

融合帧间差分与碰撞算法的快速跟踪预测算法^①

李凯强, 沈建新

(南京航空航天大学 机电学院, 南京 210016)

通讯作者: 沈建新, E-mail: cadatc@nuaa.edu.cn

摘要: 传统的运动目标跟踪预测算法难以保证机器人对高速运动目标的快速捕捉和提前预测, 尤其是运动目标在滑行过程中发生碰撞改变了原有的运动方向, 针对这一问题提出了基于帧间差分与碰撞算法相结合的运动目标跟踪预测算法. 通过帧间差分法快速识别出平面内运动物体的具体位置和运动速度, 根据其运动速度方向判别运动目标是否发生碰撞. 当运动目标在运动过程中发生碰撞, 采用 LS-DYNA 显示动力分析软件建立碰撞仿真模型, 并用 MATLAB 拟合仿真数据得到碰撞算法, 结合碰撞算法对运动目标的运动轨迹进行预测. 结果表明以帧间差分和碰撞算法相结合的运动目标检测跟踪算法对于在平面内运动目标的跟踪预测方面速度更快, 完全能够满足机器人对算法快速性的要求.

关键词: 运动目标检测; 帧间差分法; 碰撞算法; 轨迹预测

引用格式: 李凯强, 沈建新. 融合帧间差分与碰撞算法的快速跟踪预测算法. 计算机系统应用, 2018, 27(12): 136-142. <http://www.c-s-a.org.cn/1003-3254/6649.html>

Fast Tracking and Prediction Algorithm Based on Inter Frame Difference and Collision Algorithm

LI Kai-Qiang, SHEN Jian-Xin

(College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: The traditional moving target tracking and prediction algorithm is difficult to ensure that the robot can capture and predict the high-speed moving targets in advance. In particular, the collision of moving targets in the course of sliding changes the original motion direction. Aiming at this problem, a moving target tracking prediction algorithm based on frame difference and collision algorithm is proposed. Through the inter frame difference method, the specific position and velocity of the moving object in the plane are identified quickly, and the collision of the moving target is judged according to the direction of motion velocity. When the moving targets collide in the process of motion, the collision simulation model is established by using the LS-DYNA dynamic analysis software, and the collision algorithm is obtained by fitting the simulation data with MATLAB, and the motion trajectory of the moving target is predicted by the collision algorithm. The results show that the moving target detection and tracking algorithm, which combines the inter frame difference and the collision algorithm, is faster for the tracking and prediction of the moving target in the plane, and can fully meet the requirements of the robot's fast algorithm.

Key words: moving target detection; interframe difference method; collision algorithm; trajectory prediction

近年来, 桌上冰球机器人、羽毛球机器人、乒乓球机器人的相继出现, 体现了人机互动在人工智能领域的重要性, 已逐渐成为目前的热点研究方向. 机器人的工作原理大致相同: 机器人通过图像采集系统, 快速

^① 收稿时间: 2018-04-16; 修改时间: 2018-05-08; 采用时间: 2018-05-16; csa 在线出版时间: 2018-12-03

检测到运动目标的具体位置和运动状态,并实时的对运动目标实现击打.桌上冰球机器人是在二维平面内工作,对本文算法的验证较为简单,故而本文选取桌上冰球机器人作为算法的应用对象.冰球机器人通过图像采集系统获得冰球滑行过程中的位置信息并对其轨迹进行提前预测,等到冰球滑行到机器人的攻击范围内,通过伺服控制系统操纵机械臂,对冰球实现击打.由于冰球滑行的速度非常快,这就对运动目标预测算法提出了较高要求.由实际工程测试知道,算法运行时间在 30 ms 内时,冰球机器人才能实时挡住冰球,并有时间进行击打操作.对于运动目标的轨迹预测算法,目前应用最多的是卡尔曼滤波算法,使用卡尔曼滤波算法对观测到的轨迹信息进行滤波即可估计出运动目标的运动速度,用以预测轨迹,但当运动目标发生突变如碰撞后运动方向改变,可能会导致运动目标丢失.基于受力分析的运动建模方法目前应用也十分广泛,但其经常应用于无碰撞或碰撞发生后的轨迹预测情况^[1].考虑到算法运行速度的问题,本文通过帧间差分结合碰撞算法,在冰球与桌边碰撞前对冰球目标轨迹进行预测,提高了算法的运行速度,满足实时性的要求.

