(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108682026 B
(45)授权公告日 2021.08.06

(21)申请号 201810249294.X

(22)申请日 2018.03.22

(65)同一申请的已公布的文献号
申请公布号 CN 108682026 A

(43)申请公布日 2018.10.19

(73)专利权人 江大白
地址 安徽省合肥市经开区智能科技园E栋12层

(72)发明人 孙福明 杜仁鹏 蔡希彪

(74)专利代理机构 合肥市科融知识产权代理事务所(普通合伙) 34126
代理人 杨志胜

(51)Int.Cl.
G06T 7/33(2017.01)
G06T 7/55(2017.01)
G06T 7/80(2017.01)

(56)对比文件
CN 102611904 A,2012.07.25
CN 107301642 A,2017.10.27
CN 107392950 A,2017.11.24
CN 106355570 A,2017.01.25
CN 107301664 A,2017.10.27
CN 103985128 A,2014.08.13
CN 103996202 A,2014.08.20
CN 104867135 A,2015.08.26
CN 103996201 A,2014.08.20
US 2016173852 A1,2016.06.16
US 2016019437 A1,2016.01.21
林森等.双目视觉立体匹配技术研究现状和展望.《科学技术与工程》.2017,第17卷(第30期),
卢迪等.多种相似性测度结合的局部立体匹配算法.《机器人》.2016,第38卷(第1期),
Yingyun Yang等.A New Stereo Matching Algorithm Based on Adaptive Window.《2012 International Conference on Systems and Informatics (ICSAI 2012)》.2012,
Yong-Jun Chang等.Region Based Stereo Matching Method with Gradient and Distance Information.《2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)》.2017,

审查员 刘璐

权利要求书4页 说明书13页 附图4页

(54)发明名称

一种基于多匹配基元融合的双目视觉立体匹配方法

(57)摘要

双目视觉基于多匹配基元融合的立体匹配方法包括计算机、二个摄像机和多匹配基元融合的立体匹配方法。摄像机选用ZED双目摄像机。多匹配基元融合的立体匹配方法包含三个阶段:初始匹配代价阶段、代价函数聚合阶段和视差后处理阶段。在初始匹配代价阶段中,设计了包含颜色基元与梯度基元相融合的代价函数,其通过卡尔曼系数α进行自适应调节;在代价函数聚合阶段基于RGB颜色及距离关系设计了自适应匹配窗口,通过窗口内参考图像与代价函数之间相关性进行聚合;最后阶段,通过LRC左右一致性以及基于亚像素级别自适应权重中值滤波进行视差后处理。本发明对比经典自适应权重算法提高了算法的准确性与实时性,并在复杂的区域表现了较高的鲁棒性。

CN 108682026 B

1.一种双目视觉基于多匹配基元融合的立体匹配方法,包括计算机、二个摄像机和多匹配基元融合的立体匹配方法,其特征在于:

所述摄像机选用ZED双目摄像机,并对ZED双目摄像机作如下设置:

(1)、图像采集:

通过下载ZED SDK以及CUDA,并通过USB 3.0接口连接电脑;通过MATLAB内webcam函数连接ZED双目相机,并通过snapshot函数进行图片采集;

(2)、相机标定:

相机标定的目的,旨在获取准确摄像机内外参数;内参数主要包括左右镜头的焦距,基线距离;外参数主要包括两摄像机相对于世界坐标系的旋转矩阵,以及左右相机的相对平移矩阵;获得摄像机的默认内外参数;

(3)、图像参数设定:

通过标定参数进行极线校正使得采集到的左右图像满足极线约束条件;并通过ZED SDK内嵌ZED Explorer.exe插件进行参数的重设定;

(4)、立体匹配:

立体匹配作为双目视觉系统的核心部分;立体匹配的目的在于对采集到的左右图像进行成像点匹配,通过匹配点得到视差值,并获得场景的深度信息;

ZED双目摄像机可以装设在机器人或无人驾驶机上,双目摄取的实际场景经多匹配基元融合的立体匹配处理,能达到真实场景的目的,再通过设置在机器人或无人驾驶机上的计算机处理,向机器人或无人驾驶机控制与驱动系统发出导航指令;

所述多匹配基元融合的立体匹配方法,包括下述过程:

采用改进的ASW匹配方法,主要包括左右参考图像读入阶段、左右初始匹配代价阶段、左右代价函数聚合阶段和视差后处理4个阶段;所述视差后处理阶段,主要包含LRC左右一致性检测与滤波运算,其中:

1)初始匹配代价计算是:

ASW算法利用图像的灰度信息作为匹配代价计算的基元;通过对梯度基元和R、G、B三通道颜色基元的均值设置截断阈值,并通过卡尔曼滤波的思想融合像素的R、G、B颜色与梯度信息,通过控制系数α进行自适应调节从而做出改进;具体过程如下:

(1)分别设置颜色和梯度阈值t1、t2,计算初始代价e_g, e_c,如公式(10)(11)所示;

$$e_g(q,\bar{q}_d)=\min(|I_{rg}(q)-I_{lg}(\bar{q}_d)|,t1) \tag{10}$$

$$e_c(q,\bar{q}_d)=\min(\frac{1}{3}\cdot\sum_{c\in(r,g,b)}|I_c(q)-I_c(\bar{q}_d)|,t2) \tag{11}$$

(2)自适应调节α系数计算最终的初始匹配代价,最终的初始匹配代价如公式(12)所示;

$$e_0(q,\bar{q}_d)=\alpha\cdot e_c(q,\bar{q}_d)+(1-\alpha)\cdot e_g(q,\bar{q}_d) \tag{12}$$

2)、改进的自适应窗口扩展算法:

根据像素间的颜色与空间距离采用自适应窗口方法,已知待匹配的中心像素点p(x,y),在x和y方向上各邻域像素点分别为p(x-1,y),p(x+1,y)和p(x,y-1),p(x,y+1);不同于传统的自适应窗口扩展方法以中心像素点灰度信息进行像素扩展;以中心点R,G,B作为扩

展基元，当邻域像素与中心像素点三通道信息同时满足如下公式(13)条件进行窗口扩张；

$$I_{r,g,b}(x,y)-I_{r,g,b}(x-1,y)<t \quad (13)$$

t为预设颜色阈值，且t∈(0,1)；当图像中由于不连续的纹理而导致同一区域像素跳变时，很难使邻域像素三通道像素信息同时满足公式(13)；基于此特性，对传统的固定窗口做出改进；在邻域像素在满足公式(13)条件下进行窗口自适应扩张时，若场景中存在纹理重复区域导致窗口过大使得代价聚合时计算过于复杂，这不符合算法的实时性要求；根据图像几何特性对自适应窗口设置截断臂长；当满足如下公式(14)时对窗口臂长进行截断；

$$\sqrt{(p(x)-q(x))^2+(p(y)-q(y))^2}>L_{max} \quad (14)$$

其中，p(x)，p(y)为中心像素的横纵坐标值，q(x)，q(y)为邻域像素的坐标值；通过对middlebury平台下tsukuba，teddy，cones，venus四张测试图像实验设置最小臂长$L_{min}=5$，阈值$L_{max}=11$；自适应窗口，首先，对中心像素I(x,y)进行横向扩张，若最终的窗口臂长小于最短臂长，用最小臂长L_{min}代替原长度；在窗口扩张过程中，当邻域像素与中心像素空间距离大于截断阈值时，截断窗口；以中心点为参考，上下左右四个不同长度臂长L_u，L_d，L_l，L_r；对自适应窗口的行列大小取值分别如下公式(15)、(16)所示：

$Rows=L_u+L_d$ (15)

$Cows=L_l+L_r$ (16)

