

**计算建模**

**大项目结题报告**

题 目 卡尔曼滤波原理及应用

学 院 计算机科学与技术

专 业 计算机科学与技术

学 号 7203610821，7203610609，

7203610602，7203610525

学 生 张峻齐，宋浩，陈儒钊，王卓强

任 课 教 师 范晓鹏，刘绍辉

哈尔滨工业大学计算机科学与技术学院

2022.9

# 卡尔曼滤波原理及应用

1. **项目内容梳理（20分）**

为探索卡尔曼滤波的原理以及应用，本项目实现了一种基于阈值观测和卡尔曼滤波的目标跟踪方法。本项目使用HSV阈值构建二值化图像作为观测方式，并结合卡尔曼滤波器设计了算法流程，编写了相关阈值选择以及采样框模块，实现了对单目标或多目标的跟踪和预测，并在处理速率结果上与KCF算法做了比较，分析了二者算法的优缺点并做了相关阐述。

项目可以较好地追踪视频或者摄像头中的纯色物体，并利用卡尔曼滤波处理了遮挡问题，在自己的部分数据集上测试效果甚至超过了cv2库中的KCF算法。

项目成果可以应用在场干扰较小的球类运动或者简单纯色物体的跟踪场景。

小组成员分工如下：宋浩，张峻齐进行分析，建模和优化，王卓强，陈儒钊负责搜集资料，形成报告。小组定期举行会议，以沟通进度，商讨未来工作方向。

1. **特定主题(topic)或特定文献的内容介绍-发展现状和发展趋势，所研究主体的国内外现状，最新进展等（20分）**

**（一）研究现状**

卡尔曼滤波（Kalman filter）是一种高效率的[递归滤波器](https://zh.wikipedia.org/wiki/%E9%80%92%E5%BD%92%E6%BB%A4%E6%B3%A2%E5%99%A8" \o "递归滤波器)（[自回归](https://zh.wikipedia.org/wiki/%E9%A2%91%E5%9F%9F" \o "自回归模型)滤波器），它能够从一系列的不完全及包含[噪声](https://zh.wikipedia.org/wiki/%E9%9B%9C%E8%A8%8A_(%E9%80%9A%E8%A8%8A%E5%AD%B8)" \o "噪声 (通讯学))的[测量](https://zh.wikipedia.org/wiki/%E6%B5%8B%E9%87%8F" \o "测量)中，估计[动态系统](https://zh.wikipedia.org/wiki/%E5%8A%A8%E6%80%81%E7%B3%BB%E7%BB%9F" \o "动态系统)的状态。卡尔曼滤波会根据各测量量在不同时间下的值，考虑各时间下的[联合分布](https://zh.wikipedia.org/wiki/%E8%81%94%E5%90%88%E5%88%86%E5%B8%83" \o "联合分布)，再产生对未知变数的估计，因此会比只以单一测量量为基础的估计方式要准。卡尔曼滤波得名自主要贡献者之一的[鲁道夫·卡尔曼](https://zh.wikipedia.org/wiki/%E9%B2%81%E9%81%93%E5%A4%AB%C2%B7%E5%8D%A1%E5%B0%94%E6%9B%BC" \o "鲁道夫·卡尔曼)。

卡尔曼滤波在技术领域有许多的应用。常见的有飞机及太空船的[导引、导航及控制](https://zh.wikipedia.org/w/index.php?title=%E5%B0%8E%E5%BC%95%E3%80%81%E5%B0%8E%E8%88%AA%E5%8F%8A%E6%8E%A7%E5%88%B6&action=edit&redlink=1)。卡尔曼滤波也广为使用在[时间序列](https://zh.wikipedia.org/wiki/%E6%99%82%E9%96%93%E5%BA%8F%E5%88%97" \o "时间序列)的分析中，例如[信号处理](https://zh.wikipedia.org/wiki/%E4%BF%A1%E5%8F%B7%E5%A4%84%E7%90%86" \o "信号处理)及[计量经济学](https://zh.wikipedia.org/wiki/%E8%AE%A1%E9%87%8F%E7%BB%8F%E6%B5%8E%E5%AD%A6" \o "计量经济学)中。卡尔曼滤波也是[机器人](https://zh.wikipedia.org/wiki/%E6%A9%9F%E5%99%A8%E4%BA%BA" \o "机器人)运动规划及控制的重要主题之一，有时也包括在[轨迹最佳化](https://zh.wikipedia.org/w/index.php?title=%E8%BB%8C%E8%B7%A1%E6%9C%80%E4%BD%B3%E5%8C%96&action=edit&redlink=1)。卡尔曼滤波也用在中轴神经系统运动控制的建模中。因为从给与运动命令到收到感觉神经的回授之间有时间差，使用卡尔曼滤波有助于建立符合实际的系统，估计运动系统的目前状态，并且更新命令。

由鲁道夫·卡尔曼一开始发明的卡尔曼滤波只是其最基本的形式，目前该算法已经发展出一系列扩展或是广义的卡尔曼滤波，例如运作在非线性系统的[扩展卡尔曼滤波](https://zh.wikipedia.org/w/index.php?title=%E6%93%B4%E5%B1%95%E5%8D%A1%E7%88%BE%E6%9B%BC%E6%BF%BE%E6%B3%A2&action=edit&redlink=1)及无迹卡尔曼滤波（unscented Kalman filter）。底层的模型类似[隐马尔可夫模型](https://zh.wikipedia.org/wiki/%E9%9A%90%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB%E6%A8%A1%E5%9E%8B" \o "隐马尔可夫模型)，不过[潜在变量](https://zh.wikipedia.org/wiki/%E6%BD%9C%E5%9C%A8%E5%8F%98%E9%87%8F)的状态空间是连续的，而且所有潜在变量及可观测变数都是正态分布。

一种变种是不估计退化函数，仅将退化图像的信息矩阵第一行（或列）代入卡尔曼滤波预测方程作为初始值进行滤波，然后将第一次滤波图像的信息矩阵第一列（或行）代入卡尔曼滤波预测方程作为初始值进行第二次滤波获得复原图像[1]。仿真结果表明，交替卡尔曼滤波图像复原算法在去除退化噪声保证图像清晰的同时，还能快速完成图像复原，其计算时间降为维纳滤波复原法计算时间的2%。

