

征的各个通道进行分别处理即可。

如2.3节所述，KCF 采用简单的线性插值作为模型更新策略。这种更新方法已经被证明是不够理想的^[8]，因为每次模型更新时（公式2.7），仅考虑了当前帧中的目标物体外表图像，而之前帧中的目标外表的贡献，将按照学习速率 η 的指数级函数不断减小。显然，这种更新策略是不够鲁棒的。当跟踪出现偏移，或者目标外表突发剧烈变化时，KCF 原来积累的准确模型将被迅速污染，从而很可能导致更大的跟踪偏移。

为了提高 KCF 的相关滤波器的鲁棒性，本节将把线性插值方法替换为 ACT 中的鲁棒更新方法。在更新模型的参数矩阵 α 时，假设当前帧为第 i 帧，则从第 1 帧到当前帧的所有目标物体图像块可记为 $\{\mathbf{x}_j : j = 1, \dots, i\}$ 。更新参数矩阵的过程变为最小化以下目标函数的过程：

$$\begin{aligned} \min_{\alpha} \sum_{j=1}^i \phi_j (\|\mathbf{f}(\mathbf{x}_j) - \mathbf{y}\|^2 + \lambda \langle \alpha, \alpha \rangle) = \\ \min_{\alpha} \sum_{j=1}^i \phi_j (\|\mathcal{F}^{-1}(\hat{\mathbf{k}}^{\mathbf{x}_i \mathbf{x}_j} \cdot \hat{\alpha}) - \mathbf{y}\|^2 + \lambda \langle \alpha, \alpha \rangle). \end{aligned} \quad (2.9)$$

该目标函数同时考虑了当前帧和所有历史帧中的目标物体图像块，其实质为参数矩阵 α 对于所有帧中目标物体外表的平方误差加权和。 ϕ_j 即是权值，这里将其直接设置为学习速率 η 。

为了能够最小化上述目标函数，经过推导（详见 [8]），新的参数矩阵更新公式为：

$$\begin{aligned} \alpha_N &= \eta \hat{\mathbf{k}}^{\mathbf{x}_i \mathbf{x}_i} \cdot \hat{\mathbf{y}} + (1 - \eta) \alpha_N; \\ \alpha_D &= \eta \hat{\mathbf{k}}^{\mathbf{x}_i \mathbf{x}_i} \cdot (\hat{\mathbf{k}}^{\mathbf{x}_i \mathbf{x}_i} + \lambda) + (1 - \eta) \alpha_D. \end{aligned} \quad (2.10)$$

其中 α_N 和 α_D 分别为频率域 α 的分子和分母，即 $\hat{\alpha} = \alpha_N / \alpha_D$ 。值得注意的是，虽然所有历史帧中的目标图像块都在公式2.9中显式地考虑了，但它们隐含在了迭代更新过程（公式2.10）中，因此每次更新 α 时仅需要当前帧中的目标物体图像。而 KCF 模型的另一部分，目标外观 $\bar{\mathbf{x}}$ ，仍然按照公式2.8进行线性插值式的更新。

2.5 将“目标候选”嵌入跟踪器中

原 KCF 跟踪器在经过特征整合和鲁棒更新的优化后，将足以胜任对目标候选的辨别。因此本节将把目标候选生成器 EdgeBoxes 嵌入优化后的 KCF 跟踪框架中，以同时利用目标候选的灵活性和优化后的 KCF 的准确辨别力。嵌入目标候选生成器后的整个跟踪过程可视化地展示在了图2.5中。

