

一种动态跟踪尺度补偿的ORB特征点提取算法

王传传, 周雪云, 吴其滨

(广州工商学院工学院, 广东 广州 510800)
✉997459445@qq.com; zhouxu@gzgs.edu.cn; 570760876@qq.com



摘要: 在全景图像拼接背景下, 深入研究图像特征点的提取算法发现, ORB算法比SIFT、SURF算法的执行效率更高, 但ORB算法不具备SIFT、SURF算法的尺度不变性, 为了确保图像匹配的准确性, 文章提出了一种动态跟踪尺度补偿的策略, 对ORB算法进行优化, 实现成像效果和计算效率的改善, 并分析了该思路的可行性。通过OpenCV视觉库进行实验, 得出提取的特征点数量约是SIFT算法的2.44倍、SURF算法的2.24倍、原始ORB算法的1.47倍, 验证了动态跟踪尺度补偿的ORB算法的可行性。

关键词: 动态跟踪尺度补偿; ORB算法; 特征提取

中图分类号: TP311 **文献标识码:** A

An ORB Feature Point Extraction Algorithm based on Dynamic Tracking Scale Compensation

WANG Chuanchuan, ZHOU Xueyun, WU Qibin

(School of Engineering, College of Technology and Business, Guangzhou 510800, China)
✉997459445@qq.com; zhouxu@gzgs.edu.cn; 570760876@qq.com

Abstract: Based on panoramic image mosaic, in-depth study of image feature point extraction algorithm shows that ORB (Oriented FAST and Rotated BRIEF) algorithm is more efficient than SIFT and SURF algorithms, but ORB algorithm does not have the scale invariance of SIFT (Scale-invariant Feature Transform) and SURF (Speeded-Up Robust Features). In order to ensure the accuracy of image matching, this paper proposes a dynamic tracking scale compensation strategy to optimize ORB algorithm and improve imaging effect and computing efficiency. The feasibility of this idea is also analyzed. The results of the experiment with OpenCV visual library show that the number of extracted feature points is about 2.44 times of SIFT algorithm, 2.24 times of SURF algorithm, and 1.47 times of original ORB algorithm, which verifies the feasibility of ORB algorithm for dynamic tracking scale compensation.

Keywords: dynamic tracking scale compensation; ORB algorithm; feature extraction

1 引言(Introduction)

SIFT算法和SURF算法无法满足对实时性要求较高的图像处理任务, 针对该问题, RUBLEE等^[1]提出了ORB(Oriented FAST and Rotated BRIEF)算法, 极大地提高了特征点的提取速度。ORB算法是在FAST角点提取算法^[2]和BRIEF描述算子的基础上改进得来的, 具有旋转不变性, 且能抗噪声。ORB算法在速度方面比SIFT算法快两个数量级, 比SURF算法快一个数量级, 在有些图像数据上的表现更好。

ORB算法的缺点是没有很好地解决尺度不变的问题。换言之, 用ORB算法对两幅拥有相同目标物且有尺度差别的图像提取特征点后, 匹配点对的数量和质量会因尺度变化而下降。为了解决这一问题, 利用带图像金字塔的LK光流算法对跟踪尺度进行补偿, 进而优化ORB算法, 实现动态跟踪目标物的目的。为了验证该方案的有效性, 本文利用OpenCV视觉库进行实验, 在算法的运行时间和特征点提取数量方面与原始ORB等算法进行性能对比评估。结果表明, 优化后的算法

能以较小的时间代价,提高图像特征点的采集数量。

2 ORB特征点检测与描述算法(The algorithm of ORB feature point detection and description)

2.1 O-FAST角点检测算法

FAST(Features from Accelerated Segment Test)特征检测^[2]算法认为如果图像中某一像素点与其邻域内多数像素点的属性不同,则该像素点可能为一个特征角点^[3]。为了检测某一像素点是否是FAST特征角点,利用布雷森汉姆(Bresenham)直线算法画一个以该像素点为圆心的半径为 N 个像素单位的圆,FAST算子检测如图1所示。

