

立体图像矫正和匹配

王宇伟 21721099 计算机学院 计算机科学与技术 wangyuwei@zju.edu.cn 1888925199

叶智超 11721010 计算机学院 计算机科学与技术 503417472@qq.com 17682305034

摘要:

从图像中获取三维结构信息是计算机视觉中的一个基本问题，被广泛应用于三维重建，机器人导航，物体跟踪等领域。目前常用的有基于立体匹配（Stereo Matching），基于深度摄像头（RGBD），以及基于深度学习（Deep Learning）等方法。

立体匹配是通过寻找双目图像中的匹配点，通过两点间的几何约束获得该点的视差和深度。常用的匹配有局部匹配和全局匹配算法，局部匹配通过找某个距离最近的点作为匹配点，全局算法通过定义某个全局能量函数进行迭代优化，常用的全局算法有 Graph Cuts, Belief Propagation 等。

本文实现了双目图像的几种匹配方法，包括局部匹配以及基于 Graph Cut 的全局算法，并比较了其在结果和性能上的差异。

关键字

立体匹配，立体矫正，Graph Cut, Belief Propagation。

1 引言

从图像中获取三维结构信息是计算机视觉中的一个基本问题，被广泛应用于三维重建，机器人导航，物体跟踪等领域。目前常用的有基于立体匹配（Stereo Matching），基于深度摄像头（RGBD），以及基于深度学习（Deep Learning）等方法。

立体匹配是通过寻找双目图像中的匹配点，通过两点间的几何约束获得该点的视差和深度，以此进行深度恢复。常用的匹配有局部匹配和全局匹配算法，局部匹配通过找某个距离最近的点作为匹配点，全局算法通过定义某个全局能量函数进行迭代优化，常用的全局算法有 Graph Cuts, Belief Propagation 等。

局部算法利用像素点的邻域信息进行匹配，在某个范围内搜寻“距离最近”的点，作为匹配点对。此处的距离可以被重新定义，常用的距离有 SAD(Sum of Absolute Differences), SSD (Sum of Squared Differences), NCC (Normalized Cross Correlation)。由于整幅图像往往含有上百万个像素点，故在进行局部匹配前一般先会对双目图像进行立体矫正，使得相同像素点位于同一水平扫描线上，以此减小搜索范围提高算法效率。局部算法的优点是算法简单高效，而缺点是在面对重复纹理以及弱纹理区域时极易出现误匹配。

全局算法则采用优化能量函数的方法求出一个全局最优匹配。在能量优化方法方面，较传统的有模拟退火算法、最可靠有限算法等。比较主流的两种优化算法是基于图切割和置信度传递。

本文阐述了局部和全局的立体匹配方法的基本原理，着重对比了两者的优缺点，并总结了立体匹配当前的难题和可能的解决途径。

2 相关工作

国外在计算机立体视觉上的研究开展较早，Roy 最早将图割算法应用于立体匹配，并通过实验表明，图割算法能有效克服其他全局优化算法的缺点（如动态规划算法等生成视差图产生的横向条纹瑕疵），避免了视差在临近极线处不连续的问题。但该算法生成的视差图轮廓边缘模糊，视差层的区分度低。Geiger 等，针对高分辨率图像立体匹配运算时间长的问题，创造性的提出了使用强约束点（纹理或特征信息较为丰富）作为支撑点，在强约

束点之间通过三角剖分对视差图进行插值计算，结合 OpenMP 技术在通用 CPU 上实现了并行计算，操作简单易于搭建环境，在通用微型计算机上实现了实时立体匹配，但是匹配效果和基于全局优化的匹配算法有一定差距。

国内对于立体视觉的研究起步较晚，早期主要采用基于特征点匹配的方法，随着技术的进步，后序对立体匹配的改进工作主要集中在对全局优化算法性能和准确度的提升上。其中大部分方法采用对待匹配图像进行图像分割后，再结合能量最优化的方法进行立体匹配。如尹等采用均值平移算法将参考图像根据颜色信息快速聚类；之后计算初始视差图；将分割结果作为能量视差函数的一个参考项；最后采用图割算法求取使全局能量最小的视差最优分配。此种基于图像分割的立体匹配方法的理论基础认为，分割区域块内的视差变化是平滑的。因此与其他基于图像分割的立体匹配算法相比，此类算法可有效地处理大块低纹理区域，匹配精度高，更有利于估计视差图的边界。并且上述算法通过分割减少了匹配基元，使得运算速度更快，能够很好的解决的边界模糊和低纹理区域的误匹配问题。