另一种做法针对一种扩展算法发散的现象，提出了用自动加权法抑制扩展卡尔曼滤波发散的思路[2]。首先利用自动加权法判断出扩展卡尔曼滤波器是否发散；然后根据判断结果对扩展卡尔曼滤波器进行相应的处理。从而减少扩展卡尔曼滤波器发散对滤波估计的影响，大大地提高数据融合算法中扩展卡尔曼滤波器的精度，使算法能更好地完成雷达组网的数据融合功能。

1. **系统模型**

卡尔曼滤波建立在[线性代数](https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E4%BB%A3%E6%95%B0" \o "线性代数)和[隐马尔可夫模型](https://zh.wikipedia.org/wiki/%E9%9A%90%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB%E6%A8%A1%E5%9E%8B" \o "隐马尔可夫模型)（hidden Markov model）上。其基本动态系统可以用一个[马尔可夫链](https://zh.wikipedia.org/wiki/%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB%E9%93%BE" \o "马尔可夫链)表示，该马尔可夫链建立在一个被高斯[噪声](https://zh.wikipedia.org/wiki/%E5%99%AA%E5%A3%B0" \o "噪声)（即正态分布的噪声）干扰的[线性算子](https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E7%AE%97%E5%AD%90" \o "线性算子)上的。系统的[状态](https://zh.wikipedia.org/wiki/%E7%8A%B6%E6%80%81%E7%A9%BA%E9%97%B4" \o "状态空间)可以用一个元素为实数的[向量](https://zh.wikipedia.org/wiki/%E5%90%91%E9%87%8F" \o "向量)表示。随着[离散时间](https://zh.wikipedia.org/wiki/%E7%A6%BB%E6%95%A3%E6%97%B6%E9%97%B4%E4%BF%A1%E5%8F%B7" \o "离散时间信号)的每一个增加，这个[线性算子](https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E7%AE%97%E5%AD%90" \o "线性算子)就会作用在当前[状态](https://zh.wikipedia.org/wiki/%E7%8A%B6%E6%80%81" \o "状态)上，产生一个新的状态，并也会带入一些噪声，同时系统的一些已知的控制器的控制信息也会被加入。同时，另一个受噪声干扰的线性算子产生出这些隐含状态的可见输出。

为了从一系列有噪声的观察数据中用卡尔曼滤波器估计出被观察过程的内部状态，必须把这个过程在卡尔曼滤波的框架下建立模型。也就是说对于每一步k，定义[矩阵](https://zh.wikipedia.org/wiki/%E7%9F%A9%E9%98%B5" \o "矩阵)Fk, Hk, Qk, Rk，有时也需要定义Bk，如下。

卡尔曼滤波模型假设k时刻的真实状态是从(k-1)时刻的状态演化而来，符合下式：

Fk是作用在xk−1上的状态变换模型（/矩阵/向量）。

Bk是作用在控制器向量uk上的输入－控制模型。

wk是过程噪声，并假定其符合均值为零，[协方差矩阵](https://zh.wikipedia.org/wiki/%E5%8D%8F%E6%96%B9%E5%B7%AE%E7%9F%A9%E9%98%B5" \o "协方差矩阵)为Qk的[多元正态分布](https://zh.wikipedia.org/wiki/%E5%A4%9A%E5%85%83%E6%AD%A3%E6%80%81%E5%88%86%E5%B8%83" \o "多元正态分布)。

时刻k，对真实状态xk的一个测量zk满足下式：

其中Hk是观测模型，它把真实状态空间映射成观测空间，vk是观测噪声，其均值为零，协方差矩阵为Rk,且服从[正态分布](https://zh.wikipedia.org/wiki/%E6%AD%A3%E6%80%81%E5%88%86%E5%B8%83" \o "正态分布)。

初始状态以及每一时刻的噪声{x0, w1, ..., wk, v1 ... vk}都认为是互相[独立](https://zh.wikipedia.org/w/index.php?title=%E7%BB%9F%E8%AE%A1%E5%AD%A6%E7%8B%AC%E7%AB%8B&action=edit&redlink=1" \o "统计学独立（页面不存在）)的。

实际上，很多真实世界的动态系统都并不确切的符合这个模型；但是由于卡尔曼滤波器被设计在有噪声的情况下工作，一个近似的符合已经可以使这个滤波器非常有用了。

1. **改进之前的卡尔曼滤波算法**

卡尔曼滤波是一种[递归](https://zh.wikipedia.org/wiki/%E9%80%92%E5%BD%92" \o "递归)的估计，即只要获知上一时刻状态的估计值以及当前状态的观测值就可以计算出当前状态的估计值，因此不需要记录观测或者估计的历史信息。卡尔曼滤波器与大多数滤波器不同之处，在于它是一种纯粹的[时域](https://zh.wikipedia.org/wiki/%E6%99%82%E5%9F%9F" \o "时域)滤波器，它不需要像[低通滤波器](https://zh.wikipedia.org/wiki/%E4%BD%8E%E9%80%9A%E6%BB%A4%E6%B3%A2%E5%99%A8" \o "低通滤波器)等[频域](https://zh.wikipedia.org/wiki/%E9%A2%91%E5%9F%9F" \o "频域)滤波器那样，需要在频域设计再转换到时域实现。

卡尔曼滤波器的状态由以下两个变量表示：

在时刻k的状态的估计；

，后验估计误差协方差矩阵，度量估计值的精确程度。

卡尔曼滤波器的操作包括两个阶段：预测与更新。在预测阶段，滤波器使用上一状态的估计，做出对当前状态的估计。在更新阶段，滤波器利用对当前状态的观测值优化在预测阶段获得的预测值，以获得一个更精确的新估计值。

