基于单目视觉的冰壶定位和运动轨迹记录

摘　要

随着智能显卡的快速发展，深度学习神经网络近些年有了长足的进步，同时在体育行业的应用也更加广泛。本设计以当下 2022 年北京冬奥会中冰壶比赛项目为研究背景，将人工智能技术信息化，数字化带入传统的冰壶赛场，设计构造一套基于单目视觉的冰壶定位和运动轨迹记录的冰壶训练系统。该系统可以帮助教练组实时分析运动员在抛出冰壶后，冰壶运动的各项数据，以便制定更完善的训练计划。

本设计首先针对冰壶的自身结构以及场地特点进行分析，采集多张多角度冰壶照片，构建冰壶运动数据集，通过 YOLOv5 卷积神经网络进行训练识别。该数据集不仅可以为目前算法提供训练样本，更能为后续姿态检测、运动轨迹预测等方向提供样本支持。

能够实时准确锁定冰壶位置后，利用小孔成像模型及坐标系转换理论，通过相机标定和相机姿态估计，计算出相机内参和外参，建立图像坐标系到相机坐标系再到世界坐标系的对应关系，将相机拍摄的二维空间上的点映射到真实世界三维空间中，以此精确计算冰壶运动过程中的各项数据。对于冰壶自转的旋转角度，我们采用霍夫变换直线检测的方法，识别拍摄视频图片帧中冰壶把手上的黑色长条区域，通过一系列视觉图像处理，记录帧与帧之间直线夹角，实时获取冰壶旋转角度。我们还利用卡尔曼滤波方法对冰壶运动下一帧轨迹进行预测。

最后我们对该系统进行实地测验，表现出良好的功能和显示效果，可以实时精确定位冰壶在真实世界中的位置，并计算出当前状态下冰壶运动的距离，时间，平均速度以及旋转角度，同时给出下一帧冰壶预测位置，并绘制出冰壶的二维运动轨迹图。

关键词：冰壶运动；单目定位；目标检测；YOLOv5；霍夫变换；卡尔曼滤波

目 录

1. 问题分析

1.1 问题重述

为了助力 2022 年北京冬奥会，帮助冰壶运动员提高训练水平，运用计算机视觉技术对冰壶运动轨迹进行分析，由视频分析物体的如运动距离和运动时间等现实物理信息。记录由起始点 S 开始推动陆地冰壶至冰壶完全停止于停止点 E 的这段时间内的运动轨迹和运动时间，通过显示屏进行轨迹显示、时间显示和总旋转角度显示。

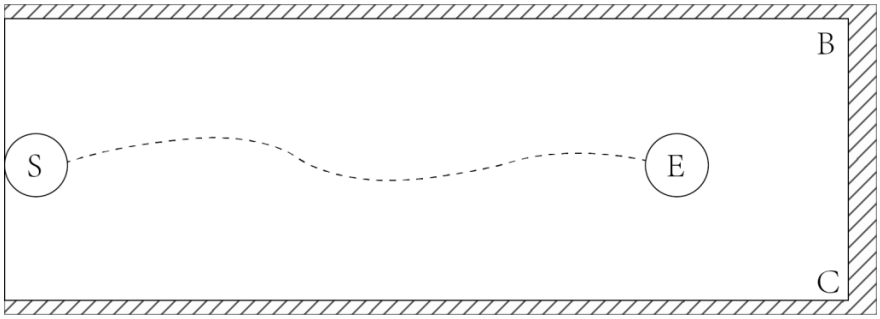

图 1 比赛场地

问题可以分为两类情况讨论，基本问题和综合问题：

基本任务：直线推动陆地冰壶，记录并显示的直线运动时间和运动距离。在基本问题中可以将冰壶抽象为质点，考虑以冰壶的中心点为目标识别冰壶的运动时间和运动距离。

综合任务：推动冰壶，冰壶运动过程中至少产生 720° 的旋转和 1m 的累计位移，记录并显示运动时间、运动距离、旋转角度和运动轨迹。在综合问题中，对运动时间、运动距离和运动轨迹分析时可以将冰壶抽象为质点，考虑以冰壶的中心点为目标进行识别，而对冰壶的总旋转角度需要考虑冰壶自转，需要对冰壶图像做进一步分析。

2. 模型假设

（1）将冰壶看作质点，冰壶的运动轨迹以冰壶的中心为基准进行记录，以便直观展现其运动轨迹。

（2）假定冰壶的运动轨迹的总旋转角度指冰壶自身总自转角度，通过计算冰壶自身自转角度作为冰壶旋转角度并记录。

（3）冰壶的直线运动指手握冰壶把手向前推出冰壶，冰壶做近似直线运动。

（4）冰壶的旋转运动指手握冰壶把手扭动手腕并向前推出冰壶，冰壶做近似曲线直线运动。

（5）仅对规定区域内的冰壶运动轨迹进行记录，若超出区域不进行记录。

3. 符号说明

符号	含义
f	焦距
(x,y)	图像坐标系
(μ,ν)	像素坐标系
(X_c,Y_c,Z_c)	相机坐标系
(X_w,Y_w,Z_w)	世界坐标系
c_x,c_y	平移距离
α,β	缩放倍数
f_x, f_y	缩放焦距
t_i	平移向量
K	内参矩阵
R	旋转矩阵
k,b	斜率，截距
θ	与极坐标轴夹角
r	极点到给定直线的垂直距离
$\hat{x}_{k-1}$， $\hat{x}_k$	后验状态估计值
$\hat{x}_{\bar{k}}$	先验状态估计值
P_k， P_{k-1}	后验状态协方差
$P_{\bar{k}}$	先验估计协方差
H	转换矩阵
z_k	滤波的输入
K_k	滤波增益矩阵
A	状态转移矩阵

Q	系统过程协方差
R	测量噪声协方差
B	系统状态转换矩阵

4. 方案分析

4.1 方案论证

在本文中需要对图像视频中冰壶的运动进行分析，可以分解为两个子任务：对冰壶的识别以及冰壶的运动分析。

4.1.1 冰壶运动分析

为了获取冰壶的精确位置信息，可以采用双目乃至多目定位法，但考虑到泛用性和可实现性，在条件有限的情况下，只需一部机位即可测定冰壶的运动轨迹，对其进行信息分析，故本文采用单目定位法。

对冰壶运动的分析，在视频数据中可以利用视频帧数进行时间片分解计算运动时间。而对运动距离可以分析目标在图像中位置再利用相关算法进行估计。根据摄像机成像原理和透视投影的性质，图像中共线四点的交比与真实世界的交比相等。因此，可以利用交比不变性来确定图像中的距离信息，分别可以采用直接几何法和间接几何法进行求解。

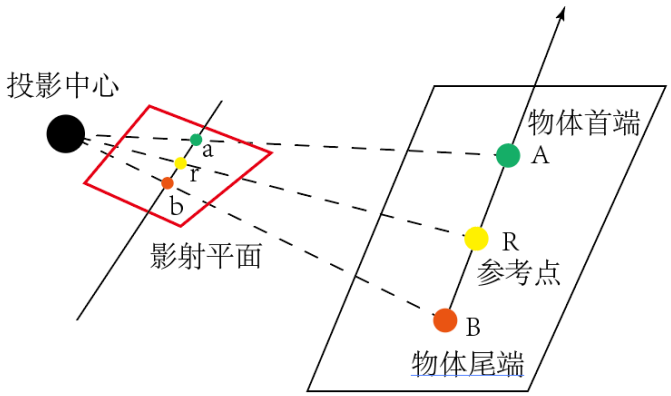

图 2 交比不变性示意图

直接几何法指通过求解出图像灭点和灭线，进而结合已知物体信息计算出距离信息。基于灭点检测算法，通过对三维空间平行线段在图像中的投影线段的检测，再利用平行线段求解灭点，灭点的准确性对后续地求解灭线，以及图像物体距离计算有重要的作用。间接几何法通过求解相机的内参矩阵和镜头的畸变系数。由于相机坐标系会随着相机的移动而改变坐标的原点和各个坐标轴的方向，需要将相机坐标转换为不变的世界坐标，从而在标准坐标系下获得距离信息。

通过阅读文献不难发现，直接几何法可以灵活获取图像信息，但对灭点检测算法要求较高且需要良好的拍摄角度。而间接几何法需要具体的坐标标定以及相机参数，但精度较

高。由于冰壶场地固定易于设置坐标标定，相机内参可由程序计算得出，故本文选择精度较高的间接几何法。

使用间接几何法就要求我们求解相机内参，镜头畸变系数和外参。

4.1.2 相机内参标定

传统的相机标定方法利用几何形状或图案等包含已知信息等特殊标定物，建立参考坐标系来求解，其中张正友相机内参标定法已有比较成熟的理论支持与实践案例，能够精准的标定相机焦距，主点信息以及畸变因子等参数。但必须要用实际相机获取一系列拍摄素材，才能进行分析和计算，由于一般情况下内参标定程序与相机分离，对源自网络等多媒体途径等相机就难以标定，限制了此方法等灵活性和自动性。

