On-line Hough Forests

Samuel Schulter[1]
schulter@icg.tugraz.at

Christian Leistner[2]
leistner@vision.ee.ethz.ch

Peter M. Roth[1]
pmroth@icg.tugraz.at

Luc Van Gool[2]
vangool@vision.ee.ethz.ch

Horst Bischof[1]
bischof@icg.tugraz.at

[1] Institute for Computer Graphics and Vision
Graz University of Technology
Austria

[2] Computer Vision Laboratory
ETH Zürich
Switzerland

Recently, Gall & Lempitsky [6] and Okada [9] introduced Hough Forests (HF), which emerged as a powerful tool in object detection, tracking and several other vision applications. HFs are based on the generalized Hough transform [2] and are ensembles of randomized decision trees, consisting of both classification and regression nodes, which are trained recursively. Densly sampled patches of the target object $\{P_i = (A_i, y_i, \mathbf{d}_i)\}$ represent the training data, where A_i is the appearance, y_i the label, and $\mathbf{d}_i$ a vector pointing to the center of the object. Each node tries to find an optimal splitting function by either optimizing the information gain for classification nodes or the variance of offset vectors $\mathbf{d}_i$ for regression nodes. This yields quite clean leaf nodes according to both, appearance and offset. However, typically HFs are trained in off-line mode, which means that they assume having access to the entire training set at once. This limits their application in situations where the data arrives sequentially, *e.g.*, in object tracking, in incremental, or large-scale learning. For all of these applications, on-line methods inherently can perform better.

Thus, we propose in this paper an on-line learning scheme for Hough forests, which allows to extend their usage to further applications, such as the tracking of arbitrary target instances or large-scale learning of visual classifiers. Growing such a tree in an on-line fashion is a difficult task, as errors in the hard splitting rules cannot be corrected easily further down the tree. While Godec *et al.* [8] circumvent the recursive on-line update of classification trees by randomly growing the trees to their full size and just update the leaf node statistics, we integrate the ideas from [5, 10] that follow a tree-growing principle. The basic idea there is to start with a tree consisting of only one node, which is the root node and the only leaf at that time. Each node collects the data falling in it and decides on its own, based on a certain splitting criterion, whether to split this node or to further update the statistics. Although the splitting criteria in [5, 10] have strong theoretical support, we will show in the experiments that it even suffices to only count the number n of samples P_i that a node has already incorporated and split when $n > \gamma$, where γ is a predefined threshold. An overview of this procedure is given in Figure 1.

This splitting criterion requires to find reasonable splitting functions with only a small subset of the data, which does not necessarily have to be a disadvantage when building random forests. As stated in Breiman [4], the upper bound for the generalization error of random forests can be optimized with a high strength of the individual trees but also a low correlation between them. To this end, we derive a new but simple splitting procedure for off-line HFs based on subsampling the input space on the node level, which can further decrease the correlation between the trees. That is, each node in a tree randomly samples a predefined number γ of data samples uniformly over all available data at the current node, which is then used for finding a good splitting function.

In the first experiment, we demonstrate on three object detection data sets that both, our on-line formulation and subsample splitting scheme, can reach similar performance compared to the classical Hough forests and can even outperform them, see Figures 2(a)&(b). Additionally, during training both proposed methods are orders of magnitudes faster than the original approach (Figure 2(c)). In the second part of the experiments, we demonstrate the power of our method on visual object tracking. Especially, our focus lies on tracking objects of a priori unknown classes, as class-specific tracking with off-line forests has already been demonstrated before [7]. We present results on seven tracking data sets and show that our on-line HFs can outperform state-of-the-art tracking-by-detection methods.

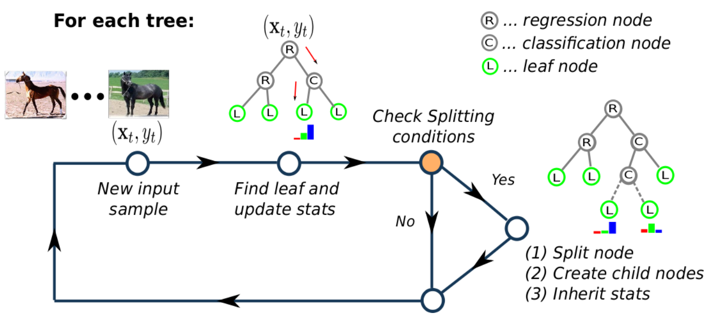

Figure 1: While labeled samples arrive on-line, each tree propagates the sample to the corresponding leaf node, which decides whether to split the current leaf or to update its statistics.

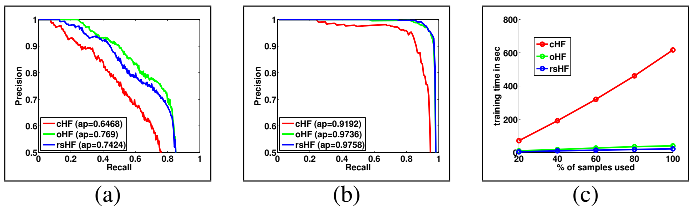

Figure 2: (a) & (b) Detection results for the two proposed methods *oHF* and *rsHF*, compared to the classic HFs (*cHF*) on the TUD dataset [1] and Weizmann dataset [3], respectively. (c) Speed-up during training.

This work has been supported by the Austrian FFG project Outlier (820923) under the FIT-IT program and the Austrian Science Foundation FWF under the project I535-N23.

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[2] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002.

[4] L. Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[5] P. Domingos and G. Hulten. Mining high-speed data streams. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000.

[6] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR*, 2009.

[7] J. Gall, N. Razavi, and L. Van Gool. On-line adaption of class-specific codebooks for instance tracking. In *BMVC*, 2010.

[8] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011. to appear.

[9] R. Okoda. Discriminative generalized hough transform for object detection. In *ICCV*, 2009.

[10] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *OLCV*, 2009.