# Multi-scale Conditional Random Fields for First-Person Activity Recognition

Kai Zhan† Steven Faux* Fabio Ramos†

†School of Information Technologies
University of Sydney
Sydney, Australia
{kai.zhan;fabio.ramos}@sydney.edu.au

*Faculty of Medicine
University of New South Wales
Sydney, Australia
sfaux@stvincents.com.au

***Abstract*— We propose a novel pervasive system to recognise human daily activities from a wearable device. The system is designed in a form of reading glasses, named 'Smart Glasses', integrating a 3-axis accelerometer and a first-person view camera. Our aim is to classify user's activities of daily living (ADLs) based on both vision and head motion data. This ego-activity recognition system not only allows caretakers to track on a specific person (such as patient or elderly people), but also has the potential to remind/warn people with cognitive impairments of hazardous situations. We present the following contributions in this paper: a feature extraction method from accelerometer and video; a classification algorithm integrating both locomotive (body motions) and stationary activities (without or with small motions); a novel multi-scale dynamic graphical model structure for structured classification over time. We collect, train and validate our system on a large dataset containing 20 hours of ADLs data, including 12 daily activities under different environmental settings. Our method improves the classification performance (F-Score) of conventional approaches from 43.32%(video features) and 66.02%(acceleration features) by an average of 20-40% to 84.45%, with an overall accuracy of 90.04% in realistic ADLs.**

## I. Introduction

The challenges associated with population ageing and nurse shortages are opening unprecedented opportunities for pervasive computing. Such systems can dramatically impact the quality of life of the ageing population by helping people through their daily living activities. A central requirement of pervasive systems is to automatically recognise human activities over time, instructing patients in hazardous situations or reminding them of important needs. For example, patients with memory loss or cognitive disorders can be reminded to take pills after having meals or bring along the walking stick while going outside. Additionally, they can provide online monitoring interfaces for nurses and caretakers to remotely access the patient's status to derive better treatments.

Daily activities can be categorised into two major groups in terms of the motion magnitudes, locomotive and stationary activities. A locomotive activity can be defined as an activity involving high energy, with specific body movements, such as walking. A stationary activity involves less or no motion, such as reading a book or watching TV. Due to the complexity and variety of daily activities, researchers have explored different approaches in activity recognition mostly based on acceleration and visual observations. Wearable accelerometers are often used to classify activities of daily living (ADLs) [1], [2], [3], [4], or to recognise occasional events, such as falls [5] or stumbles [6]. Other approaches integrate accelerometers with additional sensors to improve the system's performance, such as gyroscopes [7], microphones [8], and floor sensors [9]. Despite major achievements in automatic classification of activities from accelerometer data containing high motion magnitudes, classifying stationary activities, especially in identical static postures, still remains a key challenge, due to the similarity of the acceleration signals. For example, sitting still, reading a book or watching TV, all have the same sitting body posture but are different activities.

Visual information is an alternative method to recognise human activities. The idea is to use a single or a sequence of images containing parts or the entire human body in order to estimate the subject's posture or motion information to predict activities. Notable approaches using external cameras are described in [10], [11], [12]. These methods are mostly applied to surveillance purposes, and are often constrained to particular regions of the field of view. Recently, first-person view methods were introduced [13], [14], [15]. In these approaches, cameras are embedded into a wearable device to capture a similar field of view as the person's eyes. Image features are then extracted from objects or motion flow information. Object-oriented approaches rely on object recognition techniques and are useful to classify stationary activities involving specific objects from the video [15], [16]. However, these methods have limitations to recognise activities without clear foreground objects, for example, when the user is looking down while walking. Motion-oriented methods have reasonable performance in identifying activities using optical flow features such as those involved in sports [17].

In this paper, we introduce an automatic activity recognition system integrating both accelerometers and a first-person view camera in conventional glasses, called "Smart Glasses". As a prototype, we use a smart phone (Android OS) attached on top of safety goggles as shown in Figure 1. The device collects videos and 3-axis acceleration data. Both are synchronised and collected in parallel. We aim to develop a model able to recognise a wider range of human

Fig. 1. Our senior patients wearing Smart-Glasses prototype.

activities including dynamic and stationary. There are three contributions presented in this paper:

**Feature Extraction**: We carefully select a number of activities following healthcare professionals' directives in this study. Some of these activities have identical static postures. This makes features from the accelerometer less important. Among of static/stationary activities, a method that's able to correctly recognise if the subject is reading a book or watching a TV becomes our major concern. Therefore, in this paper, we use the video motion feature as a complementary element. This allows the system to track motion flow from consequent egocentric images in order to improve the system performance. We design separate feature extraction algorithms for both accelerometer and video data, detailed in Section II-A and II-B.

**Feature Integration**: From extensive experiments, we select the best classifier and settings for each set of features, and separate the local classification task into two categories, each assigned with weighting parameters obtained during the training process. This allows the model to choose the suitable set of feature for different activities.

**Multi-Scale Graphical Model**: We develop a Conditional Random Field (CRF) to capture the multi-scale context in a sequence of activities. This model can help to predict user's activities at different temporal scales even when the local classification is significantly noisy or ambiguous. For example, when sitting, reading or drinking, there is a period without any motion features detected from both accelerometer and video features but the activity can be recognised from the context.

In Section II, we present an overview of the system processing pipeline, followed by the details of feature extraction, feature classification and the proposed probabilistic graphical model. We evaluate the performance of each individual step of our system, and present results on real deployments in Section III. Topics for future work and conclusions are in Section IV.

## II. System Overview

The model can be described as a pipeline with three major steps: video and acceleration feature extraction, classification and structure prediction. As shown in Figure 2, both video and accelerometer data are collected and processed in parallel, feeding features into separate classifiers. The classifier result is then transformed into a class probability vector. In the final step, these vectors are associated with a unary feature function which is then combined with pairwise functions in a graphical model to perform structured prediction. The final prediction is obtained after an inference procedure on the graphical model that takes temporal relationships into account, where the relative weights of unary and pairwise features are obtained through a learning procedure. We detail these steps below.