3 方法

3.1 立体矫正（由叶智超负责）

由于整幅图像往往含有上百万个像素点，故在进行局部匹配前一般先会对双目图像进行立体矫正，使得相同点位于同一水平扫描线上，以此减小搜索范围以提高算法效率。

3.2 局部匹配方法

局部匹配方法基于假设“所有匹配点均在同一条水平扫描线上”。

矫正后的双目视觉的几何约束如图 1 所示。 Z 为物距， f 为焦距， B 为基线距离， d 为两个成像点在同一坐标系下的距离，即视差。容易证明，在焦距和基线距离已知时，物体的深度与其在双目图像中的视差成反比，因此我们可以用双目视差图来表示间接深度图。

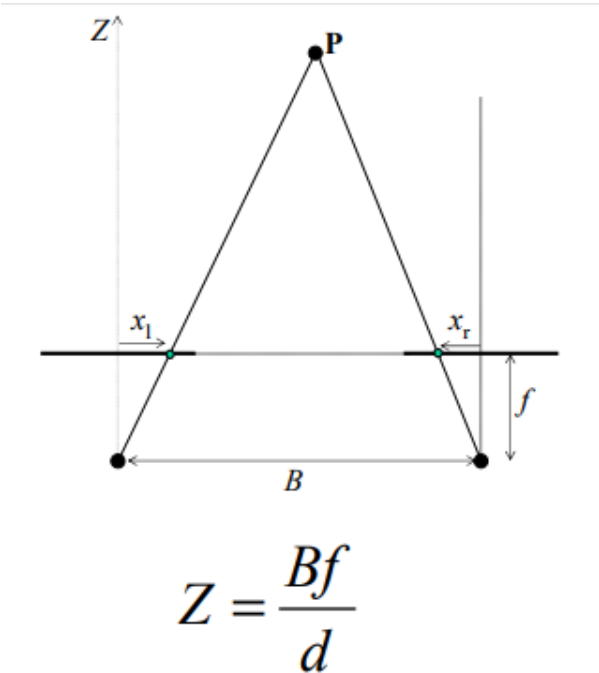


图 1

在匹配过程中，采用滑动窗口的方法，将某个像素点固定邻域内的像素值作为这一区域的向量值，通过比较两个向量之间的距离，采用 WTA (Winner Takes All) 的方式进行匹配。同时可以设定最小和最大搜索范围，以降低算法复杂度提高效率。

局部算法利用像素点的邻域信息进行匹配，在某个范围内搜寻“距离最近”的点，作为匹配点对。此处的距离可以被重新定义，常用的距离有 SAD (Sum of Absolute Differences), SSD (Sum of Squared Differences), NCC (Normalized Cross Correlation) 等。

$$\text{SAD: } \sum_{x,y \in W} \Psi(I_l(x,y), I_r(x-d,y)) = \sum_{x,y \in W} |I_l(x,y) - I_r(x-d,y)|,$$

$$\text{NCC: } \sum_{x,y \in W} \Psi(I_l(x,y), I_r(x-d,y)) = \sum_{x,y \in W} \frac{I_l(x,y) \cdot I_r(x-d,y) - \bar{I}_l \cdot \bar{I}_r}{\sigma_l \sigma_r}.$$

其中 \bar{I} 是对应窗口内的能量的均值， σ 是对应窗口内的能量的标准差。

ALGORITHM 1: 算法模板

输入：左右视图

输出：视差图

```

for x=1:xsize
  for y=1:ysize
    for d=dmin:dmax
      Sbest = max;
      S(d) = 0;
      for u = x-w:x+w
        for v = y-w:y+w
          S(d) = S(d) + y(d)
        if (S(d) < Sbest)
          Sbest = S(d)
      dbest(x,y) = d
  
