DELIVERABLE 4.4

Data fusion and activity recognition in household chores

Author(s): Ninghang Hu, Richard Bormann, Ben Kröse

Project no: 287624

Project acronym: ACCOMPANY

Project title: Acceptable robotiCs COMPanions for AgeiNg Years

Doc. Status:	**Final**
Doc. Nature:	**Report**
Version:	**0.4**
Actual date of delivery:	**15 October 2013**
Contractual date of delivery:	**Month 24**
Project start date:	**01/10/2011**
Project duration:	**36 months**
Peer Reviewer:	**UH**

Project Acronym: ACCOMPANY
Project Title: **Acceptable robotiCs COMPanions for AgeiNg Years**

EUROPEAN COMMISSION, FP7-ICT-2011-07, 7th FRAMEWORK PROGRAMME
ICT Call 7 - Objective 5.4 for Ageing & Wellbeing

Grant Agreement Number: 287624

DOCUMENT HISTORY

Version	Date	Status	Changes	Author(s)
0.4	2013-10-14	Draft	Abstract	Ben Kröse
0.3	2013-10-14	Draft	Future Work	Ninghang
0.2	2013-10-13	Draft	Intro and Conclusion	Ben Kröse
0.1	2013-10-8	Draft		Ninghang Hu
0.0	2013-10-8	Draft	Initial Draft	Ben Kröse

AUTHORS & CONTRIBUTERS

Partner Acronym	Partner Full Name	Person
UvA	University of Amsterdam	Ben Kröse
UvA	University of Amsterdam	Ninghang Hu
IPA	Fraunhofer IPA	Richard Bormann

Short description

This deliverable reports on the data fusion and the activity recognition in household chores in WP4 of the ACCOMPANY project.

At the beginning of the project we focused on data fusion for person detection and localization. In the second year we extended the person detection to human posture recognition. The basis of our work is the use of probabilistic graphical models. For the posture recognition with a top view camera we developed a novel method for posture recognition. When compared to a state-of-the-art approach of pose estimation, our posture descriptor does much better. The results show that our method is able to correctly classify 79.7% of the test sample, which outperforms the conventional approach by over 23%.

We also worked on a more robust person detection and identification, which is needed in a multi-user environment. We developed a system that seamlessly integrates the information from the robot camera and fixed external top view camera. The results show improved efficiency when the robot system is aided by the localization system of the overhead cameras.

Most effort was on our research on activity recognition. We developed a novel discriminative model for the recognition of human activities. The novel model was tested on the (CAD-120 benchmark standard benchmark data set. Experimental results on this data set indicate that our model outperforms the current state-of-the-art approach by over 5% in both precision and recall, while our model is more efficient in terms of computation.

Table of Contents

1 Introduction

This deliverable focuses on the activity recognition in the ACCOMPANY project. In the previous deliverables we focused on the fusion of depth data (either with Kinect or with a laser scanner) with the image information (D4.2) for object detection and person detection. In deliverable D4.3 we presented a system design that was targeted to combine head tracking, and head pose estimation in order to get a more robust localization and posture estimation system.

In year 2 of the project we continued with the posture estimation. The environment where we do the ACCOMPANY experiments is equipped with a top-view camera for monitoring human activities. This setup is very useful because top-view cameras can provide accurate localization and also limit inter-occlusion between persons. However, they also suffer from body parts being frequently self-occluded. Conventionally, posture recognition relies on good estimations of body part positions, which turns out to be unstable in the top-view due to occlusion and foreshortening. In our approach, we learn a posture descriptor for each specific posture category. The posture descriptor encodes how well the person in the image can be 'explained' by the model. The postures are subsequently recognized from the matching scores returned by the posture descriptors. In chapter 2 of this report we describe the model we developed.

We also worked on data fusion, where we made progress in the fusion of information from the top-view camera and the camera on the Care-o-Bot. Finding people is one of the most fundamental tasks in robot home care scenarios and it consists of many components (e.g. people detection, people tracking, face recognition, robot navigation etc.). Researchers have extensively worked on these components in isolation. But surprisingly, little attention has been paid on bridging these components as an entire system. In chapter 3 we describe our system and the evaluation of the entire system in a robot-care scenario.