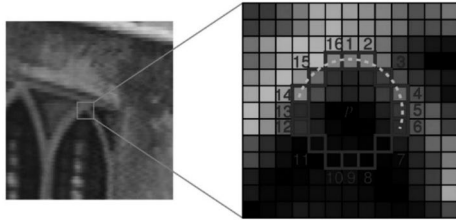


图1 FAST算子检测

Fig.1 FAST operator detection

为了便于辨识,FAST算法根据 N 值不同,标记为FAST- N 算法。当采用FAST-3时,Bresenham圆共由16个像素点组成。令点 p 的灰度值为 I_p ,检测阈值为 t 。若Bresenham圆上的16个像素点中有连续 n 个点都大于 $(I_p + t)$ 或都小于 $(I_p - t)$,则把该点作为FAST特征角点。

上述方法的缺点是必须循环遍历16个像素点才能做出判断,本文采取了一种加速特征点的获取方法:在16个点中任意选取一个点作为起始点,沿顺时针方向,每隔 90° 选取一点,如果这四个点中有三个都大于 $(I_p + t)$ 或都小于 $(I_p - t)$,就把此中心点作为FAST特征候选点;反之,此中心点不是FAST特征角点。

经试验证明,当取9时的效果最好,本文采用了FAST-9特征方法,如图2所示。



(a)原图

(b)角点效果图

图2 FAST特征角点

Fig.2 FAST feature corner

由于FAST算法不能保持特征点的旋转不变性,所以O-FAST算法在FAST算法的基础上,用强度质心^[1]方法添加方向属性,其具体做法如下。

先假设图像中某一区域的质心和几何中心为不同点,利

用质心和几何中心所构成的向量刻画方向,则该区域的矩表示如下:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (1)$$

质心 C 表示如下:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2)$$

在O-FAST中,本文选取的区域是以FAST特征角点 O 为圆心, r 为半径,即 $x,y \in [-r,r]$ 。特征角点 O 是该区域的几何中心,与质心 C 可以构成向量 $\overrightarrow{OC}$,将 $\overrightarrow{OC}$ 的方向 θ 作为FAST特征角点的方向如下:

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (3)$$

其中,atan2定义如下:

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \pi + \arctan\left(\frac{y}{x}\right) & y \geq 0, x < 0 \\ \pi + \arctan\left(\frac{y}{x}\right) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases} \quad (4)$$

2.2 R-BRIEF特征点描述算法

针对基于梯度直图的特征描述符,SIFT和SURF算法无法满足对实时性要求很高的系统,Michael Calonder等人提出了一种基于BRIEF算子^[4]的特征描述算法,其实现步骤如下。

Step1:进行定义 τ 测试。

在高斯平滑处理后的图像上选取一块大小为 $S \times S$ 的局部区域 p ,在区域 p 上进行 τ 测试:

$$\tau(p; x, y) = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

其中, $p(x)$ 为点 $x = (u, v)$ 的灰度值。

选取若干个符合特定条件的点对 $[x, y]$ 进行 τ 测试,组成由多组测试结果形成的二进制串,并把它作为一个BRIEF特征:

$$f_n(p) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (6)$$

选取BRIEF-64^[4]进行介绍,即BRIEF特征串的存储空间占64个字节,共需 $64 \times 8 = 512$ bits。

Step2:进行点对 $[x, y]$ 的选择。

由公式(5)可知,构建一个512 bits的特征串需要512个有序点对。

由于BRIEF算法不具备旋转不变性,所以需要将BRIEF改造成具有旋转感知性(Rotation-Aware)的R-BRIEF算法。ORB算法采用BRIEF-32,即描述符的串长度为256 bits。本文选取了 $n = 256$ 组测试点对,组成一个 $2 \times n$ 的矩阵 S :

$$S = \begin{pmatrix} x_1 & \cdots & x_i & \cdots & x_n \\ y_1 & \cdots & y_i & \cdots & y_n \end{pmatrix} \quad (7)$$

用公式(3)得到的方向角 θ ，计算对应的旋转矩阵 R_θ ：

$$R_\theta = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad (8)$$

构造矩阵 S 的受控(steered)矩阵 S_θ ：

$$S_\theta = R_\theta S \quad (9)$$

所以，受控的BRIEF算子(steered BRIEF)可定义如下：

$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta \quad (10)$$

为了加速计算过程，预先构建一张查找表，将角度离散化，步进单位设为 $2\pi/30$ ，只要离散化的O-FAST特征角点方向 θ 与上述角度标尺相符，则可以通过查找表查找对应的点集 S_θ ，并计算描述符。

