University of Technology, Sydney

Faculty of Engineering and Information Technology
SCHOOL OF COMPUTING AND COMMUNICATIONS

PhD Thesis

Tracking and Fine-Grained Activity Recognition in Depth Videos

Prepared By: SARI AWWAD

Principal Supervisor: Prof. Massimo Piccardi

Co-Supervisor: Dr. Richard Xu

NOVEMBER, 2016

Certificate of Authorship and Originality

Title: **Tracking and Fine-Grained Activity Recognition in Depth Videos**

Author: **Sari Awwad**

Date: **November , 2016**

Degree: **PhD**

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of author

Acknowledgements

Foremost, I would like to express my sincere gratitude to my principal supervisor, Professor Massimo Piccardi for the continuous support in my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my PhD study.

My sincere thanks also go to my home country "Jordan", my sponsor "The Hashemite University" and Dr. Ahmad Otoom for giving me the greatest opportunity to complete my PhD with a greatest superviso, and my thanks also goes to Dr. Bashar and Dawacom company.

I would like to thank my mother for her continuous prayer for me, and thanks for my brothers: Osama, Samer, Saed, Sameh, and my lovely sister Joman for their continuous supports.

I thank my colleagues: Dr. Shaukat Abidi, Mrs. Fairouz Hussein, Dr. Ava Bargi and Dr. Ehsan Zare Borzeshi, they are a great group, thank you all.

To a wonderful companion, great gratitude to my wife, Lina. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past fourteen years of my life have been built. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love.

Also, I thank Lina's parents, Mr. Asad and Mrs. Somayya, for their unending encouragement and support.

To the lovely kids, my daughters Farah and Aseel, and my son Osama who have decorated my life and made it full of happiness and joy.

Finally, this thesis is dedicated to the soul of my father, may Allah forgive him and grant him his highest paradise (Ameen).

Sari Awwad.

November, 2016, Sydney

Abstract

Tracking and activity recognition in video are arguably two of the most active topics within the field of computer vision and pattern recognition. Historically, tracking and activity recognition have been performed over conventional video such as color or grey-level frames, either of which contains significant clues for the identification of targets. While this is often a desirable feature within the context of video surveillance, the use of video for activity recognition or for tracking in privacy-sensitive environments such as hospitals and care facilities is often perceived as intrusive.

For this reason, this PhD research has focused on providing tracking and activity recognition solely from *depth* videos which offer a naturally privacy-preserving visual representation of the scene at hand. Depth videos can nowadays be acquired with inexpensive and highly-available commercial sensors such as Microsoft Kinect and Asus Xtion. The two main contributions of this research have been the design of a specialised tracking algorithm for tracking in depth data, and a fine-grained activity recognition approach for recognising activities in depth video. The proposed tracker is an extension of the popular Struck algorithm, an approach that leverages a structural support vector machine (SVM) for tracking. The main contributions of the proposed tracker include a dedicated depth feature based on local depth patterns, a heuristic for handling view occlusions in depth frames, and a technique for keeping the number of support vectors within a given budget, so as to limit computational costs. Conversely, the proposed fine-grained activity recognition approach leverages multi-scale depth measurements and a Fisher-consistent multi-class SVM. In addition to the novel approaches for tracking and activity recognition, in this thesis we have canvassed and developed a practical computer vision application for the detection of hand hygiene at a hospital. This application was developed in collaboration with clinical researchers from the Intensive Care Unit of Sydney's Royal Prince Alfred Hospital. Experiments presented through the thesis confirm that the proposed approaches are effective, and either outperform the state of the art or significantly

reduce the need for sensor instrumentation. The outcomes of the hand-hygiene detection were also positively received and assessed by the clinical research unit.

Contents

List of Figures

List of Tables

Abbreviations

DP	**D**epth **P**atterns
EM	**E**xpectation **M**aximization
FV	**F**eature **V**ector
HAI	**H**ealthcare **A**ssociated **I**nfections
KPF	**K**alman **P**article **F**ilter
LDP	**L**ocal **D**epth **P**atterns
LDPT	**L**ocal **D**epth **P**atterns for **T**racking
MEI	**M**otion **E**nergy **I**mages
MHT	**M**ultilple **H**ypothesis **T**racking
PCA	**P**rincipal **C**omponent **A**nalysis
PTB	**P**rinceton **T**racking **B**enchmark
RFID	**R**adio **F**requency **I**dentification **D**evice
ROI	**R**egion **O**f **I**nterest
RPAH	**R**oyal **P**rince **A**lfred **H**ospital
SMO	**S**equential **M**inimal **O**ptimization
SVM	**S**upport **V**ector **M**achine
WHO	**W**orld **H**ealth **O**rganisation

Chapter 1

Introduction

1.1 Background and Motivation

In computer vision, video tracking and activity recognition aim to extract detailed trajectory and activity information about the individuals in a scene. These tasks are often performed over streams of RGB or grey-level video coming from conventional video cameras. A natural by-product of these tasks is visual evidence of the tracked people, often permitting to identify them fully (i.e, to know who they are). While this may be desirable for video surveillance aims, it may be unacceptable in applications where privacy is paramount, such as within the context of patient monitoring in hospitals. While it is, in principle, possible to apply post-processing to obfuscate faces, the availability of the appearance of people poses a latent threat to privacy in the first place.

For this reason, this PhD research focuses on the use of *depth* videos that only contain the distance, or depth, of objects from the cameras focal centre. Recent advances in camera technology have led to the development of inexpensive video cameras that can acquire depth videos alongside common RGB videos. Examples

are Microsoft Kinect, Asus Xtion and Stereolabs ZED. Microsoft Kinect 1, for instance, obtains the depth information by projecting dots in the near infrared spectrum onto the scene and triangulating their depth (Jana, 2012). Since these cameras contain their own projectors, depth images can also be acquired in very low light conditions or even in the absence of light. Such a technology represents a milestone for computer vision, as it allows for much more accurate volumetric scene reconstruction, object tracking, and disambiguation of occlusions. From our point of view, this technology is conveniently privacy-preserving since it is hard or often impossible to identify the viewed people in depth videos. To illustrate the actual lack of identity clues, Figure 1.1 shows an example of a person in a depth frame (distances are rendered with the use of pseudo-colours).

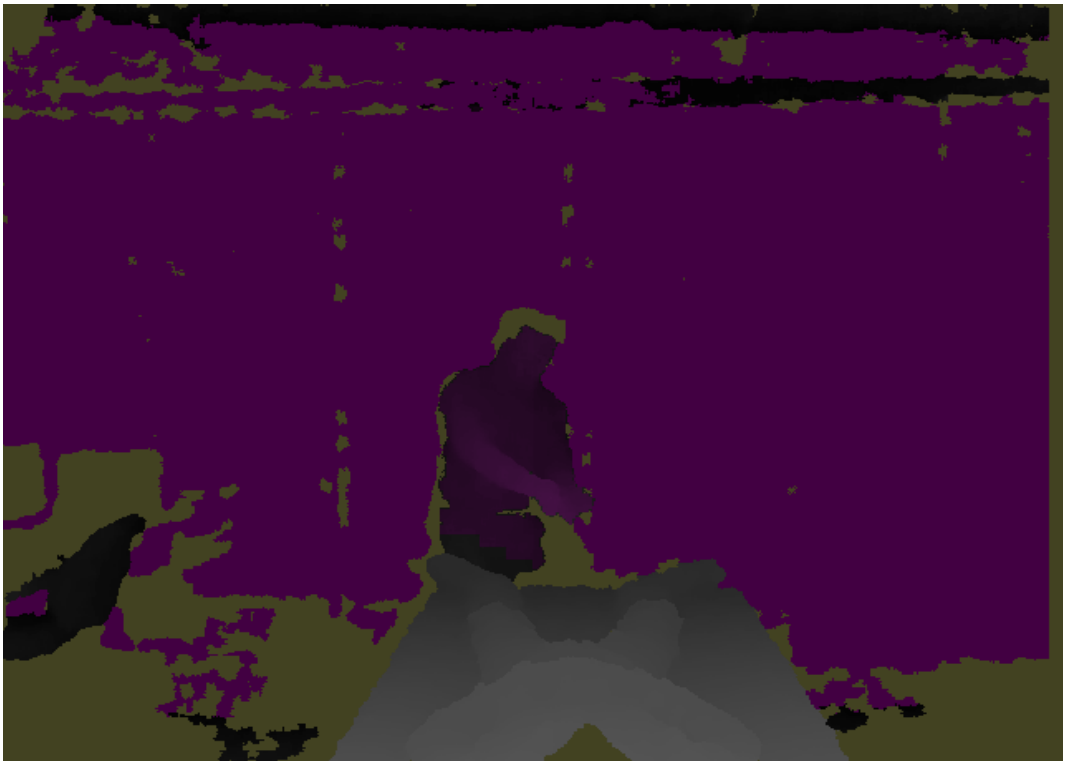


FIGURE 1.1: Depth frame example from a simulated hospital scenario.

In computer vision, "tracking" refers to the task of extracting accurate trajectories of single, moving targets in a video and it goes back a long way. Perhaps the most practiced approach nowadays is *tracking-by-detection* (Avidan, 2007, Babenko et al., 2009, Breitenstein et al., 2011, Grabner et al., 2008, Tran and Davis, 2007) where the main idea is to frame tracking as a target classification problem, while learning the classifier in an online and unsupervised manner as the targets appearance evolves. In this category, the Struck tracker from Hare *et al*. (Hare et al., 2011) has recently attracted much attention since it leverages the efficient, discriminative framework of structural SVM in the design of the classifier with a remarkable accuracy reported in a number of evaluations (Li et al., 2013, Smeulders et al., 2014, Wu et al., 2013). For this reason, we use it as the baseline tracker in this work.

The use of depth video for tracking in addition to RGB video has become increasingly popular to disambiguate occlusions and overcome illumination artifacts (Basso et al., 2013, Munaro et al., 2012, Song and Xiao, 2013). However, the possibility to track and monitor actions solely using depth videos has been largely unexplored to date. The challenge posed by depth tracking is major, as conventional trackers rely on the targets' appearance and texture to provide correct data association. In order to address this, this PhD investigates tracking of general targets based on depth data alone. Our main contributions in this area are a dedicated depth feature based on local depth patterns and a procedure for handling target occlusions in depth frames.

A major challenge in tracking-by-detection is the effective update of the target's classifier. For instance, the Struck tracker applies a number of heuristic rules to update the set of support vectors that define the detector. Such a set must respect a *budget*, i.e., an upper bound on the number of vectors, to ensure real-time performance. In an initial evaluation of Struck on depth videos, we noticed a rapid proliferation of support vectors, possibly due to the typical distributions and range of depth values. For this reason, in this PhD research, we propose to curb the number of support vectors using techniques inspired by prototype selection approaches (Riesen and

Bunke, 2010). These approaches have proven effective and flexible in many other domains and, in this study, we show how they lead to improvements in tracking accuracy.

Another line of research investigated in this thesis is the recognition of fine-grained activities in depth videos. Fine-grained activities differ from common activities in that their scale is typically smaller and their differences, more subtle, and therefore accurate recognition is more elusive. Existing approaches to fine-grained activity recognition typically leverage combinations of RFID tags, gyroscopes, accelerometers and other sensors placed on the objects, the environment and the body of the actors. While these approaches have achieved remarkable recognition accuracy, the required instrumentation is often cumbersome and makes it less likely that they will find any practical application. For this reason, in this research we have sought to investigate whether it is possible to achieve a comparable recognition accuracy by using only the depth video provided by a single, fixed-mounted depth camera. Experiments carried out over a kitchen activity dataset have confirmed our intuition and proved that the approach we propose is viable.

As a last contribution of this thesis, we have implemented a computer vision application for the detection of hand hygiene in hospital rooms. This research has been conducted in collaboration with A/Prof David Gattas and Dr. Sanjay Tarvade from the ICU of the Royal Prince Alfred Hospital in Sydney. Detection of hand hygiene in ICUs is extremely important for the prevention of infections and spreading of diseases in dehabilitated patients. The World Health Organization has established a protocol to organize hand hygiene in "moments" (before touching a patient, before an aseptic task etc) and dedicated control personnel routinely carry out monitoring of hand hygiene in hospitals. However, this type of monitoring can only be performed in limited samples and is likely to introduce an "observer effect" (the monitored parties may comply more than they would if unobserved, causing the collected samples to result in an over-estimate of the actual compliance rate). As

such, a technological solution able to monitor all actions and in a less intrusive way is highly desirable. For this reason, in the collaboration with the RPAH we have implemented a prototype for the detection of the first moment of hand hygiene using only a Microsoft Kinect camera placed behind a patient's bed. The detection approach is original and uses predominantly the depth data, with some close-ups on the clinician's hands that do not compromise the privacy-preserving nature of this approach.

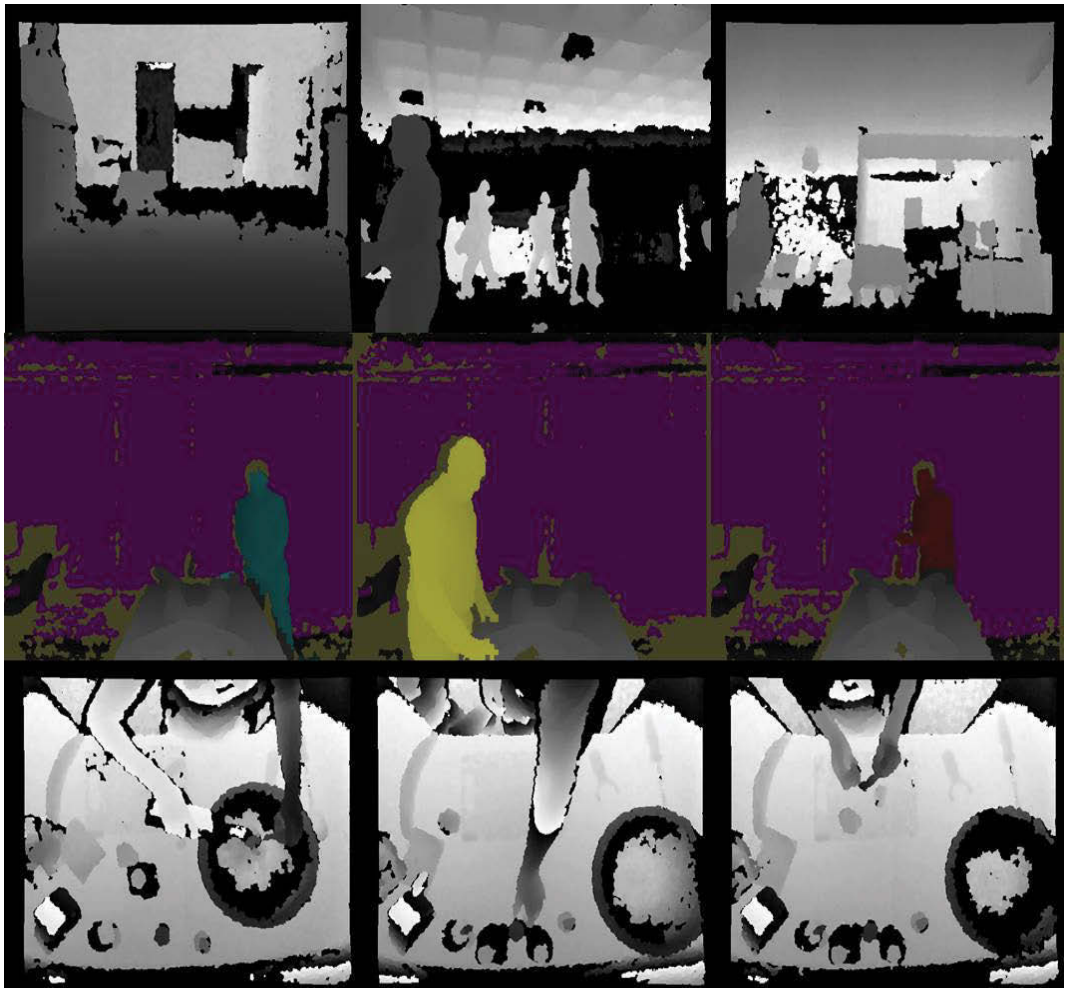


FIGURE 1.2: Depth frame examples from a simulated hospital scenario, the Princeton Tracking Benchmark, and "50 salads" datasets.

The experiments carried out for this thesis have involved three datasets: the Princeton Tracking Benchmark (PTB) dataset (Song and Xiao, 2013)] for tracking; the "50 Salads" dataset for fine-grained activity recognition (Stein and McKenna, 2013); and a simulated hospital environment dataset collected by our research group. The first dataset consists of 95 videos varying in target type (humans, animals and non-deformable objects), scene type, and presence of occlusion and bounding box distribution. The second dataset consists of 50 videos of a kitchen environment with a variety of small actions and objects related to a salad preparation. The third datasets consists of 26 videos where one or two actors simulate a visit to a patient lying on a hospital bed. Figure 1.2 displays samples of depth frames from these datasets.

1.2 Research Questions and Main Contributions

Our research aims to address the following questions:

1. Is it possible to reliably track a target based solely on depth information, or would the absence of appearance features cause tracking to fail significantly?

2. Is it possible to accurately perform fine-grained activity recognition by using only depth information, without the utilization of typical additional sensors such as gyroscopes, accelerometers and RFID tags?

3. Is it possible to perform hand-hygiene detection at a hospital in a way that is both privacy-preserving and non-intrusive on clinical operations?

All these questions have been, to an extent, answered in the affirmative. Relevant to the first question, we have developed a tracker that effectively tracks human targets

using only depth data. The main original contributions have been: i) a dedicated depth feature based on local depth patterns; ii) a heuristic for handling view occlusions in depth frames; and iii) a technique for keeping the number of the support vectors within a given ”budget” so as to limit computational costs. Relevant to the second question, we have developed a novel approach for fine-grained activity recognition that only uses local depth patterns and no additional instrumentation. Relevant to the third question, we have developed a system that detects whether a person who enters a hospital room does or does not perform hand hygiene before touching a patient (the so-called “first moment” of hand hygiene). This system leverages a combination of depth features and skin detection performed in regions where hands (not faces) are expected to be located. It therefore meets both expectations of non-intrusiveness and privacy-preservation.

This research has also led to five papers, some of which have already been published and others accepted. One more publication is under preparation. This is the complete list:

- Awwad, S., Hussein, F. and Piccardi, M., “Local Depth Patterns for Tracking in Depth Videos,” in Proceedings of the 23rd ACM international Conference on Multimedia, pp. 1115-1118, ACM, 2015.
- Hussein, F., Awwad, S., and Piccardi, M.,“Joint action recognition and summarization by sub-modular inference,” in Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2697-2701. IEEE, 2016.
- Awwad, S., and Piccardi, M., “Prototype-based budget maintenance for tracking in depth videos,” accepted (07/10/2016) in Multimedia Tools And Applications Journal.

- Awwad, S., and Piccardi, M., "Local Depth Patterns for Fine-Grained Activity Recognition in Depth Videos," accepted in (14/10/2016) Image and Vision Computing New Zealand Conference,

- Awwad, S., and Piccardi, M., Gattas, D., Tarvade, S., "Automated, Non-Intrusive Hand Hygiene Detection Using a Depth Camera," to be submitted to journal BMJ Quality and Safety.

1.3 Significance

Herewith, we wish to restate the significance of the research that we have carried out, both from a scientific and an impact perspective. The main aim of this study has been to develop methodologies and systems that would solve the problem of tracking and monitoring actions in specific environments where privacy is paramount. We believe that this research is both innovative and can have a significant impact on industrial practice. The proposed methodologies offer new approaches to solving issues previously faced when monitoring actions and tracking in depth videos. Furthermore, the results of the experiments may prove useful for other researchers and provide the foundation for future research.

The conduct of this research has led to the development of three working prototypes: 1) a tracking system; 2) a fine-grained activity recognition system; and 3) a hand-hygiene detection system, all operating on depth video. Each of these systems could potentially be extended into commercial software for the benefit of companies and end users in real life. Furthermore, these systems could be deployed to improve processes of performance monitoring and quality assurance.

Traditionally, tracking and activity recognition have been performed on colour or grey-level video; as a downside, however, they also allow the identification of the

targets in most cases. Even if post-processing can be applied to obfuscate personal traits, the collection of appearance data in the first place would pose an intrinsic, latent risk for privacy. In order to ensure anonymity, we have decided to only use depth videos, since they naturally disguise the identity of the targets. This extends the applicability of this technology to areas of application where privacy is vital such as patient monitoring in hospitals and care facilities. The outcome of this research has the potential to advance the knowledge base of this discipline with a comprehensive methodology to reliably monitor, detect, and recognize activities and events such as the detection of hand hygiene by clinical personnel.

The possibility of performing tracking using only depth videos has received limited attention to date. The challenges posed by the use of depth data alone for tracking are major since conventional trackers rely on the targets appearance and texture to provide correct data association. While it is possible to successfully fit and track skeletal models on depth data (see, for all, Shotton *et al.* (Shotton et al., 2011)), skeletal tracking is mostly designed for interaction with co-operative human users and is prone to failure in the presence of severe view occlusions. For this reasons, in this research we have deliberately excluded the use of skeletal models and focussed on the tracking of human targets in terms of overall bounding boxes. We believe that this makes our approach more general and potentially applicable to a wider variety of cases.

1.4 Thesis Structure

This doctoral thesis consists of six chapters (please see Figure1.3):

1. $Chapter$ 1 presents an overview of this research, including research issues, research motivations, research questions and contributions, and the research significance.

2. $Chapter$ 2 reviews the related research areas, and especially addresses the research background with regard to tracking algorithms, online learning for tracking, feature descriptors, feature types, depth features, event detection in hospital environments and fine-grained activity recognition techniques. Additionally, this chapter provides a basic understanding of the theory behind the support vector machines (SVMs), a describing the types of classifiers, (Binary, Multi class, Structural) and explain some related work.

3. $Chapter$ 3 presents the proposed depth tracker in detail, including an overview of the original Struck tracker, the proposed local depth features for tracking, our occlusion handling approach, and the proposed technique for support vector removal, and reviews the experimental results.

4. $Chapter$ 4 presents a novel approach for fine-grained action recognition based on the use of a single depth camera, including the local depth patterns, image gridding and fine-grained activity classification, and reviews the experimental results.

5. $Chapter$ 5 illustrates the system designed to detect compliance with Moment 1 of Hand Hygiene from a Kinect camera, including the pouring of handrub liquid from a dispenser and the vigorous rubbing of hands prior to touching a patient, and reviews the experimental results.

6. $Chapter$ 6 summarises the whole thesis and describes possible, future research.

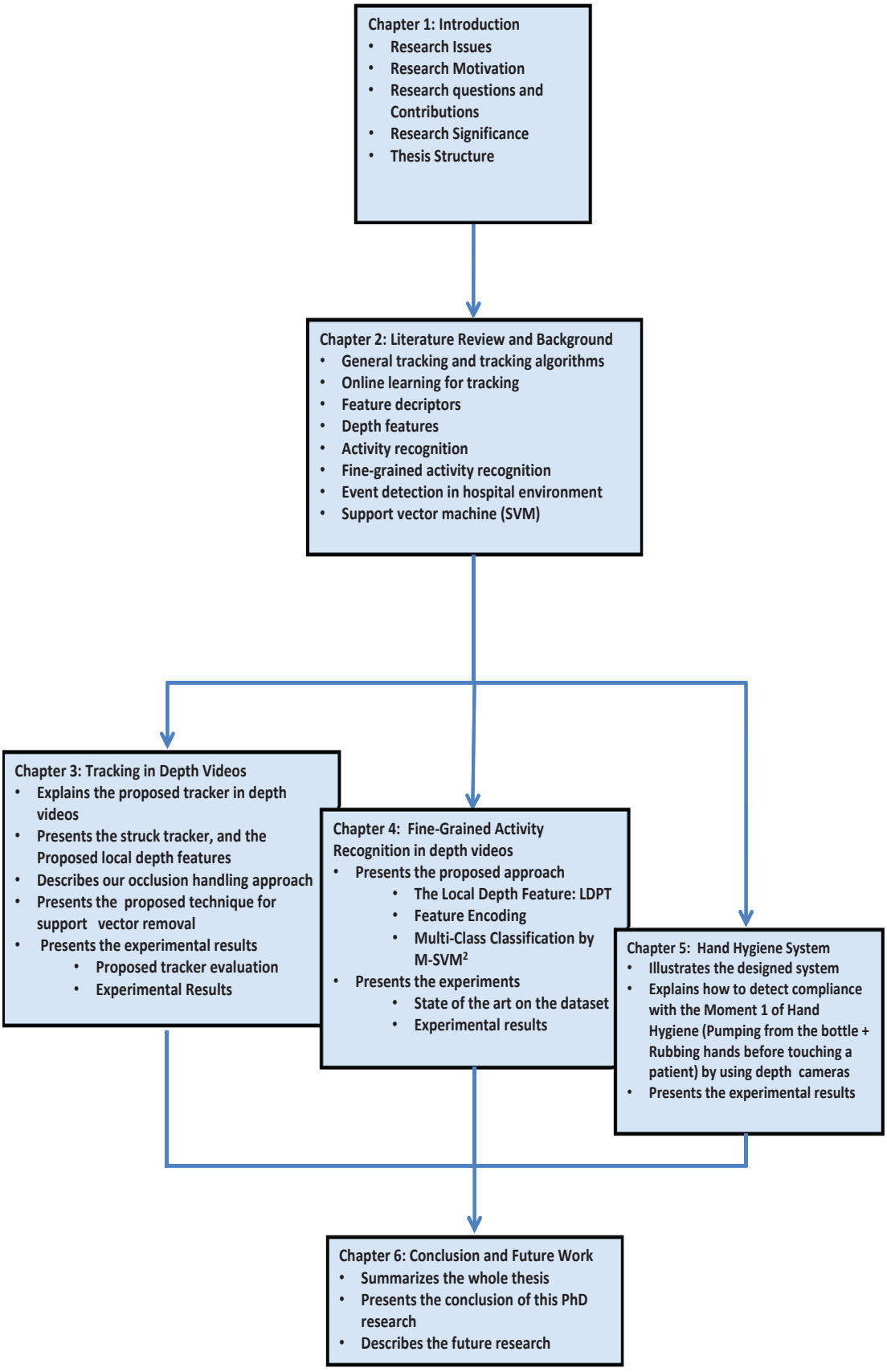


FIGURE 1.3: Thesis Structure

Chapter 2

Literature Review and Background

This chapter provides background information and a literature review relating to tracking and fine-grained activity recognition in depth videos.

This chapter is comprised of four main sections. The first section covers object tracking, which allows for an exploration of tracking algorithms and online learning for tracking purposes. Following this, information will be provided on features, which includes feature descriptors, types, browse examples for each type, and depth features. Thirdly, event detection will be discussed. This section explores event detection approaches, event detection in hospital environments, and fine-grained activity recognition techniques. Finally, SVM background will be provided, presenting a basic summary of the theory behind the topic, a describing the types of classifiers, (Binary, Multi class, Structural) and explain some related work.

2.1 Object Tracking

Object tracking refers to the task of extracting accurate trajectories of single, moving targets in a video. Usually, this process uses features including line segments, geometric points, and appearance.

Tracking of an object is an essential element in computer vision. Tracking has many useful real life applications, such as surveillance, navigation systems, event detection and action recognition.

However, tracking can become a complex process with the detection of an occlusion which blocks an object, completely or in part. These objects may be found in the background or may consist of other moving objects within the frames (Yilmaz et al., 2006).

The following are the main factors that influence the complexity of the tracking process:

- **The Complexity of the background scene**; an indoor background scene is much easier to process than an outdoor background scene;
- **The number of tracked objects**, which has the potential to increase the complexity while decreasing the accuracy;
- **The type of tracked object**; a rigid object, such as a vehicle is relatively easier to track than animals and humans. People are articulated, vary significantly in shape and their motion is non-rigid and unpredictable;
- **Light or short occlusions** normally present no difficulty to most tracking algorithms. However, substantial occlusions of several objects are almost unsolvable for any currently available tracking algorithms.

Also, the selection of certain features to represent the object is critical to the tracking process. Feature type plays a significant role in the detection of the target in each frame. It is essential to efficient tracking and activity recognition. Tracking models, an environmental dataset, and a specific target constitute the major factors in the selection of features. There are many different types of features used in tracking and activity recognition (Wang et al., 2005).