1 运动目标冰球检测算法

1.1 运动目标检测算法的种类及特点

常用的运动目标检测算法主要有光流法、帧间差分法和背景差分法三类.光流法是利用目标的矢量流特征^[2],如大小和方向等来检测运动区域,其主要优点在于能够检测出独立的运动对象,而不需预知场景的任何信息;缺点是大多数光流方法的计算比较复杂,且抗噪性能差,不能够快速检测到运动目标.背景差分法是目前目标检测中最常用的方法之一^[3,4],它将图像分为背景和前景,对背景进行建模,然后用当前帧与背景图像的差分来检测运动区域.该方法能够得到比较精确的运动目标信息,但对于动态场景的变化,如光照和外来无关事件的干扰特别敏感,因此在实际应用中需要加入背景图像的更新机制,降低了算法的运行速度.帧间差分法是在连续的图像序列中两个或三个相邻帧间,采用逐像素的差分并阈值化来提取出图像中的运动区域,它能较快地检测出视频图像中发生变化的部分^[5-8].由于连续两帧图像之间的时间间隔较短,差分图像受光线、气候等环境因素变化的影响较小,因此它具有很强的自适应性.但是如果目标速度过慢,则会造成过度覆盖,使检测的运动目标不够完整.

对运动目标冰球来说,其运动速度很快,满足了帧间差分法对运动速度的限制要求.又由于差分图像受光线、气候等环境因素的影响较小,所以当应用帧间差分法去检测冰球目标,能使运动目标检测算法具有更好的稳定性,并且帧间差分法能够快速地检测出冰球运动目标.

1.2 帧差分法对冰球目标的检测

本算法是在 VS2010+OpenCV 平台上编写运行的.首先从图像采集系统采集的序列图像中读取两张相邻帧图像,并将这两张相邻帧图像分别用 $I_{k+1}(x,y)$ 和 $I_k(x,y)$ 表示.然后,用灰度函数 `cvtColor()`; 将两张相邻帧图像转化成灰度图.其次,把得到的两张相邻帧灰度图进行差分运算如公式 (1), 得到差分图像 $D_K(x,y)$, 如图 1 所示.

$$D_k(x,y) = |I_{k+1}(x,y) - I_k(x,y)| \quad (1)$$

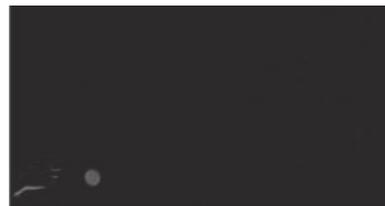


图1 帧间差分图像

为清晰明确的显示冰球目标,我们将所得到的差分图像进行二值化运算,其算法如公式 (2), 得到二值化的目标差分图如图 2.

$$R_k(x,y) = \begin{cases} 1, & D_k(x,y) > T \\ 0, & \text{其他} \end{cases} \quad (2)$$

其中, T 是二值化时所选取的阈值,其值一般由实际场景决定,本文中选取 $T=50$.



