For detection, patches are sampled from an image and matched against i.e.,
tics for the target and the background over time, for the object. In our case, these entries can be regarded as a signature set of codebook entries is activated to cast probabilistic votes for the ob-
of object parts. When matching an image against a codebook, a certain an object class. Codebooks model the spatial distribution and appearance of this end, we make use of a codebook-based detector [1] that is trained on

In this work, we demonstrate that an off-line trained class-specific detec-
tion confidence for the target, see Fig. 1.

While are already computed for the off-line creation of the codebook (2).

For tracking, however, one is not interested in the probability (2), but

Based on the posterior, the voting space is clustered (blue: foreground, red: background, green: uncertain). (c) Votes that contributed to the detected local maxima are used to update the instance-specific statistics.

The probability $p(P_i \in I|c=1, L(y))$ is estimated by counting the number of times a patch $P_i$ votes for the target instance $\{y|P_i \in I \cap \mathcal{P}_{L(y)}\}$ and the number of times it votes for other objects $\{y|P_i \notin I \cap \mathcal{P}_{L(y)}\}$:

$$p(P_i \in I|c=1, L(y)) = \frac{|\{y|P_i \in I \cap \mathcal{P}_{L(y)}\}|}{|\{y|P_i \in I \cap \mathcal{P}_{L(y)}\}| + |\{y|P_i \notin I \cap \mathcal{P}_{L(y)}\}|}. \ (4)$$

When the patch has not been previously activated for voting, we assume a fifty-fifty chance that the patch belongs to the instance $I$.

In order to compute (4), we have to estimate $\{y|P_i \in I \cap \mathcal{P}_{L(y)}\}$ and $\{y|P_i \notin I \cap \mathcal{P}_{L(y)}\}$. To this end, we assign a label to each vote based on the posterior distribution estimated by a particle filter (Fig. 2). Namely 1 (blue) or −1 (red) if we are confident that it either belongs to the instance or it does not. When the posterior is greater than zero but relatively low, we assign the label 0 (green) to it.

After labeling the elements in the Hough space, we search for strong local maxima in the positive and the negative cluster. The elements of the cluster labeled with 0 are discarded. Finally, we collect the votes that contributed to the local maxima and add them to the corresponding sets $\{y|P_i \in I \cap \mathcal{P}_{L(y)}\}$ and $\{y|P_i \notin I \cap \mathcal{P}_{L(y)}\}$. The details of the clustering and the algorithm are described in the paper.

We conclude that standard codebooks can be efficiently transformed into more instance-specific codebooks. Coupled with a particle filter, one obtains a powerful instance tracking method without the use of additional classifiers to distinguish several instances during tracking. Compared to a class-specific codebook, the accuracy is not only increased but the computation time is also reduced. Compared to on-line learning approaches, tracking is much more reliable subject to an off-line trained codebook. Although this prevents tracking arbitrary objects, it is not a practical limitation since the objects of interest usually belong to a well defined class.


