Deeply Optimized Hough Transform: Application to Action Segmentation

Adrien CHAN-HON-TONG[1], Catherine ACHARD[2], Laurent LUCAT[1], and Patrick SAYD[1]

[1] CEA, LIST, DIASI, Laboratoire Vision et Ingénierie des Contenus, FRANCE
{adrien.chan-hon-tong,laurent.lucat,patrick.sayd}@cea.fr
[2] Institut des Systèmes Intelligents et Robotique, UPMC, FRANCE
catherine.achard@upmc.fr

Abstract. Hough-like methods (Implicite Shape Model, Hough forest, ...) have been successfully applied in multiple computer vision fields like object detection, tracking, skeleton extraction or human action detection. However, these methods are known to generate false positives. To handle this issue, several works like Max-Margin Hough Transform (MMHT) or Implicit Shape Kernel (ISK) have reported significant performance improvements by adding discriminative parameters to the generative ones introduced by the Implicit Shape Model (ISM). In this paper, we propose to use only discriminative parameters that are globally optimized according to all the variables of the Hough transform. To this end, we abstract the common vote process of all Hough methods into linear equations, leading to a training formulation that can be solved using linear programming solvers.
Our new Hough Transform significantly outperforms the previous ones on HoneyBee and TUM datasets, two public databases of action and behaviour segmentation.

Keywords: Hough Transform, Learning, Action Segmentation

1 Introduction

The Hough Transform has first been introduced to detect lines in picture. The main idea of this method is to perform the detection not directly in the picture space but in the line parameter space (Hough space) where each line in the image is mapped into a single point. This method has subsequently been extended to parametric objects [1], and non-parametric objects [9] (eg. car, pedestrian, sport activities, ...). For non parametric objects, the Hough Transform first learns a probabilistic-like parametrization of the objects on a training database, and, then performs the detections as a local problem in the corresponding Hough space.

Due to this property of local detection, Hough Transform is a very fast process both in theory (time complexity theory) and practice. For this reason, it has been applied in context of real-time system like [6] for skeleton extraction

ICIAP #23

and more generally in multiple computer vision fields like tracking [5], object detection [4], human action detection [18], image segmentation [10] or human action segmentation [19].

In the context of temporal signals segmentation and recognition, the Hough Transform is composed of three steps:

1: Feature extraction and quantization to form codewords
2: Each codeword votes for each time (in a large neighborhood) and each label according to a specific learned weight
3: All the votes are agglomerated to form the Hough score from which segmentation decisions are taken

More formally, the Hough Transform (step 2-3) is based on a function $\theta()$ that links codewords, time displacements (quantified into a finite set) and labels to vote weights. Thus, a codeword w extracted at time t votes with a weight $\theta\left(w, l, \Delta_t\right)$ for the hypothesis that a label l is present at time $t+\Delta_t$ (this weight does not depend on the time t but only on the relative time displacement Δ_t). Hence, given a set of localized codewords $W = \{w, t\}$, the Hough score $\mathcal{H}$ for the label l at the time $\bar{t}$ is:

$$\mathcal{H}\left(\bar{t}, l\right) = \sum_{(w,t)\in W} \theta\left(w, l, \bar{t}-t\right) \tag{1}$$

and, the decision about the label (in $\mathcal{L}$) at time $\bar{t}$ is given by:

$$\widehat{l}\left(\bar{t}\right) = \max_{l\in\mathcal{L}} \left(\mathcal{H}\left(\bar{t}, l\right)\right) \tag{2}$$

Hence, all the purpose of the training is to select values for $\theta\left(w, l, \Delta_t\right)$ that will provide correct decisions when following the equations (1) (2) at testing time. Several works, recalled in section 2, propose to improve the generative votes used by the Implicit Shape Model (ISM method) by introducing a partial discriminative optimization process during the vote estimation step. In section 3, we propose to extend these methods by optimizing globally all the votes in a discriminative way. With this new learning process, our Hough method significantly outperforms previous ones on two public datasets of signal segmentation (the Honeybee dataset [12] and the TUM dataset [15]) as reported in section 4, before the conclusion in section 5.