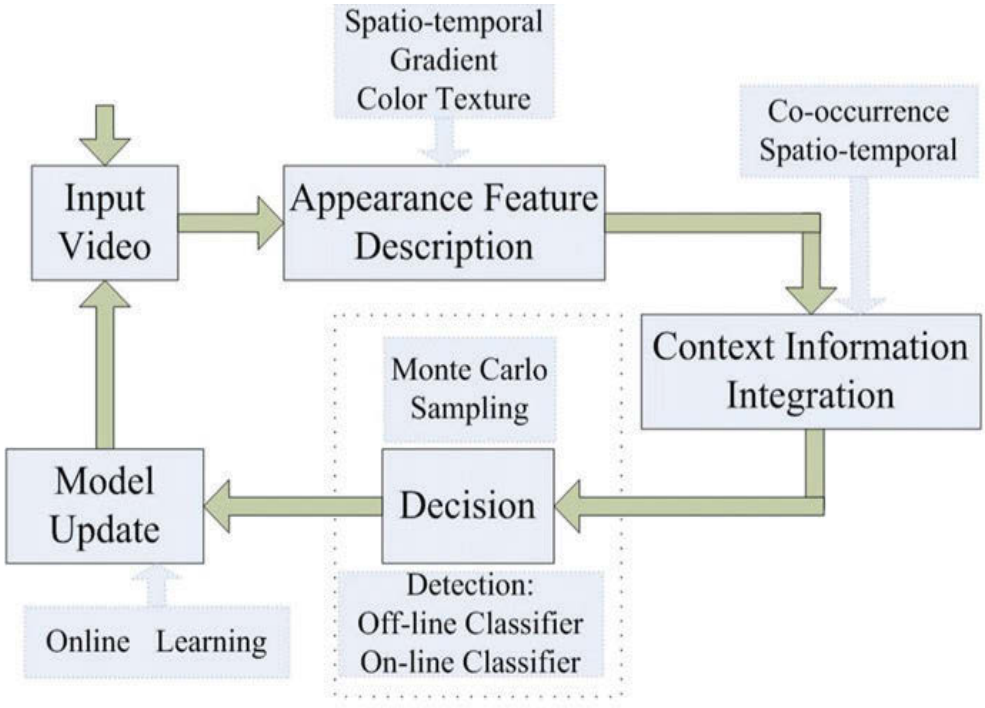


FIGURE 2.1: Tracking Stages in Computer Vision (Yang et al., 2011)

There is a great deal of tracking strategies covered in the available literature to consider. For instance, Yilmaz et al. (2006) has categorized object tracking using point tracking, kernel tracking, and silhouette tracking. It should be noted that such categories cannot fully and clearly separate all the proposed tracking methods, as numerous methods overlap between categories.

As seen in Figure 2.1, tracking in computer vision can be divided into different stages. Each stage comprises of various algorithms and approaches such as feature

descriptors, online learning methods, context information integration and classification (Yang et al., 2011).

In this thesis, an object tracking part method is distinguished based on four criteria: the selected features, the tracking model, occlusion technique and budget maintenance that controls support vectors, when its number exceeds a specific limit. Budget concept will be explained in chapter 3.

2.1.1 Tracking Algorithms

Various common tracking methods have been reviewed and classified in the following section. Generally, object tracking is a challenging endeavour. Difficulties in tracking objects can arise due to abrupt object motion, the changing patterns of both the object shape and the scene appearance, non-rigid object structures, object-to-object and object-to-scene occlusions and camera motion.

The performance of tracking is commonly conducted in the context of higher-level applications that require the location and/or shape of the object in every frame. The subsequent subheadings allow the exploration of the more common methods.

2.1.1.1 Detection-Based Tracking and Kernels

Object detection is one of the most influential factors within object tracking. Many approaches can be used for object detection, for instance, normalized cross-correlation (NCC). This approach performs target representation detection, utilizing a template specifically designed for targets and built into previous frames. Briechle and Hanebeck (2001) use this type of detection by using NCC in an effort to create a target template, then used for target detection through matching.

Other approaches can be used for object detection in tracking, such as 'appearance matching' and 'matching with histograms'. In the first case, the target region is represented through template-intensities, which is then associated with the Kalman filter (explained in subsection 2.1.1.3). In the second case, RGB-colour histogram is used to represent the target, rather than pixel spatial information. These approaches are utilised by Nguyen and Smeulders (2004) and Comaniciu et al. (2000) to design their trackers.

In the aforementioned approaches, template matching and appearance matching have been popularly used in tracking due to their relative simplicity and low computational cost. Therefore, Ross et al. (2008) and Kwon et al. (2009) used matching with an extended appearance model to design their trackers. While Han and Davis (2005), McKenna et al. (1998), Pan and Hu (2007), Schweitzer et al. (2002), Singh et al. (2004) and Zivkovic and Krose (2004) used tracking through object detection using template matching and colour matching.

By contrast, other approaches depend on sparse representation and optimisation more than target representation. These approaches aim to increase target tracking through drastic object shape changes. This is achieved by using sparse optimisation over patch pairs to handle occlusions (Kwon et al., 2009, Mei et al., 2011),

Furthermore, kernels can be used for detection in tracking. A kernel, $k(x, y)$, is a mathematical function that measures the similarity between its arguments. In this case, kernels are used for the computation of the object's motion over consecutive frames. The particular spatial domain used for the detection usually has the same shape or appearance of the target object. This could include elliptical templates and their associated colour histogram.

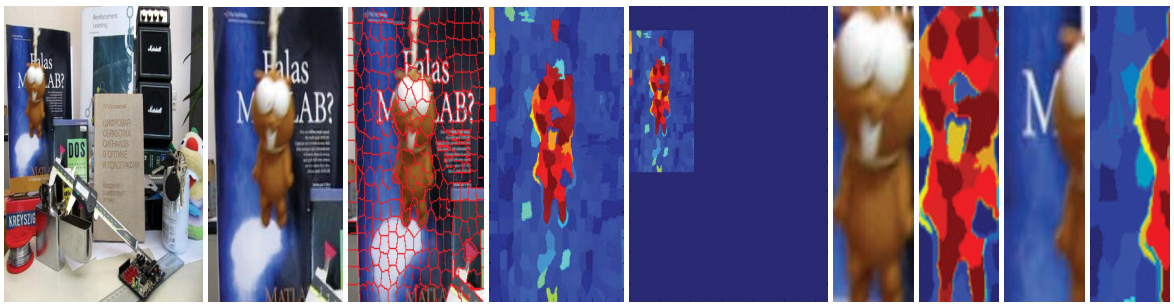
The colour histogram computation is performed through the application of a kernel that is applied to every point in the template. Moreover, the kernel motion detected between consecutive frames commonly consists of a parametric transformation such

as a rotation, translation, or general affine, or as a dense flow field that can be computed in subsequent frames.

Examples of colour histogram computation by Comaniciu et al. (2000) and Comaniciu and Ramesh (2003) proposed an approach for detection through the use of a template for the target object, using a circular spatial mask as kernel domain and a weighted colour histogram therein. This colour histogram is also created around the window for two hypothesized object locations. This process is repeated until convergence occurs. Kernel-based tracking, also conducted by Zhang et al. (2004). Their approach involves tracking translational and rotational objects. The Zhang's model allows for the application of a kernel-based spatial-colour model, enabling object representation and formulation of similar measurements.

However, many studies proposed object-tracking approaches using a kernel with classifiers to obtain the appearance of tracking objects (Li et al., 2013, Yilmaz et al., 2006). These studies consider object tracking as a continuous object detection problem.

In online and single object tracking research, Tracking-By-Detection is the recent mainstream framework popularized by Wu et al. (2013) and Smeulders et al. (2014). Existing algorithms, such as those used by Donoser and Bischof (2006), Silveira and Malis (2007), Tran and Davis (2007), Wang et al. (2011) and Hare et al. (2011) could be divided into generative approaches and discriminative approaches, (please see Figure 2.2). The generative approach does not explicitly separate the background, while the discriminative approach considers object tracking as a binary classification problem to determine whether the target object is real or not. The Struck tracker proposed by Hare (Hare et al., 2011) (please see chapter 3) is a state-of-the-art algorithm for online object tracking.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

FIGURE 2.2: Tracking by detection example; using super pixel-based discriminative appearance where (a) a new frame at time t. (b) surrounding region of the target in the last frame, i.e., at state X_t . (c) segmentation result of (b). (d) the computed confidence map of super pixels The super pixels coloured with red indicate a strong connection to the target, and those coloured with dark blue indicate a strong connection to the background. (e) the confidence map of the entire frame. (f), (g) and (h), (i) show two target candidates with high and low confidence, respectively. Model) (Wang et al., 2011)

2.1.1.2 Blob Tracking

Blob tracking is an approach which can identify and trace the movements of objects within images. A blob represents an object by a connected set of pixels.

Therefore, the most common methods for blob tracking relate to variations of image regions that correspond to moving objects. These show that the images are not stationary and allows for motion detection regions, performed through the subtraction of the background of the current image. As a result, motion detection regions are established and the object of interest can be focused on. The goal of the motion detection is to identify a particular region that is connected to that image, which is known as a 'blob'. Once this identification occurs, it is possible to track the blob over a time sequence utilizing cross-correlation.

As proposed by Collins (2003), many different techniques may be utilized in order to track blobs, such as the use of the mean shift procedure. This particular approach

starts with the detection of the scale blob by using scale-space filters. This is followed by the procedure of the mean of change and tracking the modes in space of scale. As a result, it is evident that representation of the blob occurs through the spatial position of the mode, as well as the scale of the blob, as shown in Figure 2.3. Work completed by Khansari et al. (2007) shows that the approach for tracking a blob can be extended based on temporal tracking and un-decimated wavelet features which relates to removing noise from the image.

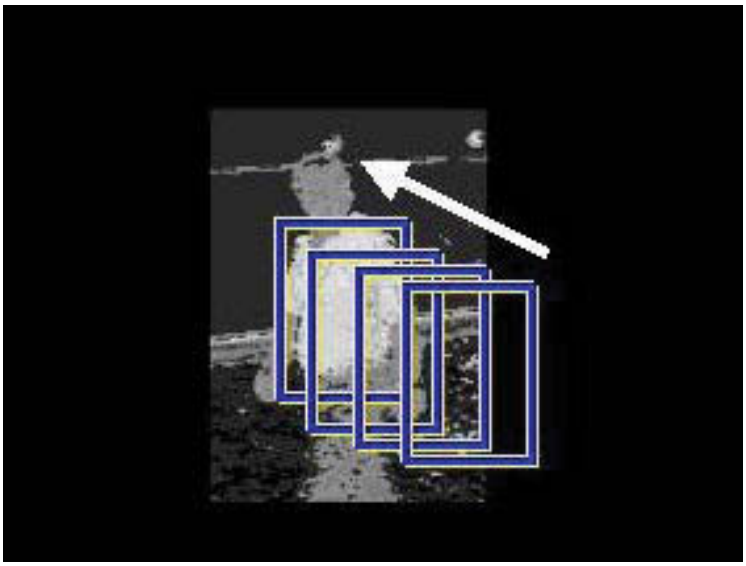


FIGURE 2.3: Blob tracking example: Two-dimensional blob tracking by applying the mean-shift algorithm to an image where pixel values represent likelihood of being on the tracked object (Collins, 2003)

Yet, this type of blob tracking has a major disadvantage; it can not handle well occlusions among objects. Furthermore, these algorithms track the blob level in order to obtain results through the use of motion detection procedures. As a result, it is not possible to obtain the objects' contours. Subsequently, blob tracking has become less popular, because new tracking algorithms have been proposed to handle occlusions among objects.

2.1.1.3 Kalman Filter

The Kalman filter, proposed by Kalman in 1960, is considered an efficient, baseline solution for estimating a trajectory from noisy measurements based on linear filtering. It consists of a set of mathematical equations which work recursively to estimate the state of a process from a series of noisy measurements by minimizing the mean square error. With the Kalman filter, the current state estimation process needs only the estimated state from the previous time step and the current measurement (Kalman, 1960).

This recursive nature makes the Kalman filter a very powerful technique for controlling noisy systems. Therefore, it has proved popular for a variety of applications:

- Object tracking (e.g., missiles, body parts)
- Fitting Bezier patches to point data (noisy, moving, ...)
- Navigation
- Many computer vision applications:
  - Stabilizing depth measurements
  - Feature tracking
  - Cluster tracking
  - Fusing data from radar, laser scanner and velocity measurements.

Also, the Kalman filter can be used for various purposes within object tracking, such as motion prediction of the target object (Gao et al., 2005, Ozyildiz et al., 2002, Weng et al., 2006), or feature estimation and smoothing (Nguyen and Smeulders, 2002).

In further detail, the Kalman filter has two distinct phases, one for prediction and the other for update. The prediction stage uses the state estimate of the previous time step to predict the condition at the current state. The predicted state $\hat{X}\prime_t$ acts as a prior for the state in the update stage. The basis for this prediction is made using the measurements obtained from previous time steps.

$$\hat{X}\prime_t = A\hat{X}_{t-1} \tag{2.1}$$

$$P\prime_t = AP_{t-1}A^T + Q \tag{2.2}$$

Equations 2.1 and 2.2 are the prediction stage; $\hat{X}\prime_t$ is the prediction for the state, $P\prime_t$ is the prediction for the covariance, Q refer to the noise covariance and A is the system matrix. The A variable has a host of typical alternatives which include: constant acceleration, periodic movement and constant velocity.

The update phase linearly combines the a-priori prediction with the current measurements to refine the estimated state. The final state estimate is called aposteriori state estimate. At each time step, the process of prediction and update are repeated by using the previous a-posteriori to predict the new a-priori estimates.

$$K_t = P\prime_t H^T (HP\prime_t H^T + R)^{-1} \tag{2.3}$$

$$\hat{X}_t = \hat{X}\prime_t + K_t(Y_t - H\hat{X}\prime_t) \tag{2.4}$$

$$P_t = (I - K_t H)P\prime_t \tag{2.5}$$

The equations above are the update stage; K_t is the Kalman gain which decides how much the a posteriori estimates should be corrected by the t^{th} observation. Y_t is the measurement, H is the measurement matrix, R is the measurement noise covariance, $Y_t - H\hat{X}\prime_t$ is the difference between the actual and predicted observations (often called innovation or measurement residual), and P_t is the a-posteriori estimated covariance (Kalman, 1960).

In many cases, the linear assumption of the Kalman filter proves restrictive. The *extended* Kalman filter is a non-linear version that only linearises arrund the current mean and covariance. However, if the state and measurement models are highly non-linear, even the extended Kalman filter cannot give a reasonable performance. This failure is due to the potential for error in the true posterior mean and covariance, which stems from having the mean and covariance propagated through linearization of the underlying non-linearity. Julier and Uhlmann (1997) proposed the use of the *unscented* Kalman filter targeted to solve this problem. They used a deterministic sampling technique to pick a minimal set of sample points around the mean to get the true mean and covariance.

2.1.1.4 Particle Filters

In the Kalman filter, one of the basic assumptions is that the state variables follow a Gaussian distribution. However, this assumption does not hold for many applications. To relax this constraint, particle filters were introduced by Kitagawa (1987) to generalise the traditional Kalman filter model.

Particle filters are sophisticated models in which a sampling approach is used, with a set of random particles (also known as samples or individuals) representing the filtering distribution. In this model, the weight of each sample is used to define the observation frequency.

Particle filters are based on repeated sampling and compute empirical averages using the samples (sequential Monte Carlo). Gordon et al. (1993) conceptualized this process. The first step in this methodology is to take N samples from the distribution in question, by generating a random number (r) between 0 and 1 and choosing the smallest j in such a way that the collective weight is less than r. This can be represented as:

$$x_t^{(l)} \sim \pi(x_t | x_{0:t-1}^{(l)}, y_{0:t}) \tag{2.6}$$

$$\hat{x}_t^{(l)} = x_{t-1}^{j} \tag{2.7}$$

For each of the samples, $\hat{x}_t^{(l)}$, another sample $x_t^{(l)}$ should be created by using a zero mean Gaussian error $w_t^{(l)}$ and a function f (non-negative). Mathematically, it is expressed as:

$$x_t^{(l)} = f(\hat{x}_t^{(l)}, w_t^{(l)}) \tag{2.8}$$

For each sample $x_t^{(l)}$ that is newly created, the $w_t^{(l)}$, which is the corresponding weight, will be calculated by using the y_t measurements. Now, the current state will be determined by the new samples.

$$w_t^{(l)} = \pi(y_t | x_t = x_t^{(l)}) \tag{2.9}$$

In the process of object tracking, particle filters have widely been in use. A multiple target tracking method was proposed by Hue et al. (2002). Their approach worked on the basis of particle filters being used to estimate the number of state processes when realizations of different kinds of observation processes were known. Furthermore, there have been approaches made by Isard and MacCormick (2001) to track

objects in 3D environments. The factors included in the state vector of the particle filter were the 3D shape, position, size and velocity of each and every object that was present in the scene. This unique approach provided for the inclusion and/or removal of objects from the scene by making changes to the prediction and correction scheme of the applied particle filtering, using this ability to increase or decrease the size of the state vector. In the cases where the state variable does not follow the Gaussian distribution, the use of particle filters have proven to be extremely effective. Thanks to the fact that they propagate multiple hypotheses, particle filters have reportedly been resilient to occlusions.

However, the downside of using particle filters is that the computational cost is very high since the algorithm considers all the particles at the same time. This problem was addressed by Li et al. (2003) when they proposed a Kalman Particle Filter (KPF). This was described as an effective combination of the Kalman filter and a particle filter. Using the techniques of both filters, the KPF works by steering a set of particles towards the areas where there is a high likelihood of a positive match. In this way, the total number of particles that is required for tracking is reduced by a large extent. The KPF was also modified for colour-based tracking by Satoh et al. (2004). Additional improvements on particle filters were achieved by Odobez et al. (2006) and many other authors.

In the field of visual object tracking, many applications of particle filters have been proposed. For example, a particle filter has been used to track objects in the presence of occlusions in a method by Zhou et al. (2004). In their method, they used a particle filter to achieve robust visual tracking and recognition. As a result, the number of particles and the velocity statistics are changed over time (please see Figure 2.4 for an example). The filter was adapted along the linear state transition equation, and the features used for tracking were:

- Observations comprised of 2D images in grey levels

- Pixel-based model for each object
- Mixture of Gaussians at each pixel

FIGURE 2.4: Particle tracking example: by using appearance-adaptive models (top row is the adaptive velocity model and the bottom row is the zero-velocity model) (Zhou et al., 2004)

Another line of tracking approaches is based on the EM (expectation/maximization) algorithm. The main idea here is simple: this algorithm is typically used to iteratively find the "optimal" parameters of a distribution from a data set. However, if the data changes, it can also be used to track the evolution of the parameters. Thus, EM is applied with every new observation to keep the model up to date. The detection and handling of occlusions with this method can be described as: whenever a pixel in the image patch is too distant from the mixture-of-Gaussian model, the pixel is labelled as an outlier. An occlusion is declared when the number of outliers in the image patch exceeds an assigned threshold, and the velocity and appearance model can not be reliably updated. This methodology can efficiently prevent the incorrect updating of the model in the presence of moderate occlusions; however, in the case of sudden illumination changes, false detections of occlusions will occur Zhou et al. (2004).

For the measurements, all the above tracking approaches have used a variety of sensors. Since the inception of depth cameras, tracking approaches have increasingly relied on the availability of both *appearance* and *depth* data. To the best of our knowledge, it appears that this thesis could be the first to attempt tracking from depth data *as the only modality*.

2.2 Feature Descriptors

This section provides a review of the *feature descriptors*. Feature descriptors are characteristic measurements from an area within a frame or a video. Features provide all the necessary information to perform certain visual tasks, including identification, classification, and tracking. An example of this would be the evaluation of the curvature of an object to classify it as a human or a vehicle.

In an ideal situation, feature descriptors should allow for the generalization of small variations in the appearance of an object, its background, the viewpoint contained within the image, or the execution of an action. These descriptors should be sufficiently rich in order to enable robust tracking and or classification of the activity of that object. Certain feature descriptors are concerned with the spatio-temporal dimension of the object, while others are concerned only with the spatial dimension. In this case, temporal vectors should be aggregated in the classification step (Herath et al., 2016).

Furthermore, there are two main categories of feature descriptors, global and local descriptors. The first addresses and encodes the visual observation as a whole, while the second concentrates on those descriptors specific to the localized setting (Jegou et al., 2012, Wang et al., 2013a,b). The following two subsections explore global and local descriptors.

2.2.1 Global descriptors

A top-down approach allows for the identification of global descriptors. In this approach, the first step consists of the localization of an object within the image using either tracking or background subtraction.

The computation of the global descriptor identified within the specified region of interest (ROI) follows. The ROI is then encoded as a whole into a single descriptor. An example of this would be the consideration of the entire human body as a global feature descriptor within a specified image or frame.

Common global descriptors include silhouettes, edges, or optical flow present within the ROI. These descriptors are typically more sensitive than local descriptors to changes in the viewpoint, background noise and occlusions. However, when the domain allows for strong levels of control over these factors, global descriptors typically perform well (Poppe, 2010).

2.2.1.1 Grid-based global descriptors

In this approach, the ROI is divided into either a fixed spatial or a temporal grid and multiple descriptors are computed. They are typically rather robust to small variations that may include noise, partial occlusions, and even changes in viewpoint within the image or frame. Each cell in the grid serves as a descriptor for local image observation, with the matching function changed at this point from a global descriptor to a dense grid of local descriptors. The application of such grid-based descriptors, while similar to general local descriptors, discussed in greater detail in subsection 2.2.2.1, requires the prior extraction of the ROI (Poppe, 2010).

Amongst others, this approach allows extracting *local binary patterns* using either the spatial or the temporal dimension in the grid, allowing for the histogram of

non-background responses to be stored (Kellokumpu et al., 2008). Following the completion of such a task, taking this stored set of non-background responses, it is then possible to calculate histograms based on specified oriented gradients (HOG) (Dalal and Triggs, 2005). Once these calculations are completed, it is possible to compute the edges of the foreground through the application of *non-negative matrix factorization* (Thurau and Hlavác, 2008). In turn, a binary silhouette response over time within each frequency domain can be computed (Ragheb et al., 2008).

Each specific cell within the spatial grid should contain the mean frequency response for the contained spatial location (Ragheb et al., 2008). Identification of optical flow, using grid-bases descriptors, is obtained from the division of the ROI into approximate horizontal slices of the head, body and legs (please see Figure 2.5) (Danafar and Gheissari, 2007).

FIGURE 2.5: Bounding box and its partitions by dividing the ROI into horizontal slices (Danafar and Gheissari, 2007)

.

2.2.1.2 Shape-contour based global descriptor

In certain applications, the simplest features that may be used to recognise movement within a given image are reliant upon shape. Each movement tracked within the frame or image contains specific poses that are identifiable based on particular movements (Poppe, 2010). An example of this would be the detection of "running"

motion, as indicated in Figure 2.6. Observation of the angles of the knees and elbows, through the application of 3D kinematics, identify this action as running.

FIGURE 2.6: Running shape: the kinematic constrains for knee and elbows angles are used for feature detection

An adaptation of this shape contour approach is based on global descriptors (Zhang et al., 2008). It has been found to be particularly effective, when each specific log-polar bin corresponds to a histogram.

2.2.1.3 Spatio-temporal based global descriptors

One of the more prominent subsets of global features is the exploration of temporal changes. In these features, descriptors focus on the motion occurring within the frame or image, as opposed to its appearance.

The movement may be directly recognized based on a given sequence of frames, even in cases where the image is blurred. While there is no structure defined for the specific features, a person can easily recognize a human frame in a sitting position. This is regardless of the blurred nature of the image.

The motion can be identified based on a person's position, detection of changed scene, or a tracked object (Ayers and Shah, 2001, Bregler, 1997, Loula et al., 2005). Deng et al. (2010) defines the viewed action as a sequence of space-time shapes through the computation of silhouettes of a human body, as displayed in Figure 2.7.

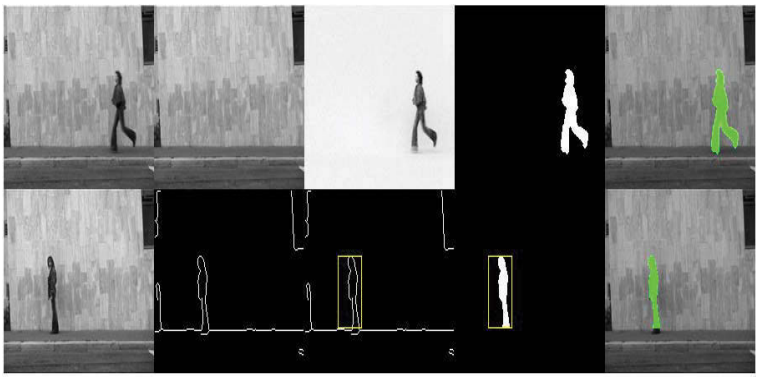


FIGURE 2.7: Movement recognition based on a sequence of space-time shapes in a video (Deng et al., 2010)

This simple, yet intuitive approach is suitable for basic motions, such as standing, sitting, walking, or running. It is however not particularly accurate for more complex motions and as a result, further methods have been explored by numerous researchers. They have used motion energy images (MEI) as a means of explaining the differences between silhouettes found in subsequent frames of video capture (Ahad et al., 2012, Skornitzke et al., 2015).

The temporal representation of silhouettes, referred to as motion history images (MHI), is another alternative. With this approach, a "history" of the motion can be obtained. This function correlates to pixel intensity based on the most recent silhouettes (Ahad et al., 2012, Skornitzke et al., 2015).

However, all approaches mentioned above are about motion or moving objects. For still images, one can resort to spatio-temporal volumetric (STV) features. These features can be applied on multiple silhouettes, stacked along a period to form a

three-dimensional STV feature. (Wang et al., 2013a, Yang and Ma, 2016). An example of this is shown in Figure 2.8.

Despite the benefits of the STV features, it does however require multiple cameras to cover all angles (Wang et al., 2013a, Yang and Ma, 2016). In order to address this issue, the silhouettes available from all viewpoints must be combined into a single three-dimensional voxel figure using motion history volumes (MHV) (Ahad et al., 2012, Wang et al., 2013b). Figure (2.8 b) displays a visual representation of MHV.

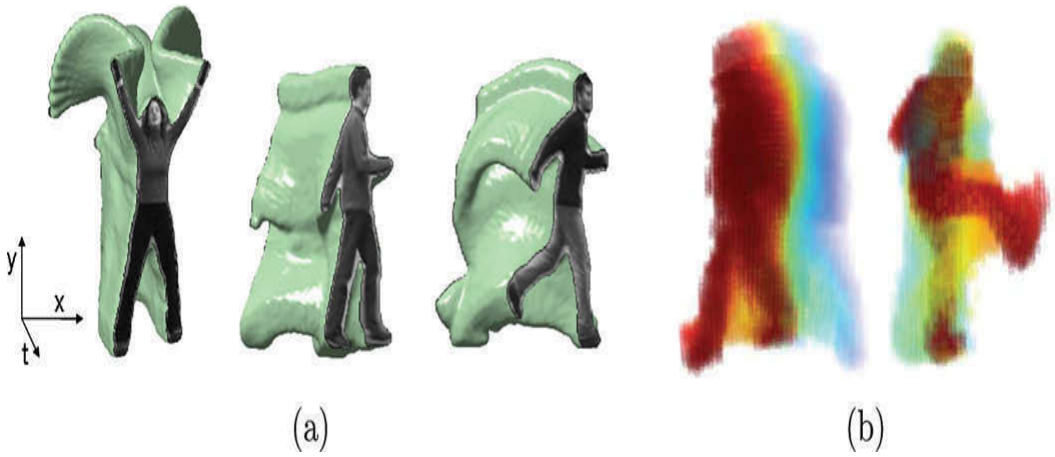


FIGURE 2.8: (a) Space-time volume of stacked silhouettes, (b) Motion history volumes. Figures reprinted from (Poppe, 2010)

2.2.2 Local descriptors

While global descriptors provide many benefits, and can aid in a more expansive identification of specific factors, there are considerable computational costs associated with their use. This is due to the initial pre-processing stages and the need for high depth background subtractions.

Further complications are presented when a high level of accuracy is necessary, not to mention the process of removing the noise from the images, both of which can

affect the overall accuracy of the recognition (Fanello et al., 2013, Hays and Efros, 2015). It is here where the use of local descriptors is relevant.