**1.预测**

（预测状态）

（预测估计协方差矩阵）

**2.更新**

首先要算出以下三个量：

（测量残差）

{\displaystyle {\textbf {S}}\_{k}={\textbf {H}}\_{k}{\textbf {P}}\_{k|k-1}{\textbf {H}}\_{k}^{T}+{\textbf {R}}\_{k}}（测量残差协方差）

{\displaystyle {\textbf {K}}\_{k}={\textbf {P}}\_{k|k-1}{\textbf {H}}\_{k}^{T}{\textbf {S}}\_{k}^{-1}}（最优卡尔曼增益）

然后用它们来更新滤波器变量**x**与**P**：

（更新的状态估计）

（更新的协方差估计）

1. **目标跟踪概述**

**1.视觉目标跟踪基本流程与框架**

视觉目标（单目标）跟踪任务就是在给定某视频序列初始帧的目标大小与位置的情况下，预测后续帧中该目标的大小与位置。

**运动模型：**生成候选样本的速度与质量直接决定了跟踪系统表现的优劣。常用的有两种方法：粒子滤波和滑动窗口。粒子滤波是一种序贯贝叶斯推断方法，通过递归的方式推断目标的隐含状态。而滑动窗口是一种穷举搜索方法，它列出目标附近的所有可能的样本作为候选样本。

**特征提取:** 鉴别性的特征表示是目标跟踪的关键之一。常用的特征被分为两种类型：手工设计的特征和深度特征。常用的手工设计的特征有灰度特征，方向梯度直方图，哈尔特征，尺度不变特征等。与人为设计的特征不同，深度特征是通过大量的训练样本学习出来的特征，它比手工设计的特征更具有鉴别性。因此，利用深度特征的跟踪方法通常很轻松就能获得一个不错的效果。

**观测模型:**大多数的跟踪方法主要集中在这一块的设计上。根据不同的思路，观测模型可分为两类：生成式模型和判别式模型. 生成式模型通常寻找与目标模板最相似的候选作为跟踪结果，这一过程可以视为模板匹配。常用的理论方法包括：子空间，稀疏表示，字典学习等。而判别式模型通过训练一个分类器去区分目标与背景，选择置信度最高的候选样本作为预测结果。判别式方法已经成为目标跟踪中的主流方法，因为有大量的机器学习方法可以利用。常用的理论方法包括：逻辑回归，岭回归，支持向量机，多示例学习，相关滤波等。

**模型更新:** 模型更新主要是更新观测模型，以适应目标表观的变化，防止跟踪过程发生漂移。模型更新没有一个统一的标准，通常认为目标的表观连续变化，所以常常会每一帧都更新一次模型。但也有人认为目标过去的表观对跟踪很重要，连续更新可能会丢失过去的表观信息，引入过多的噪音，因此利用长短期更新相结合的方式来解决这一问题。

**集成方法:**集成方法有利于提高模型的预测精度，也常常被视为一种提高跟踪准确率的有效手段。可以把集成方法笼统的划分为两类：在多个预测结果中选一个最好的，或是利用所有的预测加权平均。[3]

**2.目标跟踪的一种二值化的联通区域的目标跟踪方法**

使用上述公式计算仅在最优卡尔曼增益的时候有效。使用其他增益的话，公式要复杂一些，思考了基于跨层特征相连的办法，针对基于神经网络的模型，进行了思考，跟踪方法面临目标冗余的问题，在模块层面，针对无监督的跟踪方法，思考了目标跟踪的特征，需要针对物体是纯色的情况会更好跟踪，思考过基于模态关联的RGB-H的跟踪方法，传统多支路的方法，让网络获得两种模态图的互补信息

因为没有进行深度学习的神经网络，就只是对单色物体进行跟踪，然后用HSV方法进行阈值跟踪

可以利用联通区域标记来跟踪行人，根据连通区域的外形矩阵来跟踪一个行人的轨迹目标区域由于二值化与背景提取的偏差，目标中心往往不是目标跟踪的重心[6]

可以根据目标区域计算得到目标的重心，由于二值化得到的使得连通域变化的矩形不能很好地反应目标物体的真实情况，可以考虑进行二值化之前矩形投影，统计每行内块为白色的个数再进行投影，然后再统计每列中白色的个数

目标区域的重心轨迹分析，得到重心后，保留重心的行和列，第二帧在第一帧的附近进行搜索，如果搜索到有目标物的重心，认为两帧为同一目标物。

更新重心的行和列坐标，搜索的帧数可以自行设定，可以根据每十帧进行目标跟踪，得到的重心轨迹，来计算速度[4]。目标物的特征点选取，直接影响到目标物的跟踪，需要有代表性，特征性。

**3.视觉目标跟踪面临的挑战：**

视觉运动目标跟踪是一个极具挑战性的任务，因为对于运动目标而言，其运动的场景非常复杂并且经常发生变化，或是目标本身也会不断变化。那么如何在复杂场景中识别并跟踪不断变化的目标就成为一个挑战性的任务。

其中遮挡是目标跟踪中最常见的挑战因素之一，遮挡又分为部分遮挡和完全遮挡。解决部分遮挡通常有两种思路：（1）利用检测机制判断目标是否被遮挡，从而决定是否更新模板，保证模板对遮挡的鲁棒性。（2）把目标分成多个块，利用没有被遮挡的块进行有效的跟踪。对于目标被完全遮挡的情况，当前也并没有有效的方法能够完全解决。

形变也是目标跟踪中的一大难题，目标表观的不断变化，通常导致跟踪发生漂移。解决漂移问题常用的方法是更新目标的表观模型，使其适应表观的变化，那么模型更新方法则成为了关键。什么时候更新，更新的频率多大是模型更新需要关注的问题。

背景杂斑指得是要跟踪的目标周围有非常相似的目标对跟踪造成了干扰。解决这类问题常用的手段是利用目标的运动信息，预测运动的大致轨迹，防止跟踪器跟踪到相似的其他目标上，或是利用目标周围的大量样本框对分类器进行更新训练，提高分类器对背景与目标的辨别能力。