图2 差分图像二值化

为找到冰球的具体位置,我们要用 `findContours()` 函数去查找图像中所有轮廓.从图 2 中可以看到冰球位置和人手的干扰噪声,为准确的找到冰球位置,需要去掉噪声影响.首先,从图 2 中可以知道,冰球的面积

大于噪声的面积且人手的干扰噪声只位于图像最左侧, 所以在这里本文设置面积阈值 $area$ 结合位置阈值 $center.x$ 来除去噪声干扰, 面积阈值 $area$ 可以除去所有面积小于冰球目标的噪声, 位置阈值 $center.x$ 是所查找的轮廓中心在像素坐标系中横坐标的位置. 由程序测试知, 当面积阈值 $area$ 大于 100, 位置阈值 $center.x$ 大于 150 的时候, 只选中冰球目标. 对于冰球的具体位置我们找的是它的球心坐标, 而对于快速滑行中的冰球来说, 检测到的冰球目标会是一个椭圆轮廓, 其有大部分圆的特性, 且其最大外廓直径与原冰球目标直径相比基本保持不变, 所以这里我们用 OpenCV 自带的最小外接圆函数 $minEnclosingCircle(contours[i], center, radius)$; 去拟合图像中选中的椭圆轮廓, 拟合结果如图 3.

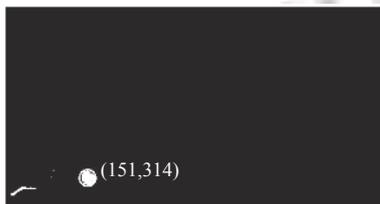


图 3 冰球轮廓最小外接圆

最小外接圆的圆心, 即为冰球在像素坐标位置的球心坐标位置, 将其圆心和半径存储起来, 如图 4 所示为所检测到的冰球的位置信息.



图 4 冰球在像素坐标中的位置

2 碰撞算法

冰球滑行过程中会与桌边产生碰撞, 从而改变其原有的运行轨迹, 增加了冰球轨迹预测的难度. 针对这一问题, 采用 ANSYS/LS-DYNA 显示非线性动力分析软件进行仿真模拟^[9,10], 建立冰球碰撞轨迹预测模型. 通过仿真模拟获得实验数据, 并用 MATLAB 对获得的数据进行数据拟合^[11], 从而得到了一种碰撞函数关系作为冰球碰撞过程的算法. 研究主要针对两种运动状态: 一种是以冰球的运动速度大小为变量, 保持碰撞前

冰球的运动速度方向与桌边的夹角不变; 另一种是保持冰球的运动速度大小不变, 只改变冰球的运动方向与桌边的夹角.

2.1 不同运动状态下仿真数据的收集

对于冰球与桌边碰撞角度不变的运动状态, 通过改变自变量速度的值, 来获得碰撞后冰球的运动状态, 其实验结果如表 1 所示.

表 1 角度不变, 改变速度大小 (单位: m/s)

碰撞前		碰撞后	
X 方向速度	Y 方向速度	X 方向速度	Y 方向速度
0.150	-0.150	0.146 081	0.126 359
0.200	-0.200	0.194 766	0.167 205
0.250	-0.250	0.243 465	0.209 038
0.300	-0.300	0.292 171	0.251 151
0.350	-0.350	0.340 849	0.292 787

对于速度大小不变的运动状态, 通过改变冰球与桌边的碰撞角度, 来获得碰撞后冰球的运动状态, 设定冰球速度恒为 0.4 m/s, 其实验结果如表 2 所示.

表 2 速度大小不变, 改变角度 (单位: m/s)

碰撞前		碰撞后	
X 方向速度	Y 方向速度	X 方向速度	Y 方向速度
0.168	-0.363	0.159 226	0.306 064
0.224	-0.332	0.215 666	0.277 472
0.284	-0.282	0.276 613	0.235 792
0.332	-0.224	0.326 132	0.187 415
0.363	-0.168	0.358 551	0.140 553

如表 2 中 X, Y 的数据值所示, 所采集的数据中当速度方向沿坐标轴正方向, 则速度为正, 否则速度为负. Y 方向的速度均为负值, 表示冰球在 Y 方向的速度沿 Y 轴负方向. 这里的 X、Y 坐标轴并不是像素坐标系的坐标轴, 而是建立的碰撞仿真模型中的坐标轴. 由于研究对象是速度矢量, 这里并不影响结果.