```

3.3 基于 Graph Cuts 的全局匹配方法

全局匹配方法的能量函数一般直接包含所有像素，通过最优化这个能量函数解出每个像素点点的视差。我们参考了 Kolmogorov 提出的基于图切割的方法，设置的能量函数包括数据项 (Data Term), 遮挡项 (Occlusion Term) 平滑项 (Smooth Term) 和单值项 (Uniqueness Term) 四部分：

$$E(f) = E_{\text{data}}(f) + E_{\text{occlusion}}(f) + E_{\text{smoothness}}(f) + E_{\text{uniqueness}}(f).$$

数据项代表左右视图中对应像素的颜色差异

$$E_{\text{data}}(f) := \sum_{a, f(a)=1} D(a) = \sum_a D(a) \cdot 1(f(a) = 1),$$

灰度图像计算公式

$$D_d(p, q) := T(|I_1(p) - I_2(q)|)^d,$$

彩色图像计算公式

$$D_d(p, q) := \frac{T(|I_1^R(p) - I_2^R(q)|)^d + T(|I_1^G(p) - I_2^G(q)|)^d + T(|I_1^B(p) - I_2^B(q)|)^d}{3},$$

遮挡项计算公式

$$E_{\text{occlusion}}(f) := \sum_{a, f(a)=0} K = \sum_a K \cdot 1(f(a)=0) = K \times \#\mathcal{A} - \sum_a K \cdot 1(f(a)=1).$$

平滑项计算公式

$$E_{\text{smoothness}}(f) := \sum_{a_1 \sim a_2} V_{a_1, a_2} \cdot 1(f(a_1) \neq f(a_2)),$$

其中

$$V_{a_1, a_2} := \begin{cases} \lambda_1 = 3\lambda & \text{if } \max(|I_1(p_1) - I_1(p_2)|, |I_2(q_1) - I_2(q_2)|) < 8 \\ \lambda_2 = \lambda & \text{otherwise.} \end{cases}$$

单值项计算公式

$$E_{\text{uniqueness}}(f) := \sum_{\substack{a_1 \bullet (p, q_1) \\ a_2 \bullet (p, q_2) \\ q_1 \neq q_2}} \infty \cdot 1(f(a_1) = f(a_2) = 1) + \sum_{\substack{a_1 \bullet (p_1, q) \\ a_2 \bullet (p_2, q) \\ p_1 \neq p_2}} \infty \cdot 1(f(a_1) = f(a_2) = 1).$$

4 实验结果与分析

4.1 数据集

采用 <http://vision.middlebury.edu/stereo> 提供的数据集 (Quarter resolution, up to 750 x 500, disp <= 200)

4.2 实验环境和配置

Win10 & Visual Studio 2015 & i5 & 8G

4.3 实验方法

采用了 3 种方法对数据集进行了深度估计，分别是基于 SAD，NCC 匹配的局部算法以及基于 Graph Cut 的全局优化算法。并将实验结果打包上传至 <http://vision.middlebury.edu/stereo> 进行在线评测。

4.4 实验结果

图 2 为几个算法在提交后的在线评测结果，采用 average error 作为评测指标，更能反应深度恢复的整体效果 (NCC_X 表示滑动窗口大小为 X)