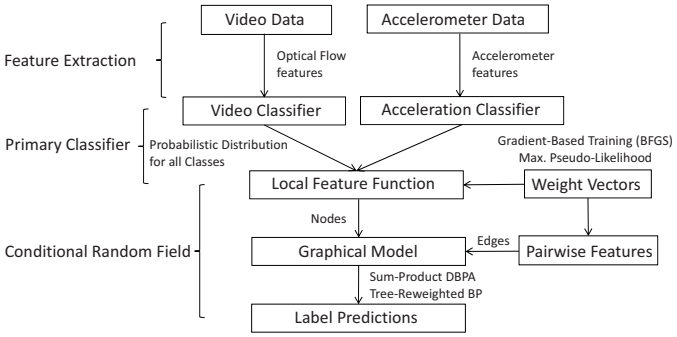


Fig. 2. System Overview.

### A. Acceleration features

Head acceleration is relatively complex because it is the result of a concatenation of motions of several body parts. However, this type of signal is very valuable, providing information about the subject's body motion in addition to information from the head acceleration. For example, drinking water often results in head raising, hand-washing usually requires a head bowing. Here, we consider a wide range of activities covering various types of motions, including: locomotive or stationary, periodic or infrequent, body or hand activities. Generally, acceleration can be represented in both time and frequency domains. In this work we extract features from a sliding window containing a short period of time. To cover the different types of activities, we conduct a comprehensive analysis of 13 and 20 features in both time and frequency domains from 3-axis accelerometer data. Time domain features include a number of basic features such as mean, standard deviations, signal magnitude, local maxima, threshold-crossing rate, etc, both extracted directly from raw sensor data. Frequency domain features are extracted with Fast Fourier Transform (FFT). We divide each feature window into 10 sub-bands. For each band, we obtain the magnitude and frequency of the peak value as our frequency feature. The cut-off threshold is set to 5Hz since daily human activities are unlikely to require higher frequencies based on our ADLs database.

### B. Video features

First-person video features work as a complement to the body acceleration data. We aim to extract motion features from egocentric images, such that the system can monitor activities even when the subject is not moving. Our video features are based on the Lucas-Kanade optical flow method [18], which estimates the motion of objects across a series of consecutive image frames, such as hands or objects. Between every pair of frames, as defined in [19], we firstly average

dense optical flow within the set of non-overlapping patches (each with $m$ pixels) and each frame is partitioned into $i$ rows and $j$ columns. Then, we conduct an average pooling process over $n$ consecutive patches from all corresponding positions to obtain a $1 \times 2ij$ vector, containing horizontal and vertical motion information for the period covering $n$ frames, written as

$$[\overline{u}_{ij}, \overline{v}_{ij}] = \frac{1}{MN} \sum_{n=1}^{N} \sum_{m=1}^{M} [u_{ij}^{m,n}, v_{ij}^{m,n}], \tag{1}$$

where $\overline{u}_{ij}$ and $\overline{v}_{ij}$ are the motion feature for patch$_{ij}$ in both horizontal and vertical directions.

### C. Classification

There are two levels of classification in our approach: local (i.i.d) and structured. The former uses features directly extracted from raw sensor data and provides predictions independently of time. The later depends on the graph structure and takes into account temporal dependencies. In this section, we focus on local classification. As shown in Figure 2 and in the previous section, our system extracts local features from 3-axis head accelerometers and an egocentric camera. These features can differ drastically in terms of the magnitude and frequency of the measurements. In this work we consider two popular off-the-shelf classifiers and compare them for the two sets of features obtained before. Our choice for the classifiers is based on their computational costs and scalability to handle high-dimensional data.

Support Vector Machine (SVM) is one of the most popular approaches for human activity recognition, especially from body acceleration [7], [20], [21], [22], [23]. SVMs are large-margin classifiers with strong generalisation properties, primarily designed for binary classification even though extensions to multi class exist [24], [25]. We use a multi-class SVM based on an "One-Versus-One(OVO)" technique which fits binary sub-classifiers and obtain the final prediction through a voting process. We also estimate the class probabilities through a pairwise coupling method [26] for the structure classification stage. Three popular kernels were selected and compared: Linear, Polynomial and Gaussian Radial Basis Function (RBF).

Boosting was first introduced by Schapire *et al.* [27] in 1990. The algorithm produces an ensemble model by greedily adding weak learners trained on data points weighted by their classification error from previous rounds. Boosting significantly improves the accuracy of a base-level binary classifier (weak learner) and can learn complex non-linear decision boundaries. It has been widely and successfully applied to many fields [28], [29], [30], [31]. Inspired by [32], we implement a LogitBoost algorithm which provides probability distributions of multi-class problems with decision stumps as the weak learner.

### D. Structured Classification

*1) Conditional Random Fields:* One of the main drawbacks in conventional i.i.d classification approaches is that context is not taken into account – classification is performed locally without considering any 'neighbours' in time or space. This can lead to mistakes that could be avoided otherwise. For example, a single sliding window might wrongly predict hand-washing due to strong image motion flow when the correct activity is walking. This could have been fixed had context been considered since a single hand-washing classification is unlikely to take place in a series of walking frames. We develop a structure classification approach for contextual activity recognition using conditional random fields (CRFs). CRFs are undirected graphical models designed for labelling sequences of data. It is a powerful tool in structured learning that allows us to model the correlations (through edges) between each defined pair of nodes in a graphical model, specifying a conditional probabilistic distribution over the query nodes given observed nodes [33].

For example, in the context of activity recognition, the nodes can be seen as containing local information of each time interval, while the edges are pairwise relations between consecutive intervals. Therefore, the order of the nodes can be explained as sequences of activities in time domain. A key advantage of CRFs compared to a generative model is that it models the conditional probability of hidden states given observations directly. This provides more flexibility to define potential functions into our system. The CRF contains a normalising partition function that groups all potentials into a general format. In activity recognition, it allows us to integrate heterogeneous sensors into the graphical model seamlessly.