2 State of the Art

In this section, we present the different published methods to select the vote weight during the training step of Hough Transform.

2.1 Implicit Shape Model

In the ISM [9], the Hough Transform (the set of $\theta\left(\right)$-values) is based on generative weights. Let $\mathcal{P}\left(l, \Delta_t | w\right)$ be the probability that the label at time $t+\Delta_t$

is l, knowing that a codeword w has been extracted at time t. This probability is estimated with statistics on the training dataset and is supposed to be independent of t (it just depends on l, Δ_t and w). Then, the weights are given by:

$$\theta_{ISM}\left(w, l, \Delta_t\right) = \mathcal{P}\left(l, \Delta_t | w\right) \tag{3}$$

In practice, the probability $\mathcal{P}\left(l, \Delta_t | w\right)$ is estimated by:

$$\mathcal{P}\left(l, \Delta_t | w\right) \approx \frac{N\left(l, \Delta_t, w\right)}{N\left(w\right)} \tag{4}$$

where $N\left(l, \Delta_t, w\right)$ is the number of times a label l has been seen with a displacement Δ_t from a codeword w and $N\left(w\right)$ is the number of occurrences of the codeword w.

These ISM-based weights have several advantages (eg. parameter-free training, robustness to over-training), but they suffer from several drawbacks. In particular, all codewords and training examples have the same importance and are considered independently from each other. Two methods, MMHT [11] and ISK [20] have been introduced to solve these drawbacks.

2.2 Max-Margin Hough Transform

In MMHT [11], a coefficient is introduced for each codeword to weight the ISM values, resulting in:

$$\theta_{MMHT}\left(w, l, \Delta_t\right) = \lambda_w \times \theta_{ISM}\left(w, l, \Delta_t\right) = \lambda_w \times \mathcal{P}\left(l, \Delta_t | w\right) \tag{5}$$

The weights λ_w give more or less importance to the different codewords w according to their discriminative power. They are learnt simultaneously in a discriminative way through an optimisation process similar to support vector machine (SVM) training [3].

2.3 Implicite Shape Kernel

In ISK [20], the votes are also based on the ISM generative ones, but some coefficients are introduced to weight the different training examples. Hence, ISK training leads to:

$$\theta_{ISK}\left(w, l, \Delta_t\right) = \sum_i \lambda_i \times \mathcal{P}_i\left(l, \Delta_t | w\right) \tag{6}$$

where $\mathcal{P}_i\left(l, \Delta_t | w\right)$ is an estimation of the probability $\mathcal{P}\left(l, \Delta_t | w\right)$ based only on the training example i. The weights λ_i are learnt simultaneously in a discriminative way using a specific kernel-SVM training [20].

MMHT and ISK report experimental improvements over ISM by adding discriminative parameters. This trend is also supported by [17] (we call this method Scaled Implicit Shape Model SISM).

2.4 Scaled Implicit Shape Model

This method [17] is also based on ISM but introduces a weighting coefficient for each displacement, resulting in:

$$\theta_{SISM}\left(w, l, \Delta_t\right) = \lambda_{\Delta_t} \times \mathcal{P}\left(l, \Delta_t|w\right) = \lambda_{\Delta_t} \times \theta_{ISM}\left(w, l, \Delta_t\right) \tag{7}$$

As in [11, 20], the weights λ_{Δ_t} are learnt simultaneously in discriminative way through a SVM training.

2.5 Hough forest

To our knowledge ISM [9] and the presented extensions [11, 17, 20] are the only published methods to estimate the weights of Hough Transform. More precisely, these methods define links between codewords and votes. There are, of course, various ways to select the features and the codewords, like, the Hough forest methods which are major methods of the state of the art. Hough forests use ISM votes, but the mapping between features (usually data patches) and codewords (a leaf in a weak binary classifier tree) is constructed such that all training features associated with a same codeword are expected to come from training examples with a same label. Several works, like [4], report that this automatic feature mapping process associated with ISM votes leads to significant experimental improvements against codewords obtained without learning, by K-means for example.

