A Hough Transform-Based Voting Framework for Action Recognition

Angela Yao[1] Juergen Gall[1] Luc Van Gool[1,2]
[1]ETH Zurich, Switzerland [2]KU Leuven, Belgium
{yaoa, gall}@vision.ee.ethz.ch vangool@esat.kuleuven.be

Abstract

We present a method to classify and localize human actions in video using a Hough transform voting framework. Random trees are trained to learn a mapping between densely-sampled feature patches and their corresponding votes in a spatio-temporal-action Hough space. The leaves of the trees form a discriminative multi-class codebook that share features between the action classes and vote for action centers in a probabilistic manner. Using low-level features such as gradients and optical flow, we demonstrate that Hough-voting can achieve state-of-the-art performance on several datasets covering a wide range of action-recognition scenarios.

1. Introduction

Recognizing human actions in unconstrained video has garnered growing interest in the computer vision community due to its potential for a multitude of applications. Early works in action recognition focused on classifying video sequences of single persons in controlled environments with simple and uniform backgrounds [32, 3]. Recent works have tried to introduce more natural and unconstrained videos, such as sequences from feature films [14], broadcast sports [20] and YouTube [18]. In addition to classification, action recognition systems are also trying to address localization, or estimating the spatio-temporal boundaries of a given action [22, 23, 21, 35, 30]. To this end, the action detection problem, i.e. the localization and recognition of the action, can be considered a form of object detection with higher dimensionality. Recent works have therefore extended object recognition and localization approaches such as interest-point detectors and bag-of-words models into the spatio-temporal domain [8, 18, 14, 22].

Interest-points, though discriminative, have the drawback of being sparse and as such, they are not robust to non-idealities such as low resolution, motion blur and camera movement more typical of uncontrolled videos. Indeed, Wang et al. showed in [34] that dense sampling outperforms several types of interest-point detectors. Furthermore, the classical bag-of-words model assumes independence between the spatio-temporal "words" and does not make use of the rich spatio-temporal relationships inherent in actions. While some works have relaxed this independence assumption [14, 22], they fail to exploit the information in a more global context within the video sequence.

Inspired by the success of Hough transform-based methods in object detection [2, 15, 16, 19, 26, 11, 25], we propose a new action detection system based on Hough transform voting. The use of spatio-temporal voting frameworks for action detection has so far been largely unaddressed, mostly due to the high dimensionality of the problem. A direct extension to the spatio-temporal domain would result in a Hough space of at least six dimensions, with votes being cast for location, scale, time and duration, as well as the action itself. If scale and location of the person changes over time during an action, $3 + 3 \cdot T$ parameters need to be estimated in this space, namely 3 for the temporal boundaries and action class, and $3 \cdot T$ for the location and scale in each frame, where T is the duration of the action.

We approach the high dimensionality problem by separating the voting into two stages and hence two lower-dimensional spaces. In an initial spatial localization stage, we apply a Hough forest trained for people detection [11] and generate detection hypotheses for each frame of the video sequence independently (Fig. 1*(a)*). Due to the strong correlation of the detections' location and scale across a sequence, the votes can be assembled across time by a particle filter into tracks (Fig. 1*(b)*). For instance, a parallelepiped structure would result for a translation. The detection tracks are then mapped to a cuboid representation, which we call "action tracks", to obtain a location and scale invariant representation of the person in time (Fig. 1*(c)*). In a subsequent classification and temporal localization stage, votes are then cast for the action's label and spatio-temporal center on the normalized action track (Fig. 1*(d)*, 2).

As far as we know, this is the first time that Hough-voting has been used for action recognition. We perform the voting with a collection of random trees, termed a Hough forest [11], and learn a mapping between densely sampled spatio-temporal feature patches and an action center. Each

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

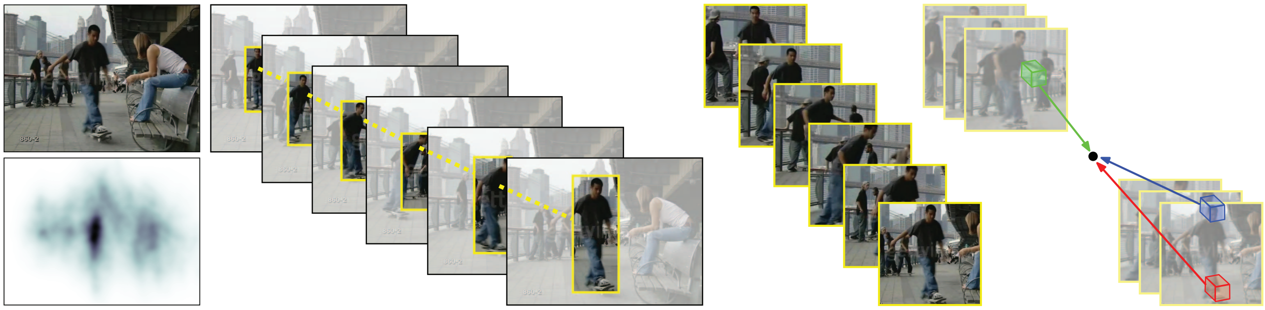


Figure 1. *(a)* **Detection hypotheses** are generated for each frame independently. *(b)* **Particle filtering** is used to link hypotheses across frames. *(c)* **Action tracks**. *(d)* **Hough voting** by 3D patches for action label and center.

tree in the Hough forest is trained to discriminate multiple action classes simultaneously. After training, the resulting set of leaf nodes in the trees can be considered a discriminative codebook with shared features across classes, since each leaf can vote probabilistically for all classes. This type of feature-sharing in such a voting framework is novel to our action recognition system and as of yet, has not been demonstrated even in related object detection systems.