Theoretically, CRFs are a special case of Markov Random Fields (MRF). It models conditional distributions of the hidden nodes $\mathbf{x}$ given observations $\mathbf{z}$, written as $p(\mathbf{x}|\mathbf{z})$. Within the graph, the hidden nodes $\mathbf{x}$ are linked by edges following a predefined graph structure. Each fully connected subset of nodes (clique) $c \epsilon C$ is described by a nonnegative clique potential function $\phi_c(x_c|z)$, which maps clique variables to a positive real number. A CRF distribution over the cliques can be written as,

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c \epsilon C} \phi_c(\mathbf{x}_c|\mathbf{z}), \tag{2}$$

where the partition function $Z(\mathbf{z})$ is expressed as

$$Z(\mathbf{z}) = \sum_{\mathbf{x}} \prod_{c \epsilon C} \phi_c(\mathbf{x}_c|\mathbf{z}). \tag{3}$$

The potentials $\phi_c(\mathbf{x}_c|\mathbf{z})$ are usually represented as log-linear combinations of feature functions $f_c(\mathbf{x}_c|\mathbf{z})$ with a weight factor $w_c$, expressed as:

$$\phi_c(\mathbf{x}_c|\mathbf{z}) = \exp(w_c^T f_c(\mathbf{x}_c|\mathbf{z})). \tag{4}$$

Combining Equations 2 and 4 in the CRF, the conditional distribution of the hidden nodes can be rewritten as

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z}, w)} \exp \left\{ \sum_{c \epsilon C} w_c^T f_c(\mathbf{x}_c|\mathbf{z})) \right\}. \tag{5}$$

*2) Graph Structure:* Since ADLs are sequential events, we build a CRF model to capture temporal relationships. The graph structure is shown in Figure 3. It contains sequences of observations $\mathbf{z}$ from sensor features, hidden nodes $\mathbf{x}$ of class probability assignments, and edges $\mathcal{E}$ representing pairwise relations. Each node in the sequence contains information from a short period, and connects to a number of nodes at different distances. For example, in Figure 3, node $x_s$ connects to 6 nodes in three different scales from shorter (1 unit) to a longer (4 units). These connections allow for contextual information to be incorporated into the entire network. This graph contains two types of potentials, named local and pairwise potentials. The local potential captures local information within an interval, while the pairwise potential explains how the nodes relate to each other. Including both node and edge features, the overall clique potential can be written as:

$$\phi_c(\mathbf{x}_c|\mathbf{z}) = \exp(\sum_{s\in\mathcal{V}} w_s^T f_s(\mathbf{x}_s|\mathbf{z}) + \sum_{sd\in\mathcal{E}} w_{sd}^T f_{sd}(\mathbf{x}_s, \mathbf{x}_d|\mathbf{z})) \tag{6}$$

where $f_s$ is the local potential for a hidden node $\mathbf{x}_s$ and $f_{sd}$ is a pairwise potential connecting a node $\mathbf{x}_s$ and $\mathbf{x}_d$. Individual weights are also assigned to each of the functions. They encode the relative importance of each potential. We now describe in detail the local and pairwise potentials used.

**Local potential** - Represented by the observation nodes $\mathbf{z}$ in Figure 3, they contain features extracted directly from data and encode local information from an interval. In our model, we have accelerometer and video features stored in two observation nodes $\mathbf{z}_A$ and $\mathbf{z}_V$. Each classifier uses the features to predict a class-probability vector $\mathbf{P}_A$ and $\mathbf{P}_V$ in the size of $\mathcal{M}$, where $\mathcal{M}$ is the number of state (activity) in this study. Interestingly, acceleration and video features lead to very different performance on the classification of separate activities (shown in the experiments section). Acceleration generally has good accuracy on locomotive activities, and video motion features are more accurate for stationary activities, especially the static activities. For this reason, the probabilistic vector $P$ is divided into two class categories: $\alpha$ (acceleration) and $\beta$ (video). $\alpha$ contains the activity classes where the acceleration feature is better suited than video features. Conversely, $\beta$ contains activities more suited for video features. Note that, we keep classification results the same for each category and set the rest to zero. For example, assume there are two activities involved, let $P_A$ be a normalised $1\times2$ vector from acceleration features, which has walking and sitting probabilities expressed as $[0.35, 0.65]$. If walking is registered as a dynamic activity, then $P_{A-\alpha}$ and $P_{A-\beta}$ become $[0.35, 0]$ and $[0, 0.65]$ respectively. In this way, we can assign a separate weight vector to different activity groups. More details are in section III-B.

**Pairwise Feature** - These are potentials defined over the edges connecting each pair of hidden nodes. It specifies a relationship from one state to another using a matrix of size $\mathcal{M} \times \mathcal{M}$. In this research we use a point-to-point weight assignment method, this allows the model to define individual transition weights between different states. Therefore, a 4-activity model requires $4^2$ (16) weights, which represent the likelihood of transitioning between activities. Such as, from sitting to drinking, walking to climbing stairs, or simply walking to walking.

To capture multi-scale temporal correlations, we introduce a distance-inference method into the model. The example shown in Figure 3 has a number of edges on the top, where the hidden node $\mathbf{x}_s$ connects the third $\mathbf{x}_{(s\pm2)}$ and fifth $\mathbf{x}_{(s\pm4)}$ neighbours. The transition matrices of $Edge_{(s,s\pm1)}$ and $Edge_{(s,s\pm4)}$ can be very different; the former refers to nodes from its immediate predecessor or successor, while the latter links to the ancestor or descendant. In this case, the node can be predicted using further contextual information from a larger group of neighbours to improve the system robustness.