相机自标定法，指依赖于视频场景中的图像序列来估计相机参数，基于内参矩阵与二次曲面的约束关系进行构造和求解，由求解思路可以进一步划分为基于绝对二次曲面的相机自标定方法，基于模约束的相机自标定方法以及基于 kruppa 方程的相机自标定方法。虽然自标定法可以对任意视频数据进行相机内存标定，但这些自标定方法与传统方法相比约束条件更少，导致方程求解的不稳定性较高和优化方案的影响较大。

综上所述，传统的相机标定方法借助现实标定参照物计算求解，在精度上有较大优势；而相机自标定法则是利用算法分析求解视频中的，不需要拍摄源相机，有较高的灵活性。由于在本任务中拍摄视频的源相机是可确定的，故本文采用传统的相机表达法进行相机内参和镜头畸变的标定任务，以提高精确度。

4.1.3 相机外参标定

三线法指在图像中利用手工标定或自动检测的方法获取三条标定线，利用空间中直线在无限远处收敛的性质，相机外参与这三条直线的斜率间存在映射关系，可以构成转换表达方程，由此获取相机外参。三线法对相机拍摄角度要求较高需要能够在图像中显示出灭点位置，在特定场景下，如自动驾驶，无人机侦察等领域有着广泛的应用。

逆透视投影模型是利用像素坐标系映射的世界坐标系中的坐标转换的过程求解透视变换的逆过程。尽管透视过程可以将三维信息转换为二维信息，但单个摄像机获取的二维信息难以通过逆透视投影模型还原至三维信息，必定在高度上的损失精度，但在二维平面中精度较高。

为了获取较高的精度和拍摄自由度，本文采用逆透视投影模型，借助标定图像协助进行相机姿态估计。

4.1.4 目标检测算法的选择

目标检测通常是指在图像中检测出目标出现的位置及对应的类别，它是计算机视觉中的根本问题之一，如图像分割、物体追踪、关键点检测等都依赖物体检测。从应用来看，物体检测已广泛应用于大家的日常生活中，如浏览器的拍照识图、自动驾驶领域的行人车辆检测、道路目标检测（人行横道检测）及图像分类等。目前主要的目标检测手段可以分为传统目标检测算法和卷积神经网络算法。

传统的物体检测算法一般包含三个部分，检测窗口选择，特征设计和分类器设计。

检测窗口的选择指如何选定目标位置，以人脸检测为例，当给出一张图片时，我们需要框出人脸的位置以及人脸的大小，那么最简单的方法就是遍历搜索候选框，把图像中所有可能出现框的位置从上往下、从左往右遍历一次。并且通过缩放一组图片尺寸，得到图像金字塔来进行多尺度搜索。

图 3 滑动窗口检测与图像金字塔

可以看出这种遍历搜索的计算浪费很大，难以实际应用。但通过对目标视觉特征分析，可以获取一定的先验知识，如人脸肤色 YCbCr 空间呈现很紧凑的高斯分布，通过肤色检测快速选定候选区域作为人脸检测搜索范围，这使得检测整体速度提升很多。但肤色提取只是用到简单的颜色先验，如果遇到和肤色很像的，比如黄色的桌子，很有可能被误判成人脸的候选检测区域。进一步提高精度衍生出如 Selective Search 或 EdgeBox 等提取的方法，基于颜色聚类、边缘聚类的方法来快速把不是所需物体的区域给去除，相对于肤色提取精度更高，极大地减少了后续特征提取和分类计算的时间消耗。

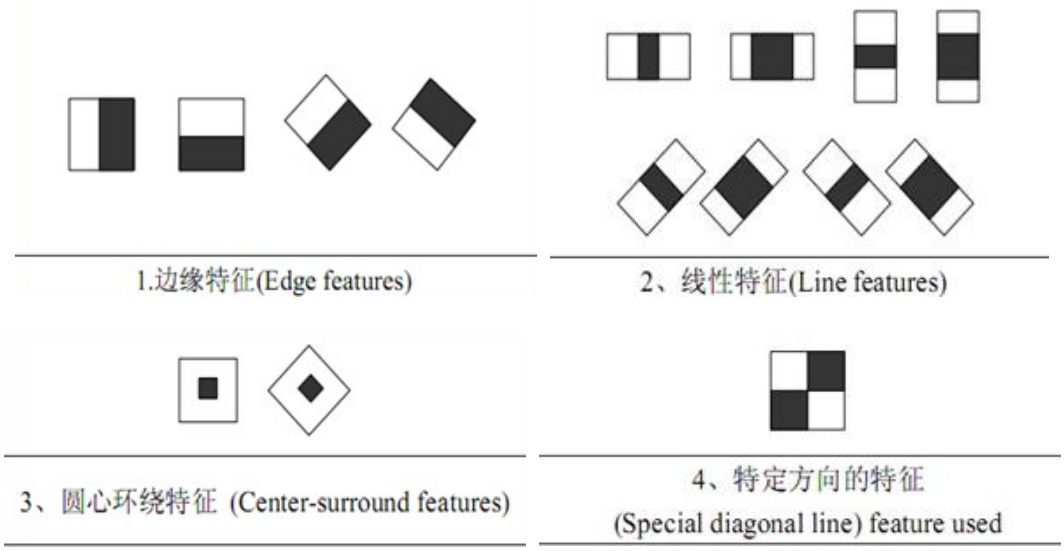

图 4 Haar 特征提取

在传统的检测中，有以下几种常用的特征提取方法。Haar 由于提取速度快，能够表达物体多种边缘变化信息，并且可以利用积分图快速计算，得到广泛的应用；LBP 更多的表达物体的纹理信息，对均匀变化的光照有很好的地适应性；HOG 通过对物体边缘使用直方图统计来进行编码，特征表达能力更强，在物体检测、跟踪、识别都有广泛的应用。传统特征设计往往需要研究人员经验驱动，更新周期往往较长，通过对不同的特征进行组合调优，从不同维度描述物体可以进一步提升检测精度，如 ACF 检测，组合了 20 种不同的特征表达。特征提取利用在计算机视觉和图像处理中用来进行物体检测的特征描述算子，不仅在传统的目标检测中有着至关重要的作用，对后来的卷积神经网络算法的设计也起到了启发作用。

目前常用的卷积神经网络目标检测模型分为Two-stage检测模型和One-stage检测模型。

4. 1. 5 Two-stage 检测模型

Ross B.Girshick 在论文中将检测抽象为两个过程，一是基于图片提出若干可能包含目标的区域，二是在提出的这些区域上运行 Alexnet 分类网络，得到目标的类别。这就是检测任务转化为区域上的分类任务的 R-CNN 框架。R-CNN 不仅能够提高候选边界框质量，还能提取高级特征，其工作可以分为三个阶段：生成候选区域、基于 CNN 的特征提取和分类与定位。R-CNN 采用选择搜索为每个图片产生 2k 个区域提议，选择性搜索方法依赖于简单的自下而上的分组和显著性提示，从而快速提供任意大小的更准确的候选框，并有效减少了搜索空间。在分类上，R-CNN 使用区域提议进行评分，然后通过边界框回归调整评分区域，并用非极大值抑制进行过滤以产生用于保留目标位置的最终边界框。流程图如图

所示。

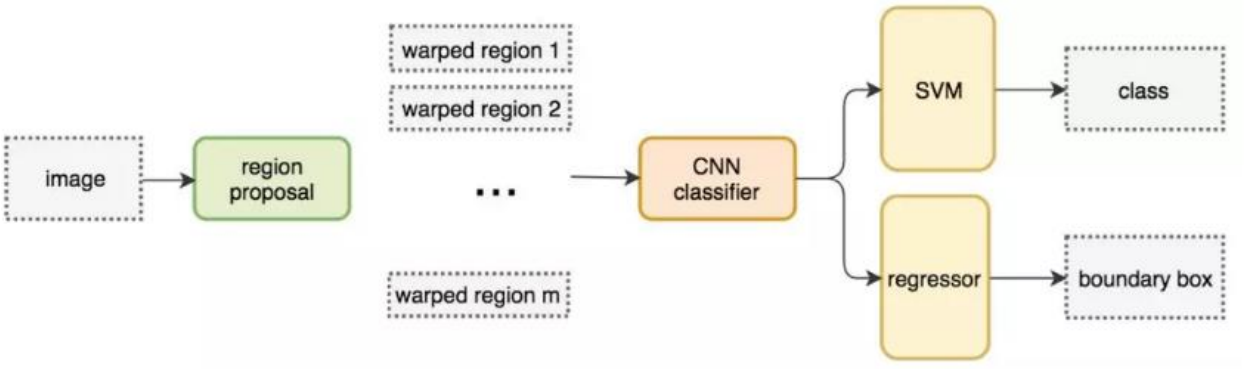

图 5 RCNN 流程图

尽管 R-CNN 在准确程度上相比于传统方法有所提高，并且将 CNN 应用于实际目标检测中具有重要意义，但是仍然存在一些缺点。

（1）全连接层的存在使得输入图像的大小固定，直接导致每个评估区域的整个 CNN 都需要重新计算，花费大量的测试时间；

（2）R-CNN 的训练是分为多阶段完成的。首先，对提议的卷积网络进行了微调。然后，将通过微调学习的 softmax 分类器替换为 SVM。最后，训练边界框回归器；

