDE	H	7.44 15	5.72 25	4.36 22	5.92 18	2.83 11	10.4 16	5.71 13	5.30 16	5.54 18	2.40 10	13.5 16	14.1 19	12.6 15	13.9 14	6.39 17	12.2 23
05/31/18	CBMV_ROB	H	7.65 16	3.48 3	3.35 8	4.80 8	3.57 20	6.32 6	6.88 22	4.84 13	3.91 7	**1.97** 1	25.4 37	11.1 9	13.1 17	15.8 27	7.34 22	13.8 28
02/28/18	SDR	H	7.69 17	5.41 17	4.22 18	4.20 6	2.73 8	10.2 15	5.40 11	6.40 18	5.76 19	4.72 38	11.2 10	15.4 25	13.4 21	16.5 31	5.22 5	13.0 25
11/28/18	MSFNetA	H	7.96 18	6.21 30	4.26 19	6.02 20	3.66 22	8.95 12	6.28 19	8.41 25	8.06 24	2.62 12	17.9 25	13.9 18	11.9 12	11.5 2	8.00 25	10.6 14
10/29/18	Dense-CNN	H	7.98 19	5.59 22	4.54 26	5.83 17	2.79 9	10.4 17	5.78 14	8.26 23	8.84 26	2.66 15	15.6 21	14.2 22	13.2 19	13.2 8	6.30 15	11.1 18
08/28/15	MC-CNN-acrt	H	8.08 20	5.59 22	4.55 28	5.96 19	2.83 11	11.4 26	5.81 15	8.32 24	8.89 28	2.71 16	16.3 22	14.1 21	13.2 20	13.0 6	6.40 18	11.1 18
11/03/15	MC-CNN+RBS	H	8.42 21	6.05 27	5.16 37	6.24 22	3.27 17	11.1 23	6.36 20	8.87 27	9.83 36	3.21 26	15.1 20	15.9 27	12.8 16	13.5 10	7.04 20	9.99 10
09/13/16	SNP-RSM	H	8.75 22	5.46 20	4.85 29	6.50 25	3.37 18	10.4 18	7.31 25	8.73 26	9.37 32	3.58 29	14.3 17	14.7 23	14.9 27	12.8 5	10.1 30	10.8 16
12/11/17	OVOD	H	8.87 23	4.74 16	3.64 11	5.51 16	4.82 26	12.8 31	6.51 21	9.91 30	9.96 37	3.13 22	16.6 23	14.8 24	14.1 25	15.4 24	6.92 19	13.2 26
01/21/16	MCCNN_Layout	H	8.94 24	5.53 21	5.63 43	5.06 11	3.59 21	12.6 30	7.23 24	7.53 22	8.86 27	5.79 48	23.0 32	13.6 17	15.0 28	14.7 19	5.85 11	10.4 12
01/26/16	MC-CNN-fst	H	9.47 25	7.35 33	5.07 35	7.18 29	4.71 25	16.8 37	8.47 30	7.37 21	6.97 21	2.82 17	20.7 28	17.4 31	15.4 30	15.1 22	7.90 24	12.6 24

立体匹配算法的应用：Intel D435



- 兼容主动/被动模式
- RGB 1920 x 1080 @ 30 fps
- depth resolution up to 1280 x 720
- depth frame rate up to 90 fps

