# Transition Hough Forest for Trajectory-based Action Recognition

Guillermo Garcia-Hernando[†] Hyung Jin Chang[†] Ismael Serrano[‡] Oscar Deniz[‡] Tae-Kyun Kim[†]
[†]Imperial College London [‡]University of Castilla-La Mancha
{ggarciah, hj.chang, tk.kim}@imperial.ac.uk {ismael.serrano, oscar.deniz}@uclm.es

## Abstract

*In this paper, we propose a new discriminative framework based on Hough forests that enables us to efficiently recognize and localize sequential data in the form of spatio-temporal trajectories. Contrary to traditional decision forest-based methods where predictions are made independently of its output temporal context, we introduce the concept of "transition", which enforces the temporal coherence of estimations and further enhances the discrimination between action classes. We start applying our proposed framework to the problem of recognizing and localizing fingertip written trajectories in mid-air using an egocentric camera. To this purpose, we present a new challenging dataset that allows us to evaluate and compare our method with previous approaches. Finally, we apply our framework to general human action recognition using local spatio-temporal trajectories obtaining comparable to state-of-the-art performance on a public benchmark.*

## 1. Introduction

A human action can be seen as an ensemble of spatio-temporal trajectories that describe human motion. Trajectories can have different levels of abstraction (see Figure 1): from low-level trajectories describing local motion of parts of a human body to high-level trajectories such as handwritten characters that have a meaning by themselves. However, different kind of trajectories have a common and important property: they are time-structured patterns.

With the recent introduction of wearable cameras a new chapter in computer vision called egocentric vision has emerged where the user is the center of the action. A distinctive characteristic of this new paradigm relative to the classic third-person vision is that hands are very present in the scene [6, 14]. As these wearable sensors lack a keyboard, an interesting way to communicate with them would involve using our hands. A natural way of doing this would consist in writing with our fingertip in front of the camera. We can think of the fingertip motion as a spatio-temporal trajectory in mid-air which can represent, for instance, a handwritten character in the English alphabet (see Figure 2). This could lead to many different new applications in the domains of human-computer interaction or augmented reality. If we are able to recognize the fingertip written trajectories in mid-air we can use them as text input for the device. If we are able to not only recognizing them but localize them, we could, for instance, write notes in a virtual blackboard. Note that these two applications need real-time performance. Motivated by the aforementioned challenges, we address the problem of recognizing and localizing fingertip written characters in mid-air. We propose a new framework based on Hough forests [8] that we evaluate presenting the first public dataset of fingertip written characters in mid-air recorded with an egocentric sensor.

Figure 1. Two examples of spatio-temporal trajectories. On the left, trajectories representing human motion. On the right, a character 'a' written in the mid-air using the fingertip and an egocentric sensor.

Decision forests-based methods have been very successful and popular in many computer vision tasks because their efficiency both in training and testing, their inherently multi-class handling ability and their capacity to handle overfitting. Their efficiency in prediction comes with the cost of assuming independence in the output variables, which is not always the case for sequential data. In the interest of enforcing temporal coherence in our forest framework, we introduce the concept of *transition*. We define a transition as the probability of observing the current output of the forest taking into account previous observations. Thus, our current prediction will consider what has been previously observed. Based on Hough forests, our method inherits the benefits of a decision forest model while enforc-

ing the temporal coherence of predictions. Finally, we show that our framework formulation is general enough to deal with different types of sequential data proving its suitability for general human action recognition.

In summary, our main contributions are three-fold:

- A new general framework based on Hough forests which can simultaneously recognize and localize spatio-temporal trajectories.
- Introduction of temporal context in a decision forest-based classifier in the form of transitions.
- The first public dataset containing fingertip written characters in mid-air in egocentric viewpoint.

