

一种新的基于特征点的立体匹配算法

陈肇君 戚飞虎

(上海交通大学计算机科学与工程系, 上海 200030)

摘要 目前,立体匹配是计算机视觉领域中最活跃的研究主题之一。为了快速并更精确的对特征点进行立体匹配,本文提出了一种新的基于特征点的立体匹配算法。该方法独立于特征点的检测算法,先以扫描线作为匹配单元,然后以鲁棒函数为匹配代价函数,最后用顺序约束对每一匹配单元的视差图进行检测与校正。实验证明,该方法的匹配精度高于传统的基于NCC(normalized cross-correlation)的立体匹配算法,并且运行时间快,可以应用于纯软件的基于特征点的立体视觉系统中。

关键词 立体匹配 鲁棒函数 特征点 顺序约束

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2005)11-1411-05

A New Feature-point-based Stereo Matching Algorithm

CHEN Pan-jun, QI Fei-hu

(Department of computer science and engineering, Shanghai Jiaotong University, Shanghai 200030)

Abstract Stereo matching is currently one of the most active research topics in domain of computer vision. A new feature-point-based stereo matching algorithm is proposed in this paper, in order to match feature points more efficiently and accurately. It is independent of feature detection algorithms, using the scan line as the matching unit, using the robust function as the matching cost function and using the ordering constraint to detect and correct errors in each matching unit in the disparity map. Experiments show that the matching accuracy of this algorithm is better than the traditional NCC (normalized cross-correlation) based stereo matching algorithm and the running time is quite fast. So, it can be used in the independent hardware feature-point-based stereo vision system.

Keywords stereo matching, robust function, feature point, ordering constraint

1 引言

立体匹配是立体视觉系统的重要组成部分,是近年计算机视觉领域备受关注的前沿方向和研究热点。它在机器人视觉、航空测绘、军事应用、医学诊断及工业检测中有着广泛的应用前景和潜在的经济价值。

前常见的立体匹配算法可以分为:稠密匹配和稀疏匹配两类。稠密匹配,即对图像中的每一像素求出视差值。近年对稠密匹配算法研究比较多,文献[1]将该类方法分成了局部方法和全局方法两类。其中局部方法包括,块匹配、特征匹配、基于梯度优化等方法;全局方法有动态规划、图切等方法。

局部方法虽实现复杂程度低,计算量小,但对于图像中的遮挡区域、低纹理区域以及周期性纹理区域比较敏感,易造成匹配精度不高;全局方法由于使用的约束是在整个图像或者扫描线上,其对于局部方法难以匹配的区域能够较好地处理,因此全局方法的匹配精度高,但由于该类方法实现复杂程度高,计算量大,所以目前能够应用于实际立体视觉系统中的稠密匹配算法往往要求相应的硬件支持,才能够达到实时的要求。对于稀疏匹配,即基于特征点的立体匹配,则只需对图像的某些像素求出视差,由于该类方法运行时间快,可以独立于硬件,因此很多应用于实际的纯软件的立体视觉系统都是基于特征点的。该类方法首先对两幅图像进行特征点提取,然

收稿日期:2005-08-16;改回日期:2005-09-08

第一作者简介:陈肇君(1980~),男。2003年获天津商学院工学学士学位,现为上海交通大学计算机科学与工程系硕士研究生。主要研究方向为计算机视觉、数字图像处理和模式识别。E-mail:chen_pj@sjtu.edu.cn

后根据特征点的视差梯度限制来求出各特征点的视差值^[2],但由于特征点提取算法不鲁棒,一幅图像的某个特征点,在另一幅图像与其相对应的特征点有可能不存在,因此,该方法的匹配精度不高。还有一些方法先求一幅图像的特征点,然后以该特征点为中心,根据极线约束和灰度一致性约束,在另一幅图像中找出该特征点的对应点。这类方法常用的匹配的代价函数有 SSD (sum of squared differences), SAD (sum of absolute differences), NCC (normalized cross-correlation) 等。由于这些函数对噪声敏感,因此这类方法的匹配精度也不高。为此本文提出了一种新的基于特征点的立体匹配算法,该方法独立于特征点的提取算法,先以扫描线作为匹配单元,然后以鲁棒函数为匹配代价函数,最后用顺序约束对每一匹配单元的视差图进行检测与校正。实验表明,该方法不仅精度高于传统的基于 NCC 的方法,而且在普通 PC 机上的运行时间能够达到实时的要求,可以用于实际的纯软件立体视觉系统中。

