Imperial College
London

UNDERSTANDING EGOCENTRIC HUMAN ACTIONS WITH TEMPORAL DECISION FORESTS

GUILLERMO GARCIA-HERNANDO

Thesis submitted for the degree of Doctor of Philosophy

Department of Electrical and Electronic Engineering
Imperial College London
September 2017

COPYRIGHT DECLARATION

London, September 2017

ABSTRACT

Understanding human actions is a fundamental task in computer vision with a wide range of applications including pervasive health-care, robotics and game control. This thesis focuses on the problem of egocentric action recognition from RGB-D data, wherein the world is viewed through the eyes of the actor whose hands describe the actions.

The main contributions of this work are its findings regarding egocentric actions as described by hands in two application scenarios and a proposal of a new technique that is based on temporal decision forests. The thesis first introduces a novel framework to recognise fingertip writing in mid-air in the context of human-computer interaction. This framework detects whether the user is writing and tracks the fingertip over time to generate spatio-temporal trajectories that are recognised by using a Hough forest variant that encourages temporal consistency in prediction. A problem with using such forest approach for action recognition is that the learning of temporal dynamics is limited to hand-crafted temporal features and temporal regression, which may break the temporal continuity and lead to inconsistent predictions. To overcome this limitation, the thesis proposes transition forests. Besides any temporal information that is encoded in the feature space, the forest automatically learns the temporal dynamics during training, and it is exploited in inference in an online and efficient manner achieving state-of-the-art results. The last contribution of this thesis is its introduction of the first RGB-D benchmark to allow for the study of egocentric hand-object actions with both hand and object pose annotations. This study conducts an extensive evaluation of different baselines, state-of-the art approaches and temporal decision forest models using colour, depth and hand pose features. Furthermore, it extends the transition forest model to incorporate data from different modalities and demonstrates the benefit of using hand pose features to recognise egocentric human actions. The thesis concludes by discussing and analysing the contributions and proposing a few ideas for future work.

DECLARATION OF ORIGINALITY

I, Guillermo Garcia-Hernando, hereby declare that this work is my own, and where it is based on or derived from the work of others, I have acknowledged this and included a reference in the bibliography. Main ideas and some figures have been published in the following listed papers involving the author:

- Chang, H.J.*, **Garcia-Hernando, G.***, Tang, D. and Kim, T.K., 'Spatio-Temporal Hough Forest for efficient detection–localisation–recognition of fingerwriting in egocentric camera', *Computer Vision and Image Understanding*, 148, pp.87-96.
  (* denotes equal contribution)

- **Garcia-Hernando, G.**, Chang, H.J., Serrano, I., Deniz, O. and Kim, T.K., Transition Hough forest for trajectory-based action recognition, in Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, New York, USA, 2016.

- **Garcia-Hernando, G.** and Kim, T.K., Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition and Detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, 2017.

- **Garcia-Hernando, G.**, Yuan, S., Baek, S. and Kim, T.K., First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, Utah, USA, 2018.

ACKNOWLEDGEMENTS

This thesis presents work resulting from multiple years of research which I could not have accomplished without the help and support of many people during my PhD.

I would like to start by thanking my supervisor, Tae-Kyun Kim, for providing me the opportunity to visit his lab and for later accepting me as his student. He also taught me the basics and introduced me to the world of computer vision and machine learning research. His deep yet prompt understanding of research ideas in addition to his vision for approaching a problem have been truly inspiring.

I am also grateful for all of the people with whom I have worked and collaborated, including the co-authors on my published papers as well as current and past members of the Imperial Computer Vision and Learning lab. Their help was truly crucial for parts of this thesis. Research is more fun when shared, and I have gained knowledge from every one of them.

I would also like to thank my friends, including those in London, those all over the world and those on the 10th floor. I have appreciated the good times, the fun, the experiences, and the support throughout these years. To those friends, you know who you are – and if you ever read this, the next drink is on me!

Finally, I would like to thank and dedicate this thesis to my family: mum, dad, brother Biel and sister Glòria. Without your love and support, this adventure would have never been possible.

CONTENTS

LIST OF FIGURES xvi

LIST OF TABLES AND ALGORITHMS xvii

LIST OF ABBREVIATIONS xix

GLOSSARY xxi

1 INTRODUCTION 1

1.1 Thesis outline and contributions 8

2 RELATED WORK 11

2.1 Overview 11

2.2 Action recognition in RGB videos 11

2.2.1 Hand-crafted feature representation methods 12

2.2.2 Learned representation methods 15

2.2.3 RGB action recognition benchmarks 17

2.3 Action recognition in RGB-D videos 18

2.3.1 RGB-D action recognition benchmarks 20

2.4 Action recognition in egocentric viewpoint 22

2.4.1 Egocentric benchmarks 23

2.5 Action recognition and decision forests 24

2.6 Evaluation criteria 26

3 UNDERSTANDING EGOCENTRIC FINGERTIP WRITING IN MID-AIR WITH A TRAJECTORY HOUGH FOREST 27

3.1 Overview 27

3.2 Related work 28

3.3 Proposed framework 32

3.3.1 Handwriting posture detection 32
3.3.2 Fingertip detection 35
3.3.3 Trajectory Hough forest 35
3.4 A dataset for egocentric fingertip writing in mid-air 41
3.5 Experiments 43
3.5.1 Hand posture recognition and fingertip detection 43
3.5.2 Character recognition and localisation 47
3.5.3 An experiment on RGB human action recognition 50
3.6 Summary 53
4 TRANSITION FORESTS 55
4.1 Overview 55
4.2 Related work 56
4.3 Transition forests 57
4.3.1 Learning transition forests 58
4.3.2 Inference 63
4.3.3 Implementation details 64
4.4 Experimental evaluation 65
4.4.1 Baselines 65
4.4.2 Action recognition experiments 66
4.4.3 Online action detection experiments 75
4.5 Summary 79
5 UNDERSTANDING EGOCENTRIC HAND-OBJECT ACTIONS WITH RGB-D VIDEOS AND 3D HAND POSE ANNOTATIONS 81
5.1 Overview 82
5.2 Related work 84
5.3 Daily hand-object actions dataset 87
5.4 Evaluated algorithms and baselines 95
5.4.1 Action recognition 95
5.4.2 Hand pose estimation 96
5.5 Evaluation results 97
5.5.1 Action recognition 97

5.5.2 Hand pose estimation 108
5.6 Summary 111

6 SUMMARY AND FUTURE IDEAS 113
6.1 Overall summary 113
6.2 Future ideas 114

A APPENDIX A: HAND POSE ANNOTATION SYSTEM DETAILS 145

B APPENDIX B: HAND POSE ESTIMATION BASELINE DETAILS 147

LIST OF FIGURES

Figure 1.1 Daily action in third-person viewpoint and in egocentric viewpoint 1
Figure 1.2 Spatio-temporal trajectories on egocentric fingertip writing 2
Figure 1.3 Different actions and data modalities 4
Figure 1.4 'Opening peanut butter jar' action in the proposed benchmark . . 5
Figure 2.1 Cuboid sampling vs. trajectory sampling 13
Figure 2.2 Two-stream approach for action recognition 16
Figure 3.1 Overview of the proposed framework for egocentric fingertip writing 27
Figure 3.2 Hand posture and fingertip detection proposed approach. 34
Figure 3.3 Spatio-temporal feature extraction for character recognition . . . 36
Figure 3.4 Procedure to construct a transition matrix 40
Figure 3.5 Examples of characters written in mid-air 42
Figure 3.6 Examples of images from the writing/no writing dataset 43
Figure 3.7 Distance weighting parameter and number of harmonics 44
Figure 3.8 Proposed framework in action 44
Figure 3.9 Qualitative fingertip detection results 46
Figure 3.10 Forest parameters for proposed trajectory Hough forest 47
Figure 3.11 Confusion matrix for character recognition 49
Figure 3.12 Character localisation qualitative results 52
Figure 4.1 Transition forest concept 58
Figure 4.2 Learning a level of a transition tree 60
Figure 4.3 Transition forest inference 64
Figure 4.4 Temporal order k comparison 66
Figure 4.5 Effect of adding a transition term 67
Figure 4.6 Objective function visualisation 73
Figure 4.7 Qualitative results in online action detection 78
Figure 5.1 Different data modalities on proposed benchmark 81

Figure 5.2 Visual examples of the proposed benchmark 86
Figure 5.3 Hand actions taxonomy 88
Figure 5.4 t-SNE visualisation of hand pose embedding 89
Figure 5.5 Hand model visualisation 90
Figure 5.6 Recording process 91
Figure 5.7 Dataset statistics 92
Figure 5.8 Correlation between objects, grasps and actions 93
Figure 5.9 Benchmark results for different data modalities 99
Figure 5.10 Transition forest model using colour, depth and pose features . . 102
Figure 5.11 Different data modalities action and object retrieval 103
Figure 5.12 Different data modalities behaviour for a subset of classes 105
Figure 5.13 Action confusion matrix on proposed benchmark 106
Figure 5.14 Qualitative examples of confused hand categories 107
Figure 5.15 Hand pose estimation error plot 110
Figure 5.16 Hand occlusion on action recognition and hand pose estimation . 112
Figure 5.17 Qualitative results hand pose estimation 112
Figure A.1 Hand pose annotation system details 146
Figure B.1 CNN baseline for hand pose estimation 148

LIST OF TABLES

Table 3.1 Characteristics of the proposed dataset 41
Table 3.2 Hand posture recognition performance 45
Table 3.3 Fingertip detection and character recognition results 48
Table 3.4 Action recognition results on UT-Interaction dataset 50
Table 4.1 State-of-the-art comparison on MSRC-12 dataset 68
Table 4.2 Comparison with forest-based baselines. 70
Table 4.3 State-of-the-art comparison on MSR-Action3D dataset. 71
Table 4.4 Effect of adding transitions in terms of node purity 72
Table 4.5 State-of-the-art comparison on Florence-3D dataset. 74
Table 4.6 Online action detection results 76
Table 5.1 First-person view datasets with hands and objects involved 94
Table 5.2 Training and test protocols 98
Table 5.3 Hand action recognition results on benchmark 101
Table 5.4 Temporal decision forest approaches on proposed benchmark . . . 103
Table 5.5 Object pose for action recognition evaluation 108
Table 5.6 Hand pose estimation error for different protocols 109

LIST OF ALGORITHMS

Algorithm 4.1 Learning level l of a transition tree 61

LIST OF ABBREVIATIONS

AR	augmented reality
CNN	convolutional neural network
DFM	dynamic forest models (Lehrmann et al., 2014)
DTW	dynamic time warping
FD	Fourier descriptor (Persoon and Fu, 1977)
FTP	Fourier temporal pyramid (Wang, Liu, Wu and Yuan, 2012)
HCI	human-computer interaction
HOF	histogram of optical flow (Laptev et al., 2008)
HOG	histogram of gradients (Dalal and Triggs, 2005)
HMM	hidden Markov model
IDT	improved dense trajectories (Wang and Schmid, 2013)
JOULE	jointly learning heterogeneous features (Hu et al., 2015)
JP	joint positions
LSTM	long-short term memory (Hochreiter and Schmidhuber, 1997)
MBH	motion boundary histograms (Dalal et al., 2006)
MP	moving pose descriptor (Zanfir et al., 2013)
NN	nearest neighbour
PCA	principal component analysis
PCRF	pairwise conditional random forests (Dapogny et al., 2015)
RF	random forest (Breiman, 2001)
RJP	pairwise relative distance of joints (Vemulapalli et al., 2014)
RNN	recurrent neural network
SVM	support vector machine

SW	sliding window forest (Fothergill et al., 2012)
TF	transition forest
THF	trajectory Hough forest

GLOSSARY

$\mathbf{x}$ input feature vector.

t temporal index.

y class label.

$\mathcal{Y}$ class label (output) space.

N number of points in sliding window.

W_t N-points sliding window starting at time step t.

S labelled set of training vectors.

$f(\cdot,\cdot)$ split function.

θ split parameter (node decision).

m a decision tree.

$\mathcal{M}$ set of trees in a forest.

$\ell(\mathbf{x})$ leaf node reached by $\mathbf{x}$.

$H(\cdot)$ Shannon entropy function.

$p(\cdot)$ probability function.

$\mathbf{p}$ trajectory point (2D).

k temporal order of a transition forest.

$\|$ and $[\cdot,\cdot]$ vector concatenation operator.

$||\cdot||^2$ L_2 norm.

$E(\cdot)$ objective function.

$\mathcal{A}$ transition matrix.

1

INTRODUCTION

Figure 1.1 presents frames that have been extracted from a video of someone pouring juice into a plastic glass. Each video has a different point of view, but they both depict the same action with one difference: on the left, we are merely spectators in the action we see, while on the right, we see the action as though we are the ones performing it. We are able to recognise the action that both frames portray with little difficulty by simply looking at the hands and the objects they are manipulating. Furthermore, we would likely be able to repeat the same action, even if we had not poured juice before by looking at her hand, her grasp on the object and the way in which she handles it. This process of understanding the human action – from recognising the performed action being to successfully imitating it – is a task that an intelligent agent should be able to accomplish. This thesis focuses on the recognition of human actions, which is an important and classic problem in computer vision that has a wide range of applications, including pervasive health-care, robotics and video game control.

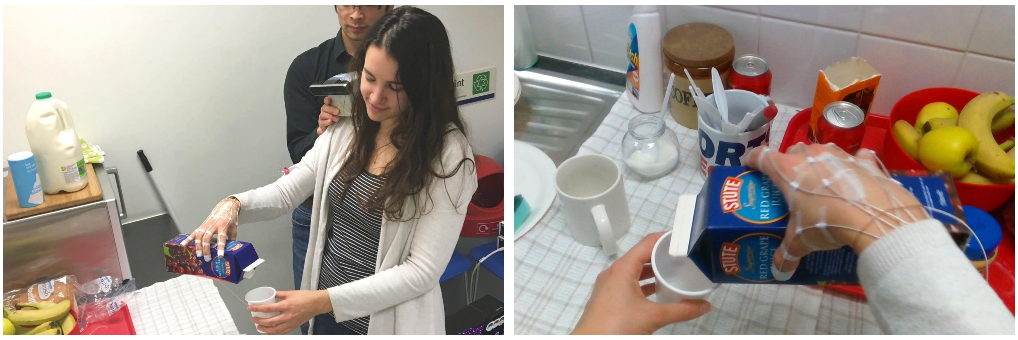

Figure 1.1: Someone pouring juice into a plastic glass from a third-person viewpoint (left) and from an egocentric viewpoint (right); we can recognise the action by looking at the hands and the object that they are manipulating.

Before proceeding to a more detailed discussion, it is necessary to define 'action'. According to Herath et al. (2017),

'Action is the most elementary human-surrounding interaction with a meaning'.

In the context of computer vision, the *meaning* of this interaction is the *class* or *category* of the action that we want to recognise. This work focuses on human actions, although the concept of actions can be broader in nature and extend to other living organisms, such as animals, or even machines and robots. In the context of this thesis, this *interaction* may produce a change in the surroundings when the human manipulates an object, such as in the juice-pouring example above, or it may not produce a change. An example of an action that does not change the surroundings can be found in the context of human-computer interaction (HCI), wherein the human uses his or her hand or body to communicate with a computer. As Figure 1.2 illustrates, writing a character in mid-air involves a certain succession of motion patterns, or *gestures*, that may not have an innate meaning. Since we have defined action as the most elementary and meaningful interaction, we can define the action category as the character that the user wants to communicate to the computer. Other definitions in terms of meaning, duration and complexity are plausible and we refer the interested reader to (Zabulis et al., 2009; Aggarwal and Ryoo, 2011; Herath et al., 2017).

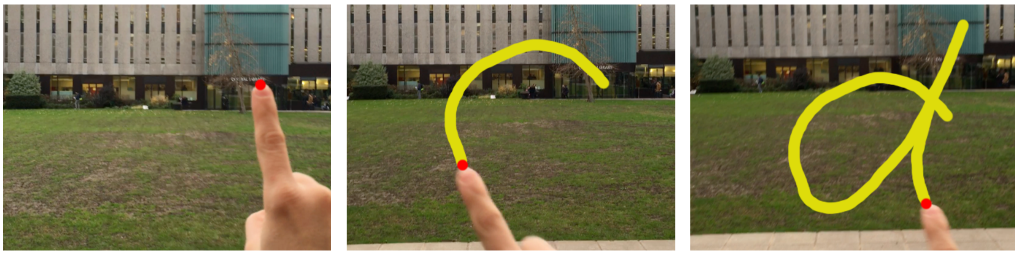

Figure 1.2: 'Writing the character "d" in mid-air': Spatio-temporal trajectories can be recognised from an egocentric viewpoint and can lead to applications in HCI and virtual and augmented reality.

In understanding human actions, different tasks are defined depending on the problem of interest and the assumptions that are made. Following the most common nomenclature in computer vision (Aggarwal and Ryoo, 2011), we refer to *action recognition* as the task of classifying an action with the assumption that the video is spatially and temporally segmented, i.e. the spatial location and temporal bounds in which the action occurs are known. If one relaxes such assumption, the problem is usually referred to as *action detection*, whereby one must determine *where* and *when* the action is occurring (Ke et al., 2005), in addition to identifying the action. For instance, in the frame of the film that Figure 1.3 (a) depicts, an action detection system should be able to localise each subject in the scene, infer each action that is performed and determine in which frames they started and ended. In contrast, in action recognition, the system would receive the trimmed video of each separate subject and would independently infer the action of each video. When action inference is performed after observation of the full video, it is usually called *offline* prediction. By contrast, *online* prediction, does not offer access to future frames to reason about an on-going video, which is a more realistic scenario in real-world applications that require instant predictions of ongoing actions (De Geest et al., 2016). This thesis focuses on action recognition of spatial and temporally localised actions; however, also presents some algorithmic extensions and experiments in the case of online action detection.

Human action recognition presents difficult challenges, such as intra-class variation and across-class similarities. Intra-class variations include different viewpoints, actor styles, aspects and execution speeds. For instance, Figure 1.3 (a) provides one example in which the actor who is reading the newspaper is wearing some ‘unexpected’ glasses, which makes more difficult for the system to determine whether the actor is reading at home or diving at the ocean. Another example of such variations is the difference in writing styles, such as that in Figure 1.2, depending on which user is writing in the system. Across-class similarities occurs when different action categories share similar characteristics, such as motion or objects involved. Figure 1.4 contains one example where a subject is manipulating a jar of peanut butter. Without more temporal context, we cannot determine whether the person is opening or closing the jar, as both action categories are highly similar in appearance and motion.

Understanding actions involves the recognition of complex spatio-temporal patterns as an actor depicts them in a video. In computer vision, this problem usually necessitates the extraction of spatio-temporal *features* that capture the meaning of the action and

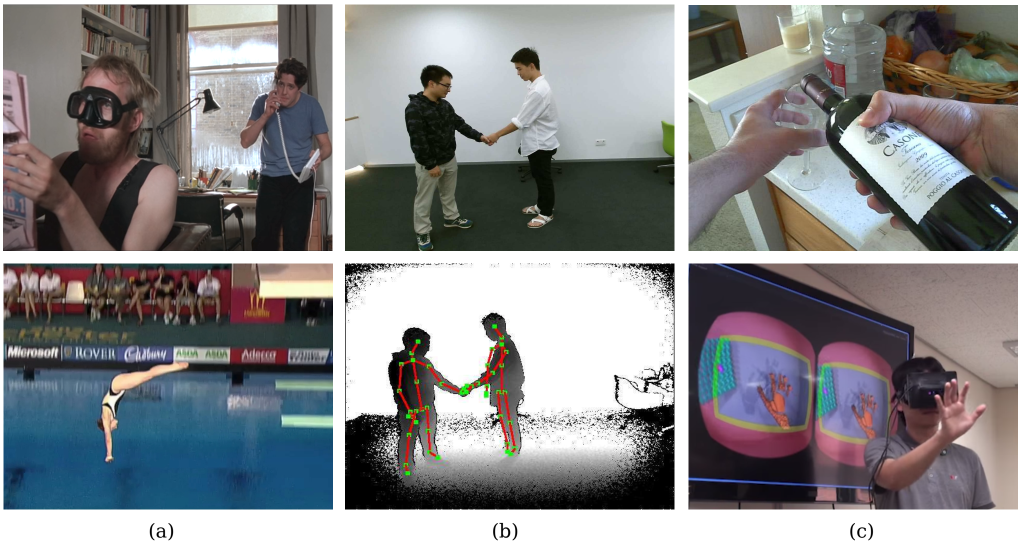

Figure 1.3: Different actions and modalities in popular action recognition datasets. (a) top: a frame from film *Notting Hill* (1999) with two on-going actions of 'reading' and 'talking by phone'; bottom: 'diving' action from popular RGB dataset (Kuehne et al., 2011); (b) third-view RGB-D action recognition and human body pose (Shahroudy, Liu, Ng and Wang, 2016); (c) egocentric viewpoint: daily action of 'pouring wine' (Rogez et al., 2015*b*) and a virtual reality game (Jang et al., 2015)

the use of a machine learning *classifier* to infer this meaning. These features can be manually designed with domain knowledge (Laptev, 2005) or automatically extracted while learning the classifier in an end-to-end fashion (Feichtenhofer et al., 2016). They can range from low-level pixel values to a more meaningful and high-level human body pose, or *skeleton* (Yao et al., 2011), depending highly on the nature of the data and the hardware in use. As for classifiers, one can broadly distinguish between two types. The first directly maps a feature vector to an action category based on the encoding of relevant temporal information in the feature space; this occurs in support vector machines (Simonyan and Zisserman, 2014) and random forests (Fothergill et al., 2012), for example. The second type tries to automatically model temporal dependencies while learning the classifier itself; state-space models (Lehrmann et al., 2014) and recurrent neural networks (Donahue et al., 2015) exemplify this type.

Action recognition (Bobick and Davis, 2001; Efros et al., 2003) has traditionally utilised standard RGB video cameras to recognise actions. The majority of the literature in the field involves RGB videos, which is still a highly active area of research (Carreira

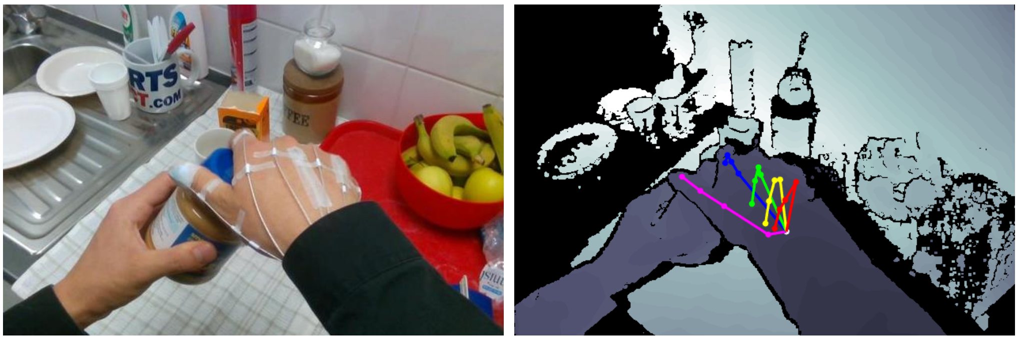

Figure 1.4: 'Opening peanut butter jar' in egocentric viewpoint with a RGB-D camera and hand pose features. Note that we need temporal context, i.e. access to previous or future frames, to differentiate it from 'closing peanut butter jar'.

and Zisserman, 2017), mainly because of its broad range of applications and the ease of obtaining RGB video data in the era of the Internet and smartphones. Successful approaches capture low-level, hand-crafted features from spatio-temporal trajectories (Wang and Schmid, 2013) and with neural networks (Simonyan and Zisserman, 2014). Extracting high-level features from RGB, such as body pose, is still an open problem, although this could change in the near future (Cao et al., 2017). Figure 1.3 (a) displays some typical examples of actions that have been considered in the RGB literature.

The study of RGB-D action recognition has emerged as a result of the irruption in the market of affordable depth sensors by products such as Microsoft Kinect$^{\text{TM}}$. In comparison to RGB video, the addition of a depth channel supports the acquisition of high-level features, such as human body pose (Shotton, Girshick, Fitzgibbon, Sharp, Cook, Finocchio, Moore, Kohli, Criminisi, Kipman et al., 2013), which can be useful for action recognition, and its effectiveness has been clearly demonstrated (Yao et al., 2011). Similar to human body pose, the study of depth has enabled reliable pose estimators for hand pose estimation compared to other low-level representations (Tang et al., 2014). Figure 1.3 (b) presents one example of body pose estimation on a depth image. Given that the depth channel only works in indoors environment RGB-D action recognition has focused primarily on recognising daily-life home actions (Wang, Liu, Wu and Yuan, 2012), gaming (Seidenari et al., 2013) and health-care (Baek, Shi, Kawade and Kim, 2017).

Wearable cameras, such as GoPro®, as well as virtual and augmented reality headsets, such as Oculus® and Hololens™, have been recently introduced. Such technologies have led to a new chapter in computer vision that is termed *egocentric* or *first-person* vision. In contrast to the *third-person* view, the camera in an egocentric viewpoint is not fixed, and one observes the world 'from the eyes' of the camera wearer, illustrated by 'pouring juice' example. Figure 1.3 (c) offers more examples of such viewpoint, wherein the user is the centre of the action and, thus, one does not have access to his or her full body. In such paradigm, the observer is no longer a passive subject and can instead influence the environment. This is the natural point of view of humans and of any intelligent agent, which makes the study of egocentric vision crucial to understanding how humans act and to designing intelligent agents (Stadie et al., 2017). This thesis examines actions performed by humans wearing a RGB-D sensor. In its conclusion, it provides insight into how this knowledge could be applied to train intelligent agents.

Given that humans naturally use their hands when interacting with the world, a distinctive characteristic of an egocentric setting is the clear presence of hands in the scene (Mayol and Murray, 2005; Fathi, Farhadi and Rehg, 2011). Hence, most approaches for recognising actions from the first-person view have concentrated on hands to extract low-level spatio-temporal features (Fathi, Farhadi and Rehg, 2011; Ishihara et al., 2015). The present research studies a variety of actions in two egocentric scenarios and uses two approaches of extracting spatio-temporal features from hands.

The first scenario is one of HCI. Since wearable cameras are usually small and lack a keyboard or similar input accessories, user hand motions can serve as natural and unobtrusive input. Spatio-temporal trajectories that are generated by fingertip movements in mid-air can represent handwritten characters, such as as in Figure 1.2, and subsequent steps can utilise these as text input in the wearable system. Other applications of these spatio-temporal trajectories are possible in, for example, making a virtual blackboard or directing a virtual orchestra. In this scenario, we extract spatio-temporal features from trajectories that represent characters by capturing how the fingertip moves on the scene. However, to successfully capture these features, one must overcome certain challenges, such as identifying whether the user is writing and detecting the fingertip in space and time.

The second scenario consists of daily actions whereby the actor interacts with quotidian objects, such as those in Figures 1.1 and 1.4. Motivated by the success of using body pose representations for third-view RGB-D action recognition, we explore the use of high-level hand pose features to recognise actions. Extracting pose and other meaningful high-level features from hands poses unique challenges compared to from the full body (Tang, 2015) in regard to self-occlusion, size and shape variances, sensor noise, segmentation and rapid motion, for example. Furthermore, in contrast to body pose estimation in depth images with reliable pose estimators and large annotated datasets, hand pose estimation is a less mature field, especially from the egocentric viewpoint and in the presence of objects (Yuan et al., 2018). As the first study to use hand pose features as cues for these kinds of scenario, the present research encountered a problem in acquiring and annotating data and designing a benchmark.

So far, we have discussed two application scenarios and considered how to extract spatio-temporal features from them. However, as previously mentioned, the extraction of meaningful features is only one part of the problem; the choice of a classifier to deal with these features is also crucial. In this research, we opted for a decision forest model (Breiman, 2001) in view of several desired properties: clusters obtained in leaf nodes, scalability, robustness to overfitting, multiclass learning and efficiency.

The main challenge in using decision forests classifiers for temporal problems concerns temporal dependencies. Previous approaches have encoded the temporal variable in the feature space by stacking multiple frames (Fothergill et al., 2012), handcrafting temporal features (Zhu et al., 2013) or creating codebooks (Yu et al., 2010). However, these methods require that temporal cues are explicitly given instead of automatically learning them. In an attempt to relieve this, Gall et al. (2011) and Yao et al. (2011) have added a temporal regression term, and frames individually vote for an action centre. This breaks the temporal continuity and thus does not fully capture the temporal dynamics. Lehrmann et al. (2014) have proposed a generative state-space model without exploiting the benefit of having rich labelled data. Dapogny et al. (2015) have grouped pairs of distant frames and grown trees by using hand-crafted split functions to cover different label transitions; however they encountered difficulty in designing domain-specific functions and making the model complexity increase with the number of labels. This thesis proposes a transition forest as a temporal decision forest model that aims to solve the aforementioned issues by automatically learning the temporal dynamics within the forest.

The following section describes the structure of this thesis and highlights its main contributions.

1.1 THESIS OUTLINE AND CONTRIBUTIONS

The thesis proceeds as follows. First, Chapter 2 presents a general literature review regarding action recognition on different data modalities. Then, Chapters 3, 4 and 5 address main contributions. Finally, Chapter 6 provides the summary, conclusions and suggestions for future work.

Highlights of main chapters are listed below:

Chapter 3: Understanding egocentric fingertip writing in mid-air with a trajectory Hough forest

In this chapter, we propose a framework for understanding fingertip writing in mid-air using an egocentric RGB-D sensor. The proposed approach first detects a writing hand posture and locates the position of the index fingertip in each frame. Fingertip points over time define a trajectory that represents a written character in mid-air. The written character is recognised and localised simultaneously. To achieve this task, we first used a contour-based view-independent hand posture descriptor that was extracted with a novel signature function. The proposed descriptor supports both posture recognition and fingertip detection. To recognise fingertip-written characters from trajectories, we propose a trajectory Hough forest that utilises sequential data as input and performs regression in both spatial and temporal domains. To encourage consistent temporal predictions, we ponder the posterior class probability of the forest with a prior probability based on clustering properties of forests. Furthermore, we introduce a new dataset that includes labels for hand postures and fingertips locations that represent written character in mid-air. For this research, we conducted experiments on posture estimation, fingertip detection, and character recognition and localisation, which indicate that our design choices are more robust than tested baselines. Additionally, we extend the framework to deal with spatio-temporal trajectories in RGB videos.

This chapter includes content that was published in the journal *Computer Vision and Image Understanding* (Chang et al., 2016) and presented at the IEEE Winter Conference on Applications of Computer Vision (WACV) in 2016 (Garcia-Hernando et al., 2016).

Chapter 4: Transition forests for action recognition and detection

This chapter introduces the novel method of transitions forests, which entails an ensemble of decision trees that learn to discriminate static frames and transitions between pairs of two independent frames. During training, node splitting is driven by alternating two criteria: the standard classification objective that maximises the discrimination power in individual frames as well as the proposed one in pairwise frame transitions. Growing the trees tends to group frames with similar associated transitions and the same action label to incorporate temporal information that was not available otherwise. Unlike conventional decision trees, whereby the best split in a node is determined independently of other nodes, the transition forests jointly seek the best split of nodes within a layer to incorporate distant node transitions. When inferring the class label of a new frame, it is passed down the trees, and an efficient, online prediction is made on the basis of previous frame predictions as well as the current one. The method has been applied to varied skeleton action recognition and online detection datasets to demonstrate its superior performance compared to several baselines and state-of-the-art approaches.

A reduced version of this chapter was presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2017 (Garcia-Hernando and Kim, 2017).

Chapter 5: Understanding egocentric hand-object actions with RGB-D videos and 3D hand pose annotations

This chapter studies the use of three-dimensional (3D) hand poses to recognise first-person hand actions in interaction with 3D objects. To this end, it proposes a RGB-D video benchmark of everyday actions that involve several different objects. To obtain high-quality hand pose annotations from real sequences, we used our own mo-cap system, which automatically infers the location of each of the 21 joints of the hand via six magnetic sensors on the fingertips and the inverse kinematics of a hand model. We present extensive experimental evaluations of RGB-D and pose-based action recognition according to baselines and state-of-the-art approaches. We also measure the impact of using appearance features and poses as well as their combinations with the extension of the transition forest that is presented in the previous chapter to manage different data modalities. Furthermore, we assess the readiness of current hand pose estimation in cases where hands are severely occluded by objects in egocentric views and investigate its influence on action recognition.

A reduced version of this chapter was presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2018 (Garcia-Hernando et al., 2018).

2

RELATED WORK

2.1 OVERVIEW

THIS chapter surveys previous work in action recognition. It divides the field into three main categories depending on the nature of the input data, which vary from RGB videos to RGB-D videos to egocentric videos. Although this thesis focuses on RGB-D and egocentric action recognition, with a small exception in Chapter 3, most techniques have their origin and inspiration in RGB action recognition; thus, for completeness, the chapter begins with a throughout review of this category. Also, since some of the main contributions of this thesis are based on decision forest models, Section 2.5 includes a review of decision forests for action recognition. Each section ends with a short review of popular benchmarks that directs particular attention to the most relevant datasets for the present research and emphasises those that were used for the experiments in subsequent chapters. For a more exhaustive list, we refer interested readers to the following survey papers: Herath et al. (2017) and Zhang, Li, Ogunbona, Wang and Tang (2016). The chapter concludes with a brief description of evaluation criteria that were found in the literature and applied in this thesis.

2.2 ACTION RECOGNITION IN RGB VIDEOS

This section reviews relevant work on human action recognition in RGB videos. It first discusses approaches that use hand-crafted feature representations before describing recent models that automatically learn feature representations. For an exhaustive review of the former methods, we refer to the review by Aggarwal and Ryoo (2011); for the latter, please see Herath et al. (2017).

2.2.1 *Hand-crafted feature representation methods*

Holistic and local approaches

This section first reviews holistic approaches that primarily describe a human action based on global representations of the human body. These representations can include low-level features that are computed over the spatio-temporal volume of the human action (Bobick and Davis, 2001; Efros et al., 2003; Yilmaz and Shah, 2005) or higher-level representations, such as body pose (Wang et al., 2013). For instance, Bobick and Davis (2001) have constructed templates of actions by using weighted projections of the space-time volume that summarise the presence and the history of motion from silhouettes of humans. Efros et al. (2003) have introduced a motion descriptor that is based on optical flow as computed over the human figure and which classifies actions by using a nearest neighbour nearest neighbour (NN) classifier. Yilmaz and Shah (2005) have exploited the differential properties of spatio-temporal volumes to characterise actions, while Blank et al. (2005) have described actions according to the spatio-temporal saliency of human silhouettes over time. The use of body pose as a feature representation can evidently improve performance compared to the use of low-level features (Yao et al., 2011), but it has been less popular in RGB approaches than in RGB-D approaches. The reason for this difference is the difficulty of accurately estimating the body pose from only RGB data; however, promising approaches (Chéron et al., 2015; Cao et al., 2017) could change this. Due to the inability to obtain satisfactory global representations, the interest shifted to local representation of actions.

Local representation of actions became popular through the work of Laptev (2005), the introduction of space-time interest points and an extension to the temporal domain of the Harris corner detection (Harris and Stephens, 1988). The main approach by Laptev (2005) was to localise points in space and time with significant spatial and temporal variation. As a result, sparse interest points could be detected, and derivative filter responses could be used to characterise actions. Instead of using a sparse set of interest points, Dollár et al. (2005) have incorporated a denser representation of interest points that reflects a more accurate modelling of actions compared to a sparse representation. This approach prompted the trend of using dense representations. A small spatio-temporal volume called a *cuboid* is associated with each interest point, and appearance features (i.e. normalised brightness, gradients, optical flow) are extracted to create a vocabulary, or codebook, via k-means and modelling actions that use histograms in a bag-of-words fashion. Following this line of work, Niebles et al. (2008) have applied a generative model to learn the probability distribution of each action class as represented by a collection of vocabulary words. Subsequently, Wong et al. (2007) have further extended the model to include structural spatio-temporal information that is relative to the centre of the action.

In addition to the simple representations by Laptev (2005) and Dollár et al. (2005), a well-studied issue is how to describe these spatio-temporal points. Most of these representations are extensions of their two-dimensional (2D) counterparts to the temporal domain: Scovanner et al. (2007) have extended the scale-invariant feature transform (Lowe, 2004) to the spatio-temporal domain (3D-SIFT), while Klaser et al. (2008) have extended the histogram of gradients (Dalal and Triggs, 2005) (HOG) descriptor (Dalal and Triggs, 2005) to propose HOG3D. Moreover, Laptev et al. (2008) have suggested the combination the HOG descriptor with a new temporal descriptor, namely the histogram of optical flow (Laptev et al., 2008) (HOF). This combination together with a spatio-temporal pyramid to embed structural information, a bag-of-words representation and a support vector machine (SVM) classifier were proven to be state-of-the-art approaches at the time and were influential for later work (Wang et al., 2009).

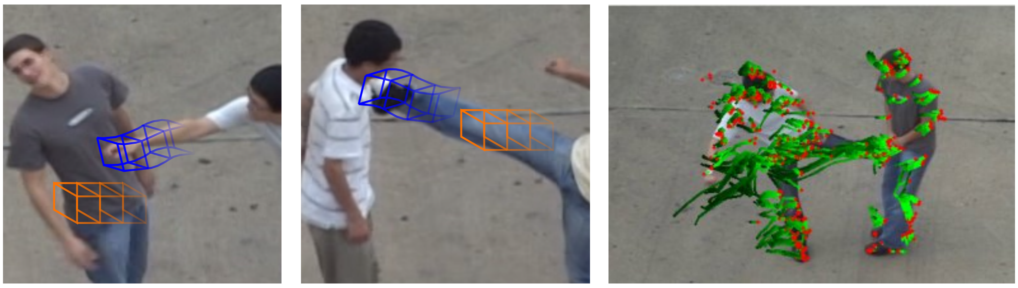

Figure 2.1: The first two images reveal the difference between cuboid sampling and trajectory sampling, and the third image depicts improved dense trajectories (Wang and Schmid, 2013); all images were extracted from UT-Interaction dataset (Ryoo and Aggarwal, 2010).