avgerr (pixels)		Res	Weight Avg																
Date	Name			Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlaytP	Recyc	Shelvs	Teddy	Vintge	
				MP: 5.7 nd: 290 im0im1 GT nonocc	MP: 1.5 nd: 256 im0im1 GT nonocc	MP: 5.2 nd: 640 im0im1 GT nonocc	MP: 5.9 nd: 280 im0im1 GT nonocc	MP: 5.9 nd: 280 im0im1 GT nonocc	MP: 5.4 nd: 260 im0im1 GT nonocc	MP: 5.4 nd: 260 im0im1 GT nonocc	MP: 5.7 nd: 300 im0im1 GT nonocc	MP: 5.3 nd: 330 im0im1 GT nonocc	MP: 5 nd: 290 im0im1 GT nonocc	MP: 5 nd: 290 im0im1 GT nonocc	MP: 5.6 nd: 260 im0im1 GT nonocc	MP: 5.9 nd: 240 im0im1 GT nonocc	MP: 2.7 nd: 256 im0im1 GT nonocc	MP: 5.5 nd: 760 im0im1 GT nonocc	
⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	⇅	
02/05/18	<div><div></div>SAD</div>	Q	67.6 71	80.3 72	85.2 72	65.0 71	57.8 71	94.8 72	61.0 71	92.1 72	48.6 71	78.5 72	69.9 71	57.7 71	65.4 71	76.3 72	37.0 71	67.1 69	
02/05/18	<div><div></div>NCC_5</div>	Q	25.5 71	21.7 68	18.0 71	59.3 69	14.7 71	14.5 67	20.0 72	36.2 71	18.3 68	28.0 69	31.1 67	20.4 72	18.9 71	25.7 71	11.6 72	81.7 72	
02/05/18	<div><div></div>NCC_7</div>	Q	21.8 69	15.8 66	17.8 71	50.9 68	12.1 70	11.9 65	16.3 70	33.6 70	17.1 68	24.1 68	27.3 66	17.1 71	13.0 67	22.7 70	9.42 71	73.3 71	
02/05/18	<div><div></div>NCC_9</div>	Q	21.0 68	13.7 65	18.6 71	50.3 68	11.9 68	11.6 65	14.9 69	32.5 70	18.3 68	22.6 68	26.3 65	15.9 70	11.3 67	21.0 69	9.46 71	71.2 70	
02/05/18	<div><div></div>NCC_11</div>	Q	21.3 68	13.0 64	20.1 71	52.1 68	12.0 68	11.7 65	14.2 69	32.0 68	20.2 70	21.9 68	27.1 66	15.7 70	11.0 67	20.0 69	10.0 71	70.7 70	
02/05/18	<div><div></div>GC</div>	Q	21.2 68	28.1 72	41.8 72	17.9 49	5.15 64	78.6 72	6.68 62	58.7 72	9.35 63	13.2 66	11.2 40	7.84 65	4.79 61	12.6 62	4.81 67	24.1 63	
02/05/18	<div><div></div>GC</div>	Q	13.2 66	4.74 58	13.8 70	17.9 49	5.15 64	27.2 69	6.68 62	64.1 72	9.35 63	13.2 66	11.2 40	7.84 65	4.79 61	12.6 62	4.81 67	24.1 63	