（3）训练阶段时间长，占用空间大。从不同区域提议提取并存储在磁盘上的特征占用很大的内存。

（4）尽管选择性搜索可以生成具有较高召回率的区域提议，但是获得的区域仍然存在多余的部分，并且这个过程需要耗费很长时间。

为了解决上述问题，Girshick 引入了多任务损失，并提出了一种名为 Fast R-CNN 的卷积神经网络。Fast-R-CNN 在卷积的最后一层采用感兴趣区域(Region of Interests，ROI)池化生成固定尺寸的特征图，另外，使用多任务损失函数，将边界回归加入到 CNN 网络中进行训练。Fast R-CNN 体系中整个图像都经过卷积层处理后生成特征图。然后，从每个感兴趣区域的区域提议中提取一个固定长度的特征向量。感兴趣区域层是特殊的 SPP 层，它只有一个金字塔。它将每个特征向量送入一系列全连接层，最后分为两个同级输出层。一个输出层产生所有 C+1 类(C 目标类加上一个“背景”类)的 softmax 概率，另一输出层用四个数值对编码的边界框的位置进行编码。这些过程中的所有参数(生成区域提议的除外)都通过端到端的多任务损失函数进行了优化。

4. 1. 6 One-stage 检测模型

Redmon 等人提出了一个名为 YOLOv1 的框架，该框架利用最上层的整个特征图来预

测多个类别和计算边界框的置信度。YOLOv1 的基本思想如图所示，YOLOv1 将输入图像划分为$S\times S$网格，每个网格单元负责预测以该网格单元为中心的目标的边界框以及其对应的置信度分数。其中置信度表示存在目标的可能性并显示其预测的置信度。与此同时，无论边界框的数量为多少，都会在每个网格单元中预测C个条件类概率。

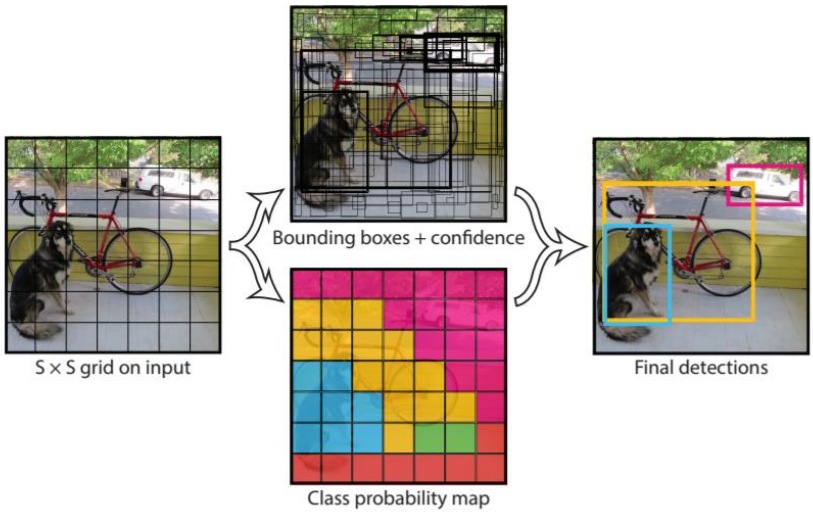

图 6 YOLO 算法检测流程

在 YOLOv1 的基础上，一种改进的版本 YOLOv2 被提出，YOLOv2 提出了一个新的拥有 19 个卷积层和 5 个池化层的分类主干网络，即 Darknet-19, 并采用了更强大的深度卷积 ImageNet 主干框架, 输入图片的分辨率由原来的 224×224 直接设置为 448×448，这使得学习到的权重对于获取微小信息更加敏感。除此之外，YOLOv2 借鉴 Faster R-CNN 中设定先验框的策略，使用全卷积网络，用 K-means 聚类算法获取先验框的宽和高，并通过预测偏移量来降低网络训练难度。最后与批归一化、多尺度处理技术一起形成训练网络。

作为 YOLOv2 的改进版本 YOLOv3，提出了更深入，更强大特征提取网络 Darknet-53。为了适应包含许多重叠标签的复杂数据集，YOLOv3 使用多标签分类。在对边界框进行预测时，YOLOv3 在 3 个不同比例的特征图进行预测，是当时速度和精度最均衡的目标检测网络。而后续 YOLOv4 和 YOLOv5 又对算法做了进一步优化，使得 YOLO 不仅可以用于学术研究，更有可能应用到实际生产中。

基于 Two-stage 的框架由几个阶段组成，区域提议生成，使用 CNN 进行特征提取，分类和边界框回归，这些阶段通常是分开训练。因此，对 Two-stage 检测模型处理不同部分都需要花费的时间成为实时目标检测的瓶颈。与之对应的 One-stage 检测模型，基于全局回归/分类的一步框架，可以直接从图像像素映射到边界框坐标和分类概率，减少时间开销。

目标检测经过几十年的研究发展，已经逐渐成为一个成熟的研究领域。凭借其强大的学习能力以及在处理遮挡，尺度转换和背景切换方面的优势，基于深度学习的目标检测已成为近年来的研究热点。实践证明深度学习算法拥有更好的泛用性和鲁棒性，自 2013 年之后，整个学术界和工业界都逐渐利用深度学习来做物体检测。而对比 Two-stage 检测模型和 One-stage 检测模型不难发现，在实时性和时间成本上 One-stage 检测模型都有较大优势，对于实时检测任务，采用 One-stage 检测框架的 YOLO 模型具有更好的表现。

综上所述，本文采用 YOLOv5 进行冰壶的检测。

4.2 理论分析与计算

4.2.1 坐标系转换

利用摄像机拍摄冰壶运动视频的过程，是将真实的三维空间上的点映射到二维空间平面上的过程。我们可以用小孔成像模型来简单理解该过程，任何来自真实世界的光线只能通过小孔才能到达成像平面。所以图像中二维像点和真实三维像点之间存在着对应关系，我们可以利用图像中的二维点信息来恢复三维场景信息。首先需要定义三种坐标系：图像坐标系，像素坐标系，相机坐标系以及世界坐标系。

图像坐标系(x,y)是物体在展现给我们的二维图像中的坐标，是以图像中心O点为原点。

像素坐标系(μ,ν)是二维图像以左上角为原点，X 轴沿着水平方向向左，Y 轴竖直向下。

相机坐标系(X_c,Y_c,Z_c)是以相机感光原件为坐标系原点，Z_c 轴与镜头的光轴平行。所有的相机坐标系中的坐标点是以摄像机的视角而言的，相机坐标系中的坐标会随着相机摆放位置而改变。

世界坐标系(X_w,Y_w,Z_w)是由人们自己定义的，真实空间中的坐标系，世界坐标系和相机坐标系没有任何关系，但是我们可以通过物理空间中某点的坐标可以在两个坐标系之间转换。

假设固定摄像机，取三维空间一点P，点P在相机坐标系下的坐标为(X,Y,Z)。点p在二维图像中的图像坐标为(x,y)。由于光轴垂直于成像平面，可知点p在相机坐标系中的坐标是(x,y,z)，其中$z=f$（f 为焦距)。

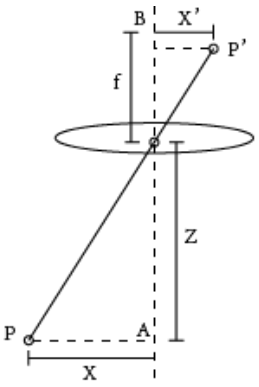

图 7 相机成像原理图

根据上图中三角形的相似关系，可以得出如下公式：

$$\begin{cases} xZ = Xf \\ yZ = Yf \\ z = f \end{cases} \tag{4-1}$$

根据此公式可以推导出相机拍摄的内参数和外参数。

相机的内参数主要由两部分构成，一个是相机焦点到成像平面的距离焦距 f，另一个是图像坐标系到像素坐标系的变换关系。上面推导的图像坐标 $p(x,y)$ 是在图像坐标系下，以成像平面的中心为原点。而实际像素点的坐标原点通常是在图像的左上角，X 轴沿着水平方向向左，Y 轴竖直向下。像素是一个矩形块，假设其长度和宽度分别为 α 和 β，所以图像坐标系和像素坐标系之间差了缩放和平移操作。

假设像素坐标的水平方向为 μ，竖直方向为 ν，那么将一个图像坐标 (x,y) 在水平方向缩放 α 倍，在竖直方向缩放 β 倍，同时对原点进行平移 (c_x,c_y)，就可以得到像素坐标系下坐标 (μ,ν)，其公式如下：

$$\begin{aligned} \mu &= \alpha \cdot x + c_x \\ \nu &= \beta \cdot y + c_y \end{aligned} \tag{4-2}$$