For action recognition, our Hough-transform based framework uses randomized tree structures, which have several appealing properties:

1. Randomized trees allow for the use of dense features during training and testing, which yields advantages over the use of sparse features [34] and is very useful e.g. in surveillance with its low image resolutions.
2. Instead of learning a separate classifier for each action, we train a single classifier and benefit from the sharing of features between classes as demonstrated in [33].
3. The trees are directly optimized for the Hough-transform, in the sense that the leaves cast probabilistic votes with small uncertainty for the label and the spatio-temporal location of the action.

In our experiments, we demonstrate and evaluate the use of Hough-voting for action recognition on four datasets covering a wide range of scenarios. We compare both the classification and localization performance of our system against other methods and demonstrate results either comparable or better than the state-of-the-art.

2. Related Works

Early human action recognition used shape or template matching [4] or multiple limb tracking [27]. Such approaches require high-level representations of the human body which are difficult to extract reliably from real-world videos. To avoid these limitations, others shifted towards the use of local, low-level features such as Gabor filter responses [12, 31] and optical flow [10]. More recently, spatio-temporal interest points have become popular, e.g. cuboids [8], 3D Harris corners [13], 3D Hessians [35] and 3D salient points [28]. Most of these are extensions of their 2D counterparts used in object detection, and many methods follow a traditional object detection approach. After detecting the interest points at multiple scales, feature descriptors are computed, clustered, and assigned to a code-book to be used in some bag-of-words representation [14, 22, 8, 18]. The temporal segmentation of sequences based on the action was addressed in [23]. A set of single-class code-books were employed as person detectors and the pattern of temporal activation of the codewords is registered. The temporal segmentation is then performed by a sliding window approach.

The use of trees and forests for action recognition has been previously explored. In [17], a shape-motion prototype tree is built from shape-motion descriptors; in [21], a vocabulary forest is constructed with local static and flow features, while in [29] a sphere/rectangle tree is built with spatio-temporal interest point features. All three works use trees as indexing structures for performing efficient nearest-neighbour searches in either a protype space [17] or in a feature space in the bag-of-words context [21, 29]. Actions are classified by weighting the n-nearest neighbours and localized either from a foreground segmentation [17] or from the features' spatial information either stored during the training stage [21] or extracted at the test stage [29]. Our approach differs fundamentally in that we are not building trees to speed up a bag-of-words approach nor nearest-neighbour search. Instead, we use randomized trees to learn the mapping between a 3D video patch and its vote in a 4D Hough space to obtain the class label and the spatio-temporal location of an action in the sense of a generalized Hough transform.

3. Voting Framework for Action Recognition

To introduce the concept of using a Hough transform-based voting framework for human action recognition, we first assume that the test sequence is already normalized as shown in Fig. 1*(c)*, i.e. the sequence is a 3D cuboid. This concept is then generalized in Section 3.3.

3.1. Training

For training, we assume that for each action class $c \in C$ a set of training sequences is available. Each training example is annotated such that it can be transformed into a normalized action track and contains roughly one action cycle, i.e. it is annotated by a 2D bounding box for each frame, the action label and the temporal boundaries of the action cycle. To learn the mapping between action tracks and a Hough space, we use the Hough forest structure [11] due to its superior efficiency compared to codebook approaches. Since Hough forests were previously developed for 2D single-class object detection, we extend the idea to multi-class detection and the spatio-temporal domain.

Each tree T in the Hough forest $\mathcal{T} = \{T_t\}$ is constructed from a set of patches $\{\mathcal{P}_i = (\mathcal{I}_i, c_i, \boldsymbol{d}_i)\}$ where,

$\mathcal{P}_i$ is a 3D patch (e.g. of 16x16x5 pixels) sampled from the action track as illustrated by the colored cuboids in Fig. 1*(d)*

$\mathcal{I}_i$ are extracted features at a patch and can be multi-channeled to accommodate multiple features, i.e. $\mathcal{I}_i = \left(I_i^1, I_i^2, ..., I_i^F\right) \in \mathbb{R}^4$, where each I_i^f is feature channel f at patch i and F is the total number of feature channels. In our system, we used six low-level feature channels: greyscale intensity, the absolute value of x, y and $time$ derivatives, and the absolute value of optical flow [6] in x and y.

c_i is the action label ($c_i \in C$)

$\boldsymbol{d}_i$ is a 3D displacement vector from the patch center to the action track center. Since the patches are being drawn from the spatially normalized action tracks, the center displacements are spatially scale-invariant but not temporally. At run-time, however, the Hough forests can be applied at multiple scales to achieve temporal scale-invariance.