## 2. Related Work

**Decision forests for structured prediction**: There has been some preliminary work using decision forest methods for spatio-temporal data modeling in diverse applications. Spatio-temporal relational probability trees [18] were proposed for weather process understanding, however the nature of their data is very different from ours. More related to our work, some approaches used tree-based methods for human action recognition [20, 35, 8, 10]. [20] proposed simultaneous action recognition and localization using local motion-appearance features method and clustering trees. [35] also used codebooks for building spatio-temporal histograms and matching them using histogram intersections and a SVM classifier. On the other hand, decision forests methods have been also used for directly mapping spatio-temporal features to space-time location and class label. In [8], dense spatio-temporal cuboids were extracted and each of them voted independently for a hypothesis in space, time and class in the Hough space. [10] proposed a spatio-temporal forest for detecting the action of finger clicking from an egocentric viewpoint, but they only considered one simple action with not much temporal structure. These approaches rely on the premise that spatio-temporal structure is adequately embedded in the feature level. In practice, noisy and incoherent labels are observed mainly caused by the output independence assumption. This is a general problem in structured prediction using decision forests and some authors have proposed solutions in other computer vision areas such as semantic image segmentation [21, 29, 30, 23, 12]. [21, 29] exploited the hierarchical nature of the trees in order to cluster similar samples and extract context information. [30, 23] used graphical models in top of decision forest predictions, while [12] proposed directly modelling the context within the forest in order to have smooth pixel-wise labellings. [3] introduced temporal context in a decision forest framework by warping map confidences using optical flow for body pose estimation.

Figure 3. Dense cuboid patches in Hough forest (orange) and trajectory patches in Transition Hough forest (blue) in two different scenes of punching and kicking from UT-interaction dataset [27].

**Fingertip writing in mid-air:** Recognizing fingertip written trajectories in mid-air has been previously explored in the last decades highly depending on the available technology and mainly from a third person viewpoint [24, 1, 28, 7, 31]. From an egocentric point of view the problem remains quite unexplored; however there are some early approaches related to our application [16, 11, 9]. [16] with the help of a wearable computer recognized fingertip trajectories using a spline-based matching algorithm. [11] and [9] recorded fingertip writing gestures with a webcam pointing a desktop and used DTW-based classifiers, but no real-time performance was achieved and localization was not performed.

**Trajectories for human action recognition:** Trajectory-based methods [17, 19, 32, 33, 34] for human action recognition have been very popular in the last years mainly to its good results in a variety of datasets. [19] extracted trajectories using an interesting point detector and proposed a graphical model to model the velocities of those trajectories. [17] extracted trajectories using a KLT tracker and clustered them in a bag of words fashion [22]. [32] densely sampled the and tracked trajectories extracting local descriptors such as HOGHOF [13] and MBH [5] along them. Densely sampling trajectories leads to the problem of capturing non meaningful trajectories; a problem that can be attenuated modelling the camera motion [33] or automatically learning the feature representation [34]. After extracting trajectory-sampled features, [32, 33, 34] use a bag of words model [22] losing important structural information.

## 3. Overview of the method

### 3.1. Hough forests for spatio-temporal trajectories

A spatio-temporal trajectory is a set of time-ordered space tracked points $P_t = (x_t, y_t)$ where $t$ is the frame number. A trajectory can be written as $(P_t, P_{t+1}, ..., P_{t+L-1})$ where $L$ is the length, in frames, of the trajectory. We store our trajectory data in the form of patches $\{\mathcal{P}_i = (f(P_i), c_i, d_i)\}$ where $f(P_i)$ are appearance

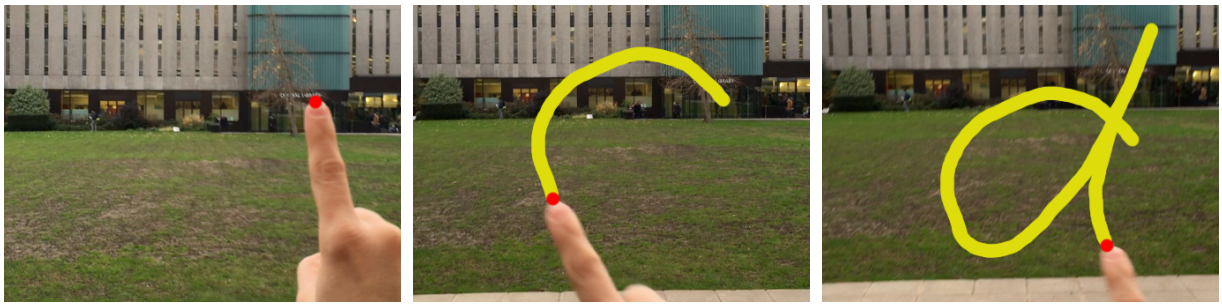

Figure 2. An example of capturing a spatio-temporal trajectory representing the character 'd' described by fingertip motion.

and motion features for a given point of a trajectory, $c_i$ is the class label and $d_i$ is a vector pointing to the spatio-temporal center of the trajectory in the fingertip writing problem and to the spatio-temporal center of the action in the human action recognition problem.