One fundamental limitation in capturing information from fixed spatio-temporal locations is that they do not necessarily capture the motion information for a sufficiently long temporal span. To overcome this limitation, a new line of research has emerged that tracks a given spatio-temporal point over time and thereby captures longer and more complex motion information along its spatio-temporal trajectory, as Figure 2.1 illustrates. For instance, Messing et al. (2009) have extracted trajectories by using a point detector (Laptev, 2005) and a Kanade–Lucas–Tomasi (KLT) feature tracker (Lucas et al., 1981), and they have proposed a graphical model for velocities of trajectories. Matikainen et al. (2009) have also determined trajectories with a KLT feature tracker and clustered and classified them in a bag-of-words fashion. Motivated by the improvement of dense sampling over sparse spatio-temporal points, Wang et al. (2011) have sampled and tracked trajectories in a dense way. In doing so, they extracted local feature descriptors, such as HOG, HOF and motion boundary histograms (Dalal et al., 2006) (MBH), along the trajectories and processed them similarly to Laptev et al. (2008). Although dense sampling can capture non-meaningful trajectories, this problem can be attenuated by modelling the camera motion according to improved dense trajectories (Wang and Schmid, 2013) (IDT). Figure 2.1 presents an example of such trajectories. Oneata et al. (2013) and Peng et al. (2014) have explored the use of Fisher vectors (Perronnin et al., 2010) for feature aggregation as an alternative to bag-of-word trajectory encoding. Wang and Schmid (2013) work with Fisher vector encoding is considered the state of the art in action recognition through hand-crafted feature representations. Section 2.2.2 introduces methods that learn the feature representation. Although they are generally able to outperform hand-crafted approaches, it is apparent that most feature-learned methods benefit from a combination with the improved trajectories of Wang and Schmid (2013).

Sequential approaches

This section concludes with an overview of approaches to the problem of action recognition as a sequence of observations. In general, these approaches extract a feature vector that describes the human in every frame, and a decision is made according to a classifier that considers the temporal structure of the observations. Early approaches that compare an observed sequence to a template sequence that represents an action class include those of Darrell and Pentland (1993) and Gavrila et al. (1995). These two works have proposed the use of dynamic time warping (DTW) to classify simple human hand gestures and upper body movements, respectively. The use of state-space models such as the hidden Markov model (HMM) for action recognition started with the seminal work by Yamato et al. (1992). An HMM represents each action class, and these are trained (i.e. observation and transition probabilities are estimated) with the labelled data for a particular class. In the inference stage, the observed sequence is evaluated for each model, and the most likely class is selected. Approaches that involve variants of HMM or similar probabilistic frameworks have appeared constantly in the literature (Oliver et al., 2000; Duong et al., 2005; Weinland et al., 2007; Lv and Nevatia, 2007; Tang et al., 2012). In view of their superior performance and efficiency, researchers have also proposed discriminative approaches as alternatives to generative approaches via models such as conditional random fields (Sminchisescu et al., 2006; Quattoni et al., 2007; Raptis and Sigal, 2013), other discriminative state-space models (Ma et al., 2017) and ranking methods that model frame order (Fernando et al., 2015). Another notable line of research has been inspired by language models that use sequential models to decompose actions in various levels or hierarchies (Ivanov and Bobick, 2000; Oliver et al., 2002; Ryoo and Aggarwal, 2009). These approaches have the advantage of tailored applicability to long and complex actions, but they are disadvantaged by their sensitivity to error propagation between layers. The research direction for sequential approaches has recently shifted to the use of deep models that involve recurrent neural networks, as these can be trained in an end-to-end fashion and thus simultaneously learn the feature representation and sequential model. The following section reviews this line of work.

2.2.2 *Learned representation methods*

After the success of learning visual representations in still images through convolutional neural networks (CNNs or ConvNets) (LeCun et al., 1989; Krizhevsky et al., 2012) over hand-crafted features for several computer vision tasks (Zhou et al., 2014; Girshick et al., 2014), the application of deep learning to action recognition has not been an exception (Herath et al., 2017). Ji et al. (2013), Karpathy et al. (2014), Tran et al. (2015) and Varol et al. (2017) have attempted to extend image convolutions to the

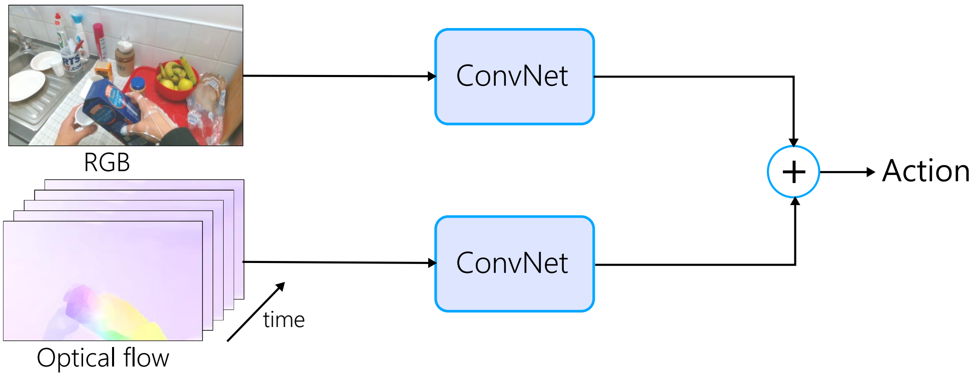

Figure 2.2: Two-stream approach (Simonyan and Zisserman, 2014); deep features are extracted from both RGB and optical flow channels and combined in prediction.

spatio-temporal domain by demonstrating effective performance through the use of only RGB cues, but their outcomes were still inferior to the state-of-the-art, hand-crafted approach of IDT. Given the difficulty of working on spatio-temporal volumes, Simonyan and Zisserman (2014) have commenced an impactful line of research (see Figure 2.2) to learn features from two streams, namely the appearance stream (image) and the motion stream (optical flow). This approach can exploit strong results from the image recognition domain by pre-training the networks on large visual datasets. Feichtenhofer et al. (2016) have recently proposed improvements to this architecture on the basis of their exploration of different temporal fusions, Varol et al.'s (2017) use of temporally longer convolutions and the addition of correspondences between streams (Feichtenhofer et al., 2017). However, a drawback of such an approach is the need to compute the computationally expensive optical flow. Bilen et al. (2016) have suggested one approach that strives to overcome this limitation without the use of temporal convolutions; this method creates an image on top of the network that summarises both appearance and motion.

To more effectively capture temporal dependencies between video frames, another line of research has incorporated the deep sequential model of recurrent neural network (RNN) with long-short term memory (Hochreiter and Schmidhuber, 1997) (LSTM) after feature learning from a CNN (Donahue et al., 2015). This feature learning can occur only on colour frames (Donahue et al., 2015), by capturing the motion on a flow channel through standard use of a two-stream model (Yue-Hei Ng et al., 2015) or by adding an attention mechanism (Li et al., 2018). In contrast to other tasks of computer vision, results for the learned representation approaches above can be improved by a combination with hand-crafted features, such as IDT, wherein deep architectures miss some spatio-temporal patterns (Feichtenhofer et al., 2016). Recently, Carreira and Zisserman (2017) have proposed a deep model that involves 3D convolutions on both colour and flow channels that, jointly with a substantially larger and curated dataset pre-training, was able to yield better results compared to all previous approaches and without the need to resort to IDT features.

2.2.3 *RGB action recognition benchmarks*

From the rich history of RGB action recognition in computer vision, it is clear that the complexity and difficulty of utilising datasets closely accompany advancements in the techniques that Chapter 2 has presented. Among the first proposed datasets are the KTH (Schuldt et al., 2004) and Weizmann (Blank et al., 2005) datasets. These datasets have a limited number of action classes and have simple categories, such as 'walk' and 'jump'. Both datasets reflect single actors from a third-person viewpoint in a controlled scenario. In a scenario of surveillance with a higher number of humans in the scene, we find the UT-Interaction (Ryoo and Aggarwal, 2010) dataset, which we use in Chapter 3 to evaluate the generalisation of our framework. This dataset consists of six classes of human-human interactions, such as 'shake hands', 'point', 'hug', 'push', 'kick' and 'punch'. In a much more challenging scenario, Marszałek et al. (2009) have proposed the Hollywood2 dataset, which is comprised of segments from movies. The difficulty of this dataset compared to the previous one derives from the variability of viewpoints and scales that naturally appear in movies. Other popular datasets include (Rodriguez et al., 2008) and Sports-1M (Karpathy et al., 2014), which describe sports. While the former is a small dataset of only 10 classes, Sports-1M contains more than one million Youtube videos in almost 500 categories. By obtaining videos from YouTube, we determined that the most popular datasets at the time of writing this thesis were UCF-101 (Soomro et al., 2012) and HMDB-51 (Kuehne et al., 2011). The challenges of these datasets concern the inclusion of subtle actions, such as 'apply make up' or 'apply lipstick'. Furthermore, videos from YouTube are not professionally recorded, which can negatively affect camera motion and resolution quality. Carreira and Zisserman (2017) have recently proposed Kinetics, a high-scale dataset of Internet videos which allowed for the training of large, deep models that yielded state-of-the-art results for the UCF-101 and HMDB-51 datasets.

2.3 ACTION RECOGNITION IN RGB-D VIDEOS

The recent introduction of commodity sensors, such as Microsoft Kinect$^{\text{TM}}$, has considerably increased interest in action recognition through RGB-D sensors. The use of depth cameras for action recognition differs from traditional RGB action recognition in the availability of an additional data modality, depth. While most successful RGB approaches (Wang et al., 2011; Feichtenhofer et al., 2016) extract information predominantly from static colour images or motion flow, these approaches are not directly applicable to the depth stream because of its noisy, texture-less and discontinuous pixel regions.

Researchers have outlined numerous approaches to extracting information from the depth channel. These methods have usually focused on the extraction of discriminative features from depth images via geometric descriptors (Ohn-Bar and Trivedi, 2014; Oreifej and Liu, 2013; Yang and Tian, 2014) that are sensitive to viewpoint changes and view-invariant approaches (Rahmani et al., 2016; Rahmani and Mian, 2016; Baek, Shi, Kawade and Kim, 2017). For instance, a depth-based descriptor that was developed from the histogram of oriented 4D normals (HON4D) (Oreifej and Liu, 2013) can successfully describe local geometry, and Yang and Tian (2014) have extended it to incorporate local information for neighbourhood pixels according to super normal vectors. However, these methods suffer when the viewpoint changes from a frontal view to another view. To overcome this disadvantage, Rahmani et al. (2016) have proposed a histogram of oriented principal components to detect and characterise interest points that are robust for viewpoint variations. Moreover, Rahmani and Mian (2016) have recently promoted the learning of view-invariant features through the use of CNNs from several synthesised depth views. Lately, a popular trend has entailed the utilisation of the depth channel to obtain robust human body pose estimates (Shotton, Sharp, Kipman, Fitzgibbon, Finocchio, Blake, Cook and Moore, 2013) and use them directly as a holistic feature or in combination with other RGB-D features to recognise actions.

Some works have combined the above data modalities to yield multimodal approaches. For instance, Wang, Liu, Wu and Yuan (2012) have captured local occupancy patterns around the estimated body joints and identified the most discriminative joints. Meanwhile, Ohn-Bar and Trivedi (2014) have combined joint-angle representations with a modified histogram of gradients (HOG^2), while Zhu et al. (2013) have utilised colour and flow information together with pose features. Shahroudy, Ng, Yang and Wang (2016) have combined pose and depth features in a hierarchical learning framework, and Shi and Kim (2017) have used skeleton information as privileged learning for training while using only RGB-D information in testing. Hu et al. (2015) have advocated for jointly learning heterogeneous features (JOULE) for action recognition from all available data streams (i.e. colour, depth and pose). Given the popularity of using only *skeleton* (pose) features for RGB-D action recognition, the following section focuses on this line of research.

Skeleton-based action recognition

Generative models (Xia et al., 2012; Wu and Shao, 2014; Lehrmann et al., 2014) pose disadvantages in the difficulty of estimating model parameters and their time-consuming learning and inference stages. Thus, discriminative approaches have been widely adopted for their superior performance and efficiency. One main line of research consists of learning discriminative features from skeleton data. For instance, Vemulapalli et al. (2014) and Vemulapalli and Chellappa (2016) have represented entire skeletons as points in a Lie group before temporally aligning sequences with DTW and capturing temporal dynamics through Fourier temporal pyramids (FTPs), which resembles the approach of Wang, Liu, Wu and Yuan (2012). This involves a moving pose descriptor (Zanfir et al., 2013) (MP) that uses both pose and atomic motion information and then temporally mining key frames through a k-NN approach in contrast to Jung and Hong (2014), who used DTW. Devanne et al. (2015), Wang, Wang and Yuille (2016) and Zhu, Zhang, Shen and Song (2016) have investigated the use of key frames or key motion units and have reported good performance, which reveals the importance of static information for action recognition. Recently, some works have utilised CNNs to automatically learn features directly from skeleton data (Ke et al., 2017) or from Lie group representations (Huang et al., 2017).

Researchers have also proposed deep sequential models that involve vanilla RNNs (Du et al., 2015) and with LSTMs (Veeriah et al., 2015; Zhu, Lan, Xing, Zeng, Li, Shen and Xie, 2016; Liu et al., 2017) to model temporal dependencies. However, these models have exhibited inferior performance compared to recent models that explicitly exploit static information (Wang, Wang and Yuille, 2016; Wang, Yuan, Hu, Li and Zhang, 2016) or well-suited time-series mining called Gram Matrix (Zhang, Wang, Gou, Sznaier and Camps, 2016). In demonstrating the benefit of combining feature learning and sequential deep models, Du et al. (2015) have first proposed a hierarchical, end-to-end architecture that uses a bi-directional HBRNN. This approach contrasts with Veeriah et al. (2015), who have directly fed hand-crafted features into a RNN with LSTM.

2.3.1 *RGB-D action recognition benchmarks*

Most datasets for action recognition that use RGB-D sensors also include human body pose annotations that have been obtained via Microsoft Kinect (Shotton, Girshick, Fitzgibbon, Sharp, Cook, Finocchio, Moore, Kohli, Criminisi, Kipman et al., 2013). Since RGB-D sensors work only indoors, action classes are usually limited to daily actions and gaming or human-computer interaction (HCI). The first proposed RGB-D benchmark was MSR-Action3D (Li et al., 2010), which depicts actions in a gaming scenario with a fixed background and camera. Other popular gaming datasets are MSRC-12 (Fothergill et al., 2012) and UT-Kinect datasets (Xia et al., 2012). Gaming actions include ‘tennis serve’ or ‘shoot a pistol’, and they are usually recorded from a single viewpoint with the user facing the camera. Popular datasets regarding daily life actions are MSR-DailyActivity3D (Wang, Liu, Wu and Yuan, 2012), CAD-60 (Sung et al., 2011) and Florence-3D (Seidenari et al., 2013), which include actions such as ‘drink’, ‘eat’ and ‘answer phone’. These are usually more challenging because they involve objects and the user might not be facing the camera. One limitation of the aforementioned datasets is that they are recorded from a single viewpoint (i.e. the camera is fixed). To combat this limitation, the UWA3D Multiview II (Rahmani and Mian, 2016) and NTU RGB+D (Shahroudy, Liu, Ng and Wang, 2016) datasets were proposed to explore other camera settings. The latter has compiled more than 56,000 videos and 60 classes to become the largest at the present time, and it is thought to be useful for data-hungry deep learning algorithms. In the next section, we provide more details for the datasets that we use in Chapter 4.

The **MSR-Action3D dataset** (Li et al., 2010) contains 20 actions performed by 10 actors as well as 567 videos with a resolution of 320-by-240 pixels. Microsoft recorded this database with a prototype of the Kinect and only provided depth and skeleton data. Two popular evaluation protocols are associated with this dataset. The first divides the dataset into three subsets of eight actions (AS1, AS2 and AS3) and performs cross-subject validation by assigning half the users to training and half to testing. It measures final accuracy by averaging the classification performance over 10-fold validation on all three sets. The second protocol is more difficult but differs only in that it does not divide the actions into three subsets and performs classification of 20 actions. Actions categories include 'high arm wave', 'horizontal arm wave', 'hammer', 'hand catch', 'forward punch', 'high throw', 'draw x', 'draw tick', 'draw circle', 'hand clap', 'two hand wave', 'side-boxing', 'bend', 'forward kick', 'side kick', 'jogging', 'tennis serve', 'golf swing', 'pickup' and 'throw'.

The **MSRC-12 dataset** (Fothergill et al., 2012) contains 12 actions performed by 30 actors. There are 6,000 instances of actions, and each actor repeats them several times. The actions are one of two types, namely iconic or metaphoric. Iconic actions focus on gaming and include 'hide', 'shoot a pistol', 'throw an object', 'change weapon', 'kick' and 'put on night vision goggles', whereas metaphoric actions concern HCI and could include 'start music', 'navigate to next menu', 'wind up the music', 'end music session', 'protest the music' and 'move up the tempo of the song'. In this thesis, we followed the protocol of Lehrmann et al. (2014); accordingly, we used the six iconic gestures and performed five-fold leave-person-out cross-validation, which required 24 actors for training and six for testing per fold.

The **Florence-3D dataset** (Seidenari et al., 2013) consists of nine actions by 10 subjects. Each subject performed every action two or three times for a total of 215 action sequences. In this research, we followed the protocol of Wang, Wang and Yuille

(2016) and Wang, Yuan, Hu, Li and Zhang (2016), which dictates a leave-one-subject-out protocol. This approach uses nine subjects for training and one for testing for a total of 10 times. Actions include 'wave', 'drink from a bottle', 'answer phone', 'clap', 'tighten lace', 'sit down', 'stand up', 'read', 'watch' and 'bow'.

The **Online Action Detection (OAD) dataset** (Li et al., 2016) is not an action recognition dataset but rather an online action detection dataset. However, we include it here for completeness, as we perform experiments on it in Chapter 4. The main difference between this dataset and the previous ones is that actions are not isolated; one can find multiple actions in the same video as well as an absence of action. The dataset consists of 59 long sequences that contain 10 daily-life actions that are performed by various actors and recorded with Kinect v2 for a total of over 216 minutes of video. Each sequence contains different actions and background periods of variable length (3,000 frames on average) in an arbitrary order with annotated starting and ending frames. Actions include 'drinking', 'eating', 'writing', 'opening cupboard', 'washing hands', 'opening microwave', 'sweeping', 'gargling', 'throwing trash' and 'wiping'.

2.4 ACTION RECOGNITION IN EGOCENTRIC VIEWPOINT

Action recognition from an egocentric viewpoint warrants a specific section to discuss its particularities in contrast to third-person videos. Unlike third-person videos, an egocentric viewpoint does not provide access to the body of the actor[1] and only depicts the actor's current perspective. As such, hands and manipulated objects become the most descriptive cues of egocentric actions (Mayol and Murray, 2005; Fathi, Farhadi and Rehg, 2011; Fathi, Ren and Rehg, 2011; Pirsiavash and Ramanan, 2012; Bambach et al., 2015; Ma et al., 2016; Singh et al., 2016), which compromises the aforementioned approaches that focus on the human body. As another important characteristic of an egocentric viewpoint, the sensor is not fixed, and abrupt motion might appear, which could complicate the application of state-of-the-art tracking methods, such as IDT (Ishihara et al., 2015; Singh et al., 2016). Some researchers have investigated the use of the human gaze as extracted with a special sensor as an attention cue for egocentric

[1] In this thesis we assume that the wearer of the camera performs the action, although in some works (Ryoo and Matthies, 2013) the wearer is an observer of the action.

vision (Fathi et al., 2012; Damen et al., 2014; Li et al., 2015). However, this thesis considers purely vision-based systems.

An early approach by Fathi, Farhadi and Rehg (2011) has found that low-level features that were extracted from hands were a rich source of information to recognise egocentric actions. However, Pirsiavash and Ramanan (2012), who have determined that recognising egocentric action is 'all about objects', have used HOG descriptors to model objects. However, most recent and successful works (Ishihara et al., 2015; Ma et al., 2016; Singh et al., 2016) have followed a hybrid approach that incorporates a detection stage from low-level pixel information to combine both hands and object features to model actions. Ishihara et al. (2015) have extracted motion features with IDT and hand features with HOG descriptors via Fisher vector encoding. Ma et al. (2016) and Singh et al. (2016) have adopted the state-of-the-art approach of two-stream networks (Simonyan and Zisserman, 2014; Feichtenhofer et al., 2016) in an egocentric setting. Ma et al. (2016) have proposed the detection of both hands and objects in dedicated networks and the fusion of the network with a motion network from the optical flow. Similarly, Singh et al. (2016) have recommended an additional third stream that specifically examines hand features. Chapter 5 explores the use of fine-grained hand pose features for egocentric action recognition in contrast to previous approaches.

2.4.1 *Egocentric benchmarks*

The first benchmark that appeared for egocentric action recognition was the CMU-Kitchen dataset (Spriggs et al., 2009). Its authors utilised a multi-camera system, which included one wearable sensor, and combined it with mo-cap system to capture full body articulations from a third-view camera. Actors who wore special clothes prepared a variety of recipes in a highly controlled environment, and temporal bounds of actions are provided. In attempting to advance this work, Fathi, Ren and Rehg (2011) have proposed *Georgia Tech Egocentric Activity* (GTEA), a dataset of daily actions that were captured in 'the wild' by users who simply manipulated objects in their kitchens while wearing a non-intrusive GoPro sensor. The dataset was first labelled with seven high-level actions, such as 'make hotdog sandwich', but was later refined and re-annotated to contain 71 actions, including 'put hotdog on bread'. Another related dataset is the

Activities of Daily Living (ADL) dataset (Pirsiavash and Ramanan, 2012), which includes 18 daily actions. Unlike GTEA, ADL includes actions that occurred outside of the kitchen, and some of them do not involve manipulated objects (e.g. 'watch television'). Authors of GTEA proposed a later version of the dataset called Gaze and Gaze+ (Fathi et al., 2012), which includes gaze annotations with an eye tracker and 44 actions. Damen et al. (2014) also included gaze annotations in the *Bristol Egocentric Object Interactions Dataset* (BEOID), which was first referenced to study the automatic discovery of manipulated objects by users but later annotated with action recognition labels and temporal bounds (Wray et al., 2016) that involve pairs of a 'verb' (action) plus a 'noun' (object). In each dataset that has been mentioned so far, the actor is alone, and no other humans are involved. By adding interactions with other humans in which the wearer of the sensor is passive, Ryoo and Matthies (2013) have proposed the JPL-Interaction dataset, wherein other actors provide the user with actions such as 'being punched' and 'hand shake'. Bambach et al. (2015) have also proposed a dataset of four actions that involve interactions with other humans and provide pixel-level annotations of hands. We present our benchmark in Chapter 5, which includes both daily and interaction classes in three scenarios.

2.5 ACTION RECOGNITION AND DECISION FORESTS

Standard decision forest approaches for action recognition, such as that of Fothergill et al. (2012), directly stack frames and grow forests to classify them. Zhu et al. (2013) and Seidenari et al. (2013) have created bags of poses that break the temporal structure and classified the entirety of sequences. Mikolajczyk and Uemura (2011) have proposed simultaneous action recognition and localisation through a local motion-appearance features method and clustering trees. By clustering the properties of trees, Yu et al. (2010) have also constructed codebooks with the help of multiple heuristic rules to capture structural information. Baek, Kim and Kim (2017) have encoded temporal information through the use of temporal features, such as MP, and the introduction of spatio-temporal contexts based on RGB-D stacks of frames. Baek, Shi, Kawade and Kim (2017) have introduced a kinematic term inside the forest to capture the geometry of the scene of indoor actions and additionally incorporated MP as a temporal feature. These approaches require that temporal cues are directly encoded in the feature space.

To relieve the necessity of manually encoding temporal cues in the feature space Gall et al. (2011), Yu et al. (2013) Jang et al. (2015) and Serrano et al. (2018) have added a temporal regression term and mapped both appearance and pose features to vote in an action Hough space. However, in Hough frameworks, temporal information is captured as temporal offsets with respect to a temporal centre of independent samples, which disrupts the temporal continuity and requires observation of the whole sequence. Pairwise conditional random forests (Dapogny et al., 2015) (PCRF) have been proposed for the related field of facial expression recognition and consist of trees for which hand-crafted split functions operate on pairs of frames. These pairs are formed to cover different facial dynamics and fed into multiple subsets of decision trees that are conditionally drawn on the basis of different label transitions, which ensures that the ensemble size is proportional to the number of labels. Generative forest-based methods include dynamic forest models (Lehrmann et al., 2014) (DFM), which are ensembles of autoregressive trees that store multivariate distributions in their leaf nodes. These distributions model observation probabilities given a short history of previous frames. Similarly to HMM, a decision forest is trained for each action label, and inference is performed to maximise the likelihood of the observed sequence. Recently, and in less relation to the scope of this thesis, Chen et al. (2016) have encouraged the learning of smooth temporal regressors for real-time camera planning, while Charles et al. (2014) have introduced temporal context in a decision forest framework by warping map confidences through the use of optical flow for body pose estimation.

A related line of work (Shotton et al., 2008; Nowozin et al., 2011; Kontschieder et al., 2013; Shotton, Sharp, Kohli, Nowozin, Winn and Criminisi, 2013) has proposed decision forest methods for image segmentation. The objective of such an approach is to obtain coherent pixel labels, and decision forests are linked with probabilistic graphical models to connect multiple pixel predictions. However, these methods focus on the spatial coherence of predictions in an image space, while this thesis concentrates on discriminative changes of data and prediction in a temporal domain.

2.6 EVALUATION CRITERIA

There are two popular evaluation criteria in action recognition. The first and most extended criterion is usually denoted as *'accuracy'* and is a per-cent measure of the number of correctly classified videos in relation to the total number of videos. A drawback of this measure is its sensitivity to dataset class imbalance, wherein some classes are much more common than others, which introduces bias to the model. To relieve this problem, another popular measure is the *'average class accuracy'*, which computes the correct number of classified videos over the total videos for a given class and considers the average as the final performance indicator. In some large benchmarks, such as Sports-1M, it is logical to define a *'top-k' accuracy*, which considers a video classification to be correct if the true label appears among the top k results of the classifier.

The online action detection experiment that we perform on the OAD dataset in Chapter 4 measures the performance with $F1$ scores. It identifies a detection as correct if the intersection over union, IoU, between the prediction action interval I and the ground-truth interval I_{gt} exceeds a fixed threshold, such as that of 60% in Li et al. (2016):

$$IoU = \frac{|I \cap I_{gt}|}{|I \cup I_{gt}|}.$$

Using the above criterion, the $F1$ scores are computed as follows:

$$F1 = 2\frac{precision \cdot recall}{precision + recall}.$$

For the evaluation of the localisation of starting and ending points of the interval of an action, where $[t_s, t_e]$ the ground-truth temporal points, the score is computed as $e^{|t-t_s|/|t-t_e|}$. For false positive and false negatives, the score is set to zero.

3

UNDERSTANDING EGOCENTRIC FINGERTIP WRITING IN MID-AIR WITH A TRAJECTORY HOUGH FOREST

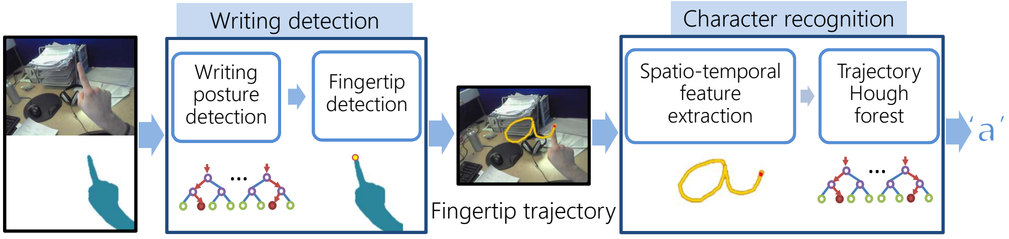

Figure 3.1: Overview of the proposed framework for recognising egocentric fingertip writing in mid-air.

Wearable cameras lack a keyboard or similar text input device that can allow the user to easily communicate with the system. In this chapter, we propose a framework for using an egocentric RGB-D sensor to recognise fingertip writing in mid-air. The framework first detects whether the user is writing and localises the fingertip. It then extracts spatio-temporal features from the trajectories that the fingertip depicts and feeds them into a trajectory Hough forest (THF). Finally, the character that the user has written is recognised and localised in the 3D space. As an extended version of the standard Hough forest (Gall et al., 2011), THF encourages temporal consistence in prediction through the clustering of forest properties. To test the suitability of the framework's components, we introduce a new dataset and perform multiple experiments on it. Furthermore, we extend the framework to engage with more general spatio-temporal trajectories (Wang and Schmid, 2013) from RGB data.

Contributions

The main contributions of this chapter are as follows:

- The proposal of a view-independent hand posture descriptor that is based on a novel signature function and which leads to robust writing pose and fingertip detection.
- A new framework termed trajectory Hough forest (THF) that extends the Hough forest (Gall et al., 2011) and encourages consistence in prediction.
- The introduction of a new fingertip-writing dataset that was captured from an egocentric view, has positive and negative poses and fingertip positions as well as character labels of trajectories.

Note on contributions

This chapter contains material that was published in Chang et al. (2016) for which the author of this thesis shares authorship. The author of this thesis was not involved in designing, implementing or experimenting with the spatio-temporal feature extraction or the forest baseline (Hough forest without transition term) that Sections 3.3.3 and 3.5 describe and thus claims no contribution to these parts.

3.2 RELATED WORK

Trajectories from writing in mid-air

Vision-based systems for recognising handwritten trajectories in mid-air are not new, and authors have proposed numerous approaches over the last two decades. Such approaches have been highly dependent on the available hardware and mainly assume a third-person viewpoint. In the context of augmented reality and the use of a sophisticated device with an infrared and colour sensor, Oka et al. (2002) have obtained promising results for the tracking of fingertips and recognition of the trajectories of simple geometric shapes. Alon et al. (2009) have proposed a mid-air handwriting recognition framework that employs a standard colour camera. The trajectories that represent digits were collected with the user facing his or her fist and wearing coloured gloves in the training stage. Schick et al.

(2012) has proposed a hand-tracking approach that involves a stereo camera system in front of a virtual blackboard to relieve users of the need to wear special sensors or clothes. With the arrival of commodity depth sensors, Raheja et al. (2011), Feng et al. (2012), Zhang et al. (2013), Vikram et al. (2013) and Aggarwal et al. (2015) have explored the use of depth to recognise fingertips and handwritten trajectories. In our application, depth sensors allow for easy segmentation of the hands with a simple distance filter; in contrast, the lighting conditions and background in other RGB camera-based approaches severely affect the segmentation quality.

The problem remains quite unexplored from an egocentric perspective; however, some early approaches relate to this work. Liu et al. (2006), Hannuksela et al. (2007), Jin et al. (2007), Shah et al. (2011) and Ishida et al. (2010) have utilised RGB cameras from a first-person perspective to recognise finger writing by applying different techniques of sequence recognition and handwriting recognition fields. These previous approaches did not consider the problem of an egocentric viewpoint, as the experiments were all undertaken in highly controlled conditions with no challenging hand postures present. By examining the dataset and applying the framework that this chapter proposes, Hameed et al. (2015) have explored the spatio-temporal design of offline spatio-temporal features. Some time after the publication of this chapter, Huang et al. (2016) proposed a framework that resembles that of the present study and uses a convolutional neural network (CNN) to detect hand postures and fingertips in RGB videos.

Hand posture recognition

Recognising hand postures is a difficult and unresolved problem in computer vision. Variation of illumination, point of view (e.g. 3D rotations, scale) and acquisition noise complicate the task. The literature presents two major families of methods: generative and discriminative approaches. Generative approaches (Oikonomidis et al., 2011*a*) aim to recover the full 3D hand pose via 3D model fitting. These are not suitable for present application since their high computational cost is unfavourable for fast hand movements. In contrast, discriminative methods (Hu and Yin, 2013; Tang et al., 2013; 2014; Rogez et al., 2014) directly construct mappings between training and testing poses, which is usually more efficient. In this chapter, we aim to recognise a particular hand posture from a binary image that describes a silhouette as a result of a previous segmentation stage. In view of this purpose, discriminative methods are the most suitable to apply.

A variety of approaches have been proposed to address the general problem of shape feature representation and recognition (Zhang and Lu, 2004; Yang et al., 2008), and some of these methods have been applied to hand posture recognition. Such previous works can be divided into region-based (Hu and Yin, 2013) and contour-based (Belongie et al., 2002; Persoon and Fu, 1977; Zhang and Lu, 2001) approaches according to whether features are extracted only from the entire shape region or from the contour. Regarding region-based techniques, Hu and Yin (2013) have proposed a topological-based feature descriptor which describes the behaviour of the holes between the hand region and its convex hull under morphological operations. This feature representation has proven to be accurate for discriminating between different hand postures from similar viewpoints, but it fails under drastic viewpoint and shape changes. Among contour-based methods, shape context (Belongie et al., 2002) has performed well in hand posture recognition under controlled conditions, but its performance drastically declines as the viewpoint varies. Another popular contour-based approach is the use of Fourier descriptors (Persoon and Fu, 1977) (FDs), which permits an invariant hand-shape representation for hand posture recognition (Chen et al., 2003; Bourennane and Fossati, 2012; Conseil et al., 2007). In advancing applications of FDs, Zhang and Lu (2001) have used signature functions, which are one-dimensional functions that represent different features – e.g. curvature, distance to the shape centroid, turning angle - that derive from the shape contour.

Fingertip detection

The detection and tracking of fingertips with both colour and depth cameras has been an active topic in the fields of human-computer interaction (HCI) and augmented reality (AR). Hand pose estimation approaches (Tang et al., 2013; 2014; Rogez et al., 2014) can detect fingertips; however, since we need only an estimation of the fingertip, we prefer a simpler approach. A popular alternative approach entails first segmenting the hand silhouette based on colour or depth cues and then detecting fingertips from the extracted binary shape. Following this line, many works have focused on the structure of the hand and exploited its contours and geometrical properties to localise fingertip points. In contrast to other parts of the hand, fingertips are high curvature points; Lee and Hollerer (2007) and Pan et al. (2010) have exploited this property in considering the contour curvature as a cue to detect fingertips. Another typical key characteristic of fingertips is their substantial distance from the hand palm. In view of this, Bhuyan et al. (2012) and Liang et al. (2012) have applied a distance metric from the hand palm to the contour's furthest points to localise candidate points, which were subsequently refined by various techniques. Raheja et al. (2012) have proposed a two-step algorithm whereby the fingertip is localised from the hand edges after estimating the hand direction, while Maisto et al. (2013) have also taken into account the topological structure of the hand in extracting points from the convex hull of the silhouette. A notable disadvantage of these methods is that the fingertips are not always over the hand silhouette's edge in all hand postures. To address this assumption, Raheja et al. (2011) have detected the fingertips as the hand points which are closer to the sensor after segmenting the hand palm and the fingers, and Yu et al. (2014) later followed and reinforced this approach with a hand graph model that is similar to that of Krejov and Bowden (2013). Krejov and Bowden (2013) have extended the distance concept by using a geodesic distance to localise fingertips in hand configurations for which previous methods had failed and thereby enforcing it with the natural structure of hand. Most of these approaches assume that the palm always faces to the camera, which is not an appropriate assumption for our application. Nevertheless, we are only interested in localising fingertip in one hand-pointing posture, which simplifies our task compared to the aforementioned works which have aimed to detect fingertip in any hand configuration.

3.3 PROPOSED FRAMEWORK

Figure 3.1 displays the proposed framework, which is composed of three main stages: 1) detection of writing hand poses; 2) detection of fingertips; 3) recognition and spatio-temporal localisation of trajectories. The following sections are organised accordingly.

3.3.1 *Handwriting posture detection*

The proposed technique for handwriting posture detection assumes that the hand has been pre-segmented successfully by, for instance, setting a depth threshold or skin colour selection. For the application of interest where the user is mainly wiring and not manipulating any object, we found that this is not a hard assumption while using a depth camera. The posture detection is modelled as a binary problem where positive values are the closed-hand pointing posture and the rest, including boundary cases, are treated as negative values.

To make the technique independent from sensors and use only depth values for segmentation, we propose a new contour-based hand posture descriptor using Fourier descriptors (Persoon and Fu, 1977) extracted from a novel shape signature function. The segmented hand is represented as a binary image as shown in Figure 3.2. A planar contour curve s_t extracted from the binary image of frame t. We propose a novel signature function based on a distance-weighted scale invariant measure of the contour curvature. The advantages of this new signature function are the following: it is a discriminative feature, which permits a high accurate description of the hand posture; it is not computationally demanding as there is only need to examine one scale; it can be reused for fingertip detection.

Scale invariant curvature measure. We propose to use a scale invariant measure of the curvature presented in Feldman and Singh (2005), which we refer to as *curvature entropy* u. Consider the extracted hand contour at frame t depicting a curve s_t of length L, which is sampled at discrete intervals $\Delta s = L/n$. The curvature κ measures the change on the tangent direction β and can be *locally* approximated by:

$$\kappa \approx \frac{\beta}{\Delta s}. \tag{1}$$

This approximation becomes exact in the limit ($\Delta s \to 0$ when $n \to \infty$). Rearranging terms an expression for β can be obtained:

$$\beta \approx \kappa \Delta s. \tag{2}$$

As shown by Feldman and Singh (2005), β follows a von Mises distribution and thus $\kappa \Delta s$ is distributed likewise, leading to the curvature entropy expression:

$$u(\kappa) \propto -cos(\kappa \Delta s). \tag{3}$$

This quantity is scale-invariant and it is locally proportional to its curvature κ. While κ is not a scale-invariant quantity, the product $\kappa \Delta s$ is. The intuition behind is that κ and Δs scale inversely by the same factor when the curve changes in size (Feldman and Singh, 2005). Computing the entropy values along the contour s_t, the series $u(\kappa(s_t))$ is obtained, which permits to localise high curvature points without exploring different scales (Lee and Hollerer, 2007) (Figure 3.2).

Signature function (Ψ)**.** A signature function of a contour $\Psi(s_t)$ is defined as the combination of the *curvature entropy* along the contour $u(\kappa(s_t))$ and a distance transform $\delta(s_t)$, which represents the distances of every contour point to a centre of mass of the hand as depicted in Figure 3.2:

$$\Psi(s_t) = u(\kappa(s_t)) \cdot \delta(s_t)^{\gamma}. \tag{4}$$

The parameter γ weights the impact of the distance in the signature function. It also attenuates the high curvature points which are not a fingertip reducing the false positives mainly caused by noise. This allows us to reuse the function to localise the fingertip.