2.2 数据拟合

为了得到碰撞前后速度之间的关系, 分别对 X 轴方向和 Y 轴方向的速度分量进行分析. 将上述仿真实验中获得的碰撞前和碰撞后的速度数据进行拟合^[10], 从而得到一种简单的函数关系, 作为碰撞模型的碰撞预测算法. 通过观察相关系数 (R -square)、和方差 (SSE) 及均方差 ($RMSE$) 的值, 来判定该碰撞预测算法的可行性.

如图 5、6 所示, 分别为以 X 轴、Y 轴方向碰撞前

速度 V_x 、 V_y 为自变量,以 X 轴、Y 轴方向碰撞后的速度 $V_{x'}$ 、 $V_{y'}$ 为因变量,进行数据拟合的显示图像。

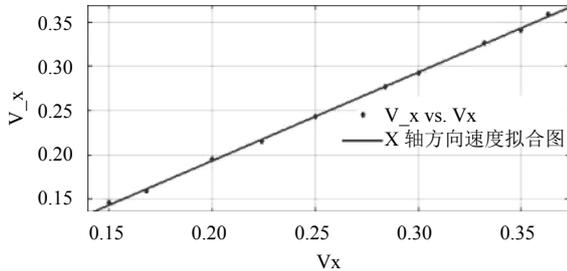


图5 X轴方向速度拟合

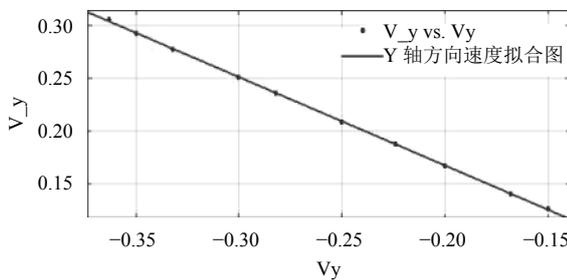


图6 Y轴方向速

如表3所示,为经过 MATLAB 拟合的结果,从而得到碰撞算法的函数关系式如下:

表3 X轴、Y轴方向速度拟合结果

Result	X轴方向	Y轴方向
Linear model poly	$f(x) = p1*x + p2$	$f(x) = p1*x + p2$
Coefficients	$p1 = 0.9974$	$p1 = -0.8392$
	$p2 = -0.00606$	$p2 = -0.000398$
SSE	$3.002e-05$	$5.445e-06$
R-square	0.9994	0.9998
RMSE	0.001937	0.000825

X轴方向速度函数关系式:

$$V_{x'} = 0.9974V_x - 0.00606 \quad (3)$$

Y轴方向速度函数关系式:

$$V_{y'} = (-0.8392)V_y - 0.000398 \quad (4)$$

上述(3)、(4)式中 V_x 和 V_y 分别表示碰撞前冰球在 X 轴方向和 Y 轴方向上的速度, $V_{x'}$ 和 $V_{y'}$ 分别表示碰撞后冰球在 X 轴方向和 Y 轴方向上的速度. 由于相关系数 (R-square) 的正常取值范围为[0, 1], 越接近 1, 表明模型函数对数据拟合得越好. 和方差 (SSE) 及均方差 (RMSE) 的值越接近 0, 说明模型函数拟合的越好, 数据预测也越成功. 在表3的拟合结果中, 上述两式的

相关系数 (R-square) 分别为: 0.9994 和 0.9998, 和方差 (SSE) 与均方差 (RMSE) 的值也近乎为 0, 所以本模型算法的建立能够很好的预测碰撞后冰球的运动状态。

