On-line Adaption of Class-specific Codebooks for Instance Tracking

Juergen Gall[1]
gall@vision.ee.ethz.ch
Nima Razavi[1]
nrazavi@vision.ee.ethz.ch
Luc Van Gool[1,2]
vangool@vision.ee.ethz.ch

[1] Computer Vision Laboratory
ETH Zurich, Switzerland
[2] IBBT, ESAT-PSI
K.U. Leuven, Belgium

In this work, we demonstrate that an off-line trained class-specific detector can be transformed into an instance-specific detector on-the-fly. To this end, we make use of a codebook-based detector [1] that is trained on an object class. Codebooks model the spatial distribution and appearance of object parts. When matching an image against a codebook, a certain set of codebook entries is activated to cast probabilistic votes for the object. For a given object hypothesis, one can collect the entries that voted for the object. In our case, these entries can be regarded as a signature for the target of interest. Since a change of pose and appearance can lead to an activation of very different codebook entries, we learn the statistics for the target and the background over time, *i.e.* we learn on-line the probability of each part in the codebook belonging to the target. By taking the target-specific statistics into account for voting, the target can be distinguished from other instances in the background yielding a higher detection confidence for the target, see Fig. 1.

A class-specific codebook as in [1, 2, 3, 4, 5] is trained off-line to identify any instance of the class in any image. It models the probability of the patches belonging to the object class $p(c{=}1|L)$ and the local spatial distribution of the patches with respect to the object center $p(\mathbf{x}|c{=}1, L)$. For detection, patches are sampled from an image and matched against the codebook, *i.e.* each patch $P(\mathbf{y})$ sampled from image location $\mathbf{y}$ ends at a leaf $L(\mathbf{y})$. The probability for an instance of the class centered at the location $\mathbf{x}$ is then given by

$$p\big(E(\mathbf{x})|L(\mathbf{y})\big) = p\big(\mathbf{y}-\mathbf{x}|c{=}1, L(\mathbf{y})\big) \cdot p\big(c{=}1|L(\mathbf{y})\big). \quad (1)$$

Since each patch P_i has been observed at a relative position $\mathbf{d}_i$ with respect to object center, the spatial distribution $p\big(\mathbf{y}-\mathbf{x}|c{=}1, L(\mathbf{y})\big)$ can be approximated by a sum of Dirac measures $\delta_{\mathbf{d}_i}$:

$$p\big(E(\mathbf{x})|L(\mathbf{y})\big) = \frac{1}{|\mathscr{P}_{L(y)}|}\left(\sum_{P_i \in \mathscr{P}_{L(y)}} p\big(c{=}1|L(\mathbf{y})\big)\cdot \delta_{\mathbf{d}_i}(\mathbf{y}-\mathbf{x})\right). \quad (2)$$

For tracking, however, one is not interested in the probability (2), but in the probability $p\big(E_I(\mathbf{x})|L(\mathbf{y})\big)$ where $E_I(\mathbf{x})$ is the evidence for a given instance I, namely the tracking target. In this case, $p\big(c{=}1|L(\mathbf{y})\big)$ needs to be replaced by the probability of a patch P_i belonging to the instance I, *i.e.* $p\big(P_i \in I|L(\mathbf{y})\big)$. Hence, we have

$$p\big(E_I(\mathbf{x})|L(\mathbf{y})\big) = \frac{1}{|\mathscr{P}_{L(y)}|}\left(\sum_{P_i \in \mathscr{P}_{L(y)}} p\big(P_i \in I|L(\mathbf{y})\big)\cdot \delta_{\mathbf{d}_i}(\mathbf{y}-\mathbf{x})\right) \quad (3)$$

$$= \frac{1}{|\mathscr{P}_{L(y)}|}\left(\sum_{P_i \in \mathscr{P}_{L(y)}} p\big(P_i \in I|c{=}1, L(\mathbf{y})\big)\cdot p\big(c{=}1|L(\mathbf{y})\big)\cdot \delta_{\mathbf{d}_i}(\mathbf{y}-\mathbf{x})\right).$$

While $p\big(P_i \in I|c{=}1, L(\mathbf{y})\big)$ needs to be estimated on-line, the other terms are already computed for the off-line creation of the codebook (2).

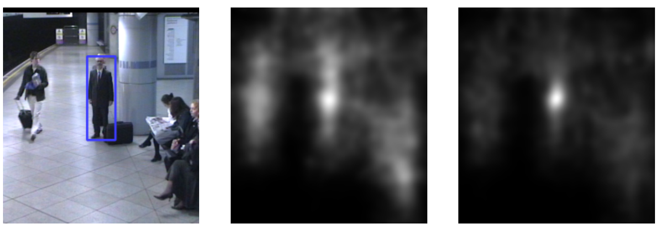

Figure 1: (a) Blue box indicates the instance of interest. (b) Voting image obtained by an off-line trained codebook for pedestrians. (c) Voting image obtained by the proposed instance-specific codebook.

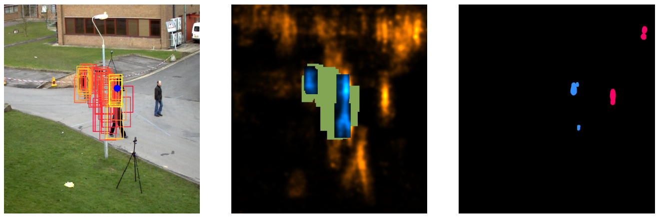

Figure 2: (a) After updating the particles, the multi-modal posterior distribution is approximated. The weights of the particles are indicated by color (*yellow: high, red: low*). The target is marked by a *blue dot*. (b) Based on the posterior, the voting space is clustered (*blue: foreground, red: background, green: uncertain*). (c) Votes that contributed to the detected local maxima are used to update the instance-specific statistics.

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

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

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 $\{y|P_i \in I \cap \mathscr{P}_{L(y)}\}$ and $\{y|P_i \notin I \cap \mathscr{P}_{L(y)}\}$. 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 $\{y|P_i \in I \cap \mathscr{P}_{L(y)}\}$ and $\{y|P_i \notin I \cap \mathscr{P}_{L(y)}\}$. 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.

[1] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

[3] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 80(1):16–44, 2008.

[4] R. R. Okada. Discriminative generalized hough transform for object dectection. In *International Conference on Computer Vision*, 2009.

[5] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, 2008.