Each leaf node L stores p_c^L, the proportion of patches per class label reaching the leaf after training, i.e. $\sum_c p_c^L = 1$, and $D_c^L = \{\boldsymbol{d}_i\}_{c_i=c}$, the patches' respective displacement vectors. Each non-leaf node B of a tree is assigned a binary test in relation to the patch appearance $\mathcal{I}$ during training. There are many possibilities in defining the binary test; we use a simple comparison of two pixels at locations $\boldsymbol{p} \in \mathbb{R}^3$ and $\boldsymbol{q} \in \mathbb{R}^3$ in feature channel f with some offset τ. The binary test at node B, t_B, can be defined as

$$t_{B,f,\boldsymbol{p},\boldsymbol{q},\tau}\left(\mathcal{I}\right) = \begin{cases} 0 & \text{if } I^f\left(\boldsymbol{p}\right) < I^f\left(\boldsymbol{q}\right) + \tau \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

The random trees in Hough forests are constructed according to a standard random forest framework [5]. Construction begins at the root by choosing a binary test, splitting the training patches according to the test results and then constructing children nodes. At each subsequent child node, the same procedure continues recursively, with each node being designated as a non-leaf node until the termination criteria is met, i.e. the child node is of a maximum depth, or there are less than a minimum number of patches remaining. Upon termination as a leaf, the remaining patches' information, $\left(p_c^L, D_c^L\right)_{c \in C}$, is stored; otherwise, another binary test is chosen and the patches are split again. Since the patches from all classes $c \in C$ that pass the binary tests and arrive at a certain leaf share the same features, the probabilities p_c^L represent the degree of sharing between the action classes (see Fig. 4*(b)* for an example).

The ideal binary test should split the patches in such a way as to minimize the uncertainty of their class label and center offsets. To do this, we developed two measures to evaluate the uncertainty for a set of patches $A = \{\mathcal{P}_i = (\mathcal{I}_i, c_i, \boldsymbol{d}_i)\}$. The first measure aims to minimize class uncertainty:

$$U_1\left(A\right) = -\left|A\right| \cdot \sum_c p_c \ln\left(p_c\right), \quad (2)$$

where $|A|$ is the number of patches in set A and p_c is the proportion of patches with label c in set A. Note that the summation expression is the standard definition of entropy for the class labels. The second measure aims to minimize center offset uncertainty:

$$U_2\left(A\right) = \sum_i \left\| \boldsymbol{d}_i - \overline{\boldsymbol{d}_A} \right\| 2, \quad (3)$$

where $\overline{\boldsymbol{d}_A} \in \mathbb{R}^3$ is the mean offset vector of set A. Note that the offset uncertainty is minimized for all classes at the same time.

At each node during training, a pool of binary tests $\left\{t^k\right\}$ is generated with random values of f, $\boldsymbol{p}$, $\boldsymbol{q}$ and τ falling within the constraints of the training data. Then, either class or offset uncertainty is chosen at random to be minimized at that given node. The set of patches arriving at the node will be evaluated with all binary tests in the pool and the binary test satisfying the following minimization objective will be chosen:

$$\underset{k}{\operatorname{argmin}} \left(U_\star\left(\left\{\mathcal{P}_i \left| t^k = 0 \right.\right\}\right) + U_\star\left(\left\{\mathcal{P}_i \left| t^k = 1 \right.\right\}\right)\right), \quad (4)$$

where $\star$ indicates the chosen uncertainty measure for the node. By randomly selecting the uncertainty measure, nodes decreasing both class and offset uncertainty are interleaved throughout the tree. As such, patches being stored at the leaves tend to have low variation in both class label and center displacement and vote with low uncertainty into the Hough-space.

3.2. Classifying and Localizing Actions

To classify and localize an action within an action track S, extracted patches are passed through each of the trees in the random forest; the leaves that the patches arrive in are then used to cast votes for the action label and the spatio-temporal center. To begin with, consider a patch $\mathcal{P}(\boldsymbol{y}) = (\mathcal{I}(\boldsymbol{y}), c(\boldsymbol{y}), \boldsymbol{d}(c(\boldsymbol{y}), \boldsymbol{y}))$ located at position $\boldsymbol{y} \in \mathbb{R}^3$ in the action track, where $\mathcal{I}(\boldsymbol{y})$ are the patch's features, $c(\boldsymbol{y})$ the patch's unknown class label and $\boldsymbol{d}(c(\boldsymbol{y}), \boldsymbol{y})$ the displacement of the patch from the unknown action's center. Let $Q_c(\boldsymbol{x})$ be the event of the action track belonging to class label c and centered at $\boldsymbol{x} \in \mathbb{R}^3$. We are interested in finding the conditional probability $p(Q_c(\boldsymbol{x}) \,|\, \mathcal{I}(\boldsymbol{y}))$, which can be decomposed as follows:

