

指导教师： 杨 涛

提交时间： 2016/3/14

# CVPR2015 Paper Translation

No: 01

姓名： 刘鹏飞

学号： 2013302555

班号： 10011304



## 鲁棒性视觉追踪的多假设轨迹分析

Dae-Youn Lee†, Jae-Young Sim‡, and Chang-Su Kim†

†School of Electrical Engineering, Korea University, Seoul, Korea

‡School of Electrical and Computer Engineering,

Ulsan National Institute of Science and Technology, Ulsan, Korea

daeyounlee@mcl.korea.ac.kr, jysim@unist.ac.kr, [changasukim@korea.ac.kr](mailto:changasukim@korea.ac.kr)

### 摘要

本文提出了强大视觉追踪的多重假设轨迹分析(MBA)的概念。我们采用了多组不同线成分的追踪器,它们独自具有纹理、颜色、光照不变性的特点。每个组件追踪器跟踪目标对象的痕迹向前,经过一定时间间隔后退回。通过分析前向和后向追踪轨迹,我们测量了组件跟踪器的鲁棒性。为此,我们从向前和向后的轨迹中提取了它们的几何相似性,循环权重和外观相似性。我们选择最佳的组件跟踪器,以产生最大的鲁棒性得分,并使用其正向轨迹作为最终的跟踪结果。实验结果表明 MTA 追踪器提高了追踪的鲁棒性和精度,比最近的基准问题测试资料集中记录的一个跟踪器做得还好。

### 1、 介绍

几十年来,人们已经提出了许多视觉跟踪算法[30], [29], [20]。由于各种原因,大多数跟踪算法经历着漂移问题,包括不能明显识别的特征描述符,闭塞和突发的照明变化。为了能够更强大地追踪,最近的追踪算

法都在试图克服这种干扰。

根据某种特征类型,依据输入的视频序列可能无法从背景中区分目标对象,因此,自适应地结合采用多个不同功能的追踪器,可以实现强大的鲁棒性追踪[19], [24], [16], [17]。更具体地说,对多个追踪器所追踪的位置进行比较和处理,可以产生一个整体的估计位置。然后,所有的追踪者在该位置使用估计信息更新自己的坐标。然而,如果多个追踪算法因为任何中断,在某一帧中失去了目标对象的位置,追踪误差可能会传播到未来的帧。

最近,人们提出了一种可以抑制误差传播的追踪系统[23], [21], [31], 它拥有内存设备,可以改进追踪器过去的轨迹或外观模型。在[23]长期追踪中,[21], 追踪器将一个目标最可能的位置存储在每一帧中,然后,同时考虑目标物体的每个位置的可靠性,以及在连续时间的帧,目标物体位置关系的可靠性,动态规划估计其运动轨迹。此外,[31], 对过去的帧的一些外观模型记录和处理,了、可以得到一个适当的外观模

型，并减少追踪误差。然而，这些采用内存的追踪使用固定的特征描述，并不能在一些序列中有效地将目标对象从背景部分区分开。因此，它们根据功能选择仍然会得到错误的追踪结果。

在本文中，我们提出了一种新的多假设追踪算法，它结合了“追踪使用多种追踪器”与“带有记忆的追踪”两种概念。我们采用了具有不同功能的三个向前跟踪器，它们分别基于纹理信息、颜色信息和光照不变的信息三个方面。从框架式到框架式，每个向前的追踪器都独立地追踪一个目标物体。然后，在框架式中，所有的反向的追踪器都被初始化在相应的前向跟踪的估计位置，然后在一个反转时间计算一个向后的轨迹。为了选择三个追踪器中最好的跟踪结果，我们计算它们的鲁棒性值。为此，我们从每对向前和向后的轨迹中提取了它们的几何相似性，循环权重和外观相似性。在选择了最佳的前进轨迹之后，所有前向追踪器的外观模型都被恢复到以前框架式中的状态，并被沿着选定轨迹的包围盒所更新。当连续帧中所有前进轨迹的几何和外观的相似性都非常小时，前向追踪器不再更新，搜索的范围将会在下一帧中扩大。这项工作的主要贡献如下。

- 采用新奇的多重假设轨迹分析从多个追踪器中提取最佳轨迹。
- 一组基于几何相似，循环量与外观相似的向前和向后轨迹的鲁棒

性值设计。

- 应用几何相似性与外观相似性的模式分析方法检测和处理轨迹追踪故障。

本文的其余部分组织如下：第2节简要回顾了相关工作原理。第3节提出了一种基于追踪器受影响时的算法。第4节描述了这种算法，第5节讨论了实验结果。第6节得出了结论。

## 2、 相关研究工作

检测追踪：为了长时间通过检测一个目标对象来跟踪它，人们使用了一个被训练了 10,000 张左右有车辆和无车辆的离线图片的支持向量机 (SVM) 分类器 [1]。由于在一般情况下，目标对象的外观会随着时间的推移而变化，人们使用正负样本在线更新跟踪目标位置周围的分类器 [2]。受影响的追踪器 [10] 采取结构化的 SVM 分类器来集成学习和协调追踪、避免样品标签产生歧义。同时，为了防止由于分类器更新而产生的漂移问题，遮挡检测通过长期分析在一个边界框内轨迹的遮挡点 [12]。