尺度变换是目标在运动过程中的由远及近或由近及远而产生的尺度大小变化的现象。预测目标框的大小也是目标跟踪中的一项挑战，如何又快又准确的预测出目标的尺度变化系数直接影响了跟踪的准确率。通常的做法有：在运动模型产生候选样本的时候，生成大量尺度大小不一的候选框，或是在多个不同尺度目标上进行目标跟踪，产生多个预测结果，选择其中最优的作为最后的预测目标。

1. **几种基于卡尔曼滤波的追踪目标应用**

由于卡尔曼滤波对线性系统出色的跟踪能力，其在实际的工程中得到了广泛应用，下面是几种基于卡尔曼滤波的追踪器应用

**1.基于卡尔曼滤波改进压缩感知算法的车辆跟踪算法**

**应用背景：**

目前土木工程领域利用卡尔曼滤波算法进行车辆目标跟踪的研究相对较少. 此外，传统的基于压缩感知技术的目标跟踪算法（Com⁃pressed sensing tracking algorithm，CT）受跟踪背景变化影响较大，存在跟踪点漂移、目标跟踪结果不稳定以及波动较大等问题。因此，本文采用卡尔曼滤波算法对压缩感知目标跟踪算法的结果进行修正，实现了较为精确的目标跟踪. 首先，通过传统压缩感知目标跟踪算法获得跟踪结果。

**算法概述：**

首先，通过传统压缩感知目标跟踪算法获得跟踪结果. 其次，基于上一帧跟踪轨迹，利用卡尔曼滤波预测本帧的跟踪轨迹，与本帧的实际跟踪轨迹相比较，利用卡尔曼系数对预测值与跟踪值进行修正，获得本帧图像目标跟踪结果. 最后，根据本帧目标跟踪的目标框坐标及尺度，利用调整后的跟踪目标范围在其周围对下一帧图像进行正负样本采样，从而实现目标跟踪轨迹的实时更新，并对采样器进行更新坐标的实时反馈[5].

**2.基于YOLOv3 与卡尔曼滤波的多目标跟踪算法**

**应用背景：**

目标跟踪一般分为单目标跟踪和多目标跟踪，对于单目标跟踪而言，一般会存在一个先验假设，因此即使只在初始位置框定范围，仍能得到一个看起来还好的跟踪结果。而通常应用于行人监测的多目标跟踪往往是一个多变量估计问题，不仅没有上述先验假设，而且还存在对象位置变化大，目标个数不固定的问题。因此多目标跟踪除了需要考虑物体形变、背景干扰等方面外，还需解决以下几个问题: (1)目标的自动初始化和自动终止; (2)目标的运动检测和相似度判别; (3)目标之间的交互和遮挡;(4)跟丢的目标再次出现时的再识别问题。针对上述问题中的一点或几点，研究者们提出相应的解决措施，主要可以分为两大类: 基于检测的数据关联算法和网络最小代价流算法。随着深度学习的快速发展，很多跟踪算法都以上述两大类方法作为基准，并加入深度学习算法[6]，因此基于YOLOv3 与卡尔曼滤波的多目标跟踪算法应运而生。

**算法概述：**

首先利用YOLOv3 网络检测出当前帧中的行人目标; 再采用卡尔曼滤波器对下一帧目标进行预测，减小长期跟踪所带来的误差; 然后通过改进匈牙利算法进行目标分配和数据关联，形成跟踪轨迹。

**3.基于卡尔曼滤波的目标识别跟踪与射击系统设计**

**应用背景：**

近年来，采用基于Siamese(孪生) 网络框架进行目标跟踪的孪生网络以其优良的性能在视觉跟踪领域受到了广泛的关注。SiamRPN是在Siamese 网络的基础上引入了一个区域候选网络，将每一帧的跟踪转换为一次局部检测任务，SiamRPN 网络强大的检测功能使得跟踪方法获得了较高的精度和速度。然而，当目标运动速度较快或者目标周围有干扰物的时候，跟丢目标的概率依然很高。为了提高模型抗干扰性和解决短时全遮挡问题，一种基于卡尔曼滤波的SiamRPN目标跟踪方法应运而生[7]。

**算法概述：**

本文提出了一种基于卡尔曼滤波的SiamRPN 目标跟踪方法，当视频输入时，SiamＲPN 和卡尔曼滤波模块同时进行跟踪，SiamRPN 模型分支首先利用卷积神经网络对模板帧和检测帧进行特征提取，然后经过RPN 网络回归得到跟踪目标的位置坐标和尺度。模型的另一分支利用卡尔曼滤波对检测帧的目标进行预测，最后通过置信度加权融合2 个子模型的跟踪结果，确定最优跟踪框。