The most important part of the work was carried out in activity recognition. Originally we planned to apply HMM and DBN to the fusion of data from sensor networks and image information, we decided to focus on activity recognition from the visual modality. The reason for this is that we wanted to go beyond the state-of-the-art in activity recognition algorithms, and explore novel methods. Probabilistic Graphical Models have been widely used for recognizing human activities in both robotics and smart home scenarios. The graphical models can be divided into two categories: generative models and discriminative models. The generative models require making of assumptions on both the correlation of data and on how the data is distributed given the activity state. The risk is that the assumptions may not reflect the true attributes of the data. The robotic and smart environment scenario environments are usually equipped with a combination of multiple sensors. Some of these sensors may be highly correlated, both in the temporal and spatial domain (e.g. a pressure sensor on the mattress and a motion sensor above the bed). , In contrast, the discriminative models only focus on modeling the posterior probability regardless of how the data are distributed. In our scenarios, the discriminative models provide us with a natural way of

implementing data fusion for human activity recognition. In chapter 4 we describe a novel discriminative model for the recognition of human activities.

The report is structured as follows: Chapter 2 describes our work on pose estimation. A paper on this work has been accepted for IROS13. Chapter 3 describes our system that integrates the information from the overhead camera with the camera on the Care-O-bot. This work has been submitted to ICRA14. Chapter 4 describes our new approach for activity recognition. This work has also been submitted to ICRA14. The full papers and submissions are attached as appendices A, B and C.

2 Posture Recognition with a Top-view Camera

Human posture recognition is one of the most important tasks for human-robot interactions (HRI), as it provides a solid base for human activity recognition [5]–[7]. There are many papers on recognizing human posture with robot sensors or ambient cameras in 2D [8], 2.5D (RGB+Depth) [9] and 3D [10]. Most of the 2D approaches observe humans from a side-view however, and recognizing human posture from the top-view still remains a challenging and unsolved problem.

For the purposes of our work, posture recognition is defined as the process of assigning semantic posture labels to people in an image (e.g. whether people are standing, sitting, bending or pointing). In contrast, pose estimation estimates the configuration of the body parts [11], and so focuses on getting accurate body part locations rather than on posture labeling. Similarly, pose refers to a configuration of the body parts, and posture refers to a category of poses that bear the same semantic label.

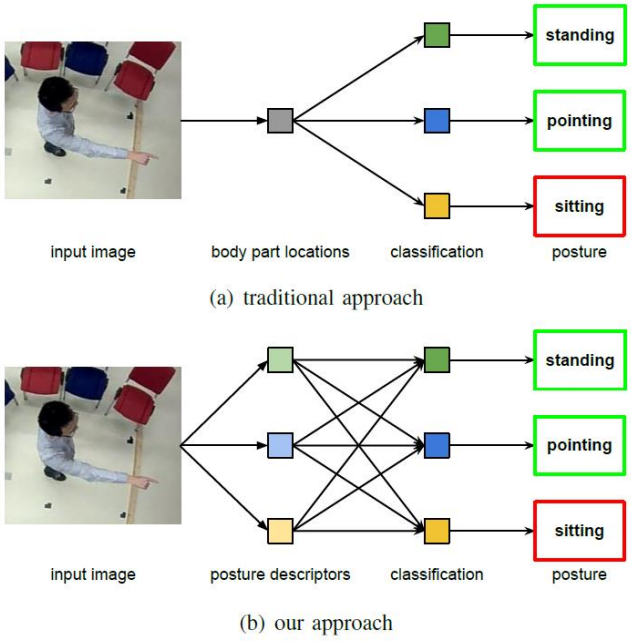


Figure 1 A comparison between the (a) traditional approach and (b) our approach.

In this section, we present a system that recognizes human postures from the still images captured by a top-view camera. An overview of our system is shown in Figure 1. Our interest in this problem stems from a robot assisted living scenario, where we use ceiling-mounted cameras as part of a domestic monitoring system to inform the robot on the human activities. Compared with robot-mounted sensors, top-view cameras give a good overview of the overall scene and a large amount of information about the person. Also, top-view cameras provide a better estimation of the human locations and allow for far less inter-occlusion

between persons when compared to side-views and robot mounted sensors. Top-view cameras do, however, suffer more from self-occlusion as compared to side view cameras.

(a) top-view (b) side-view

Figure 2: Posture recognition from the top-view (a) is a more challenging

We distinguish between two types of occlusion: inter-occlusion and self-occlusion. Inter-occlusion refers to an object being blocked by another. For example, the view of a person can be partially blocked by the person in front. In contrast, self-occlusion means that the object is occluded by itself (e.g. the limbs are occluded by the head and the torso). The two types of occlusions both happen in the side-view and top-view, but at different levels. The inter-occlusion is more frequent in the side-view because other persons also stand at the same height level. In contrast, self-occlusion is more severe in the top-view (see Figure 2). Most literature on posture recognition addresses the side-view and neglects the problem of top-view occlusion.