3)、匹配代价聚合算法：

计算匹配窗口内待匹配像素的R、G、B三通道像素值与匹配代价函数卷积的均值，如公式(17)所示；

$$m_{ce}(p)=\frac{\sum_{q\in N_p} I_c(q)\cdot e(q)}{n} \quad (17)$$

其中，p为窗口中心点像素坐标，q为窗口内包括中心点各像素坐标，Np为自适应窗口，n为窗口内像素总数；在计算完各通道卷积函数的均值后，

计算其协方差函数，如公式(18)所示：

$$v_{ce}=m_{ce}-m_c\cdot m_e \quad (18)$$

上式中，m_c，m_e分别为自适应窗口内三通道元素与匹配代价函数的均值，具体公式如下公式(19)、(20)所示：

$$m_c(p)=\frac{\sum_{q\in N_p} I_c(q)}{n} \quad (19)$$

$$m_e(p)=\frac{\sum_{q\in N_p} e_c(q)}{n} \quad (20)$$

然后，计算参考图像自适应窗口内三通道元素组成的方差矩阵θ，具体描述如下公式(21)所示；

$$\theta=\begin{bmatrix} s_{rr} & s_{rg} & s_{rb} \\ s_{rg} & s_{gg} & s_{gb} \\ s_{rb} & s_{gb} & s_{bb} \end{bmatrix} \quad (21)$$

其中，矩阵θ各元素计算如公式(22)所示：

$$s_{\substack{cc\\c\in\{r,g,b\}}}(p)=\frac{\sum_{q\in N_p} I_c(q)\cdot I_c(q)}{n}-\sum_{q\in N_p} I_c(q)\cdot\sum_{q\in N_p} I_c(q) \tag{22}$$

通过式(17)-(20)可得系数矩阵如公式(23)所示：

$$\underset{m\in\{1,2,3\}}{\gamma_{kn}}=\frac{v_{ce}}{\theta} \tag{23}$$

由于v_{ce}(c∈{r,g,b})为1*3的向量，可得γ_{kn}的值包含三通道的向量；当计算完相关系数γ_{kn}之后，通过各点代价函数的均值减去参考图像三通道像素与相关系数的卷积，使得左右两幅图像的匹配代价更加独立；最终，初始代价函数如下公式(24)所示：

$$e_0(p,\bar{p}_d)=m_c-(\sum_{\substack{q\in N_p,c\in\{r,g,b\}\\m\in\{1,2,3\}}} I_c(q)\cdot\gamma_{kn}(q)) \tag{24}$$

由于γ_{kn}为三通道向量，因此n∈(1,2,3)；在计算完最终的匹配代价之后，通过自适应窗口进行匹配代价聚合，具体公式如公式(25)所示；

$$E(d)=\frac{\sum e_0(q,\bar{q}_d)}{n}+\sum_{\substack{q\in N_p,c\in\{r,g,b\}\\m\in\{1,2,3\}}} I_c(q)\cdot\gamma_{kn}(q) \tag{25}$$

只对参考图像基础上进行聚合，并没有加入待匹配图像；实验结果表明在提高算法实时性的同时并没有降低其准确度；最后采用公式(26)，通过WTA(Winner-Takes-All)算法选取代价函数最小时的视差值为视差图的像素值；

$$D(p)=\arg\min_d(E(d)) \tag{26}$$

4)、视差后处理

(1)，LRC左右一致性检测

在立体匹配算法中，由于左右图像存在视差，遮挡问题一直不可避免；在获得最终视差图之前，本文首先采用LRC左右一致性算法进行视差后处理运算；

在以左图像为参考图像时计算得到视差d_l，以右图像为参考图像得到视差dr；当满足如下公式(27)条件：

$$|d_l-d_r|>\delta \tag{27}$$

δ为阈值，δ∈(0,1)；本文中δ取值为1；当满足左右视差差的绝对值大于δ，则认为是遮挡点；对遮挡点取左右视差中较小的视差值进行视差填充；

(2)，自适应权重中值滤波

在进行代价聚合算法之后，得到的视差图往往存在较多的椒盐噪声，有必要对图像进行中值滤波；然而传统的滤波往往忽视了像素之间的相关性；基于空间内像素之间颜色与距离的差异对窗口内像素赋予不同的权值，具体权值运算如公式(28)所示；

$$w(x,y)=\exp(\frac{-k1}{\gamma_c\cdot\gamma_c}+\frac{-k2}{\gamma_d\cdot\gamma_d}) \tag{28}$$

γ_c，γ_d为常数，通过试验获得γ_c=0.1，γ_d=9；k1,k2由中心像素与周围像素点在颜色空间与距离空间的差异得出，分别由下式(29)和(30)获得；

$$k1=\sqrt[3]{\sum_{q\in N_p} I_{c\in\{r,g,b\}}(p)-I_{c\in\{r,g,b\}}(q)} \quad (29)$$

$$k2=\sqrt[3]{\sum_{(x,y)\in N_p}(x-x')^2\cdot(y-y')^2} \quad (30)$$

窗口大小为19*19;在获得窗口内各像素的权值之后进行自适应中值滤波,具体过程如下:

(1)对窗口内除中心点外的各像素灰度值与各自的权值相乘,得到新的灰度值,利用式(31)计算获得;

$$I'(q)=w\cdot I(q) \quad (31)$$

(2)对窗口内包括中心点在内的各像素的新值进行排序,取位于中值附近最接近中心点的2个像素值I'(q1),I'(q2),取其均值得到新的亚像素级别灰度值代替原中心点像素的灰度值,由式(32)计算获得

$$I(p)=\frac{I'(q_1)+I'(q_2)}{2} \quad (32)。$$

一种基于多匹配基元融合的双目视觉立体匹配方法

技术领域

[0001] 本发明涉及一种基于多匹配基元融合的双目视觉立体匹配方法，可广泛应用于无人驾驶及机器人导航等技术领域。

背景技术

[0002] 立体匹配作为双目视觉中的核心技术广泛应用于无人驾驶及机器人导航等领域。其基本原理是通过双目相机获取场景的二维图，经匹配算法获取待匹配点的视差，进而获取场景的深度信息。立体匹配算法由于其高准确度特性受到学者的广泛关注。然而，由于自然场景中纹理的复杂性及场景深度的不连续性，一直制约着立体匹配算法的实际应用。

[0003] 目前主流的立体匹配算法可分为基于全局的立体匹配算法与基于局部的立体匹配算法。基于全局的立体匹配方法，由于涉及到能量函数的运算，计算复杂、效率低，难以满足实际需求。基于局部的立体匹配方法，在代价函数聚合阶段只对局部数据项进行操作，实时性高，但准确率相对较低，直到Yoon提出经典自适应支持权重算法(ASW)，其准确度得到质的提高，但由于其窗口固定，并不能反映图像的特征及纹理信息。

[0004] 局部立体匹配算法根据匹配基元的不同，又可细分为基于区域、基于特征以及基于相位的立体匹配方法。自适应支持权重方法由于匹配基元单一而且窗口固定，导致算法在复杂纹理区域准确度较低，并且由于复杂的权值计算导致算法实时性低。

[0005] 针对经典算法出现的问题，一些学者基于梯度及或颜色等多种匹配基元信息来表述图像的初始匹配代价，使得算法准确度得到了进一步的提高。然而，并没有摒弃复杂的权值计算，使得该算法的实时性仍较低且鲁棒性弱。

[0006] Lin等人使用线性函数来拟合原算法中的高斯权值函数，对原图像下采样得到高斯金字塔样本，并采用层次聚类算法计算代价聚合函数，在提高算法实时性的同时又提高了算法的准确度。但是，由于受到采样次数限制，采样后的图像过于模糊，导致该算法鲁棒性较差。

[0007] 一直以来，研究人员都对自适应窗口进行了研究，并通过利用参考图像与待匹配图像自适应窗口的重合度来摒弃一些明显错误的视差值，在代价聚合阶段摒弃了复杂权值的运算。但是，由于没有考虑到参考图像与匹配代价函数之间的相关性，这使得该算法的准确度并不优于经典算法。