3 轨迹预测算法及实验验证

本文运用帧间差分结合碰撞算法, 使冰球的运行轨迹只与冰球的初始运动状态有关, 算法在处理到前几帧便可以预测出冰球运行到机器人攻击点的最终位置, 极大地缩短了冰球轨迹的预测时间, 使机器人能够有充足的时间对冰球进行击打, 满足了机器人实时性的要求. 通过采集几组实际工程中连续帧图像来验证本文算法, 将预测位置与实际检测位置的差值 P 与冰球直径 D 作对比, 且由于机器人手柄直径大于冰球直径, 所以若 $P < D$, 则机器人能够挡住并实现对冰球的击打, 以此验证本文算法切实可行。

实验中所建立的坐标系为像素坐标系, 其整体尺寸为 694×370 , 数值均为像素坐标值^[12]. 由于本文主要解决算法运行速度慢的问题, 所以应该尽可能地在检测到冰球的前几帧时就能够预测出冰球到达机器人攻击范围内的目标位置(X_L, Y_L). 该目标位置是根据机器人的攻击范围合理的设定了一个 X 方向的阈值 X_L , 本文根据实际情况取 $X_L=500$, 对于预测目标只需要求出 Y_L 的值即可预测到目标位置. 本文用帧速表示冰球的速度, 通过帧间差分检测, 可以得到冰球在像素坐标中的具体位置 $I(x, y)$, 冰球在像素坐标中的半径 R , 检测出 $R=15$.

冰球在桌面上运行的轨迹主要分为两种: 一种是不与桌边发生碰撞的滑行; 另一种是滑行过程中与桌边发生碰撞, 改变了其原有的运动轨迹. 针对这两种情况, 本文用帧间差分和碰撞算法相结合的方法去进行轨迹预测实验, 其具体算法运行流程图如图7所示。

3.1 滑行中无碰撞的轨迹预测实验

这种情况对冰球轨迹的预测相对来说较为简单, 只需要知道冰球滑行的初始速度就能够预测出冰球滑行到机器人攻击范围内的具体坐标位置(X_L, Y_L), 并提前将这个坐标位置发送给机器人, 以实现对运动目标冰球的击挡。

3.1.1 滑行中无碰撞的轨迹预测算法

如图8所示, 为无碰撞时冰球轨迹预测示意图. 本文用帧差法第一次检测到冰球位置 $I_1(x_1, y_1)$, 第二次检测到冰球位置为 $I_2(x_2, y_2)$, 则冰球的初始速度为:

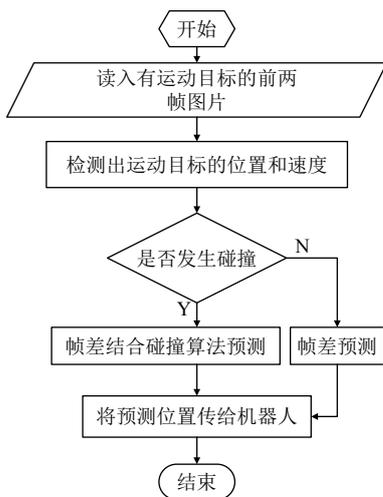


图7 算法运行流程图

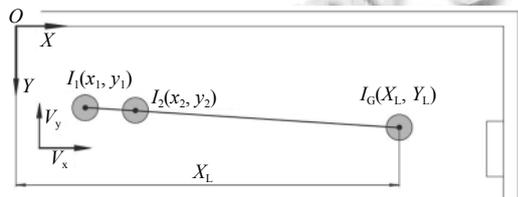


图8 无碰撞轨迹预测示意图

$$V_x = x_2 - x_1 \quad (5)$$

$$V_y = y_2 - y_1 \quad (6)$$

其中, V_x 为冰球在 X 方向上的帧速, V_y 为冰球在 Y 方向上的帧速.