In our work, we focus on recognizing human postures under the severe self-occlusion seen in top-view images. The conventional approach is to firstly estimate the human pose configuration, and then classifies postures based on the body part positions [8]. In top-view images, people are largely self-occluded. With little information about the body part locations, recovering an articulated pose from these images is a difficult task even for human annotators, let alone to further derive the posture category based on the ambiguous body part locations.

Recent work shows that, when the joint positions are accurately known, the best performance in posture recognition is obtained from the 3D joint positions [8]. In our approach, we recognize the human posture without explicitly knowing the exact location of body parts, and we will show that, in the case of heavy self-occlusion, this approach outperforms joint position based posture recognition. Unlike the conventional approach which classifies postures based on the body part locations, our idea is to use posture descriptors instead for classification. A posture descriptor provides a mapping from image features to the matching score of a posture category. Given a new test image, each posture descriptor gives a matching score that measures how well the person can be explained by that posture descriptor. For example, the standing posture descriptor returns a higher value when applied to standing people, and lower values on the others. Note that the posture categories overlap.

For instance, a standing person may be also pointing. Our posture descriptors encode such attributes in a natural way by enabling multiple data labels to be applied to a single image. Figure 1 compares our proposed system with the conventional approach.

In this work, we address the following research questions:

1) Is 2D pose competitive with 3D pose for posture recognition? Posture recognition from (perfect) 3D pose has been shown to outperform appearance-based approaches. We show that the performance of posture recognition with 2D pose is virtually identical to 3D pose, including those derived from top-view image projections.
2) How accurately can we obtain 2D pose from top-view images? To investigate this, we apply a state-of-the-art 2D pose estimation algorithm to the top-view images. We show that the performance is generally very low, but the specific models that are trained on a particular posture category perform comparably better.
3) How accurately can we recognize posture from imperfect 2D pose, and how does this performance compare to our proposed model? We show that our proposed model based on posture descriptors significantly outperforms the baseline, which consists of two state-of-the-art approaches.

For details of the paper, please refer to Appendix A.

3 Multi-User Identification by Fusing Robot and Ambient Sensors

Two fundamental tasks in robot home care scenarios are people localization and people identification. They are also the elemental components for more advanced tasks such as activity recognition [1]. In recent years, researchers have been extensively working on the tasks of people detection [2], people tracking [3], face recognition [4], robot navigation, and robot controls, but mainly as isolated tasks instead of combining these systems for real life applications.

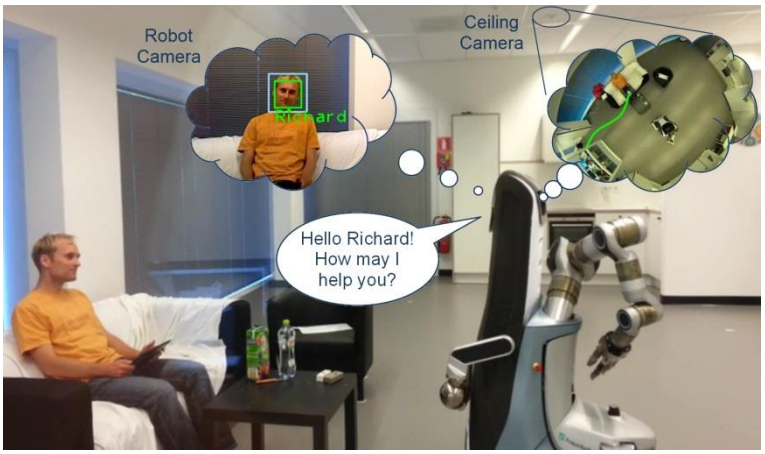


Figure 3 Fusion of robot and environment cameras for direct user approach

In our work, we study these tasks jointly, and we introduce a unified system that integrates these components in our scenarios, see Figure 3. The system is very efficient and suitable for real-time applications. Moreover, the components are complementary to help improving the robustness of the entire system. Commonly used sensors for these tasks include overhead cameras and RGB-D sensors on mobile robots. The overhead cameras are usually fixed at the ceiling, covering most of the areas in the room. The cameras only need to be calibrated once so that the coordinates of the detected person can be transformed easily from the image space to the ground-plane of the room. As the camera is mounted on the ceiling, people in the video are less likely to be occluded by each other. The overhead camera commonly has a wide field of view. Thereby one camera is often sufficient for detecting and tracking people in the whole room. Despite these benefits, it is very difficult for the overhead camera to recognize people's identity. Faces can hardly be seen at many locations. The most prominent parts of people are the clothes, but they may be changed from session to session.