We formulate the spatio-temporal trajectory recognition as a multi-class classification problem and action center localization as regression. To perform them simultaneously, we build upon Hough forest [8]. An important difference between [8] framework and ours is that we extract patches by sampling along trajectories instead of dense sampling cuboids (see Figure 3).

A Hough Forest is an ensemble of decision trees that, in addition to classification, they also perform regression. Each tree in the forest is constructed from a set of patches extracted along the trajectories $\{\mathcal{P}_i\}$. Tree training starts at the root and input data is divided and rooted left or right following a split function. Split candidates are generated randomly and the best split is chosen based on an objective function that is minimized. This objective function is randomly chosen between Shannon entropy, which minimizes class uncertainty, and variance of displacement vectors, a regression measure that tends to group similar vectors. If the current node reaches a certain depth or a good split cannot be found, it becomes a leaf node. At each leaf node $l_i$, a class histogram $p(c|l)$ is estimated by the proportion of trajectory patches per class that reached that node. Both histogram and displacement vectors $d_i$ are stored.

During inference, patches are passed through each tree in the trained forest. Starting at the root of the forest the patch traverses the tree, branching left or right according to the split node function, until reaching a leaf node. Using the stored class distribution and vector displacements at the leaf nodes, each leaf node votes for its corresponding class label and spatio-temporal center location. Each patch votes in a 4D Hough space and the most likely hypothesis can be found searching the maxima. We refer the reader to [8] for further details.

### 3.2. Transition Hough forest

A major drawback of using a decision forest based classifier for sequential data is that the forest produces each estimation independently of its temporal context. This assumption can be too strong in the problem of recognizing spatio-temporal trajectories. For instance, the human action of punching involves the movement of an arm in a particular direction and speed. This movement follows a certain temporal order that makes it different from similar actions such as pulling.

As presented in the previous section, a Hough Forest reduces both class and displacement uncertainty throughout the tree. The leaf nodes will contain similar patches both in displacement, feature-space and class, thus it can be seen as a clusters of similar patches. Such idea of using a decision forest framework for clustering is not new and it has been explored in other areas such as semantic image segmentation (*e.g.* [21, 29]), but relatively less for action recognition [35]. From this perspective, we can see a spatio-temporal trajectory as a time-indexed sequence of codebook values.

Based on this, we introduce our concept of *transition*. Our hypothesis is that different classes of spatio-temporal trajectories will have a different temporal dynamics within the forest. For example, if we observe that in a given frame $t$ the trajectory patch $P_t$ has reached the node $i$ while in the previous frame $t-1$ the corresponding patch $P_{t-1}$ reached the node $j$, we can quantify how *likely* is the transition from node $j$ to node $i$ or, more formally, $p(n(t)=i|n(t-1)=j)$ a certain class. We name this last term as *transition probability* borrowed from HMM literature [25]. We define our transitions for one time step, thus we will ignore the time index in the following sections. [29] showed that adding non-terminal nodes while constructing codebooks captured the hierarchical structure of the tree leading to a better performance. Accordingly, we consider transitions between both leaf and split nodes. Although in practice trees are not balanced and transitions can be observed between different levels of the tree, we ignore them maintaining its hierarchical nature considering only same level transitions. In order to compact this information, we define a transi-