将式（4-1）带入（4-2），并转换为矩形形式得：

$$\begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = Z \begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4-3}$$

由此我们可以计算出相机得内参矩阵 K

$$K=\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4-4}$$

其中 $f_x=\alpha\cdot f$ 和 $f_y=\beta\cdot f$ 都与相机得焦距有关。c_x 和 c_y 是平移距离，与相机拍摄图像得大小有关。因此我们在求解相机内参数时，需要固定焦点拍摄。

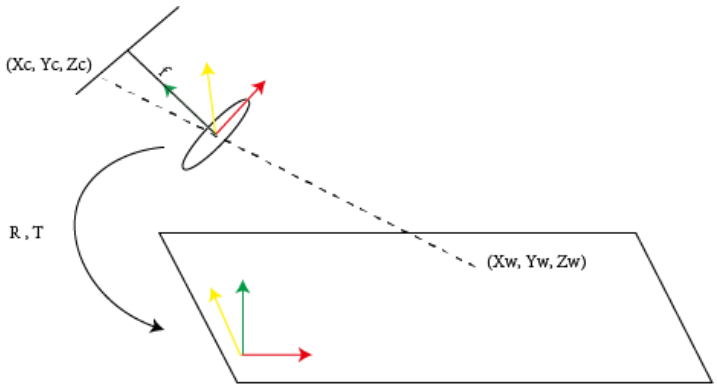

图 8 相机与场地坐标模拟图

在求解相机的外参数时，需要了解相机坐标系到世界坐标系的转换关系。假设 P_c 是点 P 的相机坐标系下的坐标 (X_c,Y_c,Z_c)，P_w 是其世界坐标系下坐标 (X_w,Y_w,Z_w)。可以通过一个旋转矩阵 R 和一个平移向量 t 将相机坐标转换为世界坐标。

$$P_c=RP_w+t \tag{4-5}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}+\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{4-6}$$

当我们知道了相机的内参和外参后，通过合并式（4-3）和（4-6）得：

$$P_i=K\left[R\,|\,T\right]P_w \tag{4-7}$$

$$\begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix}=\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4-8}$$

其中内参矩阵 K 由相机本身决定，当固定相机拍摄角度后，旋转矩阵 R 和平移向量 t 都是可以确定的。因此我们可以在任意角度下拍摄冰壶运动视频并且求出运动轨迹及距离。

4.2.2 旋转角度

在计算冰壶的旋转角度时，在冰壶把手上标记黑色长条区域，通过霍夫变换的方法检测直线，最终达到计算旋转角度的效果。

霍夫变换检测直线的原理，在笛卡尔坐标系下，一条直线的方程可以写成 $y=k\cdot x+b$，现在我们对该直线方程变形后有 $b=-k\cdot x+y$，这时我们不在把 k,b 看作是直线方程的系数，而是将其分别作为参数空间的自变量和因变量，而 x,y 则看作是系数。这样我们就实现笛卡尔坐标系与其参数空间之间的映射。相当于把笛卡尔坐标系下的点映射为参数空间中的一组直线。

对于每个图像来说，图像中每个像素坐标点经过变换都变成对直线特质有贡献的统一度量，一条直线在图像中是一系列离散点的集合，通过一个离散极坐标公式便可以表达出离散点几何等式：

$$x\cdot\cos\theta+y\cdot\cos\theta=r \tag{4-9}$$

其中，角度 θ 表示 r 与 x 轴之间的夹角，r 为到直线的垂直距离。任何在直线上的点都可以用坐标 x_i,y_i 表示，其图像展示为：

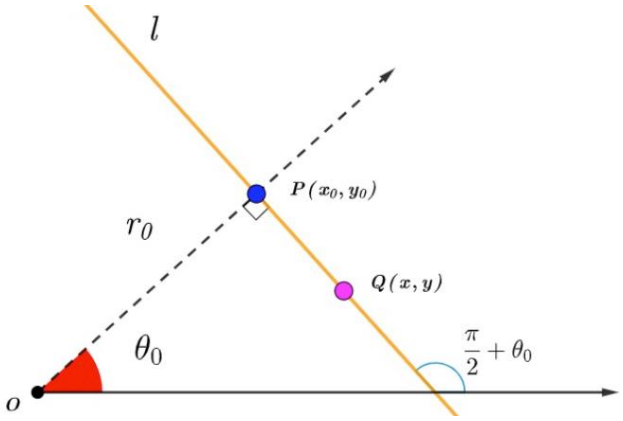

图 9 极坐标系直线方程

对于直线 l 上的任意一点 Q 而言，我们可以得出：

$$k=\frac{y-y_0}{x-x_0},\text{其中}\begin{cases}x_0=r_0\cdot\cos\theta_0\\y_0=r_0\cdot\cos\theta_0\end{cases} \tag{4-10}$$

$$k=\tan\left(\frac{\pi}{2}+\theta_0\right)=-\frac{\cos\theta_0}{\sin\theta_0} \tag{4-11}$$

联立式（4-10）和（4-11）可求得 Q 点在极坐标系下的参数空间方程：

$$r_0 = x \cdot \cos\theta_0 + y \cdot \cos\theta_0 \tag{4-12}$$

这样我们就可以实现极坐标系与参数空间之间的映射：

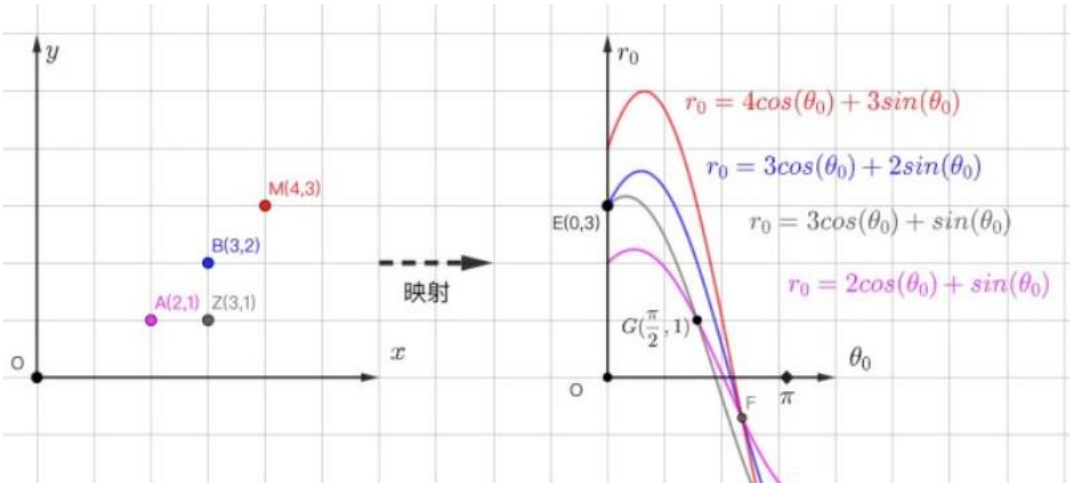

图 10 极坐标系与参数空间映射

通过极坐标系的参数空间将平面的散点集一一映射到其中，然后在参数空间中找出曲线相交较多的点，则其即为原散点集中被检测出的直线。故依据上图，我们可判定参数空间中的 F 点即为原散点集中被检测到的直线。通过霍夫变换，可以大大提高直线检测的效率。

4.2.3 轨迹预测

利用卡尔曼滤波通过对冰壶位置观察序列来预测出冰壶运动下一点的位置坐标。卡尔曼滤波器可以简单概述为 5 条公式，首先是系统的状态方程：

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{4-13}$$

这个状态方程是根据上一时刻的状态和控制变量来推测此时的状态，w_{k-1} 是服从高斯分布的噪声，相当于预测过程中的噪声，它对应 x_k 中每一个分量的噪声，期望是 0。

卡尔曼滤波器的观测方程为：

$$z_k = Hx_k + v_k \tag{4-14}$$

v_k 是观测的噪声，服从高斯分布。

卡尔曼滤波算法有两个基本假设：（1）信息过程的足够精确的模型，是由白噪声所激发的线性动态系统。（2）每次的测量信号都包含着附加的白噪声分量。当满足上述上假设时，可以应用卡尔曼滤波算法。