**三、应用方案\应用系统\实验设置和结果展示（已有方法，改进的方法(不一定要更好)等，以及实验分析部分）（30分）**

**（一）算法流程**

我们尝试结合HSV阈值来观测目标，并结合卡尔曼滤波最优估计进行对目标的跟踪和对遮挡问题的预测处理，我们设计的（某一帧）算法流程如图3.1所示：

主要思路是在采样框内进行观测，并将计算得到的质心作为观测位置，传入卡尔曼滤波器进行估计。其中各个参数均为经验设置。

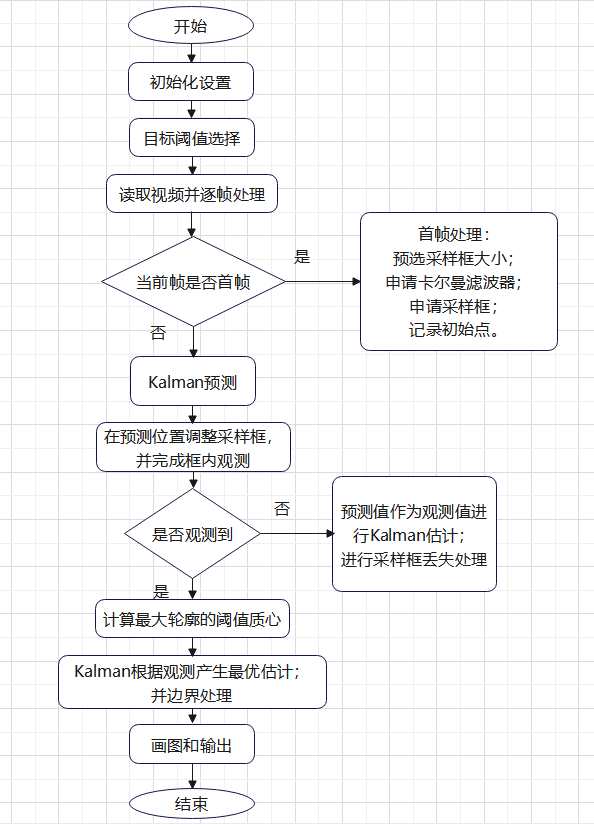


图 3.1.1 （某一帧）算法流程图

**（二）文件阐述**



图3.2 所有文件

1.main.py

主程序文件，运行即可得出算法结果。

2.SamBox.py

采样框类文件。

**作用：**为每个跟踪目标分配一个采样框。其根据首帧预选的ROI进行放大，放大倍率参考KCF论文设置为2.5倍。目的是为了在一定程度上减小背景干扰带来的影响。

使根据框中心坐标(cx, cy)在视野中生成合适的采样框。如图3.2所示：用：

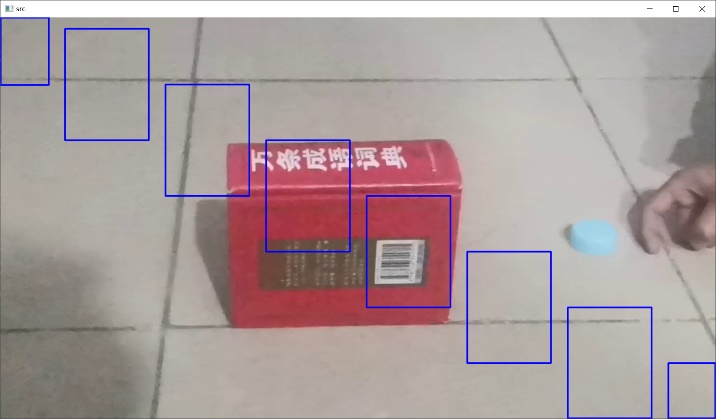


图3.2.1 采样框示意图

3.Kalman.py

卡尔曼滤波器类文件。

**作用**：为每个跟踪目标分配一个卡尔曼滤波器，主要包括预测和更新两个步骤。

**使用：**传入状态向量进行初始化，在视频处理过程中，传入观测向量，输出最优估计。

4.pickHSV.py

HSV阈值选择器类文件。

**作用：**为每个跟踪目标设置一个阈值选择器。通过手动调节阈值，选择完毕后退出。程序自动记录所选阈值。

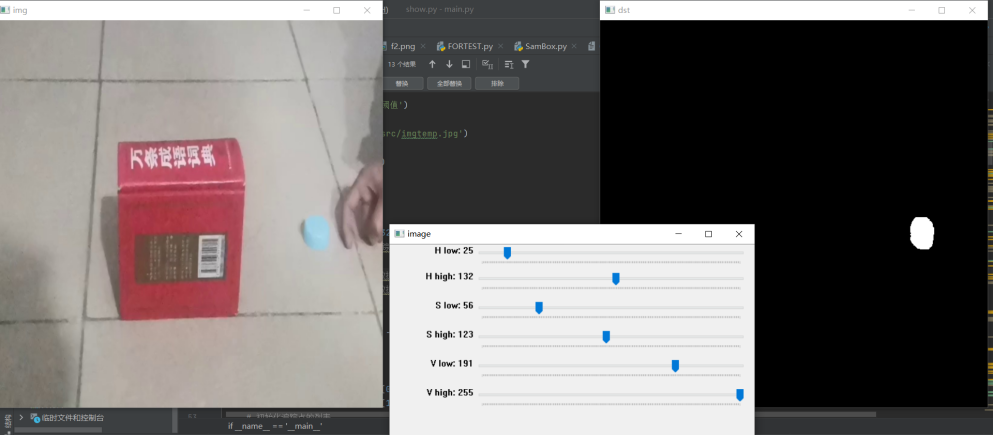


图3.2.2 阈值选择示意图

5.show.py

数据集标注文件。

**作用：**人工标注目标在视频中的真实位置。

6.mean\_square\_root.py

误差计算程序文件。

**作用：**计算最优估计txt和真实标注txt之间的误差，包括均方根误差。

7.RawVideos

原视频文件夹。

**内容：**包含已经标注的视频数据集。

8.ResultVideos

结果视频文件夹。

**内容：**包含带有轨迹点、追踪圈以及FPS等输出信息的视频。

9.TRpoints

原视频目标真实坐标文件夹。

**内容：**包含了对原视频逐帧进行人工标注真实坐标的txt文件。

10.ESpoints

最优估计坐标文件夹。

**内容：**包含了对原视频每一帧进行卡尔曼最优估计得到的轨迹坐标输出的txt文件。

11.Src

素材视频数据文件夹。

**内容：**包含未标注的数据视频，主要用于演示和测试。

**（三）四个问题**

目标跟踪领域有常见的几个问题，在我们的算法中，有相应的的对于这些问题的尝试解决方法的体现，作阐述如下：

**1.尺度问题**

尺度问题表示在目标跟踪过程中，跟踪目标的刚性形变以及非刚性形变，通俗而言就是目标会产生忽远忽近以及旋转等变化。

对于阈值跟踪而言，阈值即目标，目标有尺度变化时，阈值的二值化区域也随之变化，能灵活应对尺度问题，与此同时也产生新问题，即对于被部分遮挡的物体，仅能根据裸露部分计算质心坐标而导致与真实值误差偏大。

**2.遮挡问题**

遮挡问题有全遮挡和部分遮挡。

对于全遮挡问题，卡尔曼滤波器有着很好的解决效果，我们的策略是，可以通过卡尔曼滤波器进行预测，并将预测值作为观测值传入滤波器进行最优估计，这样做的理由是，在遮挡情况下我们没有观测值，只能完全信赖预测，但预测也存在误差，也不可完全信赖，不妨将预测值作为伪观测值，可以保证代码一致性和可移植性。

