3.1 基于卡尔曼滤波的多目标并行运动预测技术

卡尔曼滤波，也被称为线性二次估计，是一种使用一系列随时间观察到的测量结果，包括统计噪声和其他不准确因素，估计每个时间段的变量的联合概率分布，产生未知变量的估计值的算法，这些估计值往往比仅基于单一测量的变量更准确。这种数字滤波器有时被称为Stratonovich-Kalman-Bucy滤波器，该算法通过一个两阶段的过程工作，主要过程如图6所示。

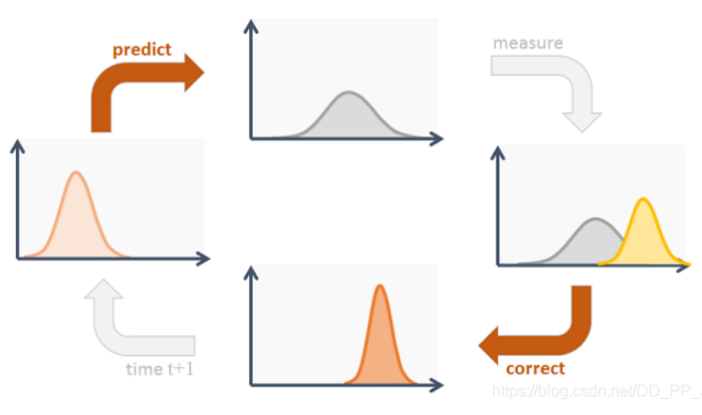


图 6 卡尔曼滤波器计算流程

在目标跟踪任务中，目标的状态变量表示为 ，其中 表示目标框的中心坐标，表示目标框的高宽比,表示目标框的高，在卡尔曼滤波算法中也表示为 mean；分别表示对应变量的速度，也称之为 covariance。

2.1 预测

预测就是根据目标在 t-1 时刻的状态来预测其在 t 时刻的状态。预测主要分为两部分：

(1)

在公式 1.1 中， 为 track 在 t-1 时刻的状态值，称为状态转移矩阵，该公式预测 t 时刻的状态值 。

在公式 1.2 中，为 track 在 t-1 时刻的协方差，为系统的噪声矩阵，代表整个系统的可靠程度，一般初始化为很小的值，该公式预测 t 时刻的 。

**2.2 更新**

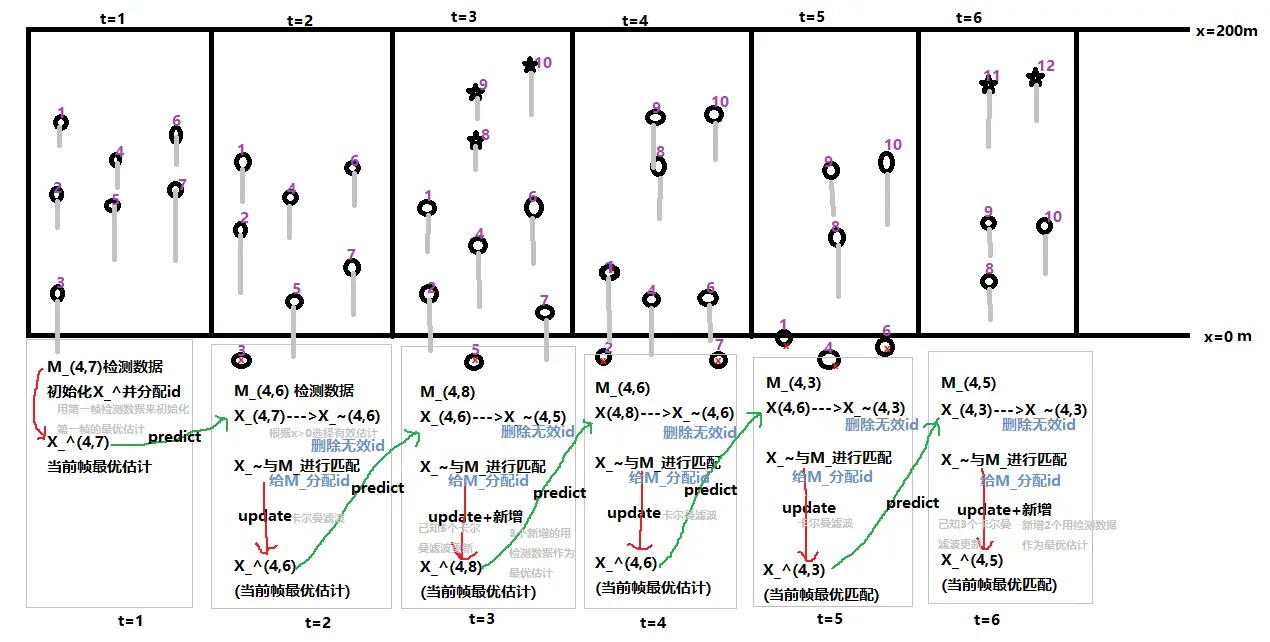
更新是基于 t 时刻的检测结果（测量值）和根据跟踪轨迹预测目标在 t 时刻的状态（预测值），得到一个在 t 时刻更精确的结果（状态）。