图 2 算法结果比较

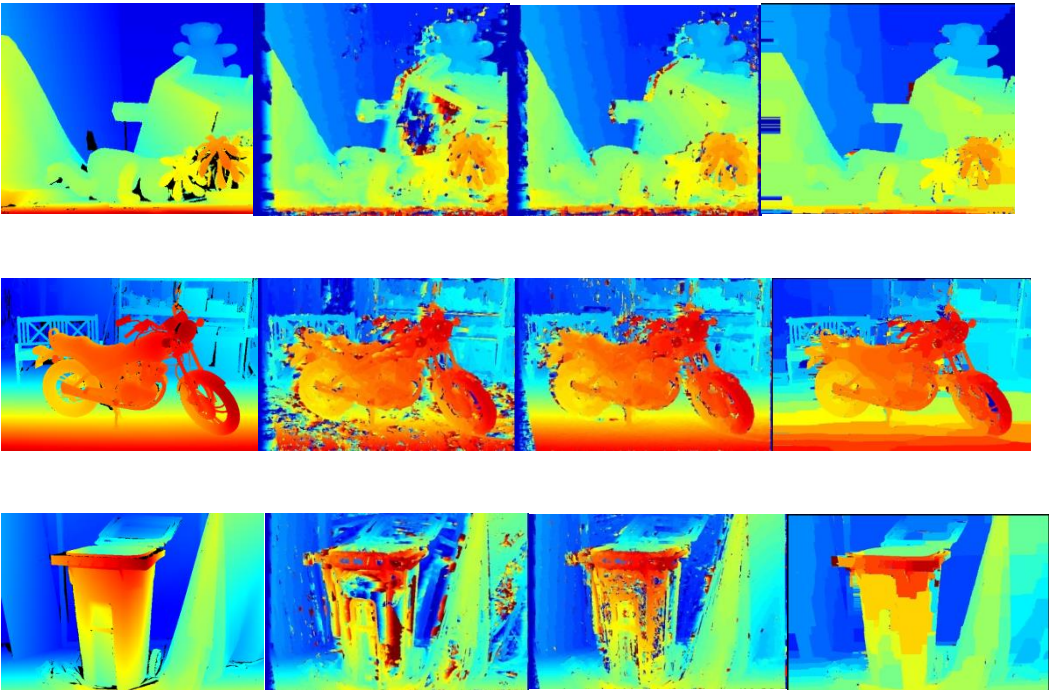


图 3 从左至右分别为 Ground Truth，基于 SAD 和 NCC 的局部算法，和基于 GC 实现的全局算法的深度估计图

4.5 实验分析

不难发现，NCC 的实验结果明显好于 SAD。

通过比较 SAD 效果特别差的几组图片，如 ArtL, MotorE, PianoL 等，发现其对光照变化较为敏感，而 NCC 在计算前窗口向量被归一化（Normalize）到同一尺度，仅仅比较角度大小，使得局部的线性（如光照等）影响被忽略，结果较为鲁棒。

在 NCC 中窗口大小设置对匹配结果也有一定影响，窗口过小导致误匹配和噪音增多，图像不够平滑，窗口过大导致边缘效果不明显，因此设置合适的窗口大小可以提高匹配的准确度。从图表中可知，当窗口大小为 9 时，匹配准确度最高。

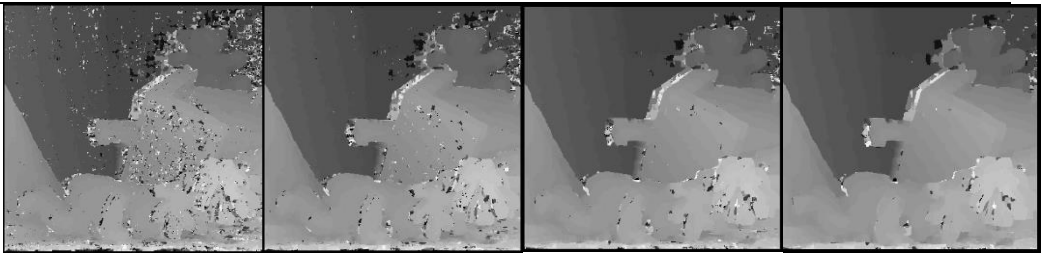


图 4 基于 NCC 算法的可视化深度估计图，从左至右窗口大小分别为 5，7，9，11

第一次提交的 GC 算法的平均误差大于效果最好的 NCC 算法。通过比较测试数据发现，大多数情况下 GC 算法优于 NCC 局部匹配，但在光照影响下，NCC 算法表现更好。通过对初始图片进行预处理调整整体亮度，进行了第二次试验，GC 算法的第二次实验结果整体优于 NCC 算法。仅有 PianoL 数据结果差于 NCC（局部光照影响）。