The overhead camera may be enough to locate a person, but it is not sufficient for people identification. The sensors on the robot, (e.g. Microsoft Kinect etc.) provide a complementary view to the overhead camera. The on-board cameras are commonly mounted at a level that keeps the human face in sight. The RGB-D sensor provides both the color image from a color camera and the depth image from a range camera. By fusion of the depth image and color image, a face can be recognized robustly [4]. However, the RGB-D sensor is limited in both the range and the view angle. When people are too close, the face is outside the field of

view; when they are far away, the accuracy and resolution of face data drops quickly. An advantage of the combination with ceiling cameras for tracking is that the robot itself does not need to keep monitoring the persons all the time. Hence, the robot may carry out other tasks, rather than allocating its resources to the task of tracking each person. In this section, we introduce the system that is used in the ACCOMPANY project.

The architecture of the proposed system is shown in Figure 4. Our system consists of three modules, a) people detection and tracking, b) people identification, and c) a joint tracker that combines both of the systems. The first module finds multiple people that are present in the room using two overhead cameras. The background probabilities are modeled with a dynamic probabilistic background model. Using the background model, people in the room are detected with a Bayesian people detector. After that the detection is associated with the tracks by comparing cues of appearance and the positions. The second module identifies people using a Kinect sensor that is mounted on the robot. We apply a fast search for all head-shaped objects using the depth camera, generating a set of candidate face locations. These candidate locations are evaluated in the color image for face detection. Once the candidate is verified as a face, features are extracted from the face for face identification.

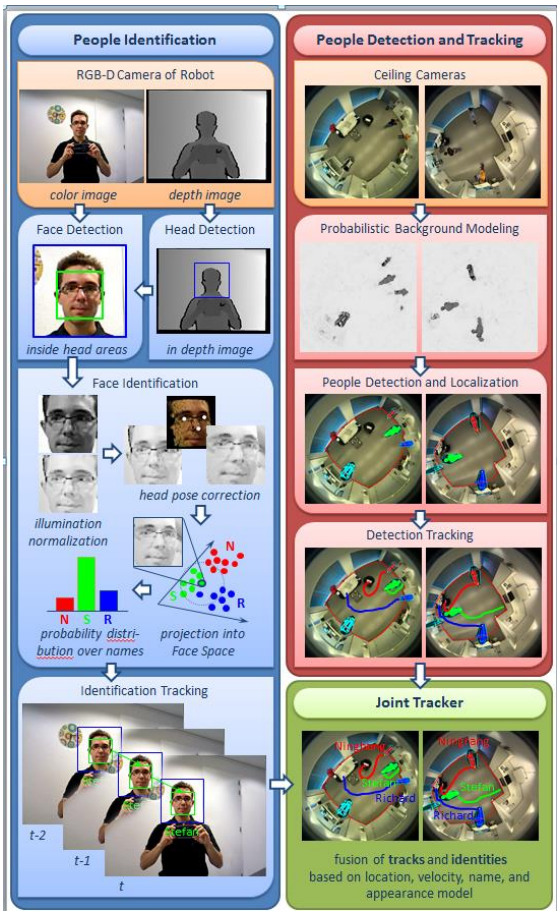


Figure 4 System overview

The third module collects information from the first two modules and associates tracks with human identities. Every time a new person is recognized, the joint tracker finds the closest tracks and labels the track with the respective name.

For details of the paper, please refer to Appendix B.

4 Learning Latent Structure for Activity Recognition

Robotic companions which help people in their daily life are currently a widely studied topic. In Human-Robot Interaction (HRI), it is very important that the human activities are recognized accurately and efficiently.

In this section, we present a novel graphical model for human activity recognition. The task of activity recognition is to find the most likely underlying activity sequence based on the observations generated from the sensors. Typical sensors include ambient cameras, contact switches, thermometers, pressure sensors, and the sensors on the robot, e.g. RGB-D sensor and Laser Range Finder.

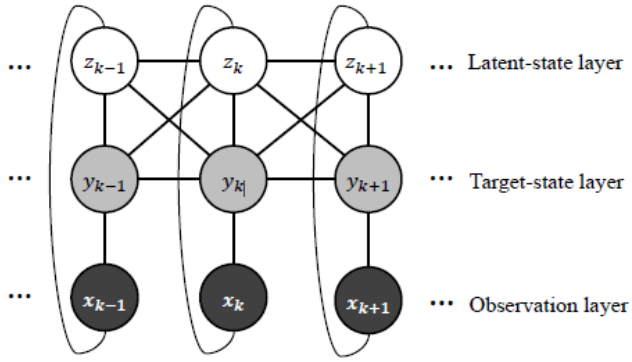


Figure 5: the proposed graphical model

Probabilistic Graphical Models have been widely used for recognizing human activities in both robotics and smart home scenarios. The graphical models can be divided into two categories: generative models [12], [13] and discriminative models [1], [14], [15]. The generative models require making assumptions on both the correlation of data and on how the data is distributed given the activity state. The risk is that the assumptions may not reflect the true attributes of the data. The discriminative models, in contrast, only focus on modeling the posterior probability regardless of how the data are distributed. The robotic and smart environment scenarios are usually equipped with a combination of multiple sensors. Some of these sensors may be highly correlated, both in the temporal and spatial domain, e.g. a pressure sensor on the mattress and a motion sensor above the bed. In these scenarios, the discriminative models provide us a natural way of data fusion for human activity recognition.