[0008] 针对以上问题，为提高算法在实际场景应用中的鲁棒性，本发明在初始匹配代价阶段，将颜色匹配基元与梯度匹配基元相融合，并利用Middlebury平台符合自然场景条件的Cones图像对颜色与梯度匹配基元的比例系数α进行自适应调节；在代价函数聚合阶段，首先针对传统方法中固定窗口导致算法在复杂纹理条件下准确度低的问题，根据像素间的颜色信息与空间距离信息对中心像素点进行扩展，然后采用初始匹配代价函数和参考图像像素之间的相关性计算来代替传统的复杂能量函数和权值运算进行代价聚合，不仅提高了算法的准确度，同时也降低了算法的时间复杂度；最后，对所得视差图采取LRC左右一致性检测以及采用一种基于亚像素级自适应权重中值滤波运算，再次提高了算法的准确度。

发明内容

[0009] 本发明的目的，是针对现有技术存在的问题而提供一种双目视觉基于多匹配基元融合的立体匹配方法，采用本发明的方法所得到的视差图的平均误匹配率得到了极大的改善，图像，清晰而精确，大幅度提高了机器人或无人驾驶机运行轨迹的精度。

[0010] 采用的技术方案是：

[0011] 一种双目视觉基于多匹配基元融合的立体匹配方法，包括计算机、二个摄像机和多匹配基元融合的立体匹配方法，其特征在于：

[0012] 所述二个摄像机选用Astrolabes公司生产的ZED双目摄像机，并对ZED双目摄像机作如下设置：

[0013] 1)图像采集：

[0014] 通过下载ZED SDK以及CUDA，并通过USB3.0接口连接电脑。通过MATLAB内webcam函数连接ZED双目相机，并通过snapshot函数进行图片采集；

[0015] 2)相机标定：

[0016] 相机标定的目的，旨在获取准确摄像机内外参数。内参数主要包括左右镜头的焦距，基线距离；外参数主要包括两摄像机相对于世界坐标系的旋转矩阵，以及左右相机的相对平移矩阵。本发明从官方手册获得摄像机的默认内外参数；

[0017] 3)图像参数设定：

[0018] 通过标定参数进行极线校正使得采集到的左右图像满足极线约束条件。并通过ZED SDK内嵌ZED Explorer.exe插件进行参数的重设定；

[0019] 4)立体匹配：

[0020] 立体匹配作为双目视觉系统的核心部分。立体匹配的目的在于对采集到的左右图像进行成像点匹配，通过匹配点得到视差值，并获得场景的深度信息。

[0021] ZED双目摄像机不仅可以获取高质量的图像信息，而且具有体积小，功耗低的优点。特别是由于内嵌了ZED SDK函数库以及CUDA并行处理，大大降低了系统的处理时间。ZED双目摄像机可以装设在机器人或无人驾驶机上。双目摄取的实际场景，经多匹配基元融合的立体匹配处理，能达到真实场景的目的，再通过设置在机器人或无人驾驶机上的计算机处理，向机器人或无人驾驶机控制与驱动系统发出导航指令；或是通过设置在机器人或无人驾驶机外部计算机，经过局域网、以太网或是导线与计算机连接，经计算机处理，向机器人或无人驾驶机控制与驱动系统发出导航指令。大幅度提高了机器人或无人驾驶机运行轨迹的精度。

[0022] 所述多匹配基元融合的立体匹配模块方法，包括下述过程：

[0023] 本发明采用改进的ASW立体匹配方法。

[0024] 目前已知的ASW立体匹配算法是：

[0025] 在立体匹配算法中，一般默认两幅图像满足极线约束条件，即左右图像对应的匹配点处于两幅图像的同一行位置。ASW算法的核心在于使用支持权重来测量图像像素之间的相似性时，仅当相邻像素来自相同深度时，这些相邻像素的支持才是有效的，它具有与待匹配像素相同的视差。因此，窗口周围像素的支持权重w与窗口视差概率Pr成比例，如公式(1)所示：

[0026] $w(p,q) \propto Pr(d_p = d_q)$ (1)

[0027] 其中，p为待匹配像素，q为窗口内除待匹配像素外的其它像素，d为所求视差。w(p，q)与图像的颜色和距离相关，如公式(2)所示：

[0028] $w(p,q)=k\cdot f(\Delta c_{pq},\Delta g_{pq})$ (2)

[0029] 其中，Δc_{pq}，Δg_{pq}分别代表p和q两点分别在LAB颜色空间和几何空间上的距离，k为比例系数，具体数值通过实验获得，f为拉普拉斯核函数，二者相互独立，如公式(3)所示：

[0030] $f(\Delta c_{pq},\Delta g_{pq})=f(\Delta c_{pq})\cdot f(\Delta g_{pq})$ (3)

[0031] 其中Δc_{pq}，Δg_{pq}计算如公式(4)(5)所示：

[0032] $$\Delta c_{pq}=\sqrt{(L_p-L_q)^2+(a_p-a_q)^2+(b_p-b_q)^2} \quad (4)$$

[0033] $$\Delta g_{pq}=\sqrt{(p_x-q_x)^2+(p_y-q_y)^2} \quad (5)$$

[0034] $c_p=|L_p,a_p,b_p|$，$c_q=|L_q,a_q,b_q|$为Lab三通道颜色空间色度值，Lab颜色空间中的L分量用于表示像素的亮度，取值范围是[0,100]，表示从纯黑到纯白；a表示从红色到绿色的范围，取值范围是[127,-128]；b表示从黄色到蓝色的范围，取值范围是[127,-128]。(p_x,p_y)与(q_x,q_y)为几何空间的坐标值。使用拉普拉斯核函数定义分组强度，如公式(6)(7)所示。

[0035] $$f(\Delta c_{pq})=\exp(-\frac{\Delta c_{pq}}{\gamma_c}) \quad (6)$$

[0036] $$f(\Delta g_{pq})=\exp(-\frac{\Delta g_{pq}}{\gamma_p}) \quad (7)$$

[0037] γ_c，γ_p通过实验获得取$\gamma_c=7$，$\gamma_p=36$

[0038] (取值一般与窗口大小相关)。之后利用公式(8)进行代价聚合。

[0039] $$E(P,\overline{P_d})=\frac{\sum_{q\in N_p,\bar{q}_d\in N_{\bar{P}_d}} w(p,q)\cdot w(\bar{p}_d,\bar{q}_d)\cdot e_0(q,\bar{q}_d)}{\sum_{q\in N_p,\bar{q}_d\in N_{\bar{P}_d}} w(p,q)\cdot w(\bar{p}_d,\bar{q}_d)} \quad (8)$$

[0040] 其中初始匹配代价$e_0(q,\bar{q}_d)$如公式(9)所示：

[0041] $$e_0(q,\bar{q}_d)=\sum_{c\in\{r,g,b\}}\left|I_c(q)-I_c(\bar{q}_d)\right| \quad (9)$$

[0042] $I_c(q)$与$I_c(\bar{q}_d)$为参考图像与待匹配图像固定窗口内视差为d的两像素的灰度值。最后通过WTA(Winner-Takes-All)方法确定最终的视差图。

[0043] 本发明采用改进的ASW匹配方法主要包括：左右参考图像读入阶段、左右初始匹配代价阶段、左右代价函数聚合阶段和视差后处理4个阶段，所述视差后处理4个阶段，主要包含LRC左右一致性检测与滤波运算。其中：

[0044] 一、初始匹配代价计算是：

[0045] ASW算法利用图像的灰度信息作为匹配代价计算的基元。本发明通过对梯度基元和R、G、B三通道颜色基元的均值设置截断阈值，并通过卡尔曼滤波的思想融合像素的R、G、B颜色与梯度信息，通过控制系数α进行自适应调节从而做出改进。具体过程如下：

[0046] (1)分别设置颜色和梯度阈值t1、t2，计算初始代价e_g，e_c，如公式(10)(11)所示。

[0047] $$e_g(q,\bar{q}_d)=\min(|I_{rg}(q)-I_{lg}(\bar{q}_d)|,t1) \quad (10)$$

[0048] $$e_c(q,\bar{q}_d)=\min\left(\frac{1}{3}\cdot\sum_{c\in(r,g,b)}|I_c(q)-I_c(\bar{q}_d)|,t2\right) \quad (11)$$