对于部分遮挡问题，目前没有良好的解决方法，但阈值锁定于剩余裸露区域，鉴于在视频追踪的场景下，目标所占比例不大，故利用剩余区域计算的质心坐标与真实值的误差也不会太大，在可以接纳的范围。

**3.目标脱离视野**

当目标消失时，此时可能是全遮挡或者目标脱离视野。我们的策略是，在卡尔曼预测的同时，进行全视频阈值检测，直到目标出现在视频画面中，这么做的好处是可以在一定程度时维持跟踪的连续性，问题是可能有将背景干扰带入计算的风险。

未来可以考虑在算法上进行改进，即判断目标丢失位置，若接近边界且具有向外的速度时，认为此丢失是目标丢失视野。

目标脱离视野具有完全遮挡问题的性质，可以当作遮挡问题处理。但出于目标位置具有随机性的考虑，区分目标丢失是被完全遮挡还是目标脱离视野的问题类别并无太大意义。

**4.背景干扰**

由于是阈值选择目标，故观测器对阈值完全依赖，很有可能将背景中的噪声代入计算，这样就会产生误差。

我们的策略是：第一，利用腐蚀和膨胀消除噪点，一般迭代次数设置为2，这个次数可以设计自适应算法，这是未来需考虑的工作；第二，设计采样框类，在一定区域而不是全图中进行观测，可以在一定程度上避免背景噪声影响。缺点是，采样框会产生新问题，即采样框丢失。

丢失的原因在于：在卡尔曼预测值上设置采样框，预测值可能不准确，也可能目标在被遮挡时，采样框根据卡尔曼预测而进行移动，导致逐渐脱离目标造成采样框丢失。采样框内无观测数据，即脱离了目标。

对于采样框丢失问题的解决方案较多，作分析如下：

第一，在首帧丢失时，由于目标的初始速度设置为0，采样框在第二帧的卡尔曼预测位置还是在原处，而此时目标可能已经移动，造成采样框丢失。解决方案是在第三帧设置采样框，这样可以记录下前两帧观测的速度信息，能让卡尔曼预测位置更可能框中目标，但这样做会忽略前两帧的信息，造成数据误差，若在其他帧发生丢失，则使用一次该方法就会造成两帧数据的误差，误差累积可想而知，故不采用。

第二，设置多个采样框。可在预测位置与原位置，或者四面八方均设置采样框，好处是可以增大采样框采中目标概率，缺点是不方便调度，也一定程度上增加了背景干扰信息，偏离了采样框设计初衷，不采用。

第三，放大处理，一旦发生丢失，即放大采样框，好处是可以增加采样框命中目标概率，但放大倍数又成新问题。

第四，在第三的想法上，直接将放大倍数设置最大，即进行全视野检测。一旦出现丢失，则在下一帧中进行全画面观测，直到目标出现。其合理性在于目标丢失时很难判断它会从何位置再现，具有随机性，甚至可能有切画面导致的尺度问题，故直接全图检测是最直接也是最无奈的策略。缺点是容易加入噪声干扰。

出于简单设计考虑，我们采用第四种方案。

**（四）数据集**

逐帧标注得真实坐标txt文件

**（五）实验结果**

**1.单目标无遮挡（视频1）：**

视频1路径：Target-Tracking/src/snk1.mp4

视频帧数：846

白色台球hsv阈值：[[24, 36, 149],[58, 102, 255]]

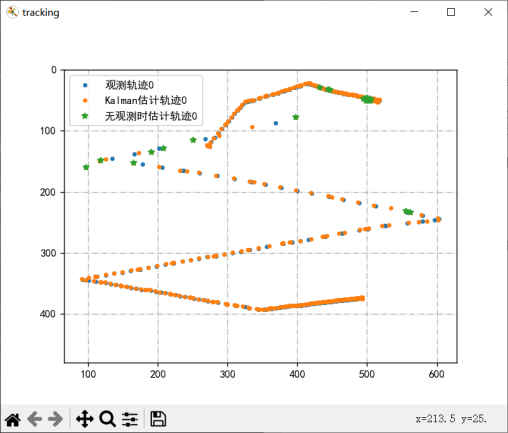


图3.6.1 视频1截图 图3.6.2 轨迹示意图

平均FPS：578

KCF结果：KCF存在丢失问题，由于白球在开始的高速移动，导致目标丢失，可见KCF对高速移动物体的适应性较差。

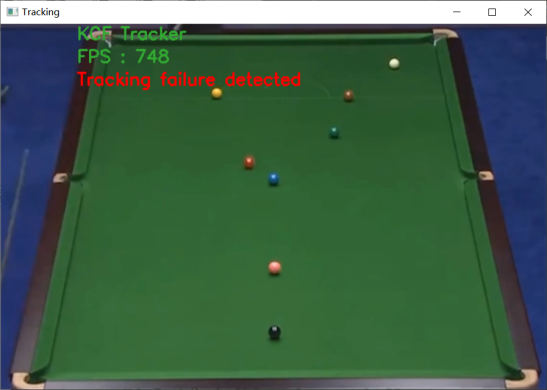


图3.6.3 KCF追踪白球示意

若将追踪目标换成红球，则可以很好地跟踪，平均FPS达到了760。

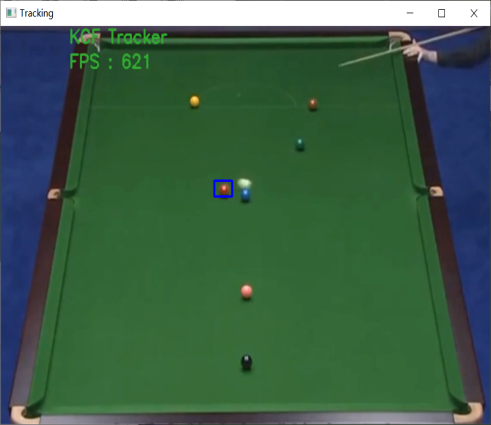


图3.6.4 KCF追踪红球示意

**2.单目标有遮挡（视频2、3）：**

视频2路径：Target-Tracking/RawVideos/1.mp4

视频帧数：69

目标阈值：[[76, 56, 191], [132, 123, 255]]