Local descriptors are image patterns that summarize a patch of an image or a video, differentiating that frame from the current, larger area. Local descriptors serve as a means of representation that are only mildly affected by background clutter, appearance changes, and occlusions (Poppe, 2010, Seidenari et al., 2014).

These local descriptors are used to define and describe observations as a collection of independent patches. They can speed up the processing time while reducing some undesirable sensitivity. This type of descriptors is typically associated with a change in image property, or in several properties of the image simultaneously. However, they are not necessarily concentrated on those changes (Fanello et al., 2013, Hays and Efros, 2015, Poppe, 2010). Figure 2.9 shows an example of local features within a still image.

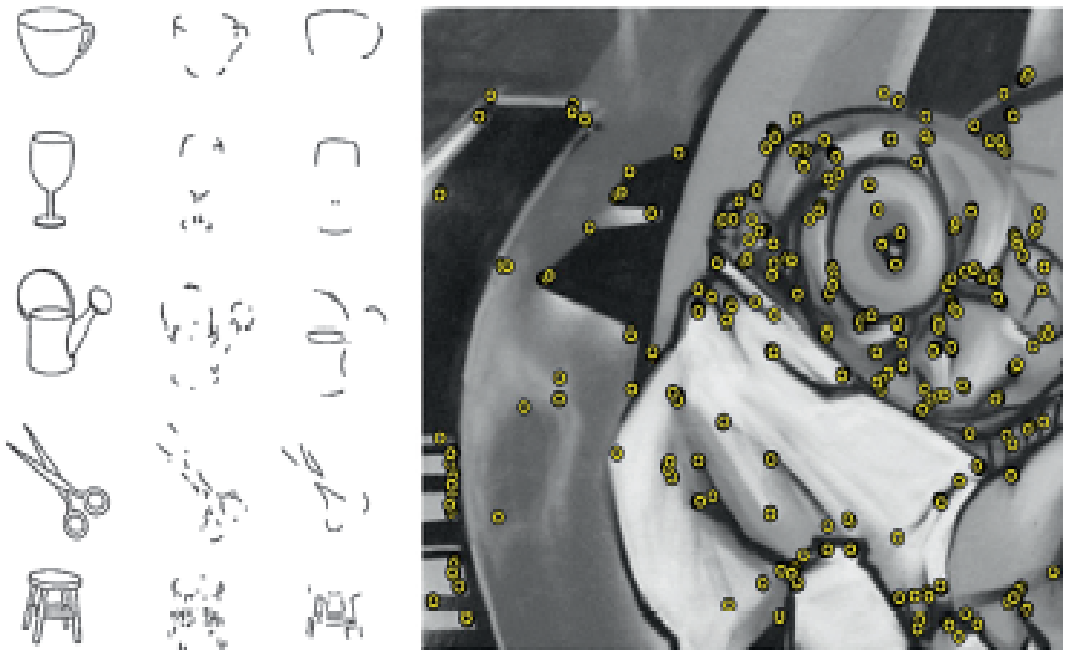


FIGURE 2.9: Examples of local descriptors within a still image (Mikolajczyk and Tuytelaars, 2015)

The calculation of local descriptors is typically conducted using a bottom-up approach, starting with the detection of spatio-temporal interest points, with local patches calculated around those points. Once such a task has been completed, the variously identified patches are combined to create feature vectors, which may then be tracked throughout the frames.

As these features do depend on a sufficient amount of relative interest points, it is necessary that the image quality is such that those points are identifiable. Otherwise, such feature vectors cannot compensate for camera movement adequately (Fanello et al., 2013, Hays and Efros, 2015, Poppe, 2010, Seidenari et al., 2014).

2.2.2.1 Grid-based local descriptors

Similarly to the approaches identified in section 2.2.1.1, it is possible to use grids to compute local spatial or temporal descriptors. These descriptors are not usually concatenated like with the global grid descriptors, but used to compute some statistics such as histograms or mixture distributions (Poppe, 2010).

The grid-based approach to local descriptors ensures that a certain degree of spatial information is maintained across the images. Within the spatial domain, there are three common approaches, described hereafter.

Within the image, the first commonly employed approach is to sample oriented rectangular patches. This approach uses rectangular cells, meaning that each image is divided into squares, and each square is a cell. Each rectangular cell has its own associated histograms that represent the distribution of the various rectangle orientations. As a result, the matching of the points of interest are ensured and identified with the initial image for action detection across multiple frames (Ikizler and Duygulu, 2009). Figure 2.10 is an example of grid-based local descriptor represented through space-time features for the action of drinking in multiple frames.

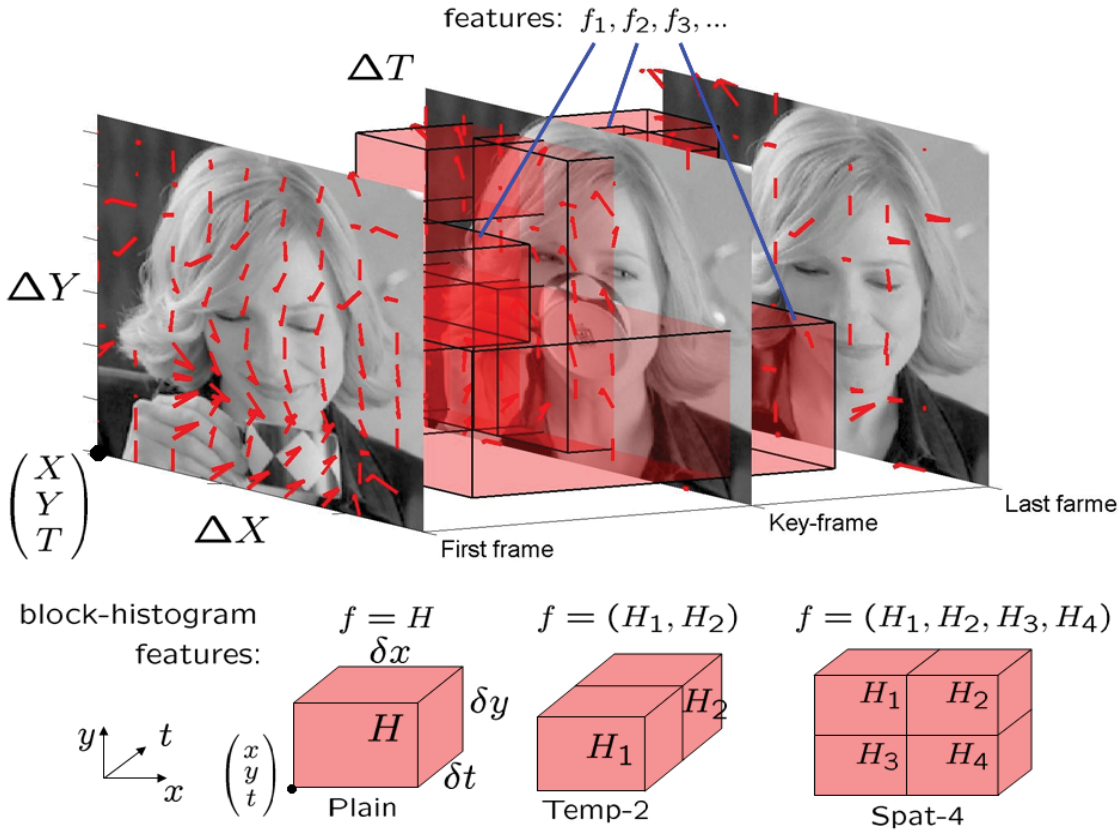


FIGURE 2.10: An example on space-time features represented by three frames of a drinking action. Three types of features with different arrangement of histogram blocks. Histograms for composed blocks (Temp-2, Spat-4) are concatenated into a single feature vector. (Ikizler and Duygulu, 2009)

A second approach is to compute the local descriptors around specific *interest points* in the image. Such points are detected with dedicated detectors such as, for instance, corner detectors. In this approach, the various patches can be assigned decreasing weights based on the spatial and temporal distance from the current pixel being analysed (Zhao and Elgammal, 2008).

A third approach refers to videos and consists of extracting histograms based on both the oriented gradients within the images and along the image flow. It is based on specific interest points within a spatio-temporal grid, allowing for the combination of the two approaches previously described. In this third approach, the grid

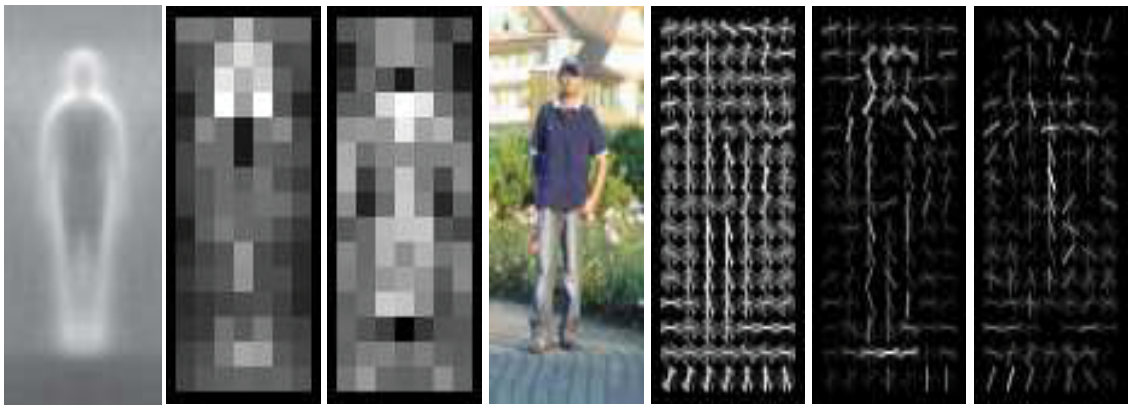
used spans a volume that is based on the position and size of a head within the image. The distribution of the histogram is identified for each spatio-temporal cell identified within the grid. This is then repeated for the whole image (Laptev and Pérez, 2007).

In order to accomplish this task, three different block types from the feature set are used. Each of these types corresponds to an individual cell, combining the temporal aspects of the two neighbouring cells and combining their spatial aspects. From this, a subset of all potential blocks within the grid that are interconnected are selected and different spatial, temporal, and overlap settings are evaluated (Laptev et al., 2008).

2.2.2.2 Texture-based local descriptors

Texture features measure a mean of the intensity of different surfaces. For example, smoothness, regularity, and weaves strength (Yang et al., 2011). These features are prominently used in the browsing and identification of specific objects for image retrieval purposes in a stream of images or frames (Manjunath and Ma, 1996).

The identification of the gradient feature is a type of texture feature. It falls into two distinct categories. The first is a standard of local features, where basic techniques are applied to identify and represent contours. The second type allows for the statistical summarization of gradients for object recognition (Dalal and Triggs, 2005, Gavrila, 2000). Figure 2.11 represent training stages and testing stages for a human detection using gradient features.



(a) (b) (c) (d) (e) (f) (g)

FIGURE 2.11: An example on texture features represented by gradients, where (a to c) are represent training stages and (d to g) represent testing stages . (Dalal and Triggs, 2005)

In recent years, more effort has been devoted to examining an image's local patterns for object detection and action recognition (Mu et al., 2008). Many local features have been proposed for tracking in conventional video, including, amongst others, the popular spatio-temporal interest points (SIFT) (Lowe, 2004), speeded-up robust features (SURF) (Bay et al., 2008) and local binary patterns (LBP) (Pietikäinen et al., 2011). To shed more light on the generic mechanism of local feature extraction, space-time interest points (STIP) descriptors will be elaborated in more detail in the following subsection.

2.2.2.3 Space-time interest points local descriptors

These descriptors consist of space-time interest points (STIPs). In videos, these points are patches in space and time with abrupt changes due to motion. These points occur when a given motion changes direction or turns a corner, resulting in a specific effect on both the spatial and temporal aspects of the image.

These descriptors can be sampled independently from background noise, clutter, occlusions and changes to appearance. The main challenge to these descriptors is

how to detect a sufficient number of STIPs per frame (Laptev, 2005, Wang et al., 2009). Figure 2.12 presents an example of STIPs which are extracted for a "walking" action. In this action, the STIPs come into correspondence with the different parts of the legs that move while walking.

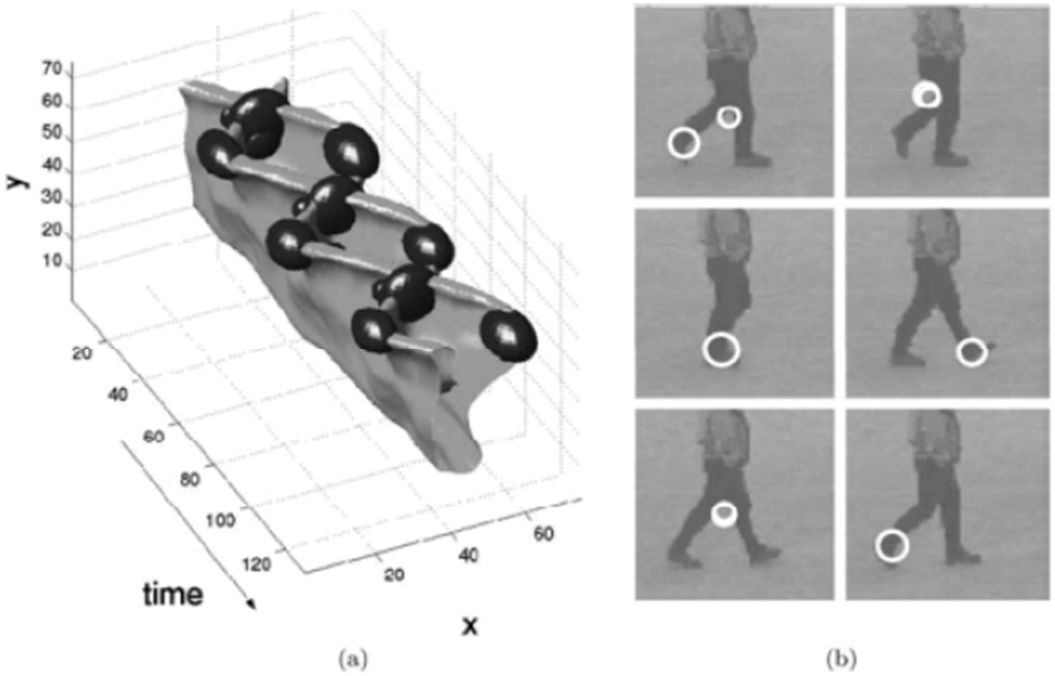


FIGURE 2.12: Examples of spatio-temporal interest points for a "walking" action: (a) 3D plot of leg pattern shown upside down to simplify interpretation; (b) spatio-temporal interest points detection overlayed on walking legs (Laptev, 2005)

Local features are usually used with the so-called *bag-of-words encoding*. In the visual field, it is also often referred to as bag-of-features (BoF). All the extracted STIPs are first clustered based on a notion of distance (Laptev, 2005), usually with the k-means algorithm. Then, each cluster is considered as a codeword and its population size counts as the frequency, forming a histogram of features.

The main advantage of using an encoding is that it summarises the STIP information while drastically reducing its size. Its main drawback is that the spatial and temporal

aspects of the features are neglected, considering only the frequency of each pattern not where and when it occurs. However, a BoF encoding can be extended to include the location and time information.

2.2.3 Depth Features

In the previous sections, we have described in great detail the different types of features that have been proposed in the literature to characterise grey-level and colour frames. All these features could also be used "as is" to characterise other types of frames, such as for instance the *depth* features which have been used extensively in this thesis. Depth features are extracted from frames that contain the distance, or "depth", of the objects in the view from the camera's focal centre. The simplest way of creating a depth map-based feature is to treat them as if they were grey-level images, "recycling" descriptors such as HOG, SIFT, STIP, HOF and kernel descriptors.

While applying such features on depth map sequences achieves reasonably good performance, these depth maps are inherently descriptors of three-dimensional shapes, not of appearances. Consequently, the classification of objects and human actions from depth frames could be better addressed through the exploitation of the properties of three-dimensional shapes (Alexiadis et al., 2013).

A depth map can be seen as a 3D shape. It naturally follows that a sequence of depth maps can be seen as a 4D spatio-temporal pattern. This sequence of depth maps thus results in a 4D video volume (Wang et al., 2014).

According to Vieira et al. (2012), the 4D spatio-temporal patterns of human actions can be usefully characterised through space-time *occupancy* patterns (STOPs). This characterisation is obtained by partitioning the human actions into 4D spatio-temporal cells. Cells are aggregated to increase the occupancy information of single

cells as shown in Figure 2.13.

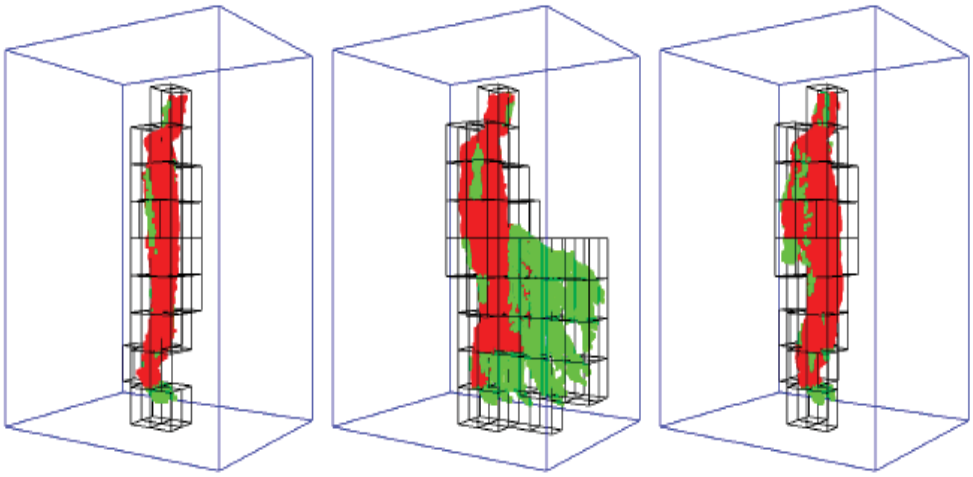


FIGURE 2.13: Space-time cells of a depth sequence of the forward kick action. For each time segment, the frames are placed together in the same space. (Vieira et al., 2012)

Other researchers such as Tang et al. (2012) estimated that a 3D normal (i.e., perpendicular) vector exists for every point on the surface of a 3D volume. This makes it possible to collect histograms of oriented normal vectors (HDNV). The work of Oreifej and Liu (2013) shows that a sequence of depth maps can be used to describe a 4D spatio-temporal shape by using the oriented 4D normald (HON4D). This allows a 4D normal to be calculated for each specific point located on this particular shape, collecting 4D normal vectors through a histogram, as shown in Figure 2.14.

Finally, Xia and Aggarwal (2013b) proposed a depth *cuboid* which is an extension of the cuboid features proposed by (Dollár et al., 2005) to depth map sequences. These researchers used a 3D filtering method to remove noise in the depth maps before applying the interest point detector. The cuboid features are extracted from a 3D volume centred around every interest point. The 3D volume is divided into a grid of 3D cells, and the pairwise similarities of the cells are computed as the features. The cuboid depth feature is reasonably invariant to spatial and temporal

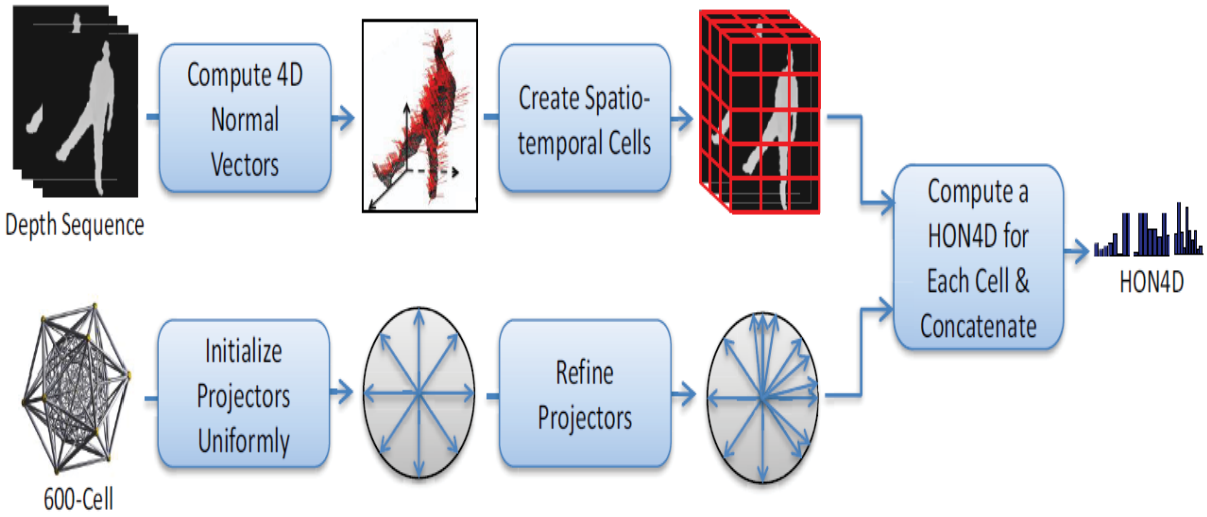


FIGURE 2.14: Example of HON4D descriptors for each cell and their concatenation. (Oreifej and Liu, 2013)

shifts, scales, background clutter, partial occlusions, and multiple motions in the scene.

However, based on our knowledge, none of the above depth features were used for tracking or fine-grained activity recognition in depth videos only.

2.3 Detecting events and activities in a video

There is a great deal of research and existing knowledge surrounding activity recognition and event detection, particularly relating to computer vision (Ke et al., 2013, Poppe, 2010). Despite the large amount of results on these topics, there are many outstanding challenges. The main research to date is explored in the following subsections.

2.3.1 Human-centred action detection

This approach focuses on the most salient people in the video sequence. It is used to detect a human by recognising its particular activities. Part-based approaches are one of the techniques used in human-centred activity detection. This technique can be used to first locate each body part; then, once the parts are detected, their trajectories can be processed using methods like the HMM (hidden Markov model) to identify the actions. The main drawback of this approach is that it does not tend to be reliable in crowded scenes (de Campos, 2014).

To tackle this issue, Dalal and Triggs (2005) and Felzenszwalb et al. (2010) devised a technique of using bounding boxes around people. The motion within each bounding box is described in a "discriminative" way (action x vs action y), without necessarily locating each limb of the person.

For instance, Gorelick et al. (2007) relied on the individual segmentation to build a 3D (space-time) shape. This shape can be described as a vector and matched to templates. De Campos et al. (2011) and Kläser et al. (2010) describe actions as space-time volumes, but builds the feature vectors based on a 3D generalisation of HoG. In this way, actor detection and tracking is required but segmentation is not. These methods can then be fruitfully applied on more realistic videos.

2.3.2 The "blind" approach

This approach is used in cluttered videos, where it is harder to depend on accurate target detectors. The "blind" approach describes video sequences as a set of unorganised space-time local descriptors. In this approach, bags of visual words (BoW) or any of the many encoding methods are used. For instance, Wang et al. (2009) presented a benchmark of local space-time features for the BoW framework,

concluding that densely sampled features gave the best results. The best local feature extraction methods were HOG3D (Klaser et al., 2008), and HOG/HOF (Laptev et al., 2008).

However, better results were obtained more recently using dense trajectories (Wang et al., 2013b). For instance, (Chatfield et al., 2011) presented a benchmark on encoding methods for feature pooling on static images, concluding that Fisher Kernels (Sánchez et al., 2013) are the most effective. The use of Fisher Kernels encoding for action classification was demonstrated on two small datasets in Atmosukarto et al. (2012) and Oneata et al. (2014), showing state-of-the-art results on action classification and temporal localisation. More details about Fisher encoding can be found in chapter 4.

Other studies combined dense trajectories and motion boundaries to build action descriptors. For instance, Wang et al. (2013b) created a hybrid between methods. Another promising alternative piece of research is the representation of video sequences by combining local event detectors to build global feature vectors. This approach is computationally more demanding, but has given much better results with challenging benchmarks (Sadanand and Corso, 2012).

2.3.3 Fine-grained activity recognition

Fine-grained activities can be defined as human activities involving small objects and small movements. Automatic recognition of such activities can prove useful for many applications, including detailed diarization of meetings and training sessions, assistive human-computer interaction, and robotics interfaces. The ability to recognize fine-grained activities has a great deal of potential applications in the real world, where it is often necessary to analyse a complex action and break it down into its individual parts or components. Other areas in which fine-grained activity

recognition can find useful application include sport activities, avian categorization, and kitchen classification (Riboni et al., 2015, Sun et al., 2015, Yao et al., 2011).

The two main interrelated issues in fine-grained activity recognition are the localization of the relevant objects and parts of interest, and the characterization and modelling of their shape and movement (Zhou et al., 2014). Accordingly, the related work is organized hereafter over two subsections concerning object localization and classification approaches, respectively.

2.3.3.1 Object localization

In many cases, the research on object localization has exploited the use of attached instrumentation such as RFID tags and accelerometers to locate and identify the objects of interest. Such sensors are often used in conjunction with video cameras so that the localization accuracy can be increased by aligning their data with that of the video (Chai et al., 2013, Donahue et al., 2013, Jia et al., 2013, Stein and McKenna, 2013, Stikic et al., 2008, Wah et al., 2011, Yao et al., 2011).

The training of these combined approaches require annotation of the bounding boxes of the objects in the visual data, and synchronization with the sensor data. Other approaches are based on *active learning* and require human intervention during the system's training. For instance, Wah et al. (2011) required user clicks as a means of guiding the machine towards correct identification. Branson et al. (2010) exploited online supervision to improve the model; and Xie et al. (2013) assumed knowledge of the true position of parts at run time. To relax the requirements on video annotation, Sun et al. (2015) recently proposed searching the Web for 'highlights' of the objects of interest. However, this approach is heavily affected by the inaccuracy of the search results. To the best of the research knowledge, Sánchez

et al. (2011) is the only work to date to have addressed fine-grained activity recognition without any annotation of the frames. However, it only addresses recognition in still images, rather than in live settings as in the scope of this work.

2.3.3.2 Classification approaches in fine-grained activity recognition

Classification approaches are based on machine learning techniques and include both supervised and unsupervised learning. Supervised learning requires the use of labelled data, i.e., pairs of measurement and corresponding class label. Conversely, unsupervised learning only uses unlabelled data, i.e., measurements with no given class labels (Chen and Khalil, 2011). Quite obviously, supervised learning, where possible, generally leads to more accurate classifiers.

As general guidelines, experimenting with a classifier is organised over four main steps. First, the data are divided into a training set and a test set. Then, the training model is built by training the classification algorithm on the training set. Following this, the classification performance of the trained algorithm is tested with the test. Finally, the approach can be deployed on run-time data (Chen and Khalil, 2011).

In fine-grained activity recognition, various classifiers and learning style have been used. For instance, Wah et al. (2011) utilised kernel descriptors with linear SVM classifiers to recognise both fine-grained objects and activities. Other studies have suggested that online supervision should be used as a means of learning better models (Branson et al., 2010). Sun et al. (2015) has assumed that only "weak" video-level annotations are available during training, using weak labels for the temporal segments of the actions and Internet search for generalizing the learned models.

2.3.4 Role of computer vision in various fields of medicine

Computer vision plays a vital role in medicine as an assistive diagnostic tool for medical imaging. Computer vision is utilized in the detection of tumors, arteriosclerosis, epilepsy, Alzheimer's, Parkinson's and other illnesses (Nicola et al., 2015). It has also been used to provide accurate organ measurement dimensions, blood flow and cardiac functions, and to explore the structure of the brain (Nicola et al., 2015).

In recent years, computer vision has also started to find its way into broader hospital automation. For instance, General Electrics has developed a system for surgical tool packaging, delivery and sterilization using robotics, RFID and computer vision (Todd et al., 2015). Pittsburgh-based company Aethon supplies a hospital delivery robot that transports medications, meals and materials and uses infrared computer vision for path finding (Aldo et al., 2015). These are just two examples of the role that computer vision can play in modern hospital automation. As autonomous intelligent systems continue to improve, applications are likely to expand rapidly.

In response to these innovative uses of computer vision, the work of this thesis has included the development of a new system for the automated detection of the "moments of hand hygiene" (please see chapter 5 for details).

2.4 Support vector machines

This section provides an overview of the main machine learning techniques used in this thesis: the multi-class support vector machine (SVM) and structural SVM. It starts by introducing the conventional binary SVM, then it moves on to review the main kernels used in SVM and then addresses multi-class and structural SVM.