Hand posture descriptor. The signature function can be represented as a time series with variable length due to the different scales of contours in images. Once the Fourier series $a(n)$ and $b(n)$ are extracted from the signature function Ψ, a normalisation

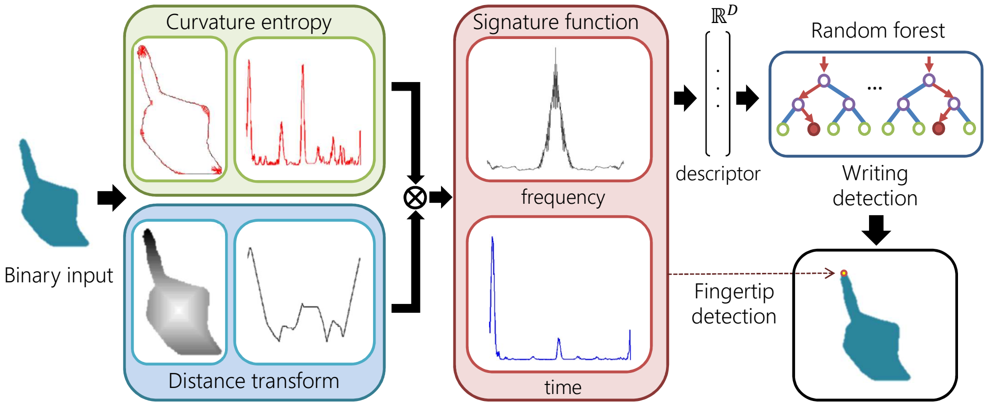

Figure 3.2: Hand posture and fingertip detection proposed approach.

step similar to Chen et al. (2003) is performed. This step makes the features invariant to rotation, translation and scale changes by defining the series:

$$S(n) = \frac{(a(n)^2 + b(n)^2)^{1/2}}{(a(1)^2 + b(1)^2)^{1/2}}, \tag{5}$$

which is sampled to conform the hand posture descriptor $\mathbf{x^h}_t = [S(1), ..., S(D)] \in \mathbb{R}^D$. The total number of samples (i.e. harmonics) D is determined experimentally and it is discussed on the experimental section.

Hand writing/no writing posture classifier. A standard decision forest (Breiman, 2001) with label space $\mathcal{Y} = \{\text{writing, no writing}\}$ is used as a classifier for the binary classification problem.

3.3.2 *Fingertip detection*

Fingertips have the property of being points of high curvature and distant from the hand centre. The signature function presents a peak on the fingertip position caused by a high curvature entropy value. This point is also highly distant from the centre of the hand, so it is kept by combining with the distance function, while false positive points mainly caused by noise are attenuated as shown in Figure 3.2.

The main advantage of the presented approach over other curvature-based ones (Lee and Hollerer, 2007; Pan et al., 2010) is that there is no need to examine several scales to find maximum curvature points and thus relieving of computational cost. The advantage over distance-based methods (Bhuyan et al., 2012) is that more accurate detections can be obtained in cases where the furthest point is not exactly the fingertip, which occurs when the user lightly bends their finger or in certain viewpoints as shown in Figure 3.9. To obtain smooth trajectories for the next stage of the algorithm a Kalman filter is used.

3.3.3 *Trajectory Hough forest*

In this section THF is presented to recognise and localise handwritten characters in mid-air depicted by fingertip trajectories. First, the feature extraction step is described. Second, the training stage of the forest is detailed. Finally, the term that encourages temporal consistency is introduced.

Spatio-temporal feature extraction. Information within a N-points sliding window, $W_t = \{\mathbf{p}_t, \dots, \mathbf{p}_{t+(N-1)}\}$, is encoded in the triplet $\mathbf{x}_t = [\mathbb{A}_t, \mathbb{C}_t, \mathbb{T}_t]$, where $\mathbb{A}_t$ is an non-parametric appearance term, $\mathbb{C}_t$ is a parametric term describing the curvature information and $\mathbb{T}_t$ encodes the temporal information within W_t (see Figure 3.3).

The **appearance term ($\mathbb{A}_t$)** is a $2 \times (N-1)$ dimension vector defined as follows:

$$\mathbb{A}_t = \mathop{\Big|\Big|}_{j=1}^{N-1} (\mathbf{p}_{t+j} - \mathbf{p}_t), \tag{6}$$

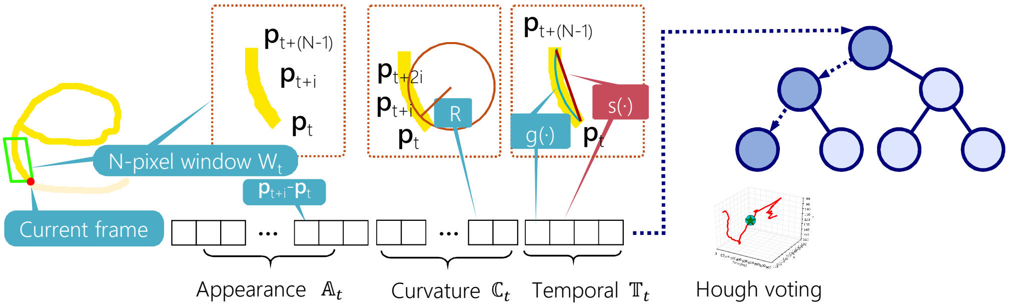

Figure 3.3: Spatio-temporal feature extraction for character recognition. It consists of three terms: appearance, curvature and temporal. The numbers in brackets indicate dimension of each term.

The **curvature term** $(\mathbb{C}_t)$ has the dimension of $\frac{N-1}{2}$. Menger curvature (Léger, 1999) is applied to capture the shape of the curvature within W_t. The aim is to approximate the curvature with a circle that is given by three points and then use reciprocal of the circle radius as representation. In pursuance of robustness, three points are selected from the curvature incrementally as follows:

$$\mathbb{C}_t = \mathop{\Big\|}\limits_{j=1}^{(N-1)/2} \frac{4Area(\mathbf{p}_t, \mathbf{p}_{t+j}, \mathbf{p}_{t+2j})}{|\mathbf{p}_t - \mathbf{p}_{t+j}||\mathbf{p}_t - \mathbf{p}_{t+2j}||\mathbf{p}_{t+j} - \mathbf{p}_{t+2j}|}, \tag{7}$$

where $Area(\mathbf{p}_t, \mathbf{p}_{t+j}, \mathbf{p}_{t+2j})$ is the area spanned by selected point triplet $(\mathbf{p}_t, \mathbf{p}_{t+j}, \mathbf{p}_{t+2j})$. The $Area(\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3)$ of three points $(\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3)$ is calculated as follows:

$$Area((\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3)) = |\frac{p_x^1(p_y^2 - p_y^3) + p_x^2(p_y^3 - p_y^1) + p_x^3(p_y^1 - p_y^2)}{2}|, \tag{8}$$

where p_x and p_y are the x and y coordinates of a point $\mathbf{p}$.

The **temporal term ($\mathbb{T}_t$)** is a 4-dimensional vector defined as:

$$\mathbb{T}_t = [g(\mathbf{p}_t, \mathbf{p}_{t+(N-1)}), s(\mathbf{p}_t, \mathbf{p}_{t+(N-1)}), \dot{g}(\mathbf{p}_t, \mathbf{p}_{t+(N-1)}), \dot{s}(\mathbf{p}_t, \mathbf{p}_{t+(N-1)})], \tag{9}$$

where $g(\cdot)$ stands for geodesic distance (i.e. along the writing trajectory of the fingertip), $s(\cdot)$ stands for Euclidean distance, $\dot{g}(\cdot)$ and $\dot{s}(\cdot)$ stand for velocity (in magnitude) in geodesic and Euclidean space respectively. This term can encode different temporal writing properties such as different stroke speeds depending on each character. By considering both the geodesic distance and Euclidean distance, this term can represent different stroke combinations (e.g. an arc after a straight line, a straight line or a circle, etc.).

Classification and localisation. Character recognition is formulated as a multiclass classification problem, while character centre localisation is formulated as regression. Hough forest (Gall et al., 2011) model is utilised as it is well-suited for the 26-class (26-character; $\mathcal{Y} = \{a, b, \dots, z\}$) problem. For each training sequence, the *character centres* $\{\bar{\mathbf{\Delta}}$ and $\bar{\mathbf{\Upsilon}}\}$ are calculated in the spatial and temporal domains respectively.

Each tree m in the forest $\mathcal{M}$ is constructed from a training set $S = \{(\mathbf{x}_t, \mathbf{d}_t^{spc}, \mathbf{d}_t^{tmp}, y_t)\}$ that is generated from the fingertip trajectories. $\mathbf{x}_t \in \mathbb{R}^{(2(N-1)+(N-1)/2+4)}$ is the input vector encoding the spatio-temporal features at frame t, $\mathbf{d}_t^{spc}$ and $\mathbf{d}_t^{tmp}$ are displacement vectors from the first point $\mathbf{p}_t$ of W_t to the spatio-temporal character centre respectively and $y_t \in \mathcal{Y}$ is the class label.

Consider a node i and a decision θ_i that consists of a threshold for a selected dimension of $\mathbf{x}_t$. According to θ_i, the instances in S_i are directed to its left or right child nodes, $2i + 1$ and $2i + 2$ respectively, as $S_{2i+1} = \{(\mathbf{x}_t, \mathbf{d}_t^{spc}, \mathbf{d}_t^{tmp}, y_t) \in S_i \mid f(\theta_i, \mathbf{x}_t) \leq 0\}$ and $S_{2i+2} = S_i \setminus S_{2i+1}$. The decision θ_i is chosen based on the minimisation of an objective function. The ideal decision θ_i is such that splits S_i to minimise the uncertainty of both class label and spatio-temporal displacement vectors. To this goal, three different objective functions to be minimised are defined as follows:

$$\text{Objective functions} = \begin{cases} E_c(\theta_i) = \sum_{n\in\{1,2\}} |S_{2i+n}|H(S_{2i+n}), \\ E_{rspc}(\theta_i) = \sum_{n\in\{1,2\}} ||(\mathbf{d}_t^{spc})^{2i+n} - \bar{\boldsymbol{\Delta}}^{2i+n}||^2, \\ E_{rtmp}(\theta_i) = \sum_{n\in\{1,2\}} ||(\mathbf{d}_t^{tmp})^{2i+n} - \bar{\boldsymbol{\Upsilon}}^{2i+n}||^2. \end{cases} \tag{10}$$

$H(\cdot)$ is the Shannon entropy computed over the class labels y_t in the training instances and $E_c(\theta_i)$ is denoted as *classification term*. $E_{rspc}(\theta_i)$ and $E_{rtmp}(\theta_i)$ are the *regression terms*, with $\bar{\boldsymbol{\Delta}}$ the spatial and $\bar{\boldsymbol{\Upsilon}}$ the temporal centres of each character respectively.

During training, a pool of candidate decisions $\{\theta_i\}$ is randomly generated at each node one and the one minimising one (randomly picked) of the above objective functions is stored. At the end of the tree growth, each leaf node $\ell \in \mathcal{L}$ stores the probability of the cropped trajectory W_t belonging to the class, estimated by the proportion of feature per class label reaching the leaf after training, and $[\mathbf{d}_t^{spc}, \mathbf{d}_t^{tmp}]$, the cropped trajectories' respective displacement vectors.

For prediction, $\mathbf{x}_t$ input vectors are passed through each tree. Starting at the root, the feature vector traverses the tree, branching left or right according to the split node function, until reaching a leaf node. Using the stored class distribution $\pi_{\ell(\mathbf{x}_t)}(y_t)$ and offsets at the leaf nodes, each leaf node votes for its corresponding class label and spatio-temporal centre location. The final class probability is averaged for all trees in the forest as follows:

$$p(y_t|\mathbf{x}_t) = \frac{1}{|\mathcal{M}|}\sum_m (\pi_{\ell(\mathbf{x}_t)}(y_t))^{(m)} \,. \tag{11}$$

Aggregating votes of all trees, the final class and centre position of the written trajectory are inferred. To find the centre point a mean shift mode seeking method is used (Comaniciu and Meer, 2002).

Encouraging temporal consistence in prediction. A drawback of using Equation 11 for inference is that it does not take consider previous predictions by the forest. As presented in the previous section, a Hough forest reduces both class and displacement uncertainty throughout the tree. The leaf nodes contain similar feature vectors both in displacement, feature space and category and thus it can be seen as clusters of similar patches. Such idea of using a decision forest for clustering is not new and it has been explored in other areas, such as semantic image segmentation (Moosmann et al., 2007; Shotton et al., 2008), but relatively less for action recognition (Yu et al., 2010). From this perspective, a spatio-temporal trajectory can be seen as a time-indexed sequence of codebook values.

Based on the above observation, we introduce a concept of *transition* between leaf nodes. Our hypothesis is that different classes of spatio-temporal trajectories have different temporal dynamics within the forest. For example, if we observe that at a given frame t the trajectory patch $\mathbf{x}_t$ has reached the node i while at the previous time step $t-1$ the corresponding patch $\mathbf{x}_{t-1}$ reached the node j, it can be quantified how *likely* is the transition from leaf node j to node i or, more formally, $p(\ell(\mathbf{x}_t) = i|\ell(\mathbf{x}_{t-1}) = j)$ for a certain class. We name this last term as *transition probability*, borrowed from the hidden Markov model (HMM) literature (Rabiner, 1989).

Shotton et al. (2008) showed that adding non-terminal nodes while constructing codebooks captured the hierarchical structure of the tree and led to an increased 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 transition matrix $\mathcal{A}^m(y,l)$ that encodes all transitions between nodes for a given class y and level l of a m tree in one time step (Figure 3.4). Rows of $\mathcal{A}^m(y,l)$ encode transition probabilities from node $i \in \mathcal{N}_l$ to all the rest of the nodes $j \in \mathcal{N}_l$ in a particular level l of the tree and they are normalised defining a probability distribution ($\sum_j p(n=j|n=i) = 1, i, j \in \mathcal{N}_l$).

To integrate this information into the predictions of the forest, the transition probability is treated as a prior probability $p(y_t)$, in a similar way to Shotton et al. (2008).

We want transitions to emphasise classes that are likely in a temporal context and reject unlikely ones. Given two temporal consecutive input vectors from a trajectory, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, both patches are passed through the forest, reaching different nodes through each tree of the forest. The prediction for $p(y_t|\mathbf{x}_t)$ is weighted with the prior probability as follows:

$$p'(y_t|\mathbf{x}_t) = p(y_t|\mathbf{x}_t)p(y_t), \tag{12}$$

with $p(y_t)$ defined as:

$$p(y_t) = \frac{1}{|\mathcal{M}|}\sum_m (\frac{1}{Z}\sum_l \mathcal{A}^m(y_t, l))^\alpha, \tag{13}$$

where Z is a normalising factor and α a constant that 'softens' the prior probability.

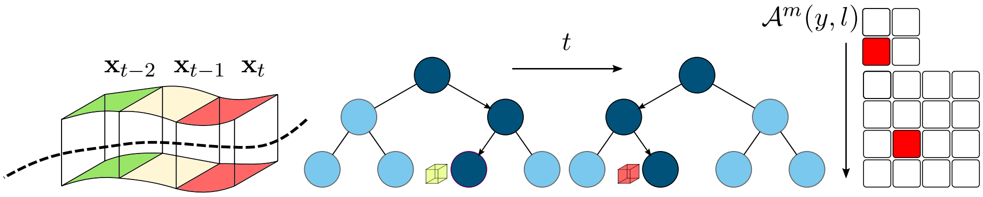

Figure 3.4: Procedure to construct a transition matrix. At current frame t the forest is fed with the current trajectory patch $\mathbf{x}_t$ of class y_t. The path through the forest, tree by tree, followed by $\mathbf{x}_t$ is compared with the one followed by $\mathbf{x}_{t-1}$. The transition matrix $\mathcal{A}(y_t, l)$ for the first two levels of the tree is shown. As there are two nodes on the first level and four on the second level, $\mathcal{A}(y_t, 1)$ is a 2-by-2 matrix and $\mathcal{A}(y_t, 2)$ a 4-by-4. In this example, $\mathbf{x}_{t-1}$ reached the 2nd node on the first level and the 3rd node on the second level. $\mathbf{x}_t$ reached the 1st node on the first level and the 2nd one on the second level. Thus, 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 at the second level are increased.

3.4 A DATASET FOR EGOCENTRIC FINGERTIP WRITING IN MID-AIR

In this section the recorded dataset to evaluate the proposed framework is introduced. The dataset is composed of depth video sequences containing fingertip written trajectories that represent the 26 English alphabet characters (from ‘a’ to ‘z’). An RGB-D sensor (Creative* Interactive Gesture Camera) is attached to a cap 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, the sequences are fully annotated with ground-truth fingertip positions after the detection and tracking stages to help research on this direction as well. Figure 3.5 displays some examples of the recorded sequences and Table 3.1 shows detailed statistics.

The hand posture dataset consists of 8,000 images from two classes: {‘writing’, ‘no writing’}. It has an approximate ratio of 1 : 3 for ‘writing / no writing’ containing challenging poses that naturally occur in egocentric vision, such as rotations out-of-plane of the hand, poses corrupted by noise and missing points due to the limitations of the sensor and the simple segmentation stage. The 2,500 images in ‘writing’ class have been manually labelled with fingertip positions.

Table 3.1: Characteristics of the proposed dataset

Classes	26	Total frames	15,792
Videos	260	Videos per class	10
Mean video frames	60.74	Resolution	320-by-240
Min. video frames	27	Max. video frames	154

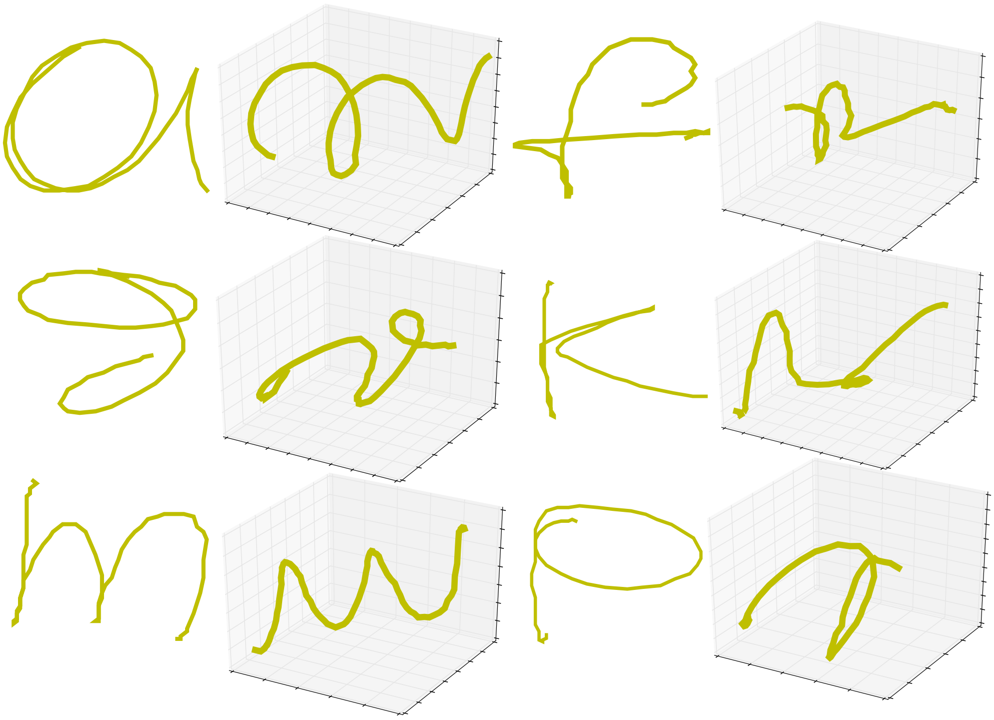

Figure 3.5: Examples of the proposed dataset of characters written in mid-air both projected in 2D space and in 3D space-time.

3.5 EXPERIMENTS

The proposed descriptor and fingertip detection are implemented on an Intel Core i7-2600 with 16 GB RAM in C++, and the THF is implemented in Python separately. Figure 3.8 shows captured images on different stages of the proposed framework.

3.5.1 *Hand posture recognition and fingertip detection*

Different experiments to test the proposed hand posture descriptors are performed. All the experiments have been done using 8,000 binary labelled images from our dataset. All the results presented in this section are with 10-fold cross validation using a standard random forest classifier (Breiman, 2001) and a resolution of 10 pixels in the computation of the curvature entropy.

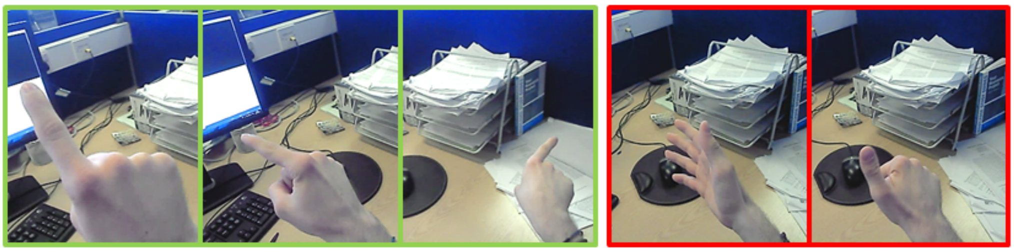

Figure 3.6: Examples of images from the introduced dataset when the user is writing (green) or not writing (red).

Hand posture recognition. The proposed approach is compared to one state-of-the art region-based method (Hu and Yin, 2013) and one contour-based method using FDs (Chen et al., 2003) extracted from contour coordinates. The proposed signature function is a combination of two signature functions: curvature entropy and distance to hand centre. For this reason, both functions individually are also tested to study the impact of their combination. Table 3.2 summarises the results for each approach varying the two important parameters of a decision forest classifier: tree number and maximum depth. The proposed descriptor shows a better performance over the baseline methods and over the individual signature functions for all the combinations of parameters. The best recognition accuracy for the proposed descriptor is achieved with an ensemble size of 40 trees and 18 levels as maximum depth.

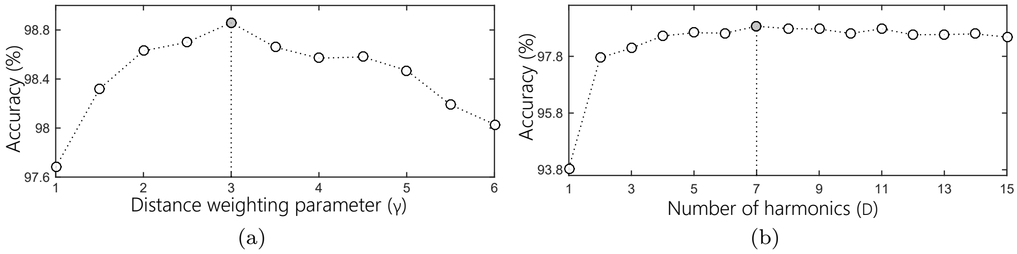

Figure 3.7: Parameter impact of the distance weighting parameter γ and the number of harmonics D on hand writing posture detection

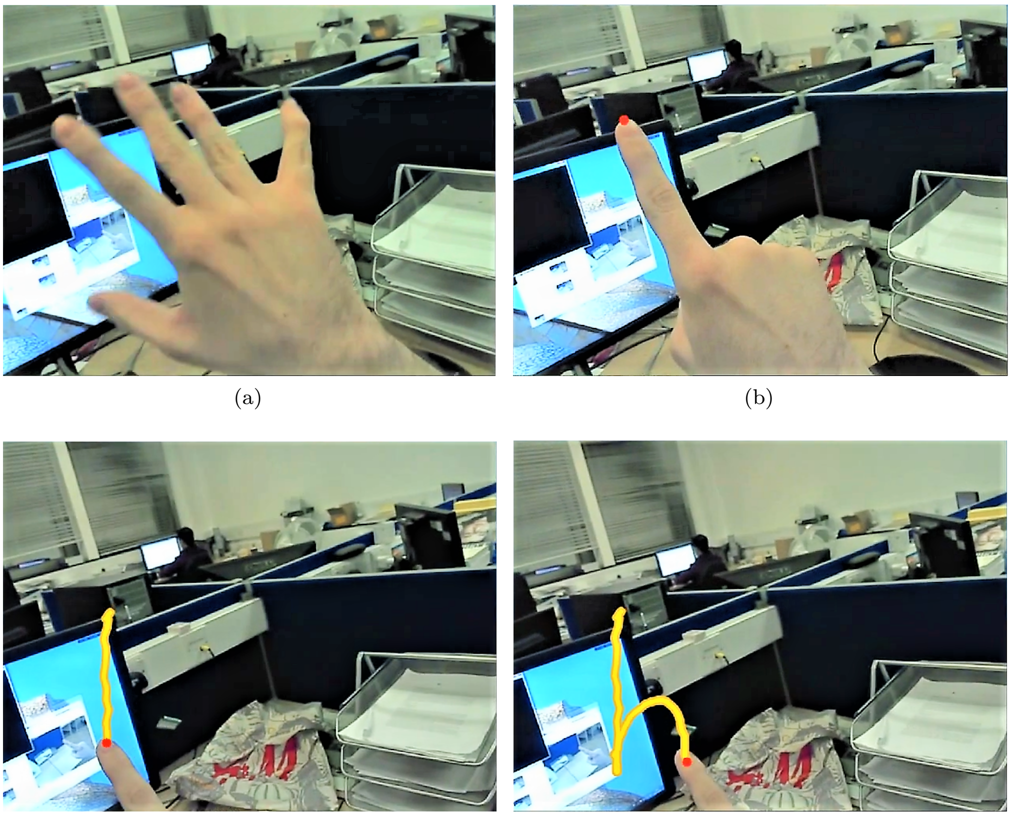

Figure 3.8: All figures show different stages of the framework in action. (a) a non-writing hand posture, no fingertip is detected and the system is in pause. (b) user starts to write, handwriting posture is detected. Fingertip is tracked in successive frames. (c) user in process of writing, when enough spatial-temporal points are buffered, on-line recognition starts. (d) user finished writing character and a ‘h’ is recognised.

Ablation experiments are performed to evaluate the influence of the parameters of the proposed descriptor: the distance weighting parameter γ and the number of harmonics extracted to conform the feature vector D. As shown in Figure 3.7b, only a small number of harmonics are needed to conform the feature vector, obtaining the highest accuracy with the first 7 of them. Using a higher number of harmonics does not improve the accuracy as all the information, in form of energy, is concentrated on the low frequencies of the spectrum as can be seen in Figure 3.2. The parameter γ describes approximately a quadratic function (Figure 3.7a) in terms of accuracy with a maximum found in 3.

Table 3.2: Hand posture recognition performance of different hand descriptors.

Max. depth	Num. of trees	Descriptor method				
		FD (Chen et al., 2003)	MSBNM (Hu and Yin, 2013)	Distance	Curvature entropy	Proposed
12	5	69.8	88.3	93.8	94.7	98.6
	10	71.8	87.8	95.9	94.9	98.7
	20	69.1	88.0	95.1	95.3	98.7
	30	69.4	88.2	95.8	95.4	98.8
	40	69.7	88.0	95.6	95.4	98.8
	50	68.6	88.4	95.8	95.3	98.9
14	5	72.4	88.4	93.8	94.7	98.6
	10	71.0	88.9	95.9	94.9	98.7
	20	69.8	89.1	95.1	95.3	98.7
	30	69.3	89.2	95.8	95.4	98.8
	40	69.1	89.2	95.6	95.4	98.8
	50	69.8	89.4	96.3	95.8	99.0
16	5	74.0	89.5	95.3	95.4	98.8
	10	72.4	89.6	95.9	95.8	98.8
	20	70.1	89.9	96.3	96.0	98.8
	30	71.1	90.0	96.7	96.0	99.0
	40	70.4	90.0	96.6	96.1	98.9
	50	70.5	90.2	96.4	96.1	98.9
18	5	**75.2**	89.8	96.1	95.5	99.0
	10	74.9	89.7	96.4	96.0	98.9
	20	71.6	90.3	96.6	**96.4**	98.9
	30	72.9	90.3	96.8	96.2	99.0
	40	72.9	90.4	96.9	96.3	**99.1**
	50	72.9	**90.4**	**97.2**	96.2	99.0

Fingertip detection. To test the proposed fingertip detection approach quantitatively the 2,500 manually labelled images from the proposed dataset are used. As a measure of error, the Euclidean distance between the estimated fingertip location $\hat{\mathbf{p}} = (\hat{p_x}, \hat{p_y})$ and the actual ground-truth $\mathbf{p} = (p_x, p_y)$ is computed; a detection is considered correct if its distance was less than 3 pixels to the ground-truth. The proposed approach is compared to two different methods. The first method (Bhuyan et al. (2012)) uses only the geodesic distance from the hand shape contour to the centre of the hand palm, without exploiting the curvature cue.

The second method is the one presented by Raheja et al. (2012), where fingertip points detection is tackled as edge detection of the hand binary shape. The results are presented on Table 3.3. The proposed approach outperforms compared approaches. The novel combination of curvature and distance information permits to have accurate estimations of fingertip positions in cases where using only distance information performs poorly (see Figure 3.9).

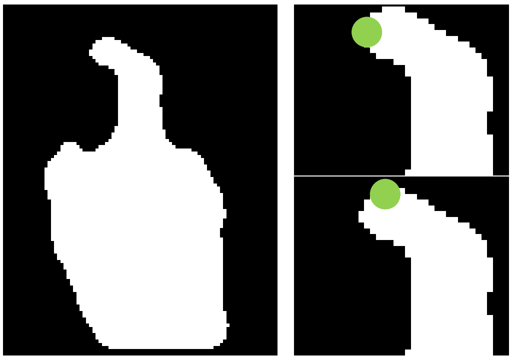

Figure 3.9: Qualitative fingertip detection results. Proposed method (top); distance-based method (bottom) (Bhuyan et al., 2012).

For this configuration, the computation time of extracting one descriptor, passing it through the forest and the fingertip detection was 2 ms on average. It can be observed from the results that the proposed hand posture recognition error is 0.9%, which compares favourably to the baselines. Furthermore, it can be concluded that both distance and curvature entropy cues are complimentary. On the other hand, fingertip detection error is 2.3%, measured as the percentage of frames where the prediction error is higher than a certain threshold (3 pixels in the experiment). In Figure 3.9 qualitative results and comparison to a distance-based method (Bhuyan et al., 2012) are depicted, showing the suitability of including the curvature term.

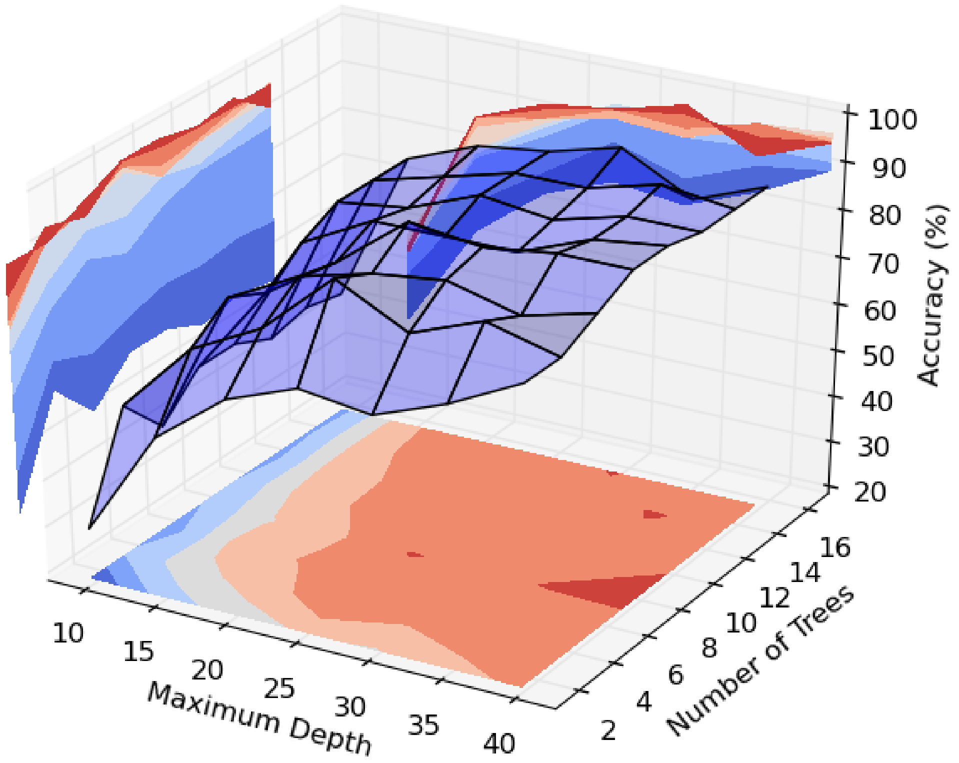

Figure 3.10: Different training parameters of THF vs. classification accuracy. A maximum at 8 trees of depth 25 is observed. Good accuracies are also obtained for deeper trees with significantly increased computational cost.

3.5.2 *Character recognition and localisation*

Character recognition. In this section the effect of several training parameters on classification accuracy is investigated. In Figure 3.10 the effect of the maximum depth and the number of trees on accuracy using 10-fold cross validation is depicted. For training, the window size N of W_t is set to 21 and on average 9,299 feature vectors are used for training and 1,033 for testing. Of all the parameters, the maximum depth appears to affect most significantly as it directly controls the model capacity of the forest. Based on the experimental results, we set the number of trees as 8 and the maximum depth as 25. Significant accuracies are also achieved for deeper trees (e.g. above 40) although it comes with a much higher computational cost due to the exponential nature of the forest.

In Table 3.3 the performance of different methods on the proposed dataset is shown. All the results have been obtained performing 10 leave-one-out cross validation (234 sequences for training and 26 for testing). Results of two classical algorithms for sequential data recognition, HMM and dynamic time warping (DTW) (Vikram et al., 2013; Chen et al., 2003) are included. Although none of these methods is suitable for the application of interest since they do not perform localisation, they are included for completeness. Furthermore, the table shows results for decision forest-based classifiers: a conventional random forest (Breiman, 2001) (RF), the forest without transition term and the full model. For all forest-based algorithms we fixed the number of trees to 8, maximum depth to 25 and a soften prior parameter $\alpha = 0.1$. It can be observed that the proposed THF performs better than the rest of the approaches. Introducing the prior probability slightly improves the accuracy by a 1.5%. Compared to the conventional random forest, it can be observed that the addition of localisation also helped classification, a similar observation found in Gall et al. (2011).

From the confusion matrix (Figure 3.11) it can be concluded 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 recognise even for humans. We believe that adding a broader temporal context could help on these cases.

Table 3.3: Performance comparison for fingertip detection and character recognition.

Problem	Method	Accuracy (%)
Fingertip detection	Distance-based (Bhuyan et al., 2012)	94.9
	Proposed	**97.7**
Character recognition	HMM (20 states) (Chen et al., 2003)	66.4
	DTW (Vikram et al., 2013)	78.5
	RF	79.6
	Proposed (no temporal term, no prior)	82.7
	Proposed (no prior)	90.4
	Proposed	**91.9**

Character centre localisation. The proposed method can also correctly localise spatio-temporal centre of each character writing by spatio-temporal offset Hough voting. Figure 3.12 shows localisation results in the 3D spatio-temporal space and it can be observed that estimated centres are similar to ground-truth ones. The writing centre

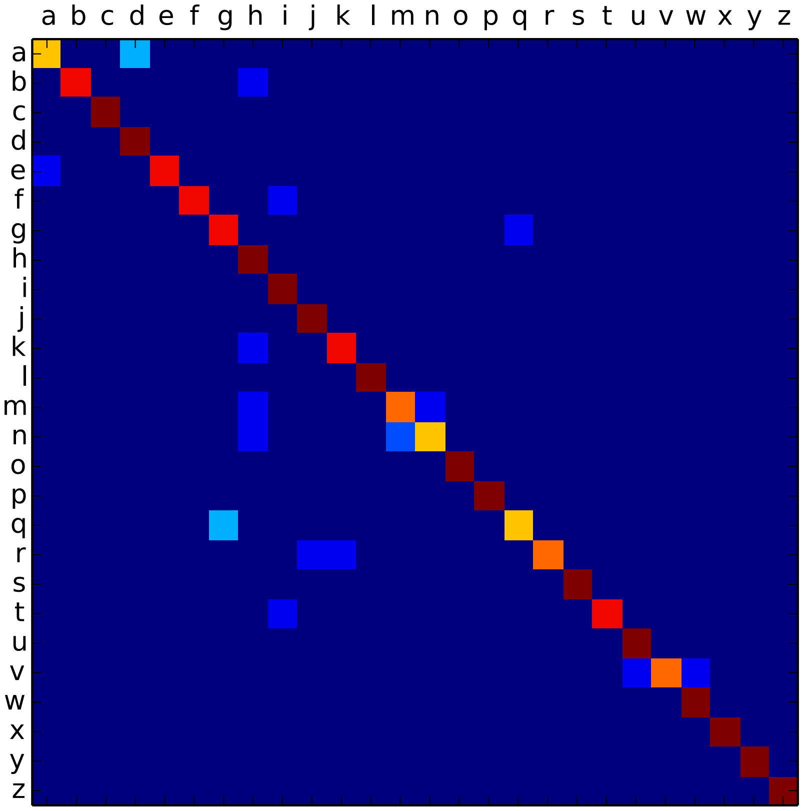

Figure 3.11: Confusion matrix of character recognition results by the proposed method.

information of each character can be used as an important to segment each character in a word or to anchor where the user wrote in an AR scenario.

3.5.3 *An experiment on RGB human action recognition*

To test the proposed THF in a more general setting, experiments are conducted on a public RGB benchmark: UT-interaction dataset (Ryoo and Aggarwal, 2010). The segmented set 1 of the dataset which contains 10 sequences per each class is used. The methodology recommended by the authors is followed and 10-fold leave-one-out cross validation to find the average performance is performed.

For tracking and extracting features along spatio-temporal trajectories on RGB video data, improved dense trajectories (Wang and Schmid, 2013) (IDT) are used. This approach is selected because of its excellent results and its publicly available code, however other spatio-temporal trajectory representation could be used. In Wang and Schmid (2013), each trajectory point is tracked at different scales using optical flow. Tracked points are sampled in small volumes and rich feature descriptors HOG, HOF and MBH are extracted. All this information is encoded in the trajectory patches $\mathbf{x}_t$. Different to Wang and Schmid (2013), the feature vectors of tracked points are not concatenated or averaged. Instead, each point of the trajectory is treated independently and stored as a patch. The trajectories are defined as ensembles of independent patches.

Table 3.4: Performance on UT-Interaction dataset of the proposed framework and extended to RGB scenarios compared to baselines and state-of-the-art approaches.