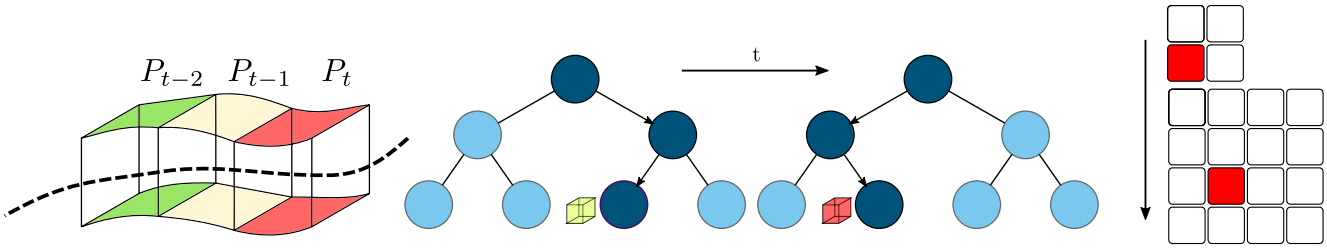


Figure 4. Process to build the transition matrix. At current frame $t$ we feed the forest with the current trajectory patch $P_t$ of class $c$. We compare the path through the forest (tree by tree) followed by $P_t$ with the one followed by $P_{t-1}$. We show the transition matrix $A(c, l)$ for the first two levels of the tree (we do not count the root level). As there are two nodes on the first level and four on the second level, $A(c, 1)$ will be a 2x2 matrix and $A(c, 2)$ a 4x4. In this example, $P_{t-1}$ reached the 2nd node on the first level and the 3rd node on the second level. $P_t$ reached the 1st node on the first level and the 2nd one on the second level. Thus, we increase the transition probability from 2nd node to 1st on the transition matrix at the first level and the transition probability from 3rd node to 2nd on the transition matrix at the second level.

tion matrix $A(c, l)$ that encode all transitions between nodes for a given class $c$ and level $l$ in one time step. Rows of $A(c, l)$ encode transition probabilities from node $i$ to all the rest of the nodes $j$ in a particular level $l$ of the tree and they are normalized defining a probability distribution ($\sum_j p(n = j|n = i) = 1, n \in l$). See Figure 4 for further details. Note that we will have a transition matrix for every tree in the forest, however we omitted this to make the notation clear.

In order to incorporate this temporal information into our predictions, we treat this transition probability as a prior probability $p(c)$ in a similar way to [29]. We want transitions to emphasize classes that are likely in a temporal context and reject unlikely ones. Given two temporal consecutive patches from a trajectory, $P_t$ and $P_{t-1}$, we pass both patches through the forest and each of them reaches different nodes through each tree of the forest. We ponder the prediction for $P_t$, $p(c|l_i)$, with the (almost independent) prior probability:

$$p'(c|l_i) = p(c|l_i)p(c) \tag{1}$$

with $p(c)$ defined as the averaged transition probability $p_{trans}$, of all $T$ trees in the forest soften by a constant $\alpha$:

$$p(c) = \frac{1}{T}\sum_{k=0}^{T} p^{\alpha}_{trans}(c, k) \tag{2}$$

$p_{trans}$ is calculated from our transition matrices defined above:

$$p_{trans}(c, k) = \frac{1}{W}\sum_{l=0}^{D_k} A^k(c, l) \tag{3}$$

where $D_k$ is the maximum level reached in the $k$-th tree by $P_{t-1}$ or $P_t$ and $W$ is a factor that ensures probability normalization. $D_k$ is not necessarily the total depth of the tree $D$ as leaf nodes can be found at any level of the tree.

### 3.3. Implementation details

#### 3.3.1 Fingertip writing trajectories

We extract fingertip writing spatio-temporal trajectories by tracking the index fingertip in space and time. Instead of standard RGB video, we decided to use a depth sensor. Using depth data makes the problems of hand segmentation and fingertip detection easier. We detect and track the fingertip using the approach of [15] where hand contour is obtained and fingertips are tracked using a distance transform and a particle filter respectively. Once obtained the spatio-temporal trajectories, we extract local features that we encode in $f(P_i)$. These features are extracted using a temporal sliding window of $n_\tau$ frames. This parameter defines the length of strokes encoded in patches. Small values of $n_\tau$ may not allow us to properly capture motion, while large ones could give us non meaningful information. At each temporal window, we concatenate the following local features: displacement vector between points, curvature, distance and velocity. Distance and velocity are defined only between the first and the last point in the window and we considered both euclidean distance and geodesic distance. Their temporal derivatives provide us complementary information about the writing stroke. Note that for these trajectories $L$ does not have a fixed size and it will depend on their category.

#### 3.3.2 Action recognition trajectories

For tracking and extracting features along spatio-temporal trajectories on video data, we follow the approach from [33]. We chose this method mainly because its excellent results and its publicly available code, however other spatio-temporal trajectory representation would be also valid. In [33], each trajectory point is tracked at different scales using optical flow. Tracked points are sampled in small volumes of size $n_\sigma \times n_\sigma \times n_\tau$ and rich feature descriptors, HOGHOF [13] and MBH [5], are extracted. We encode all

this information in our trajectory patches $f(P_i)$. In contrast to [33], we do not concatenate the descriptors of all the points in the trajectory nor we average them. Instead, we treat each point of the trajectory independently and we store it as a patch. Our trajectories are defined as ensembles of $L$ independent patches.