2.4.1 Binary SVM

The support vector machine (SVM) was proposed by Vapnik and Cortes in the mid-1990s (soft-margin version) to deal with binary classification problems (Cortes and Vapnik, 1995). SVM is concerned with mapping the input vectors into a high dimensional feature space through a non-linear mapping chosen a priori. A linear decision hyperplane needs to be constructed in this feature space, with special properties, in order to ensure that SVM has a high generalization ability.

The main, attractive properties of SVM are: a) an ability to provide non-linear, discriminative classifiers; b) a convex objective function that enjoys a number of stable and efficient solvers and c) compared to non-parametric techniques, a remarkable sparsity in the sample set. Thanks to these properties, SVM, in its various flavours, has received widespread adoption in many fields of science and engineering.

The binary SVM is used for two-class classification problems and was proposed by Vapnik (1965) to determine an "optimal" hyperplane between the two classes. Vapnik explained that not all the training data are needed to construct the optimal hyperplane, but just a small subset of "support vectors". Such an optimal hyperplane is defined as the linear decision function with maximum margin between the closest points of the two classes (Meyer and Wien, 2015). Figure 2.15 displays a separating hyperplane in binary classification. Mathematically, the training objective of binary SVM is expressed as:

$$
\begin{aligned}
& w^*, b^* = \underset{w,b,\xi \geq 0}{\operatorname{argmin}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\
& s.t.\ y_i(w^T x_i + b) \geq 1 - \xi \quad i = 1 \ldots N
\end{aligned}
\tag{2.10}
$$

Here, $w^T x + b$ is a function known as the SVM score, with w the weight vector and b the bias. Equations such as $w^T x + b = y$ define a hyperplane in x-space (i.e., a straight line in the 2D case) and $w^T x + b = 0$ is the separating hyperplane between the two classes. The bias, b, translates the hyperplane away from the origin while w establishes the rotation. The constraints require that the points of the positive class (y_i = 1) have a score greater than or equal to 1, and that the points of the negative class have a score less than or equal to -1. Where this is not possible, variables ξ_i relax the constraints. Given that the distance between the closest points of the two classes is proportional to $1/\left\|w\right\|$, minimising $\left\|w\right\|$ in the objective increases the class separation and ensures generalization. Once training is complete, classification of a new point is simply achieved as:

$$\begin{aligned} w^T x + b \geq 0 &\rightarrow \text{positive class} \\ w^T x + b < 0 &\rightarrow \text{negative class} \end{aligned} \tag{2.11}$$

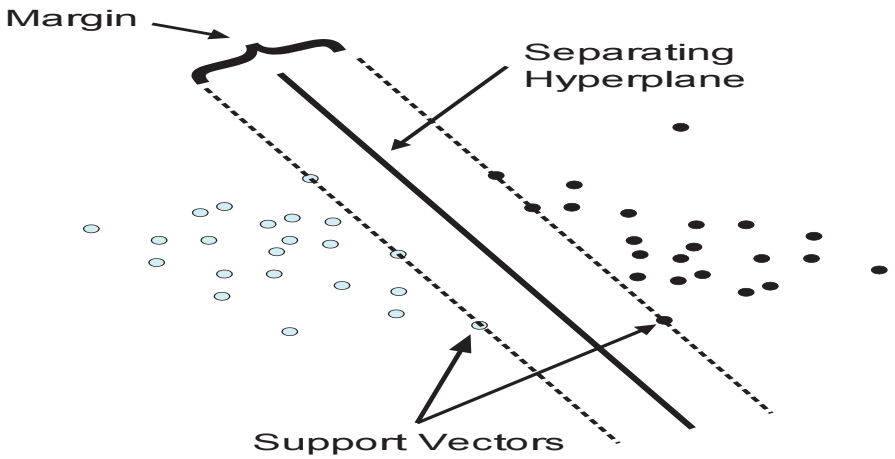


FIGURE 2.15: Binary SVM(Meyer and Wien, 2015)

Figure 2.15 shows the main geometry of SVM: the optimal hyperplane is unique and separates the training data with a maximal margin: it determines the direction

$w/|w|$ where the distance between the projections of the training vectors of two different classes is maximal.

A great deal of research has been conducted on the application of binary SVM in the completion of various studies regarding activity recognition, object detection, and/ or action recognition. Just to cite a few, Dollár et al. (2005) work in object recognition, specifically in its application to behaviour recognition through the use of sparse spatio-temporal features. Poppe (2010) worked on the use of binary SVM in vision-based human action recognition, i.e. the practice of labelling image sequences with action labels.

Binary SVM has been usilised in the proposed hand hygiene system, presented later in this thesis.

2.4.2 Kernels: SVM from Linear to Nonlinear Classifiers

In many applications, a non-linear classifier provides better accuracy than a linear counterpart. The transition to non-linear classification is imperative when the data are not linearly separable, which is the common case for almost every application. But the question is: how can we transform a linear classifier into a non-linear classifier?

The answer provided by kernel approaches is to map the data from the original space x to a feature space f using a non-linear function ϕ. In the f-space, the discriminant function is:

$$y = w^T \phi(x) + b \tag{2.12}$$

The main issue is how to find a suitable mapping function ϕ:$x \rightarrow f$. In practice, this mapping is rarely possible or restrictive. However, we can briefly say that SVM

is able to circumvent this restriction since the values of function ϕ never appear in isolation, but only in products (this would require details of the dual training and inference which we skip herewith). All is needed is to be able to compute $\phi^T(x_i)\phi(x_j)$, which is achieved by means of Mercer kernels: $K(x_i, x_j)$ (the "kernel trick"). The ensuing problem is thus how to choose a kernel for a specific problem. There is no standard criteria for choosing the kernel function. Generally, the RBF (Radial Basis Function) aka Gaussian kernel performs well. Table 2.1 presents the most popular kernels.

TABLE 2.1: Most used kernels functions (Stanevski and Tsvetkov, 2005)

Kernel function	**Description**
$K(x_i, x_j) = x_i^T x_j$	Dot product kernel
$K(x_i, x_j) = (x_i^T + 1)^p$	Polynomial kernel
$K(x_i, x_j) = e^{-\|x_i - x_j\|^2} / _{2\sigma^2}$	Radial basis (Gaussian) kernel
$K(x_i, x_j) = tanh(kx_i^T x_j - \delta)$	Sigmoidal kernel

Figure 2.16 is an example of a non-linear SVM which used a Gaussian kernel, showing a large number of support vectors.

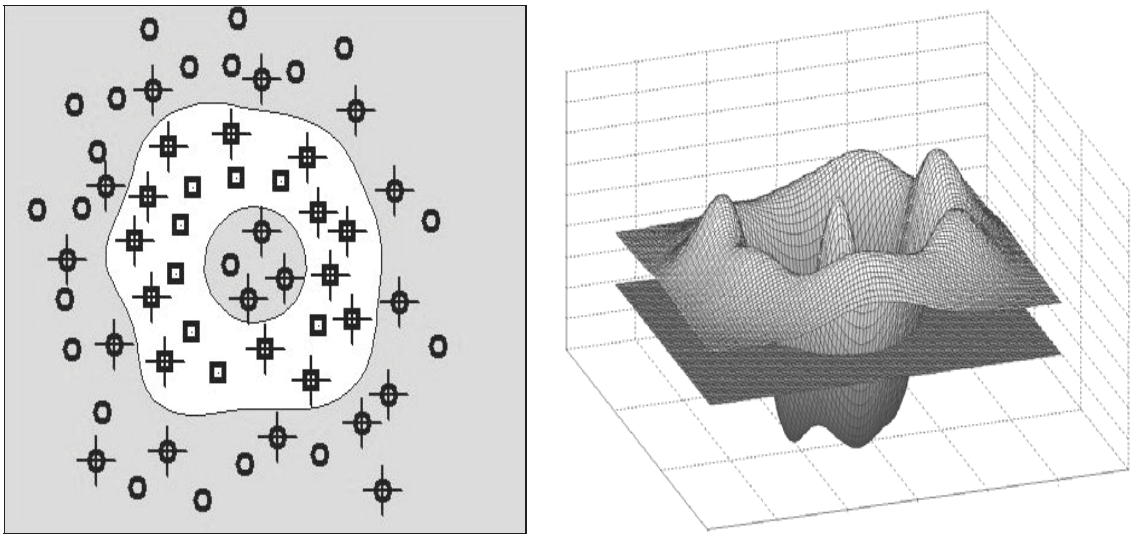


FIGURE 2.16: An example of non-linear classifiers using the Gaussian kernel (Stanevski and Tsvetkov, 2005)

In the work of Boser et al. (1992) the application of non-linear classification, through the use of the kernel trick, was employed for object classification. Within this model, the dots used to identify objects were instead replaced with non-linear kernel functions, allowing the algorithm to be applicable within a given feature space. As a result, multi-dimensional object tracking within non-linear input spaces was achieved, with high enough performance and accuracy (Jin and Wang, 2012).

2.4.3 Multi-Class SVM

Basically, the support vector machine (SVM) was designed for binary classification problems, where the number of classes, K, is equal to 2. However, common applications often require categorization for more than two classes, which are called multi-class SVM. A popular approach to multi-class SVM requires reducing the given multi-class problem into as many as necessary different binary classification problems (Duan and Keerthi, 2005, Hsu and Lin, 2002). The common strategies are to either distinguish between one class and all the others, in a one-versus-all

approach; or to distinguish between each different pair of classes, taking the one-versus-one approach.

In the one-versus-all approach, the tag or label with the highest score will be the assigned class. In this approach, we have K classes and K separate SVM models. Each ith SVM model is trained with the data form ith class as the positive examples and the data from the other $(K-1)$ classes.

By contrast in the one-versus-one approach, various binary classifiers are built, which are used to distinguish between each different pair of classes. In this approach $K(K-1)$ are trained as different binary class SVMs on all possible pairs of classes. The classification occurs through the use of a "Max Wins Voting" strategy in which every classifier assigns one of the two classes, with the vote for the assigned class increased by one until the class with the most votes determines the classification of the instance (Duan and Keerthi, 2005, Hsu and Lin, 2002). When we have a large number of classes, the one-versus-one approach requires significantly more training time than the one-versus-the-rest approach. Furthermore, the former requires much more computation to evaluate test points.

However, since at least 1999, effective, "true" multi-class support vector machine have also been available. The great advantage of these solutions is that the score functions of all the various classes are trained jointly and are, therefore, up to a "common scale". In our investigation of fine-grained activity recognition, we have used a true multi-class SVM from Lauer and Guermeur (2011).

2.4.4 Structural SVM

Structural SVM can be called the third primary type of support vector machine, where multi-class and binary SVM are the other two popular forms of SVM algorithms. Most commonly used for object tracking, text mining, and classification, the

application of structural SVM allows the researcher to train a classifier for a general structured output label. If, for example a structural SVM is applied to a sentence, the output label will be an annotated parse tree (Le Nguyen et al., 2005).

In structural SVM, x is a variable $\in X$ and y is a variable $\in Y$, and in a given set of l training of input-output pairs $(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)$ drawn from some fixed but unknown probability distribution. The general problem is to learn a score function $F : X \rightarrow Y$ between input space X and discrete output space Y based on a training sample of input-output pairs. As an example, consider the case of natural language parsing where function F maps a given sentence x to a parse tree y. This process is displayed visually in Figure 2.17.

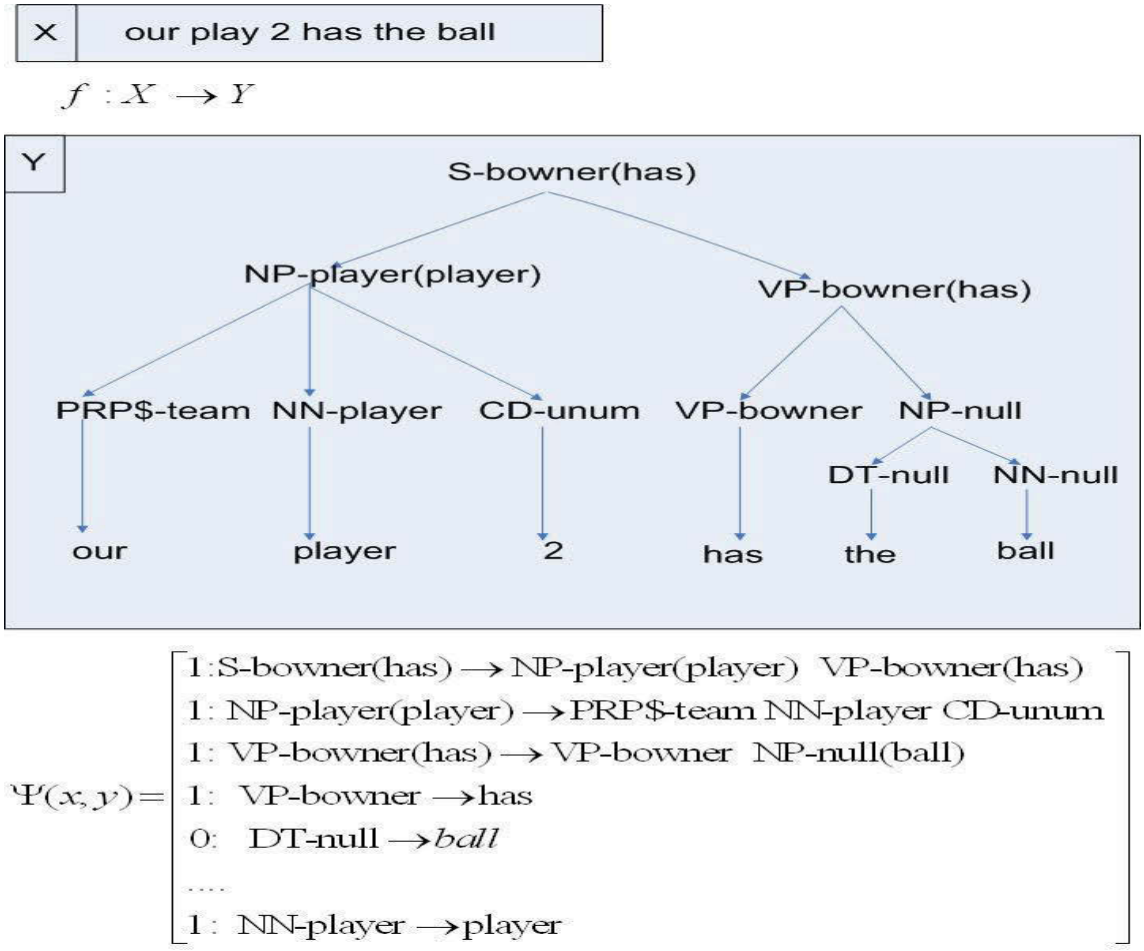


FIGURE 2.17: An example of natural language parsing by structural SVM from (Le Nguyen et al., 2005)

Once f is trained, inference can be obtained as:

$$f(x; w) = \underset{y \in Y}{\operatorname{argmax}} F(x, y; w) \tag{2.13}$$

where w denotes the usual parameter vector. By restricting F to be linear in some combined feature representation of inputs and outputs $\psi(x, y)$, we have:

$$f(x, y; w) = \langle w, \psi(x, y) \rangle \tag{2.14}$$

The training of structural SVM minimises a regularized risk using the following objective:

$$\begin{aligned} &\min_{w} \|w\|^2 + C \sum_{n=1}^{l} \max_{y \in Y} (\Delta(y_n, y) + \langle f(x_n, y; w) - f(x_n, y_n; w) \rangle) \\ &\min_{w} \|w\|^2 + C \sum_{n=1}^{l} \max_{y \in Y} (\Delta(y_n, y) + \langle w, \psi(x_n, y) - w, \psi(x_n, y_n) \rangle) \end{aligned} \tag{2.15}$$

The problem above is often rephrased by introducing explicit, extra variables, ξ_n, for each sample. The standard structural SVM primal formulation is then given as follows:

$$\begin{aligned} &\min_{w,\psi} \|w\|^2 + C \sum_{n=1}^{l} \xi_n \quad s.t. \\ &\langle w, \psi(x_n, y_n) \rangle - \langle w, \psi(x_n, y) \rangle + \xi_n \geq \Delta(y_n, y), n = 1, ..., l, \forall y \in Y \end{aligned} \tag{2.16}$$

As said above, function $\psi : X \times Y \rightarrow R^d$ is a feature function that computes a feature vector from a given sample and label. The design of this function depends on the specific structure of the output and the application. $\Delta(y_n, y)$ is a loss function that quantifies the loss associated with predicting y when the ground truth is y_n and that we wish to upper bound during training.

In addition to unnumbered applications in classification, structural SVM can also be used for object tracking. In this case, it treats the object tracking as a classification problem where the goal is to choose the best next "move", and online learning techniques are used to update the object model. This model is used to predict the tracked object location in the next frame. In this thesis, we propose a tracker that is an extension of the popular Struck algorithm (Hare et al., 2011) which leverages a structural SVM framework for tracking. More details about structural SVM for tracking can be found in Chapter 3.

However, the possibility to track and monitor actions solely using depth videos has been largely unexplored to date. The challenge posed by depth tracking is major, as conventional trackers rely on the targets' appearance and texture to provide correct data association. In order to address this, this PhD investigates tracking and fine-grained activity recognition based on depth data alone.

Chapter 3

Tracking in depth videos

In this chapter, we present a novel tracker that operates solely from depth data. The proposed tracker is designed as an extension of the popular Struck algorithm which leverages the effective framework of structural SVM. The main contributions of this work are: i) a dedicated depth feature based on local depth patterns, ii) a heuristic for handling view occlusions in depth frames, and iii) a technique for keeping the number of the support vectors within a given "budget" so as to limit computational costs. Experimental results over the challenging Princeton Tracking Benchmark (PTB) dataset report a remarkable accuracy compared to the original Struck tracker and other state-of-the-art trackers using depth and RGB data.

3.1 Introduction and Background

The aim of video tracking is to extract the trajectories of a chosen set of targets. In recent years, the release of sensors such as Microsoft Kinect has made it possible to acquire depth videos inexpensively. A significant trend in tracking research has become the use of depth data in addition to RGB data to disambiguate occlusions

and overcome illumination artifacts (Basso et al., 2013, Munaro et al., 2012, Song and Xiao, 2013).

However, the possibility to perform general tracking solely from depth videos has received only partial attention to date. One of the most popular approaches for video tracking is known as *tracking-by-detection* (Avidan, 2007, Babenko et al., 2009, Breitenstein et al., 2011, Grabner et al., 2008, Guo et al., 2014, Tran and Davis, 2007). Its main idea is to frame tracking as a target classification problem and learn the classifier in an online and unsupervised manner. In this category, the Struck tracker from Hare *et al*. (Hare et al., 2011) has recently attracted much attention since it leverages the efficient, discriminative framework of structural SVM and has reported a remarkable accuracy in a number of evaluations (Li et al., 2013, Smeulders et al., 2014, Wu et al., 2013). Its main rationale is to use patches of the video frames as the support vectors of an SVM, maintaining the set dynamically and within a "budget" so as to not compromise real-time operation. For these reasons, we have decided to adopt it as the base for our depth tracker.

In initial tests with depth video, Struck showed some limitations that motivated the extensions that are the focus of this work. One of the main challenges in tracking from depth data is the design of features effective at tracking single targets through severe occlusions. In our experiments, existing features such as appearance histograms and Haar features did not seem as effective as they are on RGB data.

3.2 Related work

Since their inception, consumer depth cameras have found increasing adoption in computer vision and multimedia. The widespread availability of depth data has led to the proposal of several dedicated features which, in most cases, are adaptations of pre-existing appearance features. For instance, (Oreifej and Zicheng, 2013) has

proposed the HON4D feature which is a histogram of oriented 4D normals suitable for recognizing activities from depth video. Xia and Aggarwal have proposed a modification of the popular STIP detector and descriptor in (Xia and Aggarwal, 2013a). In (Lu et al., 2014), the authors have proposed a simple range-depth feature computed around the location of skeletal joints. In (Zhao et al., 2012), Zhao *et al*. have introduced a depth-based version of local binary patterns (Pietikäinen et al., 2011).

In addition to these works on features, depth data have found significant use as an additional modality for tracking: for instance, (Yang et al., 2011) leverages point-cloud clustering of depth pixels; (Basso et al., 2013, Munaro et al., 2012) use depth-based hierarchical clustering for tracking both individuals and groups; while (Song and Xiao, 2013) have used the depth information to resolve occlusions between targets. Following (Song and Xiao, 2013, Zhao et al., 2012), in this work we propose to adopt a dense local descriptor aggregating depth values from the target's area.

In machine learning, an important problem is the learning of a classifier under a "budget" constraint, aiming at speeding up both the training and the classification (Singer, 2004). This problem is even more urgent in tracking-by-detection where the online learning of the classifier must be performed within real-time constraints. In the case of SVM, the budget constraint limits the number of support vectors that can be used in the classifier. The general strategy for adding a support vector is to add it at its first appearance, while the decision to remove it is more critical and arbitrary.

Therefore, several approaches have been proposed for removal: (Cavallanti et al., 2007, Vucetic et al., 2009) remove samples based on a random selection; (Dekel et al., 2008) removes the oldest support vectors; while (Schuurmans and Caelli, 2007, Wang et al., 2010, Weston et al., 2005) and also Struck (Hare et al., 2011) remove the support vector that causes the minimum L^2 norm change to the SVM

primal model. In this work, we have decided to follow a different approach based on the notion of *prototype selection* (Riesen and Bunke, 2010). The most common use of prototypes is for the embedding of non-vectorial objects such as strings, sets and graphs. Prototypes are typically selected from a given object set based on various centrality or uniformity measures (Riesen and Bunke, 2010). In this work, we have decided to evaluate three different prototype selectors to remove the support vector that is possibly the most redundant or otherwise an outlier inside the current set. The experimental results presented in Section 3.5 show the effectiveness of this approach.

3.3 The Struck tracker: overview

The Struck tracker was proposed in (Hare et al., 2011) as a principled improvement to tracking-by-detection approaches. It leverages the framework of structural SVM (Tsochantaridis et al., 2005) to provide a prediction for the movement of a target, y, from its current position, p_t. By noting as x_t the frame at time t, Struck provides the prediction as the inference of a generalized linear model:

$$\bar{y} = \underset{y}{\operatorname{argmax}}\, w^\top \phi(x_t, y) \tag{3.1}$$

where feature function $\phi(x_t, y)$ computes a feature vector from a patch of pixels inside frame x_t centred at position $p_t + y$, and product $w^\top \phi(x_t, y)$ assigns it a score. In other terms, $w^\top \phi(x_t, y)$ is a joint model for the displacement and appearance of a target.

The challenge with maintaining the parameter vector, w, is that it has to be adapted at every new frame and in an unsupervised manner. This is achieved by framing

this learning problem as a structural SVM objective and providing a heuristic for its online update. The structural SVM primal objective is expressed as:

$$\begin{aligned} &\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad s.t. \\ &w^\top\phi(x_i, y_i) - w^\top\phi(x_i, y) \geq \Delta(y_i, y) - \xi_i, \\ &\quad i = 1 \ldots N,\ \forall y \in \mathcal{Y} \end{aligned} \tag{3.2}$$

The objective in equation (3.2) is the standard SVM primal objective balancing an upper bound over the empirical loss, $\sum_{i=1}^{N}\xi_i$, with a regularization term, $\|w\|^2/2$. For brevity of notations, x_i notes the feature vector associated with displacement y_i. The constraints in equation (3.2) impose that the score assigned to the true displacement of the target, y_i, is greater than that assigned to any other displacement, $y \neq y_i$, by an amount decided by a chosen loss function, $\Delta(y_i, y)$. At its turn, the loss function is set to reflect the overlap between two bounding boxes centred, respectively, on the target's true location, y_i, and predicted location, y:

$$\Delta(y_i, y) = 1 - overlap(y_i, y) \tag{3.3}$$

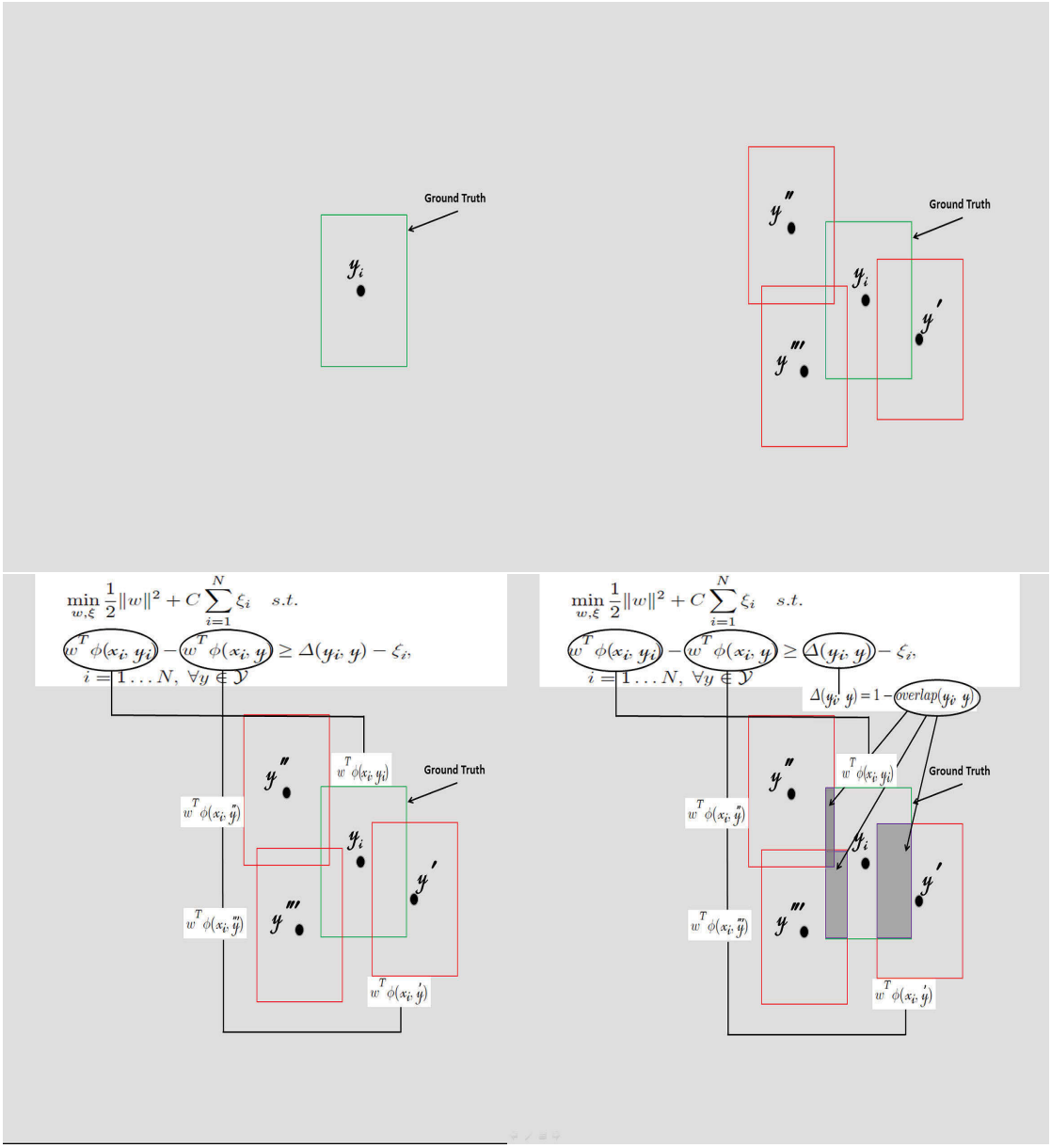


FIGURE 3.1: The main steps of Struck: a) the estimated ground-truth bounding box at frame i (a positive support vector); b) other bounding boxes around the ground truth (negative support vectors); c) the score, $w^\top \phi(x, y)$, of all bounding boxes is computed; d) the constraints in equation (3.2) impose that the score of the true displacement, y_i, is greater than that of any other displacement, $y \neq y_i$, by an amount set by the chosen loss function, $\Delta(y_i, y)$. At its turn, $\Delta(y_i, y)$ is chosen to be complementary to the overlap between bounding boxes y_i and y.