[0049] (2)自适应调节α系数计算最终的初始匹配代价,最终的初始匹配代价如公式(12)所示。

[0050] $$e_0(q,\bar{q}_d)=\alpha\cdot e_c(q,\bar{q}_d)+(1-\alpha)\cdot e_g(q,\bar{q}_d) \quad (12)$$

[0051] 二、改进的自适应窗口扩展算法:

[0052] 传统的ASW匹配算法由于窗口固定,在处理复杂纹理区域时鲁棒性较差。本发明根据像素间的颜色与空间距离采用自适应窗口方法。已知待匹配的中心像素点p(x,y),在x和y方向上各邻域像素点分别为p(x-1,y),p(x+1,y)和p(x,y-1),p(x,y+1)。不同于传统的自适应窗口扩展方法以中心像素点灰度信息进行像素扩展,本发明以中心点R,G,B作为扩展基元,当邻域像素与中心像素点三通道信息同时满足如下公式(13)条件进行窗口扩张。

[0053] $I_{r,g,b}(x,y)-I_{r,g,b}(x-1,y)<t$ (13)

[0054] t为预设颜色阈值,且t∈(0,1)。当图像中由于不连续的纹理而导致同一区域像素跳变时,很难使邻域像素三通道像素信息同时满足公式(13)。基于此特性,本发明对传统的固定窗口做出改进。在邻域像素在满足公式(13)条件下进行窗口自适应扩张时,若场景中存在纹理重复区域导致窗口过大使得代价聚合时计算过于复杂,这不符合算法的实时性要求。本发明根据图像几何特性对自适应窗口设置截断臂长。当满足如下公式(14)时对窗口臂长进行截断。

[0055] $$\sqrt{(p(x)-q(x))^2+(p(y)-q(y))^2}>L_{\max} \quad (14)$$

[0056] 其中,p(x),p(y)为中心像素的横纵坐标值,q(x),q(y)为邻域像素的坐标值。通过对middlebury平台下tsukuba,teddy,cones,venus四张测试图像实验设置最小臂长L_{min}=5,阈值L_{max}=11。

[0057] 自适应窗口,如图3图所示。首先,对中心像素I(x,y)进行横向扩张,若最终的窗口臂长小于最短臂长,用最小臂长L_{min}代替原长度。在窗口扩张过程中,当邻域像素与中心像素空间距离大于截断阈值时,截断窗口。最后结果如图4图所示,以中心点为参考,上下左右四个不同长度臂长L_u,L_d,L_l,L_r。本发明对自适应窗口的行列大小取值分别如下公式(15)、(16)所示:

[0058] $Rows=L_u+L_d$ (15)

[0059] $Cows=L_l+L_r$ (16)

[0060] 三、匹配代价聚合算法:

[0061] 在计算完初始匹配代价之后,进行代价函数聚合,不同于经典算法复杂的权重运算。本发明通过代价函数与参考图像的相关性以及参考图像自身的相关性,对参考图像自适应窗口内方差以及参考图像与代价函数协方差进行计算,得出其相关函数,并用初始匹配代价函数减去相关函数之后进行代价聚合,大大挺高了算法的准确性与实时性。具体过程如下:

[0062] 首先,计算匹配窗口内待匹配像素的R、G、B三通道像素值与匹配代价函数卷积的均值,如公式(17)所示。

[0063] $$m_{\substack{ce\\c\in\{r,g,b\}}}(p)=\frac{\sum_{q\in N_p} I_c(q)\cdot e(q)}{n} \quad (17)$$

[0064] 其中，p为窗口中心点像素坐标，I(x,y)为窗口内包括中心点各像素坐标，I(x,y)为自适应窗口，n为窗口内像素总数。在计算完各通道卷积函数的均值后，计算其协方差函数，如公式(18)所示：

[0065] $$v_{\substack{ce\\c\in\{r,g,b\}}}=m_{ce}-m_c\cdot m_e \quad (18)$$

[0066] 上式中，m_c，m_e分别为自适应窗口内三通道元素与匹配代价函数的均值，具体公式如下公式(19)、(20)所示：

[0067] $$m_{\substack{c\\c\in\{r,g,b\}}}(p)=\frac{\sum_{q\in N_p} I_c(q)}{n} \quad (19)$$

[0068] $$m_{\substack{e\\c\in\{r,g,b\}}}(p)=\frac{\sum_{q\in N_p} e_c(q)}{n} \quad (20)$$

[0069] 然后，计算参考图像自适应窗口内三通道元素组成的方差矩阵θ，具体描述如下公式(21)所示。

[0070] $$\theta=\begin{bmatrix} s_{rr} & s_{rg} & s_{rb}\\ s_{rg} & s_{gg} & s_{gb}\\ s_{rb} & s_{gb} & s_{bb}\end{bmatrix} \quad (21)$$

[0071] 其中，矩阵θ各元素计算如公式(22)所示：

[0072] $$s_{\substack{cc\\c\in\{r,g,b\}}}(p)=\frac{\sum_{q\in N_p} I_c(q)\cdot I_c(q)}{n}-\sum_{q\in N_p} I_c(q)\cdot\sum_{q\in N_p} I_c(q) \quad (22)$$

[0073] 通过式(17)-(20)可得系数矩阵如公式(23)所示：

[0074] $$\gamma_{\substack{kn\\n\in\{1,2,3\}}}=\frac{v_{ce}}{\theta} \quad (23)$$

[0075] 由于v_{ce}(c∈{r,g,b})为1*3的向量，可得γ_{kn}的值包含三通道的向量。当计算完相关系数γ_{kn}之后，通过各点代价函数的均值减去参考图像三通道像素与相关系数的卷积，使得左右两幅图像的匹配代价更加独立。最终，初始代价函数如下公式(24)所示：

[0076] $$e_0(p,\overline{p}_d)=m_e-\left(\sum_{\substack{q\in N_p,c\in\{r,g,b\}\\ n\in(1,2,3)}} I_c(q)\cdot\gamma_{kn}(q)\right) \quad (24)$$

[0077] 由于γ_{kn}为三通道向量，因此n∈(1,2,3)。在计算完最终的匹配代价之后，通过自适应窗口进行匹配代价聚合，具体公式如公式(25)所示。

[0078] $$E(d)=\frac{\sum_{q\in N_p} e_0(q,\bar{q}_d)}{n}+\sum_{\substack{q\in N_p,c\in\{r,g,b\}\\ n\in(1,2,3)}} I_c(q)\cdot\gamma_{kn}(q) \quad (25)$$

[0079] 本发明只对参考图像基础上进行聚合，

[0080] 并没有加入待匹配图像。实验结果表明在提高算法实时性的同时并没有降低其准确度。最后采用公式(26)，通过WTA(Winner-Takes-A11)算法选取代价函数最小时的视差值为视差图的像素值。

[0081] $$D(p)=\arg\min_d(E(d)) \quad (26)$$

[0082] 四、视差后处理

[0083] 1，LRC左右一致性检测

[0084] 在立体匹配算法中，由于左右图像存在视差，遮挡问题一直不可避免。在获得最终视差图之前，本发明首先采用LRC左右一致性算法进行视差后处理运算。

[0085] 在以左图像为参考图像时计算得到视差d_l，以右图像为参考图像得到视差d_r。当满足如下公式(27)条件：

[0086] $|d_l-d_r|>\delta$ (27)

[0087] δ为阈值，$\delta\in(0,1)$。本发明中δ取值为1。当满足左右视差差的绝对值大于δ，则认为是遮挡点。对遮挡点取左右视差中较小的视差值进行视差填充。

[0088] 2，自适应权重中值滤波

[0089] 在进行代价聚合算法之后，得到的视差图往往存在较多的椒盐噪声，有必要对图像进行中值滤波。然而传统的滤波往往忽视了像素之间的相关性。本发明基于空间内像素之间颜色与距离的差异对窗口内像素赋予不同的权值，具体权值运算如公式(28)所示。