$$\begin{aligned} &p\big(Q_c(\boldsymbol{x})|\mathcal{I}(\boldsymbol{y})\big)\\ =&\sum_{l\in C} p\big(Q_c(\boldsymbol{x})|\,c(\boldsymbol{y})=l,\mathcal{I}(\boldsymbol{y})\big)p\big(c(\boldsymbol{y})=l|\,\mathcal{I}(\boldsymbol{y})\big)\\ =&p\big(Q_c(\boldsymbol{x})|\,c(\boldsymbol{y})=c,\mathcal{I}(\boldsymbol{y})\big)p\big(c(\boldsymbol{y})=c|\,\mathcal{I}(\boldsymbol{y})\big)\\ =&p\big(\boldsymbol{d}(c,\boldsymbol{y})\,|\,c(\boldsymbol{y})=c,\mathcal{I}(\boldsymbol{y})\big)p\big(c(\boldsymbol{y})=c|\mathcal{I}(\boldsymbol{y})\big). \end{aligned} \qquad (5)$$

Both factors in (5) can be estimated by passing the patch $\mathcal{P}(\boldsymbol{y})$ through the trees. Suppose that the patch ends up in leaf L of tree T. The first factor can then be approximated as the Parzen-window estimate of D_c^L, the offset vectors belonging to class c, while the second factor can be approximated as p_c^L, the probability of the patch belonging to class c. We can then rewrite Equation (5) for tree T as

$$\begin{aligned} &p\big(Q_c(\boldsymbol{x})\,|\,\mathcal{I}(\boldsymbol{y}), T\big)\\ &= \left(\frac{1}{|D_c^L|}\sum_{\boldsymbol{d}\in D_c^L} G\big((\boldsymbol{y}-\boldsymbol{x})-\boldsymbol{d}\big)\right)\cdot p_c^L \end{aligned} \qquad (6)$$

where G is the 3D Gaussian Parzen window function. For the entire forest $\mathcal{T}$, we average over all the trees

$$p\big(Q_c(\boldsymbol{x})\,|\,\mathcal{I}(\boldsymbol{y}), \mathcal{T}\big) = \frac{1}{|\mathcal{T}|}\sum_t p\left(Q_c(\boldsymbol{x})\,|\,\mathcal{I}(\boldsymbol{y}), T_t\right). \quad (7)$$

Equations (6) and (7) define the probabilistic vote of a single patch $\mathcal{P}$ for action class c. Votes from all patches are integrated into a 4D Hough accumulator:

$$V(\boldsymbol{x}, c) = \sum_{\boldsymbol{y}\in S(\boldsymbol{x})} p\big(Q_c(\boldsymbol{x})\,|\,\mathcal{I}(\boldsymbol{y}), \mathcal{T}\big). \qquad (8)$$

Note that the action classes are treated independently in the Hough accumulator while the 3D votes for each class are smoothed by a Gaussian window (6).

Since the action track has already been localized in space, the Hough accumulator is marginalized across the spatial dimensions into a 2D image in class label and time. Note, however, that we vote in the spatial dimensions despite having spatially localized tracks in order to enforce spatial regularity of the patches during training, i.e. grouping together the patches which vote towards the actions's 3D center. The local maxima in the remaining Hough accumulator are used to determine the track's label and localization in time as illustrated in Fig. 2.

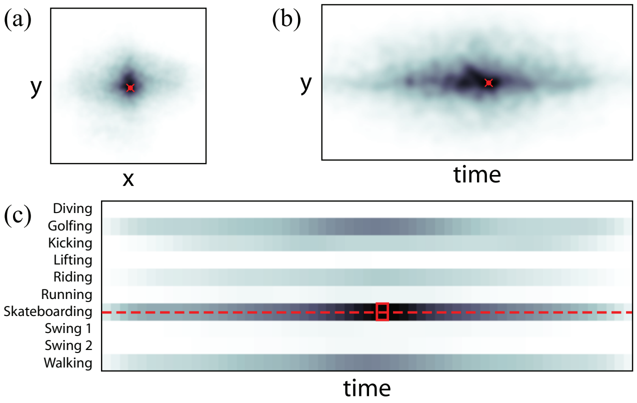


Figure 2. Hough voting space of skateboarding example in Fig. 1 in *(a)* (x, y) for the skateboarding class, *(b)* $(y, time)$ for the skateboarding class and *(c)* $(action\ class, time)$

.

To achieve time-scale invariance, the action tracks can in theory be either up- or down-sampled accordingly and the same Hough forest can then be applied to label actions at differing speeds. We note, however, that action speeds (disregarding a variation in frame-rate) typically do not vary more than by a factor of two. Furthermore, the system has some tolerance built in through variation in speed of the training data and in our current work, we found it unnecessary to apply the Hough forest in multiple time scales.