BRIEF描述符采用点对的方式进行测试，这种方法虽然简单易行，但是无法较好地抑制噪声。为了解决这一问题，ORB算法采用点集对的方法。如图3所示，以特征角点为中心选取一个边长为 $w_p = 31$ 的正方形邻域，在该邻域中用边长为 $w_t = 5$ 的正方形子窗口(点集)代替BRIEF描述符中的点，用点集对代替BRIEF描述符中的点对。这样该邻域中应有 $N = (w_p - w_t)^2 = 676$ 个子窗口。从中任取两个构成点集对，那么点集对的数量为 $M' = C_N^2 = 228,150$ ，去掉窗口相互覆盖的点集对，真正可用的点集对的数量为 $M = 205,590$ 。

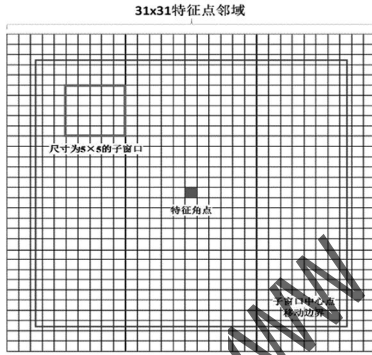


图3 R-BRIEF测试集

Fig.3 R-BRIEF test sets

未添加旋转特性的BRIEF具有的一个优点是二进制特征描述符有较大的方差且均值在0.5附近，可以很好地对特征点进行区分；而添加旋转特性的steered BRIEF却丧失了这一特性。本文采用高斯BRIEF点对100 k个样本特征点进行描述，得到均向量的位响应均值分布。同时，ORB算法通过如下操作提高steered BRIEF的辨识度。

(1)按图3的方式选取点集对，对点集对进行 τ 测试。

(2)将测试结果按照与0.5做差，得到的值按从大到小排序，得到队列 T 。

(3)应用贪婪算法：①将队列 T 的队首移入结果集 R 中；②队列 T 的新队首与结果集 R 中的所有结果进行比较，若它们的相关性大于预设阈值 t_{co} ，则放弃队首，减少结果集 R 的冗余信息，否则添加到 R 中；③当结果集 R 有256个元素时，将 R 作为

描述符，若 R 中的元素个数不及256个，则提高预设阈值 t_{co} ，然后重复前两个步骤。

3 ORB算法的改进(Improved ORB algorithm)

3.1 光流动态跟踪法

运动物体在视频中表现为相关目标特征像素点在连续帧的移动，而由于人眼的视觉暂留现象，因此像素点的移动可以用“光流”刻画，用光流的矢量速度对图像中物体的运动信息进行描述。带图像金字塔的LK(Lucas-Kanade)光流算法^[5]可以解决多尺度下“大运动”的跟踪问题^[6]。原始的LK算法(假设条件如图4所示)的实现必须具备下面三个假定条件^[7]。

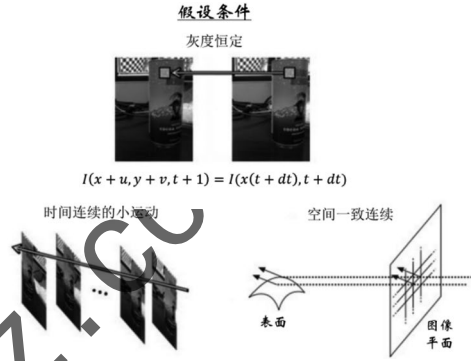


图4 LK算法的条件

Fig.4 Condition of the LK algorithm

条件1：灰度保持不变。

视频中运动的目标物在相邻帧间的灰度保持不变，则

$$f(x, t) \equiv I(x(t), t) = I(x(t + dt), t + dt) \quad (11)$$

也就是说，运动目标物的灰度不随时间的变化而发生改变，那么：

$$\frac{\partial f(x)}{\partial t} = 0 \quad (12)$$

条件2：“小运动”。

运动目标物体在连续帧间的位移较小，即属于“小运动”。由公式(11)、公式(12)和链式法则，可得到：

$$\frac{\partial I}{\partial x} \Big|_t \left(\frac{\partial x}{\partial t} \right) + \frac{\partial I}{\partial t} \Big|_{x(t)} = 0 \quad (13)$$