卡尔曼滤波算法可以分为两个步骤，预测与更新。卡尔曼滤波器可以分为时间更新方程和测量更新方程。

时间更新方程（即预测阶段）根据前一时刻的状态估计值推算当前时刻的状态变量先验估计值和误差协方差先验估计值：

$$\hat{x}_{\bar{k}} = A\hat{x}_{k-1} + Bu_{k-1} \tag{4-15}$$

$$P_{\bar{k}} = AP_{k-1}A^T + Q \tag{4-16}$$

测量更新方程（即更新阶段）负责将先验估计和新的测量变量结合起来构造改进的后验估计。时间更新方程和测量更新方程也被称为预测方程和校正方程。因此卡尔曼滤波算法是一个递归的预测——校正方法。

$$K_k = \frac{P_{\bar{k}}H^T}{HP_{\bar{k}}H^T + R} \tag{4-17}$$

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k\left(z_k - H\hat{x}_{\bar{k}}\right) \tag{4-18}$$

$$P_k = \left(I - K_kH\right)P_{\bar{k}} \tag{4-19}$$

其中 $\hat{x}_{k-1}$ 和 $\hat{x}_k$ 分别表示 $k-1$ 时刻和 k 时刻的后验状态估计值，是滤波的结果之一，即更新后的结果，也叫最优估计（估计的状态，根据理论，我们不可能知道每时刻状态的确切结果所以叫估计）。$\hat{x}_{\bar{k}}$ 表示 k 时刻的先验估计值，是滤波的中间计算结果即根据上一时刻（$k-1$ 时刻）的最优估计预测的 k 时刻的结果，是预测方程的结果。

P_k 和 P_{k-1} 分别表示 $k-1$ 时刻和 k 时刻的后验估计协方差（即 $\hat{x}_{k-1}$ 和 $\hat{x}_k$ 的协方差，表示状态的不确定度），是滤波的结果之一。$P_{\bar{k}}$ 表示 k 时刻的先验估计协方差，是滤波的中间计算结果。

H 代表着状态变量到测量的旋转矩阵，表示将状态和观测连接起来的关系，卡尔曼滤波里为线性关系，它负责将 m 维的测量值转换到 n 维，使之符合状态变量的数学形式，是滤波的前提条件之一。z_k 是测量值，相当于滤波的输入。K_k 是滤波增益矩阵，是滤波的中间计算结果，卡尔曼增益。 A 表示状态转移矩阵，实际上是对目标状态转换的一种猜想模型。例如在机动目标跟踪中， 状态转移矩阵常常用来对目标的运动建模，其模型可能为匀速直线运动或者匀加速运动。当状态转移矩阵不符合目标的状态转换模型时，滤波会很快

发散。

Q是过程激励噪声协方差（系统过程的协方差）。该参数被用来表示状态转换矩阵与实际过程之间的误差。因为我们无法直接观测到过程信号， 所以Q的取值是很难确定的。是卡尔曼滤波器用于估计离散时间过程的状态变量，也叫预测模型本身带来的噪声。 R是测量噪声协方差。滤波器实际实现时，测量噪声协方差R一般可以观测得到，是滤波器的已知条件。 B是将输入转换为状态的矩阵。

4.3 设计与实现

4.3.1 程序流程

根据前述方案选择，结合理论分析及计算，设计程序实现相应功能，文中仅介绍程序的核心思想及实现方法，核心代码见附录。

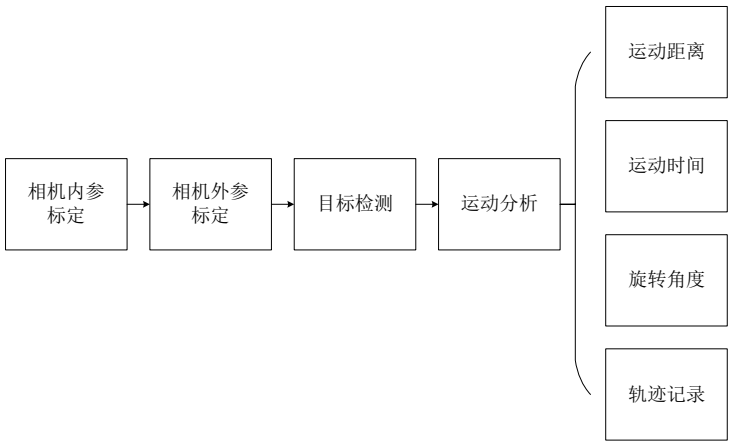

图 11 系统流程图

4.3.2 相机内参标定

首先进行相机内参标定求解，现在手机的镜头都有自动对焦功能，不同的焦点下内参是不一样的，因此需要用手动模式固定对焦点到无穷远。选取 9*6 的棋盘坐标实现内参矩阵求解，求得内参矩阵K和镜头畸变系数$dist$，

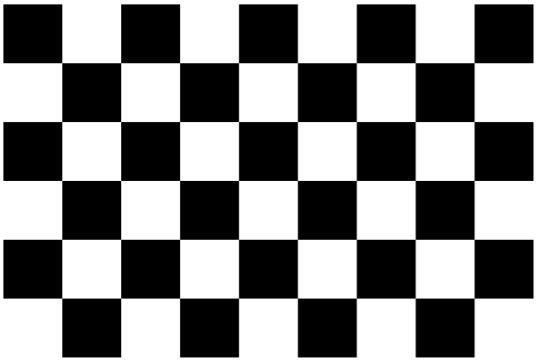

图 12 棋盘图案

然后从不同角度，不同距离，不同位置拍摄二十张左右的棋盘图片。

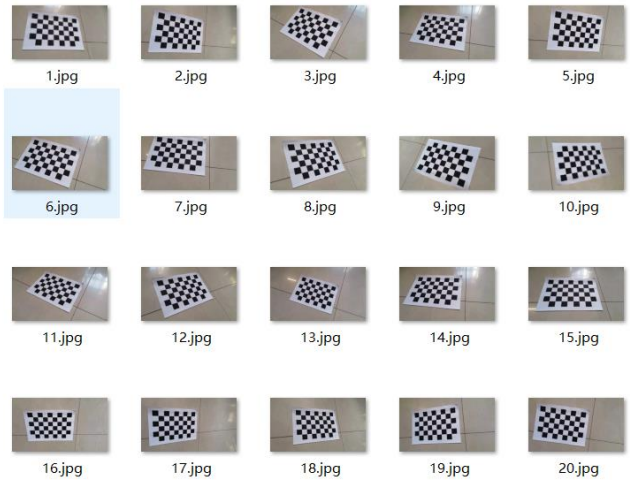

图 13 棋盘图案标志点

4.3.2 相机外参标定

得到了相机内参矩阵 K 后。接下来固定已知参数的相机的位置，保持与计算内参时一样的焦点，拍摄场地标点位置的图像，获取此次拍摄的相机外参：旋转矩阵 R 和平移向量 t。我们需要提前设定四个标点的世界坐标系，以便通过比例计算出冰壶在真实世界中的位置。

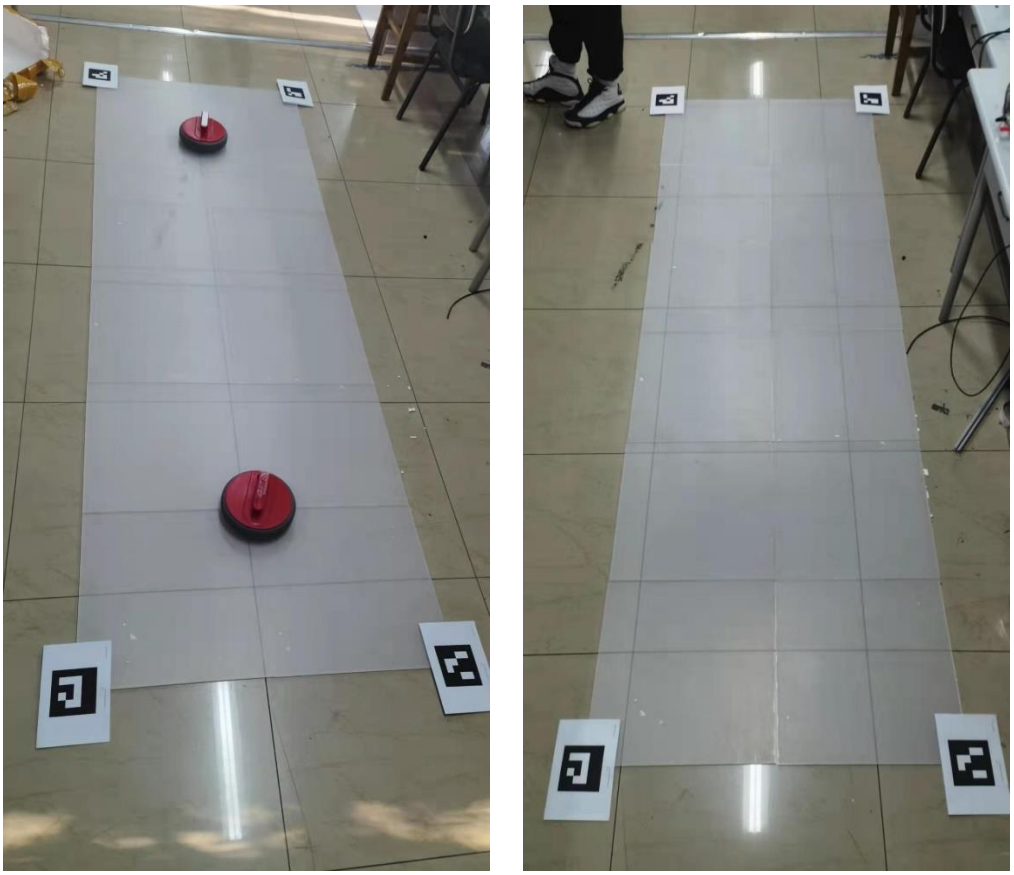

图 14 相机外参标定

有了一组物理世界中的 3D 点坐标和图像中的 2D 点坐标，又由于相机内参已知，可以求解旋转矩阵 R 和平移向量 t，也就是从世界坐标系到相机坐标系的变换关系。

确定好相机的内参和外参后，便可以拍摄冰壶运动视频，利用 YOLOv5 识别冰壶位置并将其返回到图像坐标系上，通过式（4-7）计算得出冰壶在真实世界中的位置。接下来开始完成竞赛要求。