[0090] $$w(x,y)=\exp\left(\frac{-k1}{\gamma_c\cdot\gamma_c}+\frac{-k2}{\gamma_d\cdot\gamma_d}\right) \quad (28)$$

[0091] γ_c，γ_d为常数，通过试验获得$\gamma_c=0.1$，$\gamma_d=9$。k1，k2由中心像素与周围像素点在颜色空间与距离空间的差异得出，

[0092] 分别由下式(29)和(30)获得。

[0093] $$k1=\sqrt[3]{\sum_{q\in N_p}\sum_{c\in\{r,g,b\}} I_c(p)-I_c(q)} \quad (29)$$

[0094] $$k2=\sqrt[3]{\sum_{(x,y)\in N_p}(x-x')^2\cdot(y-y')^2} \quad (30)$$

[0095] 窗口大小为19*19。在获得窗口内各像素的权值之后进行自适应中值滤波。具体过程如下：

[0096] (1)对窗口内除中心点外的各像素灰度值与各自的权值相乘，得到新的灰度值，利用式(31)计算获得。

[0097] $I'(q)=w\cdot I(q)$ (31)

[0098] (2)对窗口内包括中心点在内的各像素的新值进行排序，取位于中值附近最接近中心点的2个像素值$I'(q_1)$，$I'(q_2)$，取其均值得到新的亚像素级别灰度值代替原中心点像素的灰度值，由式(32)计算获得。

[0099] $$I(p)=\frac{I^{'}(q_1)+I^{'}(q_2)}{2} \quad (32)$$

[0100] 五、算法描述

[0101] 算法的详细执行过程如表1所述。 表1

[0102]

算法1:基于多匹配基元立体匹配算法
输入：待匹配的左右参考图像 输出：视差图
1 初始化. 设置最小臂长L_{min}，臂长截断阈值L_{max}；初始匹配代价计算中颜色与梯度阈值t_1，t_2 卡尔曼系数α；视差后处理阶段权值系数γ_c，γ_d 2 读入参考图像并根据读入图像的不同设置不同的视察搜索范围1-d_{max} 3 for $d=1:d_{max}$ 4 根据公式（10）-（12）计算初始匹配代价 5 根据公式（13）-（16）计算代价聚合过程中的自适应窗口 6 根据公式（17）-（23）计算参考图像与代价函数的相关性，并根据公式（25）得到最终代价函数的聚合值 7 根据公式（26）WTA（Winner-Takes-All）选择最小聚合值对应的视差值为对应视差图像素最终灰度值 8 对视差图进行视差后处理操作得到最终视差图 9 结束

[0103] 六、实验结果

[0104] 本发明算法实验环境为Intel core i7-6700 3.5HZ CPU,12G内存,Matlab2016a平台。为了验证本发明所提算法的有效性,本发明首先在middlebury平台下对比了经典的自适应权重算法的视差图,如图6是Cones图像误匹配率对比图中的全部区域与深度不连续区域误匹配率对比视图。

[0105] 实验中自适应窗口最大最小臂长分别为L_{min}=5,L_{max}=11,最小梯度与颜色阈值分别为t1=2,t2=7,卡尔曼系数α=0.11。自适应权值滤波权值系数γ_c=0.1,γ_d=9,窗口为19*19。本发明对所得视差图在middlebury平台下三个测试指标:nonocc(非遮挡区域误匹配率),all(所有区域误匹配率),disc(深度不连续区域误匹配率)测试,对比了传统的自适应立体匹配算法以及De-Maeztu基于梯度的ASW改进算法,具体对比如下表2所示。

[0106] 表2.匹配算法评估表(单位:%)

[0107]

算法	Tsukuba			Venus			Teddy			Cones		
	Nocc	All	Disc	Nocc	All	Disc	Nocc	All	Disc	Nocc	All	Disc
Proposed	1.65	2.0	6.5	0.26	0.4	2.3	6.73	12	15	2.78	8	7.9
ASW	1.38	1.9	6.9	0.71	1.2	6.1	7.88	13	18	3.97	10	8.3
GradAW	2.26	2.6	9.0	0.99	1.4	4.9	8.00	13	18	2.61	7	7.4

[0108] 从表1可以看出对比经典的自适应权重及其改进算法，本发明使用基于多匹配基元融合的立体匹配算法在Venus，Teddy，Cones图像下，无论是在所有区域，非遮挡区域还是深度不连续区域场景下，本发明算法都有明显的优势。在光线较暗的Tsukuba图像下，在非遮挡区域，亦有一定的提升。对比基于单梯度基元的GradAdaptWgt算法，Tsukuba，Venus，Teddy在所有区域的误匹配率更低，且Cones图像的误匹配率与原GradAdaptWgt算法相当。本发明算法在middleb ury平台下进行平均误匹配率对比如表3所示，取得了相当可观的效果。

[0109] 表3.平均误匹配率对比

[0110]

算法	ASW	GradAdaptWgt	IterAdaptWgt	HCFilter	Proposed method
Average	6.67	6.55	6.08	5.67	5.54

[0111] 为提升算法的鲁棒性，本发明对比了颜色与梯度信息的卡尔曼系数比。选取符合自然场景下的Cones图像作对比，在不同的α系数下nonocc，all，disc各匹配条件误匹配率对比，其中α∈(0.11:0.02:0.29)。

[0112] 由上图结果可得，为实现立体匹配算法鲁棒性要求，我们在实际应用中将α的值设为0.23。而且相对于经典的自适应支持权重算法，本发明摒弃了复杂权重的运算，大大提高了算法的实时性。在时间复杂度上本发明对比了经典自适应支持权重算法如表4所示。

[0113] 表4.本发明算法与经典ASW算法时间复杂度对比(单位:ms)

[0114]

算法	Tsukuba	Venus	Teddy	Cones
Proposed method	507	983	2740	2718
ASW	23754	41560	86431	86076

[0115] 实验结果显示本发明算法大大降低了算法的时间复杂度。

[0116] 本发明优点：

[0117] 本发明通过融合颜色匹配基元与梯度匹配基元，并引入代价函数与参考图像相关性，提出了一种新颖的基于多匹配基元立体匹配方法。算法大大提升无人驾驶条件下场景信息的准确度，为获得更精确的深度图做出贡献，并且大大降低了算法时间复杂度，为无人

驾驶的实际应用提供了理论依据。更进一步提高机器人或无人机的导航精度。

附图说明

[0118] 图1是经过标定及图像设定之后zed双目相机采集的左右图像。

[0119] 图2是改进的ASW算法流程图。

[0120] 图3是自适应窗口图中的横向扩张图。

[0121] 图4是自适应窗口图中的窗口扩展最终效果图。

[0122] 图5是Cones图像误匹配率对比图中的非遮挡区域误匹配率视图。

[0123] 图6是Cones图像误匹配率对比图中的全部区域与深度不连续区域误匹配率对比视图。

具体实施方式

[0124] 一种双目视觉基于多匹配基元融合的立体匹配方法，包括计算机、二个摄像机和多匹配基元融合的立体匹配方法，其特征在于：

[0125] 1、摄像机选用Astrolabes公司生产的ZED双目摄像机。

[0126] ZED双目摄像机采集的视频流分辨率分四类，

[0127] 具体参数如下表1所示：

[0128] 表1:ZED输出视频流模式

[0129]

视频模式	输出分辨率（并排）	帧速率（fps）	视野
2.2K	4416*1242	15	宽
1080P	3840*1080	30，15	宽
720P	2560*720	60，30，15	超宽

[0130]

WVGA	1344*376	100，60，30，15	超宽

[0131] 2、图像采集：

[0132] 通过下载ZED SDK以及CUDA，并通过USB 3.0接口连接电脑。通过MATLAB内webcam函数连接ZED双目相机，并通过snapshot函数进行图片采集；

[0133] 3、相机标定：

[0134] 相机标定的目的，旨在获取准确摄像机内外参数。内参数主要包括左右镜头的焦距，基线距离；外参数主要包括两摄像机相对于世界坐标系的旋转矩阵，以及左右相机的相对平移矩阵。本发明从官方手册获得摄像机的默认内外参数；