Figure 3.1 shows the main steps of Struck. The challenges with the SVM problem

in equation (3.2) are that the actual ground truth is unknown, and that the model requires updating at every new frame. To this aim, Struck predicts the ground-truth labeling of a new sample, y_i, based on the current model, and uses a heuristic to select samples and labellings for the weight updates that we briefly describe in the following. As shown in (Hare et al., 2011), the primal SVM objective equation (3.2) can be turned into this equivalent dual problem:

$$\begin{aligned} &\max_{\beta} -\frac{1}{2}\sum_{i,y}\sum_{j,y'}\beta_i^y\beta_j^{y'}\phi(x_i,y)^\top\phi(x_j,y') - \sum_{i,y}\beta_i^y\Delta(y_i,y) \\ &s.t.,\ i=1\dots N: \\ &0\le\beta_i^{y_i}\le C,\ \beta_i^y\le 0\ \forall y\neq y_i,\ \sum_y\beta_i^y=0 \end{aligned} \tag{3.4}$$

The maximization in equation (3.4) takes place over a vector of variables, β, where β_i^y denotes the variable for sample $i=1\dots N$ and labeling $y\in\mathcal{Y}$. Such variables have different sign constraints: those for the estimated ground truth, $\beta_i^{y_i}$, are constrained to be positive, while the others, $\beta_i^y, y\neq y_i$, negative. Therefore, we refer to the respective vectors, (x_i,y_i) and $(x_i,y\neq y_i)$, as "positive" and "negative" vectors for short in the following.

The solver for equation (3.4) is an SMO (sequential minimal optimization) algorithm that sequentially selects a sample, x_i, and a pair of its β coefficients for update (Platt, 1999). The two chosen coefficients, renamed as $\beta_i^{y_+}$ and $\beta_i^{y_-}$, are modified, respectively, as $\beta_i^{y_+}+\lambda$ and $\beta_i^{y_-}-\lambda$, with $\lambda\ge 0$, so as to preserve the sum-to-zero constraint of (3.4). The choice of the two coefficients is performed by identifying the direction with the highest directional derivative of the objective function, in order to gain maximum benefit from the update. By noting as $g(y)$ the derivative along β_i^y, we select $y_+=\operatorname{argmax}_y g(y)$ and $y_-=\operatorname{argmin}_y g(y)$: in this way, moving "toward" $g(y_+)$ by $+\lambda$ and "against" $g(y_-)$ by $-\lambda$ guarantees moving

along the direction with the highest derivative for any possible pair of dimensions for this sample.

If the update modifies one of the β_i^y coefficients from an initial value of zero to a different value, its (x_i, y) vector is included in the current working set of support vectors. When the budget is eventually reached, an existing support vector is selected for removal so as to minimize the change in L^2 norm to the primal model, $w = \sum_{i,y} \beta_i^y \phi(x_i, y)$. The reader can refer to (Bordes et al., 2007, Hare et al., 2011) for further details. The last component of the tracker is feature vector $\phi(x, y)$. As options, Struck provides:

- a 192-D Haar-like feature vector extracted from a grid centred at displacement y;
- a 256-D feature vector of spatially re-scaled raw pixels;
- a 480-D feature vector obtained from the concatenation of 16-bin intensity histograms computed on a four-level pyramid.

3.4 Extensions for depth tracking

In this section, we present the proposed extensions to Struck consisting of a novel depth descriptor (Section 3.4.1), a technique for support vector removal based on prototype selection (Section 3.4.2), and an occlusion handling procedure (Section 3.4.3).

3.4.1 Local depth features for tracking

In this work, we have decided to explore a local depth feature recently proposed for activity recognition in depth video. The feature, called local depth pattern (LDP),

resembles the popular local binary patterns Pietikäinen et al. (2011) in that it computes differences between cells of a local patch Zhao et al. (2012). While this feature has proved effective for activity recognition, its performance for tracking cannot be anticipated since these two tasks rely on very different characteristics of the target.

To form our tracking feature (named LDP for tracking, or LDPT for short), we divide the target's bounding box into an $HD \times VD$ grid of LDPs. As values for HD and VD, we typically select 3 and 4, respectively. At its turn, each LDP contains a 3×3 grid of cells. Given that the bounding box has variable size, the size in pixels of the LDP and its cells adjust accordingly. The value of each LDP is obtained by concatenating the differences between the average depth of each of its cells with every other. Therefore, the total size of the LDPT feature is:

$$size(LDPT) = HD \times VD \times \binom{3 * 3}{2} \tag{3.5}$$

for a total of 432 dimensions. Algorithm 1 shows the detailed steps for computing an LDPT feature.

3.4.2 Support vector removal based on prototype selection

Given a set of vectors or non-vectorial objects and a distance over them, prototype selection aims to find the sub-set of the objects that maximizes chosen properties such as the centrality in the set, uniform spread of the prototypes, proximity to the set boundaries and others. In our case, we aim to use prototype selection to determine the positive support vector (i.e., an estimated target) that is the most "disposable" according to these properties. In particular, we evaluate three different selection strategies, namely the *median*, *centre* and *marginal* support vectors Riesen and Bunke (2010):

Algorithm 1 The algorithm for computing the proposed LDPT feature.

Input: Bounding box
Output: LDPT feature

{initializes the LDPT feature to an empty set:}
$LDPT = \emptyset$
loop r = 1 : VD
  loop c = 1 : HD
    {initializes the LDP(r,c) descriptor to an empty set:}
    $LDP(r,c) = \emptyset$
    loop i = 1 : 9
      {loops over all cells in the LDP descriptor}
      loop j = i + 1 : 9
        {computes the difference with every other cell:}
        $diff(i,j) = |avgdepth(i) - avgdepth(j)|$
        $LDP(r,c) = concatenate(LDP(r,c), diff(i,j))$
      end loop
    end loop
    $LDPT = concatenate(LDP(r,c))$
  end loop
end loop

$$sv^*_{median} = \underset{i}{\operatorname{argmin}} \sum_j d(i,j) \tag{3.6}$$

$$sv^*_{centre} = \underset{i}{\operatorname{argmin}} \max_j d(i,j) \tag{3.7}$$

$$sv^*_{marginal} = \underset{i}{\operatorname{argmax}} \sum_j d(i,j) \tag{3.8}$$

where $d(i,j)$ represents the distance between two positive support vectors, sv_i and sv_j. The median support vector is defined as the support vector minimizing the sum of the distances from the remaining vectors, while the centre support vector minimizes the maximum distance from them. Both these selection strategies aim to remove a support vector that is "central" in the pool and therefore less likely to prove

discriminative. On a different rationale, the marginal support vector maximizes the sum of the distances from the other vectors, and we remove it under the assumption that it may prove an outlier. The chosen distance also plays an important role in the selection: to this aim, we have evaluated three distances: 1) a simple Euclidean distance between the feature functions of support vectors sv_i and sv_j (d_u); 2) a distance weighted by the β coefficients of the two vectors (d_w); and 3) a distance weighted by their square root (d_s):

$$d_u(i,j) = \|(\phi(x_i, y_i) - \phi(x_j, y_j\| \tag{3.9}$$

$$d_w(i,j) = \beta_i^{y_i}\beta_j^{y_j}\|(\phi(x_i, y_i) - \phi(x_j, y_j\| \tag{3.10}$$

$$d_s(i,j) = \sqrt{\beta_i^{y_i}\beta_j^{y_j}}\|(\phi(x_i, y_i) - \phi(x_j, y_j\| \tag{3.11}$$

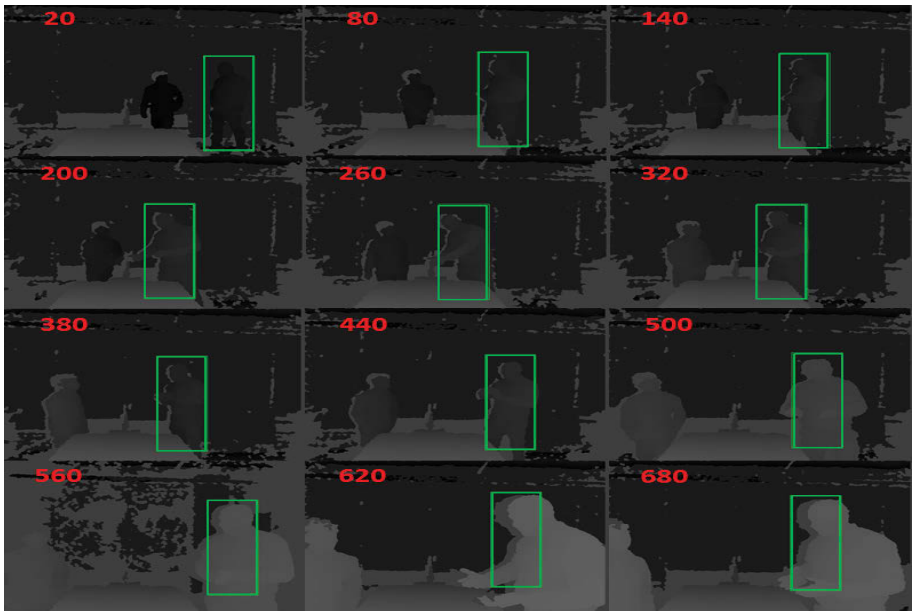
Algorithm 2 shows the main steps of the support vector removal procedure.

Algorithm 2 The proposed algorithm for support vector removal.

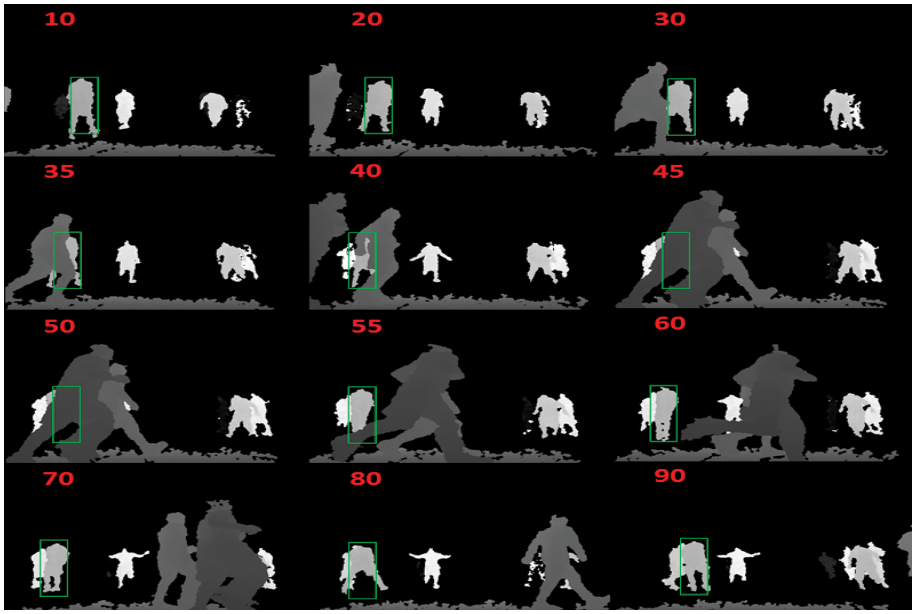
Input: Current SV = $\{sv_1, \ldots, sv_n\}$, with $n >$ size(budget)

Output: New SV = $\{sv_1, \ldots, sv_m\}$, with $m <=$ size(budget)

1: **while** size(SV) $>$ size(Budget)
2: $SV_C = central\_positive\_support\_vector(SV)$
3: **loop** 1 : size(SV)
4: $SV_N = corresponding\_negative\_support\_vector(SV_C)$
5: $SV = SV \setminus \{SV_N\}$
6: **end loop**
7: $SV = SV \setminus \{SV_C\}$
8: **end while**



(A)



(B)

FIGURE 3.2: Examples of occlusion handling in A) the hospital simulation and B) PTB datasets.

3.4.3 Occlusion handling

View occlusions from static objects and other targets are likely the main challenge of tracking. While the weakness of depth data is their lack of appearance features, their strength is the possibility to provide reliable target discrimination based on their distance from the camera. Therefore, in the proposed tracker we have built an occlusion detector that flags an occlusion whenever the depth of the candidate target, d_t, differs from its historical average, d_{avg}, more than a given threshold θ. This threshold is set in centimeters (to 50 cm) so as to have a uniform threshold value across the entire depth range. The measurements are computed at the centre of the respective bounding boxes and the historical average is maintained as a running average of update coefficient λ, updated only in the absence of detected occlusions:

$$occlusion = |d_t - d_{avg}| > \theta \tag{3.12}$$

$$d_{avg}(updated) = \begin{cases} \lambda d_t + (1-\lambda) d_{avg} & \text{if } occlusion = 0 \\ d_{avg} & \text{otherwise} \end{cases} \tag{3.13}$$

If the targets in the scene have significant changes in distance from the camera, the threshold will have to be adapted correspondingly.

Figures 3.2.A and 3.2.B show examples of successful occlusion handling in a video from our hospital simulation dataset and a challenging basketball video from the PTB dataset. The videos with the full results can be visualized from Dropbox [1].

[1] https://www.dropbox.com/s/8codeji5lnzkg22/hospital.avi?dl=0,
https://www.dropbox.com/s/dzuock30489st1u/occlusion.avi?dl=0

3.5 Experiments

3.5.1 Datasets

The proposed tracker has been evaluated both qualitatively and quantitatively using a hospital simulation dataset collected by these authors and the recent Princeton Tracking Benchmark (PTB) dataset Song and Xiao (2013). Our hospital simulation dataset consists of 26 depth videos that stage simulated visits to a patient lying on a hospital bed. These videos are characterized by ample back-and-forth target movement and static occlusions and have been used for qualitative evaluation only [2].

The work on the hospital environment was motivated by a collaboration with clinical researchers from the Intensive Care Unit of Sydney's Royal Prince Alfred Hospital, who provided guidance on the simulation and set the privacy requirement for the video footage. The PTB dataset was released as part of an ICCV 2013 publication to offer a unified, challenging benchmark for tracking in RGB and depth data. It consists of 95 videos varying in target type (humans, animals and substantially rigid objects such as toys and human faces), level of background clutter (plain living rooms, cafes, sport courts etc), and type of occlusions (different durations, appearance changes during occlusions, similarity between targets and occluders etc).

The dataset comes accompanied by an evaluation website [3] managed by the benchmark's authors which allows for an unbiased accuracy evaluation. The evaluation protocol considers three types of tracking errors: Type I errors that occur when the target is visible, but the tracker's output is far off from the target (wrong detections); Type II errors that occur when the target is invisible but the tracker still outputs a bounding box (false detections); Type III errors that occur when the target is visible

[2] the dataset can be downloaded from https://drive.google.com/file/d/0B9cAe42oTaT_-aUs4ckVrazQ1OXM/

[3] http://tracking.cs.princeton.edu/submit.php

but the tracker fails to produce any output (missed detections). Accuracy figures are divided by target type, target size, movement, occlusion and motion type.

3.5.2 Experimental results

Our experiments aim to compare the proposed tracker with the original Struck tracker and other state-of-the art trackers. The qualitative evaluation on the hospital simulation dataset is generally very positive, with the target (a visiting clinician) successfully tracked in all videos. The original Struck tracker instead tends to lose the target in the presence of large static occlusions.

The quantitative evaluation on the PTB dataset provides the test-bed for a rigorous and current performance analysis. Table 3.1, top part, reports the accuracy comparison for the proposed depth tracker against other trackers using only depth data. These include Struck with different types of features and a tracker based on HOG features Song and Xiao (2013). The results in Table 3.1 show that the proposed tracker outperforms the other trackers in 7 categories out of 11. The introduction of the LDTP feature alone achieves an average accuracy improvement of 7 percentage points over Struck with the best feature (histogram; 0.51% vs 0.44%). The addition of the proposed prototype selection approach together with the occlusion handling heuristic achieves a further improvement of 2 percentage points.

For comparison, the bottom part of Table 3.1 reports the performance of trackers using RBG data. The proposed tracker outperforms Struck operating on RGB data in almost every category (10 out of 11). This result is remarkable in that it shows that depth tracking with suitable features can outperform RGB tracking at a parity of targets and scenes. In turn, this proves that depth tracking is a viable approach to tracking under privacy-preserving operating conditions. It is also important to add that the performance of Struck on RGB data was reported in Song and Xiao (2013)

TABLE 3.1: Accuracy comparison for the proposed tracker and other trackers on the Princeton Tracking Benchmark.

Algorithm	**target type**			**target size**		**movement**		**occlusion**		**motion type**		**average**
	human	**animal**	**rigid**	**large**	**small**	**slow**	**fast**	**yes**	**no**	**passive**	**active**	
Struck (Depth videos), Haar	0.31	0.32	0.36	0.29	0.36	0.36	0.32	0.21	0.49	0.36	0.32	0.34
Struck (Depth videos), raw pixels	0.34	0.44	0.42	0.37	0.41	0.44	0.37	0.29	0.54	0.43	0.37	0.40
Struck (Depth videos), histogram	0.38	0.46	0.44	0.45	0.40	0.51	0.39	0.31	0.57	0.52	0.38	0.44
HOG (Depth videos) from Song and Xiao (2013)	0.43	0.48	**0.56**	0.47	0.50	0.52	0.47	0.38	0.63	0.54	0.48	0.50
Proposed tracker (no occl. handling, no prototype selection)	0.39	**0.61**	0.54	0.46	0.51	**0.58**	0.45	0.32	**0.69**	0.56	0.46	0.51
Proposed tracker	**0.45**	0.59	0.54	**0.49**	**0.53**	0.57	**0.49**	**0.39**	0.68	**0.57**	**0.49**	**0.53**
Struck (RGB videos) from Song and Xiao (2013)	0.35	0.47	0.53	0.45	0.44	0.58	0.39	0.30	0.64	0.54	0.41	0.46
OF tracker (RGB videos) from Song and Xiao (2013)	0.47	0.47	0.63	0.47	0.47	0.57	0.52	0.47	0.62	0.63	0.49	0.53
RGBD tracker (RGB and depth) from Song and Xiao (2013)	0.74	0.63	0.78	0.78	0.70	0.76	0.72	0.72	0.75	0.70	0.82	0.74

TABLE 3.2: Comparison of average accuracy with different prototype selection techniques and distances.

Prototype selector	**Euclidean** (d_u)	**Weighted** (d_w)	**Square root** (d_s)
Median	0.49	0.51	0.52
Centre	0.49	0.52	**0.53**
Marginal	0.48	0.49	0.50

as the best out of a pool of popular trackers including TLD Luber et al. (2011), CT Zhang et al. (2012), MIL Babenko et al. (2009), semi-B Grabner et al. (2008) and VTD Kwon and Lee (2010). The only RGB tracker that outperforms our depth tracker in a few categories is the tracker proposed by the authors of the benchmark itself (*OF tracker*, Table 3.1). Remarked improvements over depth tracking alone is only achieved by fusion of depth and RGB information (*RGBD tracker*, Table 3.1).

Figure 3.3 shows cases of success and failure for the proposed tracker and the original Struck tracker. Figure 3.3.a shows a case where Struck wrongly swaps the target with another passer-by due to a temporary occlusion. In the same case (Figure 3.3.b), the proposed tracker continues to track correctly thanks to the effective detection and handling of the occlusion. Figure 3.3.a shows a failure from the proposed tracker due to the sudden fast motion of the target (a small dog). Since the model is a joint model for the movement and appearance of the target, abrupt changes are the main potential cause of failure. The same sequence shows that the proposed tracker withstands another major occlusion around frame 30.

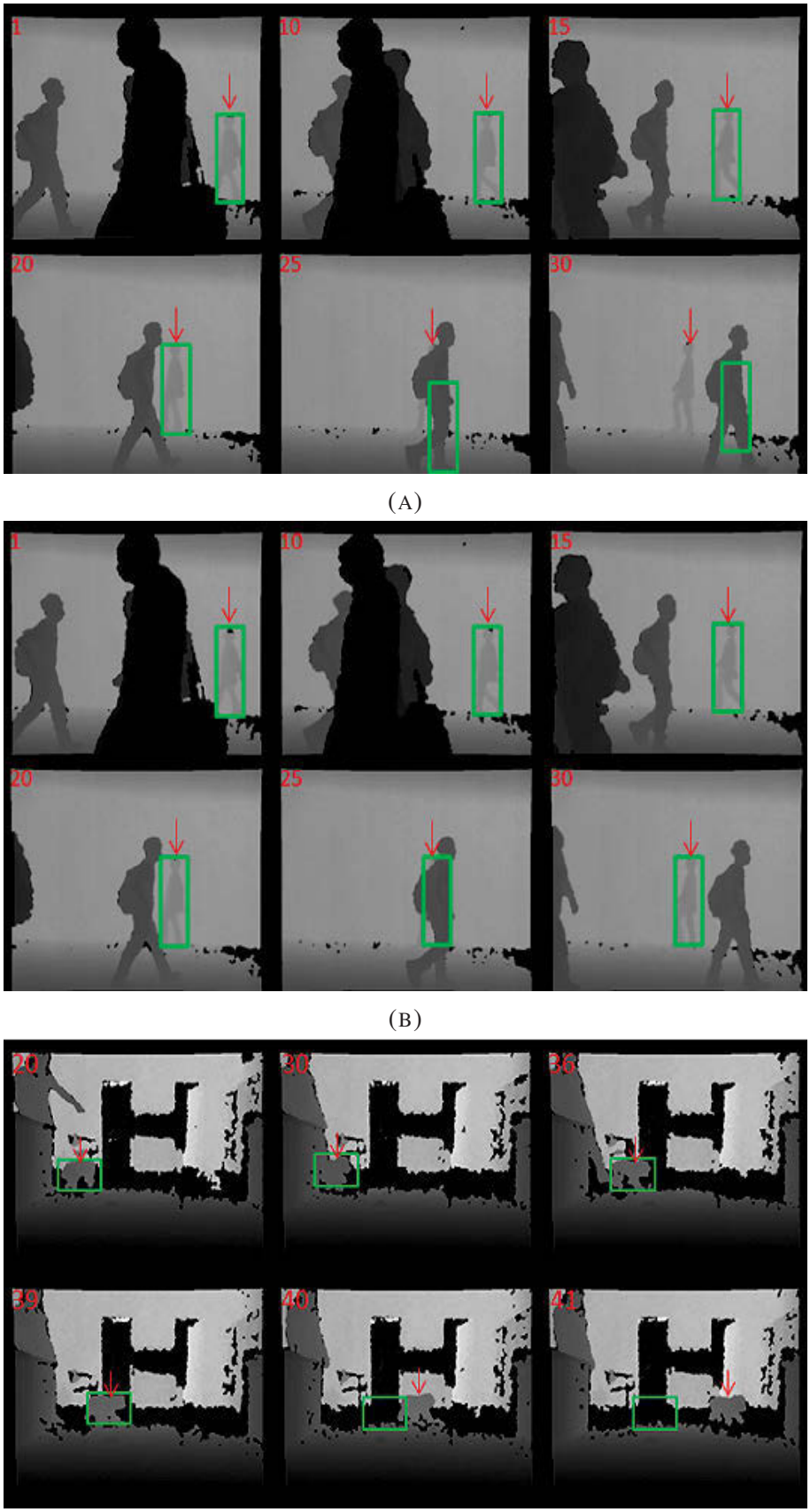
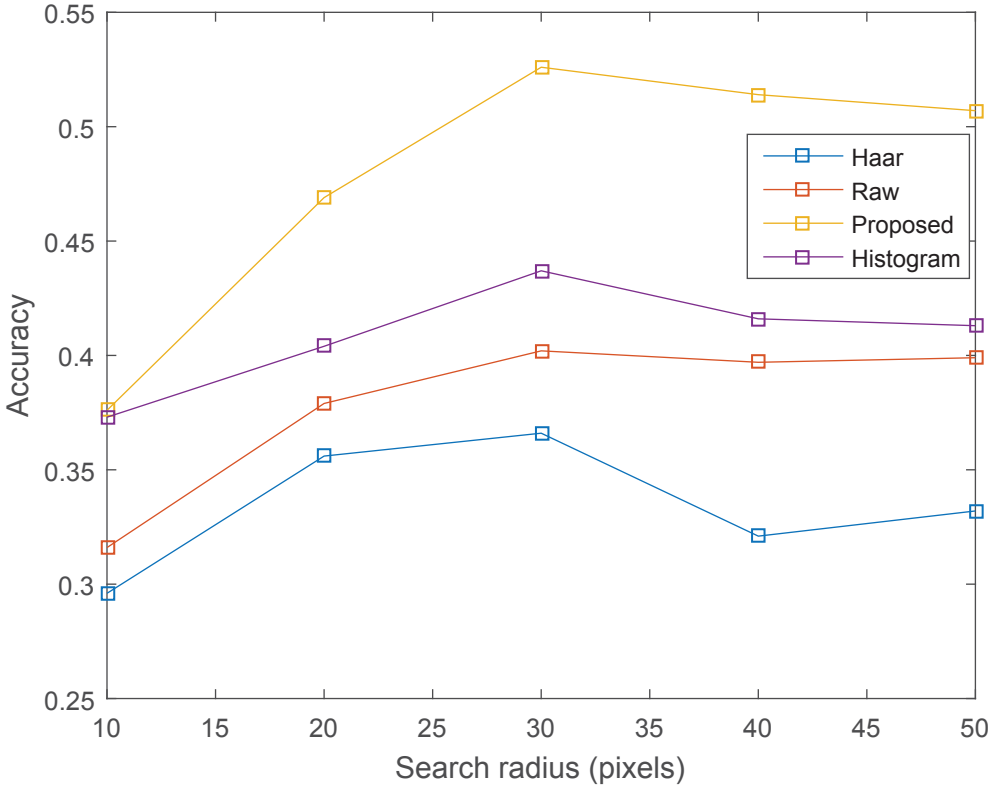
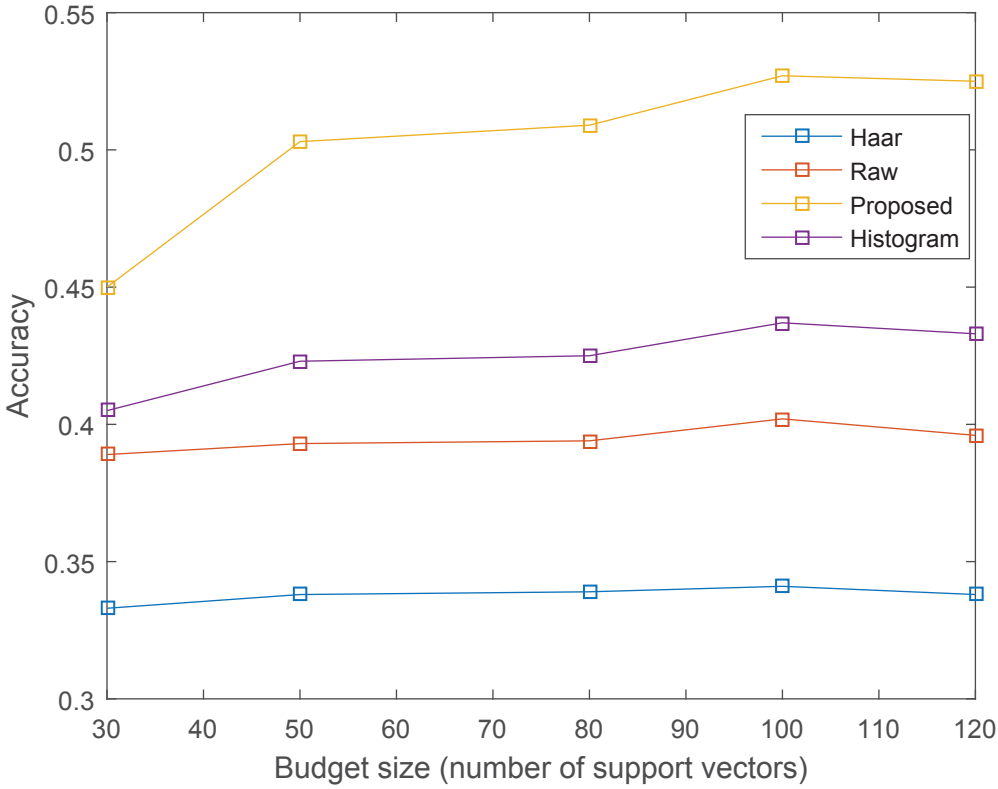


FIGURE 3.3: Cases of success and failure for the proposed tracker and the original Struck tracker.



(A)



(B)

FIGURE 3.4: Comparison between the proposed tracker and the original Struck tracker with various features; A) by varying the search radius; B) by varying the budget size.