However, in this paper, we focus on the optimisation of the weights used during the vote process and so to the link between codewords and votes which is generic whatever the features and codewords used. Thus, the proposed method can be employed in the Hough forest context by substituting the weights estimated by ISM by the weights optimized by our proposed method.

The common point between MMHT, ISK and SISM is that they add discriminative parameters to the generative ones introduced by the ISM. In this paper, we propose to use only discriminative votes strongly optimized. We call this method Deeply Optimized Hough Transform (DOHT).

3 Deeply Optimized Hough Transform

The goal of the training process is to establish a correspondence between codewords and weights. While the ISM methods only use generative weights, MMHT, ISK and SISM introduce discriminative parameters optimized according to codewords, training examples or displacements. Using these methods that optimize only one parameter of $\theta\left(w, l, \Delta_t\right)$, a small number of coefficients λ have to be determined. So the optimization process can be solved using SVM. We propose in this paper to optimize all the weights in a global way, according to all the parameters of $\theta\left(w, l, \Delta_t\right)$ in multi-class context. In this way, we do not use ISM values and the method becomes deeply discriminative. The problem is that the number of unknown parameters is more important and their optimisation using

SVM becomes intractable. So, we propose to reformulate the problem such that it becomes linear according to the unknown coefficients.

The goal is to define a function $\theta()$ such that for all training examples (whose set is denoted $\mathcal{T}$) and all times $\bar{t}$, the predicted label $\widehat{l}$ is the real one l^* (known on the training data).

Considering the definition of the predicted label $\widehat{l}$ (eq. (2)), our problem formulation is equivalent to :

$$\forall \bar{t}, l \neq l^*\left(\bar{t}\right), \mathcal{H}\left(\bar{t}, l\right) < \mathcal{H}\left(\bar{t}, l^*\left(\bar{t}\right)\right) \tag{8}$$

by dividing $\theta()$ by the minimal gap, this is equivalent to

$$\forall \bar{t}, l \neq l^*\left(\bar{t}\right), \mathcal{H}\left(\bar{t}, l\right) + 1 \leq \mathcal{H}\left(\bar{t}, l^*\left(\bar{t}\right)\right) \tag{9}$$

and, using equation 1,

$$\forall \bar{t}, l \neq l^*\left(\bar{t}\right), \left(\sum_{(w,t)\in W} \theta\left(w, l, \bar{t}-t\right)\right) + 1 \leq \left(\sum_{(w,t)\in W} \theta\left(w, l^*\left(\bar{t}\right), \bar{t}-t\right)\right) \tag{10}$$

Hence, the constraints on the function $\theta()$ are naturally linear. As in [3], to manage noisy training data, a soft margin framework is applied. For this purpose, some variables ξ are introduced leading to:

$$\forall \bar{t}, l \neq l^*\left(\bar{t}\right), \sum_{(w,t)\in W} \theta\left(w, l, \bar{t}-t\right) + 1 - \xi\left(\bar{t}\right) \leq \sum_{(w,t)\in W} \theta\left(w, l^*\left(\bar{t}\right), \bar{t}-t\right) \tag{11}$$

with the objective function: $\min\limits_{\theta \geq 0, \xi \geq 0} \left(\sum\limits_{\bar{t}} \xi\left(\bar{t}\right)\right)$.

To prevent over-fitting, a regularity term in added to the objective function as in [13]. It penalizes the gap between $\theta()$ and the uniform votes (0 here). A coefficient Υ regulates the trade off between the attachment to data and the regularity as in [3, 13]. In addition, as $\theta(w, l, \Delta_t)$ and $\theta(w, l, \Delta_t + \delta)$ should be close for a small δ and for all w and l, we regularly quantify all possible Δ_t values.