### *E. Parameter Learning*

The overall goal of parameter learning is to determine the most suitable values for the weight vector $w_s$ and $w_{sd}$ in the feature functions. It involves an optimisation process to maximise the conditional likelihood of the training set. However, directly maximising the conditional likelihood can be extremely time consuming due to the need to perform inference in every step of the algorithm; the partition function $Z$ needs to be computed in every iteration. In order to make learning tractable for these problems, we maximise the pseudo-likelihood of the training data [34]. This approximates the conditional likelihood by a product of conditional distributions over given immediate neighbours (Markov Blanket) of $x_s$. Let $N$ be the total number of nodes in our model, the pseudo-likelihood can be computed as,

$$p(\mathbf{x}) = \prod_{s=1}^{N} p(\mathbf{x}_s|MB(\mathbf{x}_s)). \tag{7}$$

For mathematical convenience, the optimisation on $p(\mathbf{x})$ can be achieved through a maximisation process in the log domain. To prevent the weights from getting too large during the optimisation process, the pseudo-log likelihood objective is typically regularised with a quadratic term,

$$\mathcal{L}_{\mathcal{R}}(w|\mathcal{D}) = \mathcal{L}(w|\mathcal{D}) - \frac{(w-\bar{w})^T(w-\bar{w})}{2\sigma^2} \tag{8}$$

where $\mathcal{D}$ is the training set expressed as $\mathcal{D} = (\mathcal{X}_s, \mathcal{Z}_s|s = 1, \ldots, N)$, and $\mathcal{L}(w|\mathcal{D})$ is the pseudo-log likelihood written as

$$\mathcal{L}(w|\mathcal{D}) = \sum_{s=1}^{N}\sum_{m=1}^{M}\Big\{w_s^T f_s(x_s|z_s) + \sum_{i=1}^{I} w_{sd(i)}^T f_{sd}(x_{s,m}|z_s) - \log\left(Z_m(z_s, MB(x_s), w_s, w_{sd})\right)\Big\}, \tag{9}$$

where the local feature function $w_s^T f_s(x_s|z_s)$ is defined as

$$w_s^T f_s(x_s|z_s) = w_1^T f_{acc.\alpha} + w_2^T f_{acc.\beta} + w_3^T f_{vid.\alpha} + w_4^T f_{vid.\beta}. \tag{10}$$

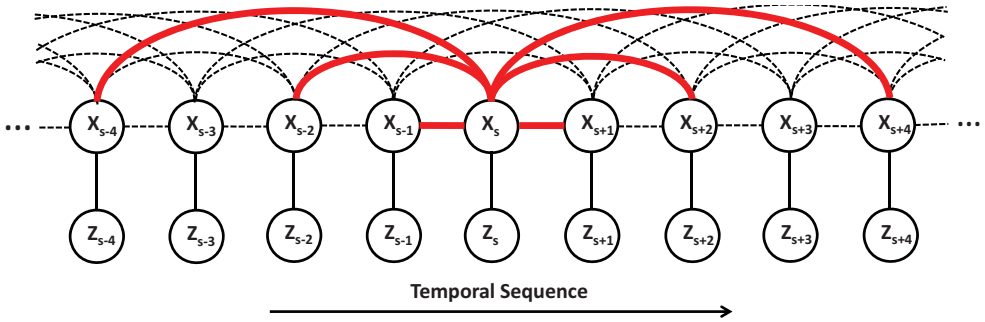


Fig. 3. Multi-scale CRF graph structure. The red edges represent the connection setting of '020305' for node $\mathbf{x}_s$.

As shown in Equation 10, the 4 local potentials, as previous explained in Section II-D.2, includes 2 acceleration and 2 video feature functions for two sensor sources, where $f_{acc.\alpha}$ represents a vector predicted from acceleration data for activities $\alpha$, and $f_{vid.\beta}$ is the video vector for the group $\beta$.

Equation 9 contains three components: local, pairwise and partition functions. The partition function $Z_m$ is a local "committee" (comparing with a global network), where Markov blanket dramatically reduces the computational cost from the original partition function. To elaborate it, we assume there are $N$ nodes with $M$ states each, in this case, $Z_m$ sums $M$ states from the local Markov blanket of $\mathbf{x}_s$. Therefore, computation is repeated $M \times N$ times for the entire space. comparing to the general form of Equation 3, where $Z(\mathbf{z})$ has to evaluate $M^N$ values. Since $\mathcal{L}_{\mathcal{R}}(w|\mathcal{D})$ is a convex function, the local maximum can be achieved by a gradient descent algorithm. In this research we use the unconstrained L-BFGS method [35] on the negative version of $\mathcal{L}_{\mathcal{R}}(w|\mathcal{D})$.

### F. Probabilistic Inference

The inference procedure computes statistics for the hidden nodes $\mathbf{x}$ given the graph structure, and the observations $\mathbf{z}$. There are two basic operations, the computation of *Marginal Distributions*– the posterior distribution $p$ for each of the variables $\mathbf{x}$, and *Maximum A Posteriori Configuration* – the most likely assignment of $\mathbf{x}$. Both the marginals and MAP configuration can be computed using belief propagation (BP) [36]. In particular, the sum-product version of BP performs marginalisation and the max-product version of BP computes the MAP configuration. Conventional BP when applied to tree graphs provides an exact answer. Messages are propagated from the leaves to the root node and back again. For arbitrary graphs with loops, there is another popular variation called Loopy Belief Propagation (LBP) [37] which provides approximate answers. It updates messages in every iteration until converge (which is not guaranteed but typically happens). Another variation with stronger convergence properties is Tree-Reweighted BP (TRBP) [38]. It provides a guaranteed bound on the log partition function. It decomposes the original graph into a convex combination of tree-structured graphs allowing efficient computations, while the convex combination allow the computation of an upper bound on the optimal solution. In our model, as shown in Figure 3, we will compare the performance of these three approaches: BP on a chain model (CBP), LBP and TRBP to select the best approach for the problem.