[0135] 4、图像参数设定：

[0136] 通过标定参数进行极线校正使得采集到的左右图像满足极线约束条件。并通过ZED SDK内嵌ZED Explorer.exe插件进行参数的重设定；

[0137] 5、立体匹配：

[0138] 立体匹配作为双目视觉系统的核心部分。立体匹配的目的在于对采集到的左右图像进行成像点匹配，通过匹配点得到视差值，并获得场景的深度信息。

[0139] ZED双目摄像机可以装设在机器人或无人驾驶机上。双目摄取的实际场景，经多匹

配基元融合的立体匹配处理，能达到真实场景的目的，再通过设置在机器人或无人驾驶机上的计算机处理，向机器人或无人驾驶机控制与驱动系统发出导航指令。

[0140] 所述多匹配基元融合的立体匹配方法，包括下述过程：

[0141] 采用改进的本发明采用改进的ASW匹配方法：

[0142] 本发明提出的改进ASW算法流程，主要包括：左右参考图像读入阶段、左右初始匹配代价阶段、左右代价函数聚合阶段和视差后处理4个阶段，所述视差后处理4个阶段，主要包含LRC左右一致性检测与滤波运算。其中：

[0143] 一、初始匹配代价计算是：

[0144] ASW算法利用图像的灰度信息作为匹配代价计算的基元。本发明通过对梯度基元和R、G、B三通道颜色基元的均值设置截断阈值，并通过卡尔曼滤波的思想融合像素的R、G、B颜色与梯度信息，通过控制系数α进行自适应调节从而做出改进。具体过程如下：

[0145] (1)分别设置颜色和梯度阈值t1、t2，计算初始代价e_g，e_c，如公式(10)(11)所示。

[0146] $$e_g(q,\bar{q}_d)=\min(|I_{rg}(q)-I_{lg}(\bar{q}_d)|,t1) \quad (10)$$

[0147] $$e_c(q,\bar{q}_d)=\min(\frac{1}{3}\cdot\sum_{c\in(r,g,b)}|I_c(q)-I_c(\bar{q}_d)|,t2) \quad (11)$$

[0148] (2)自适应调节α系数计算最终的初始匹配代价，最终的初始匹配代价如公式(12)所示。

[0149] $$e_0(q,\bar{q}_d)=\alpha\cdot e_c(q,\bar{q}_d)+(1-\alpha)\cdot e_g(q,\bar{q}_d) \quad (12)$$

[0150] 二、改进的自适应窗口扩展算法：传统的ASW匹配算法由于窗口固定，在处理复杂纹理区域时鲁棒性较差。本发明根据像素间的颜色与空间距离采用自适应窗口方法。已知待匹配的中心像素点p(x,y)，在x和y方向上各邻域像素点分别为p(x-1,y)，p(x+1,y)和p(x,y-1)，p(x,y+1)。不同于传统的自适应窗口扩展方法以中心像素点灰度信息进行像素扩展，本发明以中心点R,G,B作为扩展基元，当邻域像素与中心像素点三通道信息同时满足如下公式(13)条件进行窗口扩张。

[0151] $I_{r,g,b}(x,y)-I_{r,g,b}(x-1,y)<t$ (13)

[0152] t为预设颜色阈值，且t∈(0,1)。当图像中由于不连续的纹理而导致同一区域像素跳变时，很难使邻域像素三通道像素信息同时满足公式(13)。基于此特性，本发明对传统的固定窗口做出改进。在邻域像素在满足公式(13)条件下进行窗口自适应扩张时，若场景中存在纹理重复区域导致窗口过大使得代价聚合时计算过于复杂，这不符合算法的实时性要求。本发明根据图像几何特性对自适应窗口设置截断臂长。当满足如下公式(14)时对窗口臂长进行截断。

[0153] $$\sqrt{(p(x)-q(x))^2+(p(y)-q(y))^2}>L_{max} \quad (14)$$

[0154] 其中，p(x)，p(y)为中心像素的横纵坐标值，q(x)，q(y)为邻域像素的坐标值。通过对middlebury平台下tsukuba,teddy,cones,venus四张测试图像实验设置最小臂长$L_{min}=5$，阈值$L_{max}=11$。

[0155] 自适应窗口，如图3图所示。首先，对中心像素I(x,y)进行横向扩张，若最终的窗口臂长小于最短臂长，用最小臂长L_{min}代替原长度。在窗口扩张过程中，当邻域像素与中心像素空间距离大于截断阈值时，截断窗口。最后结果如图4图所示，以中心点为参考，上下左右

四个不同长度臂长L_u,L_d,L_l,L_r。本发明对自适应窗口的行列大小取值分别如下公式(15)、(16)所示：

[0156] $Rows = L_u + L_d$ (15)

[0157] $Cows = L_l + L_r$ (16)

[0158] 三、匹配代价聚合算法：

[0159] 在计算完初始匹配代价之后，进行代价函数聚合，不同于经典算法复杂的权重运算。本发明通过文献[11]中代价函数与参考图像的相关性以及参考图像自身的相关性，对参考图像自适应窗口内方差以及参考图像与代价函数协方差进行计算，得出其相关函数，并用初始匹配代价函数减去相关函数之后进行代价聚合，大大挺高了算法的准确性与实时性。具体过程如下：

[0160] 首先，计算匹配窗口内待匹配像素的R、G、B三通道像素值与匹配代价函数卷积的均值，如公式(17)所示。

[0161] $$\underset{c\in\{r,g,b\}}{m_{ce}(p)} = \frac{\sum_{q\in N_p} I_c(q)\cdot e(q)}{n} \qquad (17)$$

[0162] 其中，p为窗口中心点像素坐标，q为窗口内包括中心点各像素坐标，N_p为自适应窗口，n为窗口内像素总数。在计算完各通道卷积函数的均值后，计算其协方差函数，如公式(18)所示：

[0163] $$\underset{c\in\{r,g,b\}}{v_{ce}} = m_{ce} - m_c \cdot m_e \qquad (18)$$

[0164] 上式中，m_c，m_e分别为自适应窗口内三通道元素与匹配代价函数的均值，具体公式如下公式(19)、(20)所示：

[0165] $$\underset{c\in\{r,g,b\}}{m_c(p)} = \frac{\sum_{q\in N_p} I_c(q)}{n} \qquad (19)$$

[0166] $$\underset{c\in\{r,g,b\}}{m_e(p)} = \frac{\sum_{q\in N_p} e_c(q)}{n} \qquad (20)$$

[0167] 然后，计算参考图像自适应窗口内三通道元素组成的方差矩阵θ，具体描述如下公式(21)所示。

[0168] $$\theta = \begin{bmatrix} s_{rr} & s_{rg} & s_{rb} \\ s_{rg} & s_{gg} & s_{gb} \\ s_{rb} & s_{gb} & s_{bb} \end{bmatrix} \qquad (21)$$

[0169] 其中，矩阵θ各元素计算如公式(22)所示：

[0170] $$\underset{c\in\{r,g,b\}}{s_{cc}(p)} = \frac{\sum_{q\in N_p} I_c(q)\cdot I_c(q)}{n} - \sum_{q\in N_p} I_c(q)\cdot \sum_{q\in N_p} I_c(q) \qquad (22)$$

[0171] 通过式(17)-(20)可得系数矩阵如公式(23)所示：

[0172] $$\underset{n\in\{1,2,3\}}{\gamma_{kn}} = \frac{v_{ce}}{\theta} \qquad (23)$$

[0173] 由于v_{ce}(c∈{r,g,b})为1*3的向量,可得γ_{kn}的值包含三通道的向量。当计算完相关系数γ_{kn}之后,通过各点代价函数的均值减去参考图像三通道像素与相关系数的卷积,使得左右两幅图像的匹配代价更加独立。最终,初始代价函数如下公式(24)所示:

[0174] $$e_0(p,\bar{p}_d)=m_e-\Big(\sum_{\substack{q\in N_p,c\in\{r,g,b\}\\ n\in(1,2,3)}} I_c(q)\cdot\gamma_{kn}(q)\Big) \qquad (24)$$