对于需要预测的目标位置 (X_L, Y_L) 来说, 由于 $X_L=500$, 则冰球从第二次检测位置运行到该目标位置所需要的时间 T 为:

$$T = \frac{X_L - x_2}{V_x} \quad (7)$$

$$Y_L = y_2 + V_y \times T \quad (8)$$

由式 (8), 可以计算出冰球滑行到机器人攻击范围内的具体坐标位置 (X_L, Y_L) .

3.1.2 无碰撞的轨迹预测实验验证

如图 9, 10 为检测到冰球相邻两帧的位置, 根据算法判断此时冰球与桌边不发生碰撞, 其具体滑行轨迹如图 11 所示.

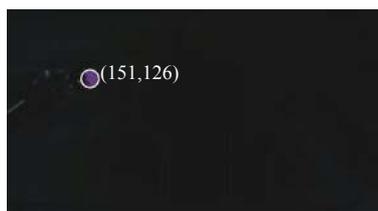


图9 检测到冰球位置 (一)

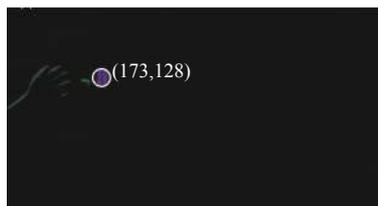


图10 检测到冰球位置 (二)

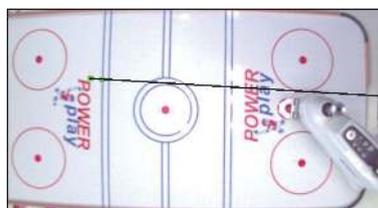


图11 无碰撞时冰球目标轨迹

将图 9, 10 中所获得的像素坐标值带入轨迹预测公式 (5) 至公式 (8), 得到预测目标位置为 $(500, 157.727)$, 对比图 12 中用帧差法检测的实际最近目标位置 $(495, 155)$ (因帧差法限制, 检测不到横坐标为 500 的位置, 故取其最邻近位置), 由于冰球直径为 $D=2R=30>2.727$, 所以冰球机器人能够挡住冰球, 该预测算法能够准确预测冰球轨迹.

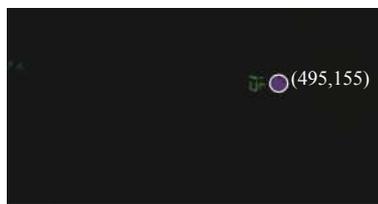


图12 检测到实际目标位置

3.2 滑行中与桌边发生碰撞的轨迹预测实验

当冰球与桌边发生碰撞时, 其情况更为复杂. 首先要根据检测到的冰球目标, 与桌边哪一侧碰撞, 并预测到碰撞点的位置. 其次根据碰撞点的位置结合碰撞算法, 预测出冰球滑行到机器人攻击范围内的具体坐标

位置 (X_L, Y_L) .

3.2.1 滑行中有碰撞的轨迹预测算法

如图 13 所示, 为冰球轨迹预测示意图, 与无碰撞时一样先用帧差法测到前几帧中冰球位置为 $I_1(x_1, y_1)$, $I_2(x_2, y_2)$, 求出冰球在碰撞前的速度为:

$$V_x = x_2 - x_1 \tag{9}$$

$$V_y = y_2 - y_1 \tag{10}$$

其中, V_x 为冰球在 X 方向上的帧速, V_y 为冰球在 Y 方向上的帧速, 带入式 (3)、(4) 可以求出碰撞后的速度 $V_{_x}$ 、 $V_{_y}$ 的值. 并且通过判断 V_y 的方向, 知晓撞向桌边的哪一侧.