## III. Experiments

We validate our model on 40 independent datasets, equally split from 5 senior volunteers, with age over 55. Each dataset contains an average of 30-minute sensors recording on realistic sequential ADLs from separated days, as detailed in Table I. Note that some actions are repeated multiple times in each video. To best represent realistic situations, the subjects are not asked to perform specific sequences of activities or detailed motions. They simply follow their normal ADLs sequences on their own preferences.

TABLE I
A LIST OF ACTIVITY DISPLAYS THE AVERAGE DURATION FOR EACH ACTIONS CONTAINED IN OUR DATASET.

| ID | Activity | Average Duration |
|---|---|---|
| 1 | Walking | 154.99 sec |
| 2 | Going Upstairs | 59.05 sec |
| 3 | Going Downstairs | 55.14 sec |
| 4 | Drinking | 15.72 sec |
| 5 | Stand Up | 1.71 sec |
| 6 | Sit Down | 2.32 sec |
| 7 | Sitting | 46.08 sec |
| 8 | Reading | 45.21 sec |
| 9 | Watching TV/monitor | 253.58 sec |
| 10 | Writing | 112.78 sec |
| 11 | Switch Water-Tap | 1.67 sec |
| 12 | Hand-Washing | 10.39 sec |

In the experiment, we use a 'Sony Ericsson Xperia mini' mobile phone as our online processing unit. The first-person viewing angle is approximately 90 degree, and video is recorded at 15Hz with 144 $\times$ 176 resolution, synchronised with a 80Hz 3-axis acceleration readings based on the android system timestamps. As a summary, we totally collect 1.08 million frames and 17.3 millions of sensor sampling points from 5 subjects. As noticed earlier, each dataset covers few activities following user's own preferred sequence. We manually label all the data based on the videos, therefore the synchronised acceleration data can be automatically annotated from video labels.

In the next section, we detailed the parametrisation, feature window size, classifier settings and multi-scale graph design. Then, we categorise the activities for both sensors followed by the inference method comparison and experimental results.

### A. Parametrisation

*1) Window Size:* The window size defines how much information (duration) is required to classify an activity. Therefore, each window should contain just sufficient data to describe the subject's current status. The overlapping percentage is another concern during data collection, it helps the classifier distinguish the contents between consequent windows with a transition period. It also helps to recognise the feature from a wider context in order to compensate the possible errors from the window itself. We run cross validation over 10 ADLs videos to determine a suitable windows size and overlap portion chosen from 9 different settings: 2, 3, 5 seconds of window length with 25%, 50% and 75% overlapping portion. After a number of cross validations from our independent training database, we estimate the 3-second window with 50% overlap has the best performance over all settings.

*2) Local Classifiers:* In this research, we use two classifiers: LogitBoost and SVM. LogitBoost requires the number of Weak Learners (WLs) to be specified for both features and is usually resilient to over-fitting. We take 5 independent video segments for each activity and each segment contains 5 minutes. We firstly extract video and acceleration features into windows and re-arrange them in a random order. Then we conduct a 10-fold cross validation on both features, and test LogitBoost with 10 different weak learners settings, from 50 to 500. Our results show that 150-WLs is the most reliable for the video features and 50-WLs for the acceleration features. Selecting a suitable kernel for SVM can be difficult. We run the same validation process on three popular kernels: Linear, Polynomial and Gaussian Radial Basis Function (RBF). The averaged precision and recall results for both classifiers are shown in Table II.

TABLE II
CLASSIFIER ACCURACY ON ACCELERATION AND VIDEO FEATURES

| Classifier | LogitBoost | SVM | | |
|---|---|---|---|---|
| Acceleration | 50-WLs | Linear | Polynomial | RBF |
| Averaged Precision | 68.99% | **69.39%** | 43.6% | 37.01% |
| Average Recall | 61.01% | **62.96%** | 34.98% | 34.89% |
| Overall Accuracy | 77.40% | **78.07%** | 60.51% | 66.92% |
| Video | 150-WLs | Linear | Polynomial | RBF |
| Averaged Precision | **53.12%** | 37.58% | 35.19% | 43.86% |
| Average Recall | **44.76%** | 30.11% | 26.39% | 42.78% |
| Overall Accuracy | **68.91%** | 56.39% | 44.63% | 61.68% |

As can be seen, SVMs with linear kernel has the best performance for the acceleration features. LogitBoost does better on video optical flow features. Note that our dataset is quite realistic including various ADLs; the activities include both dynamic and static actions. Some of the static activities, such as watching or sitting might not contain any acceleration signals. Conversely, motions such as walking and climbing stairs would create significant noise in the egocentric vision. For this reason the local classifiers in isolation generally do not achieve a very high accuracy.

*3) Graph structure:* In Section II-D.2, we introduce the multi-scale graphical model. Each node connects to multiple neighbours to create a more powerful inference network. In this section, we validate 8 different settings from 5 different scales: 2nd neighbour (chain model), 3rd neighbour, 5th neighbour, 10th neighbour and 20th neighbour. The number indicates the distance before/after a particular node in time. It represents the number of time steps between a pair of connected nodes. In the last section, we had selected the time window interval as 3 seconds with 50% overlapping. This means a 5th neighbour is $4 \times 1.5 = 6$s away from the current node, 10th neighbours is 13.5s away and 20th is 28.5s apart.

To obtain an optimal graph setting, we take 10 datasets from our database, train on 8 datasets, and test on 2 using a sum-product TRBP inference method. For each dataset, we set 8 edge configurations: '02'(chain graph), '0203', '0205', '0210', '0220', '020305', '020510' and '02051020', and use LogicBoost as our feature classifier on acceleration features. The settings, for example, '020305' means each node connects to 2nd nodes (immediate nodes), 3rd nodes and 5th nodes. An example is shown in Figure 3. In the example $\S_s$ is with the '020305' setting. Note that the edge distance impacts on the updating frequency and the contextual information. Graphs with longer edges generally require more memory and result in lower updating frequency. e.g. 20th neighbour requires to store at least 28.5s of data, and 100th neighbour would need 142.5s of data. Therefore, by balancing these factors, we choose up to 20th neighbour as our maximum inference distance. To compare these edge settings, we use the averaged precision value as our evaluation preferences. The results of selected configurations are illustrated in Figure 4, plotted in the form of error-bars. The first term is the precision of the local classification result before applying the graphical model.