2 算法原理

2.1 算法框架

基于特征点的立体匹配算法如图 1 所示。

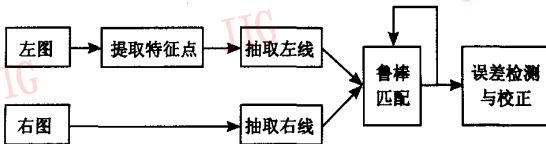


图 1 算法框架

Fig. 1 The framework of the algorithm

图 1 中左图和右图是经过校正的同一场景的左视图和右视图。

本文算法首先对左图进行特征点提取,且提取的方法应用了 SUSAN (smallest univalue segment assimilating nucleus) 算法^[3],该方法检测性能优于传统的角点检测器,其不仅能够很好地处理噪声,并且运行时间快,由于随后的鲁棒匹配没有应用特征点的特征信息,因此算法对于特征点提取方法是独立的,然后抽取左线,即以左图中的特征点为中心抽取水平线,长度为算法所定义的匹配长度;抽取右线,即以左图中的特征点的横坐标为右线的中心,按照特征点的纵坐标抽取水平线,长度与左线相同。

左右线抽取方法如图 2 所示。

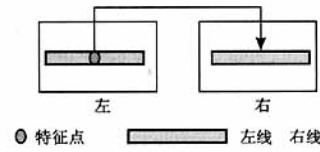


图 2 抽取左线和右线

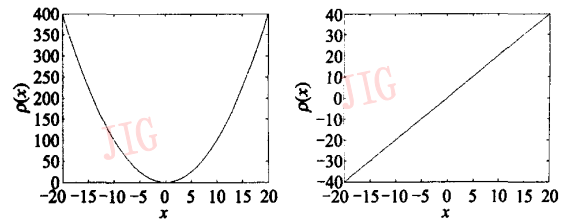
Fig. 2 Extracting the left line and right line

2.2 鲁棒匹配

常见的立体匹配算法的匹配代价函数有 SSD, SAD, NCC 等。现以 SSD 为例,该函数如下式所示:

$$SSD(x, y, d) = \sum_{x,y} [I_L(x, y) - I_R(x - d, y)]^2 \quad (1)$$

式中, $I_L(x, y)$ 为左图 (x, y) 像素点的灰度值, $I_R(x - d, y)$ 为右图 $(x - d, y)$ 像素点的灰度值, d 为视差值。该函数实际是一个二次函数 $\rho(x) = x^2$, 它的导数为 $\varphi(x) = 2x$, 两函数曲线如下图所示。



(a) $\rho(x) = x^2$ 函数曲线 (b) $\varphi(x) = 2x$ 函数曲线

图 3 $\rho(x) = x^2$ 函数及 $\varphi(x) = 2x$ 函数

Fig. 3 $\rho(x) = x^2$ function and $\varphi(x) = 2x$ function

从图 3(b) 可以看出, 由于对于立体匹配中的噪声影响是呈线性增加的, 且函数没有对这些噪声进行抑制, 因此 SSD 对于量化误差和噪声是敏感的。

为了抑制噪声的影响, 本文引入了文献[4]提出的鲁棒匹配代价函数 Lorentzian。它的函数形式及导数形式如下式所示:

$$\rho_\sigma(x) = \log\left(1 + \frac{1}{2}\left(\frac{x^2}{\sigma}\right)\right) \quad (2)$$

$$\varphi_\sigma(x) = \frac{2x}{2\sigma^2 + x^2}$$

式中, σ 为尺度参数。

两函数曲线如图 4 所示 (σ 取值为 2)。

从图 4(b) 可以看出, 该函数对噪声影响进行了平滑抑制, 使其影响趋向于零。

大家知道, 立体匹配算法是一个多约束的最佳

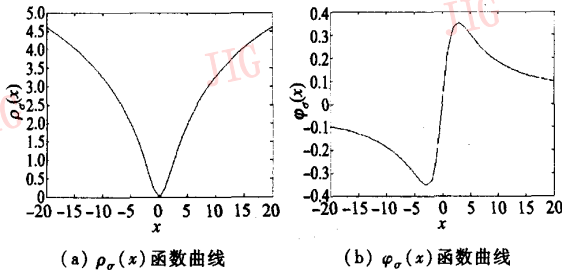


图 4 $\rho_\sigma(x)$ 函数及 $\varphi_\sigma(x)$ 函数

Fig. 4 $\rho_\sigma(x)$ function and $\varphi_\sigma(x)$ function

搜索问题。在该算法中,本文引入了灰度一致性约束,即左右两幅图像中对应点的灰度应该相等,其数学形式如下:

$$I_L(x, y) = I_R(x - d, y) \quad (3)$$

由灰度一致性约束可以导出以下代价函数:

$$E_D(d) = \sum_{x,y} \rho(I_L(x, y) - I_R(x - d, y)) \quad (4)$$

式中,下标 D 表明该代价函数是基于灰度一致性约束得到的, ρ 的形式是 Lorentzian 函数。

第 2 个约束,本文使用了视差平滑性约束,即像素点的视差值和其周围像素点的视差值相差不大。应用该约束的代价函数如下:

$$E_S(d) = \sum_{n \in G} \rho(d - d_n) \quad (5)$$

式中,下标 s 表明该代价函数是基于视差平滑性约束得到的, ρ 的形式是 Lorentzian 函数, G 是以视差值 d 的像素点为中心的上下左右像素点的集合。由式(4)、式(5),可以导出

$$E(d) = E_D(d) + \lambda E_S(d) \quad (6)$$

式中, λ 为控制灰度一致性约束和视差平滑性约束的比重。

本文还引入了极线约束,即对应点只能落在第 2 幅图像的极线上,这样通过该约束就可使 2 维搜索变为 1 维搜索。算法进行匹配时是对以特征点为中心的线进行匹配,此时代价函数 $E(d)$ 变为

$$E(d) = \sum_{x \in I_L} \left[\rho(I_L(x, y) - I_R(x - d_{x,y}, y)) + \lambda \sum_{n \in G} \rho(d_{x,y} - d_n) \right] \quad (7)$$

式中, I_L 为纵坐标为 y 的在左图中的左线像素点的集合, $d_{x,y}$ 是 (x, y) 像素点的视差值, G 是以 (x, y) 像素点为中心的前一个像素和后一个像素的集合。

由于代价函数 $E(d)$ 可能是非凸函数,并且在求特征点的视差值的同时就可获得该线所有像素的

视差值,这样就可以使用顺序约束来进一步提高匹配精度,因此本文采用同步超松弛算法来对该函数进行优化,该方法可以很快地同时获得该线所有像素的最优解,即视差值。

对 $E(d)$ 进行最小化的第 $n + 1$ 步迭代方程如下:

$$d_{x,y}^{(n+1)} = d_{x,y}^{(n)} - \omega \frac{1}{T(d_{x,y})} \frac{dE}{dd_{x,y}} \quad (8)$$

式中, ω 是超松弛参数,用于校正第 $n + 1$ 步中的 $d_{x,y}^{(n+1)}$ 的估计; $\frac{dE}{dd_{x,y}}$ 是 $E(d)$ 的一阶导数:

$$\frac{dE}{dd_{x,y}} = \sum_{x \in I_L} \left[\varphi(I_L(x, y) - I_R(x - d_{x,y}, y)) I_{R,x} + \lambda \sum_{n \in G} \varphi(d_{x,y} - d_n) \right] \quad (9)$$

其中, $\varphi(x) = \frac{\partial \rho}{\partial x}$, $I_{R,x} = \frac{\partial I_R(x - d_{x,y}, y)}{\partial x}$ 。

$T(d_{x,y})$ 是 $E(d)$ 的二阶导数的上界,其定义如下:

$$T(d_{x,y}) = \frac{\lambda I_{R,x}^2 + 4}{\sigma^2} \geq \frac{\partial^2 E}{\partial d_{x,y}^2} \quad \forall x \in I_L \quad (10)$$

2.3 扫描线视差值的误差检测与校正

通过鲁棒匹配,就获得了整个扫描线各像素的视差值,这样就可以利用顺序约束来对视差值进行误差检测与校正,以进一步提高匹配精度。

顺序约束即对于相似深度的表面及对应的特征点,一般以相同的顺序落在极线上。如图 5 所示。

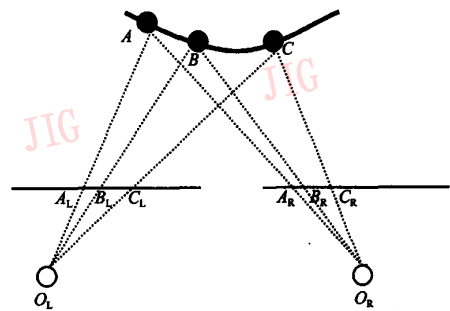


图 5 顺序约束

Fig. 5 Ordering constraint

本文应用了文献[5]的水平零交叉校正算法,当检测出误匹配点时,本文再使用鲁棒函数作为匹配代价函数,由于本文是以线作为匹配单元,优化算法则采用了穷举的方法来获得最值,因此这样耗费的时间并不是很多。