[0175] 由于γ_{kn}为三通道向量,因此n∈(1,2,3)。在计算完最终的匹配代价之后,通过自适应窗口进行匹配代价聚合,具体公式如公式(25)所示。

[0176] $$E(d)=\frac{\sum_{q\in N_p} e_0(q,\bar{q}_d)}{n}+\sum_{\substack{q\in N_p,c\in\{r,g,b\}\\ n\in(1,2,3)}} I_c(q)\cdot\gamma_{kn}(q) \qquad (25)$$

[0177] 本发明只对参考图像基础上进行聚合,

[0178] 并没有加入待匹配图像。实验结果表明在提高算法实时性的同时并没有降低其准确度。最后采用公式(26),通过WTA(Winner-Takes-All)算法选取代价函数最小时的视差值为视差图的像素值。

[0179] $$D(p)=\arg\min_d(E(d)) \qquad (26)$$

[0180] 四、视差后处理

[0181] 1,LRC左右一致性检测

[0182] 在立体匹配算法中,由于左右图像存在视差,遮挡问题一直不可避免。在获得最终视差图之前,本发明首先采用LRC左右一致性算法进行视差后处理运算。

[0183] 在以左图像为参考图像时计算得到视差d_l,以右图像为参考图像得到视差d_r。当满足如下公式(27)条件:

[0184] $|d_l-d_r|>\delta$ (27)

[0185] δ为阈值,δ∈(0,1)。本发明中δ取值为1。当满足左右视差差的绝对值大于δ,则认为是遮挡点。对遮挡点取左右视差中较小的视差值进行视差填充。

[0186] 2,自适应权重中值滤波

[0187] 在进行代价聚合算法之后,得到的视差图往往存在较多的椒盐噪声,有必要对图像进行中值滤波。然而传统的滤波往往忽视了像素之间的相关性。本发明基于空间内像素之间颜色与距离的差异对窗口内像素赋予不同的权值,具体权值运算如公式(28)所示。

[0188] $$w(x,y)=\exp\Big(\frac{-k1}{\gamma_c\cdot\gamma_c}+\frac{-k2}{\gamma_d\cdot\gamma_d}\Big) \qquad (28)$$

[0189] γ_c,γ_d为常数,通过试验获得$\gamma_c=0.1$,$\gamma_d=9$。k1,k2由中心像素与周围像素点在颜色空间与距离空间的差异得出,

[0190] 分别由下式(29)和(30)获得。

[0191] $$k1=\sqrt[3]{\sum_{q\in N_p}\ \sum_{c\in\{r,g,b\}} I_c(p)-I_c(q)} \qquad (29)$$

[0192] $$k2=\sqrt[3]{\sum_{(x,y)\in N_p}(x-x')^2\cdot(y-y')^2} \qquad (30)$$

[0193] 窗口大小为19*19。在获得窗口内各像素的权值之后进行自适应中值滤波。具体过

程如下：

[0194] (1)对窗口内除中心点外的各像素灰度值与各自的权值相乘，得到新的灰度值，利用式(31)计算获得。

[0195] $I'(q)=w\cdot I(q)$ (31)

[0196] (2)对窗口内包括中心点在内的各像素的新值进行排序，取位于中值附近最接近中心点的2个像素值$I'(q_1)$，$I'(q_2)$，取其均值得到新的亚像素级别灰度值代替原中心点像素的灰度值，由式(32)计算获得。

[0197] $$I(p)=\frac{I'(q_1)+I'(q_2)}{2} \quad (32)$$

图1

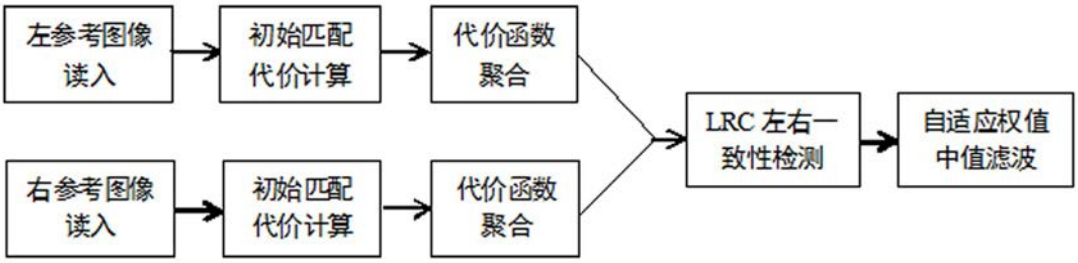
左参考图像读入
初始匹配代价计算
代价函数聚合
右参考图像读入
初始匹配代价计算
代价函数聚合
LRC 左右一致性检测
自适应权值中值滤波