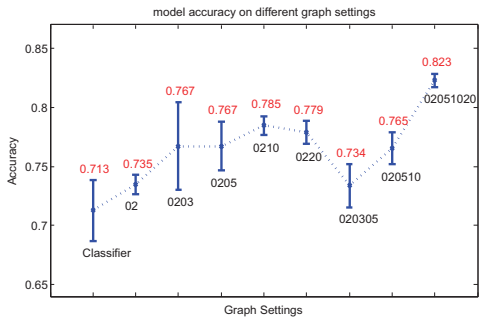


Fig. 4. Overview of model accuracy when using different graph settings

In Figure 4, the graph setting '02051020' shows the highest precision, outperforming the local feature classifier by 11% achieving 82.3%, followed by '0210' at 78.5% and '0220' at 77.9%, where '020305' and '02' have the worst

performance with accuracy of 73.4% and 73.5% respectively, but still better than the local classifiers. Therefore, we suggest the graph '02051020' as the best setting, thus to be applied in the next set of experiments.

### B. Activity Categorisation

In Section II-D.2, we introduce the local features integration method, which requires to separate the activities into two categories. However, our ultimate goal is to let the model determine which feature is better for each activity, such that the CRF is able to train and assign appropriate weights for both vision and acceleration features. In order to achieve this, a direct approach is used to inspect their current classification performance, and to find the prediction accuracy of each activity using both features. From Section III-A.2, we learned that SVM (linear kernel) has the best performance on acceleration features, and LogitBoost is good on video feature classification. Therefore, we run a 20-fold cross validation over 10 datasets using both classifiers. All activities are evaluated using precision and recall [39]. The classification results are shown in Table III.

TABLE III
RESULTS FROM ACCELERATION AND VIDEO FEATURES WITH ACTIVITIES FOLLOWING THE ORDER IN TABLE I.

| | Acceleration | | Video | | Differences | |
|---|---|---|---|---|---|---|
| **Act. ID** | **Pre.** | **Rec.** | **Pre.** | **Rec.** | **Pre.** | **Rec.** |
| **1** | 0.936 | 0.961 | 0.422 | 0.384 | 0.514 | 0.577 |
| **2** | 0.878 | 0.879 | 0.305 | 0.305 | 0.573 | 0.575 |
| **3** | 0.868 | 0.868 | 0.212 | 0.200 | 0.656 | 0.668 |
| **4** | 0.670 | 0.584 | 0.523 | 0.375 | 0.147 | 0.209 |
| **5** | 0.754 | 0.679 | 0.238 | 0.227 | 0.516 | 0.451 |
| **6** | 0.771 | 0.521 | 0.207 | 0.229 | 0.564 | 0.293 |
| **7** | 0.250 | 0.418 | 0.350 | 0.333 | -0.100 | 0.086 |
| **8** | 0.503 | 0.624 | 0.682 | 0.749 | -0.178 | -0.124 |
| **9** | 0.581 | 0.505 | 0.722 | 0.684 | -0.141 | -0.178 |
| **10** | 0.762 | 0.759 | 0.828 | 0.753 | -0.067 | 0.006 |
| **11** | 0.429 | 0.375 | 0.438 | 0.475 | -0.009 | -0.100 |
| **12** | 0.375 | 0.136 | 0.577 | 0.482 | -0.202 | -0.345 |
| **Ave.** | **0.648** | **0.609** | **0.459** | **0.433** | **0.189** | **0.176** |

From the table, it is interesting to note that classification from acceleration features has good precision on most of the dynamic motion against the statics ones, while the video features have better prediction on stationary activities (Sitting, Reading, Watching TV/monitor, Writing, Switching Water-Tap and Hand-Washing) than locomotive activities (Walking, Going Upstairs, Going Downstairs, Drinking, Stand Up and Sit Down). A reason that may explain this phenomenon is: locomotive activity usually involves a number of periodic motions or infrequent motion with high magnitude, which can be easily captured from an accelerometer. On the other hand, activities involving less mobility such as sitting or reading can be better classified with optical flow features because the user is in a stationary position, but the motions can be captured from the egocentric video. Figure 5 demonstrates three examples for both acceleration and vision features: *Walk*, *Drink* and *Write*. Note that in the *Drink* window, the signal change at the beginning represents a head-lifting motion. Writing involves only a small motion compared to the others.

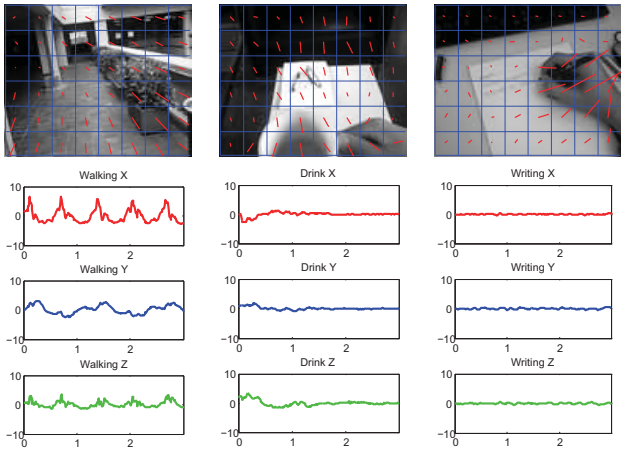


Fig. 5. Examples of optical flow features (1st row) and corresponding 3-axis acceleration window from three typical activities: Walking, Drinking and Writing.