(2)

在公式2.1中 为 检测到的的状态向量，该状态向量不包含速度变化值，即，称为测量矩阵，它将跟踪到的的均值向量映射到检测空间，该公式计算实际检测到的状态向量和跟踪到的状态向量的均值误差 ；在当前任务中，测量矩阵 H 取单位矩阵；公式 2.2中，为检测器的噪声矩阵，它是一个 4x4 的对角矩阵，对角线上的值分别为中心点两个坐标以及宽高的噪声，以任意值初始化，一般设置宽高的噪声大于中心点的噪声。该公式先将协方差矩阵 映射到检测空间，然后再加上噪声矩阵R；公式2.3中，计算 卡尔曼增益，卡尔曼增益用于估计误差的重要程度；公式2.6和公式2.7中，计算更新后的均值向量和协方差矩阵。

具体来说：

如图7所示，多目标跟踪中，有效跟踪范围为x=0到x=200之间，**红线**表示卡尔曼滤波的**初始化**及**更新**阶段，**绿线**表示**预测**阶段。



**当t=1时，**

1. **M\_(4,7)表示观测数据。**这里检测到7个目标。（这里用矩阵表示多个目标，矩阵的每一列表示一个目标，每个目标有4维数据，分别是观测到的物体该时刻的位置和速度 [px,py,vx,vy] ）
2. **X\_^ 表示当前时刻的最优状态估计。**（因为这是第一组数据，因此，用当前时刻的观测数据M\_对X\_^进行初始化）
3. 同时，对t=1时刻的目标进行**依次编号**，作为**初始化**结果。

**当t=2时，**

1. **M\_(4,6)表示观测数据。**这里检测到6个目标。（上一时刻的3号目标在当前时刻超出检测范围，因此不计入观测数据）
2. 通过上一时刻的最优估计X\_^，可以根据指定条件筛选**符合条件的目标**，得到当前时刻的**有效估计**。（X\_(4,7)为上一时刻的预测，假如判断符合条件的目标为(px>0)，则3号目标不符，去除编号3。因此，有效目标只有6个，因此**有效估计X\_~为(4,6)**）
3. 将有效估计X\_~(4,6)与当前时刻的观测数据M\_(4,6)进行**匹配**，得到每个目标在当前时刻的编号，这里可以用**匈牙利算法**进行目标之间的匹配。
4. 利用卡尔曼滤波算法进行**当前时刻的最优状态估计X\_^(4,6)**。

**当t=3时，**

1. **M\_(4,8)表示观测数据。**这里检测到8个目标，其中5个是已知的，3个是**新增目标**。（上一时刻的5号目标在当前时刻超出检测范围，因此不计入观测数据）
2. 通过上一时刻的最优估计X\_^，可以根据指定条件筛选**符合条件的目标**，得到当前时刻的**有效估计**。（X\_(4,6)为上一时刻的预测，假如判断符合条件的目标为(px>0)，则5号目标不符，去除编号5。因此，有效目标只有5个，因此**有效估计X\_~为(4,5)**）
3. 将有效估计X\_~(4,5)与当前时刻的观测数据M\_(4,8)进行**匹配**，得到每个目标在当前时刻的编号。
4. 对于**匹配**数据，利用卡尔曼滤波算法更新得到**当前时刻的最优状态估计X\_^(4,5)**；**未匹配**数据有3个，即新增的3个目标。对于这3个目标，用观测数据对其进行**初始化**，并**编号**。
5. 将两部分数据相结合，得到**当前时刻的最优状态估计X\_^(4,8)**。