To explore the sensitivity to parameters, Figure 3.4.A compares the accuracy achieved using different features as a function of the *search radius* for the target. The search radius determines the maximum distance over which the displacements are computed, and in this experiment it has been made vary between 10 and 50 in steps of 10 while leaving all the other parameters unchanged. The plot shows that the highest accuracy for every value of the search radius is achieved with the proposed LDPT features. The accuracy shows no increase beyond a radius of 30 which is a desirable result given that the computational time increases with larger radii. Likewise, Figure 3.4.B compares the accuracy achieved using different features as a function of the *budget size*.

In this experiment, the budget size has been made vary between 30 and 120 in steps of 10 while, again, leaving all the other parameters unchanged. Figure 3.4.B shows that also in this case the proposed features achieve the highest accuracy for every value in the range, with a maximum for a budget size of 100. These results further validate the usefulness and operational robustness of the proposed approach.

Eventually, Table 3.2 shows a comparison of the average accuracy for different prototype selection techniques in combination with different support vector distances. The results show that the use of weighted distances is generally preferable and that the distance with the square root of the weights' product, d_s, achieves the best accuracy in combination with the centre-based prototype selection.

Chapter 4

Fine-Grained Activity Recognition in Depth Videos

In this chapter, we investigate fine-grained activity recognition in a kitchen setting by solely using a depth camera. The primary contribution of this work is an aggregated depth descriptor that effectively captures the shape of the objects and the actors. Experimental results over the challenging ”50 Salads” dataset of kitchen activities show an accuracy comparable to that of a state-of-the-art approach based on multiple sensors, thereby validating a less intrusive and more practical way of monitoring fine-grained activities.

4.1 Introduction

Fine-grained activities are human activities involving small objects and small movements. Automatic recognition of such activities can prove useful for many applications, including detailed diarization of meetings and training sessions, assistive

human-computer interaction and robotics interfaces. Existing approaches to fine-grained activity recognition typically leverage the combined use of multiple sensors including cameras, RFID tags, gyroscopes and accelerometers borne by the monitored people and target objects. Although effective, the downside of these solutions is that they require minute instrumentation of the environment that is intrusive and hard to scale. To this end, this chapter investigates fine-grained activity recognition in a kitchen setting by solely using a depth camera. The primary contribution of this work is an aggregated depth descriptor that effectively captures the shape of the objects and the actors. Experimental results over the challenging ”50 Salads” dataset of kitchen activities show an accuracy comparable to that of a state-of-the-art approach based on multiple sensors, thereby validating a less intrusive and more practical way of monitoring fine-grained activities.

4.2 Background and related work

The main aim of fine-grained activity recognition is to correctly identify activities of limited inter-class variance, often involving small objects and short-range movements. The automated recognition of such activities can play an important role in real-life applications such as the automated diarization of events, including meetings and training sessions, the verification of compliance to protocols and procedures, and human-robot interaction (Riboni et al., 2015, Sun et al., 2015, Yao et al., 2011).

Typical approaches to fine-grained activity recognition leverage a variety of embedded sensors such as RFID tags, gyroscopes, and accelerometers attached to the body of the agents and to selected, target objects (Stein and McKenna, 2013). These sensors are often complemented by cameras to help with the fine localization of objects and the measurement of gestures and movements (Lei et al., 2012, Rohrbach

et al., 2012). In addition, inexpensive depth cameras such as Play Station Eye and Microsoft Kinect have made depth videos widely available and easily usable for this task.

Despite the progress in this area, the use of multiple sensors poses practical limitations to the applicability of this technology. As a matter of fact, the requirement of equipping people and target objects with borne sensors may prove cumbersome or impractical in many scenarios. The use of borne sensors also makes it possible to identify their carrier by association, which may not be desirable in cases where privacy is paramount.

For these reasons, in this chapter we present an approach to fine-grained activity recognition that solely leverages the use of a depth camera. This approach is applied to a kitchen scenario where only a single camera is placed unobtrusively above the cooking plane, without interfering with the actions and with no additional instrumentation. The dataset used for the experiments is the challenging ”50 Salads” dataset which was released as part of a recent publication to offer a unified and probing benchmark for fine-grained activity recognition from RGB, depth and accelerometric data (Stein and McKenna, 2013). Figure 4.1 shows an example of depth frames from this dataset. The experimental results in Section 3.5 show that the proposed approach is capable of achieving an accuracy comparable to that of a state-of-the-art method that uses both cameras and accelerometers, making it possible to apply fine-grained activity recognition in a wide range of scenarios.

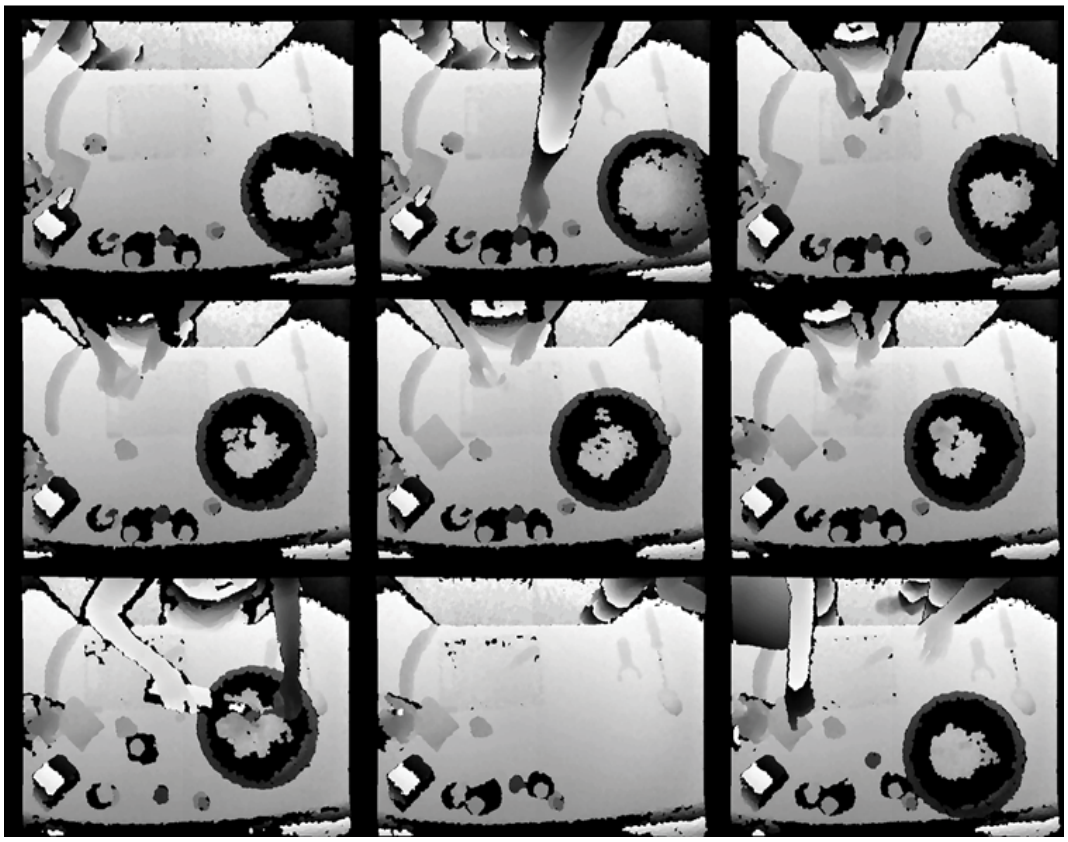


FIGURE 4.1: Examples of depth frames from the ”50 Salad” dataset.

4.3 Proposed Approach

In this chapter, we show how depth videos alone enable an accurate solution to fine-grained activity recognition. This approach is demonstrated in a kitchen environment where various actors attend to the preparation of mixed salads in a spontaneous and realistic way. Figure 4.2 shows an overview of the proposed approach, while the remainder of this section describes the main components.

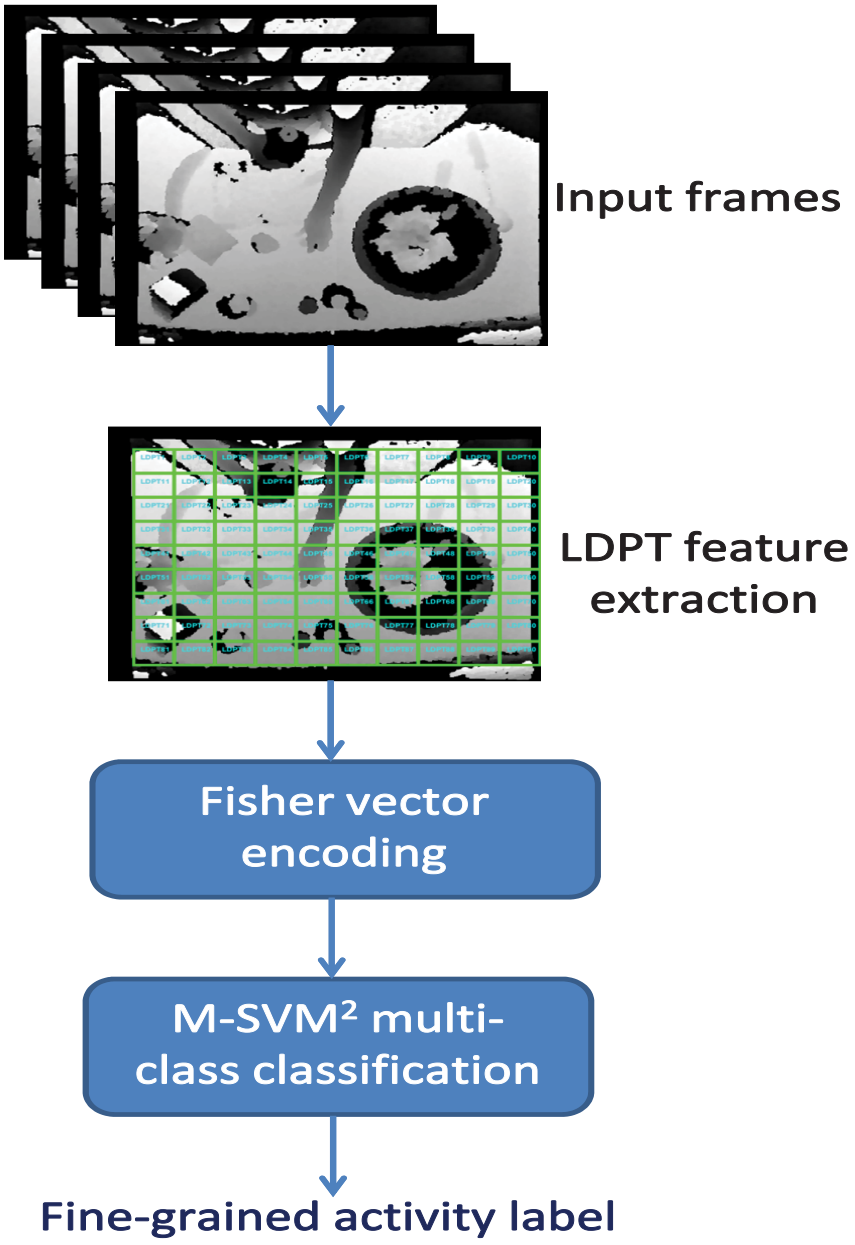


FIGURE 4.2: Overview of the proposed approach.

4.3.1 The Local Depth Feature: LDPT

Local video features have proved versatile over diverse tasks such as activity recognition, detection and tracking. In (Awwad et al., 2015), the authors have proposed

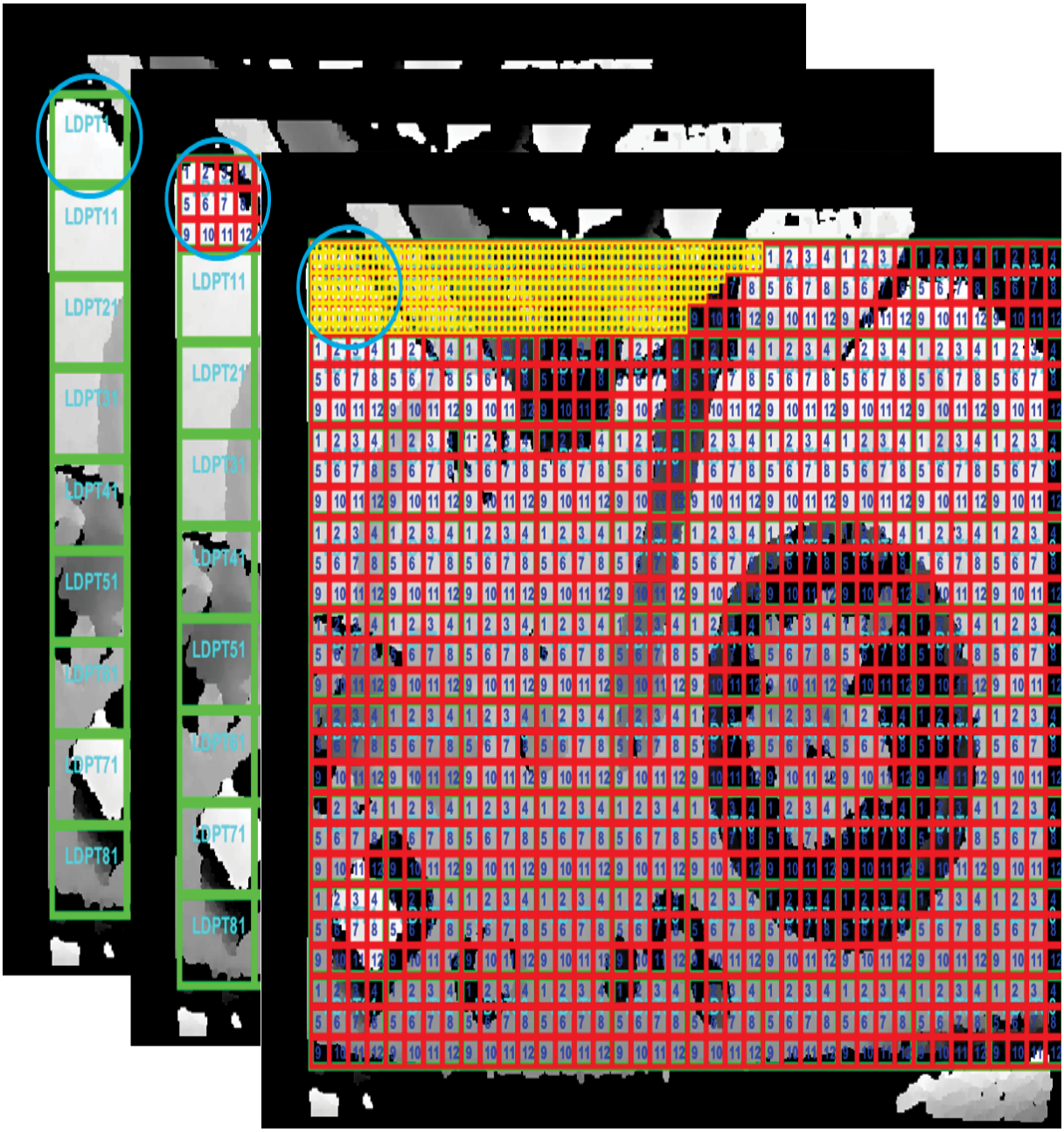


FIGURE 4.3: The hierarchy of cells (smallest), depth patterns (intermediate; numbered from 1 to 12) and LDPTs (largest). This figure should be viewed in color.

a local depth feature for tracking (LDPT) that has ranked highly in a challenging tracking benchmark of depth videos. Since the feature has proved able to represent the target shape under mild deformations and viewpoint changes, we believe that it could also be effective for representing the shape of the objects of interest in

fine-grained activity recognition. For this reason, we choose an appropriate size for the LDPT feature and we partition each depth frame into a grid of non-overlapping LDPTs with H rows and V columns. While the use of overlap between adjacent LDPTs can soften boundary effects, we found it was not beneficial in practice.

The LDPT, in turn, consists of an $HD \times VD$ grid of "depth patterns" (DP) (Pietikäinen et al., 2011) that encapsulate the directional derivatives within a small square patch. Each depth pattern first sub-divides its square patch into 3×3 cells, and then computes the absolute differences between the average depth of every pair, saving these differences in a $\binom{3*3}{2} = 36$-dimensional vector. For clarity, Figure 4.3 shows the hierarchy of the cells, depth patterns and LDPTs.

4.3.2 Feature Encoding

After the completion of the feature extraction stage, the local features of each frame are encoded into a more compact and descriptive representation called an *encoding*. Encodings are a key component of visual recognition algorithms, with the most popular being the bag-of-features, the vector of locally aggregated descriptors (VLAD) and the Fisher vector (Jegou et al., 2010, Li and Perona, 2005, Perronnin et al., 2010). In (Sánchez et al., 2011), the authors have shown that the Fisher vector is especially suitable for the recognition of fine-grained activities, thanks to its ability to retain detailed information.

Given a Gaussian mixture model (GMM) with M diagonal components and parameters $\{w_m, \mu_m, \sigma_m, m = 1 \dots M\}$ (respectively, weight, mean and standard deviation of the m-th component), the Fisher vector encodes a set of local features, $X = \{x_i, i = 1 \dots N\}$, as the gradient of their likelihood in the GMM. The equations for the gradient with respect to the mean and the standard deviation of the k-th component are:

$$G_{\mu_m} = \frac{1}{N\sqrt{w_m}} \sum_{i=1}^{N} p_{im} \left(\frac{x_i - \mu_m}{\sigma} \right) \tag{4.1}$$

$$G_{\sigma_m} = \frac{1}{N\sqrt{w_m}} \sum_{i=1}^{N} p_{im} \left(\frac{(x_i - \mu_m)^2}{\sigma} - 1 \right) \tag{4.2}$$

where p_{im} is the probability of measurement x_i in the m-th component. The Fisher vector is the concatenation of these gradients for all the M components and its dimensionality is equal to $2MD$, where D is the dimensionality of a local feature. Given that this value is typically high, we post-process the vector with principal component analysis to reduce the dimensionality to a range of $[300 - 500]$.

4.3.3 Multi-Class Classification by M-SVM^2

Notwithstanding the use of informative features, classification of fine-grained activities remains a very challenging task due to the typically small inter-class distance between the activities. Therefore, a multi-class classifier capable of discriminating subtle differences between classes is a critical requirement. The support vector machine (SVM) has a strong reputation for high empirical accuracy over multi-class problems (Hsu and Lin, 2002). However, its common binary decompositions are trained separately for each class and are prone to inconsistent predictions.

Conversely, the multi-class SVM proposed by Lee *et al.* in (Lee et al., 2004) is trained using a unified objective for all the classes while guaranteeing useful statistical properties. The main idea is to train a multi-class SVM to assign a score of 1 to the ground-truth class and a score of $-1/(K-1)$ to each of the other $K-1$ classes. The loss function that is derived from this sum-to-0 score is proven to be Fisher consistent, i.e. it tends to Bayes' optimal decision rule as the size of the training set grows. To the best of our knowledge, this is the only multi-class SVM loss which

enjoys this property over the entire parameter space. As a further improvement, Guermeur and Monfrini in (Guermeur and Monfrini, 2011) have suggested using a quadratic form over this loss to upper-bound the leave-one-out cross-validation error.

The resulting classifier - M-SVM^2 - has outperformed a number of other multi-class classifiers over a diverse range of datasets and for this reason we adopt it here (Guermeur and Monfrini, 2011, Lauer and Guermeur, 2011). Given a multi-class training set $x_i, y_i, i = 1 \dots N$, with K classes, the primal problem of M-SVM^2 is given by:

$$\begin{aligned} &\underset{w,b,\xi}{\operatorname{argmin}} \frac{1}{2}\|w\|^2 + C\xi^\top M\xi \\ &s.t., i = 1 \dots N : \\ &\quad w_k^\top x_i + b_k \leq -\frac{1}{K-1} + \xi_{ik}, \forall k \neq y_i \\ &\quad \sum_{k=1}^{K} w_k^\top x_i + b_k = 0 \end{aligned} \tag{4.3}$$

Like in a conventional SVM, (4.3) aims to minimize a trade-off between a regularization term ($\|w\|^2$) and a term accounting for the error over the training set ($\xi^\top M\xi$). Notations in (4.3) are as follows: parameter vector $w, b = \{w_k, b_k\}, k = 1 \dots K$, is the concatenation of the score parameters of each class. Vector $\xi = \{\xi_{ik}\}, i = 1 \dots N, k = 1 \dots K, k \neq y_i$, is the vector of the "slack" variables used to relax the $N(K-1)$ constraints for the satisfiability of the problem. Matrix $M = \{m_{ik,jl} = \delta_{i,j}(\delta_{k,l} + 1)\}$ is a positive semidefinite matrix that computes a quadratic term over the slack variables. The inequality constraints limit the score of classes other than the ground truth to $\leq -1/(K-1)$. As a consequence, the equality constraints make

the score of the true class, y_i, to be greater than or equal to a unit, guaranteeing a proper margin between correct and incorrect classifications.

4.4 Experiments

4.4.1 Dataset

The proposed approach has been evaluated on the challenging ”50 Salads” kitchen activities dataset that was recently released as part of a 2013 publication to offer a benchmark for fine-grained activity recognition from RGB, depth and accelerometer data (Stein and McKenna, 2013). The dataset consists of 50 videos of an individual preparing a salad in a kitchen setting, under the view of a Kinect camera and with several accelerometers attached to utensils. The activities in the ”50 Salads” dataset have been labeled at two different levels of granularity using 17 and 10 different labels, and we follow the latter for direct comparability with (Stein and McKenna, 2013). The ten activities are: *add oil*, *add pepper*, *mix the salad dressing*, *peel a cucumber*, *cut into pieces*, *place into a bowl*, *mix the ingredients*, *serve the salad onto a plate*, *add the salad dressing*, and *null*.

The challenge with this set of classes is not its size, but the fact that all activities only involve small arm movements and small objects, suggesting a very significant class overlap. Figure 4.1 shows the challenging scenario, where all the target objects are present at once and only the actor’s arms are in view. On the other hand, the camera’s position is unobtrusive and does not impinge on the activities. Each activity instance in the dataset is further annotated into three stages: pre-, core- and post-activity. The total size of the dataset is approximately 500 thousand video frames, of which around 250 thousand represent the core stage of activities. Table 4.1 displays the video frame counts for each activity.

TABLE 4.1: Dataset activities and video frame counts

Main Activity	Fine-Grained Activity	# Frames	# Core
prepare a dressing	add oil	24463	7100
	add pepper	11544	5404
	mix dressing	17291	12578
cut and mix ingredients	peel cucumber	57021	35934
	cut into pieces	194600	123836
	place into bowl	53462	27113
	mix ingredients	20525	14138
serve salad	serve salad	31237	16956
	add dressing	19227	9730
	null	62754	N/A
Total		492124	252789

4.4.2 Features extraction and classification

As measurements for the experiments, we first extracted the LDPT features of Section 4.3.1. The size of the cell was set to 5×5 pixels and HD and VD were set to 3 and 4, respectively. This made the total area covered by an LDPT equal to $(5 * 3 * 3 =) 45 \times (5 * 3 * 4 =) 60$ pixels which is appropriate for the typical size of the objects in these frames. The vector dimensionality of an LDPT was therefore $HD * VD * \binom{3*3}{2} = 432$. Each depth frame was then partitioned in a grid of $H = 9$ and $V = 10$ LDPTs, centred in the frame. This resulted in a total covered area of 405 pixels in height and 600 in width which adequately captured all the viewable activities in the scene.

The LDPT features of each frame were then encoded using $M = 16$ components, resulting in a large Fisher vector of $2DM = 2 * 432 * 16 = 13,824$ dimensions. We therefore reduced this dimensionality by PCA to the top 300 principal components. For the classification, we used the M-SVM^2 algorithm from package MSVMpack (Lauer and Guermeur, 2011) with 5-fold cross-validation which returns a realistic estimation of the run-time accuracy. As cross-validation parameters, we used

constant C over range $[1, 10]$ and the linear, polynomial and RBF kernels as the kernel.

4.4.3 State of the Art on the Dataset

The state-of-the-art accuracy on the "50 Salad" dataset is held by the approach presented in (Stein and McKenna, 2013). This approach exploits the RGB and depth videos from a vertical view of the kitchen bench and seven Axivity WAX3 wireless accelerometers attached to the following utensils: a knife, a mixing spoon, a peeler, a small spoon, a glass, an oil bottle and a pepper dispenser. A set of four types of features is computed by combining the visual and accelerometric data:

- *Object Use (OU)*: a binary variable indicating whether the object is accelerating or not, used as a proxy for the object being in use at all (7 variables in total);
- *Acceleration Statistics (AS)*: mean, energy, standard deviation and entropy for each of the three axes (relative to free fall) and estimated pitch and roll (relative to gravity) (20D per object; 140D for all objects);
- *Device Locations (DL)*: accelerometers are localized in the visual field of the camera by matching the measured acceleration of a device with the acceleration estimated along visual point trajectories (2D per object; 14D for all objects);
- *Visual Displacement Statistics (VS)*: mean, energy, standard deviation and entropy for the visual displacement components in x and y (8D per object; 56D for all objects).

4.4.4 Experimental Results and Discussion

Table 4.2 shows the recall, precision and F1 score obtained with the proposed approach for each activity class. Fig. 4.4 displays the corresponding confusion matrix (the complete matrix of ground-truth vs prediction percentages). Table 4.2 shows that there are significant differences in recall and precision between the classes: for instance, class "peel cucumber" reaches an 89.0% recall average, while class "serve salad" only achieves 55.2%. This can be explained by the different extent of class evidence in the depth data, where a repetitive activity such as peeling may prove easier to spot than an isolated action. On the other hand, Table 4.2 shows that the differences in F1 score are far less remarked and that the proposed approach achieves an F1 score above 50% for all classes but one. These results have been obtained with cross-validation parameters $C = 5$ and the linear kernel.

Table 4.3 compares the results from the proposed approach with the original results of the dataset's authors and a popular, standard SVM baseline (libsvm (Chang and Lin, 2011)) in terms of recall and precision. The table shows that the proposed approach outperforms various combinations of visual and sensor features from (Stein and McKenna, 2013), and achieves a recall higher than all of them (offset by a lower precision). The recall improvement over the best combination of visual and sensor features is 6 percentage points, while the decrease in precision is 8 percentage points, making the results roughly equivalent and supporting our main claim that our approach, based solely on a depth camera, achieves approximately the same results as an approach using a depth/RGB camera and accelerometers on every target object. Another important remark about this comparison is that the results of (Stein and McKenna, 2013) were obtained using different cross-validation parameters for each fold, whereas we only use one setting for all folds. While our choice may slightly penalize our reportable test accuracy, it is more realistic since a run-time system is only allowed one setting. Table 4.3 also shows that the adoption of the

recent M-SVM^2 algorithm allows us to achieve a marked improvement over the standard SVM baseline.

TABLE 4.2: Recall, precision and F1 score for each activity class with the proposed approach.

Class Label	**Recall %**	**Precision %**	**F1 score %**
add oil	74.0 ± 16.0	47.5 ± 2.2	57.1 ± 6.0
add pepper	88.3 ± 3.4	57.4 ± 5.9	63.6 ± 5.6
mix dressing	85.9 ± 5.1	51.0 ± 4.8	64.0 ± 5.1
peel cucumber	89.0 ± 2.7	49.4 ± 8.2	63.2 ± 6.8
cut into pieces	74.1 ± 5.5	68.7 ± 5.7	71.2 ± 5.2
place into bowl	69.8 ± 11.5	48.6 ± 7.7	57.2 ± 8.8
mix ingredients	66.4 ± 10.4	66.9 ± 9.4	65.9 ± 4.4
serve salad	55.2 ± 9.2	59.1 ± 13.5	56.0 ± 6.7
add dressing	80.7 ± 16.8	58.0 ± 5.3	66.6 ± 6.1
null	50.7 ± 6.2	82.6 ± 1.6	62.6 ± 4.8

TABLE 4.3: Comparison of recognition performance.

Feature Type	**Recall %**	**Precision %**
OU + DL (Stein and McKenna, 2013)	51 ± 3	51 ± 2
OU + VS (Stein and McKenna, 2013)	54 ± 2	53 ± 4
DL + VS (Stein and McKenna, 2013)	57 ± 4	54 ± 3
DL + AS (Stein and McKenna, 2013)	61 ± 5	64 ± 3
OU + AS (Stein and McKenna, 2013)	63 ± 5	66 ± 3
AS + VS (Stein and McKenna, 2013)	67 ± 5	67 ± 3
OU + AS + VS (Stein and McKenna, 2013)	67 ± 5	68 ± 3
libsvm	68 ± 4	57 ± 5
Our Approach	$\mathbf{73 \pm 4}$	$\mathbf{56 \pm 1}$

Finally, for internal comparison, Table 4.4 shows the accuracy improvement achieved by applying PCA to the Fisher vectors. The recall and precision proved much higher

than not using PCA (by 27 and 13 percentage points, respectively). This is in accordance with the results of (Sánchez et al., 2011) that had shown that Fisher vectors are highly compressible. In initial experiments, we had also compared this with the popular VLAD and bag-of-words encodings, but we had achieved much lower accuracies, both with and without PCA.