后向追踪：在 [27] 中，根据几何相似，时间可逆性约束可用于提高 Kanade-Lucas-Tomasi (KLT) 特征点跟踪。在向前和向后追踪之后，几何相似性测量特征点和它的估计位置之间的距离。在理想的情况下，距离应该是零。TLD 追踪器 [15] 采用几个 KLT



特征点追踪，它从一个包围盒内矩形网格上的点开始，估计它们的连续帧之间的运动矢量。一半基于几何相似的运动矢量被丢弃。然后，剩余的运动矢量的中位数变成了边界框的估计位移矢量。几何相似性也被用来检测在 [ 25 ] 中的追踪故障。

该算法还进行反向追踪，以提高追踪的鲁棒性。然而，鉴于常规算法 [27]、[15]、[25] 在两个连续的帧之间仅仅根据几何相似检测追踪故障，该算法通过采用在一段时间间隔内一对向前和向后的运动轨迹之间的几何相似、圆度和外观相似，量化了前进轨迹的可靠性。

多追踪器：协同追踪算法使用不同的特征类型训练多个支持向量机 SVM 分类器，并结合它们的追踪结果，来实现鲁棒追踪。在 [ 16 ]，使用多种观察和运动模型设计多个追踪器，然后用交互式马尔可夫链蒙特卡罗框架整合到一个整体追踪器中。另外，在 [17]，多追踪器被自适应地从追踪空间采样并组合。统一的算法 [ 9 ] 利用了那些通过测量追踪两个连续帧之间的一致性和两两之间的相关性不同的追踪器的个体之间的关系。

记忆追踪：在 [ 23 ]，[ 21 ]，目标对象的候选位置被记录在最近的帧中。这些位置成为节点，它们被连续帧之间的边缘连接着。然后，通过动态规划提取最优轨迹。MEEM 算法 [ 31 ] 采用多个基本追踪器并记下它

们以前的状态，从而恢复追踪过程中当一个跟踪器被误报的更新。此外，融合算法 [ 4 ] 提供了一个对象的精致轨迹，在动态规划的基础上，结合了在最近基准测试 [ 28 ] 的传统算法。

### 3、 追踪干扰

本节简要回顾追踪干扰 [ 10 ]。用  $x$  表示目标对象的边界框的位置，用  $d$  表示边界框从以前帧到当前帧的位移矢量。干扰使用判别函数公式  $f(x, d) = w' \phi(x, d)$ ， $\phi(x, d)$  是  $x$  和  $d$  共同特征图， $w$  是一个超平面的法向量。判别函数简化为

$$f(x, d) = \sum_{i,j} \beta_{i,j} k(\phi(x^{(i)}, d^{(j)}), \phi(x, d)) \quad (1)$$

$(x^{(i)}, d^{(j)})$  是支持向量， $k(\cdot)$  是一个联合核函数，将线性分类器转换为非线性分类器。另外， $\beta_{i,j} > 0$  表示一个正支持向量， $\beta_{i,j} < 0$  表示一个负支持向量。

在第  $t$  帧时边界框的位置  $x_t$  的公式为  $x_t = x_{t-1} + d_t$ 。估计位移矢量公式为：

$$\hat{d}_t = \arg \max_{d_t} f(x_{t-1}, d_t) \quad (2)$$

这个式子最大化了判别函数。在估计了  $x_t$  之后，系统标记的训练样本从第  $t$  帧中产生了，它被用于更新基于在线支持向量机技术 [ 5 ]，[ 6 ] 的判别函数。

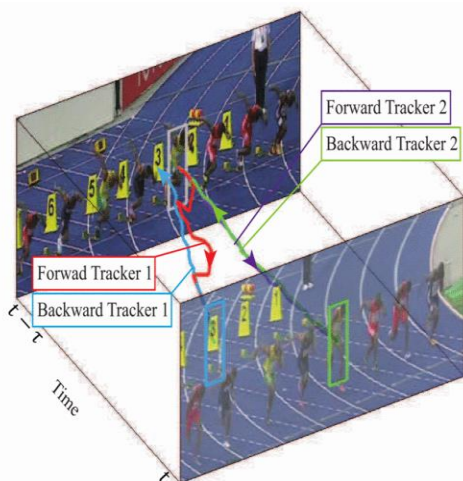


图 1. 前进后退轨迹分析：紫色跟踪器是成功的，而红色则是不成功的。

#### 4、 提出的 MTA 跟踪器

当目标对象改变它的外观或是被其他对象遮挡时，跟踪就可能会失败。例如，假设一个目标对象逐渐被非目标对象遮挡。那么，它的外观模型也会慢慢向非目标对象的特点变换，最终导致跟踪失败。然而，它是很难区分这种物体外观逐渐变化的真正外观模型。为了克服这个困难，我们采用了一个向后的跟踪器，它可以在相反的顺序检测到一个特定的对象。我们最初把后向跟踪放在前向跟踪器的检测位置，获得了后向轨迹。通过比较后向轨迹与前向的，我们可以大致知道正向跟踪是否成功了。此外，我们采用了多个前向跟踪器，提供多种轨迹的假设。在前后向分析的基础上，我们选择最佳的前向轨迹，以提高跟踪的精度和鲁棒性。因此，我们将这种提出的算法称为多