The linear-chain Conditional Random Field (CRF) is one of the most popular discriminative models and has been used for many applications. Linear-chain CRFs are efficient models because the exact inference is tractable. However, they are limited in the way that they cannot capture the intermediate structures within the target states [16]. By adding an extra layer of latent variables, the model allows for more flexibility and therefore it can be used for modeling more complex data. The names of these models are interchangeable in the literature, such as Hidden-Unit CRF [17], Hidden-state CRF [16] or Hidden CRF [18].

In this section, we present a latent CRF model for human activity recognition. For simplicity, we use “latent variables” to refer to the augmented hidden layer, as they are unknown either in training or testing. The “target variables”, which is observed during training but not testing, represent the target states that we would like to predict, e.g. the activity labels. See Figure 5 for the graphical model and the difference between latent variables and target variables. We evaluate the model using the RGB-D data from the benchmark dataset [14]. The results show that our model performs better than the state-of- the-art approach [14], while the model is more efficient in inference.

Our contributions can be summarized as follows:

1) We propose a novel Hidden CRF model for predicting underlying labels based on the sequential data. For each temporal segment, we exploit the full connectivity among observations, latent variables, and the target variables, from which we can avoid making inappropriate conditional independence assumptions.
2) We show an efficient way of applying exact inference in our graph. By collapsing the latent states and the target states, our graphical model can be considered as a linear-chain structure. Applying exact inference under such a structure is very efficient.
3) Our software is open source and will be fully available for comparison.

Details of this work can be found in Appendix C.

5 Conclusion and Future Work

The problem of posture detection from a top-view camera was studied and a novel method for posture recognition was developed. When compared to a state-of-the-art approach of pose estimation our posture descriptor does much better. The results show that our method is able to correctly classify 79.7% of the test sample, which outperforms the conventional approach by over 23%.

The identification and localization of multiple persons in a robot home setting was solved by developing a system that seamlessly integrates the information from robot camera and top view camera. The results show largely improved efficiency when the robot system is aided by the localization system of the overhead cameras.

The novel model for activity recognition was tested on a standard benchmark data set (CAD-120 benchmark). Experimental results on this data set show that our model outperforms the state-of-the-art approach by over 5% in both precision and recall, while our model is more efficient in computation.

We are currently extending our system with a hierarchical model that can jointly estimate both high-level activities (e.g. cooking, drinking, etc.) and low-level activities (e.g. grasping, placing, eating, drinking, etc.). As the observations may not be complete in practice, we will also look into developing a model that can handle partially observed data.

6 References

[1] N. Hu, G. Englebienne, and B. Krose, “Posture Recognition with a Top-view Camera,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[2] G. Gate, A. Breheret, and F. Nashashibi, “Centralized fusion for fast people detection in dense environment,” in *Proc. IEEE International Conference on Robotics and Automation*, 2009, pp. 76–81.

[3] M. Montemerlo, S. Thrun, and W. Whittaker, “Conditional particle filters for simultaneous mobile robot localization and people-tracking,” in *Proc. IEEE International Conference on Robotics and Automation*, 2002, vol. 1, pp. 695–701.

[4] R. Bormann, T. Zwölfer, J. Fischer, J. Hampp, and M. Hägele, “Person Recognition for Service Robotics Applications,” in *accepted for publication at the 13th International IEEE-RAS International Conference on Humanoid Robots*, 2013.

[5] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, “People Watching: Human Actions as a Cue for Single-View Geometry,” in *ECCV*, 2012.

[6] B. Yao and L. Fei-Fei, “Modeling mutual context of object and human pose in human-object interaction activities,” in *CVPR*, 2010, pp. 17–24.

[7] A. Yao, J. Gall, and L. Van Gool, “A hough transform-based voting framework for action recognition,” in *CVPR*, 2010, pp. 2061–2068.

[8] J. P. Wachs, D. Goshorn, and M. Kölsch, “Recognizing human postures and poses in monocular still images,” in *IPCV*, 2009.

[9] E. Weng and L. Fu, “On-line human action recognition by combining joint tracking and key pose recognition,” in *IROS*, 2012, pp. 4112–4117.