3.3. Building Action Tracks

In order to obtain action tracks from the test sequences, i.e. a cuboid representation normalized by scale and position of the human, we learn the mapping from the image domain to a 3D Hough space that encodes position and scale of a human. We train a Hough forest for the spatial domain similar to Section 3, where we use cropped and scale-normalized images of humans as positive examples and background images as negative examples. As features, we use color and histograms of gradients [7]. The voting is performed for each frame independently and detections are used to initialize an action track as illustrated in Fig. 1.

Since humans show a large variation in pose and appearance, particularly for sport clips (Fig. 4*(c)*), we do not get a correct detection for each frame. The votes, however, can be efficiently assembled into tracks by a particle filter [9] due to the strong correlation of location and scale across a sequence. We model the state $\boldsymbol{s} \in \mathbb{R}^6$ of a human by the position, scales, and velocity of its bounding box. As dynamical model $p(\boldsymbol{s}_t|\,\boldsymbol{s}_{t-1})$, we use a simple Gaussian process and camera motion compensation which is estimated from optical flow. The likelihood for an image I_t is given by

$$p(I_t|\,\boldsymbol{s}_t) = \frac{1}{Z}\exp\Big(-\big(\alpha V_1(\boldsymbol{s}_t) + \beta\sum_f V_2(\boldsymbol{s}_t, f)\big)\Big) \quad (9)$$

with f features extracted from the bounding box. The parameters $\alpha = 40$ and $\beta = 100$ were experimentally determined. The term V_1 measures the response in the Hough space V (Fig. 1*(a)*) for the candidate $\boldsymbol{s}_t$:

$$V_1(\boldsymbol{s}_t) = -\log\Big(\sum_{\boldsymbol{x}\in V} G(\boldsymbol{s}_t - \boldsymbol{x})\Big), \quad (10)$$

i.e. we sum the votes in the neighborhood of $\boldsymbol{s}_t$ weighted by a Gaussian kernel G. Note that the sum is in the range of $[0, 1]$. The term V_2 measures the similarity of the current particle with the initial detected bounding box $\boldsymbol{s}_0$, i.e. the appearance is not updated, using the Bhattacharya distance:

$$V_2(\boldsymbol{s}_t, f) = 1 - \sum_k \sqrt{h_k^f(\boldsymbol{s}_0)h_k^f(\boldsymbol{s}_t)}. \quad (11)$$

As image features, we use the HSV colour space and a local binary pattern operator [24]. The extracted bounding boxes for a track are then normalized into spatial- and scale-invariant cuboids as shown in Fig. 1.

4. Experiments

4.1. Datasets

We evaluated our system on four datasets, covering a variety of action-recognition scenarios. The first two, Weizmann [3] and KTH [32], are popular benchmarks used in action recognition and consist of single persons performing actions in front of static backgrounds. Current state-of-the-art recognition systems have saturated in performance on these two datasets, but we include their evaluation for comparison purposes against other systems. We also evaluate our system on two more challenging datasets: the UCF sports dataset [20], consisting of broadcast sports action videos with a wide range of scenes and viewpoints in unconstrained environments, and the UCR Videoweb Activities Dataset [1], featuring multiple people interacting in surveillance scenarios.

4.2. Evaluation Measures

4.2.1 Classification

We evaluated our system's ability to apply the correct action label to a given video sequence and call this classification. Classification was measured with three variations of training and testing data: (A) training and testing on tracks generated from ground-truth annotations (B) training on tracks from ground truth and testing on automatically extracted tracks and (C) training and testing on automatically extracted tracks. We refer to these as data variations A, B, and C respectively from this point on.

4.2.2 Localization

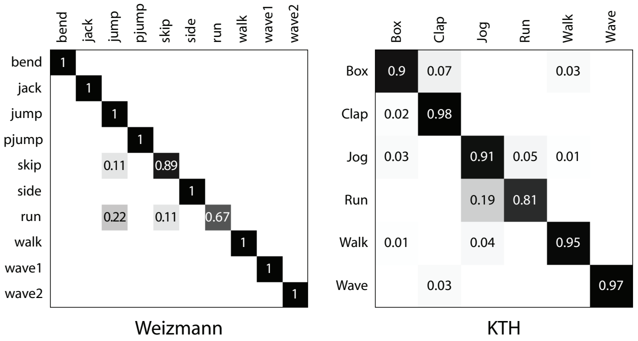
For the KTH and UCF sports dataset, we also evaluate the accuracy of detections in the automatically extracted action tracks and call this localization. The localization evaluation is the same as [21]; a detection is considered correct if (1) the action track that it belongs to was correctly classified and (2) the intersection-union ratio of the detection and ground truth bounding box is greater than 0.5. For the KTH dataset, selected frames were hand-annotated with bounding boxes and the bounding boxes for the frames in between were generated by linear interpolation. For the UCF dataset, bounding boxes were provided as part of the ground truth annotation released with the data. As a measure of localization, we present the average precision, or the area below the precision-recall curve.

5. Results

5.1. Benchmarks: Weizmann and KTH

The Weizmann dataset consists of 90 videos of nine actors performing ten different actions. Evaluations were done with a leave-one-out cross-validation. Eight actors were used for training and the ninth for testing; this was repeated over all nine actors and the results were averaged.