假设跟踪轨迹分析（MTA）。

图 1 说明了从帧  $t-\tau$  到帧  $t$  的两个轨迹假设。在这个例子中，我们跟踪的目标运动物在帧  $t-\tau$  时处于灰度边界框范围内。红色和紫色的曲线描绘了由两个不同的前向跟踪器跟踪得到的前向轨迹，而蓝色和绿色曲线分别是对应的反向跟踪器反向的轨迹。请注意，在不断地用不准确的样品更新相关的外观模型情况下，红色跟踪器跟踪目标失败。在  $t$  时，蓝色跟踪器被初始化在蓝色边界框内。然后它在时间顺序相反时遵循非目标对象向后，并在反向跟踪过程中的  $t-\tau$  时，与目标完全不同。相比之下，绿色跟踪器在  $t-\tau$  时提供了与原来的目标对象匹配成功的结果。因此，我们选择紫色的向前跟踪器作为有效，并丢弃了红色跟踪器的结果，以实现鲁棒的目标跟踪。

##### 4.1. 多组件跟踪器

我们采用三分量跟踪器  $\Gamma_1$ ， $\Gamma_2$  和  $\Gamma_3$ ，这些是基于干扰的。这些跟踪器使用不同的特征描述，被用于独立地确定目标轨迹。组件跟踪器利用有以下功能：

$\Gamma_1$ ：第一个跟踪器采用了一个

Haar-like 特征来代表目标对象的纹理信息。Haar-like 特征包括了六个不同的类型，在两尺度涵盖了边界框 [ 10 ] 的  $4 \times 4$  块。

因此，特征向量的维数为 192。矢量中的每一个元素被归一化到  $[-1, 1]$ 。

$\Gamma_2$ ：第二个追踪器使用颜色直方图来考虑目标对象的局部颜色分布。一个边界框也分为  $4 \times 4$  块，CIE Lab 颜色直方图有 48 箱是从每个块中提取。因此，特征向量的维数为 768。

$\Gamma_3$ ：第三个追踪器采用了类似于 [31], [8] 的光照不变特征。首先，我们从 L 通道图像中得到一个梯度幅度图像。然后，我们得到了幅度的累积直方图。通过采用累积直方图，我们将幅度图像转换成一个等级的图像，其中每个幅度被对应的累积直方图值替换。单级图像和三通道的 CIE Lab 图像集成到一四通道图像。最后，1024 维特征向量的空间通过向下采样的四通道边界框转换成  $16 \times 16$  块。

为了测量特征向量  $u$  和  $v$  的相似性，我们使用交叉核结合这两者，给出公式

$$k(u, v) = \frac{1}{D} \sum_{i=1}^D \min(u_i, v_i) \quad (3)$$

其中  $D$  是特征尺寸。

#### 4.2. 轨迹鲁棒性分析值

组件追踪器  $\Gamma_1$ ,  $\Gamma_2$  和  $\Gamma_3$  产生了三条轨迹的假设。我们测量每个追踪

器的鲁棒性，并选择鲁棒性最强大的轨迹假设作为最终轨迹。

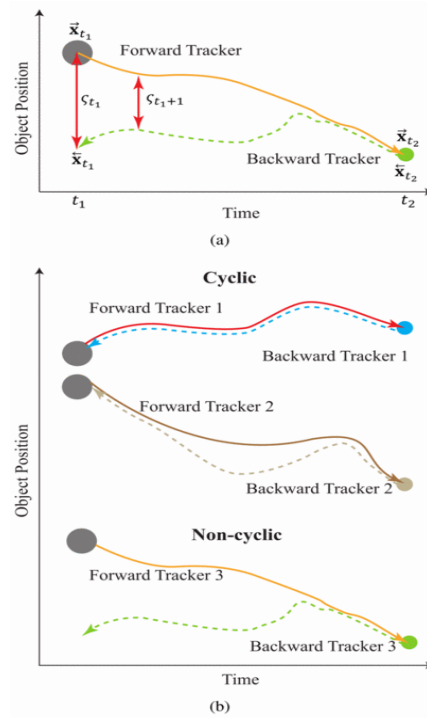


图 2. 轨迹分析：(a) 一对向前和向后的轨迹的几何相似性和 (b) 一对向前和向后的轨迹的循环特性。

让我们来描述如何计算一个追踪器的鲁棒性值。首先，追踪器一直向前追踪目标对象。让  $\bar{i}_t$  表示第  $t$  帧的边界框位置，这是由追踪器在这种向前的方式中估计得到的。从第  $t_1$  帧到第  $t_2$  帧的前向轨迹表示为

$$\bar{X}_{t_1 t_2} = \{\bar{X}_{t_1}, \bar{X}_{t_1-1}, \dots, \bar{X}_{t_2}\} \quad (4)$$

其中， $t_1 < t_2$ 。然后，在第  $t_2$  帧的  $\bar{X}_{t_2}$  位置处，我们初始化一个同样的追踪器在时间的倒序中向后追踪目标。 $S_t$  是第  $t$  帧时的向后估计位置。从帧  $t_2$  到帧  $t_1$  的后向轨迹表示为

$$\bar{X}_{t_2 t_1} = \{\bar{X}_{t_2}, \bar{X}_{t_2-1}, \dots, \bar{X}_{t_1}\} \quad (5)$$



需要注意的是，在时间间隔  $[t_1, t_2]$  中的最后一帧  $t_2$  时，有  $\bar{X}_{t_2} = \bar{X}_{t_1}$ 。