## 4. New dataset: Egocentric fingertip writing

As there exists no public dataset of fingertip written trajectories in mid-air using an egocentric sensor, we recorded our own one, which we plan to make public for further research. Our dataset is composed of depth video sequences containing fingertip written trajectories that represent the 26 English alphabet characters (from 'a' to 'z'). We attached a depth sensor (Creative* Interactive Gesture Camera) to a cap in order to be able to record gestures in egocentric viewpoint. In total, 10 sequences of 26 different characters performed by a single actor have been recorded (making a total of 260). Furthermore, we fully annotated the sequences with $(x, y, t)$ fingertip positions after our detection and tracking stage to help research on this direction as well. See Figure 5 for some examples of our recorded sequences and Table 1 for further details.

| | | | |
|---|---|---|---|
| Classes | 26 | Total frames | 15792 |
| Clips | 260 | Clips per class | 10 |
| Mean clip frames | 60.74 | Resolution | 320x240 |
| Min clip frames | 27 | Max clip frames | 154 |

Table 1. Characteristics of the dataset

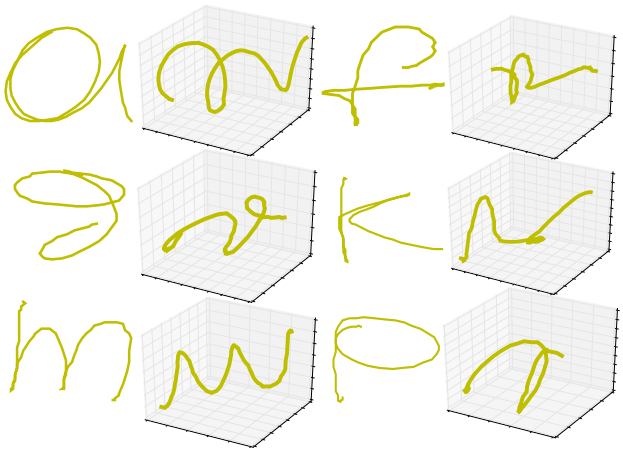

Figure 5. Examples of our new dataset of characters written in mid-air both projected in 2D space and in 3D space-time.

## 5. Experiments

### 5.1. Egocentric fingertip writing

**Character recognition:** In table 2 we present the performance of different methods on our new dataset. All the results have been obtained performing 10 leave-one-out cross validation (234 sequences for training and 26 for testing). We chose empirically a sliding window size of $n_\tau = 7$. We first show the results of two classical algorithms for sequential data recognition, Hidden Markov Model (HMM) [25] and Dynamic Time Warping (DTW) [31, 4]. Although neither of these methods is suitable for our application since they do not perform localization, we include them for completeness of this work. Next, we show the results for decision forest-based classifiers: a conventional decision forest [2], our framework without transition term and the full framework. For all forest based algorithms we fixed $T = 8$ and $D = 25$, optimizing cross validated results. We see that our proposed Transition Hough forest outperform all the other approaches. Introducing the transition term slightly improves the accuracy in a 1.5%. Comparing with the conventional random forest, we note that including localization also helped classification, as it was already pointed in [8].

| Recognition | Method | Accuracy (%) |
|---|---|---|
| Character recognition | HMM [25] | 66.4 |
| | DTW [31] | 78.5 |
| | Decision forest [2] | 79.6 |
| | Trajectory Hough forest | 90.4 |
| | **Transition Hough forest** | **91.9** |

Table 2. Recognition performance of fingertip writing.

From the confusion matrix (Figure 6), we can observe that most errors came from similar characters such as 'a-d', 'm-n', 'g-q' and 'v-w', which are all of them very similar and sometimes difficult to recognize even for humans. We believe that adding a broader temporal context could help on these cases.

**Character center localization:** Our method can also correctly localize spatio-temporal center of each character writing by spatio-temporal offset Hough voting. Figure 7 shows some localization results in spatio-temporal space. As we can see, estimated centers are similar to ground-truth ones. The writing center information of each character can be used as an important clue for segmenting each character in a word or to anchor where the user wrote in an augmented reality scenario.