[10] N. Zouba, B. Boulay, F. Bremond, and M. Thonnat, “Monitoring activities of daily living of elderly based on 3D key human postures,” *Cogn. Vis.*, pp. 37–50, 2008.

[11] L. Sigal, “Human Pose Estimation,” *Encycl. Comput. Vis.*, 2011.

[12] C. Zhu and W. Sheng, “Human Daily Activity Recognition in Robot-assisted Living using Multi-sensor Fusion,” in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2154–2159.

[13] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Unstructured human activity detection from rgbd images,” in *International Conference on Robotics and Automation (ICRA)*, 2012, pp. 842–849.

[14] H. S. Koppula, R. Gupta, and A. Saxena, “Learning Human Activities and Object Affordances from RGB-D Videos,” *Int. J. Robot. Res.*, 2012.

[15] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, “Activity recognition using semi-markov models on real world smart home datasets,” *J. Ambient Intell. Smart Environ.*, vol. 2, no. 3, pp. 311–325, 2010.

[16] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, “Hidden Conditional Random Fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848–1852, 2007.

[17] L. Maaten, M. Welling, and L. Saul, “Hidden-Unit Conditional Random Fields,” *Int. Conf. Artif. Intell. Stat.*, pp. 479–488, 2011.

[18] Y. Wang and G. Mori, “Max-margin hidden conditional random fields for human action recognition,” in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 872–879.

Appendix A

Posture Recognition with a Top-view Camera

Ninghang Hu[1], Gwenn Englebienne[1], and Ben Kröse[1,2]

***Abstract*— We describe a system that recognizes human postures with heavy self-occlusion. In particular, we address posture recognition in a robot assisted-living scenario, where the environment is equipped with a top-view camera for monitoring human activities. This setup is very useful because top-view cameras lead to accurate localization and limited inter-occlusion between persons, but conversely they suffer from body parts being frequently self-occluded. The conventional way of posture recognition relies on good estimation of body part positions, which turns out to be unstable in the top-view due to occlusion and foreshortening. In our approach, we learn a *posture descriptor* for each specific posture category. The posture descriptor encodes how well the person in the image can be 'explained' by the model. The postures are subsequently recognized from the matching scores returned by the posture descriptors. We select the state-of-the-art approach of pose estimation as our posture descriptor. The results show that our method is able to correctly classify 79.7% of the test sample, which outperforms the conventional approach by over 23%.**

I. INTRODUCTION

Human posture recognition is one of the most important tasks for human-robot interactions (HRI), as it provides a solid base for human activity recognition [1], [2], [3]. There are many papers on recognizing human posture with robot sensors or ambient cameras in 2D [4], 2.5D (RGB+Depth) [5] and 3D [6]. Most of the 2D approaches observe humans from a side-view, however, and recognizing human posture from the top-view still remains a challenging and unsolved problem.

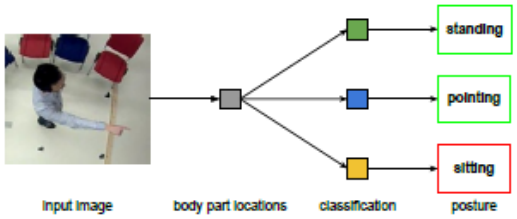
For the purposes of this paper, *posture recognition* is defined as the process of assigning semantic posture labels to people in an image, *e.g.* whether people are standing, sitting, bending or pointing. In contrast, *pose estimation* is to estimate the configuration of the body parts [7], which focuses on getting the accurate body part locations rather than on posture labeling. Similarly, *pose* refers to a configuration of the body parts, and *posture* refers to a category of poses that bare the same semantic label.

In this paper, we present a system that recognizes human postures from the still images captured by a top-view camera. An overview of our system is shown in Fig. 1b. Our interest in this problem stems from a robot assisted-living scenario, where we use ceiling-mounted cameras as part of a domestic monitoring system to inform the robot on the human activities. Compared with robot-mounted sensors,

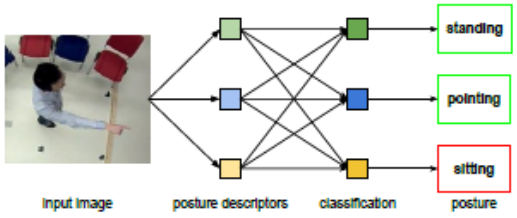
The research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287624, and partly from the SIA project BALANCE-IT.