Finally, the problem to solve is formulated as:

$$\begin{gathered} \min_{\theta \geq 0, \xi \geq 0} \left(\sum_{(w,l,\Delta_t)} \theta(w, l, \Delta_t) + \Upsilon \sum_{\bar{t}} \xi\left(\bar{t}\right)\right) \\ \text{under constraints: } \forall W \in \mathcal{T}, \bar{t}, l \in \mathcal{L} \backslash \left\{l^*\left(\bar{t}\right)\right\}, \\ \sum_{(w,t)\in W} \left(\theta\left(w, l^*\left(\bar{t}\right), \bar{t}-t\right) - \theta\left(w, l, \bar{t}-t\right)\right) + \xi\left(\bar{t}\right) \geq 1 \end{gathered} \tag{12}$$

As previously stated, a significant difference between MMHT,ISK or SISM and DOHT is that we optimize simultaneously all values $\theta\left(w, l, \Delta_t\right)$ of the theta function, and not only some variables in order to improve the θ_{ISM} function.

Hence, our set of variables is indexed by codewords w (as in HHMT), displacements Δ_t (as in SISM) and also by labels l as we consider a multi-class context and not only a binary context (MMHT, ISK, SISM). These differences are summarized in table 1. An other difference between our method and MMHT, SISM or SVM is that the penalization of the gap between *θ()* and 0 is measured in L_1-norm and not in L_2-norm. The L_1-norm allows to obtain linear equations and so, to solve the problem efficiency (for example using the solver CPLEX[3] available freely for academic purpose) as it is a linear program which is a well studied problem in literature (eg. [8]).

methods	θ	variables
ISM [9]	$\theta_{ISM}(w,l,\Delta_t) = \mathcal{P}(l,\Delta_t\|w)$	-
MMHT [11]	$\theta_{MMHT}(w,l,\Delta_t) = \lambda_w \times \mathcal{P}(l,\Delta_t\|w)$	λ_w
ISK [20]	$\theta_{ISK}(w,l,\Delta_t) = \sum_i (\lambda_i \times \mathcal{P}_i(l,d\|w))$	λ_i
SISM [17]	$\theta_{SISM}(w,l,\Delta_t) = \lambda_{\Delta_t} \times \mathcal{P}(l,\Delta_t\|w)$	λ_{Δ_t}
DOHT (our)	$\theta_{DOHT}(w,l,\Delta_t) = \lambda_{w,l,\Delta_t}$	λ_{w,l,Δ_t}

$P(l,\Delta_t|w)$ is the probability that the label at time $t+\Delta_t$ is l knowing that a codeword w has been extracted at time t. $P_i(l,\Delta_t|w)$ is the same probability estimated using only the training example i.

Table 1. The different learning methods of the Hough Transform

In the next section, we evaluate the different methods (ISM, HHMT, SISM, DOHT) in action segmentation or behavior segmentation contexts. As ISK is only adapted to detection and can not be straightforwardly extended to segmentation, we can not compare it to the others methods.

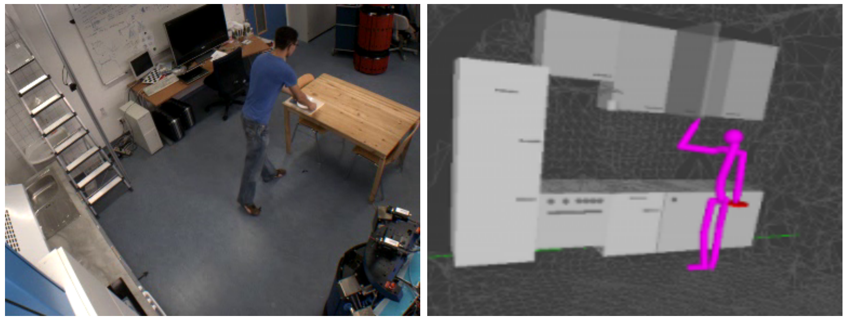
4 Experimental Results

Experiments have been conducted on the TUM [15] and Honeybee [12] datasets. These datasets are well designed for segmentation as each frame (here, frames and times are equivalent) is associated with a label.