4.3.3 目标检测算法

由于运动检测要求高实时性，小目标敏感和识别精度高等特性，结合冰壶外观特性选择了在测试过程中变现亮眼的 YOLOv5 深度检测网络。YOLOv5 在目标检测算法中采用图像划分为多个区域，按区域进行检测及定位工作，在卷积神经网络的基础上进行改进，实现了卷积的滑动窗口检测，仅通过一次向前传播即可获得检测结果，在不是准确性的同时实现了高实时性的检测。

YOLO 网络主要由三个主要组件组成，Backbone，在不同图像细粒度上聚合并形成图像特征的卷积神经网络；Neck，一系列混合和组合图像特征的网络层，并将图像特征传递到预测层；Head，对图像特征进行预测，生成边界框并预测类别。

YOLOv5 使用 CSPDarknet 作为 Backbone，从输入图像中提取丰富的信息特征。这是一种跨阶段局部网络，采用这种结构解决了其他大型卷积神经网络框架 Backbone 中网络优化的梯度信息重复问题，将梯度的变化从头到尾地集成到特征图中，因此减少了模型的参数量和 FLOPS 数值，既保证了推理速度和准确率，又减小了模型尺寸。这样可以有效缓解梯度消失问题(通过非常深的网络很难去反推丢失信号) ，支持特征传播，鼓励网络重用特征，从而减少网络参数数量。Neck 主要用于生成特征金字塔。特征金字塔会增强模型对于不同缩放尺度对象的检测，从而能够识别不同大小和尺度的同一个物体。Head 主要用于最终检测部分。它在特征图上应用锚定框，并生成带有类概率、对象得分和包围框的最终输出向量。

为了实现使用 YOLOv5 对冰壶进行识别，首先要制作数据集，对冰壶在多个角度方位和不同的遮挡情况下进行拍照，然后利用标注工具将冰壶人工标注出来并输出为训练数据。利用制作的数据集即可训练 YOLO 网络模型，训练模型过程中各损失函数的变化见下图。

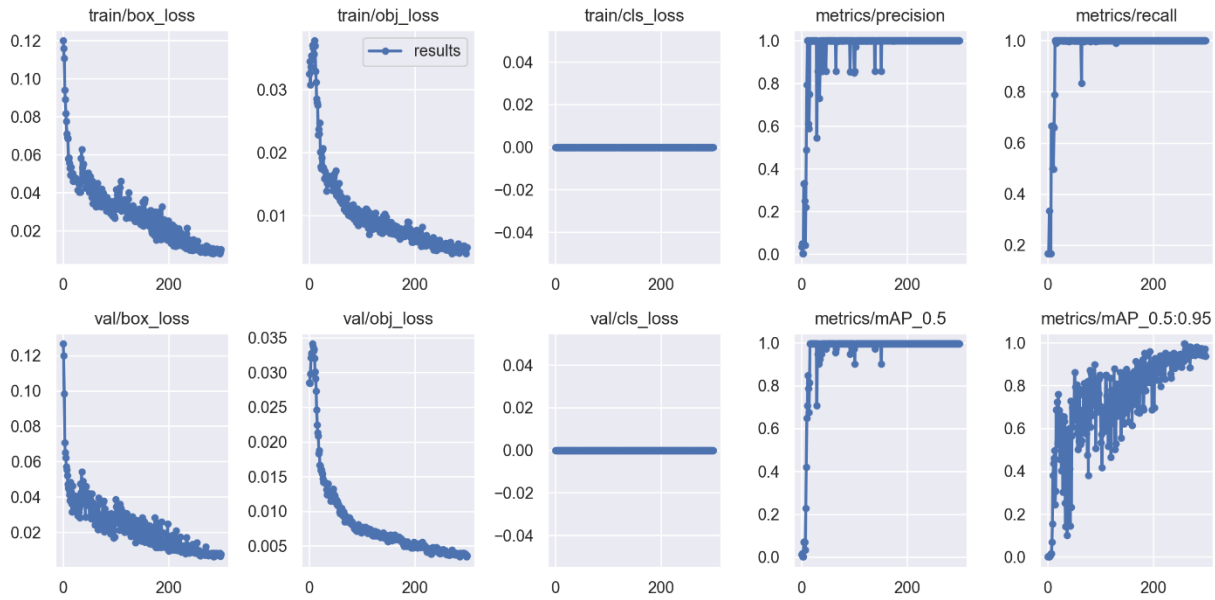

图 15 训练过程中的 loss 函数

目标检测结果测试，如下图所示，可以看到冰壶被准确的框选出。

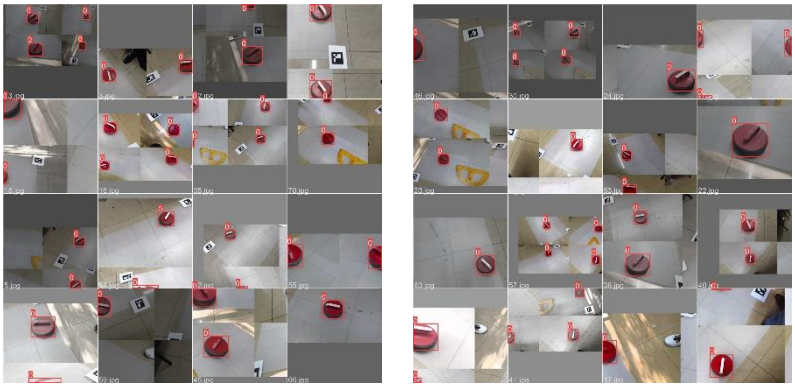

图 16 测试识别冰壶

4.3.4 冰壶运动分析

在目标检测程序中我们可以获取被框定的物体在图像中的位置信息，通过理论分析中的坐标系转换方法结合已标定的世界坐标系，不难计算出出冰壶在真实世界中的坐标。通过计算视频中每一帧的冰壶真实坐标，即可对冰壶的运动时间，距离和轨迹进行分析。

通过检测冰壶的进入规定区域后坐标是否发生变化来判断冰壶是否发生运动，在发生运动后，如果一段时间内冰壶坐标没有发生变化，那么判断冰壶停止运动。利用程序截取发生运动到停止运动见的视频数据，判断截取的视频数据中包含的帧数，在利用视频总时长与帧数的比值计算，每一帧对应的时间片长度从而计算运动时间。

在检测到冰壶发生运动后，记录冰壶的坐标，并将其映射到对应比例的场景二维坐标系中，并在对应位置打印红色圆点完成轨迹的记录。而运动距离的计算采用微元法，对相邻的运动记录点计算其位移距离并累计其在运动过程中产生的位移和，将运动开始到运动结束之间的位移和作为最终的运动距离。

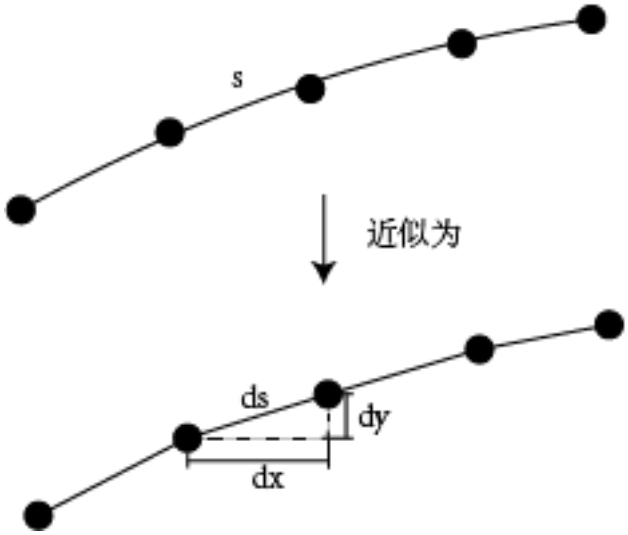

图 17 微元法计算距离

在冰壶的把手上做黑色条形标识，利用霍夫曼直线检测算法对标识区域进行提取，获取冰壶把手的角度。在运动过程中记录每一帧冰壶把手的角度变化，累计其旋转总角度变化即可获取冰壶在运动过程中的旋转角度。