[1] N. Hu, G. Englebienne, and B. Kröse are with Intelligent System Lab Amsterdam, University of Amsterdam, 1098XH Amsterdam, The Netherlands {n.hu,g.englebienne,b.j.a.krose} @ uva.nl

[2] B. Kröse is also with the Amsterdam University of Applied Science



(a) traditional approach



(b) our approach

Fig. 1. A comparison of the (a) traditional approach and (b) our approach. The traditional approach classifies posture categories based on the estimated body part locations. In contrast, our proposed system uses the matching scores from the *posture descriptors*. By combining the matching scores from all posture descriptors into a single feature vector, we apply a binary classifier to determine whether the image belongs to a certain posture category or not.

top-view cameras give a good overview of the overall scene and a large amount of information about the person. Besides, the top-view cameras also provide a better estimation of the human locations and allow for far less inter-occlusion between persons when compared to side-views and robot-mounted sensors. Top-view cameras do, however, suffer more from a different form of occlusion than side-view cameras, *i.e.* self-occlusion.

We distinguish between two types of occlusion: inter-occlusion and self-occlusion. Inter-occlusion refers to an object being blocked by another, *e.g.* when the view of a person is partially blocked by the person in front. In contrast, self-occlusion means that the object is occluded by itself, *e.g.* the limbs are occluded by the head and the torso. The two types of occlusions both happen in the side-view and the top-view, but at different levels. The inter-occlusion is more frequent in the side-view because other persons also stand at the same height level. In contrast, self-occlusion is more severe in the top-view (see Fig. 2). Most literature on posture recognition addresses the side-view and neglects the problem of the top-view.

In this paper, we focus on recognizing human postures

(a) top-view (b) side-view

Fig. 2. Posture recognition from the top-view (a) is a more challenging task than from the side-view (b) due to the severe self-occlusion.

under the severe self-occlusion seen in top-view images. The conventional approach is to firstly estimate the human pose configuration, and then classifies postures based on the body part positions [8]. In top-view images, people are largely self-occluded. With little information about the body part locations, recovering an articulated pose from these images is already a quite difficult task even for human annotators, let alone to further derive the posture category based on the ambiguous body part locations.

Recent work shows that, when the joint positions are accurately known, the best performance in posture recognition is obtained from the 3D joint positions [8]. In our approach, we recognize the human posture without explicitly knowing the exact location of body parts, and we will show that, in the case of heavy self-occlusion, this approach outperforms joint position based posture recognition. Unlike the conventional approach which classifies postures based on the body part locations, our idea is to use *posture descriptors* instead for classification. A posture descriptor provides a mapping from image features to the matching score of a posture category. Given a new test image, each posture descriptor gives a matching score that measures how well the person can be explained by that posture descriptor. For example, the standing posture descriptor returns a higher value when applied to standing people, and lower values on the others. Note that the posture categories overlap. For instance, a standing person may be also pointing. Our posture descriptors encode such attributes in a natural way by enabling multiple data labels to be applied to a single image. Fig. 1 compares our proposed system with the conventional approach.

In this work, we address the following research questions:

1) *Is 2D pose competitive with 3D pose for posture recognition?* Posture recognition from (perfect) 3D pose has been shown to outperform appearance-based approaches. We show that the performance of posture recognition with 2D pose is virtually identical to 3D pose, including for top-view projections.
2) *How accurately can we obtain 2D pose from top-view images?* To investigate this, we apply a state-of-the-art 2D pose estimation algorithm to the top-view images. We show that the performance is generally very low, but the specific models that are trained on a particular posture category perform comparably better.
3) *How accurately can we recognize posture from imperfect 2D pose, and how does this performance compare to our proposed model?* We show that our proposed model based on posture descriptors significantly outperforms the baseline, which consists of two state-of-the-art approaches.

II. RELATED WORK

Previous work on human posture recognition is mostly based on the images taken from the side-view. The top-view, which has been extensively used in domestic monitoring, receives surprisingly little attention.

Only recently did researchers start to work on the top-view to classify human postures [9], [10]. These approaches use the silhouettes of humans, which are extracted by background subtraction and represented as a vector of features. The features include the height-width ratio, the position, and the polar histograms of the silhouettes. These approaches rely on accurate foreground-background segmentation, which is difficult to obtain in practice due to noise, the change of lighting conditions or incorrect segmentation of the foreground blobs.

The more conventional method of posture recognition relies on side-view images to perform pose estimation and then predicts posture categories based on the estimated articulated pose. The state-of-the-art approach estimates body part locations using the Histogram of Oriented Gradient (HOG) features [11], and fits a human skeleton model to still images. In the human skeleton model, the joints of articulations are represented as body part detectors, and two joints are positioned in a way that the deformation costs are minimized.