我们通过采用三种测量方式：几何相似性，循环重量，外观相似，使用后向轨迹来检查前向轨迹的可靠性。如图 2 (a) 所示，在帧  $t$  时的几何相似  $\delta_t$  的定义为

$$\zeta_t = \exp\left(-\frac{\|\bar{X}_t - \bar{X}_t\|^2}{\sigma_1^2}\right) \quad (6)$$

这个公式使用了正向位置  $\bar{X}_t$  和后向  $\bar{X}_t$  之间的距离，其中  $\sigma_1^2 = 500$ 。理想的情况下，后向轨迹应该与前向轨迹一致。在这种情况下，几何相似性  $\zeta_t$  为 1。

接下来，我们评估的向前和向后的轨迹  $\bar{X}_{t_1:t_2}$  和  $\bar{X}_{t_2:t_1}$  的循环权重。虽然  $\bar{X}_{t_2} = \bar{X}_{t_1}$ ，后向轨迹的末端位置与前向轨迹的开始位置可能是不一致的，i. e.  $\bar{X}_{t_1} \neq \bar{X}_{t_2}$ ，这是由于追踪失败才引起的。在这种情况下，这两个轨迹连接不形成循环。图 2 (b)，追踪 1 和 2 形成循环，但追踪 3 不。追踪 1 产生相同的向前和向后的轨迹，形成一个周期，这表明成功的追踪的可能性很高。在这种情况下，在整个区间  $[t_1, t_2]$  中的几何相似性  $\zeta_t$  为 1。追踪器 2 也形成一个周期，但在中间的时间间隔，当  $\zeta_t$  减少时，

后向的轨迹与前向的偏离了。然而，这可能是由于暂时的遮挡，毕竟，追踪 2 是成功的。与之相反，非循环轨迹追踪 3 表面了一个可能的追踪故障。

在实践中，我们首先计算两个相应的边界框重叠率，给定的公式

$$\zeta_t = \frac{\Delta(\bar{X}_t, \bar{X}_t)}{\Delta(\bar{X}_t) + \Delta(\bar{X}_t)} \quad (7)$$

其中， $\Delta(i_t)$  和  $\Delta(K_t)$  分别是边界框为  $i_t$  和  $K_t$  时的面积， $\Delta(2, S_t)$  表示的是两个框的重叠区域面积。当  $\zeta_t \leq 0.3$  时，它表明  $a_{e_t}$  和  $K_t$  不匹配。

我们在一个很短的区间  $[t_1, t_1 + \varepsilon]$  计算了不匹配帧的数目  $\Omega$  以检验  $\bar{X}_{t_1:t_2}$  和  $\bar{X}_{t_2:t_1}$  是否构成一个周期。然后，我们定义了  $\chi$  的循环权  $\bar{X}_{t_1:t_2}$  和  $\bar{X}_{t_2:t_1}$  为

$$\chi = \begin{cases} 10^6 \dots \text{if } \cdot \Omega \cdot \text{is} \cdot 0 \cdot \text{or} \cdot 1 & (8) \\ 1 \dots \text{otherwise} \end{cases}$$

请注意， $10^6$  是任意的大数，因此，在两者之间的情况下设置不同的循环权  $\chi$ ，从而区分循环与非循环的轨迹。此外，计算  $\chi$ ，我们只考虑第一个  $\varepsilon$  帧中的正向轨迹，以允许临时不匹配的  $\bar{X}_t$  和  $\bar{X}_t$  在  $[t_1, t_2]$  中的很短的时间。在这项工作中，当  $\varepsilon=4$  时， $t_2 - t_1 = 30$ 。

同时，我们定义外观相似性来衡量在  $\bar{X}_{t_1:t_2}$  中的后向估算位置  $\bar{X}_t$  的可靠性。假设从输入序列的第一帧到第  $t_1$  帧的轨迹  $\bar{X}_{1:t_1}$  已经用多个跟追踪器定下来了。我们保持一组四个图像斑块  $S_{t_1}$ ，这些是从边界框中沿着  $\bar{X}_{1:t_1}$  选定的。在第一帧中的边界框默认被选中的情况下，其他三个箱子被更新，以得到最高的判别函数值直到第  $t_1$  帧。 $P(X)$  表示以  $X$  为中心的图像斑块。然后，外观相似性  $P(\bar{X}_t)$  到  $S_{t_1}$  的定义为

$$\phi_t = \exp \left( - \frac{\sum_{Q \in S_{t_1}} \|K \cdot (P(\bar{X}_t) - Q)\|^2}{4wh\sigma_2^2} \right) \quad (9)$$

其中， $\sigma_2^2=900$ ， $w$  和  $h$  是分别为边界框的宽度和高度。 $K$  是一个高斯权重网络，“ $\cdot$ ”表示的像素量倍增。一个小  $\phi_t$  表明，在  $\bar{X}_t$  的边界框会迅速从过去的状态改变其外观，这可能是追踪误差。