Method	Accuracy (%)
Yu et al. (2010)	83.3
Raptis and Sigal (2013)	93.3
Zhang et al. (2012)	**95.0**
Hough forest (cuboids) (Gall et al., 2011)	88.0
Proposed (no prior)	90.0
Proposed	**93.3**

In Table 3.4 the performance of the proposed method compared to baseline and other state-of-the-art methods is presented. The parameters for extracting trajectories were the recommended by Wang and Schmid (2013). A total of 15 patches per trajectory were generated. The Hough forest model using trajectory-based patches and the conventional Hough forest using dense cuboid sampling (Gall et al., 2011) are set as baselines. Forests parameters are $|\mathcal{M}| = 4$ and maximum depth 35. Using trajectory sampled descriptors instead of dense cuboids slightly improves the recognition accuracy. Furthermore, adding the proposed transition term further improves the performance, making it comparable to state-of-the-art performance. The proposed approach presents a similar result to Raptis and Sigal (2013), which used high level features (pose). Compared to Zhang et al. (2012), the proposed method performs worse likely because we rely on local spatio-temporal context, while Zhang et al. (2012) also considered long range spatio-temporal relations. To conclude, the result from Yu et al. (2010) where the clustering capability of a decision forest was also used is presented, indicating that important spatio-temporal information was lost on the histogram quantisation stage.

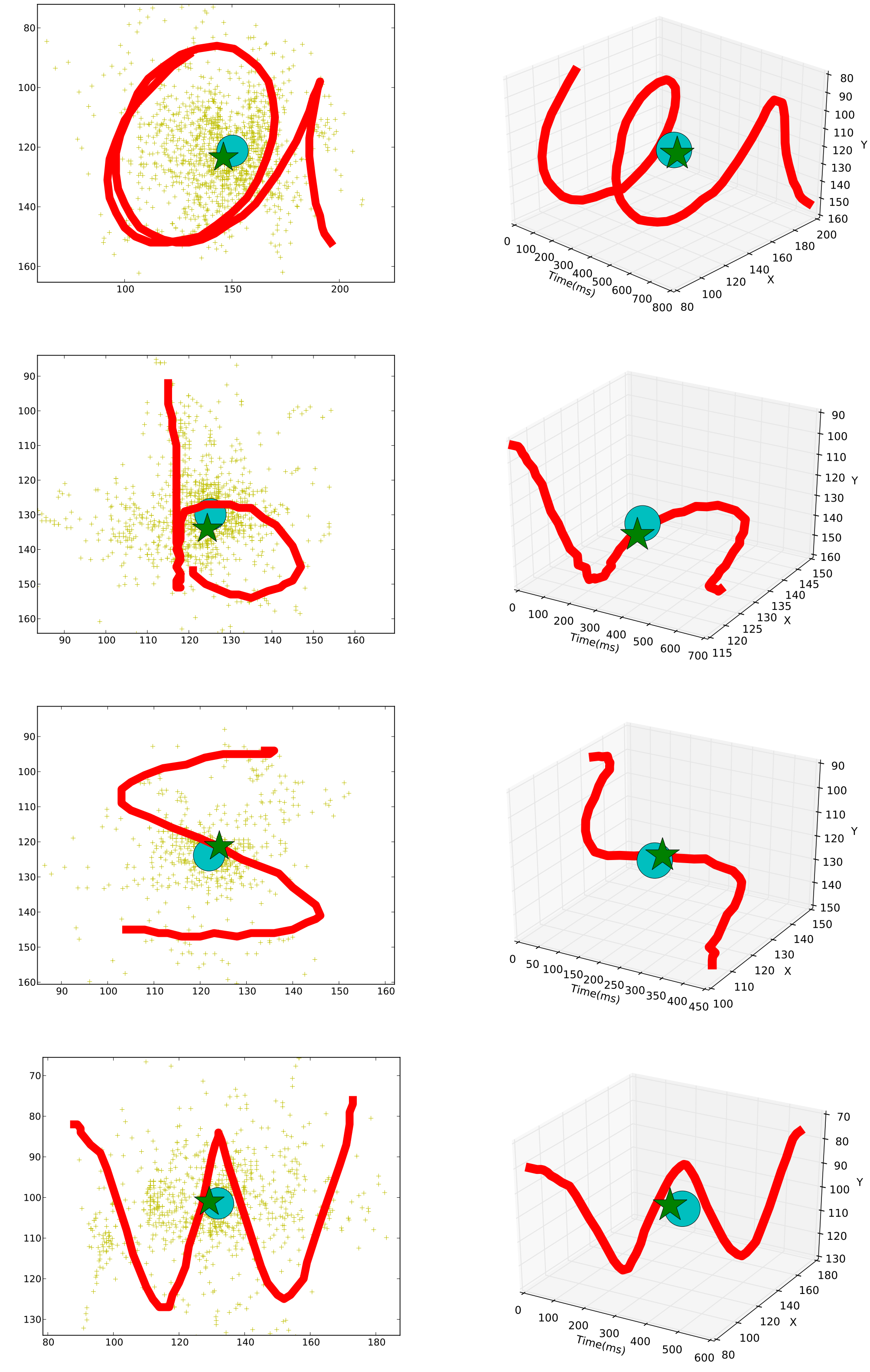

Figure 3.12: Character centre localisation results. Small yellow crosses are spatio-temporal offset voting points. Blue circles are estimated centre positions of each character and green stars indicate ground-truth centre locations.

3.6 SUMMARY

This chapter has presented trajectory Hough forest (THF) as a new framework for egocentric fingertip writing recognition. By introducing a new hand posture descriptor, we were able to improve simultaneous writing hand posture recognition and fingertip localisation compared to baselines. However, the system has some practical limitations due to the strong assumptions that we have made. For instance, we have assumed that the writing would be performed in mid-air instead of, for instance, upon a surface, which would require a more sophisticated hand segmentation module. Furthermore, we have assumed that a certain hand posture indicates whether the user is writing. Relaxing this assumption would complicate the problem of fingertip detection and temporal segmentation but be necessary to make the proposed system work outside of the lab.

The proposed forest algorithm also presents limitations. First, the size of the introduced transition matrices increased exponentially with the depth of the tree and linearly with the number of trees, which complicates its application to large datasets. Second, we only considered first-order time steps, and it is likely that we could improve the results with a longer temporal span. Finally, the transitions were simply estimated once the forest had already been trained, and they are thus not automatically learned within the forest. The next chapter of this thesis addresses these limitations.

4

TRANSITION FORESTS

4.1 OVERVIEW

In the previous chapter, we have presented the trajectory Hough forest (THF) as a variant of a Hough forest model (Gall et al., 2011) to encourage consistent temporal predictions. We have also noted a few drawbacks. First, the computed histograms at each level of the tree were computationally expensive to apply for big forests and larger datasets. Second, the temporal order under consideration was only limited to a one-time step. More importantly, the temporal dynamics were not learned inside the forest and only empirically estimated.

Section 2.5 has discussed approaches that use decision forests to deal with temporal dynamics and identified their limitations. Such approaches typically encode temporal dependencies that stack multiple frames (Fothergill et al., 2012), design hand-crafted temporal features (Zhu et al., 2013) and codebooks (Yu et al., 2010) and, as in the previous chapter, use Hough voting (Gall et al., 2011). We have also reviewed approaches to learning the temporal dynamics within the forest, such as the generative dynamic forest models approach (Lehrmann et al., 2014) (DFM) and the discriminative pairwise conditional random forests (Dapogny et al., 2015) (PCRF).

In this chapter, we propose a transition forest (TF) as an ensemble of randomised tree classifiers that discriminately learns both static pose information and temporal transitions. Temporal dynamics are learned while training the forest in addition to any temporal dependencies in the feature space, and predictions are made on the basis of previous predictions. The introduction of previous predictions complicates the learning problem as a consequence of the 'chicken and egg' paradox: making a decision in a node depends on the decision in other nodes, and vice versa. To tackle this problem, we outline a training procedure that iteratively groups pairs of frames that have similar associated frame transitions and class label in a given level of the tree. We combine both static and transition information by randomly assigning nodes to optimisation by classification or transition criteria. At the end of the tree growth, training frames that arrive at leaf nodes effectively represent a class label and associated transitions. We find that the addition of such temporal relation in training contributed to more robust single-frame predictions. The use of single frames mitigated the complexity and facilitated online predictions, which were two crucial conditions for the applicability of our approach to real-life scenarios.

Contributions

- The proposal of a new temporal decision forest model that can learn to discriminate both static frames and temporal transitions between pairs of frames.
- Exhaustive experiments in both action recognition and online action detection benchmarks according to our method and other decision forest baselines.

4.2 RELATED WORK

Skeleton-based online action detection

The detection of actions on streaming data (De Geest et al., 2016) has been less explored than the recognition of segmented sequences despite being more interesting in real scenarios. Early approaches (Fothergill et al., 2012) have included short sequences of frames or short motion information (Zanfir et al., 2013) to vote if an action is being performed. Sharaf et al. (2015) have proposed a similar approach that adds multi-scale information, while Meshry et al. (2016) have suggested a dynamic bag of features. Recently, Li et al. (2016) have introduced a more realistic dataset and baseline methods and demonstrated state-of-the-art performance with a classification and regression recurrent neural network (RNN), which Baek, Kim and Kim (2017) later improved through the use of RGB-D spatio-temporal contexts and decision forests.

4.3 TRANSITION FORESTS

Suppose we are given a training set S composed of temporal sequences of input-output pairs $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_t, y_t)\}$ where $\mathbf{x}_t$ is a frame feature vector and y_t is its corresponding action label (or background in detection setting). Our objective is to infer y_t for every given $\mathbf{x}_t$ using decision trees. On a decision tree, an input instance $\mathbf{x}_t$ starts at the root and traverses different internal nodes until it reaches a leaf node. Each internal node $i \in \mathcal{N}$ contains a binary split function f with parameters θ_i deciding whether the instance should be directed to the left or to the right child nodes.

Consider the set of nodes $\mathcal{N}_l \subset \mathcal{N}$ at a level l of a decision tree. Let S_i denote the set of labelled training instances $(\mathbf{x}_t, y_t)$ that reached node i (see Figure 4.1). For each pair of nodes $i, j \in \mathcal{N}_l$, we can compute the set of pairs of frames T_i^j that travel from node i to node j in d time steps as:

$$T_i^j = \{\{(\mathbf{x}_{t-d}, y_{t-d}), (\mathbf{x}_t, y_t)\} \mid (\mathbf{x}_{t-d}, y_{t-d}) \in S_i \wedge (\mathbf{x}_t, y_t) \in S_j\}, \tag{14}$$

where we term the set of pairs of frames T_i^j as *transitions from node i to j*. Note that T_i^j depends on frames that reached nodes i and j and time distance d. To capture different temporal patterns, we vary the distance d from one to a k-distant frame. In the following, we refer to parameter k as the *temporal order of the transition forest*.

In the example shown in Figure 4.1 we observe that the decision $f(\theta_0, S_0)$ is quite good as it separates S_0 in two sets, S_1 and S_2, in which one action label predominates. If we examine the transitions associated to this split, we see that we obtain two pure sets, T_1^1 and T_2^2, one mixed set T_2^1 and one empty set T_1^2. Imagine now that we observe the 'kick' frame in S_1 and we would have to decide based on this split, we would certainly assign the wrong label 'duck' with an uncertainty of $2/3$. Alternatively, if we check the previous observed frame (in S_2) and inspect its associated transition T_2^1, the uncertainty is now $1/2$ and thus we would be less inclined to make a wrong decision.

From the above example, we deduce that if we had obtained a better split and both child nodes were pure, we would certainly make a good decision by only looking at child nodes. However, good splits are difficult to learn if the temporal dynamics are not well

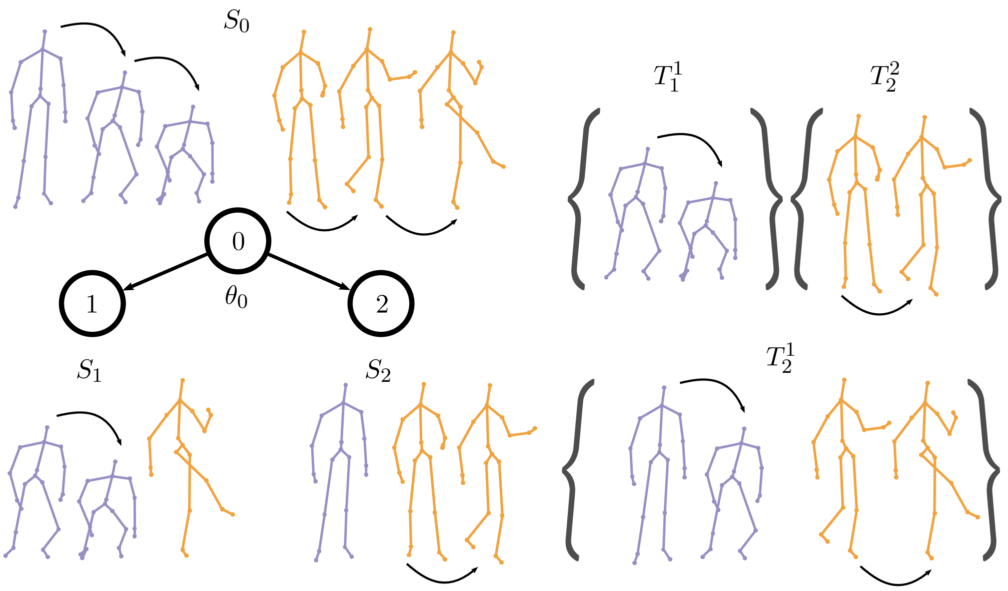

Figure 4.1: Consecutive frames representing two different actions (in purple 'duck', in orange 'kick') arrive at node 0. These frames are split in two different subsets S_1 and S_2 corresponding to child nodes 1 and 2. We compute the transitions as pairs of d-distant frames ($d = 1$ in this example) and we group them according to the route of each individual frame. T_1^1 and T_2^2 present only one transition, while T_2^1 two (one per class) and T_1^2 is empty. T_i^j are determined by θ_0.

captured on the feature space. On the other hand, if we had obtained a split that made transitions pure, we could also make a good decision. These observations motivate us to study how learning transitions between frames can help us to improve our predictions by introducing temporal information that was not available otherwise.

4.3.1 *Learning transition forests*

Our method for training a transition tree works by growing a tree one level at a time, similar to the entangled model of Shotton, Sharp, Kohli, Nowozin, Winn and Criminisi (2013) but limiting ourselves to standard binary trees. At each level, we randomly assign one splitting criterion to each node, choosing between classification and transition. The classification criterion maximises the class separation of static poses while the transition criterion groups frames that share similar transitions. As mentioned above, to maximise the span of temporal information learned, we learn transitions between d-distant pairs of frames, Equation 14, from previous frame up to the temporal order of the forest, k. For each tree, we randomly assign a value of d in the mentioned range and we keep it constant during the growth of that particular tree. For a total ensemble of $\mathcal{M}$ trees we will have subsets of trees trained with different d value: $\mathcal{M} = \mathcal{M}_1 \cup ... \cup \mathcal{M}_k$.

Consider a node $i \in \mathcal{N}_l$ and a decision θ_i. According to θ_i, the instances in S_i are directed to its left or right child nodes, $2i+1$ and $2i+2$ respectively, as $S_{2i+1} = \{(\mathbf{x}_t, y_t) \in S_i \mid f(\theta_i, \mathbf{x}_t) \leq 0\}$ and $S_{2i+2} = S_i \setminus S_{2i+1}$. Note that the split function f operates on a single frame, which will be shown important in the inference stage. After splitting, we can compute the sets of transitions between their child nodes $\{2i+1, 2i+2\} \subseteq \mathcal{N}_{l+1}$ as $\{T_{2i+m}^{2i+n}\}_{m,n\in\{1,2\}}$. Note that T_i^i is split in four disjoints sets, each one related to the combination of transitions associated to its child nodes. The decision θ_i is chosen based on the minimisation of an objective function.

Objective function. The objective function has two associated terms: one for single frame classification E_c and one for transitions between child nodes denoted as E_t. The classification term E_c is the weighted Shannon entropy of the class distributions over the set of samples that reach the child nodes $\{S_{2i+m}\}_{m\in\{1,2\}}$ as in standard classification forests. Willing to decrease the uncertainty of transitions while growing the tree, the transition term aims to learn node decisions in a way that subsets of transitions are purer in the next level. For a node $j \in \mathcal{N}_l$, the transition term is a function of the transitions between its child nodes and it is defined as follows:

$$E_t(\theta_j) = \sum_{m,n\in\{1,2\}} |T_{2j+m}^{2j+n}| H(T_{2j+m}^{2j+n}) \, , \tag{15}$$

where $T_{(\cdot)}^{(\cdot)}$ is defined in Equation 14 and $H(T_{(\cdot)}^{(\cdot)})$ is the Shannon entropy computed over the different label transitions. These two terms could be alternated or weighted-summed as single node optimisations. However, to reflect transitions between more distant nodes and capture further temporal information, we extend E_t to consider the set of all available nodes in a given level of a tree as shown in Figure 4.2 (a). For this, we randomly assign a subset of parent nodes N_c and N_t to be optimised by E_c and E_t respectively. Given that transitions between nodes depend on the split decisions at

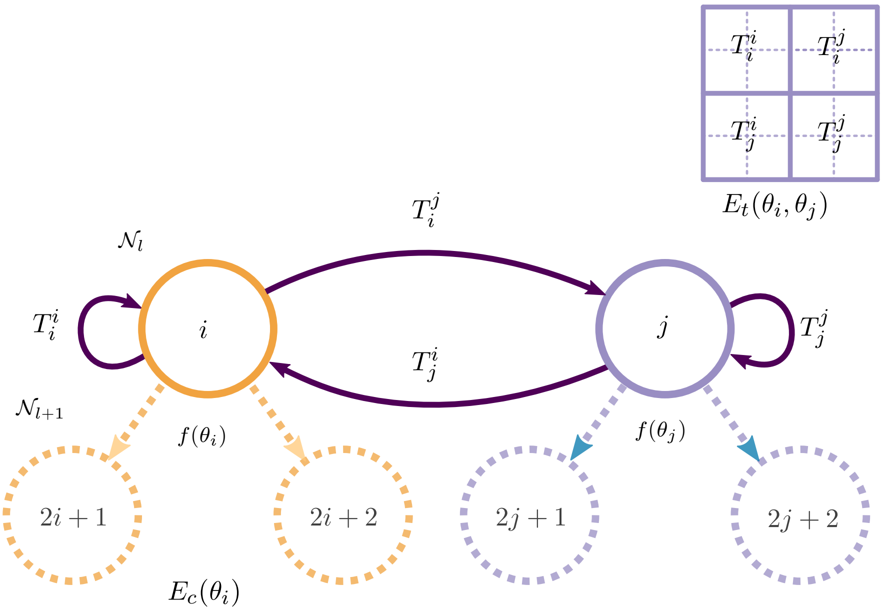

Figure 4.2: Growing a level l of a transition tree depends on all the node decisions θ_i and θ_j at the same time. Each T_i^j divides in four disjoint sets according to the different routes that a pair of samples can follow.

different nodes, the task of learning a level can be formulated as the joint minimisation of an objective function over the split parameters associated to the level nodes as:

$$\min_{\{\theta_i\}} E_c(\{\theta_i\}_{i \in N_c}) + E_t(\{\theta_i\}_{i \in N_c \cup N_t})\,. \tag{16}$$

Optimisation. The problem of minimising the objective function, Equation 16, is hard to solve. One could think of randomly assigning values to $\{\theta_i\}$ and pick the values that minimise the objective in a similar way to standard greedy optimisation in decision trees. However, the search space grows exponentially with the depth of the tree and evaluating E_t for all nodes and samples at the same time is computationally expensive. Our strategy to relieve these problems is presented in Algorithm 4.1. Given that E_c only depends on decisions in N_c nodes, we can optimise these nodes using the standard greedy procedure. Once optimised and fixed all nodes in N_c, we iterate over every node in N_t to find the split function that minimises a local version of E_t, denoted as E_t', that keeps all the split parameters fixed except the one of the considered node. It is defined for a node $j \in N_t$ and it depends on the transitions between its child nodes and all the transitions *from* and *to* these child nodes:

$$\begin{aligned} E'_t(\theta_j|\{\theta_i\}_{i \neq j \in N_c \cup N_t}) = \sum_{m,n \in \{1,2\}} & \overbrace{|T^{2j+n}_{2j+m}|H(T^{2j+n}_{2j+m})}^{\text{between j's child nodes (c.n.)}} \\ + \sum_{\substack{i \\ m,n \in \{1,2\}}} & \underbrace{|T^{2i+n}_{2j+m}|H(T^{2i+n}_{2j+m})}_{\text{from j's c.n. to i's c.n.}} + \underbrace{|T^{2j+n}_{2i+m}|H(T^{2j+n}_{2i+m})}_{\text{to j's c.n. from i's c.n.}} . \end{aligned} \quad (17)$$

The value of E'_t decreases (or does not change) at each iteration, thus indirectly minimising E_t. Following this strategy, it is not likely to reach a global minimum, but in practice we found that is effective to our problem. Note that computing Equation 17 needs the split parameters in other nodes to be available, forcing us to initialise them before the first iteration. We found that an initialisation of nodes using E_c helped the algorithm to converge faster than using a random initialisation relieving us of computational cost.

```
Input: Set of nodes $\mathcal{N}_l$ at level $l$ and temporal order $d$
Output: Set of split function parameters $\{\theta_i\}$
procedure LearnLevel($\mathcal{N}_l$)
    randomly assign nodes in $\mathcal{N}_l$ to $N_c$ and $N_t$
    for all $i \in N_c$ do
        optimise $N_c$ using $E_c$
        save and fix $\theta_i$
    end for
    initialise $\{\theta_j\}$ for $j \in N_t$
    while something changes do
        for all $j \in N_t$ do
            $\Theta \leftarrow$ random feature/threshold selection
            $\theta_j \leftarrow \text{argmin}_{\theta' \in \Theta} E'_t(\theta_j|\{\theta_i\}_{i \neq j \in N_c \cup N_t})$
        end for
    end while
end procedure
```

Algorithm 4.1: Learning level l of a transition tree

Discussion on training strategy. We made various decisions when designing the training and inference stage of our model. When training a transition forest, we passed the order of the forest k as a parameter. This parameter can be viewed as a truncation of the number of previous time steps from which the forest will try to learn transitions. When growing a particular tree, its temporal order was randomly assigned to a number between one and k, and it remained constant for all nodes in that tree (denoted as d in Algorithm 4.1). On average, we had the same number of trees for each time order, and we thus forced the forest to learn equally for all temporal distances. Although we did not explore this direction in this thesis, other training strategies are plausible. For instance, regarding the standard training strategy of recurrent neural networks whereby gradients for different time steps are combined when optimising the network weights, one could similarly compute and add E_t for all values between one and k and choose the split value that minimises the total objective function for all time steps. This strategy would significantly increase the computational cost of training a tree, as we would need to evaluate and memorise the objective function for each time order, which, as the previous section has mentioned, is the computational bottleneck of transitions. Another plausible training strategy would be to randomise the temporal order and include it in the parameter pool Θ, which would ensure that the temporal distance is learned; in other words, we could seek the value d that maximises the objective function. This task would require a rethinking of Equation 19 and potentially induce normalisation issues when deciding on the optimal d value. We believe this could be an interesting means to extend the transition forest, as decision trees models usually benefit from further randomisation (Geurts et al., 2006).

4.3.2 *Inference*

Restricting ourselves to the set of leaf nodes $\mathcal{L}$, we assign each transition subset $\{T_i^j\}_{i,j\in\mathcal{L}}$ a conditional probability distribution over label transitions denoted $\pi_i^j(y_t|y_{t-d})$. This is different from classification forests where the *classification probability* $\pi_i(y_t)$ is estimated over all the set of training instances S_i that reached the leaf node i. Instead, we focus on subsets of transitions that depend on the leaf node (prediction) that previous d-distant frame reached. Note that the split function f is defined for a single frame, enabling us to perform individual frame predictions. For an ensemble of $\mathcal{M}_d$ transition trees, we define a prediction function given two d-distant frames:

$$p_d(y_t|\mathbf{x}_t, \mathbf{x}_{t-d}, y_{t-d}) = \frac{1}{|\mathcal{M}_d|} \sum_{m\in\mathcal{M}_d} (\pi_{\ell(\mathbf{x}_t)}^{\ell(\mathbf{x}_{t-d})}(y_t|y_{t-d}))^{(m)}\,, \tag{18}$$

where $\ell(\mathbf{x}_t)$ and $\ell(\mathbf{x}_{t-d})$ are the leaf nodes reached by $\mathbf{x}_t$ and $\mathbf{x}_{t-d}$ at m-th tree respectively. We name this probability as *transition probability.* We combine the transition probability for different previous pairs of frames up to k with the *classification probability* (see Figure 4.2 (b)). Combining the static classification probability with the temporal transition probability defines our final prediction equation for a transition forest of temporal order k:

$$\begin{split} p(y_t|\mathbf{x}_t, \mathbf{x}_{t-1}, ..., \mathbf{x}_{t-k}, y_{t-1}, ..., y_{t-k}) = \\ \frac{1}{|\mathcal{M}|}\sum_m (\pi_{\ell(\mathbf{x}_t)}(y_t))^{(m)} \frac{1}{k} \sum_{1\le d\le k} p_d(y_t|\mathbf{x}_t, \mathbf{x}_{t-d}, y_{t-d})\,. \end{split} \tag{19}$$

For each frame $\mathbf{x}_t$ we obtain a probability of the frame belonging to one action (plus background in detection setting) based on k previous predictions. In the *action recognition* setting we average the per-frame results to predict the whole sequence. On the other hand, for *online action detection*, we define two thresholds, β_s and β_e, to locate the start and the end frame of the action. When the score for one action exceeds β_s, we aggregate the results since the start of the action and we do not allow any action change until the score is less than β_e.

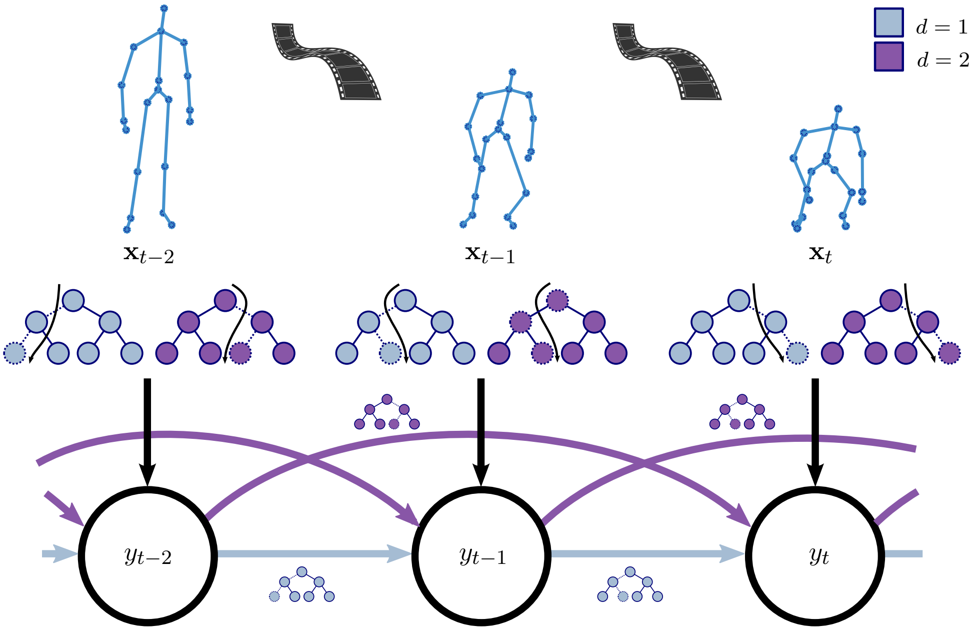

Figure 4.3: In inference, each individual frame is passed down the forest and static pose classification is combined with transition probability. Transition probability is computed using the trees trained for specific d-distant frames (shown in different colour). In this example $k = 2$ and $|\mathcal{M}| = 2$.

4.3.3 *Implementation details*

If the training data is not enough, we may encounter empty transition subsets at low levels of the tree. For this reason, we set a minimum number of instances needed to estimate their probability distribution and we empirically set this parameter to ten in our experiments. This parameter is conceptually the same as the stopping criterion of requiring a minimum number of samples to keep splitting a node.

4.4 EXPERIMENTAL EVALUATION

In the following we present experiments to evaluate the effectiveness of the transition. We start evaluating the proposed model for action recognition task and we follow with the evaluation on online action detection task. In all experiments we performed standard pre-processing on given joint positions similar to (Vemulapalli et al., 2014) making them invariant to scale, rotation and point of view.

4.4.1 *Baselines*

We compare the transition forest with five different forest-based baselines detailed next. For fair comparison, we always use the same number of trees in all methods and we adjust the maximum depth for best performance.

Random forest (Breiman, 2001) (RF). To assess how well a decision forest performs while only using static information, we implement a single frame-based random forest only using E_c.

Sliding window forest (Fothergill et al., 2012) (SW). To compare our learning of temporal dynamics with the strategy of stacking multiple frames, we implement a forest using the sliding window setting in which the temporal order k the number of previous frames in the window.

Dynamic forest model (Lehrmann et al., 2014) (DFM). To compare our discriminative forest approach with a generative one, our third baseline is a generative forest, where k is the order of their non-linear Markov model. With no public implementation available, we directly report results in Lehrmann et al. (2014).

Pairwise conditional random Forest (Dapogny et al., 2015) (PCRF). To assess the discriminative pairwise information, we implement a pairwise forest similar to the one used for expression recognition by Dapogny et al. (2015). We grow and combine classification trees for different pairwise temporal distance up to k.

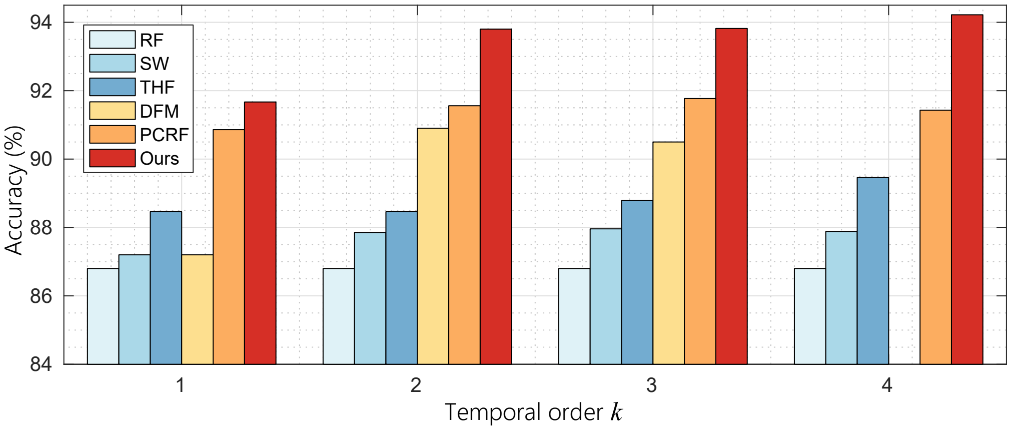

Figure 4.4: Temporal order k for different baselines and our approach on MSRC-12 dataset. The proposed TF performs better for all temporal orders. The temporal order defines the number of previous time steps considered to make a prediction.

Trajectory Hough Forest (Chapter 3) (THF) To compare with a temporal regression method, we implement the THF presented in the previous chapter and adapt the colour trajectories to poses and the histograms to deal with a temporal order of k.

4.4.2 *Action recognition experiments*

We evaluate the proposed algorithm on three different action recognition benchmarks: MSRC-12 (Fothergill et al., 2012), MSR-Action3D (Li et al., 2010) and Florence-3D (Seidenari et al., 2013). First, we perform detailed control experiments and parameter evaluation on MSRC-12 dataset. Next, we evaluate our approach comparing with baselines and state-of-the-art on all datasets.

MSRC-12 experiments

The MSRC-12 (Fothergill et al., 2012) dataset consists of 12 iconic and metaphoric gestures performed by 30 different actors. We follow the experimental protocol in (Lehrmann et al., 2014): only the 6 iconic gestures are used, making a total of 296 sequences and we perform 5-fold leave-person-out cross-validation, i.e. 24 actors for training and 6 actors for testing per fold.

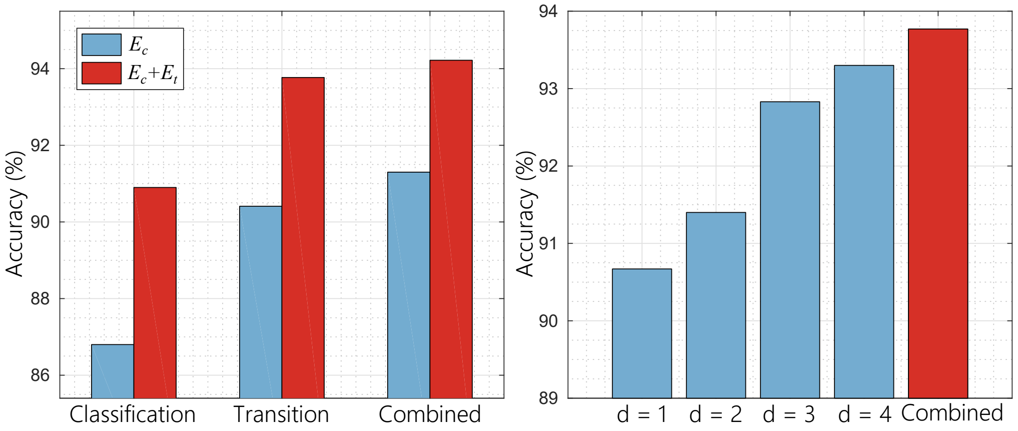

Figure 4.5: (a) E_c vs $E_c + E_t$ and terms in Equation 19. (b) contribution of different d order trees to transition probability shown in (a) and defined in Equation 18 on MSRC-12.

Temporal order k and comparison with baselines. In Figure 4.4 we show experimental results varying the temporal order parameter k for all approaches. We observe that using only static information on single frames (RF) to recognise action is limited and it can be improved by stacking multiple frames (SW). Adding a regression term as in THF helps to increase the accuracy. DFM uses the same exact input window as SW, while being more robust because of their explicit modelling of time. Better than the rest of baselines, PCRF shows that capturing pairwise information is effective to model the temporal dynamics of the actions. On the other hand, TF shows the best performance for all temporal orders. This shows that both combining static and temporal information in a discriminative way is very effective. In the next two paragraphs we analyse the contribution of both sources of information.

Discriminative power of learned transitions. We measure the impact of the transition training procedure presented in Section 4.3.1. For this, we train two different transition forests, one using only E_c and one using E_c and E_t. For each forest, we show the performance by breaking down the terms of Equation 19: (i) using only the classification probability; (ii) using only the transition probability (Equation 18); (iii) combining both terms (Equation 19).

Results are shown in Figure 4.5 (a). We observe that our proposed training algorithm increases the performance of both static and transition terms, leading to an important overall improvement. The static classification term improves substantially, meaning that

Table 4.1: MSRC-12: Comparison with state-of-the-art using different frame representations.

Method	Real-time	Online	Acc. (%)
DFM (Lehrmann et al., 2014)	✓	✓	90.90
ESM (Jung and Hong, 2014)	✗	✗	**96.76**
Riemann (Devanne et al., 2015)	✗	✗	91.50
PCRF (Dapogny et al., 2015)	✓	✓	91.77
Bag-of-poses (Zhu, Zhang, Shen and Song, 2016)	✗	✗	94.04
Proposed (JP)	✓	✓	**94.22**
Proposed (RJP)	✓	✓	**97.54**
Proposed (MP)	✓	✓	**98.25**

E_t helps to separate categories on the feature space by introducing temporal information that was not available otherwise. In Figure 4.5 (b) we show the contribution of each temporal distance to the overall transition probability in Equation 18.

Frame representation. In addition to joint positions (JP) from above experiments, we experimented with two different frame representations: one static and one dynamic. The static one consists of pairwise relative distance of joints (Vemulapalli et al., 2014) (RJP), proven to be more robust than JP while being very simple. The dynamic one, named moving pose descriptor (Zanfir et al., 2013) (MP), incorporates temporal information by adding velocity and acceleration of joints using nearby frames. In Table 4.1 we observe that RJP and MP perform similarly well performing better than JP, showing that the TF can benefit of different static and dynamic feature representations.

Initialisation. We initialised the transition nodes N_t in two ways: randomly and using E_c. We found that the latter initialisation provided slightly better results by 0.35% after ten iterations. However, after doubling the number of iterations, the difference was reduced to 0.07%, leading to the conclusion that our algorithm is robust to initialisation, but correctly initialising reduces the training time. Based on this, we limited the number of iterations to ten.

Ensemble size. A single tree of maximum depth 10 gave us an accuracy of 86.42%, six trees 93.10% and twelve 94.22%. As a tree-based algorithm, adding more trees is expected to increase the performance (up to saturation) at the cost of computational time.

Comparison with the state-of-the-art. In Table 4.1 we compare the proposed approach with the state-of-the-art. We observe that using the simple JP representation, we achieve the best except for 'enhanced sequence matching' (Jung and Hong, 2014) (ESM). However, ESM uses a slow variant of DTW and MP representation. Using both RJP and MP representation our approach achieves the best performance while being able to run in real time (1,778 fps).

MSR-Action3D experiments.

The MSR-Action3D (Li et al., 2010) dataset is composed of 20 actions performed by 10 different actors. Each actor performed every action two or three times for a total of 557 sequences. We perform our main experiments following the setting proposed by Li et al. (2010). In this protocol, the dataset is divided into three subsets of eight actions, named AS1, AS2 and AS3. The classification is performed on each subset separately and the final classification accuracy is the average over the three subsets. We perform a cross-subject validation in which half of the actors are used for training and the rest for testing using ten different splits. We use RJP frame representation, $k = 4$ and 50 trees of maximum depth 8.

Baselines and state-of-the-art comparison are shown in Tables 4.2 and 4.3 respectively. The proposed approach achieves better performance than all baselines. Offline state-of-the-art methods (Zhang, Wang, Gou, Sznaier and Camps, 2016; Wang, Wang and Yuille, 2016) achieve the best performance. Focusing on methods that are both real-time and online, the best performance is achieved by HURNN-L (Du et al., 2015), which uses a deep architecture to learn an end-to-end classifier. We obtain better results than (Du et al., 2015) on both their online and offline flavours.

Some authors (Zanfir et al., 2013; Veeriah et al., 2015) show results using a different protocol (Wang, Liu, Wu and Yuan, 2012) in which all 20 actions are considered. For comparison, using this protocol we achieved an accuracy of 92.8%, which is superior to state-of-the-art online approaches of MP, 91.7%, and dLSTM (Veeriah et al., 2015),

Table 4.2: Comparison with forest-based baselines.