其中， $I_x = \frac{\partial I}{\partial x} \Big|_t$ ， $I_t = \frac{\partial I}{\partial t} \Big|_{x(t)}$ ，而 $v = \frac{\partial x}{\partial t}$ 就是所要求的光流速度，即

$$v = -\frac{I_t}{I_x} \quad (14)$$

实际场景中，不能完全保证满足前两个假定条件，导致计算的光流速度 v 会有微小的偏差。因此，只要按照上述方法求得的光流速度足够接近真实的速度，就可以利用牛顿法迭代得到相对准确的收敛解。如图5所示，将首次获得的有偏差的光流速度作为初始值并进行下一次迭代，之后重复该过程。依据第一假定条件可知，空域导数 I_x 始终保持不变，而时域导数 I_t 每次迭代前都要重新计算，正常经过5次迭代收敛即可获得满意的光流速 v 。

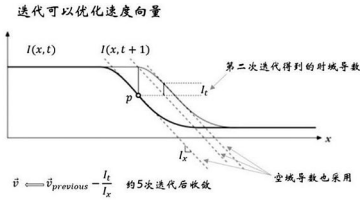


图5 牛顿法求光流速度的收敛解

Fig.5 Convergence solution of optical flow velocity by Newton method

当将一维的光流场扩大到二维空间时, 添加y方向的分量, 光流速度在x、y上的分量分别用 u 、 v 表示, 则

$$I_x u + I_y v + I_t = 0 \quad (15)$$

如图6所示, 由于已知条件少, 导致无法求出二维空间下的光流速度, 只能获取垂直运动分量, 这也被称作孔径问题^[7]。通过小孔, 观察不到物体的下移运动, 只能观察到物体的右向运动, 如图7所示。

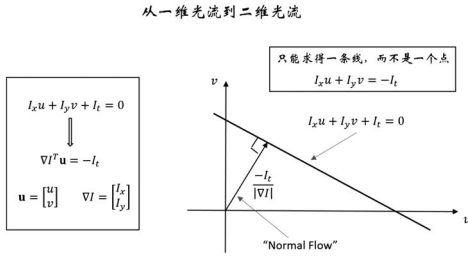


图6 二维光流速度的求解

Fig.6 Solution of two dimensional optical flow velocity

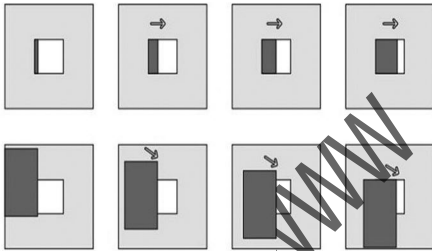


图7 孔径问题

Fig.7 The aperture problem

为了解决孔径问题, 就需要用到第三个假设条件。

条件3: 运动区域的空间连续性。

在运动场中, 由于目标物在图像中的投影是区域性的, 表现为连续帧之间位移的像素点, 其邻域的像素点也具有类似的位移趋势。本文通过建立中心点邻域像素的方程组增加约束条件, 从而避免了孔径问题的出现。如果采用 5×5 的邻域描述中心点的运动状态, 则

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix}_{25 \times 2} \begin{bmatrix} u \\ v \end{bmatrix}_{2 \times 1} = \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}_{25 \times 1} \quad (16)$$

当两条或两条以上的边缘出现在 5×5 的窗口邻域中, 可

利用最小二乘法求解该系统方程。通过公式(17)求解得残差平方和 $\|Ad - b\|^2$ 的最小值:

$$\underbrace{(A^T A)}_{2 \times 2} \underbrace{d}_{2 \times 1} = \underbrace{A^T b}_{2 \times 1} \quad (17)$$

则光流速度在x、y上的分量表示为 u 、 v :

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{A^T d} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b} \quad (18)$$

当 $(A^T A)$ 秩为2时可逆, 方程可解:

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b \quad (19)$$

实际场景中, 视频获取到的动态目标相对于图像尺度来说位移较大, 导致LK算法处理跟踪问题所得的结果也不理想。当用大尺度窗口跟踪“大运动”时, 又会破坏第三条运动连贯性的假设条件。本文引入图像金字塔模型, 采用由粗至精的方式(Coarse-to-Fine)对光流进行估计, 金字塔顶层的一个像素点可以表征其最底层的若干个像素点, 从金字塔最顶层对光流进行估计, 并将估算的结果作为下一层估计的初始值。这不仅避免了“小运动”的约束限制, 而且随着计算的向下进行, 计算精度也会提高^[8]。算法演示过程如图8所示。

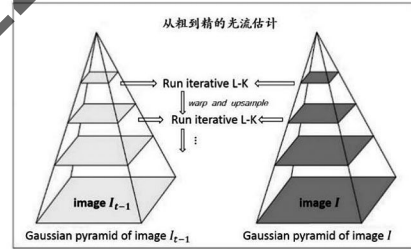


图8 带图像金字塔的LK光流算法

Fig.8 LK optical flow algorithm with image pyramid

3.2 基于动态跟踪的尺度补偿策略

由于ORB算法中的O-FAST角点检测和R-BRIEF特征描述都没有解决尺度问题, 也就是说ORB算法不具备尺度不变性, 在全景拼接的过程中, 场景的尺度通常不会发生剧烈的变化, 为了避免视图场景中(比如在高速公路上行驶的汽车)突发的极端情况, 对尺度不变性的保证还是必要的, 而且这么做可以很好地提高拼接的准确度和精确性。

本文利用带图像金字塔的LK光流算法对跟踪尺度进行补偿^[9-10], 并进行实验。视频中产生的尺度变化是由目标物与背景产生消逝点方向的相对运动导致, 如图9所示。静止场景往往不会发生尺度变化, 当目标物的尺度发生变化时, ORB算法所检测到的特征点数可能会下降, 当添加运动跟踪算法后, 目标物可以最大限度地保留原始的特征点, 这样就保证了ORB算法检测的特征点不因尺度变化而丢失。如图10(a)中, 汽车的部分特征信息因尺度变化而丢失, 图10(b)中, 带图像金字塔的LK算法对ORB进行补偿, 保留了上一帧中汽车

的部分特征信息。



图9 消逝点方向相对运动产生尺度上的变化

Fig.9 Scale changes produced by the relative movement of the vanishing point



(a)汽车部分特征信息因尺度变化而丢失

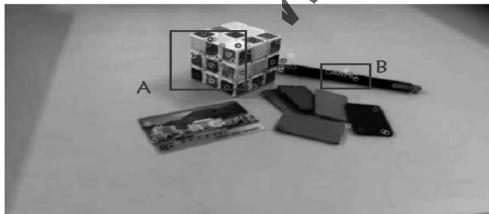


(b)算法补偿保留汽车的部分特征信息

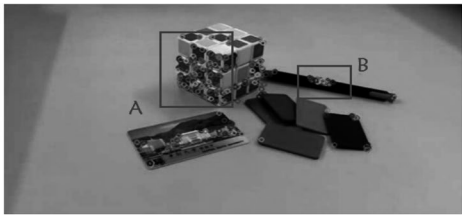
图10 光流法对ORB算法的尺度补偿

Fig.10 Scale compensation of ORB algorithm by optical flow method

本文通过带图像金字塔的LK光流算法对视频中的物品进行动态跟踪实验。在视频的同一帧，分别利用原始ORB算法和经尺度补偿后的ORB算法提取特征点，对比采集结果，如图11所示。



(a)原始ORB算法提取特征点



(b)尺度补偿后的ORB算法提取特征点

图11 特征点提取对比实验

Fig.11 Comparative experiment of feature point extraction

从图11(a)和(b)可以看出，图片尺度上有明显的差异，从A、B区域提取的特征点密集程度可以看出，经尺度补偿后的ORB算法提取的特征点数量要明显多于原始ORB算法。

此外，通过摄像头采集视频数据，选择采集视频前30帧的数据，运用OpenCV视觉库进行实验，对尺度补偿后的ORB算法的运行时间和特征提取数量进行对比统计，结果详见表1。