### 5.2. Action Recognition: UT-Interaction dataset

To demonstrate the effectiveness of our proposed method for human action recognition, we conducted experiments on a public benchmark: UT-interaction dataset [27]. The UT-interaction dataset consists in 6 different classes of human-human interactions in a surveillance scenario: shake-hands,

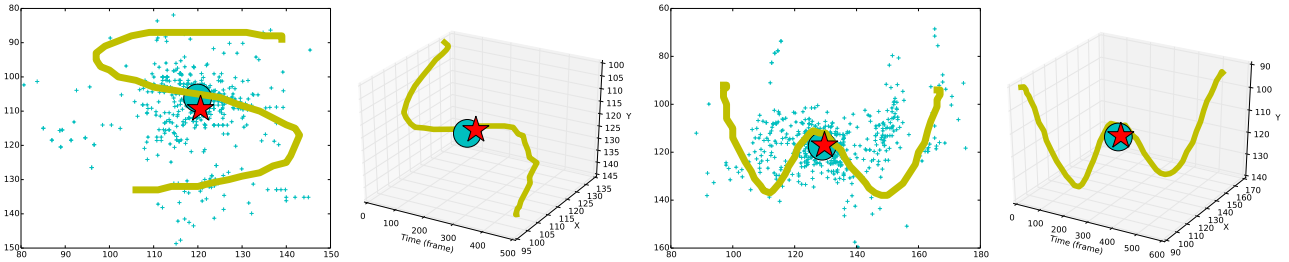

Figure 7. Character center localization results. Small cyan crosses are displacement voting points. Cyan circles are estimated center positions of each character and red stars ground-truth center locations.

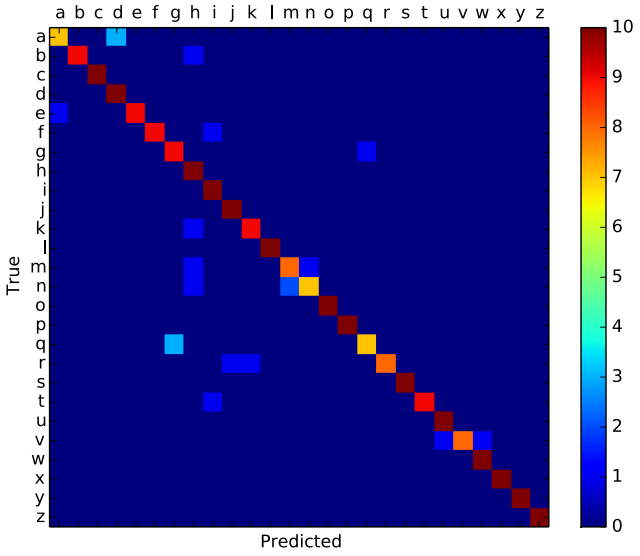


Figure 6. Confusion matrix of character recognition results by our proposed method

point, hug, push, kick and punch (an example of kick and punch is shown in Figure 3). We used the segmented set 1 of the dataset which contains 10 sequences per each class. We followed the methodology recommended by the authors and we performed 10-fold leave-one-out cross validation to find the average performance.

In table 3 we present the performance of our method compared to baseline and other state-of-the-art methods. The parameters for extracting trajectories were the recommended by [33] $n_\sigma = 2$ and $L = 15$. We used $n_\tau = 1$, meaning that a patch was generated for every frame in the trajectory. We defined our baseline as the Hough forest using trajectory-based patches and we also compared to the conventional Hough forest using dense cuboid sampling [8]. Forests parameters are $T = 4$ and $D = 35$.