在初始化时，第一帧中，记用户标注的或者目标检测得到的目标物体边界框

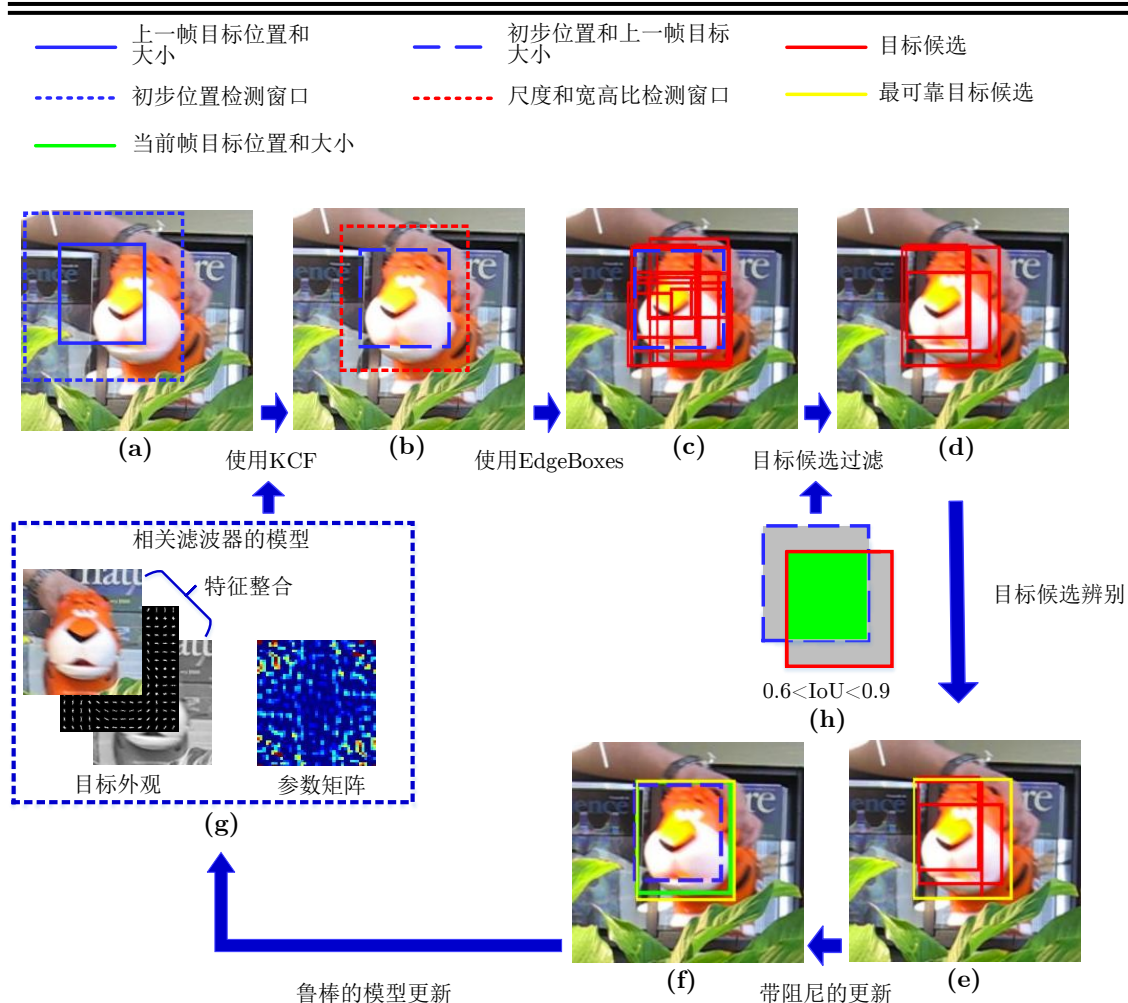


图 2.5 本章跟踪器的可视化跟踪过程

大小为 $w_1 \times h_1$ ，中心位于 \mathbf{l}_1 。将该边界框扩大为 $s^d w_1 \times s^d h_1$ ，即得到了位于 \mathbf{l}_1 的上下文窗口。这里的 s^d 被称为尺度因子，必须大于 1 以包含一定上下文信息，且覆盖下一帧中目标物体可能出现的位置。为了初始化 KCF 的模型，此时将根据该上下文窗口提取出图像块 \mathbf{x}_1 ，并用 \mathbf{x}_1 初始化参数矩阵 α （公式 2.4）。至于目标外观 $\bar{\mathbf{x}}$ ，将其直接初始化为 \mathbf{x}_1 。