### C. Inference Approaches

In this section, we compare 3 BP inference approaches with leave-one-out cross validation, using the same training datasets from Section III-B. We train and test them on both vision and acceleration features, and evaluate the model performance in terms of the classification accuracy and processing speed. The average performance is detailed in Figure 6.

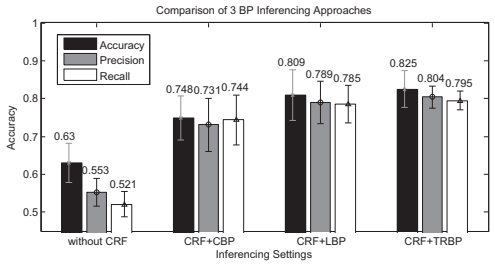


Fig. 6. Comparison of different inference methods.

The chart displays the comparisons of 3 BP approaches, TRBP has the best performance with an averaged precision of 80.4%, follow by LBP at 78.9% and then CBP at 73.1%. We also test the processing speed by conducting all three approaches on one 30-minute dataset. The average running times are: *CBP-15.19s*, *LBP-30.20s*, *TRBP-5.17s*. The result shows that both TRBP and LBP have a higher accuracy compare to CBP, but TRBP is faster than the others. We conclude, the TRBP is the most appropriate inference algorithm from our validation.

### D. Results

In this experiment, we run leave-one-out cross validation on 40 independent datasets using the optimal settings determined from previous sections. We run experiments to observe

the results among several different approaches. The averaged precision, recall and overall accuracy are then compared. Our evaluation focuses on the performance of structure classification, as well as the benefits of local-pairwise feature combined method. In this section, we observe the system performance from 7 settings: 1) VID: video feature (classified LogitBoost) only. 2) ACC: acceleration feature (SVM) only. 3) LBAV: LogitBoost Classifier on one combined feature vector of acceleration and vision. 4) CRFV: Apply CRF MODEL on top of setting 1. (5) CRFA: Use of CRF on setting 2. 6) CRFLB: CRF on setting 3. 7) CRFAV: Integrates both feature classifiers with a CRF, as explained in Equation 9 and 10. We plot the error-bar for all configurations in Figure 7. This includes the average Precision, Recall and F-Score [39] of all 12 activities. We also plot the overall accuracy over the datasets. Note that the overall accuracy is generally higher than others. This is due to the unbalanced distribution of different activities. e.g. video contains more 'Walking' than 'Hand-Washing'.

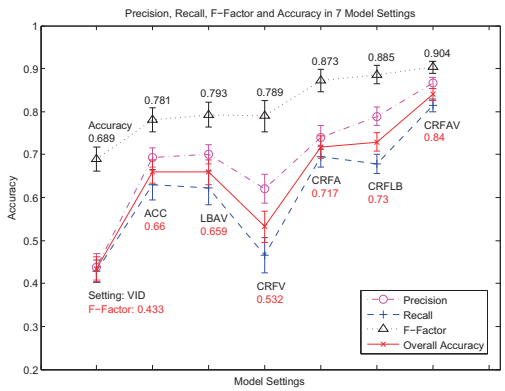


Fig. 7. System performance from 7 model configurations, the darker the diagonal line, the better the system performance.

This plot shows the benefits when the model is integrated with the CRF. Settings 1-3 use local classifier only, and 4-6 are with an additional CRF structural learning process. It increases the system overall accuracy by an average of 10.5% AND improves the F-Score by 7.6%. Notably, the CRFAV algorithm achieves the best result with an accuracy of 90.38% and F-Score of 84.45%. It also indicates that video features are not as good as acceleration features when used on ADLs. This might be caused by the unbalanced distribution between locomotive motions and stationary activities from our ADLs database. To be more specific, we include 4 confusion matrices in Figure 8.

The top two matrices show the local classifier results. The dynamic activities are misclassified with video features only, including climbing stairs, sit-down and stand-up. The acceleration feature classifier has relatively poor results in hand activities. Mistakes occur among sitting, watching and reading. The 'CRFLB' matrix removes most of the incorrect predictions, has fewer outliers between stationary activities, as well as between switching water-tap and hand-washing. Overall, 'CRFAV' is the best option. It removes most of

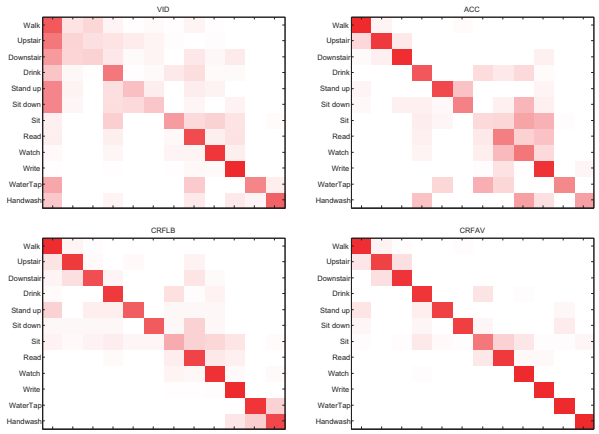


Fig. 8. Confusion matrices for 4 settings: 'VID', 'ACC', 'CRFLB' and 'CRFAV' (see Section III D for details). The colour intensity represents the magnitude of the predicted and correct assignment.

the outliers, and it only has minor errors between up and downstairs, sitting and reading.

### *E. Discussion*

In this study, we explore the use of context learning in the area of ADLs. In real life, recognising varied and realistic human activities is a challenge in terms of reliability. We address the problem for ADLs recognition, and our experimental results confirm the benefits of multi-scale CRFs compared to traditional classification approaches. This involves a distance inference method in the graphical model, and a feature function integration method. Multi-scale context improves the performance by learning the subject's ADLs sequence with temporal correlations. The feature integration method allows the system to learn the relative weights among different sources, as well as linking local features into a global network. The model boosts the local classification by an average of 10%-20%, and improves the accuracy to nearly 90%. Note that our model can now be implemented from the Smart-Glasses prototype with a 1GHz CPU processor and 512MB RAM memory. The tasks, such as 'video/sensor data collection', 'video/sensor features extraction' and 'local feature classifications' can be executed almost instantaneously. However, the TRBP inference procedure generally requires an average of 2-3 seconds delay from a 30-second updating block, which is equivalent to 20 window slides.