对于目标穿过遮挡物的实验结果如下：

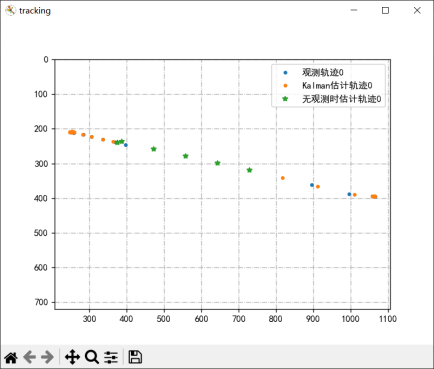


图3.6.5 视频2截图 图3.6.6 轨迹点图

平均FPS:241

均方根误差：8.14

KCF结果：同样出现了上述问题，

图3.6.7 KCF追踪示意

视频3路径：Target-Tracking/RawVideos/0.mp4

视频帧数：328

篮球阈值：[[ 0, 67, 108], [85, 255, 236]]

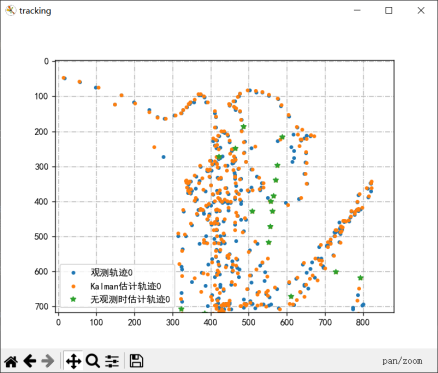


图3.6.8 视频3截图 图3.6.9 轨迹点图

平均FPS：230

均方根误差：33.96

KCF结果：KCF存在目标丢失问题。



图3.6.10 KCF追踪示意

推测原因：高速和慢速移动下的简单物体是有区别的，即物体在高速移动时会产生模糊幻影，而KCF在选择要追踪的目标时难以将二者辨别为同一物体从而导致误差。

**3.多目标遮挡（视频4）：**

视频4路径：Target-Tracking/src/12test.mp4

视频帧数：285

橘子和瓶盖的阈值分别为：[[0, 142, 72], [21, 255, 255]], [[80, 84, 163], [123, 146, 249]]

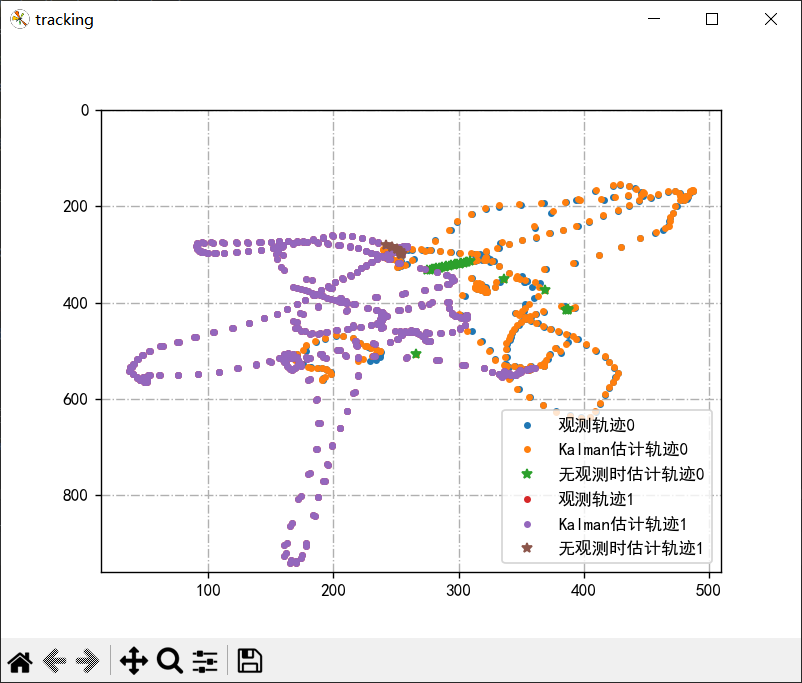


图3.6.11 视频3截图 图3.6.12 轨迹点图

平均FPS：255

KCF平均FPS：120

其过程图如下：

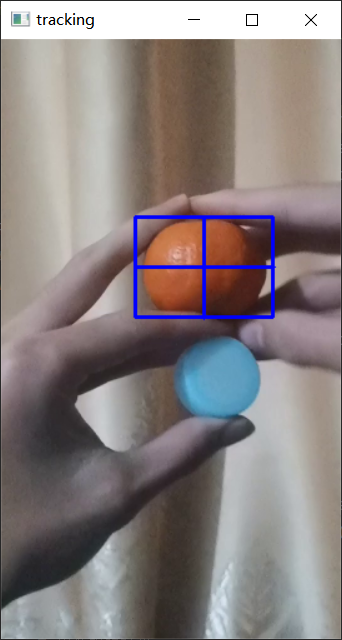
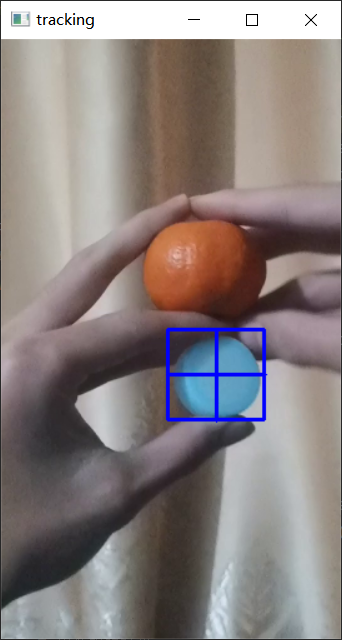