根据以上步骤,就获得了左线各像素点的视差

值,特征点的视差值即为左线中心像素点的视差值。

3 实验及其结果分析

用 4 组 5 对图片(如图 6 所示),分别是昏暗场景的一对图片、正常场景的一对图片、有遮挡区域的一对图片和纹理复杂场景的两对图片进行了测试实验,比较本文算法和基于 NCC 算法的性能。

在测试中,根据经验定义的匹配长度为 150 个像素,鲁棒匹配的初值为 0,并进行了分层处理,鲁棒匹配的次数为 3 次,前一次的结果是后一次的初值,每次匹配的迭代次数为 5 次。 λ 设定为 8, σ 设定为 2。

误差评判标准由下式定义:

$$R = \frac{N_1}{N} \quad (11)$$

其中, R 为误匹配率, N_1 为误匹配点数, N 为特征点总数。

测试结果和精确值相差两个像素即为误匹配点。

本文方法和 NCC 方法都是在配置为 CPU 赛扬 2.4GHz,内存 512MB 普通 PC 环境下运行的。测试结果如表 1 所示。



(a) 正常场景匹配 (b) 有遮挡场景匹配
(左图是 NCC 方法的实验结果, 误匹配率为 11%, 右图是本文方法的实验结果, 误匹配率为 5%)
(左图是 NCC 方法的实验结果, 误匹配率为 28%, 右图是本文方法的实验结果, 误匹配率为 10%)

图 6 部分实验结果

+ 误匹配点 + 正确匹配点

Fig. 6 Partial results of experiments

+ wrong matching point + right matching point

表 1 实验结果

Tab. 1 The results of experiments

图片对	特征点数 N	误匹配点数(N_1)		误匹配率(%)		运行时间(ms)	
		NCC	本文方法	NCC	本文方法	NCC	本文方法
Dark	398	81	59	20	15	417	212
Normal	375	41	20	11	5	394	203
Overlap	216	60	22	28	10	227	108
Texture1	688	202	155	29	23	727	396
Texture2	750	187	110	25	15	799	514
	平均值			22.6	13.6	512.8	286.6

从表 1 可以看出,本文方法的平均误匹配率低于基于 NCC 算法 9 个百分点,并且运行时间是 NCC 算法的 56%。

4 结论

本文在基于特征点的立体匹配框架中,由于引入了鲁棒代价函数,并用顺序约束对每一匹配单元的视差图进行检测与校正,从而使算法对于噪声以及遮挡区域,低纹理区域和周期性纹理区域得到较好的处理,不仅在匹配精度上高于传统的基于 NCC 的方法,并且在匹配中,是以线作为匹配单元,并使用了快速的优化算法同步超松弛算法,加快了运行时间,是基于 NCC 算法的 56%,因此该算法可以应用于纯软件的基于特征点的立体视觉系统中,并能够取得良好的匹配结果。

今后,进一步的研究是在匹配过程中引入特征点的特征信息,如用特征点的边缘方向、边缘强度组

成特征向量来进行匹配,同时研究新的优化算法,以便使匹配精度继续提高。

参考文献 (References)

- Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame Stereo correspondence algorithms[J]. International Journal of Computer Vision, 2002, 47(1/2/3):7~42.
- Pollard S B, Mayhew J E, Frisby G P. PMF: A stereo correspondence algorithm using a disparity gradient limit[J]. Perception, 1985, 14(4):449~470.
- Smith Stephen M, Brady J M. SUSAN—A new approach to low level image processing[J]. International Journal of Computer Vision, 1997, 23(1):45~78.
- Black M J, Anandan P. A framework for the robust estimation of optical flow[A]. In: IEEE International Conference on Computer Vision[C]. Berlin, Germany, 1993:231~236.
- JIA Bo, ZHANG Yu-jin, LIN Xing-gang. General and fast algorithm for disparity error detection and correction[J]. Journal of TsingHua University, 2000, 40(1):28~31. [贾波, 章毓晋, 林行刚. 视差图误差检测与校正的通用快速算法[J]. 清华大学学报, 2000, 40(1):28~31.]