Method	MSRC-12	MSR-Action3D	Florence-3D
RF	86.83	87.77	85.46
SW	87.81	90.48	88.44
THF (Chapter 3)	89.46	91.31	89.06
DFM	90.90	-	-
PCRF	91.77	92.09	91.23
Proposed	**94.22**	**94.57**	**94.16**

92.0%, but inferior to the offline approach of Gram matrix (Zhang, Wang, Gou, Sznaier and Camps, 2016), 94.7%. It is important to note that the inference complexity of both Zanfir et al. (2013) and Zhang, Wang, Gou, Sznaier and Camps (2016) increases with the number of different actions, which is not the case of our approach, making it more suitable for realistic scenarios. Zhang, Wang, Gou, Sznaier and Camps (2016) reported an inference time (ten runs over whole testing set) of 1,523 seconds, for the same setting we report a significantly lower time of 289 s.

Table 4.3: State-of-the-art comparison on MSR-Action3D dataset.

Method	Real-time	Online	AS1 (%)	AS2 (%)	AS3 (%)	Avg (%)
Zhu et al. (2013)	✗	✗	-	-	-	90.90
Vemulapalli et al. (2014)	✗	✗	95.29	83.87	98.22	92.46
Du et al. (2015)	✓	✗	93.33	94.64	95.50	94.49
Wang, Yuan, Hu, Li and Zhang (2016)	✗	✗	93.75	95.45	95.10	94.77
Zhang, Wang, Gou, Sznaier and Camps (2016)	✓	✗	98.66	94.11	98.13	96.97
Wang, Wang and Yuille (2016)	✓	✗	-	-	-	**97.44**
Dapogny et al. (2015)	✓	✓	94.51	85.58	96.18	92.09
Du et al. (2015)	✓	✓	92.38	93.75	94.59	**93.57**
Proposed	✓	✓	96.10	90.54	97.06	**94.57**

Table 4.4: Optimising transitions reduces the class uncertainty for both classification and transition probabilities leading to more robust predictions by the proposed forest.

Training	Prob.	Mean entropy (bits)	Accuracy (%)
E_c	Class.	0.5006	70.27
$E_c + E_t$	Class.	0.4454	75.68
E_c	Trans.	0.1815	81.92
$E_c + E_t$	Trans.	0.0752	91.89

Evaluation of forest leaf nodes and their transitions. The transition forest aims to produce *pure* probability distributions on both classification leaf nodes and their associated transitions. To evaluate the quality of the resulting leaf posteriors in both classification and transition probabilities, we learn two different transition forests, one using E_c and the other one using $E_c + E_t$. We compute the mean entropy of leaf nodes and their associated transitions. To have a significant number of leaf nodes, we grow a forest with 500 trees of maximum depth 8 and $k = 4$ on a random training/testing subject split on MSR-Action3D AS2. The results are shown in Table 4.4. The highest entropy for a 8-class problem would result from an uniform distribution and the value would be 3 bits. We observe that the mean entropy of leaf nodes in $E_c + E_t$ forest is lower than in E_c and thus purer. This supports the results obtained in the previous section where we observed that adding E_t helped to obtain better results by only using the classification probability. On the other hand, the mean entropy of transitions between leaf nodes is lower in both forests, which is consistent with the obtained higher accuracies by using transition probabilities. For the $E_c + E_t$ forest, the mean entropy is less than a half of the one in E_c, which is coherent with the fact that we are directly optimising the transitions subsets an thus making them more pure while growing the tree. For all the results, we observe that lower mean entropy values lead to higher accuracies.

Visualisation and details of our proposed optimisation. In Figure 4.6 we show how our proposed objective function develops as a function of the training iterations at the last level of the tree with the proposed initialisation. To plot the figure, we used the same experimental setting as in the previous section and we show the averaged value of the summation of Equation 17 for all level nodes over ten randomly selected trees. We observe that the function value rapidly decreases with the number of iterations until

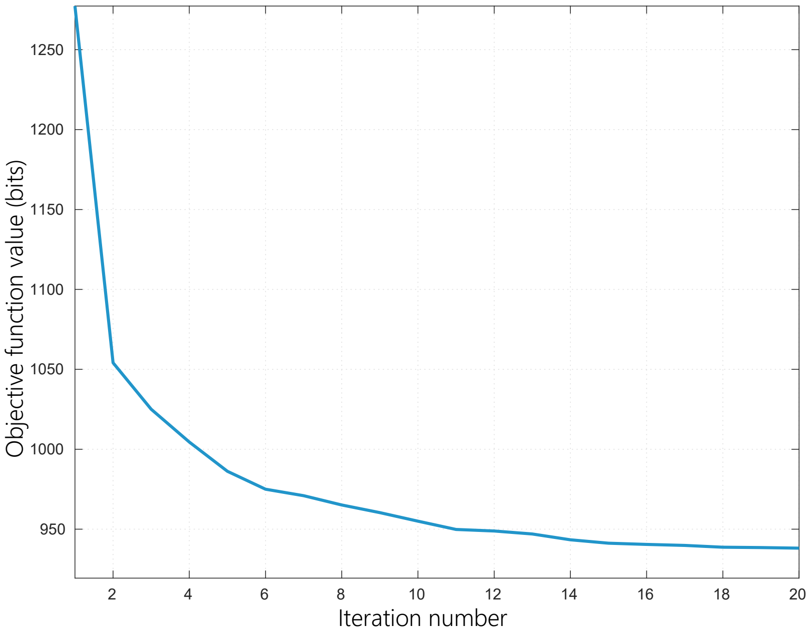

Figure 4.6: Value of objective function by number of iterations for one level of the tree.

it converges. We evaluated the accuracy for one, ten and twenty iterations, obtaining 88.2%, 91.89% and 92.01% respectively. Increasing the number of iterations after ten does not significantly increase the accuracy, while it increments the training time (7.92 seconds/tree for ten iterations, 15.44 seconds/tree for twenty, evaluating 100 random features/thresholds per node). Additionally, we believe that a very high number of iterations would reduce the randomness of individual trees making the ensemble more prone to overfitting.

Florence-3D experiments

The Florence-3D dataset (Seidenari et al., 2013) consists of 9 different actions performed by 10 subjects. Each subject performed every action two or three times making a total of 215 action sequences. Following previous work (Wang, Wang and Yuille, 2016; Wang, Yuan, Hu, Li and Zhang, 2016), we adopt a leave-one-subject-out protocol, i.e. nine subjects are used for training and one for testing for ten times. We used the same parameters as in the previous experiment.

Table 4.5: State-of-the-art comparison on Florence-3D dataset.

Method	Real-time	Online	Acc. (%)
Seidenari et al. (2013)	✗	✗	82.15
Vemulapalli et al. (2014)	✗	✗	90.88
Dapogny et al. (2015)	✓	✓	91.23
Vemulapalli and Chellappa (2016)	✗	✗	91.40
Wang, Yuan, Hu, Li and Zhang (2016)	✗	✗	91.63
Wang, Wang and Yuille (2016)	✓	✗	92.25
Proposed	✓	✓	**94.16**

We compare the proposed approach with baselines and state-of-the-art in Tables 4.2 and 4.5 respectively. We can see that our approach achieves the best performance over all baselines and state-of-the-art. Note that on this dataset we outperform the recent Key-poses approach (Wang, Wang and Yuille, 2016), which achieved the best performance on MSR-Action3D dataset.

Fingertip writing in mid-air experiment

We performed an additional experiment in the dataset presented in Chapter 3, using the same frame representation as in THF and same parameters as in previous experiment. We obtained a result of 94.2%, which is considerable better than the result of 91.9% obtained by THF. Note that TF does not perform spatio-temporal regression and performance might be improved by adding a regression term, with the cost of losing computational efficiency and online capability.

4.4.3 *Online action detection experiments*

We end our experimental evaluation in a more realistic scenario. We test the proposed transition forest for online action detection on the very recently proposed Online Action Detection (OAD) dataset Li et al. (2016). The dataset consists of 59 long sequences containing 10 different daily-life actions performed by different actors. Each sequence contains different action/background periods of variable length in arbitrary order annotated with start/end frames. We use the same splits and evaluation protocol as Li et al. (2016). Previous work Li et al. (2016) fixed the number of considered previous frames to 10, in consequence we set $k = 10$. We use RJP representation and 50 trees of maximum depth 20. Thresholds β_s and β_e were empirically set to 0.79 and 0.16 respectively.

Table 4.6: Performance comparison on Online Action Detection (OAD) dataset.

	Baselines			State-of-the-art		
				RNN	JCR-RNN	
Action	RF	SW	PCRF	(Zhu, Lan, Xing, Zeng, Li, Shen and Xie, 2016)	(Li et al., 2016)	Proposed
drinking	0.598	0.387	0.468	0.441	0.574	**0.705**
eating	0.683	0.590	0.550	0.550	0.523	**0.700**
writing	0.640	0.678	0.703	**0.859**	0.822	0.758
open cupboard	0.367	0.317	0.303	0.321	**0.495**	0.473
washing hands	0.698	**0.792**	0.613	0.668	0.718	**0.740**
open microwave	0.525	0.717	0.717	0.665	0.703	**0.717**
sweeping	0.539	0.583	0.635	0.590	0.643	**0.645**
gargling	0.298	0.414	0.464	0.550	0.623	**0.633**
throwing trash	0.340	0.205	0.350	**0.674**	0.459	0.518
wiping	0.823	0.765	0.823	0.747	0.780	**0.823**
Overall	0.578	0.556	0.607	0.600	0.653	**0.712**
SL	0.361	0.366	0.378	0.366	0.418	**0.514**
EL	0.391	0.326	0.412	0.376	0.443	**0.527**
Inference time (s)	0.59	0.61	3.58	3.14	2.60	**1.84**

In Table 4.6 we report class-wise and overall F1-score for baselines, state-of-the-art and the proposed approach. We also report the accuracy of start and end frame detection 'SL' and 'EL' respectively. We observe that the proposed approach outperforms all baselines. PCRF forest shown the best results among the baselines with a performance comparable to RNN, showing that temporal pairwise information is important. On the other hand, RF performs particularly well on this dataset, revealing that distinguishing static poses is important in addition to temporal information. Combining both static and temporal information results on better performance than the current state-of-the-art JCR-RNN (Li et al., 2016), which added a regression term on a LSTM to predict both start and end frames of actions.

Efficiency. We measure the average inference time on 9 long sequences of 3,200 frames on average. We present the results at the bottom of Table 4.6 with a C++ implementation on a Intel Core i7 (2.6 GHz) and 16 GB RAM. All compared approaches are real-time, with JCR-RNN achieving 1,230 fps for 1,778 fps of the proposed approach, showing that high performance can be obtained while keeping the complexity low.

Qualitative results. In Figure 4.7 we present qualitative results for the Online Action Detection (OAD) dataset for the proposed model and the baselines. We observe that TF is more robust to false positives and false negatives than other forest baselines and more accurate predicting the start and the end of the ongoing action. Note that Li et al. (2016) did not report any qualitative result and thus we cannot compare.

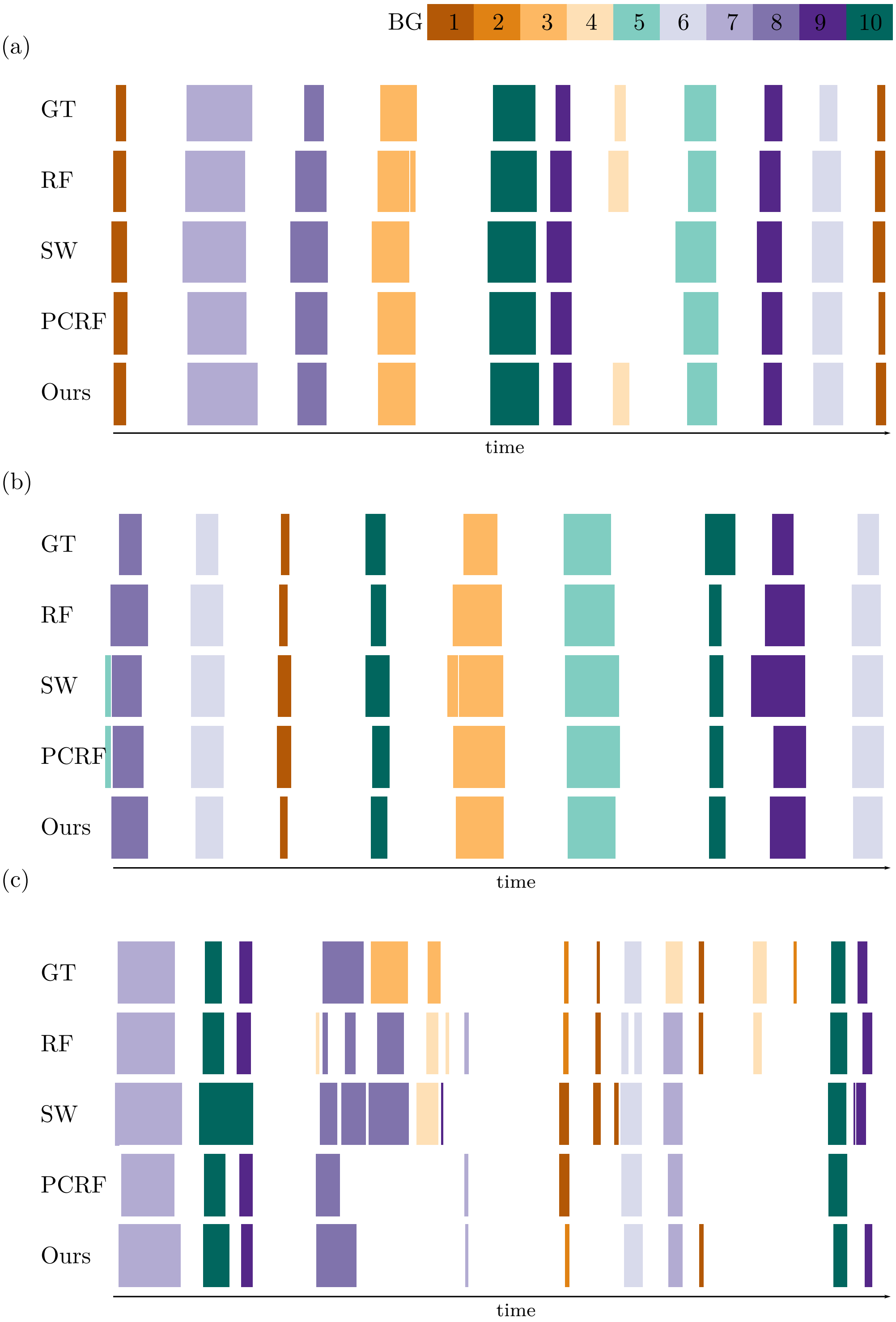

Figure 4.7: Qualitative results in OAD dataset. We show two sequences (a,b) in which the transition forest obtains good performance and one (c) in which it fails to detect several actions. Each action instance is represented with different colour depending on its category. First row in each figure represents the ground-truth (GT) temporal bounds and BG in 'white' represents background (i.e. no meaningful actions ongoing).

4.5 SUMMARY

This chapter has proposed a new forest-based classifier that can discriminatively learn both static poses and transitions. Our proposed training procedure enhances the capture of temporal dynamics compared to other strong forest baselines. The introduction of temporal relationships while growing the trees and their use in inference yield more robust frame-wise predictions and demonstrate state-of-the-art performance for challenging problems of both action recognition and online action detection.

In contrast to the trajectory Hough forest (THF) in the previous chapter, transition forest (TF) learns the transitions within the forest in a discriminative way. Furthermore, actively decreasing the uncertainty of transitions during the tree growth and storing them at the leaf nodes diminishes the complexity of storing histograms at every level of the tree and renders the approach more suitable for real-world applications.

However, our presented work has some limitations. For example, our learning stage was limited to pairwise transitions, and we believe that it would have been interesting to follow one of the training strategies that were discussed in page 62 and incorporate different time orders within the same tree learning. Also, given the generality of our work, it could be interesting to test its performance when using other data modalities, such as RGB and depth frame features, or when applied to other temporal problems that require efficient online classification. The next chapter explores the use of other data modalities.

5

UNDERSTANDING EGOCENTRIC HAND-OBJECT ACTIONS WITH RGB-D VIDEOS AND 3D HAND POSE ANNOTATIONS

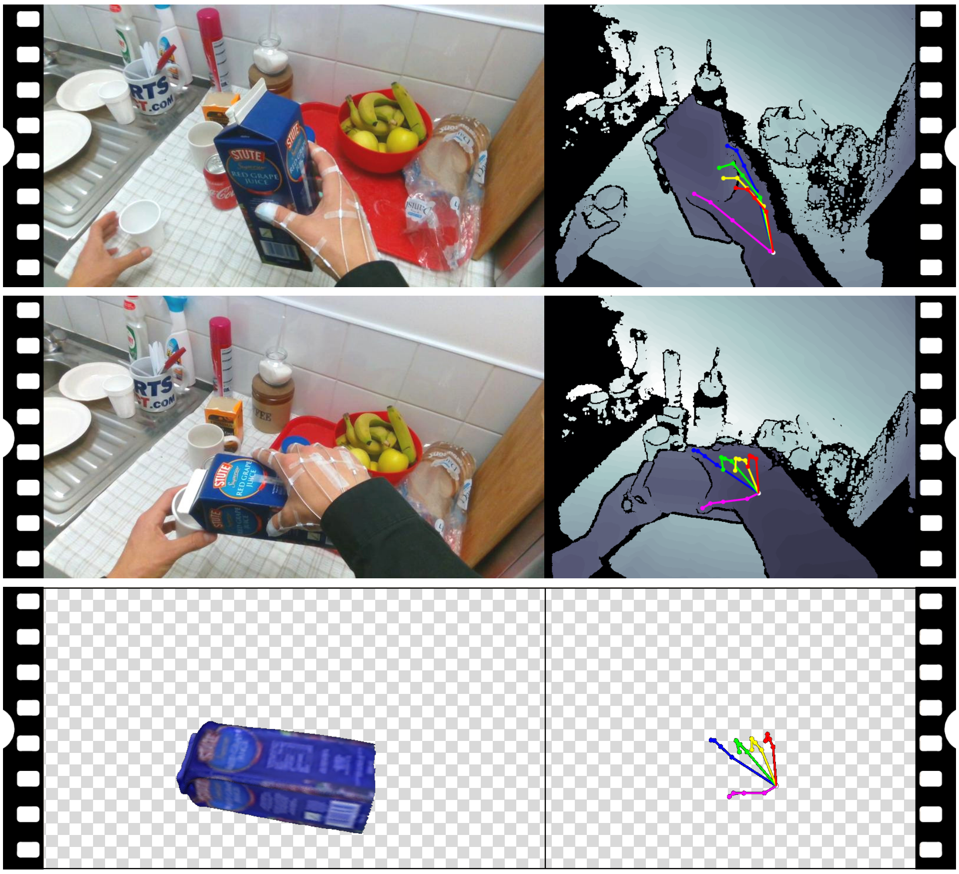

Figure 5.1: Two frames belonging to the action class ‘pour juice’ (two top rows). In this chapter a novel first-person action recognition dataset with RGB-D sequences and 3D hand pose annotations is proposed. Magnetic sensors and inverse kinematics of a hand are used to capture the pose. Depth image and hand pose (right); 6D object pose for a subset of hand-object actions is captured to enable further research by the object pose community (bottom).

5.1 OVERVIEW

MOST previous work on RGB-D action recognition has focused on actions that are performed by the whole human body. The exceptions have been mainly application-oriented for contexts such as hand gestures for HCI (Liu and Shao, 2013; Ohn-Bar and Trivedi, 2014; Molchanov et al., 2016; De Smedt et al., 2016) and sign language recognition (Wang, Liu, Chorowski, Chen and Wu, 2012). As Chapter 2 has discussed, the use of skeleton features has led action recognition in RGB-D because it offers a powerful holistic representation of the body.

Previous work on first-person action recognition (Ishihara et al., 2015; Cai et al., 2016; Ma et al., 2016; Singh et al., 2016) has found that daily actions are effectively explained by examining hands and a similar observation was found for the third-person view (Yang et al., 2015). These approaches extract hand information from hand silhouettes (Ma et al., 2016; Singh et al., 2016) or discrete grasp classification (Ishihara et al., 2015; Cai et al., 2016; Rogez et al., 2015*b*) that employs low-level image features. Rogez et al. (2015*b*) have provided static actions and hand poses in synthetic data, whereas our work involves dynamic actions and hand poses in real sequences. In full-body human action recognition, the inclusion of higher-level features, such as body pose, can benefit action recognition (Yao et al., 2011; Wu and Shao, 2014; Zhang, Wang, Gou, Sznaier and Camps, 2016; Shahroudy, Liu, Ng and Wang, 2016). Compared to full-body actions, hand actions present unique differences; for example, style and speed variations across subjects are more pronounced, as there is a higher degree of mobility for fingers, and the motion can be remarkably subtle.

A setback of using hand rather than full-body poses for action recognition is the absence of reliable and immediately available pose estimators (Shotton, Sharp, Kipman, Fitzgibbon, Finocchio, Blake, Cook and Moore, 2013; Wei et al., 2016). This absence is mainly due to the lack of hand pose annotations on real data in contrast to synthetic data sequences, especially those involving objects (Rogez et al., 2014; 2015*a*;*b*). To bridge this gap, we collected RGB-D video sequences of more than 100,000 frames of 45 daily hand action categories that involved 25 objects in several hand-grasp configurations. To obtain high-quality hand pose annotations from real sequences, we utilised our own mo-cap system, which automatically infers the location of each of the 21 joints of the hand via six magnetic sensors on the finger tips and the inverse kinematics of a hand model. To the best of our knowledge, this is the first benchmark for RGB-D hand action sequences with 3D hand poses. Additionally, we recorded the 3D rotations and 3D locations of objects.

This chapter presents an extensive experimental evaluation of RGB-D and pose-based action recognition baselines and state-of-the-art approaches, including the temporal forest models from the previous chapter. The evaluation measures the impact of employing appearance features, poses and their combinations. It also assesses the readiness of the current hand pose estimation when hands are severely occluded by objects in egocentric views and considers its influence on action recognition. The results evidence clear benefits of using hand pose as a cue for action recognition compared to other data modalities.

Contributions

- **Dataset.** We propose the first fully annotated dataset to support the study of egocentric hand actions and poses. This dataset is the first to combine both fields in the context of hands in real sequences and quality hand pose labels.

- **Action recognition.** We evaluate several baselines and state-of-the-art approaches in RGB-D and pose-based action recognition according to our proposed dataset. Our selected methods cover most of the research trends in both methodology and use of different data modalities. Furthermore, we extend the transition forest (TF) model from the previous chapter to use colour and depth features.

- **Hand pose.** We evaluate the state-of-the-art hand pose estimator in our real dataset, i.e. the occluded setting of hand-object manipulations, and evaluate its performance for action recognition.

5.2 RELATED WORK

Egocentric vision and manipulation actions

The leading role of hands in object manipulation has attracted the interest of both the computer vision and robotics communities. From an action recognition perspective and through the use of only RGB cues, recent work (Fathi, Farhadi and Rehg, 2011; Fathi, Ren and Rehg, 2011; Pirsiavash and Ramanan, 2012; Ma et al., 2016; Singh et al., 2016; Bambach et al., 2015) has delved into the recognition of daily actions and discovered that both objects and hands are important cues for the recognition problem. Another related line of work is the study of the human grasp from a robotics perspective (Bullock et al., 2015; Cai et al., 2015) as a cue for action recognition (Yang et al., 2015; Ishihara et al., 2015; Cai et al., 2016; Fermüller et al., 2018) or force estimation (Rogez et al., 2015*b*; Fermüller et al., 2018) and as a recognition problem itself (Huang et al., 2015; Rogez et al., 2015*b*). These previous works have modelled hands with low-level features or intermediate representations that follow empirical grasp taxonomies (Bullock et al., 2015), and these were thus limited in comparison to the 3D hand pose sequences in the present work. From a hand pose perspective, Rogez et al. (2014) have proposed a small synthetic dataset of static poses that is limited to train recent data-hungry algorithms. In a more relevant development for the scope of this chapter, Moghimi et al. (2014) have mounted a depth sensor to recognise egocentric activities and modelling hands with low-level skin features. In a similar setting, Damen et al. (2012) have utilised an egocentric depth sensor for workspace monitoring and were able to track manipulated objects in 3D space. Through an approach that is similar to our own but from a third-person view, Yang et al. (2014) and Lei et al. (2012) have used a hand tracker to obtain noisy estimates of hand pose in kitchen manipulation actions, while De Smedt et al. (2016) have recognised basic hand gestures for HCI that does not involve objects. The low quality of the hand tracker drastically limited the performed actions and pose labels in this work, but our own research provides accurate hand pose labels to study more realistic hand actions.

Hand pose estimation

Due mainly to the recent availability of RGB-D sensors, the field has made significant progress regarding an objectless third-person view (Oikonomidis et al., 2011*a*; Keskin et al., 2012; Tang et al., 2013; Ionescu et al., 2014; Liang et al., 2014; Qian et al., 2014; Neverova et al., 2014; Oberweger et al., 2015; Sharp et al., 2015; Ye et al., 2016) as well as more modest advances with the first-person view (Rogez et al., 2014; Oberweger et al., 2016; Mueller et al., 2017; Choi et al., 2017) have investigated the 3D tracking of a hand as it interacts with an object from a third-person view. Hamer et al. (2010) have previously studied the use of object grasp as a hand pose, while Romero et al. (2013) have considered the object shape as a cue. An important limitation is the difficulty of obtaining accurate 3D hand pose annotations, which has led researchers to resort to synthetic (Rogez et al., 2014; Sharp et al., 2015; Mueller et al., 2017; Choi et al., 2017; Baek et al., 2018), manually or semi-automatically annotated (Tang et al., 2014; Tompson et al., 2014; Sun et al., 2015; Oberweger et al., 2016) datasets that have in turn yielded non-realistic images, low numbers of samples and frequently inconsistent annotations. With the help of magnetic sensors for annotation and in a similar approach to Wetzler et al. (2015), Yuan, Ye, Stenger, Jain and Kim (2017) have proposed a major benchmark that includes egocentric poses without objects and demonstrated that a convolutional neural network (CNN) baseline can achieve state-of-the-art performance when sufficient training data is available. Yuan, Ye, Garcia-Hernando and Kim (2017) have confirmed this insight in a public challenge that utilised a subset of our proposed dataset, which was followed by an analysis by Yuan et al. (2018) of the current state of the art in the field.

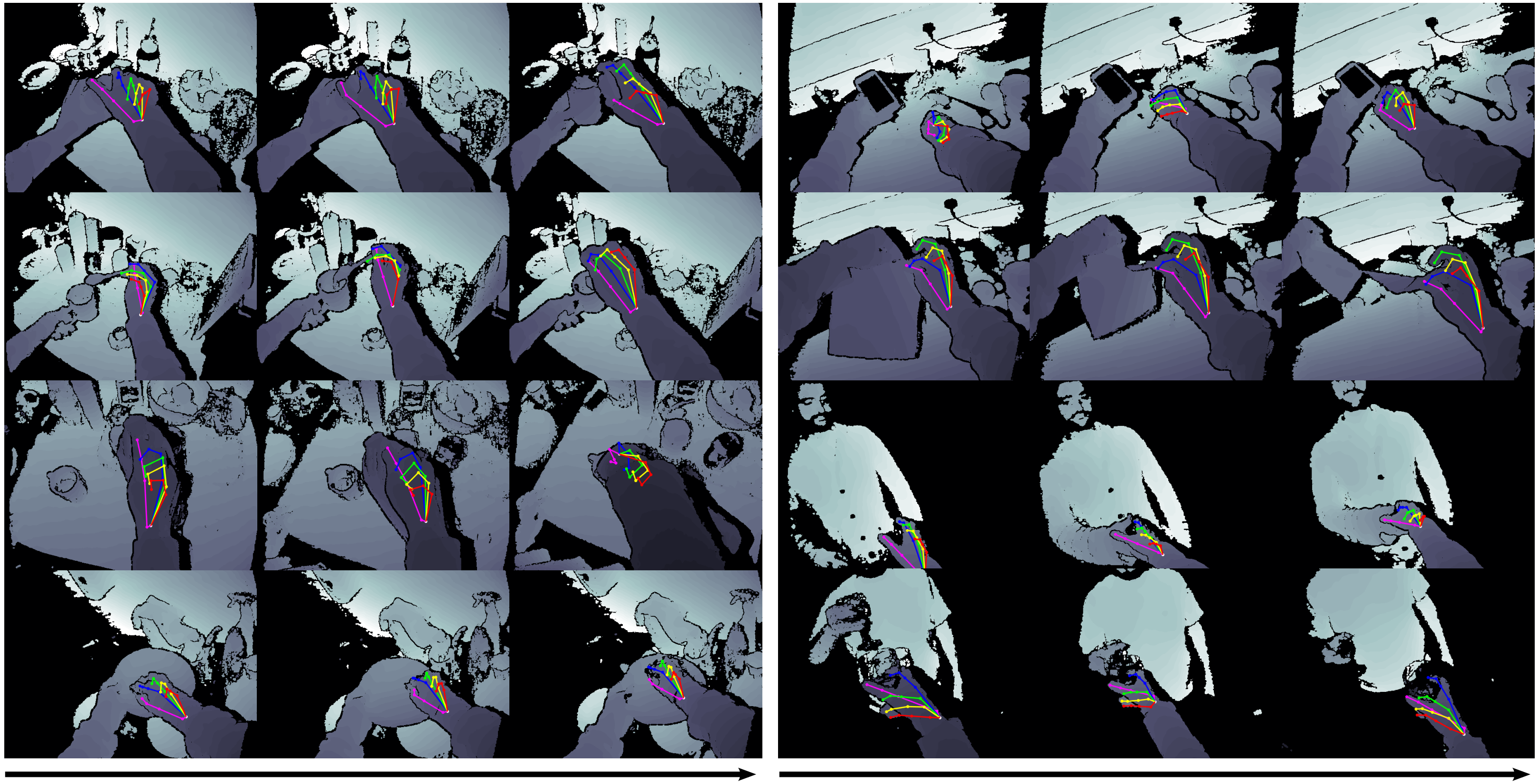

Figure 5.2: Hand actions: We have captured daily hand actions using a RGB-D sensor and used mo-cap to annotate hand pose. Left - from top to bottom: ‘open peanut butter’, ‘put sugar’, ‘pour milk’ and ‘wash with sponge’ (all in kitchen). Right - from top to bottom: ‘charge cell phone’ and ‘tear paper’ (office); ‘handshake’ and ‘toast with wine glass’ (social).

5.3 DAILY HAND-OBJECT ACTIONS DATASET

In this section, the proposed Daily Hand-Object Actions dataset is described, and a variety of relevant statistics are shown.

Dataset overview

The dataset contains 1,175 action videos belonging to 45 different action categories in three different scenarios and performed by six actors. A total of 105,459 RGB-D frames are annotated with accurate hand pose and action category. Action sequences present high both inter-subject and intra-subject variability of style, speed, scale and viewpoint. Object 6-dimensional (location and angle in 3D) pose and mesh model are also provided for four objects involving ten different action categories. In Figure 5.2 we show some example frames for different action categories and hand-pose annotation visualisation.

Hand-object actions

45 different daily hand action categories involving 25 different objects were captured (Figure 5.7 (a)). Action categories are designed to span a high number of different grasp configurations and be diverse in both hand pose and action space (see Figure 5.4). Each object has a minimum of one associated action (e.g. pen-'write') and a maximum of four (e.g. sponge-'wash', 'scratch', 'squeeze' and 'flip'). These 45 hand actions were recorded and grouped in three different scenarios: kitchen (25), office (12) and social (8). Kitchen scenario (Figure 5.2 left) comprises actions such as 'stir', 'sprinkle', 'prick' and 'pour', while some of the office actions (Figure 5.2 top-right) include 'write', 'type' and 'tear paper'. The social scenario (Figure 5.2 bottom-right) contains interactions with other humans such as 'handshake', 'high five' and 'toast with a glass of wine'. We also provide 6-dimensional object pose and mesh models for the following objects: 'milk bottle', 'salt', 'juice carton' and 'liquid soap'. These objects are involved in 10 different hand-object action categories in the kitchen scenario.

In this work we have considered each hand-object manipulation as a different action category similar to previous datasets (e.g. Fathi, Ren and Rehg (2011)), although other definitions are plausible. For example, one could label 'open juice carton' and 'open

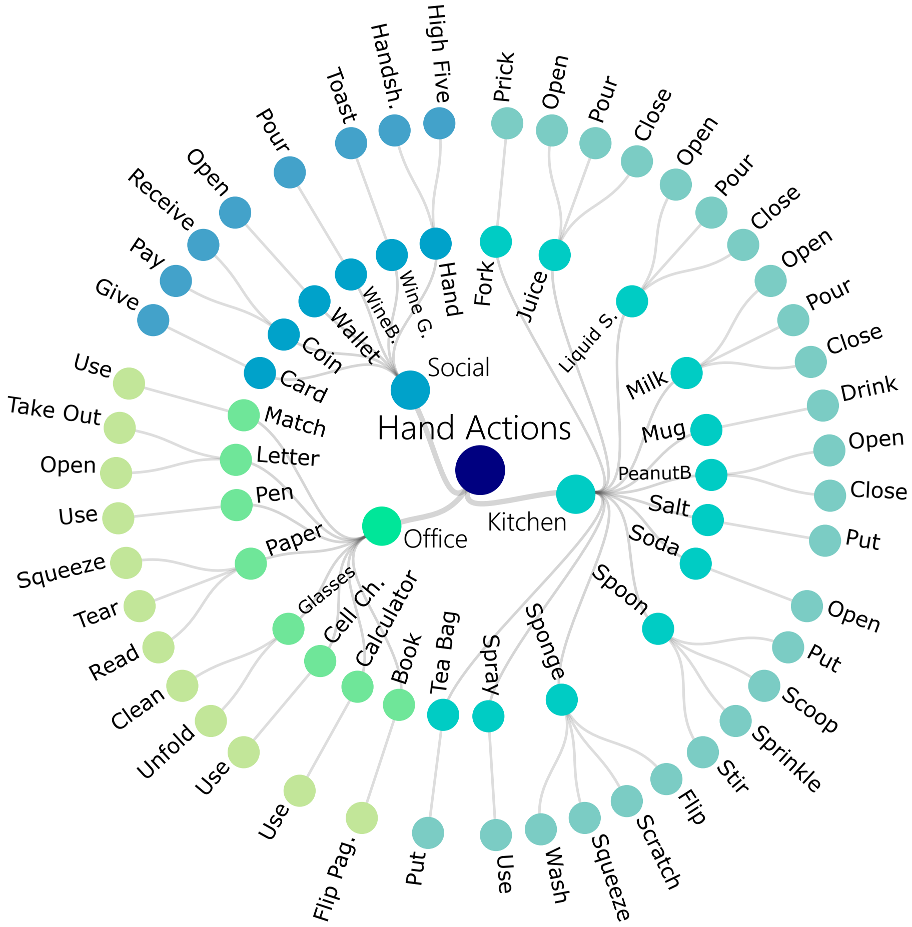

Figure 5.3: Taxonomy of our hand actions involving objects dataset. Some objects are associated with multiple actions (e.g. spoon, sponge, liquid soap) while some others have only one linked action (e.g. calculator, pen, cell charger)

peanut butter' as same action 'open' and/or make grammar combinations (Yang et al., 2014) of nouns (objects) and verbs (actions) (Wray et al., 2016). Our criteria to select objects and action categories was two-fold: we tried to have as many grasp configurations as possible following the same taxonomy as in Rogez et al. (2015*b*) and we tried to inject ambiguity in the action space by selecting objects that had multiple uses.

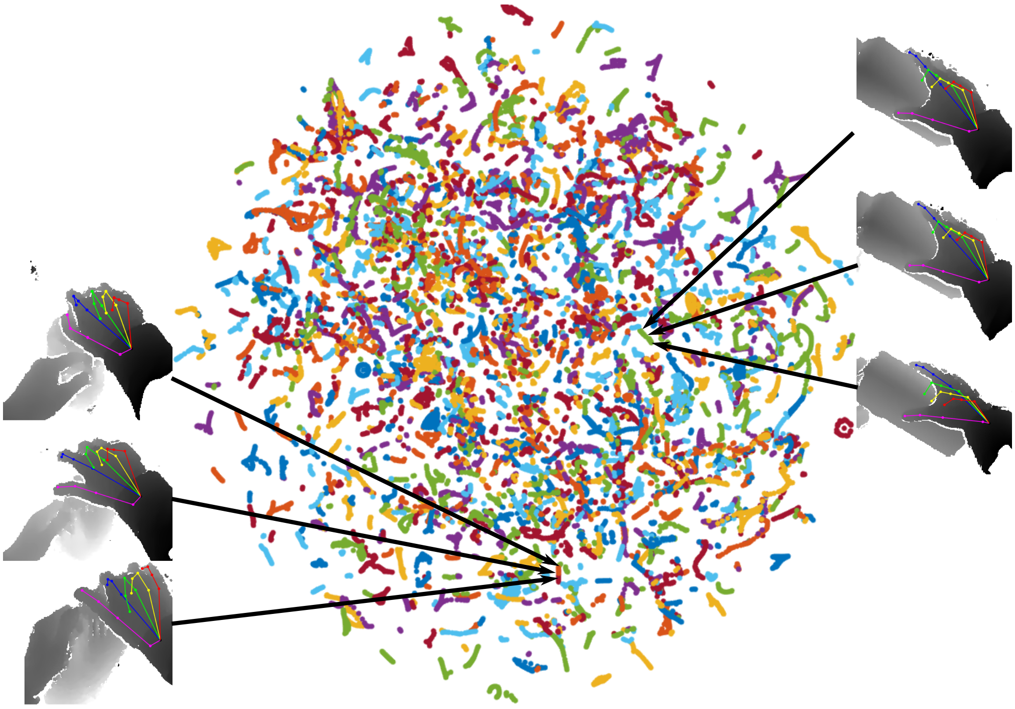

Figure 5.4: t-SNE (Maaten and Hinton, 2008) visualisation of hand pose embedding over our dataset. Each coloured dot represents a full hand pose and each trajectory an action sequence. Our dataset is rich in both hand pose configurations and actions space.

Sensors and data acquisition

Visual data. To capture visual data, the most recent version of the Intel RealSense SR300 RGB-D camera was mounted on the shoulder of the subject. RGB and depth streams were captured in the highest possible resolutions (i.e. 1920-by-1080 and 640-by-480 for the colour and depth stream respectively). The same frame rate of 30 fps was used in both streams.