**当t=4时，**

1. **M\_(4,6)表示观测数据。**这里检测到6个目标。（上一时刻的2号和7号目标在当前时刻超出检测范围，因此不计入观测数据）
2. 通过上一时刻的最优估计X\_^，可以根据指定条件筛选**符合条件的目标**，得到当前时刻的**有效估计**。（X\_(4,8)为上一时刻的预测，假如判断符合条件的目标为(px>0)，则2号和7号目标不符，去除编号2和编号7。因此，有效目标只有6个，因此**有效估计X\_~为(4,6)**）
3. 将有效估计X\_~(4,6)与当前时刻的观测数据M\_(4,6)进行**匹配**，得到每个目标在当前时刻的编号。
4. 利用卡尔曼滤波算法进行**当前时刻的最优状态估计X\_^(4,6)**。

**当t=5时，**

1. **M\_(4,3)表示观测数据。**这里检测到3个目标。（上一时刻的1号、4号和6号目标在当前时刻超出检测范围，因此不计入观测数据）
2. 通过上一时刻的最优估计X\_^，可以根据指定条件筛选**符合条件的目标**，得到当前时刻的**有效估计**。（X\_(4,6)为上一时刻的预测，假如判断符合条件的目标为(px>0)，则1号、4号和6号目标不符，去除编号1、4、6。因此，有效目标只有3个，因此**有效估计X\_~为(4,3)**）
3. 将有效估计X\_~(4,3)与当前时刻的观测数据M\_(4,3)进行**匹配**，得到每个目标在当前时刻的编号。
4. 利用卡尔曼滤波算法进行**当前时刻的最优状态估计X\_^(4,3)**。

**当t=6时，**

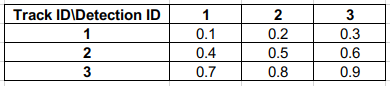
1. **M\_(4,5)表示观测数据。**这里检测到5个目标，其中3个是已知的，2个是**新增目标**。
2. 通过上一时刻的最优估计X\_^，可以根据指定条件筛选**符合条件的目标**，得到当前时刻的**有效估计**。（这里上一时刻的3个目标都为有效目标）
3. 将有效估计X\_~(4,3)与当前时刻的观测数据M\_(4,5)进行**匹配**，得到每个目标在当前时刻的编号。
4. 对于**匹配**数据，利用卡尔曼滤波算法更新得到**当前时刻的最优状态估计X\_^(4,3)**；**未匹配**数据有2个，即新增的2个目标。对于这2个目标，用观测数据对其进行**初始化**，并**编号**。
5. 将两部分数据相结合，得到**当前时刻的最优状态估计X\_^(4,5)**。

由此，多目标跟踪算法中，卡尔曼滤波的主要作用就是当前的一系列运动变量去预测下一时刻的运动变量，但是第一次的检测结果用来初始化卡尔曼滤波的运动变量。

3.2 基于匈牙利算法的多目标并行关联匹配技术

匈牙利算法通常用来求二分图的**最大匹配数**和**最小点覆盖数**。二分图就是分为两组的数据，前后两组直接可以相连，但是同组直接不能相连。就好比目标跟踪中，视频前后帧中的检测框，可以看做是两组数据，这两组数据之间存在匹配关系（同一个目标，在前后帧中的检测框为一对），而同一帧中的目标框经过 NMS 之后，我们认为他们都是不同的目标，不存在匹配关系。显然 MOT 中前后帧中目标框的匹配问题就是一个求二分图的最大匹配数的问题（尽量匹配所有目标）。

匈牙利算法需要输入一个代价矩阵（或者利益矩阵），那么在目标跟踪问题中，代价矩阵可以是从前后帧中提取的 ReID 特征的距离，也可以是 IoU 的距离，显然距离越小，匹配的就越好，所以整个问题就变成找到一组匹配，使得总的代价最小。



比如上面的表格中的列（Track ID）是之前跟踪到的目标，行（Detection ID）是当前检测到的目标，[1, 1] 表示的是 Track ID 1 和 Detection ID 1 的 IoU 距离。利用匈牙利匹配算法则是从候选目标中获取最为匹配的当前帧下的检测目标。

多目标跟踪算法中，匈牙利算法的作用简单来讲即为解决分配问题，把图像帧中的检测框和卡尔曼预测的框做分配，让卡尔曼预测的框找到和自己最匹配的检测框，达到追踪的效果。

2.2

多人2D姿态估计，是理解图像中人体的一种任务。输入一张图片，其目标是检测每一个人和定位到他们对应的关节点。推测图片中多人的姿态是有挑战的，由于图像中存在人体数量的变化，尺度变化，身体部位遮挡，人体非刚性等因素。