最后，我们结合几何相似性，循环权重和外观相似性来量化追踪器在  $[t_1, t_2]$  时的鲁棒性。给定公式

$$\psi_{t_1:t_2} = \chi \sum_{t=t_1}^{t_2} \zeta_t \phi_t \quad (10)$$

一个很大的鲁棒性值  $\psi_{t_1:t_2}$  可表明，追

踪的前向轨迹  $\sum_{t_1:t_2}$  是可靠的。

### 4.3. 最优轨迹选择追踪

我们通过每个组件追踪器得到一对向前和向后的轨迹，分析这些轨迹完成追踪。然而，为了减轻计算负担，我们在每一组的连续  $\tau$  帧执行轨迹分析，但我们分享连续 2 组之间的边界帧。更具体地说，如果在时间间隔  $[t-\tau, t]$  中进行轨迹分析，它是通过共享帧  $t$  在下一个区间  $[t, t+\tau]$  中进行的。因此，在每  $\tau$  帧时，轨迹分析程序才被调用。

对于区间  $[t-\tau, t]$ ，首先，我们分别采用组件追踪器  $\Gamma_1$ ， $\Gamma_2$  和  $\Gamma_3$  获得三个前向轨迹和相应的后向轨迹。然后，我们在 (10) 中计算三个追踪器的鲁棒值，并选择收益率最高的追踪器作为分数最优。用

$$\bar{X}_{t-\tau:t}^* = \left\{ \bar{X}_{t-\tau}^*, \bar{X}_{t-\tau+1}^*, \dots, \bar{X}_t^* \right\}$$

表示最优追踪轨迹。这被用来作为追踪结果。然后，我们恢复所有组件追踪器到以前帧  $t-\tau$  时的状况，并利用沿着最终轨迹  $R_{t-\tau:t}^*$  的样品更新它们的分类器。然而，为了防止损坏样品的使用，如果在帧  $t$  时几何相似性和外观相似相应的的乘积  $\sigma_t \phi_t^*$  比预先指定的阈值  $\delta_1=0.2$  小，我们就在此帧跳过更新。然后，三个组件追踪器都



在  $X_i^*$  处开始，接下来的时间间隔为  $[t, t + \tau]$ 。

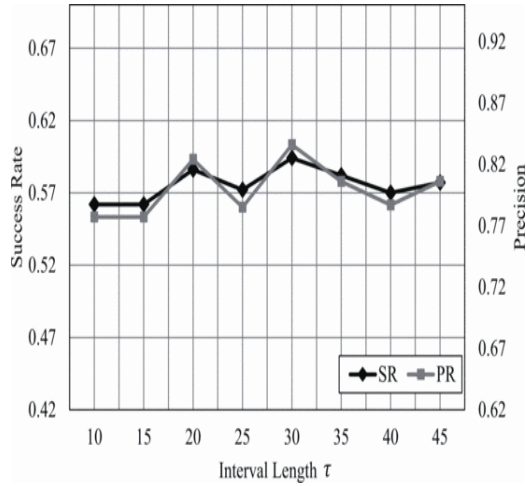


图 3. 基于基准数据集[28], 通过时间间隔长度  $t$  的平均成功率 (pr) 和高准确率 (pr)。

当一个非目标对象位于目标对象附近时，我们不能用一个短的区间长度  $\tau$  来选择最优追踪。对于非目标对象，该间隔长度应该足够长，以充分地从目标对象中分离。另一方面，若具有较长的间隔长度，追踪故障的影响就会变得更加严重。因此，重要的是选择一个适当的  $\tau$ 。按照

$\tau$ ，图 3 画出了平均成功率 (SR) 和平均高精度 (PR)。SR 是一个成功曲线下的面积，PR 是帧的百分比，这些帧的目标对象的估计位置在地面事实的 20 像素范围内。注意，SR 和 PR 在  $\tau=30$  时是最大的。我们因此设定  $\tau=30$ 。

#### 4.4. 故障处理

当 (8) 中最佳追踪器的循环权是 1 时，我们声明，一个追踪故障发生在当前区间。当所有组件追踪器连续 2 帧满足  $\zeta_i \phi_i \leq \delta_2$ ，且持续时间较长，超过  $2\tau/3$  时，我们也会检测到追踪故障。在这项工作中， $\delta_2 = 0.004$ 。当检测到追踪失败时，我们不更新所有组件追踪器的分类器。此外，当一个追踪故障发生时，目标对象可以位于普通搜索范围之外。因此，我们增加的搜索范围从 4 到  $4 \times R$ ，但只检查每一个在增加的搜索范围内的 64 个样本位置，以减少计算复杂性。

	STRUCK	SCM	TLD	ASLA	CXT	VTD	KCF	MEEM	MTA <sub>1</sub>	MTA <sub>2</sub>	MTA <sub>3</sub>	MTA
SR	0.475	0.499	0.437	0.434	0.426	0.416	0.514	<u>0.579</u>	0.508	0.538	0.524	<b>0.595</b>
PR	0.647	0.649	0.608	0.532	0.575	0.576	0.740	<u>0.836</u>	0.694	0.742	0.704	<b>0.838</b>

表 1. (28) 中基准序列的平均成功率 (SR) 和平均准确率 (PR) 的对比。八个传统的算法进行了测试: STRUCK [10], SCM [32], TLD [14], ASLA [13], CXT [7], VTD [16], KCF [11] 和 MEEM [31]。最后四列对应的算法: MTA 使用所有组件追踪器  $\Gamma_1, \Gamma_2$  和  $\Gamma_3$ ，而 MTA 采用单一追踪器  $\Gamma_i (i=1,2,3)$ 。最佳的结果为粗体字，