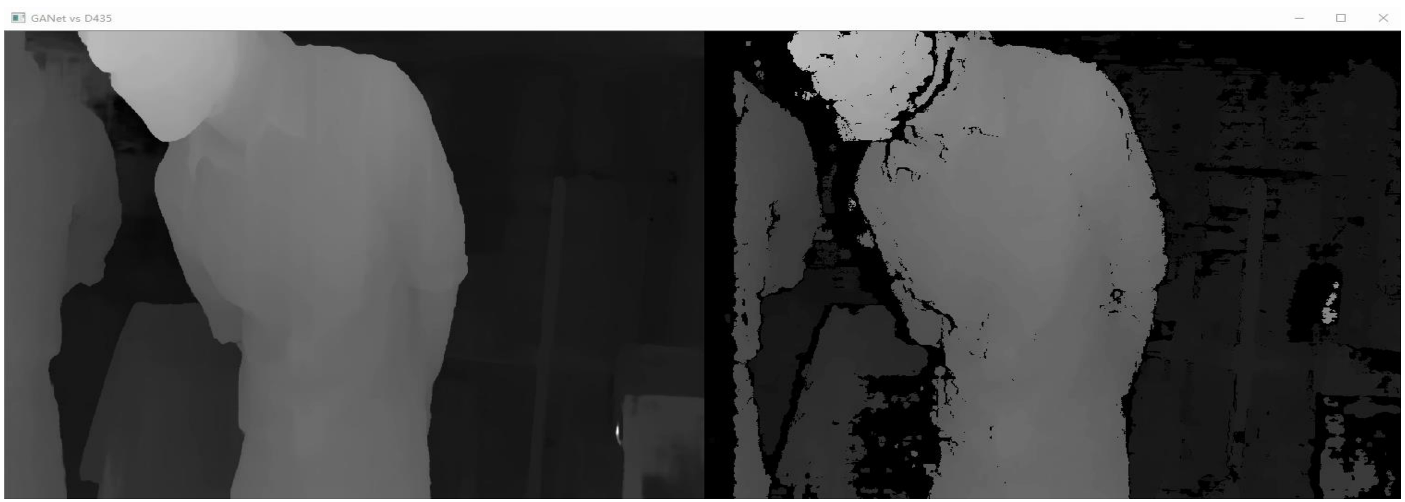
算法：

- AD-census代价计算
- 十字臂保边滤波
- SGM代价聚合
- second-peak threshold
- texture threshold

...

Leonid Keselman, et al. Intel R RealSenseTM Stereoscopic Depth Cameras. CVPRW. 2017.

D435 vs 深度学习方法（被动模式）



参考文献

1. [Yoon 2006] Yoon K J, Kweon I S. Adaptive support-weight approach for correspondence search. IEEE TPAMI, 2006.
2. [Wang 2006] Wang L, Liao M, Gong M, et al. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. 3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE, 2006.
3. [Hirschmuller 2008] Hirschmuller H. Stereo processing by semiglobal matching and mutual information. IEEE TPAMI, 2008.
4. [Zhang 2009] Zhang Ke, Lu J, Lafruit G. Cross-based local stereo matching using orthogonal integral images. IEEE TCSVT, 2009.
5. [He 2010] K. He, J. Sun, and X. Tang. Guided Image Filtering. ECCV 2010.
6. [Mei 2011]Mei X, Sun X, Zhou M, et al. On building an accurate stereo matching system on graphics hardware. ICCV Workshops. 2011.
7. [Rhemann 2011] Rhemann C, Hosni A, Bleyer M. Fast cost-volume filtering for visual correspondence and beyond. IEEE CVPR, 2011.
8. [Bleyer 2011] Bleyer M, Rhemann C, Rother C. PatchMatch Stereo-Stereo Matching with Slanted Support Windows. Bmvc. 2011.
9. [Yang 2012] Q X Yang. A non-local cost aggregation method for stereo matching. CVPR 2012.
10. [Kowalczuk 2013] Kowalczuk J, Psota E T, Perez L C. Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences. IEEE TCSVT, 2013.
11. [Ma 2013] Ma Z, He K, Wei Y, et al. Constant time weighted median filtering for stereo matching and beyond. ICCV 2013.
12. [Zhang 2014] Zhang Kang, Fang Y, Min D, et al. Cross-scale cost aggregation for stereo matching. CVPR 2014.
13. Yang Q. Hardware-efficient bilateral filtering for stereo matching. IEEE TPAMI, 2014.
14. [Wang 2015] Wang W, Yan J, Xu N, et al. Real-time high-quality stereo vision system in FPGA. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(10): 1696-1708.
15. [Kim 2016] K.-R. Kim and C.-S. Kim. Adaptive smoothness constraints for efficient stereo matching using texture and edge information. ICIP 2016.
16. [Li 2016] Li A, Chen D, Liu Y. Coordinating multiple disparity proposals for stereo computation. CVPR 2016
17. [Zbontar 2016] Zbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research, 2016.
18. [Park 2017] Park H, Lee K M. Look wider to match image patches with convolutional neural networks. IEEE Signal Processing Letters, 2017.
19. [Li 2017] Li L, Yu X, Zhang S, et al. 3D cost aggregation with multiple minimum spanning trees for stereo matching. Applied optics, 2017
20. D. Scharstein, T. Taniai, and S. Sinha. Semi-global stereo matching with surface orientation priors. 3DV 2017.
21. Leonid Keselman, et al. Intel R RealSenseTM Stereoscopic Depth Cameras. CVPRW. 2017.