跟踪过程开始后，每当新的一帧（记为第 i 帧）到来时，首先根据上一帧中的目标位置 \mathbf{l}_{i-1} 和目标大小 $w_{i-1} \times h_{i-1}$ ，设置一个中心位于 \mathbf{l}_{i-1} ，大小为 $s^d w_{i-1} \times s^d h_{i-1}$ 的“初步位置检测窗口”，如图 2.5(a) 所示。然后根据该窗口提取图像块 \mathbf{z}^d 。由于跟踪过程中，检测到的目标物体大小会不断变化，因此需要利用双线性插值 (Bilinear Interpolation) 将 \mathbf{z}^d 缩放到 $s^d w_1 \times s^d h_1$ ，才能利用无尺度适应力的 KCF 相关滤波器。将 KCF 应用于图像块 \mathbf{z}^d 之后（公式 2.6），即可根据 \mathbf{f} 中最大元素的位置，判断出新的目标位置 \mathbf{l}_i^d 。该目标位置称作“初步位置”，其对应的 \mathbf{f} 中最大元素值记为 v 。

下一步如图2.5(b)所示,以初步位置 \mathbf{l}_i^d 为中心, $s^e w_{i-1} \times s^e h_{i-1}$ 为大小,构建一个“尺度和宽高比检测窗口”。 s^e 也是一个尺度因子,但是应当设置得比 s^d 小,因为目标物体尺度的变化通常小于其位移。根据尺度和宽高比检测窗口提取图像块 \mathbf{z}^p 后,将其输入目标候选生成器 EdgeBoxes 中。如图2.5(c)所示,EdgeBoxes 将会输出大量的目标候选,即大量可能包含物体的边界框。这些边界框已按照各自所得“分数”进行了排序,而该分数代表的就是边界框内包含物体的置信度。

对于 EdgeBoxes 输出的边界框,本节仅提取其前 N 个,并且对它们进行过滤:计算每一个目标候选边界框与“初步目标物体边界框”的重叠率,如果重叠率大于 0.9 或者小于 0.6,则将该目标候选剔除。这里的初步目标物体边界框位于 \mathbf{l}_i^d ,大小为 $w_{i-1} \times h_{i-1}$,即一个位于初步位置,大小等于前一帧中目标大小的边界框。如图2.5(h)所示,重叠率使用 IoU (Intersection over Union) 进行度量,其计算方式为将两个边界框的交集面积除以它们的并集面积。IoU 大于 0.9 的边界框几乎和初步目标物体边界框相同,无需纳入考虑;而 IoU 小于 0.6 的边界框很有可能是错误的目标候选,或者其中包含的是非目标物体,均需要被剔除。

如图2.5(d)所示,通过过滤后的目标候选数目已大幅下降。但是,仍然需要用相关滤波器对这些目标候选进行再次辨别,才能找出图2.5(e)中所示的“最可靠目标候选”。对于每一个经过过滤的目标候选边界框,按尺度因子 s^d 进行扩大,并从当前帧中提取出对应的图像块,记为 \mathbf{p} 。由于 \mathbf{p} 可能具有任意的尺度和宽高比,因此也需要将它缩放到 $s^d w_1 \times s^d h_1$ 后,才能使用相关滤波器进行辨别。辨别一个目标候选图像块和当前目标外观的相似度时,使用下式:

$$f(\mathbf{p}) = \text{sum}(\mathbf{k}^{\bar{\mathbf{x}}\mathbf{p}} \cdot \boldsymbol{\alpha}), \quad (2.11)$$

这里的 $\text{sum}(\cdot)$ 代表对矩阵中所有元素的叠加操作, $f()$ 和公式2.3中的 $f()$ 意义相同,是一个代表着 \mathbf{p} 和 $\bar{\mathbf{x}}$ 的相似度的标量。由于这里仅需要对目标候选图像块本身进行辨别,而无需辨别它的所有循环位移,因此用公式2.11来代替公式2.6。事实上,公式2.11就是公式2.6转换到空间域的结果。在辨别完所有目标候选之后,本节将找出具有最大 f 值(记为 f_{max})的目标候选,并记录其中心位置为 \mathbf{l}_i^p ,大小为 $w_i^p \times h_i^p$ 。