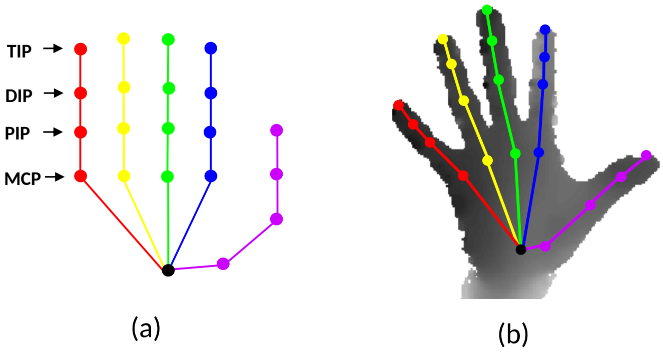

Figure 5.5: Hand model used (Yuan, Ye, Stenger, Jain and Kim, 2017) in our benchmark. (a) The hand model has 21 joints and moves with 31 degrees of freedom. (b) Example of the model over a depth image.

Pose annotation. To obtain quality annotations of hand and object pose, we follow the approach of Yuan, Ye, Stenger, Jain and Kim (2017). Hand pose is captured using six magnetic sensors (NDI trakSTAR) attached to the user's hand (five fingertips and one wrist). Each sensor provides position and orientation with 6 degrees of freedom and the full hand pose is inferred using inverse kinematics over a defined 21-joint hand model depicted in Figure 5.5. Each sensor is 2 mm wide and when attached to the human hand does not influence the depth image. The colour image is affected as the sensors and the tape to attach them can be seen, however the hand is fully visible and actions distinguishable by using the colour image (see Figure 5.1). For the object pose, we attach one another sensor to the closest point we can reach to the centre of mass. More details about the capture system can be found in Yuan, Ye, Stenger, Jain and Kim (2017) and in Appendix A.

Recording process. We asked six people, all right-handed, to perform the actions while having the mo-cap sensor and the camera attached as shown in Figure 5.6. Instructions on how to perform the action in a safe manner (for the sensitive attached sensors) were given, however no instructions about style or speed were provided in order to capture realistic data. We found difficulties while acquiring the data due to the magnetic nature of the sensor as any metallic object would interfere and make the hand pose impossible to recover. This limited the objects to use and, in some cases, we resorted to their plastic versions (e.g. fork and spoon). Action labels were annotated manually.

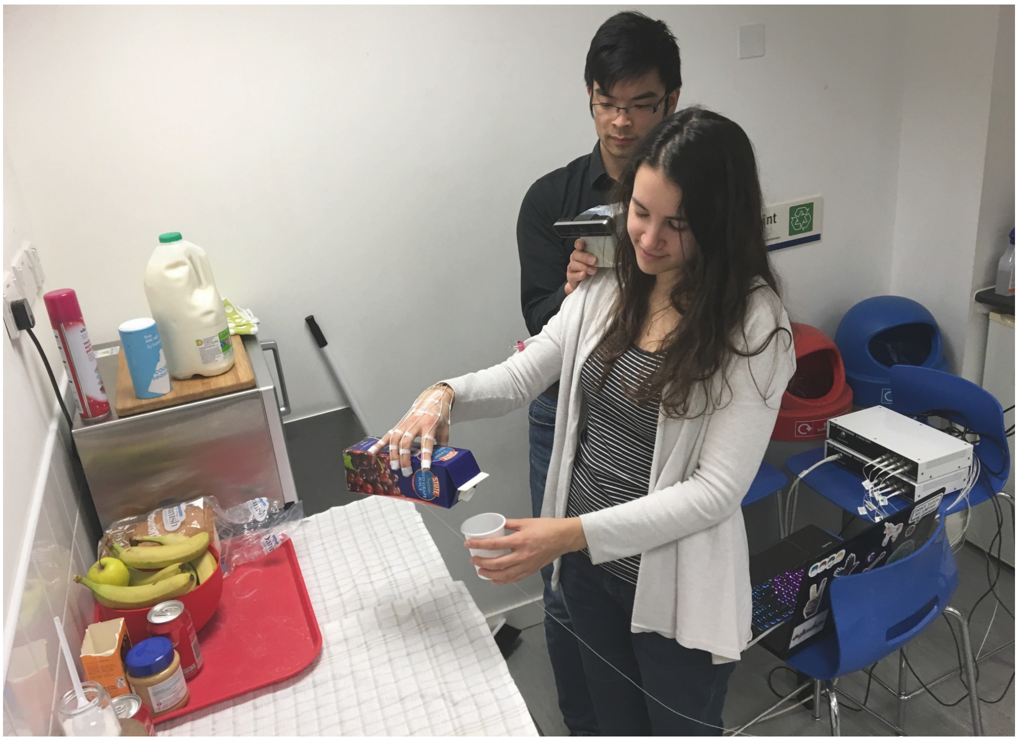

Figure 5.6: Dataset recording process during action 'pour juice' in kitchen scenario. The RGB-D sensor is placed on the shoulder of the subject for egocentric setting. Magnetic sensors are attached to both subject's fingers and manipulated object. The magnetic transmitter is attached to the camera to facilitate the mapping between magnetic sensors and camera coordinates.

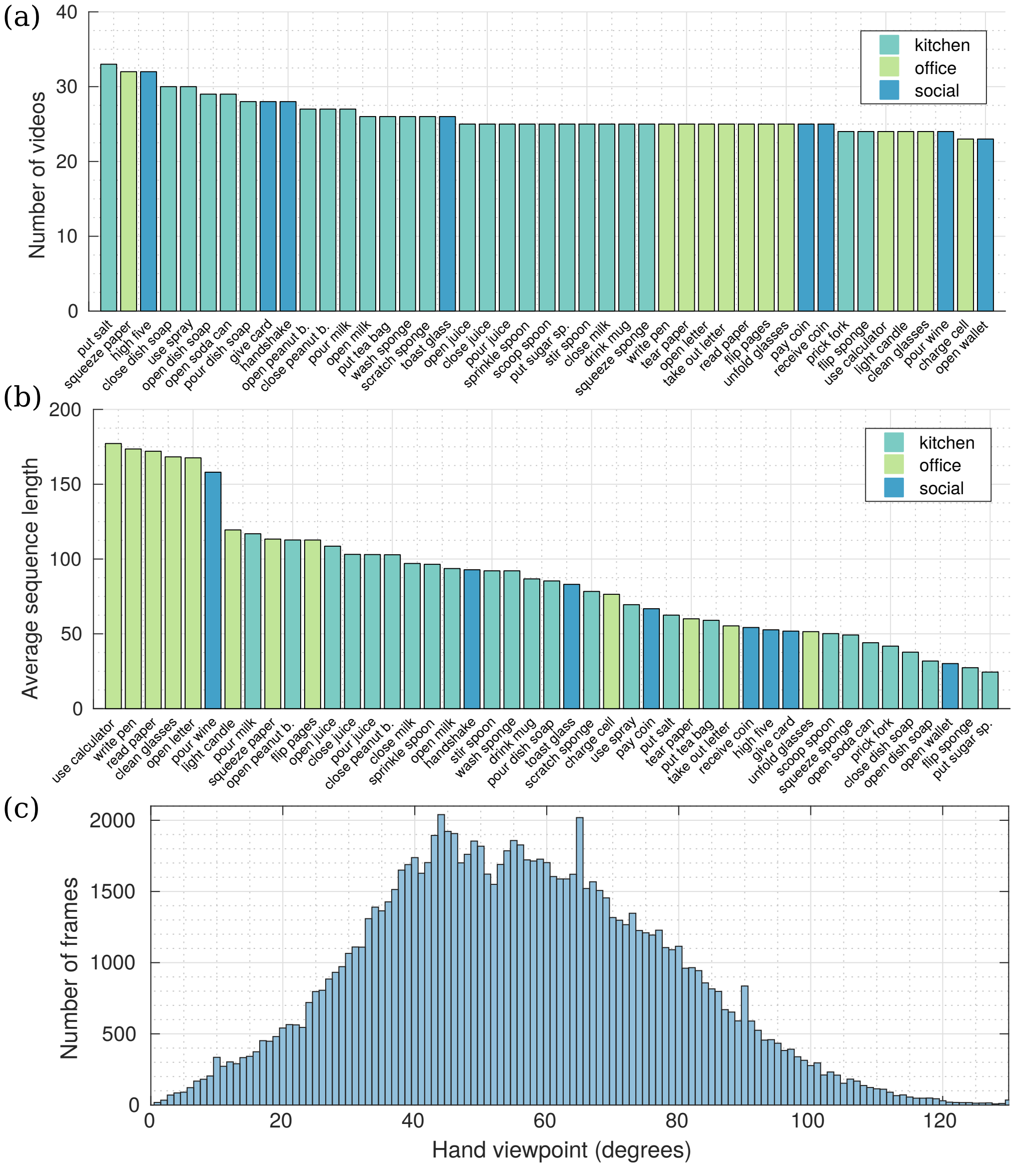

Figure 5.7: (a) Number of action instances per hand action class. (b) Average number of frames in each video per hand action class. Our dataset contains both atomic and more temporally complex action classes. (c) Distribution of hand viewpoints, defined as angles between the direction of the camera and the direction of the palm of the hand.

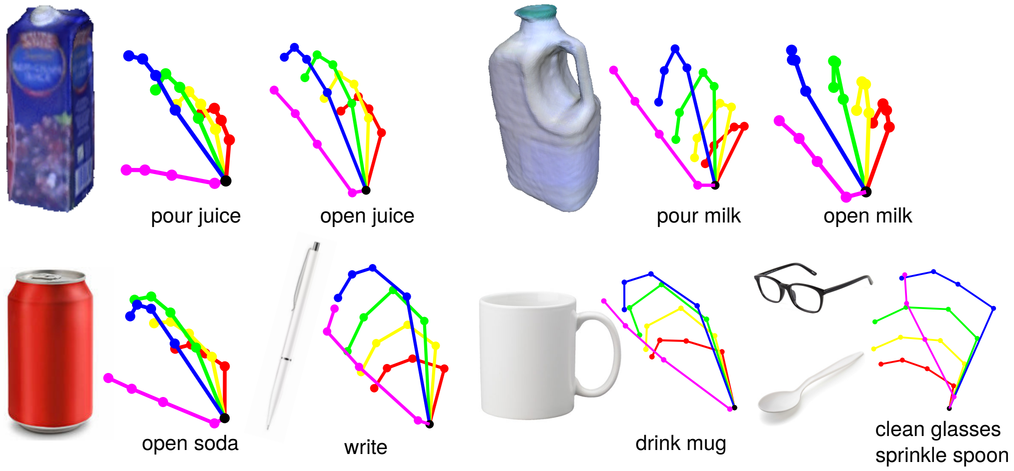

Figure 5.8: Correlation between objects, grasps and actions. Shown poses are the average pose over all action sequences of a certain class. One object can have multiple grasps associated depending on the action performed (e.g. 'juice carton' and 'milk bottle') and one grasp can have multiple actions associated (e.g. lateral grasp present at 'sprinkle' and 'clean glasses').

Dataset statistics

Taxonomy. Figure 5.3 shows the distribution of different actions per involved object. Some objects such as spoon have multiple actions ('stir', 'sprinkle', 'scoop', 'put sugar') while some objects have one specific action ('use calculator'). Although it is not an object per se, we include 'hand' as an object in actions 'handshake' and 'high five'.

Videos per action class. On average there are 26.11 sequences per class action and 45.19 sequences per object. For detailed per class numbers see Figure 5.7 (a).

Duration of videos. Figure 5.7 (b) shows the average number of video duration for the 45 action classes. Some action classes such as 'put sugar' and 'open wallet' involve short atomic movements (on average one second) while others such as 'open letter' require more time to be executed.

Grasps. We identified 34 different grasps following the same taxonomy as in (Rogez et al., 2015*b*), including the most frequently studied ones (Cai et al., 2016) (i.e. precision/power grasps for different object attributes such as prismatic/round/flat/deformable). In Figure 5.8 we show some examples of correlation between objects, hand poses and actions.

Viewpoints. In Figure 5.7 (c) we show the distribution of frames per hand viewpoint. We define the viewpoint as the angle between the camera direction and the palm of the hand. The dataset presents challenging viewpoints, in contrast to typical easy hand poses in third person view with palm facing the camera (around 90 degrees).

Hand occlusion. Figure 5.16 (bottom) shows the average number of visible (not occluded by object or viewpoint) hand joints per action class. Most actions present a high degree of occlusion (on average 10 visible joints out of 21).

Comparison with other datasets

Table 5.1: First-person view datasets with hands and objects involved. The proposed dataset is the first providing both hand pose and action annotations on real data (cf. synthetic).

Dataset	Sensor	Real?	Class.	Seq.	Frames	Labels
Yale (Bullock et al., 2015)	RGB	✓	33	-	9100	Grasp
UTG (Cai et al., 2015)	RGB	✓	17	-	-	Grasp
GTEA (Fathi, Ren and Rehg, 2011)	RGB	✓	61	525	31,222	Action
EgoHands (Bambach et al., 2015)	RGB	✓	4	48	4,800	Action
GUN-71 (Rogez et al., 2015*b*)	RGB-D	✓	71	-	12,000	Grasp
UCI-EGO (Rogez et al., 2014)	RGB-D	✗	-	-	400	Pose
Choi *et al.*(Choi et al., 2017)	RGB-D	✗	33	-	16,500	Grasp+Pose
SynthHands (Mueller et al., 2017)	RGB-D	✗	-	-	63,530	Pose
EgoDexter (Mueller et al., 2017)	RGB-D	✓	-	-	3190	Fingertips
Ours	RGB-D	✓	45	1175	105,459	Action+Pose

In Table 5.1 we summarise popular egocentric datasets that involve hands and objects in both dynamic and static fashion depending on their problem of interest. For conciseness, we have excluded from the table related datasets that do not partially or fully contain objects manipulations, such as Pirsiavash and Ramanan (2012), Oberweger et al. (2016) and Yuan, Ye, Stenger, Jain and Kim (2017). Note that previous datasets in action recognition (Bambach et al., 2015; Fathi, Ren and Rehg, 2011) do not include hand pose labels. On the other hand, pose and grasp datasets (Rogez et al., 2014; 2015*b*; Bullock et al., 2015; Cai et al., 2015; Choi et al., 2017; Mueller et al., 2017) do not contain dynamic actions and hand pose annotation is obtained by generating synthetic images or rough manual annotations (Mueller et al., 2017). Our dataset 'fills the gap' of egocentric dynamic hand action using pose and compares favourably in terms of diversity, number of frames and the use of real data.

5.4 EVALUATED ALGORITHMS AND BASELINES

5.4.1 *Action recognition*

To evaluate the current state-of-the-art in action recognition we chose a variety of approaches that, we believe, cover the most representative trends in the literature (Table 5.3). As the nature of our data is RGB-D and we acquired hand pose annotations, we focus our attention to RGB-D and pose-based action recognition approaches, although we also evaluate two RGB action recognition methods (Feichtenhofer et al., 2016; Hu et al., 2015). Note that as discussed above, most of previous work in RGB-D action recognition involve full body poses instead of hands and some of them might not be tailored for hand actions, we elaborate further on this in Section 5.5.1.

We start with one baseline to assess how the current state-of-the-art in colour action recognition performs in our dataset. For this and given that most successful (colour) action recognition approaches (Ma et al., 2016; Singh et al., 2016) use CNNs to learn descriptors from colour and motion flow, we evaluate a recent two-stream architecture (Feichtenhofer et al., 2016) fine-tuned on our dataset.

About the depth modality, we first evaluate two local depth descriptor approaches, HOG^2 (Ohn-Bar and Trivedi, 2014) and HON4D (Oreifej and Liu, 2013), that exploit gradient and surface normal information as a feature for action recognition. As a global-scene depth descriptor, we evaluate the recent approach 'novel view' by Rahmani and Mian (2016) that learns view invariant features using a CNN from several synthesised depth views of human body pose.

We continue the evaluation with pose-based action recognition methods. As our main baseline, we implemented a recurrent neural network (RNN) with long-short term memory (Hochreiter and Schmidhuber, 1997) (LSTM) module inspired in the architecture by Zhu, Lan, Xing, Zeng, Li, Shen and Xie (2016). We also evaluate several state-of-the-art pose action recognition approaches. We start with descriptor-based methods such as moving pose descriptor (Zanfir et al., 2013) (MP) that encodes atomic motion information and (Vemulapalli et al., 2014) who represents poses as points on a Lie group. For methods focusing on learning temporal dependencies, we evaluate HBRNN (Du et al., 2015) and Gram matrix (Zhang, Wang, Gou, Sznaier and Camps, 2016). HBRNN consists of a bidirectional recurrent neural network with hierarchical layers designed to learn features from the body pose. Gram matrix is currently the best performing method for body pose and uses Gram matrices to learn the dynamics of actions. Furthermore, we evaluate one hybrid approach jointly learning heterogeneous features (Hu et al., 2015) (JOULE). JOULE uses a three-step iterative optimisation algorithm to learn features jointly considering all the data channels (colour, depth and hand pose).

To conclude, we evaluate the TF presented in Chapter 4 using different features extracted from different data modalities: colour, depth, hand pose and their combination. We compare these results to the state-of-the-art approaches discussed above and to the forest baselines studied in previous chapters: random forest (Breiman, 2001) (RF), sliding window forest (Fothergill et al., 2012) (SW), pairwise conditional random forests (Dapogny et al., 2015) (PCRF) and trajectory Hough forest (THF).

5.4.2 *Hand pose estimation*

To evaluate the state-of-the-art hand pose estimation we use the same CNN architecture as Yuan, Ye, Stenger, Jain and Kim (2017). We choose this approach as it is easy to interpret and provided the best performance in a cross-benchmark evaluation (Yuan, Ye, Stenger, Jain and Kim, 2017). The chosen method is a discriminative approach operating on a frame-by-frame basis, which does not need any initialisation and manual recovery when it fails in tracking (Oikonomidis et al., 2011*a*; Intel, 2013). Most existing methods focus on hands alone, while some tracking-based methods deal with occlusions in hand-object interactions (Oikonomidis et al., 2011*b*). We include details of this model in Appendix B.

5.5 EVALUATION RESULTS

5.5.1 *Action recognition*

In the following we present our experiments in action recognition. In this section we assume the hand pose is given, i.e. we use the hand pose annotations obtained using the magnetic sensors and inverse kinematics. We evaluate the use of estimated hand poses without the aid of the sensors for action recognition in Section 5.5.2. As a measure of performance, we use the total accuracy of predicted actions or, in other words, correct number of predictions over total number of predictions. When possible, we used publicly available codes with default parameters.

Following the standard practice in body-pose action recognition (Zanfir et al., 2013; Vemulapalli et al., 2014), we compensate for anthropomorphic differences by normalising the hand poses to all have the same distance between pairs of joints. Furthermore, to be invariant to viewpoints, we define the center of coordinates to be the hand wrist. We found in our experiments that this normalisation is important to obtain acceptable results.

A baseline: LSTM

We start our experimental evaluation with a simple yet powerful baseline: a recurrent neural network (RNN) with a long-short term memory (Hochreiter and Schmidhuber, 1997) (LSTM). The architecture of our network is inspired by Zhu, Lan, Xing, Zeng, Li, Shen and Xie (2016) with two differences: we do not 'go deep' and we use a more conventional unidirectional network instead of bidirectional. We set the number of neurons to 100 and a probability of dropout of 0.2. We use TensorFlow and Adam optimiser. We feed the normalised hand poses sequences into the LSTM.

Training and test protocols. We experiment with two protocols. The first protocol consists of using different partitions of the data for training and the rest for test and we tried two different training:test ratios of 1 : 1 and 3 : 1. The second protocol is a 6-fold 'leave-one-person-out' cross-validation, i.e. each fold consists of 5 subjects for training and one for test. Results are presented in Table 5.2. We observe that following

Table 5.2: Results for different training-test protocols. 3:1 stands for '75% of the dataset is used for training and 25% for test'. In cross-person protocol perform 6-fold leave-one-person-out cross-validation.

Protocol	1:1	3:1	cross-person
Accuracy (%)	78.73	84.82	62.06

a cross-person protocol yields the worst results taking into account that in each fold we have similar training/test proportions to the 3 : 1 setting. This can be explained by the difference in hand action styles between subjects. In the rest of the chapter we perform our experiments using the 1:1 setting with 600 action sequences for training and 575 for test. The result for this protocol is 78.73% using 1-layer LSTM. We also experimented adding with more layers, for example using 2-layer LSTM the accuracy improved to 80.14%. We did not observe any significant improvements by stacking more layers likely due to the given size of our dataset.

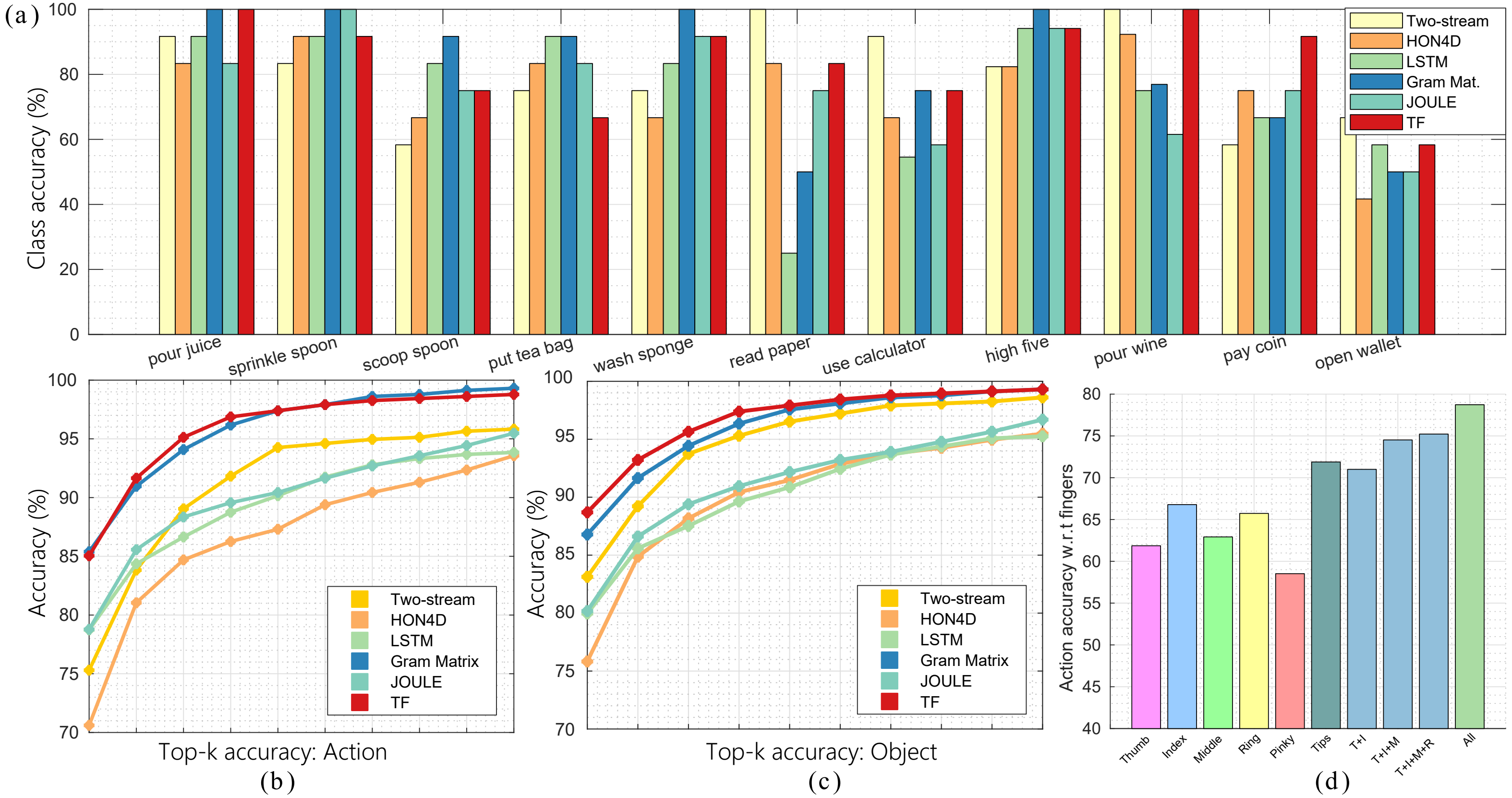

Figure 5.9: (a) We show class accuracies of some representative methods for different data modalities on a subset of classes. (b) Top-k action accuracy: true action label is in the top-k action prediction hypothesis. **(c)** Top-k object accuracy: manipulated object is in the top-k action prediction hypothesis. **(d)** Impact of each of the five fingers, combinations of them and fingertips on action recognition.

State-of-the-art discussion

In Table 5.3 we show results for state-of-the-art approaches in different data modalities and the TF introduced in the previous chapter. We observe that the two-stream method (Feichtenhofer et al., 2016) performs better when combining both colour and flow cues. Depth methods tend to perform slightly worse than the rest of the methods, suggesting that they are not able to fully capture neither the object cues nor the hand pose. Note that for 'novel view' (Rahmani and Mian, 2016) we extracted deep features from a network trained on several synthetic views of bodies, which might not generalise well to hand poses and fine-tuning in our dataset did not help. From all approaches, we observe that the ones using hand pose are the ones that achieve the best performance, with Gram matrix (Zhang, Wang, Gou, Sznaier and Camps, 2016) and Lie group (Vemulapalli et al., 2014) performing particularly well, a result in line with the ones reported in body pose action recognition.

In Figure 5.9 we select some of the most representative methods and analyse their performance in more detail. We observe that the pose method Gram matrix outperforms the rest in most of the measures, particularly when we retrieve the top k action hypothesis (Figure 5.9 (b)), showing the benefit of using hand pose for action recognition. Looking at Figure 5.9 (a), we observe that the two-stream network outperforms the rest of methods in some categories in which the object is big and the action does not involve much motion, e.g. 'use calculator' and 'read paper'. This good performance can be due to the pre-training of the spatial network on a big image recognition dataset. We further observe this in Figure 5.9 (c) where we analyse the top k hypothesis given by the prediction and look whether the predicted action contains the object being manipulated, suggesting that the network correctly recognises the object but fails to capture the temporal dynamics.

Combining different data modalities can improve the performance. For instance, we observe that HOG^2 can benefit when learning features with the help of the poses. The same is observed for JOULE and TF: hand pose features are the most discriminative ones, although the performance can be increased by combining them with RGB and depth cues. This result suggests that hand poses provide complementary information

Table 5.3: Hand action recognition performance by different evaluated approaches on the proposed dataset.

Method	Colour	Depth	Hand pose	Acc. (%)
Two stream-colour (Feichtenhofer et al., 2016)	✓	✗	✗	61.56
Two stream-flow (Feichtenhofer et al., 2016)	✓	✗	✗	69.91
Two stream-all (Feichtenhofer et al., 2016)	✓	✗	✗	75.30
HOG^2-depth (Ohn-Bar and Trivedi, 2014)	✗	✓	✗	59.83
HOG^2-depth+pose (Ohn-Bar and Trivedi, 2014)	✗	✓	✓	66.78
HON4D (Oreifej and Liu, 2013)	✗	✓	✗	70.61
Novel view (Rahmani and Mian, 2016)	✗	✓	✗	69.21
1-layer LSTM	✗	✗	✓	78.73
2-layer LSTM	✗	✗	✓	80.14
MP (Zanfir et al., 2013)	✗	✗	✓	56.34
Lie group (Vemulapalli et al., 2014)	✗	✗	✓	82.69
HBRNN (Du et al., 2015)	✗	✗	✓	77.40
Gram matrix (Zhang, Wang, Gou, Sznaier and Camps, 2016)	✗	✗	✓	85.39
JOULE-colour (Hu et al., 2015)	✓	✗	✗	66.78
JOULE-depth (Hu et al., 2015)	✗	✓	✗	60.17
JOULE-pose (Hu et al., 2015)	✗	✗	✓	74.60
JOULE-all (Hu et al., 2015)	✓	✓	✓	78.78
TF (Chapter 4) only colour deep features	✓	✗	✗	53.74
TF (Chapter 4) only depth deep features	✗	✓	✗	49.56
TF (Chapter 4) only hand pose features	✗	✗	✓	80.69
TF (Chapter 4) with all above features	✓	✓	✓	85.04

to RGB and depth cues as previously observed in body pose action recognition. In the next section we analyse in depth the results obtained by TF and the contribution of each data modality.

Temporal forests evaluation

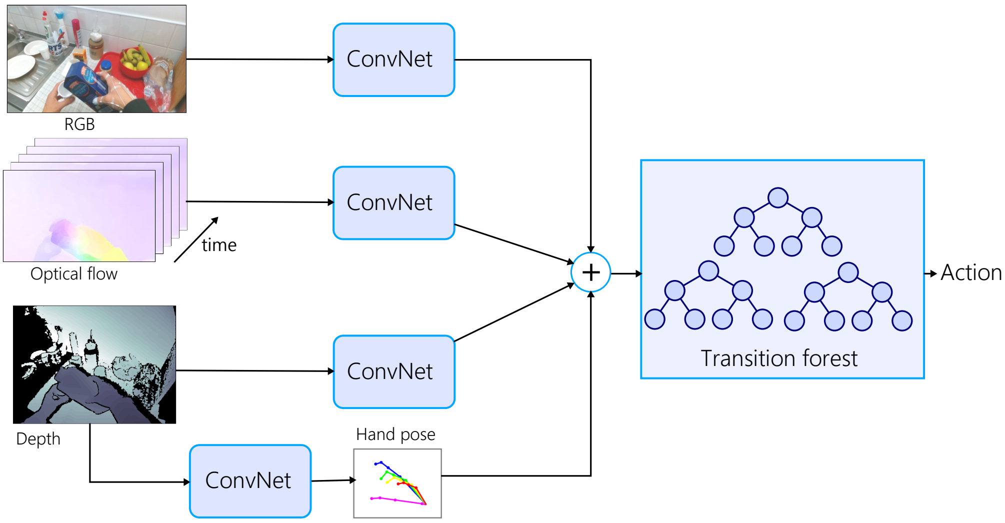

Figure 5.10: Transition forest model using colour, depth and hand pose features. For a particular frame, colour features are extracted from both RGB and optical flow channels. Hand pose features can be estimated using our hand pose baseline.

We evaluate the forest-based algorithms presented in previous chapters using four different experimental settings depending on the data channel considered: colour, depth, hand pose and their combination. To be able to perform such experiments, we need to obtain features at frame-level for both colour and depth channels. To evaluate the colour channel, we extract deep features for both RGB and flow streams from the last fully connected layer, fc7, of the two-stream network presented in Simonyan and Zisserman (2014) fine-tuned on our dataset. Features are extracted in a per-frame basis before any temporal fusion and the feature dimension is 8,192. For the depth channel, we use the 4,096-dimensional features from Rahmani and Mian (2016) before any temporal fusion and for each frame. We found that the high dimensionality and the sparsity of these features were a problem when combined. To minimise this effect, using principal component analysis (PCA) we reduce each channel dimension to 150, a number found empirically and that allowed us to reduce action recognition errors in the order of 25% for all forest-based approaches. We use the same parameters for all methods: 50 trees of maximum depth 10 and temporal order $k = 4$.

Table 5.4: Temporal decision forest approaches on proposed benchmark.

Method	Colour	Depth	Hand pose	All combined
RF	53.04	51.48	74.78	78.43
SW	53.39	49.91	76.34	79.30
PCRF	54.96	50.01	74.96	80.35
THF (Chapter 3)	58.26	50.21	73.39	78.95
TF (Chapter 4)	53.74	49.56	80.69	85.04

In Table 5.4 we present the results for the considered forest-based approaches. All the evaluated methods have similar performance for colour and depth channels, being THF the most robust one when colour features are used likely because of its temporal regression. We observe that TF, performs the best by a significant margin when all modalities are combined due to a better capture of temporal dynamics. Consistent with the results in the previous chapter but by a narrow margin, PCRF is the most robust forest-based method after TF. We found that the recognition accuracy for both colour and depth channels is rather low compared to state-of-the-art approaches in Table 5.3 and to results on the hand pose channel. We believe there are two reasons for this behaviour.

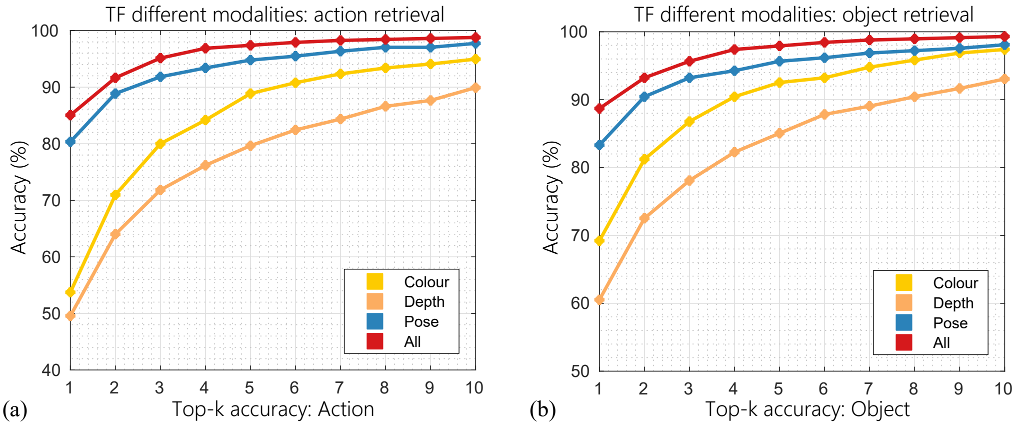

Figure 5.11: Different data modalities action and object retrieval with a TF. (a) Top-k action accuracy: true action label is in the top-k action prediction hypothesis. (b) Top-k object accuracy: manipulated object is in the top-k action prediction. Colour and depth features mainly encode coarse hand-object and background information making hard to learn temporal dynamics in contrast to hand pose features. Combining all feature channels leads to a more robust result.

The first reason is the challenge of making online per-frame predictions compared to full video predictions (Feichtenhofer et al., 2016; Rahmani and Mian, 2016; Hu et al., 2015; Vemulapalli et al., 2014; Zhang, Wang, Gou, Sznaier and Camps, 2016). To verify this, we did a simple experiment where we performed temporal averaging of deep colour features over a video and using this video-level descriptor with a RF, increasing from 53.05% to 66.09%. Also, note the only offline forest, THF, achieves a considerable better performance than the rest of the forest methods.

The second reason, also found in the previous section, is the limitation of the used deep features compared to hand pose: they are good at obtaining a gist of the scene but not at capturing a fine-grained description of the hand-object interaction. To confirm this, we investigate the hypothesis generated by each modality TF and analyse them in Figure 5.11. We observe that both colour and depth streams are better at recognising the object of interest than the action itself, reducing the accuracy gap with hand pose features. Furthermore, if instead of objects and actions we check whether the hypothesis is correct regarding the background (i.e. kitchen, office and social) we obtain the following accuracies: 97.74% for colour, 91.83 for depth, 90.96% for hand pose and 96.35% for all combined. Combining all channels improves the performance in all the tested approaches, mainly because colour and depth can reduce ambiguity on similar hand poses by giving object and background information.

In Figure 5.12 we show the performance in a subset of classes for different modalities using a TF. We observe that the forest is able to deal with temporal ambiguities (e.g. 'open peanut butter' and 'close peanut butter') when using hand poses but it fails when using only colour or depth. Mostly encoding static background and object cues makes difficult to TF to learn transitions between frames when using colour and depth features compared to clean and high level pose features. Feature learning could be improved by using some attention mechanism to focus on hands (Ishihara et al., 2015). For some actions such as 'prick with fork' and 'stir spoon', colour and depth features show to be complimentary to hand poses. For some actions such as 'charge cell', using colour and depth cues can degrade the performance compared to hand poses. If we observe the confusion matrix on Figure 5.13, we see that 'charge cell' is often confused with 'tear paper'. To understand why this high confusion occurs, we visualise in Figure 5.14 one

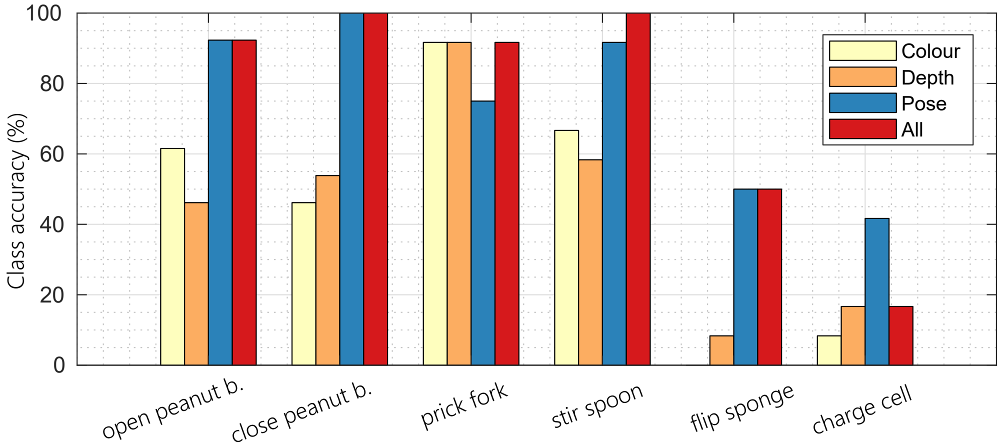

Figure 5.12: Performance of a subset of classes for different feature channels with a TF. We show that combining feature channels is beneficial, unless the performance of one channel is very poor (e.g. 'charge cell' and 'flip sponge').

colour frame for each action category and their temporally averaged hand pose. We observe that in the colour stream both actions are very similar in terms of background, colour of objects and configuration of both hands, making them almost indistinguishable when using colour and depth features. In the pose space we also observe similarities with slight differences in the thumb pose, making it more distinguishable when looking to pose and their dynamics.