表1 特征提取算法性能对比

Tab.1 The performance comparison of feature extraction algorithms

序号	SIFT算法		SURF算法		ORB算法		尺度补偿后的ORB算法	
	特征点 数/个	运行时 间/ms	特征点 数/个	运行时 间/ms	特征点 数/个	运行时 间/ms	特征点 数/个	运行时 间/ms
1	213	2,215	225	147	314	68	314	157
2	211	2,347	225	130	314	36	422	138
3	216	2,321	225	152	314	35	420	114
4	218	2,285	226	124	315	34	435	133
5	218	2,267	227	134	314	35	427	133
6	220	2,246	229	127	312	49	461	144
7	221	2,269	227	148	313	36	465	132
8	219	2,313	237	121	315	35	476	131
9	228	2,306	230	150	316	34	462	127
10	224	2,308	228	123	314	34	434	122
11	200	2,076	212	121	314	62	433	149
12	201	2,041	214	160	310	34	492	118
13	188	1,947	200	133	302	61	501	145
14	179	2,138	199	131	301	74	421	158
15	183	2,250	189	117	299	60	464	144
16	182	2,001	190	126	298	33	459	113
17	152	2,182	173	120	289	33	423	113
18	157	2,224	184	120	288	33	407	109
19	149	2,369	176	126	286	45	492	121
20	154	2,173	184	145	287	32	445	114
21	153	2,110	175	138	286	51	440	135
22	151	2,049	174	115	286	59	446	143
23	154	2,358	179	116	287	33	436	113
24	157	2,142	180	117	287	32	461	117
25	154	2,233	171	118	285	31	447	115
26	158	2,271	170	115	286	31	424	115
27	146	2,284	162	149	284	31	430	114
28	142	2,361	166	148	284	32	433	111
29	134	1,995	167	114	283	38	397	117
30	130	1,934	151	115	282	31	436	109
均值	180.4	2,200.5	196.5	130	298.83	41.07	440.1	126.8

由表1可知，原始的ORB算法比SIFT算法快两个数量级，比SURF算法快一个数量级。经动态跟踪尺度补偿后的ORB算法和SURF算法的时间损耗基本相当，但补偿后的ORB算法提取的特征点数量约是SIFT算法的2.44倍、SURF算法的2.24倍、原始ORB算法的1.47倍。结果表明，经动态跟踪尺度补偿后的ORB算法性能更优。

4 结论(Conclusion)

本文介绍了ORB算法的特征提取和特征描述算法,指出ORB算法不具有尺度不变性。在全景拼接中,为了确保图像匹配的准确性,需要对原始的ORB算法进行改进。由于带图像金字塔的LK光流算法,可以解决多尺度下“大运动”的跟踪问题,即该算法对“大运动”、多尺度下的动态跟踪有着良好的效果。基于此,本文运用动态跟踪的方法对ORB算法进行尺度补偿,分析了该设计思路的可行性。为了验证动态尺度补偿后的ORB算法性能,在Windows系统下,通过摄像头采集视频数据,利用OpenCV视觉库进行实验,在算法的运行时间和特征点提取数量上与原始ORB等算法进行对比,结果表明:经过动态尺度补偿后的算法能以较短的时间代价,提高特征点的采集数量,实验结果符合预期,为提高图像特征点提取性能提供了一种理论支持。

参考文献(References)

- [1] RUBLEE E, RABAU D V, KONOLIGE K, et al. ORB: An efficient alternative to SIFT or SURF[C]// BARCELONA. 2011 IEEE International Conference on Computer Vision, ICCV 2011, Spain: IEEE, 2011:2564-2571.
- [2] BHAT A. Makeup Invariant Face Recognition using Features from Accelerated Segment Test and Eigen Vectors[J]. International Journal of Image and Graphics, 2017, 17(1):1750005.
- [3] 李慧, 蔺启忠, 刘庆杰. 基于FAST和SURF的遥感图像自动配准方法[J]. 国土资源遥感, 2012(2):28-33.
- [4] MOHANNA S, MORRIS T. Binary robust independent elementary feature features for texture segmentation[J].