如果 f_{max} 小于 v ,说明最可靠的目标候选边界框的准确性仍然不如“初步目标物体边界框”。此时应放弃所有目标候选,认为初步位置 \mathbf{l}_i^d 就是当前帧中的目标位置 \mathbf{l}_i ,且当前目标物体大小 $w_i \times h_i$ 不变,仍然等于 $w_{i-1} \times h_{i-1}$ 。如果 f_{max} 大于 v ,则通过一个带阻尼的更新过程,将初步目标物体边界框和最可靠目标候选

相结合：

$$\begin{aligned} \mathbf{l}_i &= \mathbf{l}_i^d + \gamma(\mathbf{l}_i^p - \mathbf{l}_i^d); \\ (w_i, h_i) &= (w_{i-1}, h_{i-1}) + \gamma((w_i^p, h_i^p) - (w_{i-1}, h_{i-1})). \end{aligned} \quad (2.12)$$

上式的 γ 为阻尼因子，其作用如图2.5(f) 所示。用带阻尼的更新来最终确定目标的位置和大小，可以防止过于敏感的目标状态变化，同时还能降低跟踪错误或是目标候选错误带来的影响，从而让跟踪器更为鲁棒。

在当前帧中完成跟踪后，还需要提取一个新的目标物体图像块来更新 KCF 的参数矩阵 α 和目标外观 $\bar{\mathbf{x}}$ 。显然，该图像块位于新目标位置 \mathbf{l}_i ，大小为 $s^d w_i \times s^d h_i$ ，可被记为 \mathbf{x}_i 。将 \mathbf{x}_i 输入公式2.10，即可鲁棒地更新 KCF 的参数矩阵 α ；而更新目标外观 $\bar{\mathbf{x}}$ 则需要根据公式2.8。完成模型更新后，将进入针对下一帧的循环。

除了图2.5所示的可视化过程，本节的跟踪过程还规范化地列在了算法2.1中。更多可视化的目标候选辨别和带阻尼更新的例子可见图2.6。图2.6还清楚地展示出了目标候选方法是如何让本章的跟踪器适应尺度和宽高比变化的。正是由于目标候选的极大灵活性，本章的跟踪器可以轻易地找出被遮挡的目标（图2.6右上），并且还能准确地框出旋转的目标（图2.6左下）。

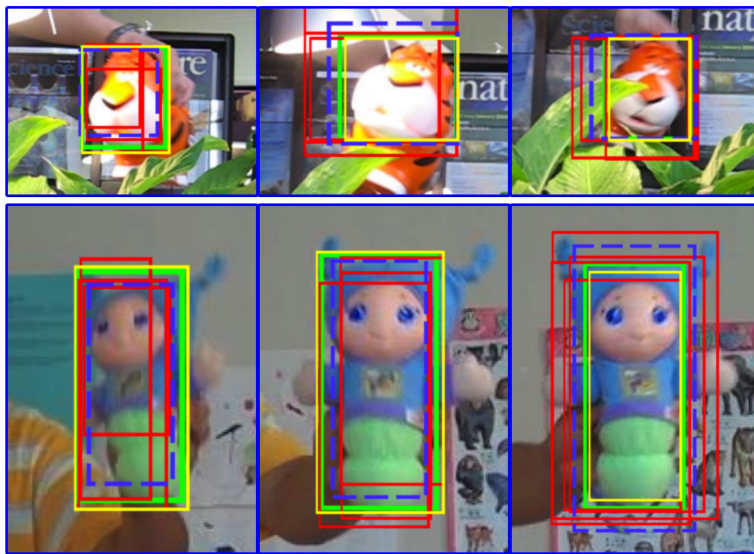


图 2.6 目标候选辨别和带阻尼更新的可视化例子（图例见图2.5）

2.6 参数设置

从上文中可以看出，本章的跟踪算法引入了大量的参数。这些参数对跟踪器的性能有着极大的影响，因此参数设置也是跟踪算法设计中至关重要的步骤。简单粗暴地进行大范围的参数遍历，寻找能够在测试集上取得最优效果的参数组合，看似合理，但却是不可行的。如果参数数目较多，各参数取值范围较大，若再加