当前对于姿态估计的方法主要分为两大类：自下而上和自上而下的。自上而下或者双阶段的方法是目前最领先的方案。他们首先采用一个强大的人体检测器，然后针对每个人体做单人的姿态估计。自上而下的方法的复杂度随着图像中人体数量而线性增加。多数实时应用需要限制运行时间，不倾向于采用自上而下的方法，由于其复杂度较高。反而，自下而上的方法提供了恒定的运行时间，因为他们在单次检测中依赖热力图检测所有关键点，然后进行复杂的后处理将他们分组成个体。后处理可能涉及到如像素级别的NMS，线积分，细化，分组等步骤。坐标调整和细化减少了下采样热力图的量化误差，其中NMS用于寻找热力图中的局部最大值。即使经过后处理，热力图也许不够清晰，去区分同一类型的两个靠得很近的关键点。同样，自下而上的方法也不能端到端训练，因此后处理步骤是不可差分的，因此在卷积网络之外发生。他们在方法上存在很大差异，从线性规划到各种启发式算法。很难用CNN加速器去加速他们，因此速度也很慢。有单次推理的方法，虽然避免了分组任务，但不能与自下而上的方法相提并论。他们依赖额外的后处理来提升性能。

本项目算法拟解决不使用热力图进行姿态估计，而与目标检测保持一致，因为目标检测中的挑战与姿态估计类似，例如尺度多样性，遮挡，人体非刚性等。因此，如果一个人体检测网络能够处理这些问题，它也能处理姿态估计。调研发现，最近的目标检测框架尝试解决尺度变化的问题，通过在多个尺度上做预测。这里，研究人员拟采用相同的策略去针对每个检测去预测多尺度的人体姿态。

对于当前应用场景下的人体姿态估计，它归结为一个三种类别的人体姿态检测问题，每个人有6个相关的关键点，每个关键点输出一个位置和可信度：。因此，与一个锚点关联的6个关键点总共有18个元素。因此，对于每个锚点，关键点头预测18个元素。在神经网络的训练过程中，关键点置信度是基于该关键点的可见性标志进行训练的。如果一个关键点是可见的或者被遮挡的，那么地面真实置信度将设置为1在视场之外，置信度则设置为零。在推理过程中，我们保留了可信度大于0.5的关键点。预测的关键点置信度不用于评估。然而，由于网络预测了每个检测的所有6个关键点，我们需要过滤掉视场之外的关键点。否则，就会有摆动的关键点，导致输出是骨骼变形的。

为了使网络更好的学习到人体拓扑关系结构，感知人员姿态规律，提高人体关键点感知准确率，研究人员重新设计了人员姿态损失函数，核心思想是利用anchor中心，不考虑框的大小。

OKS是评估关键点的最流行的策略。传统上讲，基于热图的自下而上的方法采用L1损失去检测关键点。但是，L1损失可能不合适去获取优化的OKS。因为L1损失是最朴素的，并不考虑一个物体的尺度或者关键点的类型。由于热图是概率图，不可能去使用OKS作为纯热图方法的损失函数。OKS能当做损失函数仅当我们回归关键点位置的情况下。有人提出尺度归一化的L1损失用于关键点回归，这是迈向OKS损失的一步。

因此，我们直接相对于锚框中心回归关键点，所以我们可以优化评估度量本身，而不是一种代理的损失函数。我们将IOU损失的概念从边界框扩展到关键点。在关键点中，OKS被当做IoU。因此OKS损失本质上是尺度不变的，针对特定的关键点倾斜重要性。也就说，一个人头部的关键点如耳朵、鼻子和眼睛等，会比身体如肩膀，膝盖和臀部等在像素级别上受到更多的错误惩罚。这个权重因子是COCO作者从冗余的标注验证集中根据经验选择的。不像原始的IoU损失，会在非重叠情况下出现梯度消失，OKS损失不会停滞。因此，OKS损失更像dIoU损失。

对应于每个边界框，我们存储整个姿态信息。针对每一个单独的关键点计算OKS，并累加到最终的OKS损失或者关键点IoU，即：

这里所采用的OKS(Object Keypoint Similarity)是用来表示关键点相似度的指标，它的值在(0,1)之间分布，当OKS的值接近1时预测结果越好。其中，每个预测关键点与标注的真实关键点间的欧几里得距离为d，v用于表示关键点是否可视，物体尺度为s，控制衰减常数为k；置信度损失函数为：

最终，人员姿态损失函数为：

为0.1，取0.5