图3.6.13 KCF追踪目标选择

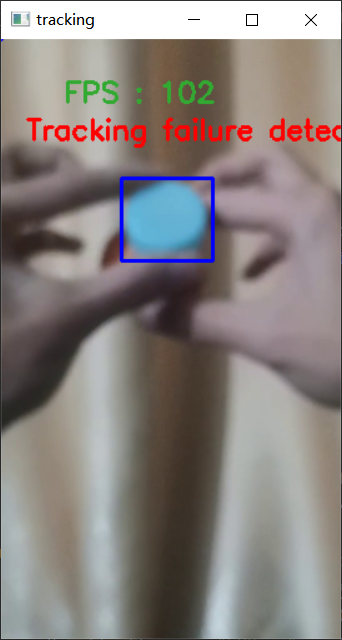
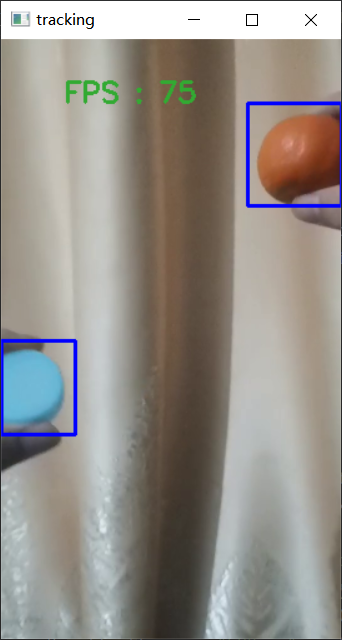


图3.6.14 KCF跟踪过程

对于多目标，KCF能进行很好的跟踪，但无法处理遮挡问题，后续可以考虑将KCF与卡尔曼滤波结合来对遮挡问题进行处理。

**（六）结论：**

可以得知，KCF难以适应尺度变化和快速变化的运动场景，而我们的算法在面对这些问题时效果较好。对我们算法的优缺点总结如下：

**1.算法优点**

在纯色目标、干扰较小和运动连续的视频场景下效果较好，卡尔曼滤波对于遮挡问题的处理效果也较好。

**2.算法缺点**

一些初始化操作对首帧的依赖性强，适用场景较为单一，目标种类受限，结果误差受阈值影响较大。

1. **思考、自己的见解、改进的基本想法、结论（20分）**

**（一）关于未来可考虑工作（改进）**

**1.自适应设计：**对于一些参数可以在迭代过程中自适应，比如卡尔曼协方差矩阵，采样框大小，腐蚀膨胀次数。

**2.部分遮挡问题：**对于部分遮挡问题产生的误差可以设法尽量缩小。

**3.自动提取阈值：**手动选取阈值费时费力，考虑一种自动阈值选取的方法，或者舍去阈值，简化算法，仅依靠轮廓进行检测。

**4.增加特征：**增加目标特征，比如轮廓形状，或者结合目标检测算法，比如yolo等，在目标检测方面完善算法。

**5.与KCF结合：**可以对KCF引入卡尔曼滤波器进行相关问题处理

**6.多目标改进**：多目标有目标丢失超过一定时长时可以考虑丢弃目标。

**7.卡尔曼：**可以在卡尔曼模型上进行升级。

**8.提高对帧率的补充：**比如说补帧。

**9.增大对速度的阈值的范围：**比如说增加最大识别速度，降低最小识别速度阈值。

**10.增加环境的影像的参数**

**11.考虑加入神经网络，**进行数据处理

**（二）改进思路**

我们所完成的卡尔曼滤波器是基于颜色阈值确定目标之后进行追踪，但这种方法存在背景噪声影响大、容易失去目标等缺点。在后续改进过程中，我们将这种方式与其他建模方式相结合，来保证目标识别的连续性和鲁棒性。本文中提到的Yolo3与卡尔曼滤波算法结合就是使用深度学习建模，从而提高了算法的稳定性。

|  |  |  |
| --- | --- | --- |
| **建模方法** | **优点** | **缺点** |
| 核跟踪 | 实现简单、速度快 | 复杂条件下效果不好 |
| 子空间 | 降低了追踪时间 | 不稳定 |
| 稀疏表示 | 有效解决目标遮挡 | 计算代价高 |
| Boosting算法等 | 判别学习能力强 | 信息冗余 |
| 随机学习 | 并行、速度快 | 跟踪性能不稳定 |
| 深度学习 | 特征建模能力优秀 | 训练集少、计算量大 |

**图4.2.1 几种常见的建模方法及优缺点**

**五、参考文献（10分）**

1. 张琦,张慧,潘健,刘松林.一种新的卡尔曼滤波图像复原算法[J].湖北工业大学学报,2022,37(05):23-27.
2. 尹建军,陈涛,岳海燕.基于自动加权法的扩展卡尔曼滤波数据融合算法研究[J].无线互联科技,2022,19(18):139-141.
3. 张丰冯平 视频目标跟踪综述 -《计算机时代》2022年1期
4. 孟琭杨旭-《自动化学报》目标跟踪算法综述 2019年7期
5. 周云, 胡锦楠, 赵瑜,等. 基于卡尔曼滤波改进压缩感知算法的车辆目标跟踪.
6. 任珈民, 宫宁生, 韩镇阳. 基于YOLOv3与卡尔曼滤波的多目标跟踪算法[J]. 计算机应用与软件, 2020, 37(5):8.
7. 张子龙,王永雄.基于卡尔曼滤波的SiamRPN目标跟踪方法[J].智能计算机与应用,2020,10(03):44-50.
8. [基于关联分析的目标跟踪方法研究](https://kns.cnki.net/kns8/Detail?sfield=fn&QueryID=0&CurRec=1&FileName=1022785047.nh&DbName=CMFDTEMP&DbCode=CMFD" \t "https://kns.cnki.net/kns8/defaultresult/_blank),陈高伟，南京邮电大学，2022-12-05
9. 曾巧玲文贡坚 运动目标跟踪综述 《重庆理工大学学报（自然科学版）》2016年7期