TABLE 4.4: Recall and precision for the proposed method with and without PCA.

Feature Type	Recall %	Precision %
Proposed approach without PCA	46 ± 2	43 ± 2
Proposed approach + PCA	$\mathbf{73 \pm 4}$	$\mathbf{56 \pm 1}$

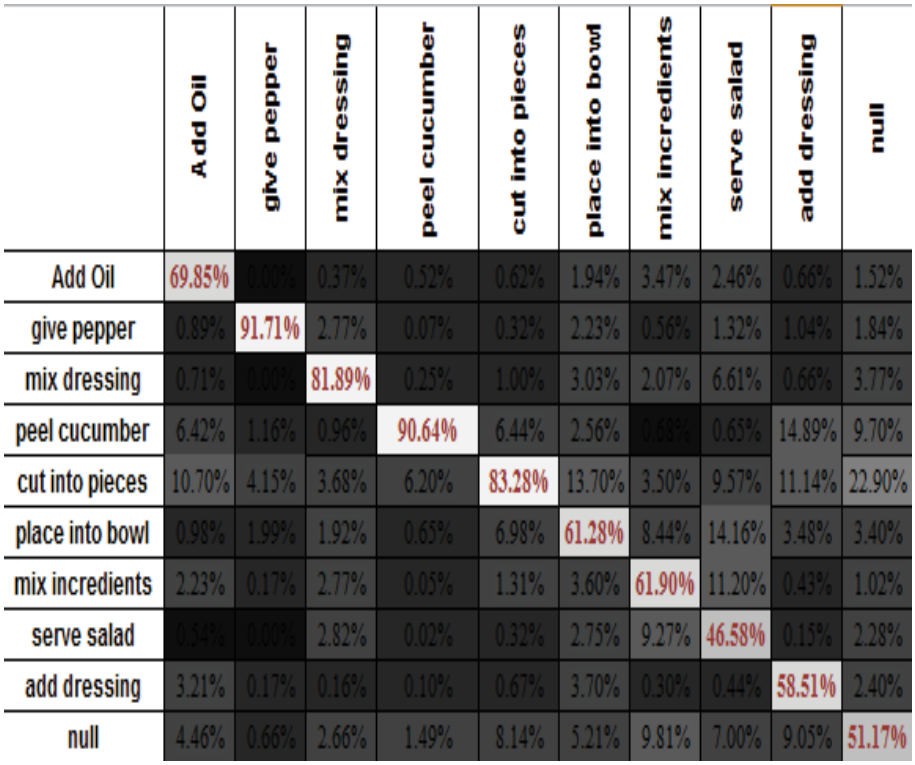


FIGURE 4.4: Confusion matrix for the proposed method. Rows and columns represent ground-truth and predicted class labels, respectively. Numbers represent frequencies in percentages and the cells' gray-levels visually encode the frequencies from 0% = black to 100% = white.

Chapter 5

Automated Hand Hygiene Detection

After having explored activity recognition in the previous chapter using a benchmark kitchen activity dataset, in this chapter we tackle a real-life problem of automated hand hygiene detection motivated by a collaboration with clinical researchers from Sydney's Royal Prince Alfred Hospital.

5.1 Introduction and background

Healthcare associated infections (HAI) are an important cause of morbidity and mortality in healthcare facilities; 5 -15% of patients admitted to hospital in developed countries will acquire an HAI (Spelman, 2002). The problem is even greater (9 - 37% of admissions) in high-risk environments such as intensive care units. HAIs affect almost 200000 patients in Australian healthcare facilities and result in approximately 2 million extra hospital bed days annually. Pathogens can be transmitted to susceptible patients by the hands of healthcare workers. Inadequate hand hygiene among healthcare workers was identified as an important cause of HAI by Ignaz Semmelweis in 18461 and remains a problem today (Vincent, 2003).

Properly performed hand hygiene effectively reduces HAI. Current World Health Organisation (WHO) and Hand Hygiene Australia guidelines describe the 5 moments of hand hygiene that must be performed. Unfortunately, compliance rates with hand hygiene are frequently low. Hand hygiene compliance rates in Australia across 860 hospitals were estimated to be 82.8% in June 2015. Low compliance rates are widespread, and vary between 5 % and 81% globally (Organisation, 2009, Sax et al., 2007).

Surveillance of hand hygiene and the collection of quality assurance data is difficult; an ideal method is not available. Direct observation of the 5 moments is currently the most common method for auditing hand hygiene compliance. The five moments are identified as: I) before touching a patient; II) before a procedure; III) after a procedure or body fluid exposure risk; IV) after touching a patient; and V) after touching a patient's surroundings, These moments are displayed in Figure 5.1 . Policies of hand hygiene education and adequate work flows have been put in place in hospitals world-wide to encourage adherence to the five moments. The WHO Hand Hygiene technical reference manual recommends observing a minimum of 200 opportunities per observation period and per unit of observation (eg. a single ward area) to reliably compare results before and after hand hygiene improvement interventions (EFORE, 2009). Direct observation has major limitations - it is expensive, laborious, and prone to bias (Organisation, 2009). It is subject to an observation bias (Hawthorne effect) where healthcare workers change their behavior whilst being audited, as well as observation and selection biases. Periods of audit are extremely short compared to the breadth of usual clinical care, resulting in gross under sampling. Bias and under sampling are threats to the accuracy of hand hygiene data, and its validity as a performance indicator (Organisation, 2009, Sax et al., 2007).

Computer vision is a branch of artificial intelligence that studies how to automatically understand the content of images and video in a human-like manner (Shah,

1997). While computer vision is well established in the area of medical imaging (medical image computing) (Handels et al., 2012), it is used extremely rarely in clinical medicine where patient (and healthcare worker) privacy is an utmost concern (Palmore and Henderson, 2011). However, concerns about the use of video surveillance in privacy-sensitive environments have been recently mollified by the introduction of $depthimages$. Unlike video (RGB) images, depth (or range) images only record the distance of the objects from the camera and do not permit identification of the viewed subjects or to distinguish features beyond outlines. Depth image cameras have become cheap and widely available. We decided to investigate whether computer vision and depth image cameras could be used to surveil hand hygiene in a way that was both clinical feasible and privacy protecting.

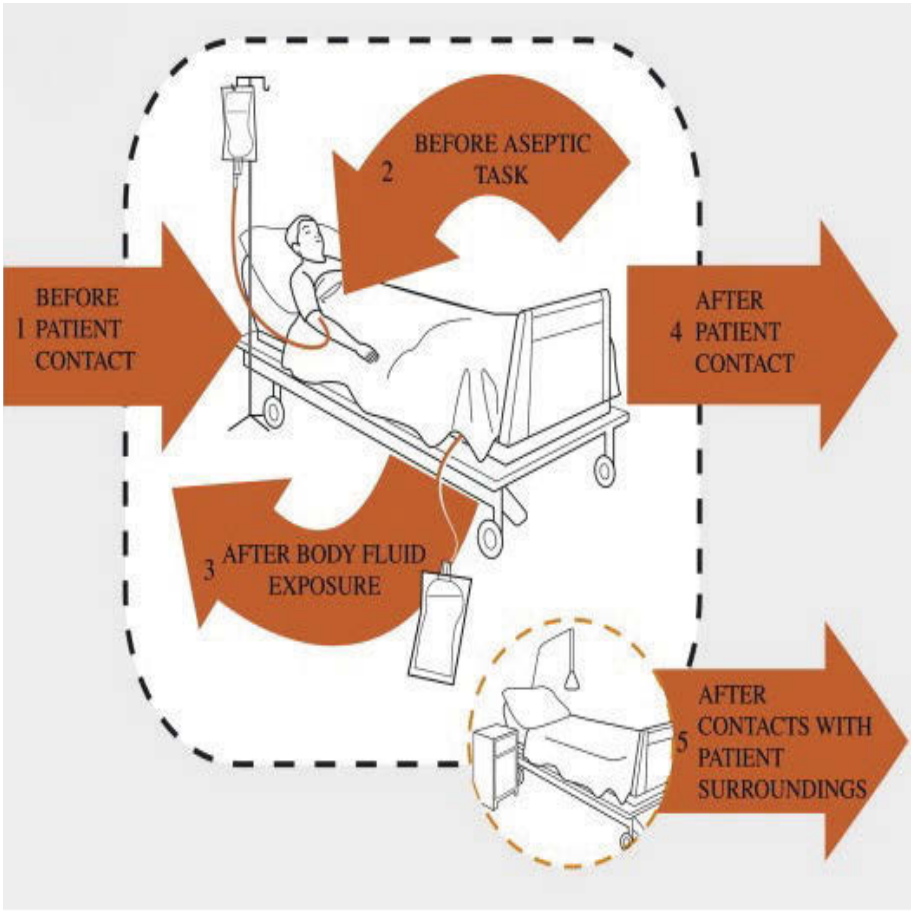


FIGURE 5.1: The Five Moments of Hand Hygiene

5.1.1 Study Objectives

This work aims to the following objectives:

1. To demonstrate the feasibility of automated, direct observation and collection of hand hygiene data.
2. To develop computer visual methods capable of reporting compliance with moment 1 (the performance of hand hygiene before touching a patient).
3. To report the diagnostic accuracy of automated, direct observation of moment 1.

5.2 Methods

5.2.1 Simulation of the clinical environment and the first moment of hand hygiene

We simulated a hospital bed space in a laboratory at the University of Technology, Sydney. Four volunteers performed the roles of patient and healthcare worker, acting as patient and healthcare worker in turn. The camera was placed above the patients head and pointed toward the foot of the bed. Alcohol-based hand rub was placed on a pedestal at the foot of the bed, in the centre of the cameras frame of view. When the patient was supine, the top of their head was visible to the camera; their face was not. Healthcare workers approached the patient on the bed, with the interaction ending with usual physical examination contact with the patient. Clinically realistic approaches by healthcare workers to the bedside were simulated - this included various combinations with/without dispensing of hand rub, and with/without rubbing of the hands together.

5.2.2 Capture and processing of RGB and depth images

The distances from the camera to the bottle and from the camera to the bed, were fixed and measured. We used a Kinect camera (Microsoft Corp) to capture range imagedepth images along with RGB images. This provides for accurate volumetric scene reconstruction, object tracking and disambiguation of occlusions14. Depth images are formed by projecting dots on the scene in the near infrared spectrum and triangulating their distance.

To capture the images, we have used a commercial software, nuiCapture version 1.4.0. Its main advantages are that it synchronously records the depth and RGB images and it automatically extracts the skeleton and face of the tracked subjects from multiple Kinect cameras. It also provides visualisation of the recorded data in a built-in 3D media player and allows exporting the recorded data as video, BVH, FBX, Matlab, OpenEXR, and other formats. We have used the Matlab format since it provides access to the all the data in a numerical format that it is immediately suitable for processing.

5.2.3 Maintenance of privacy during development of the image analysis

To automatically detect the hand hygiene events, we only need to acquire depth images and small RGB patches centred on the hand rub bottle. Since the bottle is placed at waist height, the RGB patches only depict the hands or waist of passers-by and do not allow their identification from facial traits. As evidence of that, Figure 5.3 shows a screen shot from our dataset with a person approaching the hand rub bottle. It is worth noting that wherever the acquisition of such RGB patches could be perceived as a threat to privacy, they could be disposed of altogether and replaced

by the hands information provided by the skeleton-like figure extracted by the acquisition software (Figure 5.2). The skeleton describes the position of the 20 main points of the subject, including the hands, and is extracted solely from the depth frames.

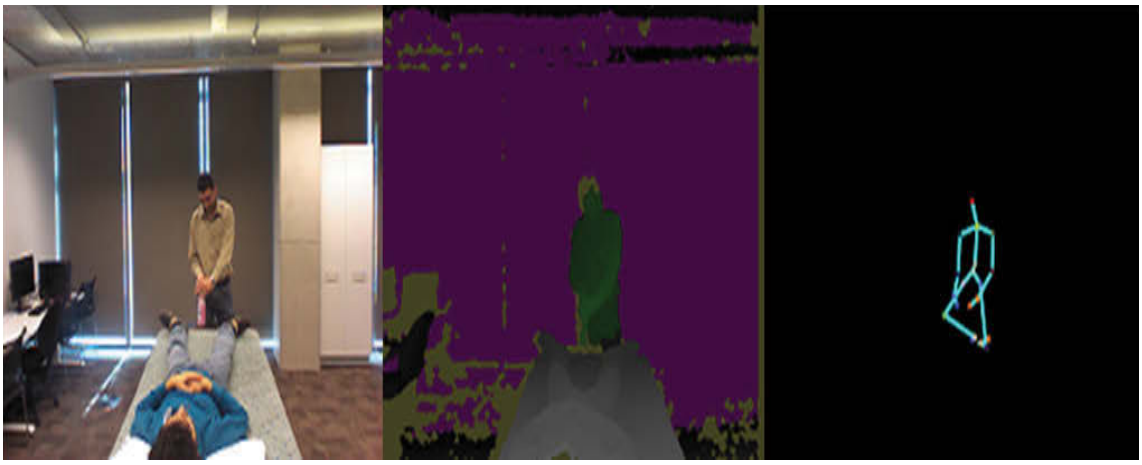


FIGURE 5.2: An example of acquisition software that includes depth, RGB, and skeleton

FIGURE 5.3: An example of the images that were used in this work. For the sake of visualization, the small RGB patch is superimposed to the bottle area

5.2.4 Outcome measures and diagnostic accuracy

The proposed procedure requires learning optimal operating parameters from a set of manually-annotated data. This stage is commonly referred to as learning or training stage. After training, the procedure can be tested and its accuracy estimated. For such an estimate to be realistic, it is mandatory to use a fresh set of data that have not been used for the training phase, to reflect a real situation of use where new clinicians and new patients are monitored. Otherwise, testing over the training data will lead to overly-optimistic estimates of the accuracy. In addition, this training/test set protocol should be repeated several times and results averaged in order to marginalise the impact of the data set as a random variable in the experiment (Dietterich, 1998).

For this reason, our experiments have been carried out following an n-fold cross validation protocol. The data set was divided into three subsets, A, B, and C, and in each experiment, we have used two joined for training and the third one for testing. This process was repeated three times and the accuracy averaged.

The gold standard for compliance with moment 1 was observation of the RGB images by study personnel. We developed automated computer visual methods to detect 3 events necessary to determine compliance with moment 1: (1A) dispensing of hand rub by the healthcare worker, (1B) rubbing together of hands by the healthcare worker, and (2) touching of patient. For each of these three events, we measured true positive (TP), false negative (FN), true negative (TN) and false positive (FP) detections. Compliance with moment 1 was defined as the complete performance of events 1A, 1B, and 2 in the correct order. Violation of moment 1 is defined as the performance of event 2 without preceding performance of events 1A and 1B in correct order, or at all.

5.3 Hand Hygiene approach and experiments

5.3.1 Dataset

A total of 26 videos (both depth and colour frames) were acquired. The simulated healthcare worker correctly complied with moment 1 in 18 videos (positive samples), and failed to do so in 8 videos (negative samples).Figure 1 shows typical RGB and depth images during the development of our simulation.

5.3.2 Hand Hygiene Events

Compliance with moment 1 by the healthcare worker comprised two events, which must be carried out in the correct order. Event 1 was the performance of hand hygiene using alcohol-based hand rub at the foot of the bed. This was subdivided into event 1A (dispensing of hand rub) and event 1B (rubbing of both hands together (vigorously and for a minimum amount of time). Event 2 was the subsequent touching of the patient. Event 2, when not preceded by Event 1 was considered to be a violation of Moment 1. The computer vision techniques for the detection of the events are shown in Figures 5.4 and 5.5 and describes below.

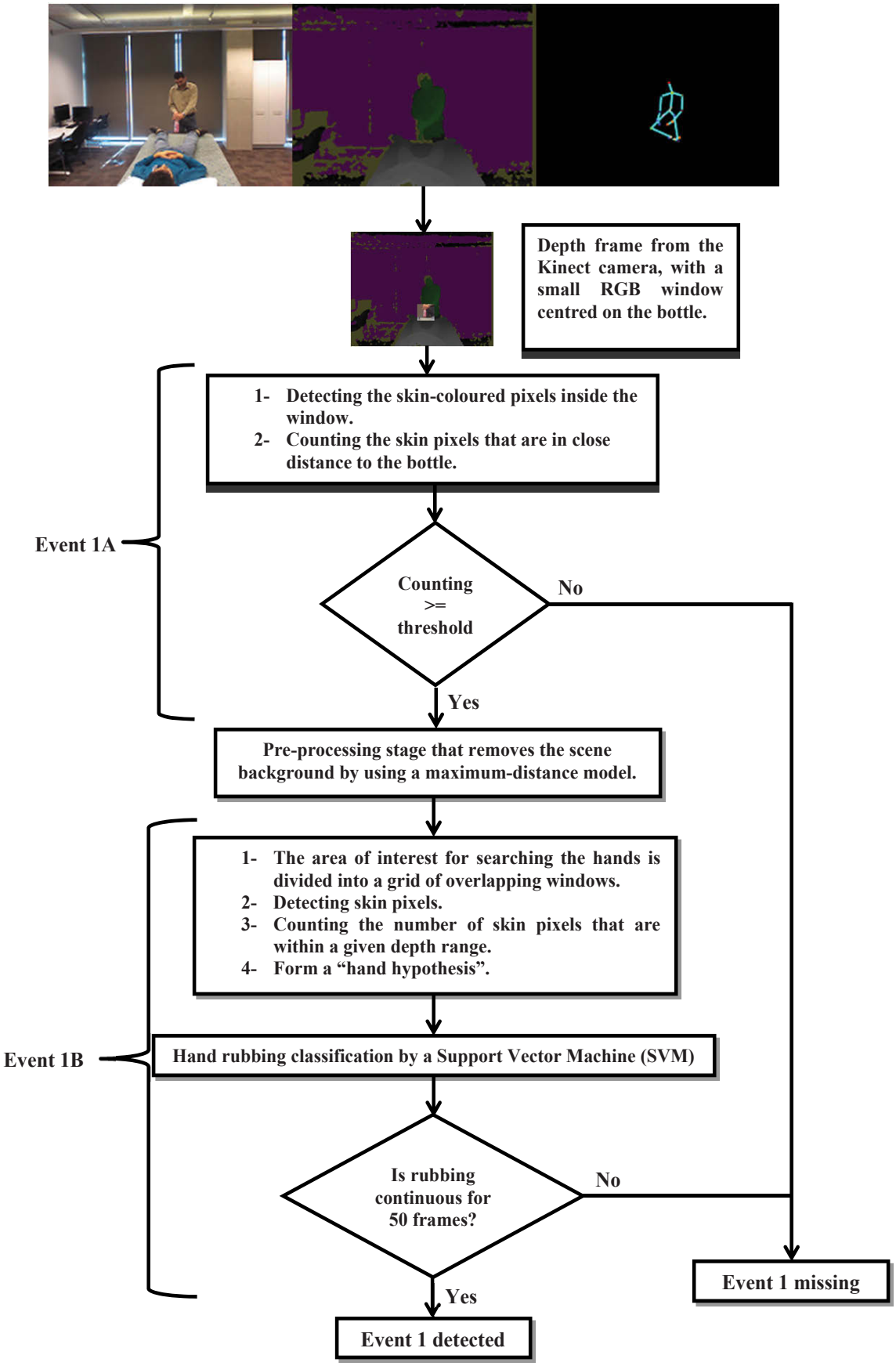


FIGURE 5.4: Event 1 of Hand Hygiene detection approach

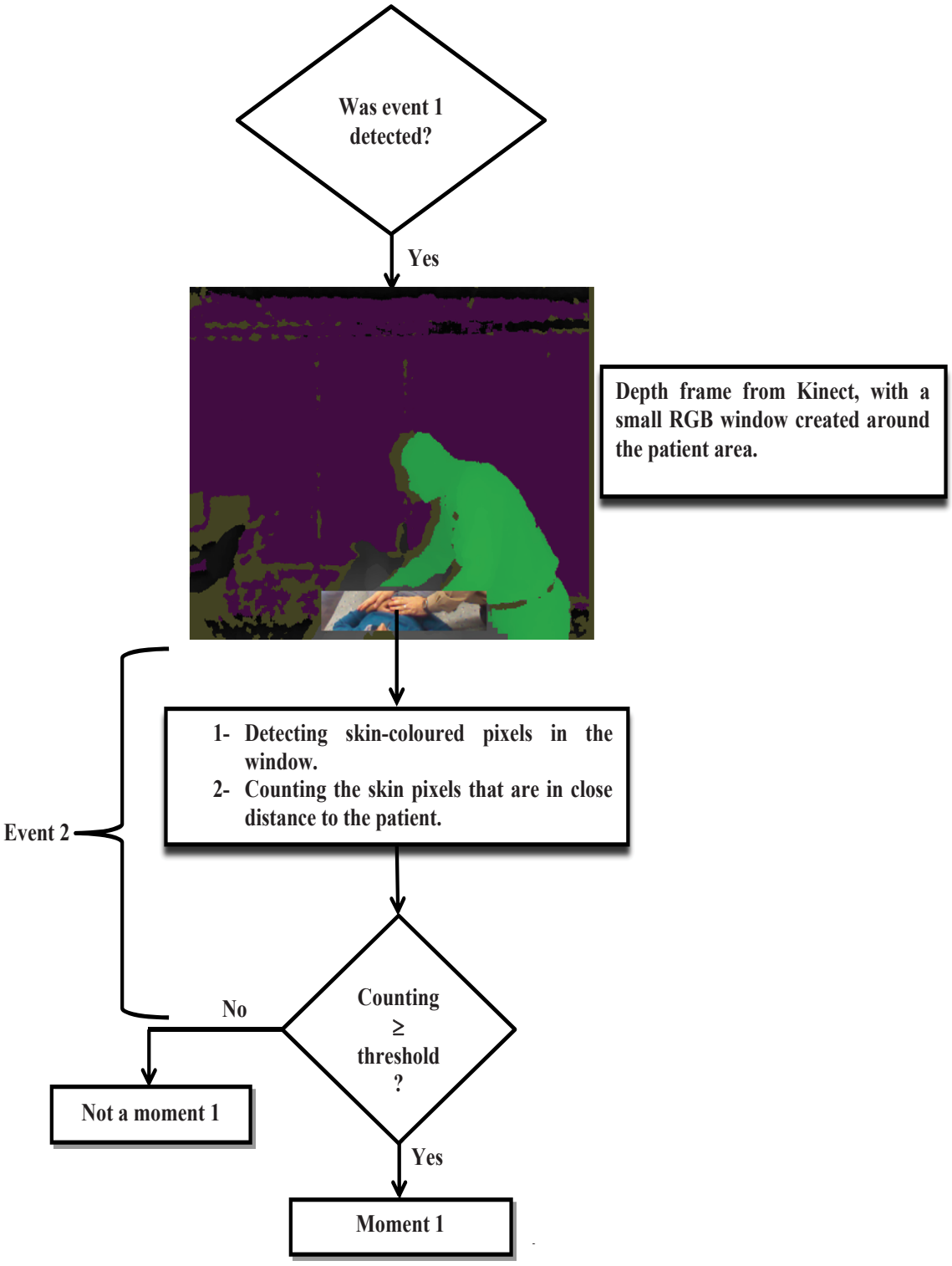


FIGURE 5.5: Event 2 of Hand Hygiene detection approach

5.3.3 Computer vision techniques for detection of dispensing alcohol-based hand rub (Event 1A)

In each frame, we selected a window of pixels centred on the handrub bottle. Dispensing of handrub was inferred if a hand remained in contact with the bottle for a minimum duration (set to 10 frames). Detection consisted of: (i) skin segmentation (detection of the presence of skin-coloured pixels in the pixel window), (ii) counting of skin pixels in close proximity to the hand rub bottle, and (iii) declaring detection if the pixel count was above a given threshold and persisted for a minimum of 10 frames.

5.3.4 Computer vision techniques for detection hand rubbing (Event 1B)

This followed only if Event 1A was detected. This detection included (i) detection and removal of the static background scene to highlight the subjects. Detection of the background scene was achieved by running a temporal filter that returned the maximum depth recorded at each pixel location over a period of time (assuming that the background scene would be in view at some point in time), please see Figure 5.6; (ii) the area of interest (hands) is divided into a grid of overlapping windows, and (iii) pixels are selected in each window if they are a) within a given depth range, b) they are segmented as skin and c) they change depth value over time (i.e., are moving objects).

A hand hypothesis was then formed if the number of selected pixels was above a threshold. When a hand hypothesis was detected, we used a machine learning classifier (a support vector machine) to detect the rubbing of hands. The classifier was trained with 600 manually-annotated images, half depicting hand rubbing and half,

still hands. Hand rubbing was declared if its occurrence was detected continuously for at least 50 frames.

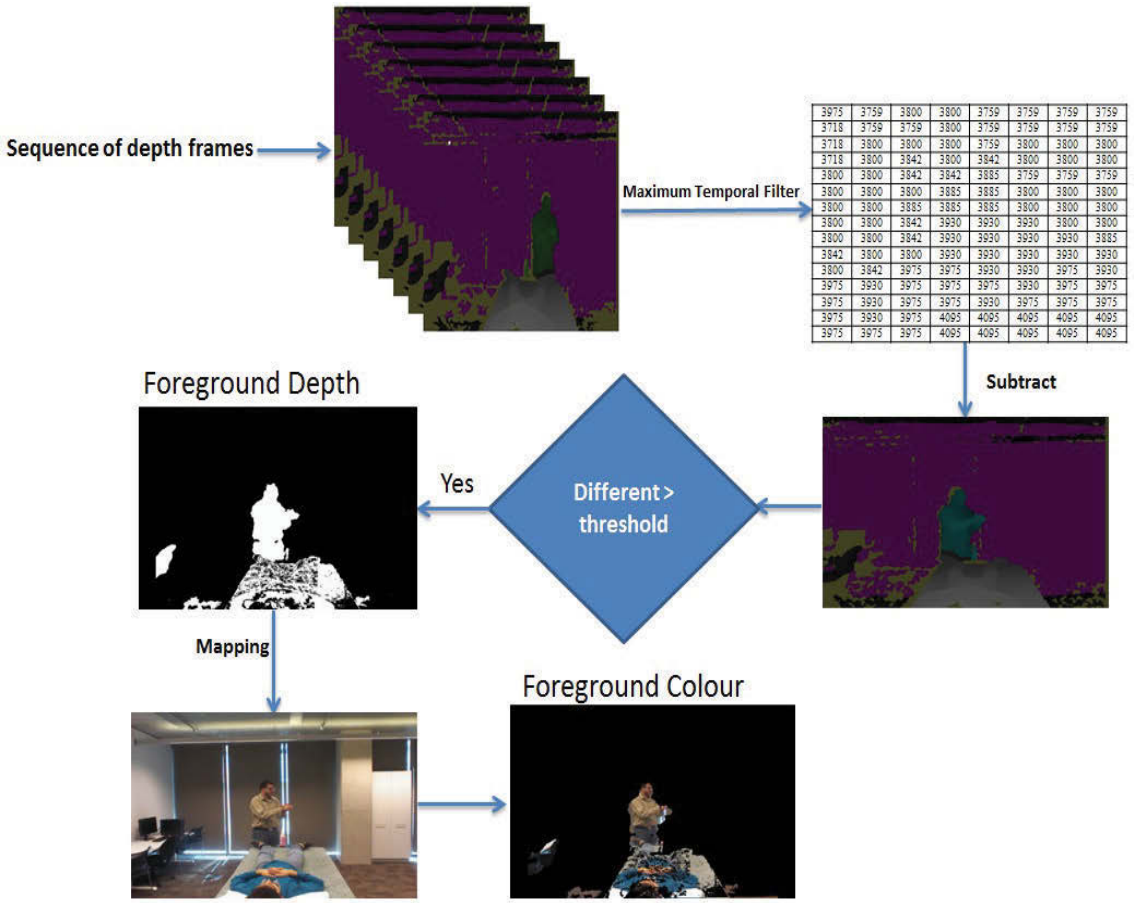


FIGURE 5.6: Background removal procedure

5.3.5 Computer vision techniques for detection of touching the patient (Event 2)

This was similar to the method used for Event 1A (dispensing hand rub). The area of interest around the bed/patient was selected, and detection of skin pixels above threshold was used as a proxy for the detection of bed/patient contact by the health-care worker's hands.