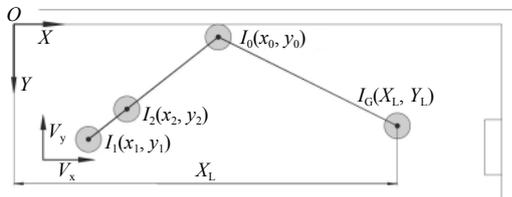


图 13 冰球轨迹预测示意图

(1) 冰球在 Y 方向上的帧速沿负方向

当冰球在 Y 方向上的帧速为负值时, V_y 沿负方向, 冰球从第二次检测位置到达碰撞点的时间 T_1 为:

$$T_1 = \frac{y_2 - R}{V_y} \tag{11}$$

到达碰撞点时, 冰球的坐标位置为 (x_0, y_0) , 其中 $y_0 = R$, x_0 为:

$$x_0 = x_2 + T_1 \times V_x \tag{12}$$

冰球从碰撞点到预测目标位置所用的时间为 T_2 , 其值为:

$$T_2 = \frac{X_L - x_0}{V_{_x}} \tag{13}$$

由此可以求出冰球到达机器人攻击范围内的目标位置 (X_L, Y_L) , 其中 $X_L = 500$, Y_L 的值则为:

$$Y_L = R + V_{_y} \times T_2 \tag{14}$$

(2) 冰球在 Y 方向上的帧速沿正方向

当冰球在 Y 方向上的帧速为正值时, V_y 沿正方向, 冰球从第二次检测位置到达碰撞点的时间 T'_1 为:

$$T'_1 = \frac{370 - R - y_2}{V_y} \tag{15}$$

到达碰撞点时, 冰球的坐标位置为 (x'_0, y'_0) , 其中

$y'_0 = 370 - R$, x_0 为:

$$x_0 = x_2 + T'_1 \times V_x \tag{16}$$

冰球从碰撞点到预测目标位置所用的时间为 T'_2 , 其值为:

$$T'_2 = \frac{X_L - x'_0}{V_{_x}} \tag{17}$$

同理可以求出冰球到达机器人攻击范围内的目标位置 (X_L, Y_L) , 其中 $X_L = 500$, Y_L 为:

$$Y_L = 370 - R - V_{_y} \times T'_2 \tag{18}$$

3.2.2 碰撞轨迹预测实验验证

如图 14, 15 为检测到的碰撞前冰球相邻两帧的位置, 根据算法判断此时冰球与桌边发生碰撞, 其具体滑行轨迹如图 16 所示.

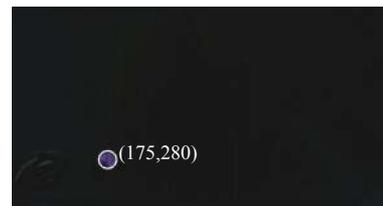


图 14 检测到冰球位置 (一)

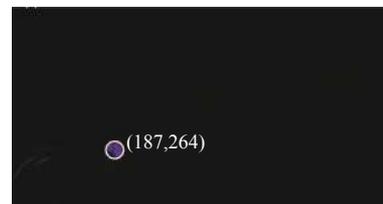


图 15 检测到冰球位置 (二)

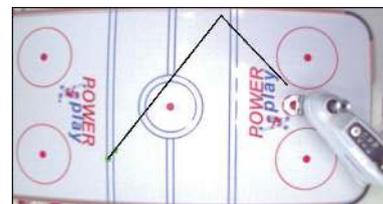


图 16 碰撞时冰球目标轨迹

将图 14, 15 中所获得的像素坐标值带入轨迹预测公式 (9)~公式 (14), 得到预测目标位置为 $(500, 156.693)$. 对比用帧差法检测到的实际目标位置, 如图 17, 因为冰球直径 $D = 2R = 30 > 6.693$, 所以机器人能够成功挡住冰球, 由此也验证了本文算法的可行性.