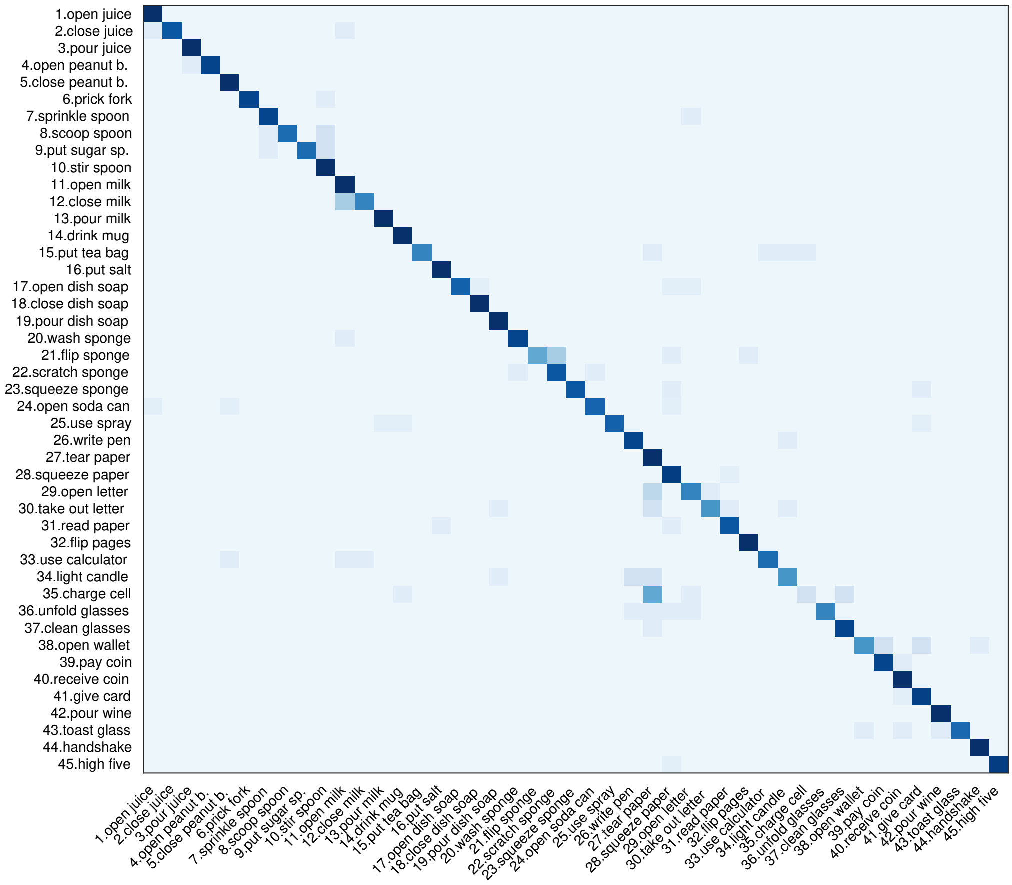

Figure 5.13: Action confusion matrix for a transition forest trained with colour, depth and hand pose features. Some action categories are classified without problems (e.g. 'pour juice' and 'put salt'), while others are often confused (e.g. 'flip sponge' and 'charge cell').

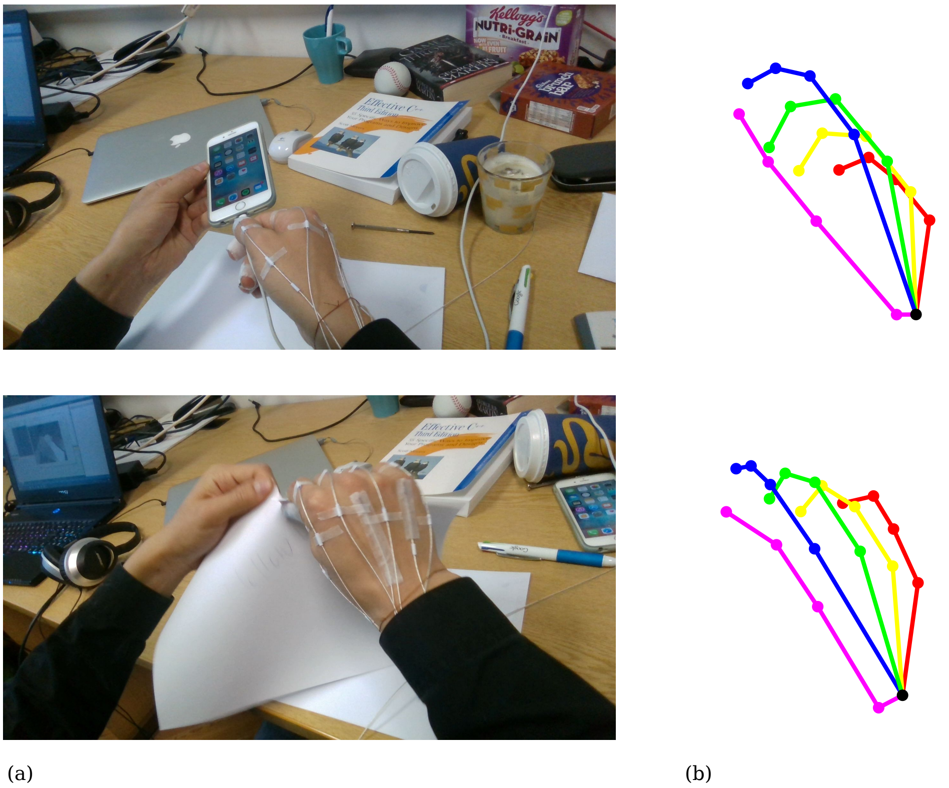

Figure 5.14: ‘charge cell’ and ‘tear paper’ action categories are often confused; (a) both categories look similar in terms of background, colour of objects and configuration of both hands. (b) time-averaged hand poses of each category show also similarities with a slight difference in the thumb.

Object pose. An extra experiment is performed by using the object pose as an additional feature for action recognition on the subset of actions that have annotated object poses: a total of 261 sequences for 10 different classes and 4 objects. The LSTM baseline is trained on half of the sequences and using three different inputs: hand pose, object pose and both combined. In Table 5.5 the results are shown and it can be concluded that combining both pose features can help on the action recognition task. Although in this thesis we do not explore this direction, we believe this is an interesting line of research as we discuss in the next chapter.

Table 5.5: The use of 6D object pose for action recognition is evaluated on a subset of the proposed dataset. We observe the benefit of combining them with the hand pose.

Pose feature	Hand	Object	Hand+Object
Action accuracy (%)	87.45	74.45	91.97

5.5.2 *Hand pose estimation*

Input pre-processing. Most approaches for hand pose estimation require to have as an input the depth channel containing only the hand. To crop the hand, we used the bounding boxes automatically annotated using the magnetic sensors. Detecting hands in depth images is still an open problem (Rogez et al., 2015*b*) that we do not investigate further. The quality of the detection is likely to affect the hand pose estimation.

Training with objects vs. no objects. One question raised while designing our experiments was whether we needed to annotate the hand pose in a close to ground-truth accuracy to experiment with hand dynamic actions. We try to answer this question by estimating the hand poses of the proposed hand action dataset in two ways partitioning our data as in our *action split*: using the nearly 300,000 *object-free* egocentric samples from Yuan, Ye, Stenger, Jain and Kim (2017) and using the images in the training set of our hand action dataset. As observed in Table 5.6 and Figure 5.15, the results suggest that having hand-object images in the training set is crucial to train state-of-the-art hand pose estimators likely because occlusions and object shapes need to be *seen* by the estimator beforehand. To confirm this, we conducted two extra experiments: *cross-subject* (half of the users in training and half in testing, all objects seen in both splits) and *cross-object* (half of the objects in training and half in testing, all subjects seen in both splits). In Figure 5.15 and Table 5.6 we observe that the network is able to generalise to unseen subjects but struggles to do so for unseen objects, suggesting that recognising the shape of the object and its associated grasp is crucial to train hand pose estimators. This shows the need of having annotated hand poses interacting with objects and thus why our dataset can be of interest for the hand pose community. In Figure 5.17 we show some qualitative results in hand pose estimation in our proposed dataset and observe, that while not perfect, they are *good enough* for action recognition.

Table 5.6: Average hand pose estimation error (Euclidean distance between magnetic poses and estimates) for different protocols and its impact on action recognition. The hand pose estimation baseline generalises better to unseen subjects than to unseen objects.

Hand pose protocol	Hand pose error (mm)	Action Acc. (%)
Cross-subject	11.25	-
Cross-object	19.84	-
Action split (training without objects)	31.03	29.63
Action split (training with objects)	14.34	72.06
Action split (GT magnetic+IK poses)	-	78.73

Hand pose estimation and action recognition. In this section we try to answer the following key question: 'how good the current hand pose estimation is for recognising hand actions?'. In Table 5.6 we show results of hand action recognition by swapping the hand pose labels by the estimated ones in the testing set. We observe that reducing the hand pose error in a factor of two yields a more than twofold improvement in action recognition. The difference in hand action recognition between using the hand pose labels in testing and using the estimated ones is 6.67%. We also tested the two best performant methods from previous section, Lie group Vemulapalli et al. (2014) and Gram matrix Zhang, Wang, Gou, Sznaier and Camps (2016). For Lie group we obtained an

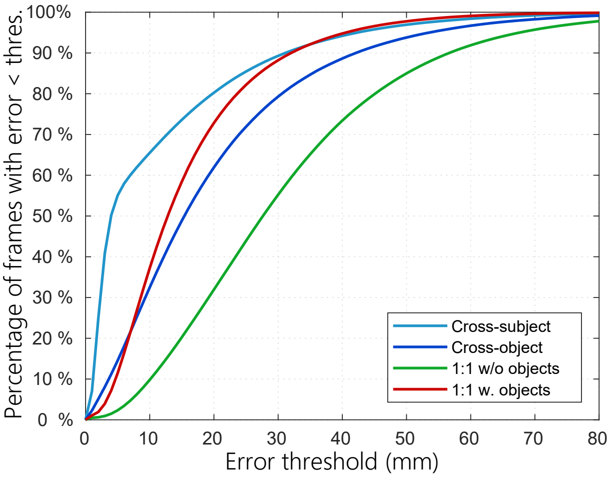

Figure 5.15: Percentage of frames for different hand pose estimation error thresholds and different protocols. Seeing objects and occlusions in training is crucial to have a robust hand pose estimator in a manipulation scenario.

accuracy of 69.22%, while for Gram matrix a poor result of 32.22% likely due to their strong assumptions in the noise distribution. TF showed to be more sensitive to noise than Lie group or LSTM dropping the accuracy to 66.78%.

In Figure 5.16 we show how the hand occlusion affects the pose estimation quality and its impact on class recognition accuracies. Although some classes present a clear correlation between hand pose error and action accuracy degradation (e.g. 'receive coin', 'pour wine'), the LSTM is still able to obtain acceptable recognition rates likely due to being able to infer the action from temporal patterns and correctly estimated joints. For more insight, we analysed the pose error per finger (thumb: 12.45; index: 15.48; middle: 18.08; ring: 16.69; pinky: 18.95; all in mm). Thumb and index joints are the best estimated ones and, according to previous section where we found that motion from these two fingers were a high source of information, this can be a plausible explanation why we can still obtain a good action recognition performance while having noisy hand pose estimates.

5.6 SUMMARY

This chapter has proposed a novel benchmark and presented experimental evaluations for RGB-D and pose-based hand action recognition in first-person settings. The benchmark provides temporal action labels, full 3D hand pose labels and six-dimensional (6D) object pose labels within the dataset. Both RGB-D action recognition and 3D hand pose estimation are relatively new fields, and this research represents a first attempt to relate both of them as happened for full human body. As the first benchmark of its kind, we believe that this study can encourage future work in multiple fields, including action recognition, hand pose estimation, object pose estimation and grasp analysis in addition to emerging topics, such as joint hand-object pose estimation.

We have evaluated several baselines in our dataset and concluded that hand pose features are a rich source of information for recognising manipulation actions. We extended the transition forest (TF) model from the previous chapter to engage with colour and depth cues by extracting deep features that lead to state-of-the-art performance. We found that combining hand pose and object pose features can enhance action recognition performance. Although colour and depth features are complementary to hand poses, the incorporation of all of these features into a practical scenario is not currently possible in view of the high computational costs of running four networks in parallel and computing the optical flow in real time. Regarding the optical flow channel, we did not apply any technique for camera motion compensation, as there was relatively stable ego-motion in our benchmark. However, in a real scenario, such application should be considered. Nevertheless, our hand pose estimator baseline can run at 25 fps, and its frame-based nature renders it not sensitive to camera motion. Thus, it is suitable for real-world application.

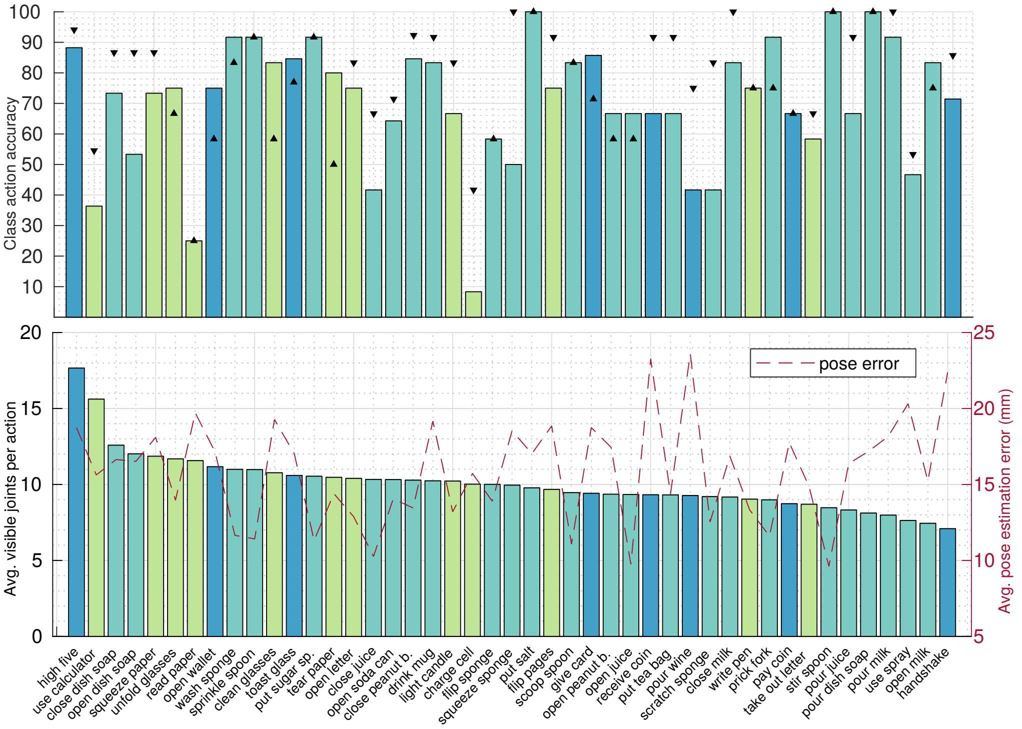

Figure 5.16: Effect of hand occlusion in action recognition and hand pose estimation; (bottom): average number of visible (i.e. not occluded) joints for hand actions on our dataset and its impact on hand pose estimation; (top) class action recognition accuracies for our LSTM baseline using estimated hand poses (accuracies with ground-truth poses are represented with black triangles).

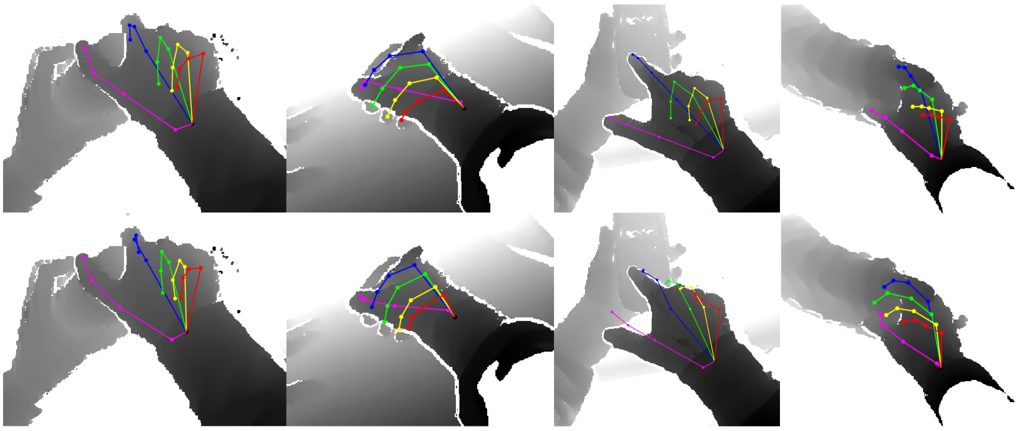

Figure 5.17: Top: pose labels obtained using the magnetic sensors; (bottom) hand pose estimates. Some estimates are noisy but good enough for action recognition.

6

SUMMARY AND FUTURE IDEAS

6.1 OVERALL SUMMARY

This thesis proposes a novel temporal decision forest to investigate egocentric actions in two scenarios. This chapter summarises the research findings and discusses future directions.

Chapter 1 first provided a general overview of action recognition and motivated the study of egocentric hand actions with examples of applications that motivated Chapters 3 and 5. It further introduced limitations to the use of decision forest classifiers for action recognition, which motivated the contributions of Chapter 4.

Chapter 2 categorised existing work in human action recognition according to data modalities and methodologies and provided an overview of several benchmarks. It additionally discussed the evolution of action recognition depending on the available hardware and complexity of solutions.

Chapter 3 proposed a trajectory Hough forest (THF) as the framework for the application of fingertip writing in mid-air with an egocentric RGB-D sensor. Moreover, it suggested a hand posture descriptor that can enable fingertip detection in depth image. The chapter introduced the extension of the Hough forest to encourage temporal consistence in predictions. The framework proved to be effective for the recognition of fingertip writing in mid-air as well as for generalising spatio-temporal trajectories that were extracted from RGB videos.

Chapter 4 proposed a transition forest (TF) as a new temporal decision forest model that can learn both static and temporal dynamics from skeleton data. The proposed approach overcomes most of the drawbacks of applying decision forests to action recognition that Chapters 2 and 3 have noted. We compared TF with several powerful forest-based baselines and state-of-the-art approaches and demonstrated the suitability of the approach to recognise and detect actions in an efficient and online manner.

Chapter 5 introduced a new RGB-D benchmark for egocentric action recognition and extended the TF to include various data modalities. The main novelty of this benchmark is its introduction of accurate hand pose and object pose labels, which enable the study of pose-based approaches in egocentric action recognition. The introduced benchmark has potential for use in other applications, such as 3D hand pose estimation and robot imitation learning.

6.2 FUTURE IDEAS

Fingertip writing recognition

Several aspects can be explored from our starting point. First, similarly to Huang et al.'s (2016) work with RGB video, it could be interesting to consider applications of deep learning architectures to the problem of fingertip detection in depth images, especially in a end-to-end manner. Second, our approach only considers isolated characters, but this could be extended to recognise complete words with the help of grammar that is similar to the NLP field. Finally, our framework considers only a limited scenario of indoor recognition with characters from only one actor, which restricts its application to real-life technology. Additional studies should examine how to generalise to any background, user and sensor in order to implement this idea in a product, and the application of an end-to-end deep learning approach should likely accompany such research.

Decision forests for temporal data

Our presented approach learns one temporal transition order per tree. However, in this regard, there is room for improvement by adding more temporal complexity to the tree learning. In Chapter 4 (page 62) we discussed a variety of training strategies that could extend our transition forest with additional temporal context in the node splitting or by including the temporal distance in the optimisation process. Another interesting option could be to extend the framework to address longer temporal relationships in a hierarchical manner that resembles the hierarchical recurrent neural network of Du et al. (2015).

Egocentric hand action recognition

Our benchmark is the first to include hand poses in an egocentric setting. We believe that there are many lines of research to explore. One interesting idea would be to jointly learn hand actions and hand poses in a similar fashion to Iqbal et al. (2017). Moreover, future research could incorporate hand poses, object poses and actions in the same learning process. As Chapter 5 has mentioned, the proposed extension of transitions forests is not currently feasible in real time, as there is a need to process different channels independently and estimate optical flow, which is a slow process. Features could be fused in the first stage of a joint network to accelerate the learning process and free the process of data redundancy. Moreover, motion features could be learned from hand poses instead of from the optical flow, which would accelerate the whole pipeline. Unlike Rogez et al. (2015*b*), we did not explore other interesting properties of hand manipulations, such as contact points and force estimation, which could be of interest to the robotics community. We believe that using high-level hand pose features to understand actions can support multiple applications that require high precision, such as hand rehabilitation (Allin and Ramanan, 2007), virtual or augmented reality (Jang et al., 2015), teleoperation (Fritsche et al., 2015) and robot imitation learning (Argall et al., 2009).

Imitation learning using hand pose demonstrations

An interesting application of the benchmark and framework in Chapter 5 is the use of the recorded actions to demonstrate an imitation-learning framework. Recent work (Amor et al., 2012; Deimel and Brock, 2016; Rajeswaran et al., 2017) has explored the use of anthropomorphic hand models for dexterous manipulation task imitation by humans. An important drawback of these approaches is the need for specialised hardware, such as gloves, to record trajectories of human interaction with objects. Our dataset reflects a step forward in training hand pose and object pose estimators in unconstrained environments, and our recorded sequences can thus serve as human demonstrations. Defining the state of the environment according to fine-grained features, such as hand pose and object pose, has the advantage of easy transformation between domains compared to the use of raw images (Stadie et al., 2017). However, the employment of human hand poses to guide a robot is not straightforward, and two components warrant further study. First, hand poses should be mapped to the robotic hand, which may have different degrees of freedom and kinematics. Second, our hand pose estimator produces noisy estimates, which should be taken into account when transferring the knowledge to the robot. An interesting approach would be to record trajectories in a physics simulator, such as that in Kumar and Todorov (2015) and similarly to Rajeswaran et al. (2017), of humans grasping virtual objects. This approach could possibly reduce the amount of noise due to occlusion from objects. Furthermore, working on a virtual environment would permit the exploitation of recent advances that require a vast amount of sampling in imitation learning (Ho and Ermon, 2016) and reinforcement learning (Rajeswaran et al., 2017; Peng et al., 2018; Zhu et al., 2018).

BIBLIOGRAPHY

Aggarwal, J. K. and Ryoo, M. S. (2011), 'Human activity analysis: A review', *ACM Computing Surveys (CSUR)* . (Cited on pages 2 and 11.)

Aggarwal, R., Swetha, S., Namboodiri, A. M., Sivaswamy, J. and Jawahar, C. (2015), Online handwriting recognition using depth sensors, *in* 'Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR)', IEEE, pp. 1061–1065. (Cited on page 29.)

Allin, S. and Ramanan, D. (2007), Assessment of post-stroke functioning using machine vision., *in* 'Proceedings of the IAPR Conference on Machine Vision Applications (MVA)', pp. 299–302. (Cited on page 115.)

Alon, J., Athitsos, V., Yuan, Q. and Sclaroff, S. (2009), 'A unified framework for gesture recognition and spatiotemporal gesture segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **31**(9), 1685–1699. (Cited on page 28.)

Amor, H. B., Kroemer, O., Hillenbrand, U., Neumann, G. and Peters, J. (2012), Generalization of human grasping for multi-fingered robot hands, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 2043–2050. (Cited on page 115.)

Argall, B. D., Chernova, S., Veloso, M. and Browning, B. (2009), 'A survey of robot learning from demonstration', *Robotics and Autonomous Systems* **57**(5), 469–483. (Cited on page 115.)

Baek, S., Kim, K. I. and Kim, T.-K. (2017), Real-time online action detection forests using spatio-temporal contexts, *in* 'Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)', pp. 158–167. (Cited on pages 24 and 56.)

Baek, S., Kim, K. I. and Kim, T.-K. (2018), Augmented skeleton space transfer for depth-based hand pose estimation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 8330–8339. (Cited on page 85.)

Baek, S., Shi, Z., Kawade, M. and Kim, T.-K. (2017), Kinematic-aware random forest for static and dynamic action recognition from depth sequences., *in* 'Proceedings of the British Machine Vision Conference (BMVC)'. (Cited on pages 5, 18, and 24.)

Bambach, S., Lee, S., Crandall, D. J. and Yu, C. (2015), Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 1949–1957. (Cited on pages 22, 24, 84, and 94.)

Belongie, S., Malik, J. and Puzicha, J. (2002), 'Shape matching and object recognition using shape contexts', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **24**(4), 509–522. (Cited on page 30.)

Bhuyan, M., Neog, D. R. and Kar, M. K. (2012), 'Fingertip detection for hand pose recognition', *International Journal on Computer Science and Engineering* **4**(3), 501. (Cited on pages 31, 35, 46, and 48.)

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A. and Gould, S. (2016), Dynamic image networks for action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3034–3042. (Cited on page 16.)

Blank, M., Gorelick, L., Shechtman, E., Irani, M. and Basri, R. (2005), Actions as space-time shapes, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', Vol. 2, pp. 1395–1402. (Cited on pages 12 and 17.)

Bobick, A. F. and Davis, J. W. (2001), 'The recognition of human movement using temporal templates', *IEEE Transactions on Pattern Aanalysis and Machine Intelligence (TPAMI)* **23**(3), 257–267. (Cited on pages 4 and 12.)

Bourennane, S. and Fossati, C. (2012), 'Comparison of shape descriptors for hand posture recognition in video', *Signal, Image and Video Processing* **6**(1), 147–157. (Cited on page 30.)

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32. (Cited on pages xix, 7, 34, 43, 48, 65, and 96.)

Bullock, I. M., Feix, T. and Dollar, A. M. (2015), 'The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments', *The International Journal of Robotics Research* **34**(3), 251–255. (Cited on pages 84 and 94.)

Cai, M., Kitani, K. M. and Sato, Y. (2015), A scalable approach for understanding the visual structures of hand grasps, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)', pp. 1360–1366. (Cited on pages 84 and 94.)

Cai, M., Kitani, K. M. and Sato, Y. (2016), Understanding hand-object manipulation with grasp types and object attributes, *in* 'Robotics: Science and Systems (RSS)'. (Cited on pages 82, 84, and 93.)

Cao, Z., Simon, T., Wei, S.-E. and Sheikh, Y. (2017), Realtime multi-person 2d pose estimation using part affinity fields, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 7291–7299. (Cited on pages 5 and 12.)

Carreira, J. and Zisserman, A. (2017), Quo vadis, action recognition? a new model and the kinetics dataset, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4724–4733. (Cited on pages 4 and 17.)

Chang, H. J., Garcia-Hernando, G., Tang, D. and Kim, T.-K. (2016), 'Spatio-temporal hough forest for efficient detection–localisation–recognition of fingerwriting in egocentric camera', *Computer Vision and Image Understanding* **148**, 87–96. (Cited on pages 9 and 28.)

Charles, J., Pfister, T., Magee, D., Hogg, D. and Zisserman, A. (2014), Upper body pose estimation with temporal sequential forests, *in* 'Proceedings of the British Machine Vision Conference 2014', BMVA Press, pp. 1–12. (Cited on page 25.)

Chen, F.-S., Fu, C.-M. and Huang, C.-L. (2003), 'Hand gesture recognition using a real-time tracking method and hidden markov models', *Image and Vision Computing* **21**(8), 745–758. (Cited on pages 30, 34, 43, 45, and 48.)

Chen, J., Le, H. M., Carr, P., Yue, Y. and Little, J. J. (2016), Learning online smooth predictors for realtime camera planning using recurrent decision trees, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4688–4696. (Cited on page 25.)

Chéron, G., Laptev, I. and Schmid, C. (2015), P-cnn: Pose-based cnn features for action recognition, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3218–3226. (Cited on page 12.)

Choi, C., Yoon, S. H., Chen, C.-N. and Ramani, K. (2017), Robust hand pose estimation during the interaction with an unknown object, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3123–3132. (Cited on pages 85 and 94.)

Comaniciu, D. and Meer, P. (2002), 'Mean shift: a robust approach toward feature space analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **24**(5), 603–619. (Cited on page 38.)

Conseil, S., Bourennane, S. and Martin, L. (2007), Comparison of fourier descriptors and hu moments for hand posture, *in* 'European Signal Processing Conference (EUSIPCO)', pp. 1960–1964. (Cited on page 30.)

Dalal, N. and Triggs, B. (2005), Histograms of oriented gradients for human detection, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', Vol. 1, pp. 886–893. (Cited on pages xix and 13.)

Dalal, N., Triggs, B. and Schmid, C. (2006), Human detection using oriented histograms of flow and appearance, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 428–441. (Cited on pages xix and 14.)

Damen, D., Gee, A., Mayol-Cuevas, W. and Calway, A. (2012), Egocentric real-time workspace monitoring using an rgb-d camera, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1029–1036. (Cited on page 84.)

Damen, D., Leelasawassuk, T., Haines, O., Calway, A. and Mayol-Cuevas, W. W. (2014), You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video., *in* 'Proceedings of the British Machine Vision Conference (BMVC)', Vol. 2, p. 4. (Cited on pages 23 and 24.)

Dapogny, A., Bailly, K. and Dubuisson, S. (2015), Pairwise conditional random forests for facial expression recognition, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3783–3791. (Cited on pages xix, 7, 25, 55, 65, 68, 71, 74, and 96.)

Darrell, T. and Pentland, A. (1993), Space-time gestures, *in* 'Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 335–340. (Cited on page 15.)

De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C. and Tuytelaars, T. (2016), Online action detection, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 269–284. (Cited on pages 3 and 56.)

De Smedt, Q., Wannous, H. and Vandeborre, J.-P. (2016), Skeleton-based dynamic hand gesture recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 1–9. (Cited on pages 82 and 84.)

Deimel, R. and Brock, O. (2016), 'A novel type of compliant and underactuated robotic hand for dexterous grasping', *The International Journal of Robotics Research* **35**(1-3), 161–185. (Cited on page 115.)

Devanne, M., Wannous, H., Pala, P., Berretti, S., Daoudi, M. and Del Bimbo, A. (2015), Combined shape analysis of human poses and motion units for action segmentation and recognition, *in* 'Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)', Vol. 7, pp. 1–6. (Cited on pages 19 and 68.)

Dollár, P., Rabaud, V., Cottrell, G. and Belongie, S. (2005), Behavior recognition via sparse spatio-temporal features, *in* '2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance.', pp. 65–72. (Cited on pages 12 and 13.)

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T. (2015), Long-term recurrent convolutional networks for visual recognition and description, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2625–2634. (Cited on pages 4 and 16.)

Du, Y., Wang, W. and Wang, L. (2015), Hierarchical recurrent neural network for skeleton based action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1110–1118. (Cited on pages 20, 69, 71, 96, 101, and 115.)

Duong, T. V., Bui, H. H., Phung, D. Q. and Venkatesh, S. (2005), Activity recognition and abnormality detection with the switching hidden semi-markov model, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', Vol. 1, pp. 838–845. (Cited on page 15.)

Efros, A. A., Berg, A. C., Mori, G. and Malik, J. (2003), Recognizing action at a distance, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', Vol. 2, p. 726. (Cited on pages 4 and 12.)

Fathi, A., Farhadi, A. and Rehg, J. M. (2011), Understanding egocentric activities, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 407–414. (Cited on pages 6, 22, 23, and 84.)

Fathi, A., Li, Y. and Rehg, J. M. (2012), Learning to recognize daily actions using gaze, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 314–327. (Cited on pages 23 and 24.)

Fathi, A., Ren, X. and Rehg, J. M. (2011), Learning to recognize objects in egocentric activities, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3281–3288. (Cited on pages 22, 23, 84, 87, and 94.)

Feichtenhofer, C., Pinz, A. and Wildes, R. P. (2017), Spatiotemporal multiplier networks for video action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 7445–7454. (Cited on page 16.)

Feichtenhofer, C., Pinz, A. and Zisserman, A. (2016), Convolutional two-stream network fusion for video action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1933–1941. (Cited on pages 4, 16, 17, 18, 23, 95, 100, 101, and 104.)

Feldman, J. and Singh, M. (2005), 'Information along contours and object boundaries.', *Psychological review* **112**(1), 243. (Cited on pages 32 and 33.)

Feng, Z., Xu, S., Zhang, X., Jin, L., Ye, Z. and Yang, W. (2012), 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'. (Cited on page 29.)

Fermüller, C., Wang, F., Yang, Y., Zampogiannis, K., Zhang, Y., Barranco, F. and Pfeiffer, M. (2018), 'Prediction of manipulation actions', *International Journal of Computer Vision (IJCV)* **126**(2-4), 358–374. (Cited on page 84.)

Fernando, B., Gavves, E., Oramas, J. M., Ghodrati, A. and Tuytelaars, T. (2015), Modeling video evolution for action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 5378–5387. (Cited on page 15.)

Fothergill, S., Mentis, H., Kohli, P. and Nowozin, S. (2012), Instructing people for training gestural interactive systems, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', pp. 1737–1746. (Cited on pages xx, 4, 7, 20, 21, 24, 55, 56, 65, 66, and 96.)

Fritsche, L., Unverzag, F., Peters, J. and Calandra, R. (2015), First-person teleoperation of a humanoid robot, *in* 'Proceedings of the IEEE-RAS 15th International Conference onHumanoid Robots (Humanoids)', pp. 997–1002. (Cited on page 115.)

Gall, J., Yao, A., Razavi, N., Van Gool, L. and Lempitsky, V. (2011), 'Hough forests for object detection, tracking, and action recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **33**(11), 2188–2202. (Cited on pages 7, 25, 27, 28, 37, 48, 50, 51, and 55.)

Garcia-Hernando, G., Chang, H. J., Serrano, I., Deniz, O. and Kim, T.-K. (2016), Transition hough forest for trajectory-based action recognition, *in* 'Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)', pp. 1–8. (Cited on page 9.)

Garcia-Hernando, G. and Kim, T.-K. (2017), Transition forests: Learning discriminative temporal transitions for action recognition and detection, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 407–415. (Cited on page 9.)

Garcia-Hernando, G., Yuan, S., Baek, S. and Kim, T.-K. (2018), First-person hand action benchmark with rgb-d videos and 3d hand pose annotations, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (Cited on page 10.)

Gavrila, D. M., Davis, L. S. et al. (1995), Towards 3-d model-based tracking and recognition of human movement: a multi-view approach, *in* 'Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)', pp. 272–277. (Cited on page 15.)

Geurts, P., Ernst, D. and Wehenkel, L. (2006), 'Extremely randomized trees', *Machine learning* **63**(1), 3–42. (Cited on page 62.)

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014), Rich feature hierarchies for accurate object detection and semantic segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 580–587. (Cited on page 15.)

Hameed, M. Z., Garcia-Hernando, G. and Kim, T.-K. (2015), Novel spatio-temporal features for fingertip writing recognition in egocentric viewpoint, *in* 'Proceedings of the IAPR International Conference on Machine Vision Applications (MVA)', pp. 484–488. (Cited on page 29.)

Hamer, H., Gall, J., Weise, T. and Van Gool, L. (2010), An object-dependent hand pose prior from sparse training data, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 671–678. (Cited on page 85.)

Hannuksela, J., Huttunen, S., Sangi, P. and Heikkila, J. (2007), Motion-based finger tracking for user interaction with mobile devices, *in* 'Proceedings of the 4th European Conference on Visual Media Production', IET, pp. 1–6. (Cited on page 29.)

Harris, C. and Stephens, M. (1988), A combined corner and edge detector., *in* 'Alvey vision conference', Vol. 15, Manchester, UK, pp. 10–5244. (Cited on page 12.)

Herath, S., Harandi, M. and Porikli, F. (2017), 'Going deeper into action recognition: A survey', *Image and Vision Computing* **60**, 4–21. (Cited on pages 2, 11, and 15.)

Ho, J. and Ermon, S. (2016), Generative adversarial imitation learning, *in* 'Advances in Neural Information Processing Systems (NIPS)', pp. 4565–4573. (Cited on page 116.)

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780. (Cited on pages xix, 16, 96, and 97.)

Hu, J.-F., Zheng, W.-S., Lai, J. and Zhang, J. (2015), Jointly learning heterogeneous features for rgb-d activity recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 5344–5352. (Cited on pages xix, 19, 95, 96, 101, and 104.)

Hu, K. and Yin, L. (2013), Multi-scale topological features for hand posture representation and analysis, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 1928–1935. (Cited on pages 29, 30, 43, and 45.)

Huang, D.-A., Ma, M., Ma, W.-C. and Kitani, K. M. (2015), How do we use our hands? discovering a diverse set of common grasps, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 666–675. (Cited on page 84.)

Huang, Y., Liu, X., Zhang, X. and Jin, L. (2016), A pointing gesture based egocentric interaction system: Dataset, approach and application, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 16–23. (Cited on pages 29 and 114.)

Huang, Z., Wan, C., Probst, T. and Van Gool, L. (2017), Deep learning on lie groups for skeleton-based action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1243–1252. (Cited on page 20.)

Intel (2013), Perceptual computing sdk. (Cited on page 96.)

Ionescu, C., Carreira, J. and Sminchisescu, C. (2014), Iterated second-order label sensitive pooling for 3d human pose estimation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1661–1668. (Cited on page 85.)

Iqbal, U., Garbade, M. and Gall, J. (2017), Pose for action-action for pose, *in* 'Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)', pp. 438–445. (Cited on page 115.)

Ishida, H., Takahashi, T., Ide, I. and Murase, H. (2010), 'A hilbert warping method for handwriting gesture recognition', *Pattern Recognition* **43**(8), 2799–2806. (Cited on page 29.)

Ishihara, T., Kitani, K. M., Ma, W.-C., Takagi, H. and Asakawa, C. (2015), Recognizing hand-object interactions in wearable camera videos, *in* 'Proceedings of the IEEE International Conference on Image Processing (ICIP)', pp. 1349–1353. (Cited on pages 6, 22, 23, 82, 84, and 104.)

Ivanov, Y. A. and Bobick, A. F. (2000), 'Recognition of visual activities and interactions by stochastic parsing', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **22**(8), 852–872. (Cited on page 15.)

Jang, Y., Noh, S.-T., Chang, H. J., Kim, T.-K. and Woo, W. (2015), '3D finger CAPE: Clicking action and position estimation under self-occlusions in egocentric viewpoint', *IEEE Transactions on Visualization and Computer Graphics* (99). (Cited on pages 4, 25, and 115.)

Ji, S., Xu, W., Yang, M. and Yu, K. (2013), '3d convolutional neural networks for human action recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **35**(1), 221–231. (Cited on page 15.)

Jin, L., Yang, D., Zhen, L.-X. and Huang, J.-C. (2007), 'A novel vision-based fingerwriting character recognition system', *Journal of Circuits, Systems, and Computers* **16**(03), 421–436. (Cited on page 29.)

Jung, H.-J. and Hong, K.-S. (2014), Enhanced sequence matching for action recognition from 3d skeletal data, *in* 'Asian Conference on Computer Vision (ACCV)', Springer, pp. 226–240. (Cited on pages 19, 68, and 69.)

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014), Large-scale video classification with convolutional neural networks, *in* 'Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1725–1732. (Cited on pages 15 and 17.)

Ke, Q., Bennamoun, M., An, S., Sohel, F. and Boussaid, F. (2017), A new representation of skeleton sequences for 3d action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4570–4579. (Cited on page 20.)