(上接第41页)

改进, 然后使用聚类算法对锚框尺寸进行优化, 得到更为准确的锚框大小, 从而得到了YOLOv5_BM口罩人脸检测器。实验结果表明, 本文提出的方法相比其他的经典目标检测算法在准确性方面具有优势, 可为机场、地铁等存在大量混淆小目标的密集人群场所的口罩人脸检测提供模型支持, 具有良好的应用前景。

参考文献(References)

- [1] 谢晶仁. 国外传染病疫情防控对中国应对突发疫情的启示[J]. 中国公共安全(学术版), 2020(1):1-4.
- [2] 牛作东, 覃涛, 李捍东, 等. 改进RetinaFace的自然场景口罩人脸检测算法[J]. 计算机工程与应用, 2020, 56(12):1-7.
- [3] 邓黄潇. 基于迁移学习与RetinaFace的口罩人脸检测的方法[J]. 电子技术与软件工程, 2020(5):209-211.
- [4] ROMERA E, ÁLVAREZ J M, BERGASA L M, et al. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(1):263-72.
- [5] 冯小雨, 梅卫, 胡大帅. 基于改进Faster R-CNN的空中目标检测[J]. 光学学报, 2018, 38(06):250-258.
- [6] 王少韩, 刘森. 基于YOLOv5和树莓派的火灾识别研究[J]. 农业装备与车辆工程, 2022, 60(08):115-118.

Advanced Science Letters, 2017, 23(6):5178-5182.

- [5] 王艳红. 基于OpenCV的运动目标检测与跟踪算法的研究[D]. 杭州: 杭州电子科技大学, 2013.
- [6] 陈胜勇. 基于OpenCV的计算机视觉技术实现[M]. 北京: 科学出版社, 2008:338-340.
- [7] BAKER S, MATTHEWS I. Lucas-kanade 20 years on: A unifying framework[J]. International Journal of Computer Vision, 2004, 56(3):221-255.
- [8] ABDAT F, MAAOUI C, PRUSKI A. Real time facial feature points tracking with pyramidal Lucas-Kanade algorithm[C]// IEEE SYMP. Robot and Human Interactive Communication, Munich: IEEE, 2008:71-76.
- [9] PRIBULA V, CANOSA R L. Body-part estimation from Lucas-Kanade tracked Harris points[J]. Image Processing: Algorithms and Systems XI. SPIE, 2013, 8655:41-48.
- [10] MUKHINA M, YEREMEYEVA T, KUZMENKO A, et al. Analysis of efficiency of use of Harris and Kanade-Lucas-Tomasi detectors for visual navigation tasks[J]. Proceedings of the National Aviation University, 2015 (3):126-132.

作者简介:

王传传(1989-), 男, 硕士, 助教. 研究领域: 计算机视觉与图像处理, 大数据技术.

周雪云(1977-), 女, 硕士, 讲师. 研究领域: 计算机视觉, 大数据分析.

吴其滨(1993-), 男, 硕士, 助教. 研究领域: 图像处理, 软件测试.

[7] 王艺皓, 丁洪伟, 李波, 等. 复杂场景下基于改进YOLOv3的口罩人脸检测算法[J]. 计算机工程, 2020, 46(11):12-22.

[8] 管军霖, 智鑫. 基于YOLOv4卷积神经网络的口罩人脸检测方法[J]. 现代信息科技, 2020, 4(11):9-12.

[9] 王洋. 改进YOLOv5的口罩和安全帽佩戴人工智能检测识别算法[J]. 建筑与预算, 2020(11):67-69.

[10] 李鑫, 汪诚, 李彬, 等. 改进YOLOv5的钢材表面缺陷检测算法[J]. 空军工程大学学报(自然科学版), 2022, 23(02):26-33.

[11] 王飞, 王林, 张儒良, 等. 基于融合FPN和Faster R-CNN的行人检测算法[J]. 数据采集与处理, 2019, 34(03):530-537.

[12] 鲁博, 瞿绍军. 融合BiFPN和改进Yolov3-tiny网络的航拍图像车辆检测方法[J]. 小型微型计算机系统, 2021, 42(08):1694-1698.

作者简介:

于晓(1985-), 男, 博士, 副教授. 研究领域: 人工智能, 红外图像处理及其应用.

张茂松(1997-), 男, 硕士生. 研究领域: 目标检测, 深度学习.

周子杰(1993-), 男, 博士生. 研究领域: 目标识别, 红外图像处理.