The KTH dataset consists of 599 videos of 25 actors performing six actions under four scenarios. We group the scenarios together as one large dataset rather than treating each scenario separately. Evaluations were done with a five-fold cross-validation. 20 actors were used for training and five for testing; this was repeated five times, such that all actors were used for testing once, and the results were averaged over all folds. As each sequence lasts several hundred frames, we limited each sequence to only one or two cycles of the action in our evaluation.



	bend	jack	jump	pjump	skip	side	run	walk	wave1	wave2
bend	1									
jack		1								
jump			1							
pjump				1						
skip			0.11		0.89					
side						1				
run			0.22		0.11		0.67			
walk								1		
wave1									1	
wave2										1

Weizmann

	Box	Clap	Jog	Run	Walk	Wave
Box	0.9	0.07			0.03	
Clap	0.02	0.98				
Jog	0.03		0.91	0.05	0.01	
Run			0.19	0.81		
Walk	0.01		0.04		0.95	
Wave		0.03				0.97

KTH

Figure 3. Confusion matrices for Weizmann and KTH dataset using ground truth action tracks for training and automatically extracted action tracks for testing.

Classification results over the three variations of training / testing data are shown in Table 1 and compared with state-of-the-art results. In addition, the confusion matrix for training on ground-truth action tracks and classification on automatically extracted action tracks (data variation B) are shown in Fig. 3.

With data variation B, we report an average classification of 95.6% for Weizmann and 92.0% for KTH, both of which are comparable with state-of-the-art action recognition systems. The closest comparison is that of [29], in which a random forest of 50 trees were trained as a comparison against the sphere/rectangle trees; our performance, with only 5 trees in the random forest, is significantly higher and highlights the strength of the Hough-voting framework.

Method	Weizmann	KTH
Hough forest (data A)	97.8%	93.5%
Hough forest (data B)	95.6%	92.0%
Hough forest (data C)	92.2%	93.0%
voc. forest [21]	-	93.2%
SR tree [29]	-	90.3%
random forest [29]	-	72.9%
prototype tree [17]	100%	93.4%
temp. segment [23]	-	81.2%
Niebles et al. [22]	90.0%	83.3%
Schindler et al. [31]	100%	92.7%
Laptev et al. [14]	-	91.8
Liu et al. [18]	-	93.8%

Table 1. Comparison of KTH and Weizmann classification with other methods. Results of all other methods presented are comparable with data variation B, with the exception of Schindler [31], which is comparable with data variation A. See 4.2.1 for definitions of the data variations.

Localization results of each action in the dataset are presented in Table 2; both our method and the vocabulary forest method [21] achieve an average precision of 0.89 over all classes.

Method	Box	Clap	Jog	Run	Walk	Wave
Hough forest	0.88	0.96	0.84	0.72	0.95	0.98
voc. forest [21]	0.98	0.97	0.79	0.78	0.86	0.96

Table 2. Comparison of KTH localization results

5.2. Broadcast Sports: UCF Sports

The UCF sports dataset is a collection of 150 broadcast sports sequences from network news videos and features ten different actions: diving, golfing, kicking, weightlifting, horseback-riding, running, skateboarding, swinging 1 (gymnastics, on the pommel horse and floor), swinging 2 (gymnastics, on the high and uneven bars) and walking. Evaluations were done with a five-fold cross-validation. Due to an unequal number of sequences in each action category, each fold consisted of approximately one-fifth of the total number of sequences per category. Four folds were used for training while the fifth was used for testing; this was repeated five times, such that all clips were used once for testing and results were averaged over all folds.

Classification results over the the three variations of training and testing data are shown in Table 3 and compared with the results reported from other methods. We outperform [20] and [36], and have comparable results with [34] though the we use much simpler features (their best results were achieved using 3D-HOG descriptors).

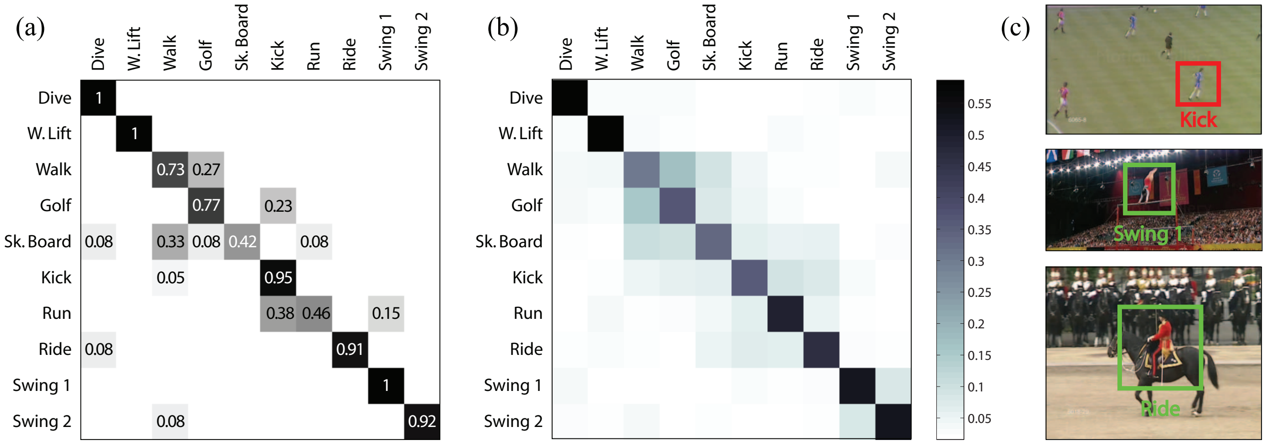
Method	Mean Performance
Hough forest (data A)	86.6%
Hough forest (data B)	81.6%
Hough forest (data C)	79.0%
Rodriguez et al. [20]	69.2%
Yeffet & Wolf [36]	79.2%
Wang et al. [34]	85.6%