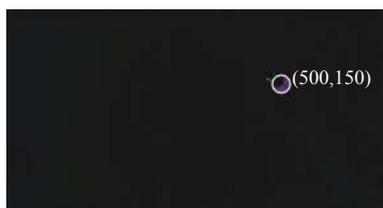


图 17 实际冰球位置

为验证本文算法的快速性, 本文用 *GetTickCount()* 函数来记录系统运行算法的时间, 该函数可以返回从算法启动到结束所经过的毫秒数, 精度非常高. 如图 18 所示, 本文算法运行时间为 $16\text{ ms} < 30\text{ ms}$, 完全能够满足机器人对目标跟踪预测算法快速性的要求.

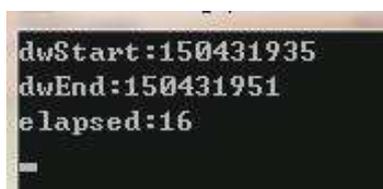


图 18 算法运行时间

4 结束语

本文通过帧间差分法快速找到冰球目标, 再结合碰撞算法, 对冰球轨迹进行预测. 算法仅通过两次帧间差分, 就能够预测出运动目标的位置信息, 满足了冰球机器人对算法快速性的要求, 使冰球机器人能够成功挡住冰球, 并且有充足的时间进行击打操作. 相对于传统的运动目标跟踪预测算法, 本文算法对于运动目标的跟踪预测速度更快, 实时性更好. 对于算法速度要求高的工程性问题, 本文算法具有很高的参考价值.

参考文献

1 章逸丰. 快速飞行物体的状态估计和轨迹预测[博士学位论文]. 杭州: 浙江大学, 2015.

- 2 石龙伟, 邓欣, 王进, 等. 基于光流法和卡尔曼滤波的多目标跟踪. 计算机应用, 2017, 37(S1): 131-136.
- 3 王彬, 马永杰, 李鹏飞. 结合分块的改进三帧差和背景差的运动目标检测. 计算机系统应用, 2015, 24(8): 155-159. [doi: 10.3969/j.issn.1003-3254.2015.08.027]
- 4 孙挺, 齐迎春, 耿国华. 基于帧间差分和背景差分的运动目标检测算法. 吉林大学学报(工学版), 2016, 46(4): 1325-1329.
- 5 胡平, 周文洪. 基于帧间差分和边缘差分的遗留物检测算法. 计算机系统应用, 2015, 24(3): 171-175. [doi: 10.3969/j.issn.1003-3254.2015.03.029]
- 6 Cheng YH, Wang J. A motion image detection method based on the inter-frame difference method. Applied Mechanics and Materials, 2014, 490-491: 1283-1286. [doi: 10.4028/www.scientific.net/AMM.490-491]
- 7 张娜, 魏海平, 于红绯. 一种融合特征点与运动补偿信息的运动目标检测新算法. 计算机应用与软件, 2017, 34(11): 210-216. [doi: 10.3969/j.issn.1000-386x.2017.11.039]
- 8 Asif M, Taj IA, Ziauddin SM, et al. An efficient inter prediction mode selection scheme for advanced video coding based on motion homogeneity and residual complexity. IEEE Transactions on Electrical and Electronic Engineering, 2016, 11(6): 760-767. [doi: 10.1002/tee.2016.11.issue-6]
- 9 于英华, 郎国军. 基于 LS-DYNA 的汽车保险杠碰撞仿真研究. 计算机仿真, 2017, 24(12): 235-238.
- 10 Liu CK, Song XP, Wang J. Simulation analysis of car front collision based on LS-DYNA and hyper works. Journal of Transportation Technologies, 2014, 4(4): 337-342. [doi: 10.4236/jtts.2014.44030]
- 11 Zhang WQ. Dynamic simulation and curve fitting of folding furniture based on geometric analysis and MATLAB. Proceedings of the 2nd International Conference on Materials Science, Machinery and Energy Engineering. Hong Kong, China. 2017, 123: 697-701.
- 12 章梦杰, 邵培南, 于铭华. 多维空间中基于模式的移动对象轨迹预测. 计算机系统应用, 2018, 27(1): 113-119. [doi: 10.15888/j.cnki.csa.006158]