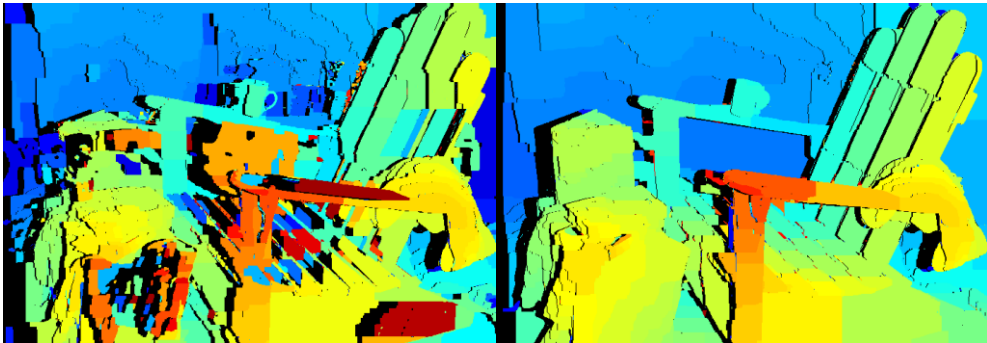


图 5 左图为未亮度预处理的深度图，右图为光照预处理后的深度图

此外，我们进行了算法效率的比较，如图 6 所示。其中 SAD 平均每组数据用时为 4.66 秒，NCC 算法平均每组数据用时为 13.4 秒，GC 算法每组数据用时为 68.8 秒。由于 SAD 计算较简单，故效率最高，而 NCC 的归一化计算较 SAD 复杂，效率次之，GC 算法采用迭代求最优解的方式，效率较低。

Date	time (sec)	Name	Res	Weight	Avg	Adiron		ArtL		Jadepl		Motor		MotorE		Piano		PianoL		Pipes		Playrm		Playt		PlaytP		Recyc		Shelvs		Teddy		Vintge	
						MP: 5.7	nd: 290	MP: 1.5	nd: 256	MP: 5.2	nd: 640	MP: 5.9	nd: 280	MP: 5.9	nd: 280	MP: 5.4	nd: 260	MP: 5.4	nd: 260	MP: 5.7	nd: 300	MP: 5.3	nd: 330	MP: 5	nd: 290	MP: 5	nd: 290	MP: 5.6	nd: 260	MP: 5.9	nd: 240	MP: 2.7	nd: 256	MP: 5.5	nd: 760
						im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	im0 im1	
						GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT	GT		
						nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	nonocc	
						↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	
02/05/18		SAD	Q	4.66	19	4.89	19	1.11	18	9.09	18	4.85	19	4.67	18	3.98	18	4.15	18	4.98	19	5.00	19	4.22	18	4.23	19	4.31	18	4.15	18	1.99	18	10.9	18
02/05/18		NCC_9	Q	13.4	22	14.0	23	3.19	22	27.5	23	13.8	22	13.8	22	12.0	21	12.0	22	14.3	23	14.1	22	11.8	22	11.8	22	11.9	21	11.5	22	5.45	21	30.7	25
02/05/18		GC	Q	68.8	34	74.7	37	13.6	29	127	33	74.9	35	68.5	33	60.6	34	71.7	37	71.2	35	74.9	35	68.8	35	68.3	35	62.9	36	73.3	36	23.7	27	141	36

图 6 算法效率比较

5 结论

局部匹配算法实现简单，算法复杂度较低，总体效果一般，对于弱纹理区域较容易出现误匹配。SAD 受光照影响较大，而 NCC 对整体和局部光照都不敏感。全局匹配算法复杂度较高，实现较复杂，但总体效果较好，预处理后对整体光照不敏感，但对局部光照仍然敏感。

在局部匹配算法中窗口大小设置也很重要，窗口过小容易出现较多误匹配，生成图像成碎片化，但边缘较清晰，而窗口过大图片较平滑，但边缘模糊，故设置一个适当大小的窗口较为合理。

全局匹配算法由于需要优化一个全局的能量函数，故对局部光照较敏感，此时效果不如局部匹配的 NCC 算法。

参考文献

- [1] Scharstein D, Szeliski R, Zabih R, et al. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International Journal of Computer Vision, 2001, 47(1): 7-42.
- [2] Boykov Y, Veksler O, Zabih R, et al. Fast approximate energy minimization via graph cuts[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(11): 1222-1239.
- [3] Sun J, Zheng N N, Shum H Y. Stereo matching using belief propagation[J]. IEEE Transactions on pattern analysis and machine intelligence, 2003, 25(7): 787-800.
- [4] V. Kolmogorov, Graph Based Algorithms for Scene Reconstruction from Two or More Views., PhD thesis, Cornell University, 2003.
- [5] V. Kolmogorov, P. Monasse, and P. Tan, "Kolmogorov and Zabih's graph cuts stereo matching algorithm," Image Processing On Line, vol. 4, pp. 220-251, 2014.