4.4 测试方案与测试结果

首先进行相机标定，计算出相机内参外参，并固定相机拍摄图片获取世界坐标系，经现场验证，定位冰壶位置精度达到厘米级。

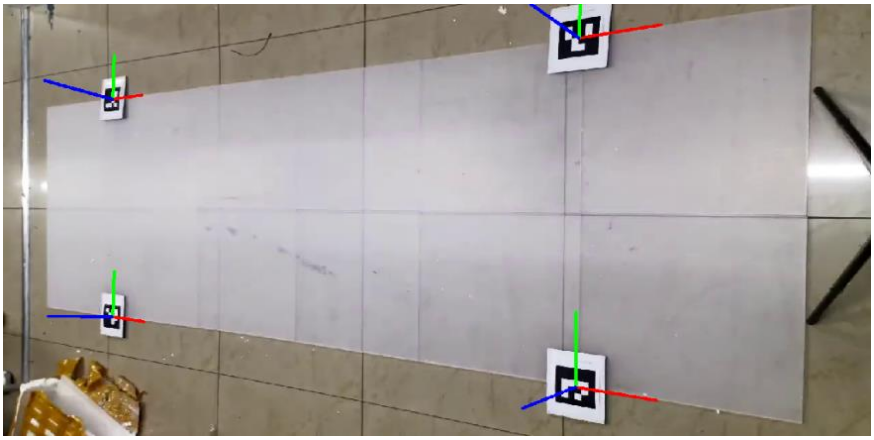

图 18 世界坐标系建立

4.4.1 冰壶直线运动

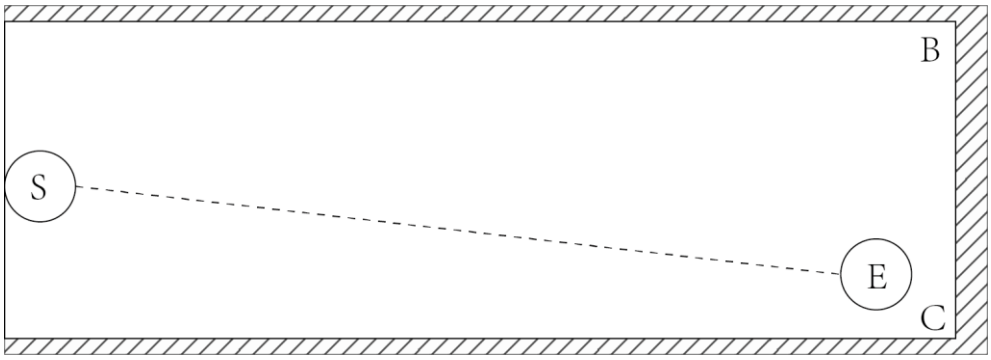

图 19 冰壶直线运动

根据题目要求，抛出冰壶使其做直线运动，测试并记录多种不同运动情况下的运动时间及运动距离，并对计算结果的正确性和精确度进行评价。

（1）冰壶进行直线运动并在场地中部停止运动，记录运动时间、距离及平均速度，并在显示背景板上显示冰壶运动轨迹。

测试结果如图：

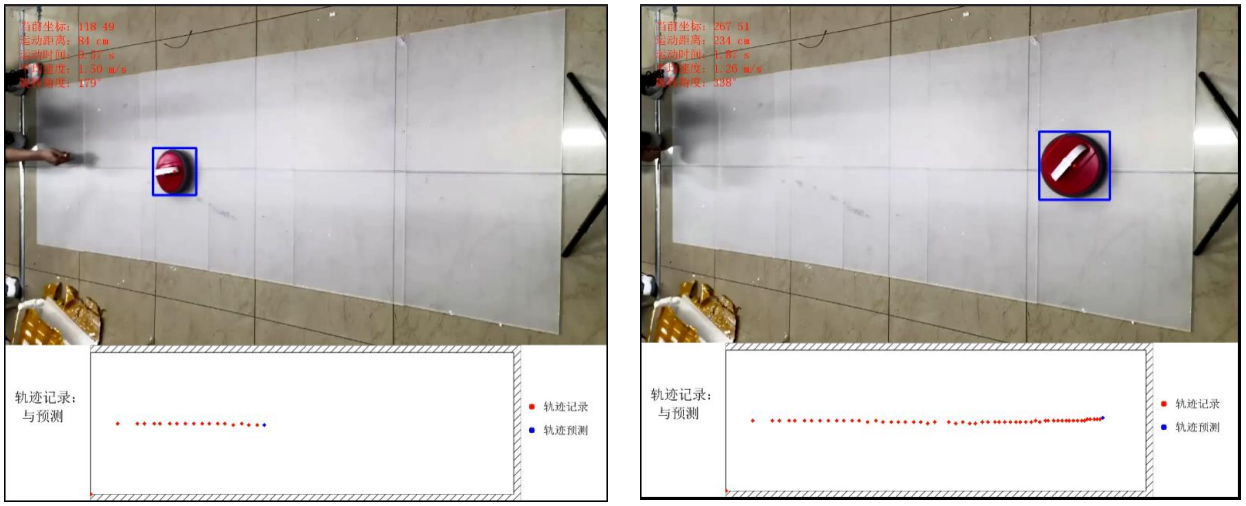

图 20 测试冰壶直线运动

（2）冰壶进行直线运动并撞击到 BC 处挡板回弹到场地中部停止运动，记录运动时间、距离及平均速度，并在显示背景板上显示冰壶运动轨迹。

由于测试时场地环境有限，没有实现回弹效果。对于冰壶在运动过程中超出场地范围的情况，并不记录其运动时间，距离及旋转角度。

4.4.2 冰壶旋转运动

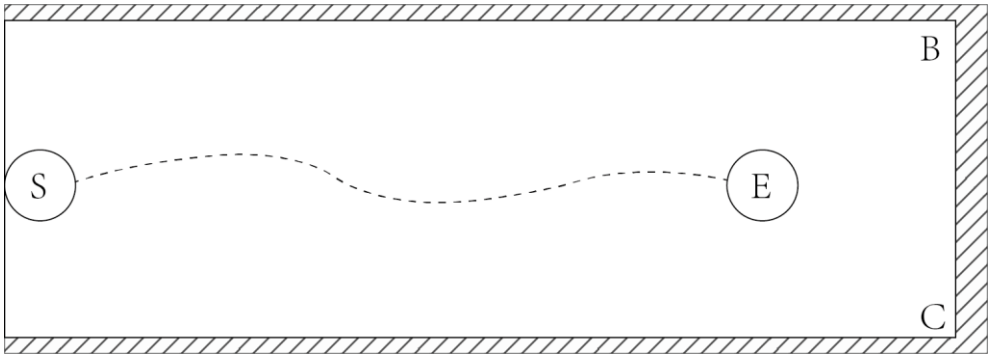

图 21 冰壶旋转运动

抛出冰壶使其做旋转运动，测试并记录不同运动情况下运动时间，运动距离及旋转角度，并对计算结果的正确性和精确度进行评价。

冰壶进行旋转运动并在场地中部停止运动，记录运动时间，距离，平均速度及旋转角度，并在显示背景板上显示冰壶运动轨迹。

测试结果如图：

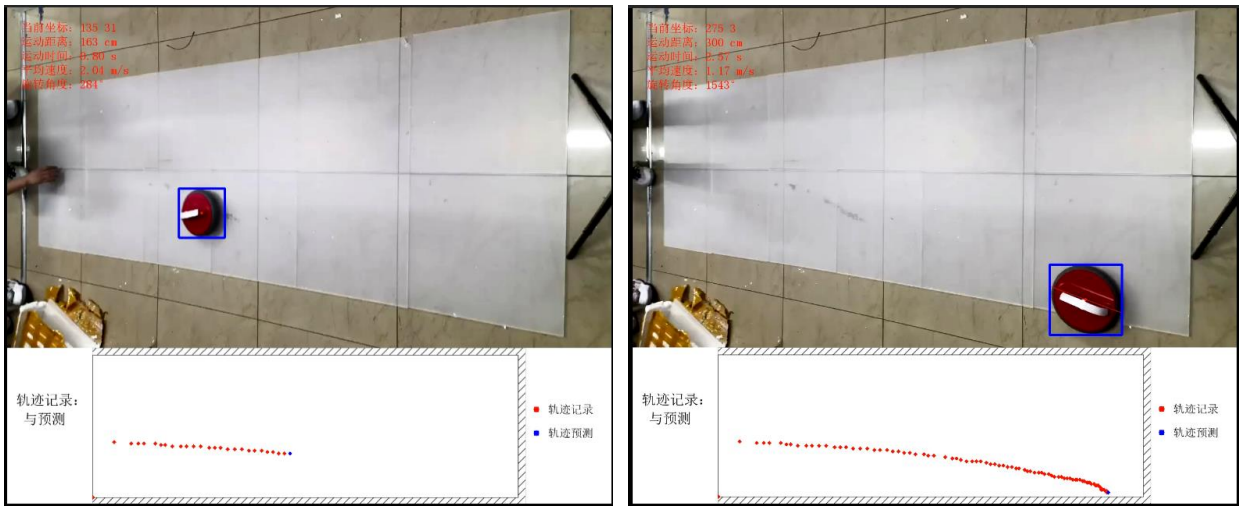

图 22 测试冰壶旋转运动

经过两次测试冰壶运动实验，得出以下结果：

表 1 测试结果

	运动距离（cm）	运动时间（s）	平均速度（m/s）	旋转角度(度)	轨迹预测精度
直线运动	247	2.20	1.12		厘米级
旋转运动	301	2.57	1.17	1543	厘米级