下划线是第二个最好的结果。

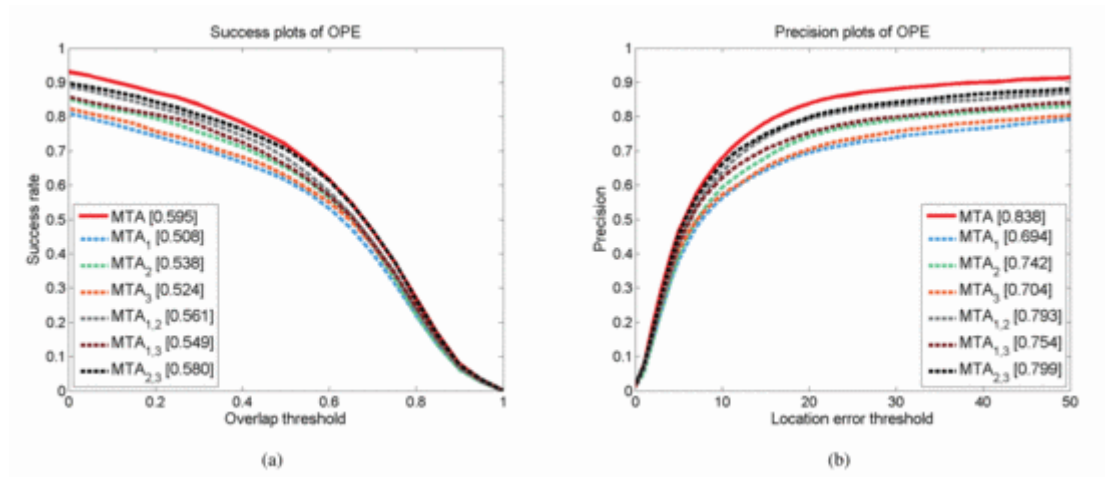


图 4. 采用不同的组合成分追踪器的算法的性能：(a) 和 (b) 分别为基准数据集 [ 28 ] 上的成功图与精度图。MTA 使用所有三种组合，MTA<sub>i</sub> 意味着只有  $\Gamma_i$  被用，MTA<sub>i,j</sub> 意味着  $\Gamma_i$  和  $\Gamma_j$  被用。

	STRUCK	KCF	MEEM	MTA
Badminton [18]	0.65(0.46)	0.22(0.15)	0.52(0.37)	<b>0.89(0.63)</b>
Bird2 [26]	0.10(0.10)	0.56(0.64)	0.99(0.75)	<b>1.00(0.77)</b>
Board [22]	0.68(0.69)	0.70(0.73)	0.60(0.72)	<b>0.79(0.80)</b>
GirlMov [26]	0.19(0.18)	0.08(0.08)	0.87(0.63)	<b>0.92(0.67)</b>
SnowBoard [18]	0.18(0.17)	0.08(0.10)	0.19(0.14)	<b>0.41(0.28)</b>
Surfer [3]	0.97(0.58)	<b>1.00(0.68)</b>	0.98(0.62)	0.95(0.50)
Youngki [18]	0.06(0.15)	0.07(0.21)	0.59(0.55)	<b>0.67(0.60)</b>
Average	0.40(0.33)	0.39(0.37)	0.68(0.54)	<b>0.80(0.61)</b>

表 2. 最先进的算法与关于非基准序列的传统的算法的比较。对于每一个测试，这 2 个数字是 PR (SR) 的结果。最好的结果用粗体标出。

## 5、 实验结果

我们在最近的基准测试数据集 [ 28 ] 上测试了建议算法，该数据集包含 50 个具有挑战性条件的测试序列，如光照变化，遮挡与超出视野。我们八个传统的追踪器与提出的算法比较：STRUCK [10]，SCM [32]，TLD [14]，ASLA [13]，CXT [7]，VTD [16]，

KCF [11]，和 MEEM [31]。前六个追踪器在基准中，而最后两个追踪器也因其优良的追踪结果被加了进去。

表 1 比较了建议算法与传统算法的平均 SR 和 PR。这个建议算法包括第 4.4 节中的追踪故障处理，来抑制误差传播，实现鲁棒性追踪。因此，我们执行 OPE [ 28 ] 来证明所提出算法的优势。在表 1 中，所提出的算法进行测试有四种方式：MTA 使用所有组件追踪器  $\Gamma_1$ ， $\Gamma_2$  和  $\Gamma_3$ ，而 MTA<sub>i</sub> 只使用一个单一的组件追踪器

$\Gamma_i (i=1,2,3)$ 。注意，组件追踪器是基于 STRUCK 的。然而，即使只用单一的组件追踪器，该算法由于具有跟追故障处理程序，也是优的。此外，当基于多假设轨迹分析追踪器  $\Gamma_1$ ， $\Gamma_2$  和  $\Gamma_3$  结合时，MTA 比传统的算法提供了更好的性能。特别是，MTA 的 SR 和 PR

分别优于STRUCK 25.3%和29.5%。表2表明该MTA在非基准测试序列中也优于STRUCK, KCF和MEEM追踪器。

图4显示了采用不同的组合成分追踪算法的成功图和精密图。我们测试了每一个可能的组合。通过采用 $\Gamma_2$ , MTA<sub>2</sub>使用颜色特征来提供了比其它分别使用了Haar和光照不变特点的追踪器MTA<sub>1</sub>, MTA<sub>3</sub>更精确的追踪结