4.1 Application to Human Action Segmentation

TUM is a multi-sensor dataset and in particular it contains skeleton streams (fig. 1). It is composed of 19 sequences around 2 minutes each containing 9 kinds of actions (each action is a label) like *Lowering an object*, *Opening a drawer* performed by 5 peoples. To provide results comparable to [19], the same experimental protocol is applied for splitting data between training and testing set, and, results are given in terms of accuracy (number of correctly labelled frames divided by the total number of frames).

[3] www-01.ibm.com/software/websphere/products/optimization/academic-initiative/



example of action: *Lowering an object* Provided skeleton

Fig. 1. TUM dataset [15]

As [19] reports better performances using skeleton features (than visual or visual plus skeleton ones), we decide to consider only skeleton based features. Hence, the input signal of our algorithm is the 3D positions of each articulation at each time.

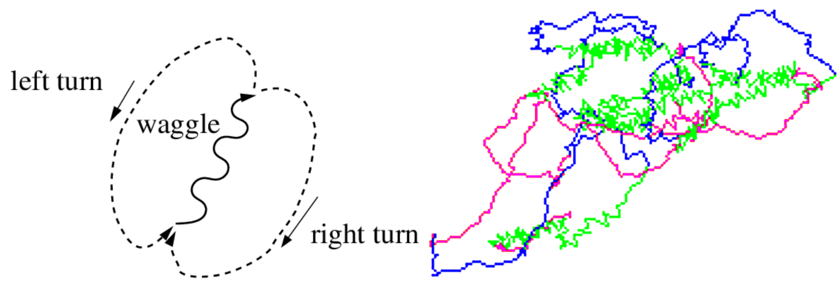
We use the same preprocessing (features and codewords) than the bag-of-gestures from [2] which achieves the best published performance on this dataset (with a manual segmentation). First, the positions are normalized (positions are expressed in a system of coordinate linked to the subject to be invariant to camera viewpoint, global body position, rotation and size). Then, we consider short temporal series of 3D positions of each articulation as features: let the vector $(p_1, ..., p_T)$ be the normalized trajectory of one articulation, then, we consider the vector $(p_{t-\tau}, ..., p_{t+\tau})$ as a feature extracted at time t. Similar features are also considered in [14, 19, 16] which report the efficiency of interest points trajectories for human action recognition. Finally, all these features are clustered by K-means. The cluster centers defines the codebook and features are mapped to their nearest codeword.

More precisely, we consider the 8 main articulations: feet, hands, knees, elbows with $\tau = 6$. The quantization with K-means is performed independently for each articulation with $K = 10$, resulting in 80 codewords. The few parameters of this experiments (τ, K, Υ and the quantification granularity of Δ_t for the optimization process (see section 3)) empirically provide the best performances. Results of this experiment are presented in table 2.

In this experiment, DOHT significantly outperforms ISM, MMHT and SISM and achieves equivalent performance than a SVM based on the same features and codeword applied on the optimal segmentation (obtained from the ground truth) from [2]. Hence, for this dataset, we achieve equivalent performance than the best published (82.6% against 84.3%) without using the optimal segmentation.

4.2 Application to Behaviour Segmentation

Experiments have also been conducted on the Honeybee dataset [12]. The Honeybee dataset provides tracking output of honey bees having 3 kinds of behaviours (each behaviour corresponds to a label) correlated with their trajectories (figure 2). It composed of 6 large sequences. To provide results comparable to [12], the same leave-one-out cross validation is applied. A global measure is obtained by averaging accuracy from all runs.



(a) theoretic behaviour (b) tracking output

Green correspond to waggle, magenta to right turn and blue to left turn.