Ke, Y., Sukthankar, R. and Hebert, M. (2005), Efficient visual event detection using volumetric features, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', Vol. 1, pp. 166–173. (Cited on page 3.)

Keskin, C., Kıraç, F., Kara, Y. E. and Akarun, L. (2012), Hand pose estimation and hand shape classification using multi-layered randomized decision forests, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 852–863. (Cited on page 85.)

Klaser, A., Marszałek, M. and Schmid, C. (2008), A spatio-temporal descriptor based on 3d-gradients, *in* 'Proceedings of the British Machine Vision Conference (BMVC)', pp. 275–1. (Cited on page 13.)

Kontschieder, P., Kohli, P., Shotton, J. and Criminisi, A. (2013), Geof: Geodesic forests for learning coupled predictors, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 65–72. (Cited on page 25.)

Krejov, P. and Bowden, R. (2013), Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima, *in* 'Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)', pp. 1–7. (Cited on page 31.)

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Advances in Neural Information Processing Systems (NIPS)', pp. 1097–1105. (Cited on page 15.)

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. and Serre, T. (2011), Hmdb: a large video database for human motion recognition, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 2556–2563. (Cited on pages 4 and 17.)

Kumar, V. and Todorov, E. (2015), Mujoco haptix: A virtual reality system for hand manipulation, *in* 'Proceedings of the IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)', pp. 657–663. (Cited on page 116.)

Laptev, I. (2005), 'On space-time interest points', *International Journal of Computer Vision (IJCV)* **64**(2-3), 107–123. (Cited on pages 4, 12, 13, and 14.)

Laptev, I., Marszalek, M., Schmid, C. and Rozenfeld, B. (2008), Learning realistic human actions from movies, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1–8. (Cited on pages xix, 13, and 14.)

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989), 'Backpropagation applied to handwritten zip code recognition', *Neural computation* **1**(4), 541–551. (Cited on page 15.)

Lee, T. and Hollerer, T. (2007), Handy ar: Markerless inspection of augmented reality objects using fingertip tracking, *in* 'Proceedings of the 11th IEEE International Symposium on Wearable Computers'. (Cited on pages 31, 33, and 35.)

Léger, J. C. (1999), 'Menger curvature and rectifiability', *Annals of Mathematics* **149**(3), 831–869. (Cited on page 36.)

Lehrmann, A. M., Gehler, P. V. and Nowozin, S. (2014), Efficient nonlinear markov models for human motion, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1314–1321. (Cited on pages xix, 4, 7, 19, 21, 25, 55, 65, 66, and 68.)

Lei, J., Ren, X. and Fox, D. (2012), Fine-grained kitchen activity recognition using rgb-d, *in* 'Proceedings of the 2012 ACM Conference on Ubiquitous Computing', pp. 208–211. (Cited on page 84.)

Li, W., Zhang, Z. and Liu, Z. (2010), Action recognition based on a bag of 3d points, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 9–14. (Cited on pages 20, 21, 66, and 69.)

Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C. and Liu, J. (2016), Online human action detection using joint classification-regression recurrent neural networks, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 203–220. (Cited on pages 22, 26, 56, 75, 76, and 77.)

Li, Y., Ye, Z. and Rehg, J. M. (2015), Delving into egocentric actions, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 287–295. (Cited on page 23.)

Li, Z., Gavrilyuk, K., Gavves, E., Jain, M. and Snoek, C. G. (2018), 'Videolstm convolves, attends and flows for action recognition', *Computer Vision and Image Understanding* **166**, 41–50. (Cited on page 17.)

Liang, H., Yuan, J. and Thalmann, D. (2012), 3d fingertip and palm tracking in depth image sequences, *in* 'Proceedings ACM international Conference on Multimedia', pp. 785–788. (Cited on page 31.)

Liang, H., Yuan, J. and Thalmann, D. (2014), 'Parsing the hand in depth images', *IEEE Transactions on Multimedia* **16**(5), 1241–1253. (Cited on page 85.)

Liu, J., Wang, G., Hu, P., Duan, L.-Y. and Kot, A. C. (2017), Global context-aware attention lstm networks for 3d action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', Vol. 7, p. 43. (Cited on page 20.)

Liu, L. and Shao, L. (2013), Learning discriminative representations from rgb-d video data., *in* 'Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)', pp. 1493–1500. (Cited on page 82.)

Liu, Y., Liu, X. and Jia, Y. (2006), Hand-gesture based text input for wearable computers, *in* 'Proceedings of the IEEE International Conference on Computer Vision Systems', pp. 8–8. (Cited on page 29.)

Lowe, D. G. (2004), 'Distinctive image features from scale-invariant keypoints', *International Journal of Computer Vision (IJCV)* **60**(2), 91–110. (Cited on page 13.)

Lucas, B. D., Kanade, T. et al. (1981), An iterative image registration technique with an application to stereo vision, *in* 'Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)', Vol. 2, pp. 674–679. (Cited on page 14.)

Lv, F. and Nevatia, R. (2007), Single view human action recognition using key pose matching and viterbi path searching, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', Vol. 2, pp. 194–201. (Cited on page 15.)

Ma, M., Fan, H. and Kitani, K. M. (2016), Going deeper into first-person activity recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1894–1903. (Cited on pages 22, 23, 82, 84, and 95.)

Ma, S., Zhang, J., Sclaroff, S., Ikizler-Cinbis, N. and Sigal, L. (2017), 'Space-time tree ensemble for action recognition and localization', *International Journal of Computer Vision (IJCV)* pp. 1–19. (Cited on page 15.)

Maaten, L. v. d. and Hinton, G. (2008), 'Visualizing data using t-sne', *Journal of machine learning research* **9**(Nov), 2579–2605. (Cited on page 89.)

Maisto, M., Panella, M., Liparulo, L. and Proietti, A. (2013), 'An accurate algorithm for the identification of fingertips using an rgb-d camera', *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **3**(2), 272–283. (Cited on page 31.)

Marszałek, M., Laptev, I. and Schmid, C. (2009), Actions in context, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2929–2936. (Cited on page 17.)

Matikainen, P., Hebert, M. and Sukthankar, R. (2009), Trajectons: Action recognition through the motion analysis of tracked features, *in* 'Proceedings of the IEEE International Conference on Computer Vision Workshops', pp. 514–521. (Cited on page 14.)

Mayol, W. W. and Murray, D. W. (2005), Wearable hand activity recognition for event summarization, *in* 'Proceedings of the IEEE International Symposium on Wearable Computers', pp. 122–129. (Cited on pages 6 and 22.)

Meshry, M., Hussein, M. E. and Torki, M. (2016), Linear-time online action detection from 3d skeletal data using bags of gesturelets, *in* 'Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)'. (Cited on page 56.)

Messing, R., Pal, C. and Kautz, H. (2009), Activity recognition using the velocity histories of tracked keypoints, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 104–111. (Cited on page 14.)

Mikolajczyk, K. and Uemura, H. (2011), 'Action recognition with appearance–motion features and fast search trees', *Computer Vision and Image Understanding* **115**(3), 426–438. (Cited on page 24.)

Moghimi, M., Azagra, P., Montesano, L., Murillo, A. C. and Belongie, S. (2014), Experiments on an rgb-d wearable vision system for egocentric activity recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops'. (Cited on page 84.)

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S. and Kautz, J. (2016), Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4207–4215. (Cited on page 82.)

Moosmann, F., Triggs, B. and Jurie, F. (2007), Fast discriminative visual codebooks using randomized clustering forests, *in* 'Advances on Neural Information Processing Systems (NIPS)', pp. 985–992. (Cited on page 39.)

Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D. and Theobalt, C. (2017), Real-time hand tracking under occlusion from an egocentric rgb-d sensor, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 1163–1172. (Cited on pages 85, 94, and 95.)

Neverova, N., Wolf, C., Taylor, G. W. and Nebout, F. (2014), Hand segmentation with structured convolutional learning, *in* 'Asian Conference on Computer Vision (ACCV)', Springer, pp. 687–702. (Cited on page 85.)

Niebles, J. C., Wang, H. and Fei-Fei, L. (2008), 'Unsupervised learning of human action categories using spatial-temporal words', *International Journal of Computer Vision (IJCV)* **79**(3), 299–318. (Cited on page 13.)

Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B. and Kohli, P. (2011), Decision tree fields, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 1668–1675. (Cited on page 25.)

Oberweger, M., Riegler, G., Wohlhart, P. and Lepetit, V. (2016), Efficiently creating 3d training data for fine hand pose estimation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4957–4965. (Cited on pages 85 and 94.)

Oberweger, M., Wohlhart, P. and Lepetit, V. (2015), Training a feedback loop for hand pose estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3316–3324. (Cited on page 85.)

Ohn-Bar, E. and Trivedi, M. M. (2014), 'Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations', *IEEE Transactions on Intelligent Transportation Systems* **15**(6), 2368–2377. (Cited on pages 18, 19, 82, 95, and 101.)

Oikonomidis, I., Kyriazis, N. and Argyros, A. A. (2011*a*), Efficient model-based 3d tracking of hand articulations using kinect, *in* 'Proceedings of the British Machine Vision Conference (BMVC)', Vol. 1, p. 3. (Cited on pages 29, 85, and 96.)

Oikonomidis, I., Kyriazis, N. and Argyros, A. A. (2011*b*), Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 2088–2095. (Cited on page 96.)

Oka, K., Sato, Y. and Koike, H. (2002), 'Real-time fingertip tracking and gesture recognition', *IEEE Computer Graphics and Applications* **22**(6), 64–71. (Cited on page 28.)

Oliver, N., Horvitz, E. and Garg, A. (2002), Layered representations for human activity recognition, *in* 'Proceedings of the IEEE International Conference on Multimodal Interfaces'. (Cited on page 15.)

Oliver, N. M., Rosario, B. and Pentland, A. P. (2000), 'A bayesian computer vision system for modeling human interactions', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **22**(8), 831–843. (Cited on page 15.)

Oneata, D., Verbeek, J. and Schmid, C. (2013), Action and event recognition with fisher vectors on a compact feature set, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 1817–1824. (Cited on page 14.)

Oreifej, O. and Liu, Z. (2013), HON4D: histogram of oriented 4D normals for activity recognition from depth sequences, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 716–723. (Cited on pages 18, 95, and 101.)

Pan, Z., Li, Y., Zhang, M., Sun, C., Guo, K., Tang, X. and Zhou, S. (2010), A real-time multi-cue hand tracking algorithm based on computer vision, *in* 'Proceedings of the IEEE Virtual Reality Conference (VR)', pp. 219–222. (Cited on pages 31 and 35.)

Peng, X. B., Abbeel, P., Levine, S. and van de Panne, M. (2018), 'Deepmimic: Example-guided deep reinforcement learning of physics-based character skills', *arXiv preprint arXiv:1804.02717* . (Cited on page 116.)

Peng, X., Zou, C., Qiao, Y. and Peng, Q. (2014), Action recognition with stacked fisher vectors, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 581–595. (Cited on page 14.)

Perronnin, F., Sánchez, J. and Mensink, T. (2010), Improving the fisher kernel for large-scale image classification, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 143–156. (Cited on page 14.)

Persoon, E. and Fu, K.-S. (1977), 'Shape discrimination using fourier descriptors', *IEEE Transactions on Systems, Man and Cybernetics* **7**(3), 170–179. (Cited on pages xix, 30, and 32.)

Pirsiavash, H. and Ramanan, D. (2012), Detecting activities of daily living in first-person camera views, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2847–2854. (Cited on pages 22, 23, 24, 84, and 94.)

Qian, C., Sun, X., Wei, Y., Tang, X. and Sun, J. (2014), Realtime and robust hand tracking from depth, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1106–1113. (Cited on page 85.)

Quattoni, A., Wang, S., Morency, L.-P., Collins, M. and Darrell, T. (2007), 'Hidden conditional random fields', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **29**(10). (Cited on page 15.)

Rabiner, L. R. (1989), 'A tutorial on hidden markov models and selected applications in speech recognition', *Proceedings of the IEEE* **77**(2), 257–286. (Cited on page 39.)

Raheja, J. L., Chaudhary, A. and Singal, K. (2011), Tracking of fingertips and centers of palm using kinect, *in* 'Proceedings of the IEEE International Conference on Computational Intelligence, Modelling and Simulation', pp. 248–252. (Cited on pages 29 and 31.)

Raheja, J. L., Das, K. and Chaudhary, A. (2012), 'Fingertip detection: a fast method with natural hand', *International Journal of Embedded Systems and Computer Engineering* **3**(2), 85–89. (Cited on pages 31 and 46.)

Rahmani, H., Mahmood, A., Huynh, D. and Mian, A. (2016), 'Histogram of oriented principal components for cross-view action recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **38**(12), 2430–2443. (Cited on page 18.)

Rahmani, H. and Mian, A. (2016), 3d action recognition from novel viewpoints, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1506–1515. (Cited on pages 18, 20, 95, 100, 101, 102, and 104.)

Rajeswaran, A., Kumar, V., Gupta, A., Schulman, J., Todorov, E. and Levine, S. (2017), 'Learning complex dexterous manipulation with deep reinforcement learning and demonstrations', *arXiv preprint arXiv:1709.10087* . (Cited on pages 115 and 116.)

Raptis, M. and Sigal, L. (2013), Poselet key-framing: A model for human activity recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2650–2657. (Cited on pages 15, 50, and 51.)

Rodriguez, M. D., Ahmed, J. and Shah, M. (2008), Action mach a spatio-temporal maximum average correlation height filter for action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1–8. (Cited on page 17.)

Rogez, G., Khademi, M., Supančič III, J., Montiel, J. M. M. and Ramanan, D. (2014), 3d hand pose detection in egocentric rgb-d images, *in* 'Workshop at the European Conference on Computer Vision', Springer, pp. 356–371. (Cited on pages 29, 30, 82, 84, 85, and 94.)

Rogez, G., Supancic, J. S. and Ramanan, D. (2015*a*), First-person pose recognition using egocentric workspaces, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4325–4333. (Cited on page 82.)

Rogez, G., Supancic, J. S. and Ramanan, D. (2015*b*), Understanding everyday hands in action from rgb-d images, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3889–3897. (Cited on pages 4, 82, 84, 88, 93, 94, 108, and 115.)

Romero, J., Kjellström, H., Ek, C. H. and Kragic, D. (2013), 'Non-parametric hand pose estimation with object context', *Image and Vision Computing* **31**(8), 555–564. (Cited on page 85.)

Ryoo, M. S. and Aggarwal, J. K. (2009), 'Semantic representation and recognition of continued and recursive human activities', *International Journal of Computer Vision (IJCV)* **82**(1), 1–24. (Cited on page 15.)

Ryoo, M. S. and Aggarwal, J. K. (2010), 'UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA)', http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html. (Cited on pages 13, 17, and 50.)

Ryoo, M. S. and Matthies, L. (2013), First-person activity recognition: What are they doing to me?, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2730–2737. (Cited on pages 22 and 24.)

Schick, A., Morlock, D., Amma, C., Schultz, T. and Stiefelhagen, R. (2012), Vision-based handwriting recognition for unrestricted text input in mid-air, *in* 'Proceedings of the ACM international Conference on Multimodal Interaction', pp. 217–220. (Cited on page 28.)

Schuldt, C., Laptev, I. and Caputo, B. (2004), Recognizing human actions: a local svm approach, *in* 'Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)', Vol. 3, pp. 32–36. (Cited on page 17.)

Scovanner, P., Ali, S. and Shah, M. (2007), A 3-dimensional sift descriptor and its application to action recognition, *in* 'Proceedings of the ACM International Conference on Multimedia', pp. 357–360. (Cited on page 13.)

Seidenari, L., Varano, V., Berretti, S., Bimbo, A. and Pala, P. (2013), Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 479–485. (Cited on pages 5, 20, 21, 24, 66, 73, and 74.)

Serrano, I., Deniz, O., Bueno, G., Garcia-Hernando, G. and Kim, T.-K. (2018), 'Spatio-temporal elastic cuboid trajectories for efficient fight recognition using hough forests', *Machine Vision and Applications* **29**(2), 207–217. (Cited on page 25.)

Shah, S., Ahmed, A., Mahmood, I. and Khurshid, K. (2011), Hand gesture based user interface for computer using a camera and projector, *in* 'Proceedings of the IEEE International Conference on Signal and Image Processing Applications', pp. 168–173. (Cited on page 29.)

Shahroudy, A., Liu, J., Ng, T.-T. and Wang, G. (2016), Ntu rgb+d: A large scale dataset for 3d human activity analysis, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1010–1019. (Cited on pages 4, 20, and 82.)

Shahroudy, A., Ng, T.-T., Yang, Q. and Wang, G. (2016), 'Multimodal multipart learning for action recognition in depth videos', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **38**(10), 2123–2129. (Cited on page 19.)

Sharaf, A., Torki, M., Hussein, M. E. and El-Saban, M. (2015), Real-time multi-scale action detection from 3d skeleton data, *in* 'Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)', IEEE, pp. 998–1005. (Cited on page 56.)

Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y. et al. (2015), Accurate, robust, and flexible real-time hand tracking, *in* 'Proceedings of the ACM Conference on Human Factors in Computing Systems', pp. 3633–3642. (Cited on page 85.)

Shi, Z. and Kim, T.-K. (2017), Learning and refining of privileged information-based rnns for action recognition from depth sequences, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4684–4693. (Cited on page 19.)

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A. et al. (2013), 'Efficient human pose estimation from single depth images', *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **35**(12), 2821–2840. (Cited on pages 5 and 20.)

Shotton, J., Johnson, M. and Cipolla, R. (2008), Semantic texton forests for image categorization and segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (Cited on pages 25 and 39.)

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M. and Moore, R. (2013), 'Real-time human pose recognition in parts from single depth images', *Communications of the ACM* **56**(1), 116–124. (Cited on pages 18 and 82.)

Shotton, J., Sharp, T., Kohli, P., Nowozin, S., Winn, J. and Criminisi, A. (2013), Decision jungles: Compact and rich models for classification, *in* 'Advances in Neural Information Processing Systems (NIPS)', pp. 234–242. (Cited on pages 25 and 58.)

Simonyan, K. and Zisserman, A. (2014), Two-stream convolutional networks for action recognition in videos, *in* 'Advances in Neural Information Processing Systems (NIPS)', pp. 568–576. (Cited on pages 4, 5, 16, 23, and 102.)

Singh, S., Arora, C. and Jawahar, C. (2016), First person action recognition using deep learned descriptors, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2620–2628. (Cited on pages 22, 23, 82, 84, and 95.)

Sminchisescu, C., Kanaujia, A. and Metaxas, D. (2006), 'Conditional models for contextual human motion recognition', *Computer Vision and Image Understanding* **104**(2), 210–220. (Cited on page 15.)

Soomro, K., Zamir, A. R. and Shah, M. (2012), 'Ucf101: A dataset of 101 human actions classes from videos in the wild', *arXiv preprint arXiv:1212.0402* . (Cited on page 17.)

Spriggs, E. H., De La Torre, F. and Hebert, M. (2009), Temporal segmentation and activity classification from first-person sensing, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 17–24. (Cited on page 23.)

Stadie, B. C., Abbeel, P. and Sutskever, I. (2017), 'Third-person imitation learning', *arXiv preprint arXiv:1703.01703* . (Cited on pages 6 and 116.)

Sun, X., Wei, Y., Liang, S., Tang, X. and Sun, J. (2015), Cascaded hand pose regression, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 824–832. (Cited on page 85.)

Sung, J., Ponce, C., Selman, B. and Saxena, A. (2011), Human activity detection from rgbd images., *in* 'Proceedings of the 16th AAAI Conference on Plan, Activity, and Intent Recognition', pp. 47–55. (Cited on page 20.)

Tang, D. (2015), 3D hand pose regression with variants of decision forests, PhD thesis, Imperial College London. (Cited on page 7.)

Tang, D., Chang, H. J., Tejani, A. and Kim, T.-K. (2014), Latent regression forest: Structured estimation of 3d articulated hand posture, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3786–3793. (Cited on pages 5, 29, 30, and 85.)

Tang, D., Yu, T.-H. and Kim, T.-K. (2013), Real-time articulated hand pose estimation using semi-supervised transductive regression forests, *in* ‘Proceedings of the IEEE International Conference on Computer Vision (ICCV)’, pp. 3224–3231. (Cited on pages 29, 30, and 85.)

Tang, K., Fei-Fei, L. and Koller, D. (2012), Learning latent temporal structure for complex event detection, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)’, pp. 1250–1257. (Cited on page 15.)

Tompson, J., Stein, M., Lecun, Y. and Perlin, K. (2014), ‘Real-time continuous pose recovery of human hands using convolutional networks’, *ACM Transactions on Graphics* **33**(5), 169. (Cited on page 85.)

Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M. (2015), Learning spatiotemporal features with 3d convolutional networks, *in* ‘Proceedings of the IEEE International Conference on Computer Vision (ICCV)’, pp. 4489–4497. (Cited on page 15.)

Varol, G., Laptev, I. and Schmid, C. (2017), ‘Long-term temporal convolutions for action recognition’, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* . (Cited on pages 15 and 16.)

Veeriah, V., Zhuang, N. and Qi, G.-J. (2015), Differential recurrent neural networks for action recognition, *in* ‘Proceedings of the IEEE International Conference on Computer Vision (ICCV)’, pp. 4041–4049. (Cited on pages 20 and 69.)

Vemulapalli, R., Arrate, F. and Chellappa, R. (2014), Human action recognition by representing 3d skeletons as points in a lie group, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)’, pp. 588–595. (Cited on pages xix, 19, 65, 68, 71, 74, 96, 97, 100, 101, 104, and 109.)

Vemulapalli, R. and Chellappa, R. (2016), Rolling rotations for recognizing human actions from 3d skeletal data, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)’, pp. 4471–4479. (Cited on pages 19 and 74.)

Vikram, S., Li, L. and Russell, S. (2013), Handwriting and gestures in the air, recognizing on the fly, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', Vol. 13, pp. 1179–1184. (Cited on pages 29 and 48.)

Wang, C., Wang, Y. and Yuille, A. L. (2013), An approach to pose-based action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 915–922. (Cited on page 12.)

Wang, C., Wang, Y. and Yuille, A. L. (2016), Mining 3d key-pose-motifs for action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2639–2647. (Cited on pages 19, 20, 21, 69, 71, 73, and 74.)

Wang, H., Kläser, A., Schmid, C. and Liu, C.-L. (2011), Action recognition by dense trajectories, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3169–3176. (Cited on pages 14 and 18.)

Wang, H. and Schmid, C. (2013), Action recognition with improved trajectories, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3551–3558. (Cited on pages xix, 5, 13, 14, 27, 50, and 51.)

Wang, H., Ullah, M. M., Klaser, A., Laptev, I. and Schmid, C. (2009), Evaluation of local spatio-temporal features for action recognition, *in* 'Proceedings of the British Machine Vision Conference (BMVC)', pp. 124–1. (Cited on page 13.)

Wang, J., Liu, Z., Chorowski, J., Chen, Z. and Wu, Y. (2012), Robust 3d action recognition with random occupancy patterns, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 872–885. (Cited on page 82.)

Wang, J., Liu, Z., Wu, Y. and Yuan, J. (2012), Mining actionlet ensemble for action recognition with depth cameras, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1290–1297. (Cited on pages xix, 5, 19, 20, and 69.)

Wang, P., Yuan, C., Hu, W., Li, B. and Zhang, Y. (2016), Graph based skeleton motion representation and similarity measurement for action recognition, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 370–385. (Cited on pages 20, 22, 71, 73, and 74.)

Wei, S.-E., Ramakrishna, V., Kanade, T. and Sheikh, Y. (2016), Convolutional pose machines, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4724–4732. (Cited on page 82.)

Weinland, D., Boyer, E. and Ronfard, R. (2007), Action recognition from arbitrary views using 3d exemplars, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 1–7. (Cited on page 15.)

Wetzler, A., Slossberg, R. and Kimmel, R. (2015), Rule of thumb: Deep derotation for improved fingertip detection, *in* 'Proceedings of the British Machine Vision Conference (BMVC)'. (Cited on page 85.)

Wong, S.-F., Kim, T.-K. and Cipolla, R. (2007), Learning motion categories using both semantic and structural information, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1–6. (Cited on page 13.)

Wray, M., Moltisanti, D., Mayol-Cuevas, W. and Damen, D. (2016), Sembed: Semantic embedding of egocentric action videos, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 532–545. (Cited on pages 24 and 88.)

Wu, D. and Shao, L. (2014), Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 724–731. (Cited on pages 19 and 82.)

Xia, L., Chen, C.-C. and Aggarwal, J. (2012), View invariant human action recognition using histograms of 3d joints, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 20–27. (Cited on pages 19 and 20.)

Yamato, J., Ohya, J. and Ishii, K. (1992), Recognizing human action in time-sequential images using hidden markov model, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 379–385. (Cited on page 15.)

Yang, M., Kpalma, K., Ronsin, J. et al. (2008), 'A survey of shape feature extraction techniques', *Pattern recognition* pp. 43–90. (Cited on page 30.)

Yang, X. and Tian, Y. (2014), Super normal vector for activity recognition using depth sequences, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 804–811. (Cited on page 18.)

Yang, Y., Fermuller, C., Li, Y. and Aloimonos, Y. (2015), Grasp type revisited: A modern perspective on a classical feature for vision, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 400–408. (Cited on pages 82 and 84.)

Yang, Y., Guha, A., Fermuller, C. and Aloimonos, Y. (2014), 'A cognitive system for understanding human manipulation actions', *Advances in Cognitive Sysytems* **3**, 67–86. (Cited on pages 84 and 88.)

Yao, A., Gall, J., Fanelli, G. and Gool, L. V. (2011), Does human action recognition benefit from pose estimation?., *in* 'Proceedings of the British Machine Vision Conference (BMVC)', pp. 67–1. (Cited on pages 4, 5, 7, 12, and 82.)

Ye, Q., Yuan, S. and Kim, T.-K. (2016), Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 346–361. (Cited on page 85.)

Yilmaz, A. and Shah, M. (2005), Actions sketch: A novel action representation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', Vol. 1, pp. 984–989. (Cited on page 12.)

Yu, T.-H., Kim, T.-K. and Cipolla, R. (2010), Real-time action recognition by spatiotemporal semantic and structural forests., *in* 'Proceedings of the British Machine Vision Conference (BMVC)', Vol. 2, p. 6. (Cited on pages 7, 24, 39, 50, 51, and 55.)

Yu, T.-H., Kim, T.-K. and Cipolla, R. (2013), Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3642–3649. (Cited on page 25.)

Yu, Y., Song, Y. and Zhang, Y. (2014), Real time fingertip detection with kinect depth image sequences, *in* 'Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)', pp. 550–555. (Cited on page 31.)

Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J. Y., Lee, K. M., Molchanov, P., Kautz, J., Honari, S., Ge, L., Yuan, J., Chen, X., Wang, G., Yang, F., Akiyama, K., Wu, Y., Wan, Q., Madadi, M., Escalera, S., Li, S., Lee, D., Oikonomidis, I., Argyros, A. and Kim, T.-K. (2018), Depth-based 3d hand pose estimation: From current achievements to future goals, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (Cited on pages 7 and 85.)

Yuan, S., Ye, Q., Garcia-Hernando, G. and Kim, T.-K. (2017), 'The 2017 hands in the million challenge on 3d hand pose estimation', *arXiv preprint arXiv:1707.02237* . (Cited on page 85.)

Yuan, S., Ye, Q., Stenger, B., Jain, S. and Kim, T.-K. (2017), Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2605–2613. (Cited on pages 85, 90, 94, 96, 108, 145, 146, 147, and 148.)

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. and Toderici, G. (2015), Beyond short snippets: Deep networks for video classification, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4694–4702. (Cited on page 16.)

Zabulis, X., Baltzakis, H. and Argyros, A. A. (2009), 'Vision-based hand gesture recognition for human-computer interaction.', *The universal access handbook* **34**, 30. (Cited on page 2.)

Zanfir, M., Leordeanu, M. and Sminchisescu, C. (2013), The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 2752–2759. (Cited on pages xix, 19, 56, 68, 69, 70, 96, 97, and 101.)

Zhang, D. and Lu, G. (2001), A comparative study on shape retrieval using fourier descriptors with different shape signatures, *in* 'Proceedings of the International Conference on Intelligent Multimedia and Distance Education', pp. 1–9. (Cited on page 30.)

Zhang, D. and Lu, G. (2004), 'Review of shape representation and description techniques', *Pattern recognition* **37**(1), 1–19. (Cited on page 30.)

Zhang, J., Li, W., Ogunbona, P. O., Wang, P. and Tang, C. (2016), 'Rgb-d-based action recognition datasets', *Pattern Recognition* **60**, 86–105. (Cited on page 11.)

Zhang, X., Wang, Y., Gou, M., Sznaier, M. and Camps, O. (2016), Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4498–4507. (Cited on pages 20, 69, 70, 71, 82, 96, 100, 101, 104, and 109.)

Zhang, X., Ye, Z., Jin, L., Feng, Z. and Xu, S. (2013), 'A new writing experience: Finger writing in the air using a kinect sensor', *MultiMedia, IEEE* **20**(4), 85–93. (Cited on page 29.)

Zhang, Y., Liu, X., Chang, M.-C., Ge, W. and Chen, T. (2012), Spatio-temporal phrases for activity recognition, *in* 'European Conference on Computer Vision (ECCV)', Springer, pp. 707–721. (Cited on pages 50 and 51.)

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. and Oliva, A. (2014), Learning deep features for scene recognition using places database, *in* 'Advances in Neural Information Processing Systems (NIPS)', pp. 487–495. (Cited on page 15.)

Zhu, G., Zhang, L., Shen, P. and Song, J. (2016), 'Human action recognition using multi-layer codebooks of key poses and atomic motions', *Signal Processing: Image Communication* **42**, 19–30. (Cited on pages 19 and 68.)

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L. and Xie, X. (2016), Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', pp. 3697–3703. (Cited on pages 20, 76, 96, and 97.)

Zhu, Y., Chen, W. and Guo, G. (2013), Fusing spatiotemporal features and joints for 3d action recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 486–491. (Cited on pages 7, 19, 24, 55, and 71.)

Zhu, Y., Wang, Z., Merel, J., Rusu, A., Erez, T., Cabi, S., Tunyasuvunakool, S., Kramár, J., Hadsell, R., de Freitas, N. et al. (2018), 'Reinforcement and imitation learning for diverse visuomotor skills', *arXiv preprint arXiv:1802.09564* . (Cited on page 116.)

A

APPENDIX A: HAND POSE ANNOTATION SYSTEM DETAILS

We follow the approach described by Yuan, Ye, Stenger, Jain and Kim (2017) to annotate the 3D hand pose with the help of six 6D magnetic sensors that provide their 3D location and 3D orientation. In Figure A.1 (a-b) the used 21-joint hand model is shown. Some physical constraints are applied: (1) wrist and 5 MCPs are relatively fixed (Figure A.1 Top (c)) (2) bone lengths are maintained, and (3) MCP, PIP, DIP and TIP are in the same plane for each finger.

As shown in Figure A.1 (top), five magnetic sensors are attached to the five fingertips: from thumb to pinky and denoted as S_1, S_2, S_3, S_4 and S_5. The sixth sensor, S_6, is attached to the back of the palm and it is used to infer the wrist and the five MCPs joints. For each finger i, sensor's orientation is used to find the three orthogonal axes, with V_1^i in the direction along the finger and V_2^i orthogonal to V_1^i pointing to the inside of the finger (Figure A.1 (bottom)). Using this, TIP's location T^i and DIP's location D^i can be inferred:

$$\begin{aligned} T^i &= L(S_i) + b^i l_1 V_1^i + r^i V_2^i, \\ D^i &= L(S_i) - b^i l_2 V_1^i + r^i V_2^i, \end{aligned}$$

where $L(S_i)$ is the location of the i sensor, b^i and r^i are the (manually measured) bone length connecting D and T and its thickness respectively. l_1 and l_2 are length ratios and sum to 1.

To infer the last joint position, PIP (P), the following conditions are used to derive a unique solution: (1) T, D, M are given, (2) $\|P - D\|$ and $\|P - M\|$ are fixed, (3) T, D, M and P are in the same plane, and (4) T and P are on different sides of the line $\overleftrightarrow{MD}$.

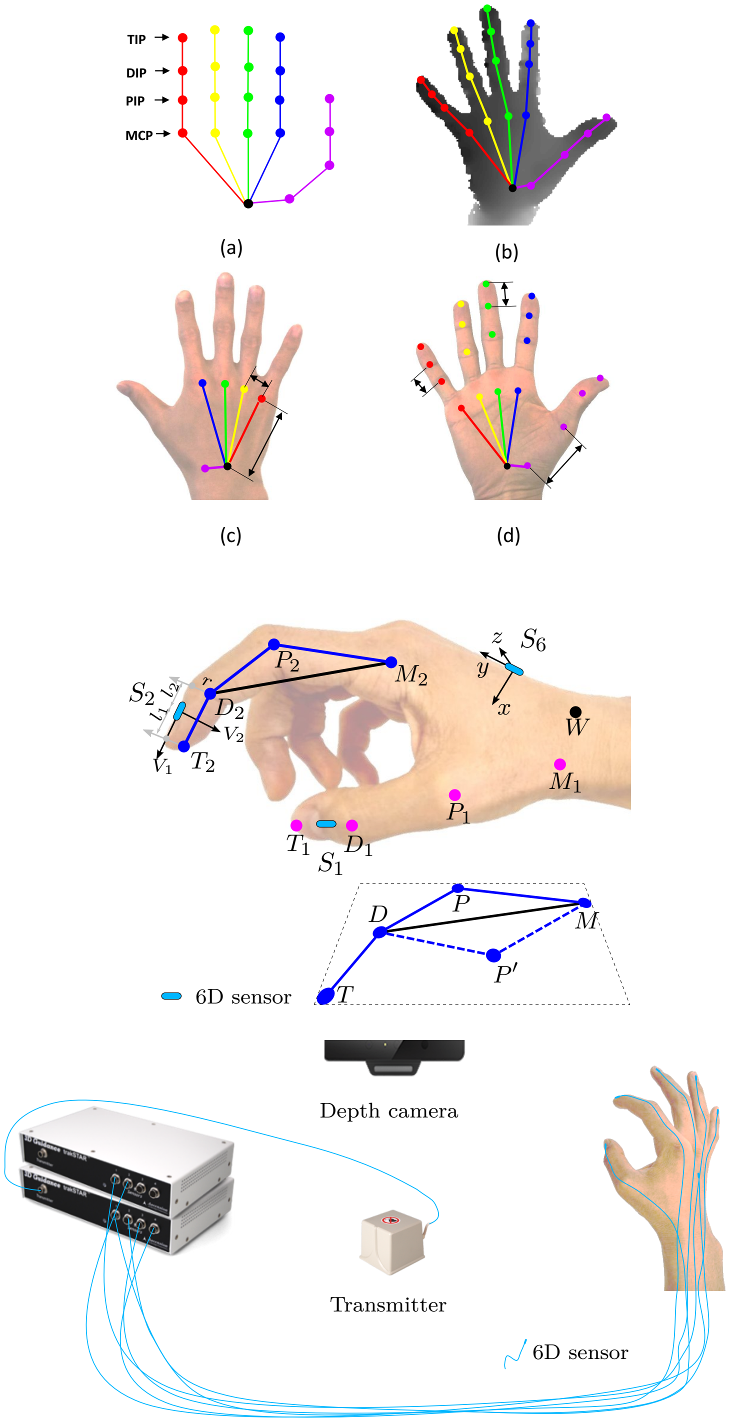

Figure A.1: Top: (a-b) 21-joint hand model used for our 3D hand pose annotations, (c-d) Physical constraints. Middle: 3D hand pose is inferred using six 6D magnetic sensors and inverse kinematics. Bottom: Equipment used to annotate hand pose. In our first-person setting we mounted the depth camera and the transmitter on the user's shoulder. Figure credit: Yuan, Ye, Stenger, Jain and Kim (2017).

B

APPENDIX B: HAND POSE ESTIMATION BASELINE DETAILS

In Figure B.1 the architecture of the CNN baseline inspired in the architecture of Yuan, Ye, Stenger, Jain and Kim (2017) is shown. The input image is the cropped depth hand image projected and normalised to 96-by-96 pixels. This normalised image is subsampled two times to sizes of 48-by-48 and 24-by-24 and all three normalised images are fed into the CNN. The cost function is the mean squared distance between joint location estimates and annotated locations. We used the same implementation parameters as Yuan, Ye, Stenger, Jain and Kim (2017).

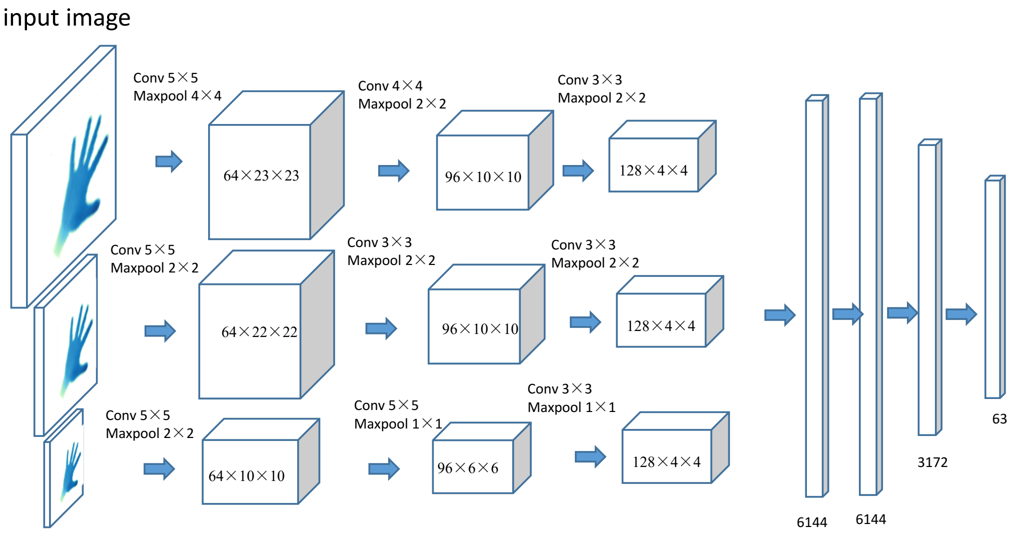

Figure B.1: Architecture of the CNN baseline for hand pose estimation used on experiments (Section 5.5.2). This architecture has been shown to achieve state-of-the-art performance in Yuan, Ye, Stenger, Jain and Kim (2017). Figure credit: Yuan, Ye, Stenger, Jain and Kim (2017).