5. 总结与展望

5.1 模型评价

在内参标定过程中通过实际计算与理论结合，最终确定利用不同角度，不同距离，不同位置拍摄二十张左右的棋盘图片，可以获得较为准确的相机内参。而相机外参的标定对设定的世界坐标系标点要求较高，会受到地板的镜面反射，标点的摆放位置，标点距离镜头的远近等因素的影响。

根据前述获取的相机内外参，对冰壶的定位精度可以达到厘米级。冰壶检测模型使用70张数据集训练而得，对冰壶的识别可以达到90%以上的置信度，对冰壶运动轨迹记录精度达到厘米级，对冰壶运动时间记录精度达到毫秒级，但是算法仍可以在检测速度和精度方面进行优化。

5.2 总结

本设计基于单目视觉对冰壶运动进行定位，计算显示运动距离及时间，并画出其运动轨迹。构建了一套完整的智能目标检测和视觉测量的实时反馈系统。实验测试表明该系统可以精确锁定冰壶运动的全过程，实现实时跟踪与测量计算，圆满完成竞赛规定的要求。本设计主要完成工作如下：

（1）对于如何识别冰壶并实现其运动过程中的跟踪，本设计对比其他检测方法，例如：基于颜色聚类、边缘聚类方法等。最终选择了实时性和鲁棒性较高的YOLOv5卷积神经网络进行冰壶识别。

（2）针对冰壶的自身结构，运动特点和赛道场景设计单目相机采集视频，并利用人工标定的方法确定比赛场地在拍摄视频中位置。通过一系列坐标系转换，从图像坐标系到像素坐标系再到相机坐标系，最后到世界坐标系，能够精确的实现二维坐标到三维坐标的转换，以达到计算冰壶运动距离及时间的要求。

（3）检测在冰壶把手上标记黑色长条区域，通过霍夫变换检测直线方法，确定冰壶进入比赛场地时冰壶把手与场地之间的夹角。对每一帧识别图像中的冰壶把手与上一帧中的做夹角分析，获取单位帧之间的旋转角度差，并求其角度和，最终获得冰壶整个运动过程中的旋转角度。

（4）创建一个与比赛场地长宽比例相同的画布，并将拍摄视频中冰壶的识别位置同步显示到画布上，实现对冰壶轨迹的实时跟踪定位。

未来我们将在目前研究的基础上，将进一步优化相关算法，考虑加入选手抛出冰壶的

姿态分析、冰壶博弈策略、操作控制等技术完整智能冰壶识别系统。同时，也能够为计算机视觉技术在现实世界中的应用研究提供学术基础和研究平台。

6. 参考文献

[1]李文佳. 基于智能视觉的冰壶运动检测与测量[D].哈尔滨工业大学,2020.DOI:10.27061/d.cnki.ghgdu.2020.004616.

[2]伏选杰,田畅.基于单目视觉的 FMS 人体姿态识别[J].计算机时代,2021(10):23-27.DOI:10.16644/j.cnki.cn33-1094/tp.2021.10.006.

[3] Francesco Barone and Marco Marrazzo and Claudio J. Oton. Camera Calibration with Weighted Direct Linear Transformation and Anisotropic Uncertainties of Image Control Points[J]. Sensors, 2020, 20(4) : 1175-1175.

[4] Y.I Abdel-Aziz and H.M. Karara and Michael Hauck. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry *[J]. Photogrammetric Engineering & Remote Sensing, 2015, 81(2) : 103-107.

[5]王特特,霍春宝,白博,苏锦瑾.基于 OpenCV 的车道线智能检测方法[J].信息技术与信息化,2021(09):224-227.

[6]朱鸿宇,杨帆,高晓倩,李学娇.基于级联霍夫变换的车道线快速检测算法[J].计算机技术与发展,2021,31(01):88-93.

[7] Hesheng Yin et al. SLAM-Based Self-Calibration of a Binocular Stereo Vision Rig in Real-Time[J]. Sensors, 2020, 20(3) : 621.

[8] Fengli Liu et al. Calibration Technique of a Curved Zoom Compound Eye Imaging System[J]. Micromachines, 2019, 10(11) : 776-776.

[9]张桂杰,郭泽旸,韩睿,杨雁升,王帅.基于 YOLOv5 的人车流量监测系统设计与实现[J].吉林师范大学学报(自然科学版),2021,42(04):118-126.DOI:10.16862/j.cnki.issn1674-3873.2021.04.021.

[10]朱鸿宇,杨帆,高晓倩,李学娇.基于级联霍夫变换的车道线快速检测算法[J].计算机技术与发展,2021,31(01):88-93.

[11] Yue Zhao and Fengli Yang and Xuechun Wang. Camera calibration using spheres and the line connecting two projections of the spherical centers[J]. Journal of Optics, 2019, 48(14) : 491-498.

[12]范丽丽,赵宏伟,赵浩宇,胡黄水,王振.基于深度卷积神经网络的目标检测研究综述[J].光学精密工程,2020,28(05):1152-1164.

7. 附录

设计核心代码

```
def camera_calibration(h, w, image_root):
    # 设置阈值
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
    filepath = 'parameters/calibration.npz'

    # 世界坐标系下的交界点坐标
    checkerboard_point = np.zeros((w * h, 3), np.float32)
    # 去掉 Z 轴，记为二维矩阵,例如(0,0,0), (1,0,0), (2,0,0) ....,(8,5,0)
    checkerboard_point[:, :2] = np.mgrid[0:w, 0:h].T.reshape(-1, 2)

    # 在世界坐标系中的三维点
    checkerboard_points = []
    # 在图像平面的二维点
    image_points = []

    images = glob.glob(image_root)
    print('./Calibration/*.jpg')
    if len(images) == 0:
        logging.error('Not find Target Images')
    for fname in images:
        image = cv2.imread(fname)
        try:
            gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        except:
            logging.error('File format appear Error')
            return
        # 找到棋盘格角点
        ret, corners = cv2.findChessboardCorners(gray_image, (w, h), None)
        if ret is True:
            cv2.cornerSubPix(gray_image, corners, (11, 11), (-1, -1), criteria)
            checkerboard_points.append(checkerboard_point)
            image_points.append(corners)
            logging.info('{} find corners'.format(fname))
            # 将角点在图像上显示
            # cv2.drawChessboardCorners(image, (w, h), corners, ret)
            # cv2.imshow('findCorners', image)
            # cv2.waitKey(2)

        else:
            logging.warning('{} not find corners'.format(fname))
    cv2.destroyAllWindows()
```

```
    # # 标定
    ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(checkerboard_points, image_points, gray_image.shape[::-1], None, None)
    print(mtx)
    print(dist)

    h, w = gray_image.shape[:2]
    newcameramtx, roi = cv2.getOptimalNewCameraMatrix(mtx, dist, (w, h), 1, (w, h))
    print(newcameramtx)

    if os.path.exists(filepath):
        os.remove(filepath)
    # np.savez('parameters/calibration.npz', mtx, dist)
    np.savez(filepath, newcameramtx, dist)

def get_datum_coordinates(image, mtx, dist):
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # detect aruco markers
    # DICT_6X6_50 字典
    aruco_dict = aruco.Dictionary_get(aruco.DICT_4X4_50)
    parameters = aruco.DetectorParameters_create()

    # rejectedImgPoints 可以暂时忽略，适用于调试目的和“重新查找”策略
    corners, ids, rejectedImgPoints = aruco.detectMarkers(gray_image, aruco_dict, parameters=parameters)
    rvecs, tvecs, _ = aruco.estimatePoseSingleMarkers(corners, 0.05, mtx, dist)

    for i in range(len(rvecs)):
        rvec = rvecs[i]
        tvec = tvecs[i]
        aruco.drawAxis(image, mtx, dist, rvec, tvec, 0.1)
    aruco.drawDetectedMarkers(gray_image, corners)
    # cv2.namedWindow('detect', cv2.WINDOW_NORMAL)
    # cv2.resizeWindow('detect', 1024, 512)
    cv2.namedWindow('detect1', cv2.WINDOW_NORMAL)
    cv2.resizeWindow('detect1', 1024, 512)
    # cv2.imshow("detect", gray_image)
    cv2.imshow("detect1", image)
    cv2.waitKey(0)
    return corners, ids
```