Table 3. Comparison of UCF classification with other methods. Results of all other methods presented are comparable with data variation B.

The confusion matrix for training on ground-truth action tracks and classification on automatically extracted action tracks are shown in Fig. 4 along with some example classification results. There is some confusion between running and kicking, since kicking sequences often open with a individual running before kicking a ball. Similarly, walking and golfing sequences are also confused since several walking sequences are drawn from individuals walking on a golf course, suggesting that the trees take some context into account when splitting the patches.

Localization results for the UCF dataset are presented in Table 4; no other works at this time have published a similar evaluation for comparison. Over all classes, we achieve an average precision of 0.54. The low average precision can be attributed to the fact that the ground truth annotations have changing aspect ratios, while we assumed a fixed aspect ratio when generating the action tracks since we are interested only in a cuboid representation. This is particularly relevant for the sports in which the people have irregular and rapidly changing articulations, such as diving, kicking and the swinging classes. Classification performance in these classes, however, are still very high because the cuboid representation is sufficient to capture the action.

To emphasize the effects of feature sharing, we plot the probability of a patch from a given class sharing features with a patch from another class (Fig. 4*(b)*). The diving and weight-lifting classes are very distinct and share little to no features with other actions. On the other hand, the two gymnastics swing classes are very similar to (and only with) each other, and as such, share features with each other.



(a)

	Dive	W. Lift	Walk	Golf	Sk. Board	Kick	Run	Ride	Swing 1	Swing 2
Dive	1									
W. Lift		1								
Walk			0.73	0.27						
Golf				0.77		0.23				
Sk. Board	0.08		0.33	0.08	0.42		0.08			
Kick			0.05			0.95				
Run						0.38	0.46		0.15	
Ride	0.08							0.91		
Swing 1									1	
Swing 2			0.08							0.92

(b)

	Dive	W. Lift	Walk	Golf	Sk. Board	Kick	Run	Ride	Swing 1	Swing 2
Dive										
W. Lift										
Walk										
Golf										
Sk. Board										
Kick										
Run										
Ride										
Swing 1										
Swing 2										

Figure 4. *(a)* Confusion matrix for UCF sports dataset using data variation B. *(b)* Probability of feature sharing in a patch between action classes. *(c)* Example classifications.

Class	Precision	Class	Precision
Dive	0.52	Kick	0.28
W.Lift	1	Run	0.37
Walk	0.67	Ride	0.66
Golf	0.77	Swing 1	0.44
Sk. Board	0.39	Swing 2	0.26

Table 4. UCF localization results

There also exists less distinct groupings, such as walking, golfing, skateboarding and kicking, suggesting that both body position and context are accounted for in the feature sharing. For example, walking, golfing, and skateboarding all involve upright individuals with legs in relatively straight alignment with the body. On the other hand, several walking sequences are drawn from people walking on golf-courses with green fields, which also resemble the soccer fields in the kicking sequences.

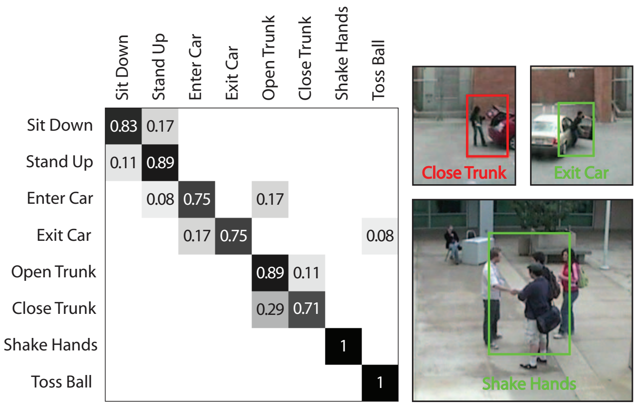
5.3. Surveillance: UCR Videoweb Activities

The UCR Videoweb Activities Dataset consists of about 2.5 hours of video footage from four to eight cameras in various surveillance scenarios. From this footage, we selected 110 sequences of eight actions that not only illustrate changes in body configuration (sitting down, standing up), but also interaction with the environment (entering and exiting a car, opening and closing a trunk), interaction with other people (shaking hands) and interaction with objects (tossing a ball). Evaluations were done with a five-fold cross-validation in the same manner as the UCF sports dataset. As our system handles only monocular views, we treat the same action instance recorded by different cameras as different sequences. As this is a newly released dataset, there are no other works with comparable results that we know of.