图2

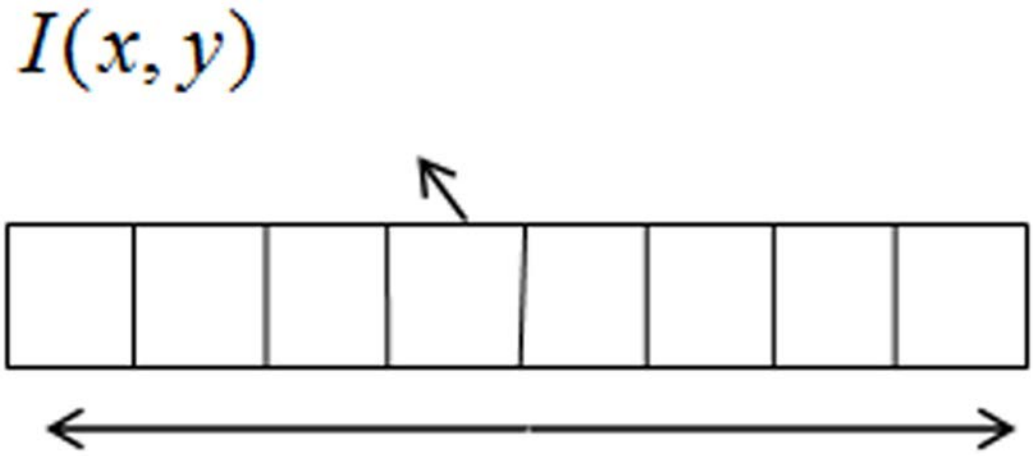
$I(x,y)$

图3

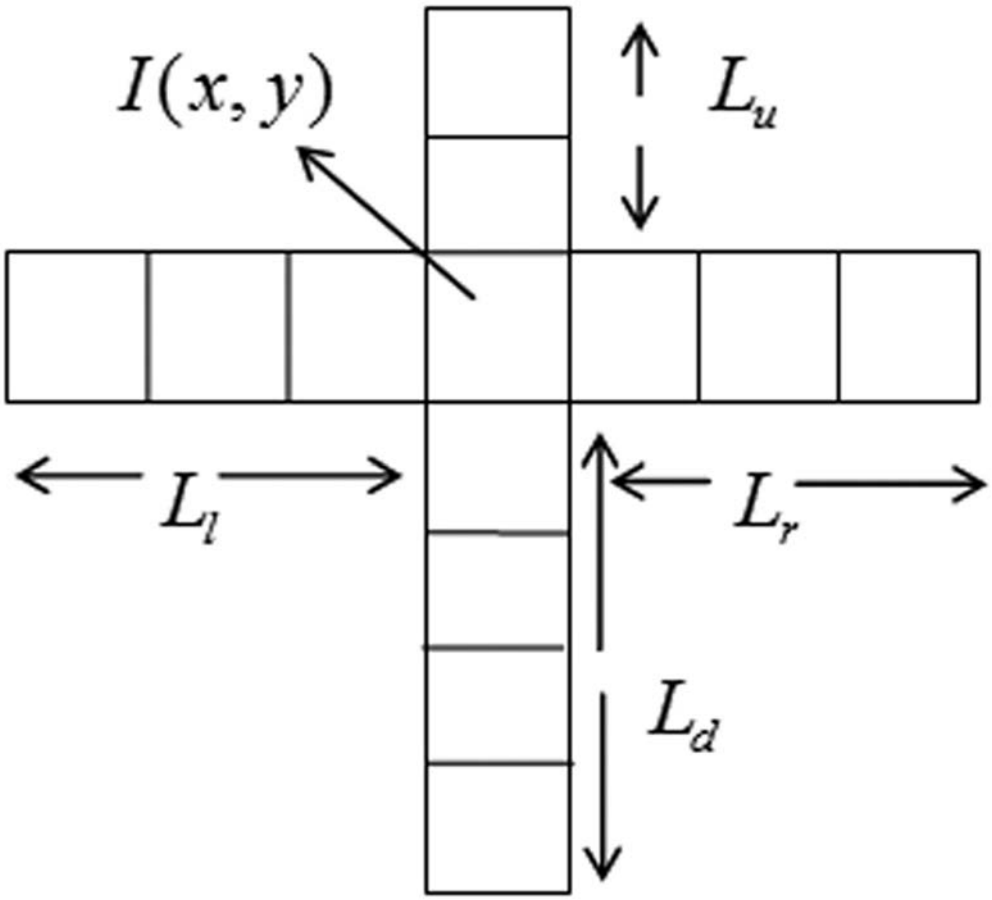

图4

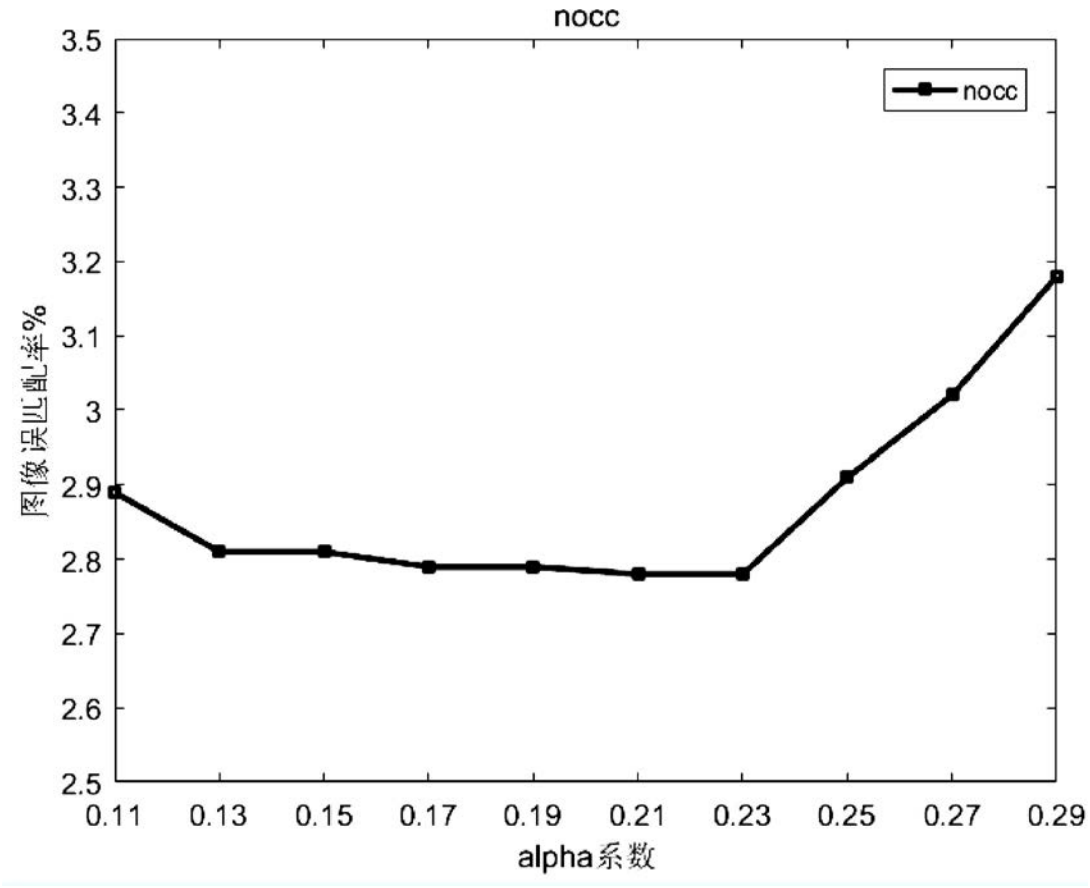
nocc
nocc
图像误匹配率%
alpha系数
3.5
3.4
3.3
3.2
3.1
3
2.9
2.8
2.7
2.6
2.5
0.11
0.13
0.15
0.17
0.19
0.21
0.23
0.25
0.27
0.29

图5

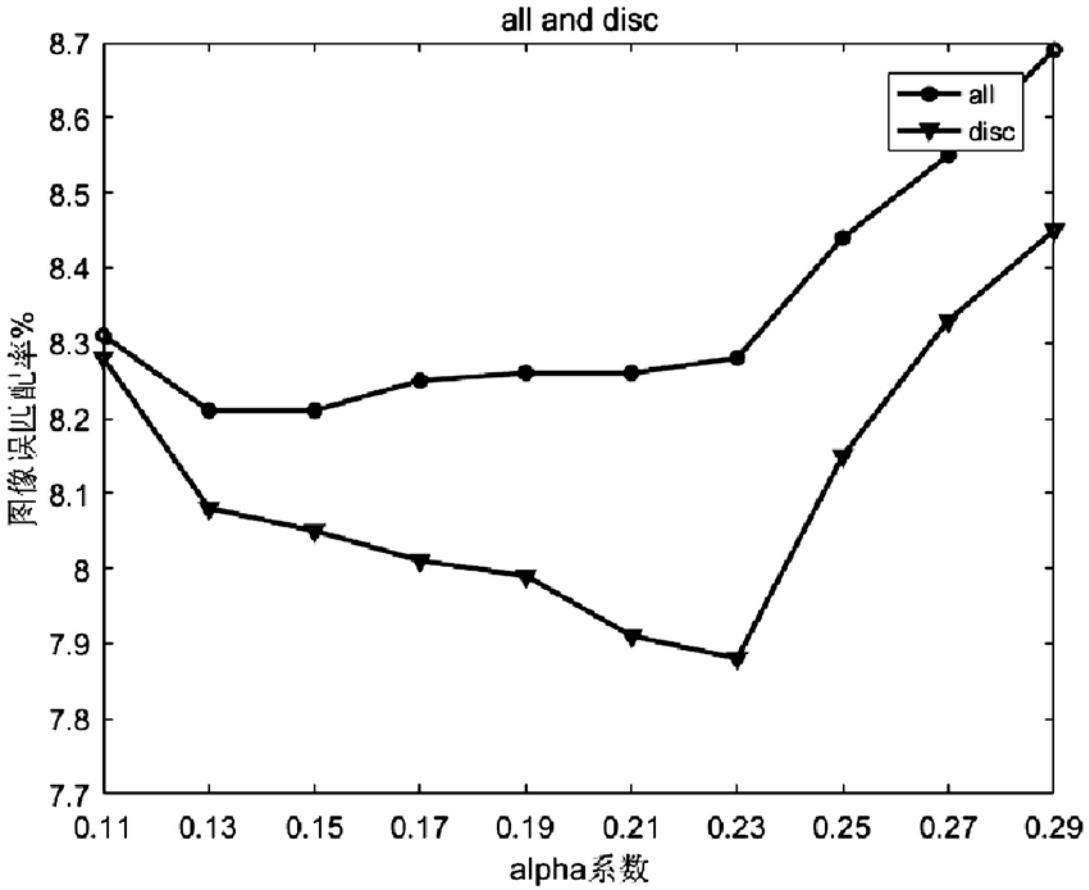
all and disc
all
disc
图像误匹配率%
alpha系数
8.7
8.6
8.5
8.4
8.3
8.2
8.1
8
7.9
7.8
7.7
0.11
0.13
0.15
0.17
0.19
0.21
0.23
0.25
0.27
0.29

图6