We observe that using trajectory sampled descriptors instead of dense cuboids slightly improves the recognition accuracy in a 2%, which is in line with what was reported in [32]. In top of that, we show that adding our transition term further improves the performance in a 3.3% from the baseline making it comparable to state-of-the-art performances. Our approach offers a similar performance to [26] even if they use more sophisticated medium-level features encoding the pose of humans in scene, which are usually hard to annotate and obtain. Compared to [36], our method performs not as well as theirs. The main reason for this is that we rely on very local spatio-temporal context while in [36] they also consider long range spatio-temporal relations. Finally, we also show the result from [35] where they also used the clustering capability of a decision forest, however important spatio-temporal information is lost when histogram quantization is performed.

| Method | Accuracy (%) |
|---|---|
| Yu *et al.* [35] | 83.3 |
| Raptis and Sigal [26] | 93.3 |
| Zhang *et al.* [36] | **95.0** |
| Hough forest (cuboids) [8] | 88.0 |
| Trajectory Hough forest | 90.0 |
| **Transition Hough forest** | **93.3** |

Table 3. Overall recognition performance for set 1 of UT-Interaction dataset of our method compared to state-of-the-art.

## 6. Conclusion and Future Work

We have presented a novel framework for recognizing and localizing spatio-temporal trajectories using a Hough forest-based classifier and showed that it is general enough to be applied in different scenarios. We have introduced a new concept of transition that makes forest predictions sensitive to their output temporal context without losing efficiency. As a future work, we plan to investigate how can we make these transitions more discriminative within the forest, enforcing the transitions at feature level or designing a novel split criteria that privileges transitions. Furthermore, we also plan to explore the introduction of long-range temporal context in contrast to only exploiting the context of time-consecutive patches.

## References


[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(9):1685–1699, 2009.

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

[3] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Upper body pose estimation with temporal sequential forests. In *Proceedings of the British Machine Vision Conference 2014*, pages 1–12. BMVA Press, 2014.

[4] F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745–758, 2003.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV 2006*, pages 428–441. Springer, 2006.

[6] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011.

[7] Z. Feng, S. Xu, X. Zhang, L. Jin, Z. Ye, and W. Yang. Real-time fingertip tracking and detection using kinect depth sensor for a new writing-in-the air system. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, ICIMCS '12, 2012.

[8] 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 (TPAMI)*, 33(11):2188–2202, Nov. 2011.

[9] H. Ishida, T. Takahashi, I. Ide, and H. Murase. A hilbert warping method for handwriting gesture recognition. *Pattern Recognition*, 43(8):2799–2806, 2010.

[10] Y. Jang, S.-T. Noh, H. J. Chang, T.-K. Kim, and W. Woo. 3D finger CAPE: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2015.

[11] L. Jin, D. Yang, L.-X. Zhen, and J.-C. Huang. A novel vision-based finger-writing character recognition system. *Journal of Circuits, Systems, and Computers*, 16(03):421–436, 2007.

[12] P. Kontschieder, P. Kohli, J. Shotton, and A. Criminisi. Geof: Geodesic forests for learning coupled predictors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 65–72. IEEE, 2013.

[13] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[14] C. Li and K. M. Kitani. Pixel-level hand detection in egocentric videos. In *CVPR*, 2013.

[15] H. Liang, J. Yuan, and D. Thalmann. 3d fingertip and palm tracking in depth image sequences. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 785–788. ACM, 2012.

[16] Y. Liu, X. Liu, and Y. Jia. Hand-gesture based text input for wearable computers. In *IEEE International Conference on Computer Vision Systems (ICVS)*, pages 8–8. IEEE, 2006.

[17] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 514–521. IEEE, 2009.

[18] A. McGovern, N. Hiers, M. Collier, D. Gagne, and R. Brown. Spatiotemporal relational probability trees: An introduction. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 935–940, 2008.

[19] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 104–111. IEEE, 2009.

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

[21] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Twentieth Annual Conference on Neural Information Processing Systems (NIPS'06)*, pages 985–992. MIT Press, 2007.

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

[23] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1668–1675. IEEE, 2011.

[24] K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *IEEE Computer Graphics and Applications*, 22(6):64–71, 2002.

[25] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[26] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2650–2657. IEEE, 2013.

[27] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[28] A. Schick, D. Morlock, C. Amma, T. Schultz, and R. Stiefelhagen. Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, pages 217–220, New York, NY, USA, 2012. ACM.

[29] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[30] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[31] S. Vikram, L. Li, and S. Russell. Handwriting and gestures in the air, recognizing on the fly. In *Computer Human Interaction (CHI)*, 2013.

[32] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[33] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558. IEEE, 2013.

[34] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4305–4314, 2015.

[35] T.-H. Yu, T.-K. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forests. In *BMVC*, 2010.

[36] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition. In *Computer Vision–ECCV 2012*, pages 707–721. Springer, 2012.