果。然而, 当使用追踪器 $\Gamma_1$ 和 $\Gamma_3$ 时, MTA<sub>1,3</sub>比MTA<sub>2</sub>产生了更好的结果。此外, 通过结合三个追踪器 $\Gamma_1, \Gamma_2$ 和 $\Gamma_3$ 时, MTA提供了最好的结果。总的来说, 提出的多重假设轨迹分析总是提供了更好的结果, 因为它拥有更多的组件追踪器。

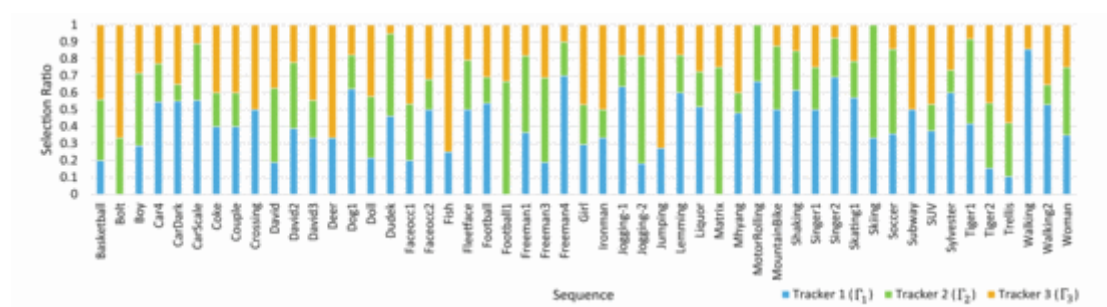


图5. 该组件跟踪器为每个测试序列的选择比。

图5显示了对于每个序列, 组件追踪器 $\Gamma_1, \Gamma_2$ 和 $\Gamma_3$ 的选择率。在一些序列中, 一个特定跟踪的所有前向轨迹都没被选。例如, 由于Haar特征描述符没有从附近的编号锥中区分目标运动员, 在“Bolt”序列中的 $\Gamma_1$ 没有被选择, 如图1所示。平均而言, 追踪器 $\Gamma_1, \Gamma_2$ 和 $\Gamma_3$ 的选择比率分别为41%, 28%和31%。由于每个组件追踪器在不同的现场有它自己的优点, 三个追踪器的选择率相互类似。

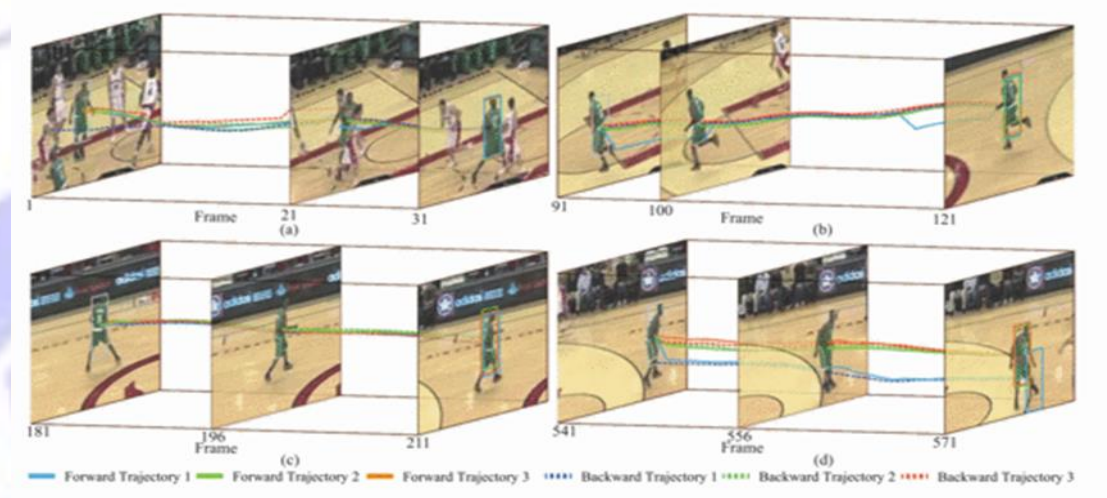


图6. 关于“篮球”序列的前向和后向轨迹的例子。



图6为提出的MTA追踪器在“篮球”序列中的追踪过程。固体和虚线分别描绘向前和向后轨迹。此外，蓝色，绿色和橙色的颜色分别代表组件跟踪1, 2, 和3。在这个例子中，2, 3, 1, 和2的正向轨迹如图6，分别为（甲）、（乙）、（丙）。在图6（甲），

因为目标球员在绿色的军装是部分遮挡的另一个玩家在白色制服在21帧， $\Gamma_1$ 后向轨迹跟随的白色球员后，闭塞。后向追踪器 $\Gamma_3$ 在闭塞中也变得不稳定。因此，基于多重假设的分析，提出了最后选择 $\Gamma_2$ 的轨迹。



图7. 追踪的例子，该MTA追踪器和传统的STRUCK, KCF和MEEM追踪器。

图7比较了MTA的追踪结果追踪与STRUCK, KCF和MEEM。“篮球”和“慢跑”序列有着杂乱的背景和闭塞。因此，常规追踪器追踪目标失败了。相反，该MTA选择可靠的轨迹追踪，利用几何相似性，循环权重，和外观相似成功地追踪对象。同时，在“两人舞蹈”和“滑雪”序列，有些功能不能区分目标背景。因此，传统的追踪器的故障视特征选择而定。不过，MTA鲁棒追踪目标与使用多个不同特点的追踪器。