We achieve an average performance of 91.5%, 85.2% and 92.6% for variations A, B, and C of training and testing data (see 4.2.1). The confusion matrix for variation B is shown in Fig. 5, together with some example classification results. As expected, there are some confusions between action pairings such as sit down/stand up, enter/exit car and open/close trunk. Performance is remarkable considering the small size of the people in the surveillance sequences (typically 40 to 60 pixels high).

5.4. Dense patch sampling

The use of randomized trees allows for dense sampling of features, which has been shown to outperform sparse features at spatio-temporal interest points [34]. We investigate our system's performance with respect to decreased patch sampling rates on the KTH dataset. In Fig. 6, the average classification performance is plotted with respect to sampling density. Since we are using dense sampling in three dimensions, there is considerable overlap between patches. Performance does not drop until around one per-



	Sit Down	Stand Up	Enter Car	Exit Car	Open Trunk	Close Trunk	Shake Hands	Toss Ball
Sit Down	0.83	0.17						
Stand Up	0.11	0.89						
Enter Car		0.08	0.75		0.17			
Exit Car			0.17	0.75				0.08
Open Trunk					0.89	0.11		
Close Trunk					0.29	0.71		
Shake Hands							1	
Toss Ball								1

Figure 5. Confusion matrix for Videoweb Activities dataset for data variation B and example classifications

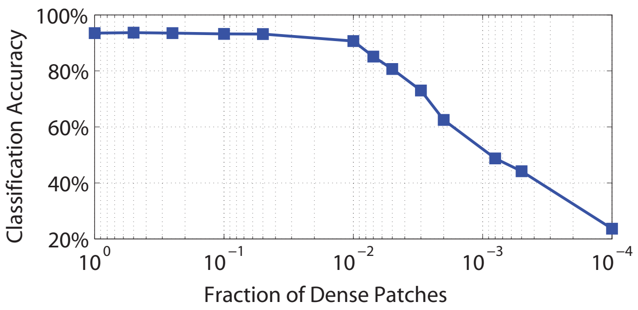


Figure 6. Performance decreases when patch sampling is reduced

cent of the original sampling density but from this point onwards, the performance decrease is smooth. This shows that the amount of data processing can be reduced by a factor up to 100 for time-critical applications.

On 100 frames of the KTH dataset, it takes around 10s to classify pre-existing action tracks and 170s to generate an action track on a standard PC.

6. Conclusion

In this paper, we present a novel method for action classification and localization using a Hough-transform based voting framework. We approach the problem from a object detection perspective, but split the otherwise high-dimensional and intractable problem into two lower-dimensional spaces. Our system can classify and localize actions in unconstrained video sequences, achieving state-of-the-art performance on two benchmarks datasets and also more challenging and realistic video sequences from sports broadcasts and surveillance scenarios. In our current work, our system shows promising performance using only low-level features; with more sophisticated features, we expect performance to improve even more. Future work includes coupling the building of action tracks with their classification and temporal localization simultaneously.

Acknowledgements The authors would like to thank Thomas Brox for sharing his code for optical flow. This research has been supported by funding from the Swiss National Foundation NCCR project IM2. Angela Yao was also supported by funding from NSERC Canada.

References

[1] Ucr videoweb activities dataset. http://vwdata.ee.ucr.edu.

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

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[4] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267, 2001.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[9] A. Doucet, N. D. Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[10] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

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

[12] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[13] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[14] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[15] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008.

[16] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *CVPR*, 2008.

[17] Z. Lin, Z. Jian, and L.S.Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[18] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos 'in the wild'. In *CVPR09*, 2009.

[19] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009.

[20] M.D.Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[21] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *CVPR*, 2008.

[22] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.

[23] A. Oikonomopoulos, I. Patras, and M. Pantic. An implicit spatiotemporal shape model for human activity localization and recognition. In *CVPR4HB09*, 2009.

[24] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[25] B. Ommer and J. Malik. Multi-scale object detection by clustering lines. In *ICCV*, 2009.

[26] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 80(1), 2008.

[27] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *IJCV*, 50(2):203–226, 2002.

[28] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In *CVPR*, 2009.

[29] K. K. Reddy, J. Liu, and M. Shah. Incremental action recognition using feature-tree. In *ICCV*, 2009.

[30] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[31] K. Schindler and L. J. V. Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008.

[32] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR04*, 2004.

[33] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *TPAMI*, 29(5):854–869, 2007.

[34] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[35] G. Willems, J. Becker, T. Tuytelaars, and L. V. Gool. Exemplar-based action recognition in video. In *BMVC*, 2009.

[36] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, 2009.