Fig. 2. Honeybee dataset [12]

The input signals in this dataset are the sequences of bee 2D positions and orientations (x_t, y_t, α_t). As in the previous experiment, normalized short temporal series of (2D here) positions are considered as features. Let us call $R(\beta)$ the matrix of the 2D rotation of angle $-\beta$ and $p(t) = (x_t, y_t)$, then we consider the vector $(R(\alpha_t)(p_{t-\tau} - p_t), ..., R(\alpha_t)(p_{t+\tau} - p_t))$ as the feature extracted at time t. All these features are clustered using K-means. The cluster centers defines the codebook and features are mapped to their nearest codeword.

More precisely, short series of size $\tau = 0, 3, 6$ are considered in this experiment. K-means is performed independently for each τ with $K = 16, 32, 64$ respectively, resulting in 112 codewords. The few parameters of this experiments empirically provide the best performances. Results of this experiment are presented in table 2.

In this experiment, DOHT significantly outperforms ISM, MMHT and SISM.

In addition, DOHT achieves equivalent performances than the best published results [7]. In [7], a multi-class SVM is applied on each temporal windows (with similar kind of features and codewords). Then, segmentation is computed using dynamic programming. As scores are computed on each temporal windows, this method is **quadratic** in the maximal length of an activity while our is **linear**. This quadratic property is a common drawback caused by performing scoring as a global problem. Hence, for this dataset, we achieve equivalent performances

(86.5% against 89.3%) than the best published results while being significantly faster.

Method	Accuracy on TUM	Accuracy mean on Honeybee
ISM [9]	58.4	71.9
MMHT [11]	69.6	78.8
SISM [17]	68.5	77.5
DOHT (our)	**82.6**	**86.5**

Table 2. Global results on TUM [15] and Honeybee [12]

5 Conclusion

In this paper, we propose to use Hough transform to segment and recognize temporal series. In a non parametric context, the training of Hough transform consists to properly select the weights used in the voting process. The simple way (Implicit Shape Model) consists in computing some probabilities on the training database, leading to a generative model. Some methods (Max-Margin Hough Transform, Implicit Shape Kernel) propose to add some parameters optimized on a training database in a discriminative way. In this article, we propose to skip the first step based on a generative model and to globally learn all the parameters of the Hough transform on the training database, resulting a deeply discriminative model. This required to reformulate the voting process to express it in a linear form in order to use linear programming solvers.

We performed several experiments on public datasets where the Hough Transform trained with our method significantly outperforms other Hough Transform methods and provides equivalent results than best published results for these datasets while being significantly faster than the corresponding algorithms.

In future works, we will evaluate our method on other contexts eg. object segmentation in image, video spatio-temporal segmentation, automatic speech segmentation, sign language segmentation.

References

1. D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 1981.
2. Adrien Chan-Hon-Tong, Nicolas Ballas, Catherine Achard, Bertrand Delezoide, Laurent Lucat, Patrick Sayd, and Françoise Prêteux. Skeleton point trajectories for human daily activity recognition. In *Proceedings of International Conference on Computer Vision Theory and Application*, 2013.
3. C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 1995.

ICIAP #23

4. J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1022–1029. IEEE, 2009.
5. J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
6. R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *IEEE International Conference on Computer Vision*. IEEE, 2011.
7. M. Hoai, Z.Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011.
8. N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984.
9. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision*, 2004.
10. Sreedevi M and Jeno Paul P. An efficient image segmentation using hough transformation. In *Asian Journal of Information Technology*, 2011.
11. S. Maji and J. Malik. Object detection using a max-margin hough transform. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
12. S.M. Oh, J.M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 2008.
13. J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 1998.
14. J. Sun, X. Wu, S. Yan, L.F. Cheong, T.S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
15. M. Tenorth, J. Bandouch, and M. Beetz. The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *IEEE 12th International Conference on Computer Vision Workshops*. IEEE, 2009.
16. Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, United States, 2011.
17. Paul Wohlhart, Samuel Schulter, Martin Kostinger, Peter Roth, and Horst Bischof. Discriminative hough forests for object detection. In *BMVC*, 2012.
18. A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
19. Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation? In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011.
20. Yimeng Zhang and Tsuhan Chen. Implicit shape kernel for discriminative learning of the hough transform detector. In *BMVC*, 2010.