## 6、 总结

在本文中，我们为鲁棒的视觉轨迹追踪提出了多重假设的分析。该MTA基于三元追踪。过一段时间，每一个组件追踪计算前向轨迹后向轨迹。通过分析向前和向后的运动轨迹，MTA提取几何相似性，循环权，外观相似性，然后结合到鲁棒性跟踪器得分。MTA然后选择最优跟踪器收益率最高的得分，然后使用它的前进轨迹作为最终的跟踪结果。实验结果

表明MTA实现更准确的和强大的比传统的国家的最先进的跟踪器跟踪。

未来的研究问题包括多重假设的应用其他组件跟踪轨迹分析。例如，一些最先进的跟踪器可以作为部分跟帖，而当前实现使用三部分跟踪，都是基于来袭。

## 致谢

本研究部分由基础科学研究方案，通过韩国（NRF）国家研究基金会，由教育部资助（2013r1a1a2011920），另一部分由NRF给予资助科技部、信息和通信技术及未来规划（MSIP）（2009号—0083495）。

## 参考文献

- [1] S. Avidan. Support vector tracking. TPAMI, 26(8):1064–1072, Aug. 2004. 2
- [2] S. Avidan. Ensemble tracking. TPAMI, 29(2):261–271, Feb.2007. 2
- [3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. TPAMI,33(8):1619–1632, Aug. 2011. 6
- [4] C. Bailer, A. Pagani, and D. Stricker. A superior tracking approach: Building a strong tracker through fusion. In ECCV,pages 170–185. 2014. 2
- [5] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solvingmulticlass support vector machines with LaRank. In ICML,pages 89–96, 2007. 2
- [6] A. Bordes, N. Usunier, and L. Bottou. Sequence labellingSVMs trained in one pass. In ECML-PKDD, pages 146–161,2008. 2
- [7] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploringsupporters and distracters in unconstrained environments.In CVPR, pages 1177–1184, 2011. 6
- [8] U. Fecker, M. Barkowsky, and A. Kaup. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. TCSVT, 18(9):1258–1267, Sep. 2008. 3
- [9] Y. Gao, R. Ji, L. Zhang, and A. Hauptmann. Symbiotic tracker ensemble toward a unified tracking framework.TCSVT, 24(7):1122–1131, Jul. 2014. 2
- [10] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In ICCV, pages 263–270, 2011. 2, 3,6
- [11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Highspeed tracking with

- kernelized correlation filters. (in press)TPAMI, 2015. 6
- [12] Y. Hua, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In ECCV, pages 172–187, 2014. 2
- [13] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In CVPR, pages 1822–1829, 2012. 6
- [14] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In CVPR, pages 49–56, 2010. 6
- [15] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. TPAMI, 34(7):1409–1422, Jul. 2012. 2
- [16] J. Kwon and K. M. Lee. Visual tracking decomposition. In CVPR, pages 1269–1276, 2010. 1, 2, 6
- [17] J. Kwon and K. M. Lee. Tracking by sampling trackers. In ICCV, pages 1195–1202, 2011. 1, 2
- [18] J. Kwon and K. M. Lee. Wang-Landau Monte Carlo-based tracking methods for abrupt motions. TPAMI, 35(4):1011–1024, Apr. 2013. 6
- [19] I. Leichter, M. Lindenbaum, and E. Rivlin. A probabilistic framework for combining tracking algorithms. In CVPR, pages II–445–II–451 Vol.2, 2004. 1
- [20] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel. A survey of appearance models in visual object tracking. ACM TIST, 4(4):58:1–58:48, Oct. 2013. 1
- [21] Y. Lu, T. Wu, and S.-C. Zhu. Online object tracking, learning, and parsing with and-or graphs. In CVPR, pages 3462–3469, 2014. 1, 2
- [22] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In CVPR, pages 723–730, 2010. 6
- [23] J. Supančič and D. Ramanan. Self-paced learning for longterm tracking. In CVPR, pages 2379–2386, 2013. 1, 2
- [24] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In ICCV, pages 1–8, 2007. 1, 2
- [25] L. Wang, H. Yan, H.-Y. Wu, and C. Pan. Forward-backward



- mean-shift for visual tracking with local-backgroundweighted histogram. TITS, 14(3):1480–1489, Sep. 2013. 2
- [26] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In ICCV, pages 1323–1330, 2011. 6
- [27] H. Wu, R. Chellappa, A. Sankaranarayanan, and S. Zhou. Robust visual tracking using the time-reversibility constraint. In ICCV, pages 1–8, 2007. 2
- [28] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In CVPR, pages 2411–2418, 2013. 2, 5, 6
- [29] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. Neurocomputing, 74(18):3823–3831, 2011. 1
- [30] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. ACM CSUR, 38(4), Dec. 2006. 1
- [31] J. Zhang, S. Ma, and S. Sclaroff. MEEM: Robust tracking via multiple experts using entropy minimization. In ECCV, pages 188–203, 2014. 1, 2, 3, 6
- [32] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In CVPR, pages 1838–1845, 2012. 6