## IV. CONCLUSION AND FUTURE WORK

This work presented a novel activity recognition approach from first person perspective. The algorithm is developed using a CRF model, which exploits two important factors for ADLs recognition, feature interpretation and contextual structure, in order to cover a wide range of human activities. We utilise the head acceleration and egocentric vision as our primary sources of information, which have great potential to capture general body motion and hand activities. Embedding the sensors into glasses makes the system simple and of easy adoption for elderly populations.

In this research, we annotate, train, and validate our method using a large realistic ADL dataset covering several motion types, different environmental settings and various locations. Results demonstrate the model outperforms a number of existing methods, and the system is tested and proved reliable in both indoor and outdoor environments. In future work, we will investigate the robustness of our model in more challenging activities of disable and elder patients, most of them receiving rehabilitation treatment following illness or injury. We will validate our model through a cross-person patient platform from different illness/injury categories.

## REFERENCES


[1] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," *Pervasive Computing, IEEE*, pp. 1–17, 2004.

[2] T. Brezmes, J.-L. Gorricho, and J. Cotrina, *Activity Recognition from Accelerometer Data on a Mobile Phone Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5518, pp. 796–799.

[3] Y. Chen, J. Qi, Z. Sun, and Q. Ning, "Mining user goals for indoor location based services with low energy and high qos," *Computational Intelligence*, vol. 26, no. 3, pp. 318–336, 2010.

[4] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *IJCAI 2011*, 2011, pp. 2545–2550.

[5] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering and Physics*, vol. 30, no. 1, pp. 84–90, 2008.

[6] N. H. Chehade, P. Ozisik, J. Gomez, F. Ramos, and G. Pottie, "Detecting stumbles with a single accelerometer," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, 2012, pp. 6681–6686.

[7] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, *Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, ch. 30, pp. 216–223.

[8] J. Ward, P. Lukowicz, G. Trster, and T. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1553–1567, 2006.

[9] M. Sousa, A. Techmer, A. Steinhage, C. Lauterbach, and P. Lukowicz, "Human tracking and identification using a sensitive floor and wearable accelerometers," in *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, 2013, pp. 166–171.

[10] L. Wang, L. Cheng, T. H. Thi, and J. Zhang, "Human action recognition from boosted pose estimation," in *Proceedings of the 2010 International Conference on Digital Image Computing: Techniques and Applications*, ser. DICTA '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 308–313.

[11] B. Yao, A. Khosla, and L. Fei-Fei, "Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses," in *International Conference on Machine Learning (ICML)*, 2011.

[12] W. Shandong, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1419–1426.

[13] L. Yong Jae, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1346–1353.

[14] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1226–1233.

[15] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2847–2854.

[16] A. Fathi, R. Xiaofeng, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3281–3288.

[17] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Fast unsupervised ego-action learning for first-person sports videos," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3241–3248.

[18] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (darpa)," in *Proceedings of the 1981 DARPA Image Understanding Workshop*, April 1981, pp. 121–130.

[19] K. Zhan, F. Ramos, and S. Faux, "Activity recognition from a wearable camera," in *2012 12th International Conference on Control Automation Robotics and Vision (ICARCV)*, 2012, pp. 365–370.

[20] Y. Nam and J. W. Park, "Child activity recognition based on cooperative fusion model of a triaxial accelerometer and a barometric pressure sensor," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 2, pp. 420–426, 2013.

[21] H. Zhenyu, "Activity recognition from accelerometer signals based on wavelet-ar model," in *2010 IEEE International Conference on Progress in Informatics and Computing (PIC)*, vol. 1, 2010, pp. 499–502.

[22] J.-K. Min and S.-B. Cho, "Activity recognition based on wearable sensors using selection/fusion hybrid ensemble," in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 1319–1324.

[23] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, "Multisensor data fusion for physical activity assessment," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 3, pp. 687–696, 2012.

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

[25] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[26] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *The Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[27] R. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[28] S. B. Kotsiantis, "Bagging and boosting variants for handling classifications problems: a survey," *The Knowledge Engineering Review*, vol. FirstView, pp. 1–23, 2013.

[29] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Proceedings of the 5th Iberian conference on Pattern recognition and image analysis*. Las Palmas de Gran Canaria, Spain: Springer-Verlag, 2011, pp. 289–296.

[30] Y. Jongmin and K. Daijin, "Frontal face classifier using adaboost with mct features," in *2010 11th International Conference on Control Automation Robotics and Vision (ICARCV)*, 2010, pp. 2084–2087.

[31] Y. Cai, K. Feng, W. Lu, and K. Chou, "Using logitboost classifier to predict protein structural classes," *Journal of theoretical biology*, vol. 238, no. 1, pp. 172–176, 2006.

[32] J. Friedman, T. Hastie, and R. Tibshirani, "Special invited paper. additive logistic regression: A statistical view of boosting," *The annals of statistics*, vol. 28, no. 2, pp. 337–374, 2000.

[33] C. Sutton and A. McCallum, "An introduction to conditional random fields," *arXiv preprint arXiv:1011.4088*, 2010.

[34] J. Besag, "Statistical analysis of non-lattice data," *The statistician*, pp. 179–195, 1975.

[35] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[36] J. Pearl, "Reverend bayes on inference engines: A distributed hierarchical approach," in *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, Pittsburgh, PA, 1982, pp. 133–136.

[37] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

[38] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.

[39] C. van Rijsbergen, *Information Retrieval. 1979*. Butterworth, 1979.