An example of a privacy protecting imagery demonstrating compliance with the first moment of hand hygiene is shown in Figure 5.5. Where the small RGB patches could be seen as a potential threat to privacy, location of the hands of the healthcare worker can also be extracted using the ”skeletal models” provided by the Microsoft Kinect software, dispensing with the need for RGB image capture altogether.

5.3.6 Hand Hygiene Detection Experimental results

Experiments have been carried out following an n-fold cross validation procedure. With this procedure, the data are divided into various ”training” and ”test” sets, and measurements of accuracy are repeated and averaged. All the parameters in the detectors were chosen based solely on the data in the training set, while the accuracy was measured ”blindly” on the test set without any further tuning of the parameters. In this way, the experimental accuracy proves a realistic estimate of the actual accuracy. Table 5.1 reports the achieved accuracy in terms of true positives (TP), false negatives (FN), true negatives (TN) and false positives (FP).

TABLE 5.1: Accuracy of Moment 1 monitoring.

Event Type	**Sub-Event**	**Positive**		**Negative**	
-	**-**	**TP**	**FN**	**TN**	**FP**
Hand Wash	Handrub Pumping	100.0%	0%	100.0%	0.0%
	Hands Rubbing	83.3%	16.7%	87.5%	12.5%
Patient Touching	-	100.0%	0%	0%	100%

Overall, the experiment should be regarded as substantially successful, with 100% correctly detected actions of handrub pumping and patient touching, and a high percentage of correctly detected hands rubbings with only a small percentage of

false positives. Although these results are achieved in a simulated scenario, the set-up is comparable to that of actual hospital rooms and we expect the in-field accuracy to prove similar.

5.4 Discussion

We have demonstrated the feasibility of auditing hand hygiene using depth imagery and computer vision. Our methods were excellent at detecting the dispensing of hand rub and subsequent manual contact of the patient by the healthcare worker (100% detection). Detection occurred in real time and without the need for video (RGB) images, which pose large privacy concerns in clinical care. We used widely available, affordable consumer technology (a Microsoft Kinect camera).

Our findings are significant because HAI and inadequate hand hygiene are a very important public health problem, and the existing strategies for measuring it and managing it are lacking. The bias, under sampling and cost problems of direct observation by human auditors could all potentially be improved by an objective, continuous and inexpensive electronic method such as the one we have described. There is a large Hawthorne effect of auditing on hand hygiene compliance. This can decrease the validity of performance indicator data, but is good for hand hygiene when auditing is going on. Auditing of hand hygiene may be an effective therapeutic intervention for HAI if it can be applied for long enough. We think automated electronic methods are the only way to achieve this.

Technological approaches to improving hand hygiene have been employed before. Remote video auditing with feedback is effective but is unlikely to be feasible or affordable on a large scale. Electronic devices can improve training, but are not always effective at improving compliance. Other methods involving sensors on hand rub dispensers, health care workers or both are also relatively expensive and require

special equipment. Our methods do not require special equipment, do not require transmitters or sensors to be applied in the bed area, and are readily deployable anywhere (a single depth image camera is mounted above the head of the bed).

In spite of the potential for this approach, our study had important limitations. The clinical setting was simulated and highly controlled: a single healthcare worker approached a supine patient, and used alcohol-based hand rub that was positioned in an elevated position at the foot of the bed. Real clinical care is relatively chaotic, and we have not evaluated these methods in that environment. Our methods were not as accurate at detecting the rubbing together of hands by the healthcare worker (83% true positive rate). Skin segmentation relies on skin coloured pixel detection and is reasonably accurate, but untested in clinical areas where non-skin coloured gloves are frequently worn. We do not know how our methods would perform with multiple healthcare workers in the same area, or with other moments of hand hygiene detection.

We have avoided the substantial ethical and privacy concerns that would arise if electronic surveillance measures were deployed in clinical areas by conducting this work in a laboratory simulation. These concerns would be insurmountable if our methods required the capture (and especially storage) of video (RGB) images. By excluding the patient's face from the field of view, and the exclusive use of non-identifying depth imagery, we believe our methods provide a substantial level of inherent privacy protection. Further development and deployment in clinical areas would need to be conducted with great care and sensitivity.

In conclusion, the potential for clinical application is significant. No video imagery needs to be stored (or even captured). The equipment needed is widely available and can be deployed anywhere. It could be paired with real-time feedback to healthcare workers by the use of something like a traffic light to permit touching of the patient. It could generate continuous auditing data for use by managers in real time,

or provide aggregate reports whilst avoiding identification or video surveillance of staff. The next logical step would be to evaluate these methods in a real clinical area using volunteers instead of patients. The technology should only be applied widely or outside of a research setting if it is known to reduce HAI, raises no significant privacy concerns, is affordable, and robust.

Chapter 6

Conclusion and Future Work

In this thesis, I have described the PhD research I have conducted over the past three years and the results I have achieved to date. I have focussed on computer vision from depth videos and my main contributions have been 1) a novel feature for effective tracking of people, 2) its application to fine-grained activity recognition, and 3) a novel system for hand hygiene detection in hospital environments.

In addition, I have proposed a set of extensions for the popular Struck tracker to improve its tracking performance on depth videos. The extensions include a dedicated depth feature based on local depth patterns, a heuristic for handling occlusions in depth frames, and a technique for maintaining the number of support vectors within a given budget to limit computational costs. The feature, called local depth pattern for tracking (LDPT), suitably extends a recently-proposed feature for activity recognition from depth data.

On the automated monitoring of fine-grained human activities such as holding a small object, chopping food, stirring, mixing, pouring and the like, I have proposed a pipeline that achieves remarkable accuracy. Monitoring such activities can empower applications such as "smart" living environments and advanced human-robot

interfaces. Unlike other approaches that employ a number of sensors for this task, the proposed approach only requires a common, commercial depth camera. This work leverages an aggregated depth descriptor that effectively captures the shape of the objects and the actors.

Eventually, I have endeavoured to tackle a real-life problem motivated by a collaboration with clinical researchers from Sydney's Royal Prince Alfred Hospital. Together with my tean, I have designed a novel system to detect compliance with the so-called "Moment 1" of hand hygiene (the hand sanitisation before touching a patient).

We believe that, in its entirety, our work has led to valuable conclusions:

- In tracking from depth videos, the lack of appearance information has not proved a major impediment to the achievement of an interesting tracking accuracy. Rather, in the experiments, tracking from depth data has outperformed tracking from RGB data at a parity of targets and scene (Table 3.1);
- The proposed tracker has achieved a higher accuracy than existing results on depth data in 7 categories of the benchmark out of 11, and on average (Table 3.1);
- The proposed extensions have lead to an average improvement of 9 percentage points over Struck with the best feature (Table 3.1). Amongst the various prototype selection methods and distances, centre-based selection and the distance weighted by the square root of the vectors' weights have reported the best accuracy (Table 3.2).
- In fine-grained activity recognition, the experimental results over a probing dataset of kitchen activities have shown that the proposed approach is capable of providing accuracy comparable to that of a state-of-the-art approach that

uses a combination of depth/RGB video and accelerometers. We believe that these results pave the way for less intrusive and more pervasive implementations of fine-grained activity monitoring.

- Although still in a prototype stage, the hand hygiene detection system has proved highly effective. In addition, the sample frames displayed in Fig. 4.1 give visual evidence that depth data are very privacy-preserving and can mollify concerns in relation to the adoption of fine-grained activity classification in a variety of environments, including privacy-sensitive organizations such as hospitals and aged care facilities.

Although these results advance the state of the art on these topics, we believe that there are still significant unresolved challenges ahead. Should we be given the opportunity to continue this work, two natural prolongments would be extension of the proposed hand-hygiene detection system to detect the compliance with all the Moments of hand hygiene (1 to 5), and the investigation of *cooperative* tracking and action recognition from depth videos, given that these two tasks have indeed the potential to behave synergistically.

Bibliography

Ahad, M. A. R., Tan, J. K., Kim, H., and Ishikawa, S. Motion history image: its variants and applications. *Machine Vision and Applications*, 23(2):255–281, 2012.

Aldo, Z., Peter, S., Robert, R., Spencer, A., Tony, M., Brian, C., and David, W. TUG. `http://www.aethon.com/tug/benefits/`, 2015. [Online; accessed 9 March 2015].

Alexiadis, D. S., Zarpalas, D., and Daras, P. Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras. *IEEE Transactions on Multimedia*, 15(2):339–358, 2013.

Atmosukarto, I., Ghanem, B., and Ahuja, N. Trajectory-based fisher kernel representation for action recognition in videos. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3333–3336. IEEE, 2012.

Avidan, S. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007.

Awwad, S., Hussein, F., and Piccardi, M. Local depth patterns for tracking in depth videos. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, pages 1115–1118. ACM, 2015.

Ayers, D. and Shah, M. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing*, 19(12):833–846, 2001.

Babenko, B., Ming-Hsuan Yang, and Belongie, S. Visual tracking with online Multiple Instance Learning. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pages 983–990, 2009.

Basso, F., Munaro, M., Michieletto, S., Pagello, E., and Menegatti, E. Fast and robust multi-people tracking from rgb-d data for a mobile robot. In *Intelligent Autonomous Systems 12*, pages 265–276. Springer, 2013.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

Bordes, A., Bottou, L., Gallinari, P., and Weston, J. Solving multiclass support vector machines with larank. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 89–96, 2007.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., and Belongie, S. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010*, pages 438–451. Springer, 2010.

Bregler, C. Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 568–574. IEEE, 1997.

Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, Sept 2011.

Briechle, K. and Hanebeck, U. D. Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 95–102. International Society for Optics and Photonics, 2001.

Cavallanti, G., Cesa-Bianchi, N., and Gentile, C. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 69(2-3):143–167, 2007.

Chai, Y., Lempitsky, V., and Zisserman, A. Symbiotic segmentation and part localization for fine-grained categorization. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 321–328. IEEE, 2013.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Chatfield, K., Lempitsky, V. S., Vedaldi, A., and Zisserman, A. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, volume 2, page 8, 2011.

Chen, L. and Khalil, I. Activity recognition: Approaches, practices and trends. In *Activity Recognition in Pervasive Intelligent Environments*, pages 1–31. Springer, 2011.

Collins, R. T. Mean-shift blob tracking through scale space. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–234. IEEE, 2003.

Comaniciu, D. and Ramesh, V. Real-time tracking of non-rigid objects using mean shift, July 8 2003. US Patent 6,590,999.

Comaniciu, D., Ramesh, V., and Meer, P. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

Danafar, S. and Gheissari, N. Action recognition for surveillance applications using optic flow and svm. In *Asian Conference on Computer Vision*, pages 457–466. Springer, 2007.

de Campos. A survey on computer vision tools for action recognition, crowd surveillance and suspect retrieval. 2014.

De Campos, T., Barnard, M., Mikolajczyk, K., Kittler, J., Yan, F., Christmas, W., and Windridge, D. An evaluation of bags-of-words and spatio-temporal shapes for action recognition. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 344–351. IEEE, 2011.

Dekel, O., Shalev-Shwartz, S., and Singer, Y. The forgetron: A kernel-based perceptron on a budget. *SIAM Journal on Computing*, 37(5):1342–1372, 2008.

Deng, C., Cao, X., Liu, H., and Chen, J. A global spatio-temporal representation for action recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1816–1819. IEEE, 2010.

Dietterich, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

Donoser, M. and Bischof, H. Efficient maximally stable extremal region (mser) tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 553–560. IEEE, 2006.

Duan, K.-B. and Keerthi, S. S. Which is the best multiclass svm method? an empirical study. In *International Workshop on Multiple Classifier Systems*, pages 278–285. Springer, 2005.

EFORE, B. Hand hygiene technical reference manual. 2009.

Fanello, S. R., Gori, I., Metta, G., and Odone, F. Keep it simple and sparse: real-time action recognition. *Journal of Machine Learning Research*, 14(1):2617–2640, 2013.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

Gao, J., Kosaka, A., and Kak, A. C. A multi-kalman filtering approach for video tracking of human-delineated objects in cluttered environments. *Computer Vision and Image Understanding*, 99(1):1–57, 2005.

Gavrila, D. M. Pedestrian detection from a moving vehicle. In *Computer VisionECCV 2000*, pages 37–49. Springer, 2000.

Gordon, N. J., Salmond, D. J., and Smith, A. F. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113. IET, 1993.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247–2253, 2007.

Grabner, H., Leistner, C., and Bischof, H. Semi-supervised on-line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 234–247, 2008.

Guermeur, Y. and Monfrini, E. A quadratic loss multi-class SVM for which a radius-margin bound applies. *Informatica, Lith. Acad. Sci.*, 22(1):73–96, 2011.

Guo, Y., Chen, Y., Tang, F., Li, A., Luo, W., and Liu, M. Object tracking using learned feature manifolds. *Computer Vision and Image Understanding*, 118:128–139, 2014.

Han, B. and Davis, L. On-line density-based appearance modeling for object tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1492–1499. IEEE, 2005.

Handels, H., Deserno, T., Meinzer, H.-P., Tolxdorff, T., et al. Image analysis and modeling in medical image computing. *Methods of information in medicine*, 51 (5):395–397, 2012.

Hare, S., Saffari, A., and Torr, P. H. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011.

Hays, J. and Efros, A. A. Large-scale image geolocalization. In *Multimodal Location Estimation of Videos and Images*, pages 41–62. Springer, 2015.

Herath, S., Harandi, M., and Porikli, F. Going deeper into action recognition: A survey. *arXiv preprint arXiv:1605.04988*, 2016.

Hsu, C.-W. and Lin, C.-J. A comparison of methods for multiclass support vector machines. *Trans. Neur. Netw.*, 13(2):415–425, March 2002.

Hue, C., Le Cadre, J.-P., and Pérez, P. Sequential monte carlo methods for multiple target tracking and data fusion. *Signal Processing, IEEE Transactions on*, 50(2): 309–325, 2002.

Ikizler, N. and Duygulu, P. Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image and Vision Computing*, 27(10):1515–1526, 2009.

Isard, M. and MacCormick, J. Bramble: A bayesian multiple-blob tracker. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 34–41. IEEE, 2001.

Jana, A. *Kinect for Windows SDK Programming Guide*. Packt Publishing Ltd, 2012.

Jegou, H., Douze, M., Schmid, C., and Pérez, P. Aggregating local descriptors into a compact image representation. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3304–3311, 2010.

Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., and Schmid, C. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

Jia, Y., Vinyals, O., and Darrell, T. Pooling-invariant image feature learning. *arXiv preprint arXiv:1302.5056*, 2013.

Jin, C. and Wang, L. Dimensionality dependent pac-bayes margin bound. In *Advances in Neural Information Processing Systems*, pages 1034–1042, 2012.

Julier, S. J. and Uhlmann, J. K. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

Ke, S.-R., Thuc, H. L. U., Lee, Y.-J., Hwang, J.-N., Yoo, J.-H., and Choi, K.-H. A review on video-based human activity recognition. *Computers*, 2(2):88–131, 2013.

Kellokumpu, V., Zhao, G., and Pietikäinen, M. Human activity recognition using a dynamic texture based method. In *BMVC*, volume 1, page 2, 2008.

Khansari, M., Rabiee, H. R., Asadi, M., and Ghanbari, M. Occlusion handling for object tracking in crowded video scenes based on the undecimated wavelet features. In *Computer Systems and Applications, 2007. AICCSA'07. IEEE/ACS International Conference on*, pages 692–699. IEEE, 2007.

Kitagawa, G. Non-gaussian statespace modeling of nonstationary time series. *Journal of the American statistical association*, 82(400):1032–1041, 1987.

Klaser, A., Marszałek, M., and Schmid, C. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

Kläser, A., Marszałek, M., Schmid, C., and Zisserman, A. Human focused action localization in video. In *Trends and Topics in Computer Vision*, pages 219–233. Springer, 2010.

Kwon, J., Lee, K. M., and Park, F. C. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In *Computer Vision and*

Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 991–998. IEEE, 2009.

Kwon, J. and Lee, K. M. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

Laptev, I. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

Laptev, I. and Pérez, P. Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

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

Lauer, F. and Guermeur, Y. Msvmpack: A multi-class support vector machine package. *J. Mach. Learn. Res.*, 12:2293–2296, July 2011.

Le Nguyen, M., Shimazu, A., and Phan, H. X. A structured svm semantic parser augmented by semantic tagging with conditional random field. In *Institute of Linguistics, Academia Sinica, The 19th Pacific Asia Conference on Langauge, Information and Computation*, 2005.

Lee, Y., Lin, Y., and Wahba, G. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99:67–81, 2004.

Lei, J., Ren, X., and Fox, D. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 208–211. ACM, 2012.

Li, F.-F. and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02, pages 524–531. IEEE Computer Society, 2005.

Li, L., Huang, W., Gu, I. Y., and Tian, Q. Foreground object detection from videos containing complex background. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 2–10. ACM, 2003.

Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., and Hengel, A. V. D. A survey of appearance models in visual object tracking. *ACM transactions on Intelligent Systems and Technology (TIST)*, 4(4):58, 2013.

Loula, F., Prasad, S., Harber, K., and Shiffrar, M. Recognizing people from their movement. *Journal of Experimental Psychology: Human Perception and Performance*, 31(1):210, 2005.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

Lu, C., Jia, J., and Tang, C.-K. Range-sample depth feature for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 772–779, 2014.

Luber, M., Spinello, L., and Arras, K. O. People tracking in rgb-d data with on-line boosted target models. In *IROS*, pages 3844–3849, 2011.

Manjunath, B. S. and Ma, W.-Y. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):837–842, 1996.

McKenna, S. J., Raja, Y., and Gong, S. Object tracking using adaptive colour mixture models. In *Computer VisionACCV'98*, pages 615–622. Springer, 1998.

Mei, X., Ling, H., Wu, Y., Blasch, E., and Bai, L. Minimum error bounded efficient 1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264. IEEE, 2011.

Meyer, D. and Wien, F. T. Support vector machines. *The Interface to libsvm in package e1071*, 2015.

Mikolajczyk, K. and Tuytelaars, T. Local image features. *Encyclopedia of Biometrics*, pages 1100–1105, 2015.

Mu, Y., Yan, S., Liu, Y., Huang, T., and Zhou, B. Discriminative local binary patterns for human detection in personal album. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Munaro, M., Basso, F., and Menegatti, E. Tracking people within groups with rgb-d data. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2101–2107. IEEE, 2012.

Nguyen, H. T. and Smeulders, A. W. Template tracking using color invariant pixel features. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–569. IEEE, 2002.

Nguyen, H. T. and Smeulders, A. W. Fast occluded object tracking by a robust appearance filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1099–1104, 2004.

Nicola, B., Andrea, V., Viktor, G., and Luis, G. MIBISOC: Medical Imaging Using Bio-inspired and Soft Computing. `http://www.softcomputing.es/metaspace/portal/7/265-about`, 2015. [Online; accessed 9 March 2015].

Odobez, J.-M., Gatica-Perez, D., and Ba, S. O. Embedding motion in model-based stochastic tracking. *Image Processing, IEEE Transactions on*, 15(11):3514–3530, 2006.

Oneata, D., Verbeek, J., and Schmid, C. Efficient action localization with approximately normalized fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2552, 2014.

Oreifej, O. and Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

Oreifej, O. and Zicheng, L. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *CVPR*, pages 716–723, 2013.

Organisation, W. H. WHO Guidelines on Hand Hygiene in Health Care: a summary. `http://www.who.int/gpsc/5may/tools/who_guidelines-handhygiene_summary.pdf`, 2009. [Online; accessed 1 August 2015].

Ozyildiz, E., Krahnstöver, N., and Sharma, R. Adaptive texture and color segmentation for tracking moving objects. *Pattern recognition*, 35(10):2013–2029, 2002.

Palmore, T. N. and Henderson, D. K. Big brother is washing video surveillance for hand hygiene adherence, through the lenses of efficacy and privacy. *Clinical infectious diseases*, page cir781, 2011.

Pan, J. and Hu, B. Robust occlusion handling in object tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Perronnin, F., Sánchez, J., and Mensink, T. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, pages 143–156. Springer-Verlag, 2010.

Pietikäinen, M., Zhao, G., Hadid, A., and Ahonen, T. *Computer Vision Using Local Binary Patterns*. Number 40. Springer, 2011.

Platt, J. C. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, 1999.

Poppe, R. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

Ragheb, H., Velastin, S., Remagnino, P., and Ellis, T. Human action recognition using robust power spectrum features. In *2008 15th IEEE International Conference on Image Processing*, pages 753–756. IEEE, 2008.

Riboni, D., Bettini, C., Civitarese, G., Janjua, Z. H., and Bulgari, V. From lab to life: Fine-grained behavior monitoring in the elderlys home. In *Proc. of PerCom Workshops. IEEE Comp. Soc*, 2015.

Riesen, K. and Bunke, H. *Graph classification and clustering based on vector space embedding*. World Scientific Publishing Co., Inc., 2010.

Rohrbach, M., Amin, S., Andriluka, M., and Schiele, B. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201. IEEE, 2012.

Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

Sadanand, S. and Corso, J. J. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1234–1241. IEEE, 2012.

Sánchez, J., Perronnin, F., and Akata, Z. Fisher vectors for fine-grained visual categorization. In *FGVC Workshop in IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

Satoh, Y., Okatani, T., and Deguchi, K. A color-based tracking by kalman particle filter. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 502–505. IEEE, 2004.

Sax, H., Allegranzi, B., Uckay, I., Larson, E., Boyce, J., and Pittet, D. My five moments for hand hygiene: a user-centred design approach to understand, train, monitor and report hand hygiene. *Journal of Hospital Infection*, 67(1):9–21, 2007.

Schuurmans, L. C. S. and Caelli, S. Implicit online learning with kernels. *Advances in neural information processing systems*, 19:249, 2007.

Schweitzer, H., Bell, J., and Wu, F. Very fast template matching. In *Computer VisionECCV 2002*, pages 358–372. Springer, 2002.

Seidenari, L., Serra, G., Bagdanov, A. D., and Del Bimbo, A. Local pyramidal descriptors for image recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1033–1040, 2014.

Shah, M. Fundamentals of computer vision1. 1997.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, 2011.

Silveira, G. and Malis, E. Real-time visual tracking under arbitrary illumination changes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

Singer, K. Online classification on a budget. *Advances in neural information processing systems*, 16:225, 2004.

Singh, M., Mandal, M., and Basu, A. Robust klt tracking with gaussian and laplacian of gaussian weighting functions. In *null*, pages 661–664. IEEE, 2004.

Skornitzke, S., Fritz, F., Klauss, M., Pahn, G., Hansen, J., Hirsch, J., Grenacher, L., Kauczor, H., and Stiller, W. Qualitative and quantitative evaluation of rigid and deformable motion correction algorithms using dual-energy ct images in view of application to ct perfusion measurements in abdominal organs affected by breathing motion. *The British journal of radiology*, 88(1046):20140683, 2015.

Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. Visual tracking: an experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468, 2014.

Song, S. and Xiao, J. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 233–240. IEEE, 2013.

Spelman, D. W. 2: Hospital-acquired infections. *Medical Journal of Australia*, 176 (6):286–295, 2002.

Stanevski, N. and Tsvetkov, D. Using support vector machines as a binary classifier. In *International Conference on Computer Systems and Technologies–CompSys Tech*, 2005.

Stein, S. and McKenna, S. J. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. ACM, 2013.

Stikic, M., Huynh, T., Laerhoven, K. V., and Schiele, B. Adl recognition based on the combination of rfid and accelerometer sensing. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 258–263. IEEE, 2008.

Sun, C., Shetty, S., Sukthankar, R., and Nevatia, R. Temporal localization of fine-grained actions in videos by domain transfer from web images. *arXiv preprint arXiv:1504.00983*, 2015.

Tang, S., Wang, X., Lv, X., Han, T. X., Keller, J., He, Z., Skubic, M., and Lao, S. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Asian conference on computer vision*, pages 525–538. Springer, 2012.

Thurau, C. and Hlavác, V. Pose primitive based human action recognition in videos or still images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Todd, A., Natalia, A., Joey, Y., and Holly, G. Ge global research newsroom. `http://www.geglobalresearch.com/news/press-releases/`, 2015. [Online; accessed 9 March 2015].

Tran, S. and Davis, L. Robust object trackinng wvith regional affine invariant features. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005.

Vieira, A. W., Nascimento, E. R., Oliveira, G. L., Liu, Z., and Campos, M. F. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In *Iberoamerican Congress on Pattern Recognition*, pages 252–259. Springer, 2012.

Vincent, J.-L. Nosocomial infections in adult intensive-care units. *The Lancet*, 361 (9374):2068–2077, 2003.

Vucetic, S., Coric, V., and Wang, Z. Compressed kernel perceptrons. pages 153–162, 2009.

Wah, C., Branson, S., Perona, P., and Belongie, S. Multiclass recognition and part localization with humans in the loop. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2524–2531. IEEE, 2011.

Wang, B., Liu, Y., Wang, W., Xu, W., and Zhang, M. Multi-scale locality-constrained spatiotemporal coding for local feature based human action recognition. *The Scientific World Journal*, 2013, 2013a.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, pages 124–1. BMVA Press, 2009.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013b.

Wang, J., Liu, Z., and Wu, Y. *Human Action Recognition with Depth Cameras*. Springer, 2014.

Wang, J., Chen, X., and Gao, W. Online selecting discriminative tracking features using particle filter. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1037–1042. IEEE, 2005.

Wang, S., Lu, H., Yang, F., and Yang, M.-H. Superpixel tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1323–1330. IEEE, 2011.

Wang, Z., Crammer, K., and Vucetic, S. Multi-class pegasos on a budget. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1143–1150, 2010.

Weng, S.-K., Kuo, C.-M., and Tu, S.-K. Video object tracking using adaptive kalman filter. *Journal of Visual Communication and Image Representation*, 17 (6):1190–1208, 2006.

Weston, J., Bordes, A., and Bottou, L. Online (and offline) on an even tighter budget. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 413–420, 2005.

Wu, Y., Lim, J., and Yang, M.-H. Online object tracking: A benchmark. In *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013.

Xia, L. and Aggarwal, J. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2013, pages 2834–2841, 2013a.

Xia, L. and Aggarwal, J. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2834–2841, 2013b.

Xie, L., Tian, Q., Hong, R., Yan, S., and Zhang, B. Hierarchical part matching for fine-grained visual categorization. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1641–1648. IEEE, 2013.

Yang, H., Shao, L., Zheng, F., Wang, L., and Song, Z. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

Yang, J. and Ma, Z. Action recognition using hierarchical stip saliency and mixed neighborhood features. *International Journal of Control and Automation*, 9(3): 245–260, 2016.

Yao, B., Khosla, A., and Fei-Fei, L. Combining randomization and discrimination for fine-grained image categorization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1577–1584. IEEE, 2011.

Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

Zhang, H., Huang, Z., Huang, W., and Li, L. Kernel-based method for tracking objects with rotation and translation. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 728–731. IEEE, 2004.

Zhang, K., Zhang, L., and Yang, M.-H. Real-time compressive tracking. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 864–877, 2012.

Zhang, Z., Hu, Y., Chan, S., and Chia, L.-T. Motion context: A new representation for human action recognition. In *European Conference on Computer Vision*, pages 817–829. Springer, 2008.

Zhao, Y., Liu, Z., Yang, L., and Cheng, H. Combining rgb and depth map features for human activity recognition. In *2012 Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–4, 2012.

Zhao, Z. and Elgammal, A. Human activity recognition from frames spatiotemporal representation. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

Zhou, S. K., Chellappa, R., and Moghaddam, B. Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Processing, IEEE Transactions on*, 13(11):1491–1506, 2004.

Zhou, Y., Ni, B., Yan, S., Moulin, P., and Tian, Q. Pipelining localized semantic features for fine-grained action recognition. In *Computer Vision–ECCV 2014*, pages 481–496. Springer, 2014.

Zivkovic, Z. and Krose, B. An em-like algorithm for color-histogram-based object tracking. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–798. IEEE, 2004.