To perform posture recognition, [8] assumes that the body part locations are known and transforms the 3D body part locations into a feature vector of geometric distances. The postures are then recognized using a random forest. The results are compared with the approach in [3], where a Hough Forest [12] was trained to learn the mapping from appearance patches to action labels. The results show that the pose-based distance features outperform the appearance-based features.

From the top-view, the body parts become largely self-occluded, which makes conventional approaches less suitable. It is very difficult to estimate the body part locations accurately from top-view images, and the resulting posture recognition performance is substandard. To solve this problem, we perform posture recognition with the matching scores from [11] instead of using the estimated poses. In this way, the exact body part locations need not to be extracted accurately for recognizing the postures.

III. APPROACH

Our system consists of two parts. First the image is transformed into a vector of posture scores by using the posture descriptors. These posture scores are then used as features by a posture classifier, which returns the final posture label.

A. Posture Descriptor

The posture descriptor is a component that transforms the input image into a vector of features that can be used for posture recognition. Normally, the posture descriptor consists of body part locations [8] or transformed low-level features [13]. In this paper, we capture the posture descriptor at a higher level. Each posture descriptor is a measurement of how likely the input image belongs to a certain posture category. Specifically, we adopt the posture descriptor from the state-of-the-art approach in human pose estimation [11], where the poses are estimated by finding the optimal skeleton configuration with respect to the local body part detection. Similar to the structure of a Support Vector Machine (SVM), each posture descriptor returns a matching score along with the estimated body part positions. Since the body part positions are often unknown due to the occlusion, we disregard them and use only the matching score to perform posture recognition in our approach.

We now formulate the problem and give a brief introduction to the posture descriptor. For more details, please refer to [11].

Let I be an input image, and k is a posture category that follows $k \in \{1, ..., K\}$. Given the input image, each posture descriptor gives a matching score $S_k(I)$ by maximizing the energy function $Q_k(I, l, t)$ over all possible body part locations L and all types of the body parts T

$$S_k(I) = \max_{l \in L, t \in T} Q_k(I, l, t) \tag{1}$$

where l is a vector of body part locations in the discretized image space and t is a vector of type assignments over all the body parts.

Solving a general problem of (1) takes exponential time. But when Q_k are computed within a tree structure, the non-maximum suppression of the function can be computed efficiently using dynamic programming [14]. We define a tree structure following the human skeleton, where the vertices V of the tree are the body parts and the edges E are the pair-wise connections between the vertices.

We write the energy function of the tree structure as

$$Q_k(I, l, t) = \sum_{i \in V} \omega_i^{t_i} \cdot \phi(I, l_i) + \sum_{ij \in E} \omega_{ij}^{t_i t_j} \cdot \psi(l_i, l_j) + S(t) \tag{2}$$

where $\omega_i^{t_i} \cdot \phi(I, l_i)$ is a linear filter of the body parts. It gives high scores if the image at location l_i looks like the type t_i of the i^{th} body part. The second term $\omega_{ij}^{t_i t_j} \cdot \psi(l_i, l_j)$ is a quadratic spring model that makes connections between two body parts with a spatial deformation cost. $S(t)$ is the bias that models the prior of seeing a particular type as well as the prior of seeing the pair-wise type combination. The term of the bias is formulated as

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j} \tag{3}$$

Note that (2) is a linear equation that is parameterized by ω and b, therefore it can be rewritten as

$$Q_k(I, l, t) = \beta_k \cdot \Phi(I, l, t) \tag{4}$$

where β is the concatenation of ω and b. Knowing β, we are able to solve $S_k(I)$ in polynomial time [11].

The parameter β can be learned from the training data within a structured-SVM framework [15]. Note that $S_k(I)$ is bounded by $Q_k(I, l, t)$ with respect to all combinations of l and t, therefore the constraint equation of the SVM can be drawn as: a) $Q_k(I, l, t)$ needs to be larger than or equal to 1 on all positive examples. b) For all negative samples, $Q_k(I, l, t)$ should be smaller than or equal to -1 with respect to all possible l and t. Under such constraint, we would like to maximize the margin between two classes, which is a typical optimization problem that can be solved by using quadratic programming (QP) [16].

B. Posture Recognition

We learn a separate posture descriptor with respect to each of the posture categories by selecting and training the descriptors on specific subsets of the training data. The posture descriptors estimate the body part locations in the image and simultaneously generate a score associated to the best approximation of the pose articulation. We note that the quality of generated part positions is extremely low due to the severe self-occlusion. Rather than using the positions, we use the corresponding scores for posture recognition. After applying the set of posture descriptors to the input image, we get a vector of scores $\{S_1(I), ..., S_K(I)\}$ from the descriptors. The score reflects the confidence of that image belonging to a certain posture category. One straight-forward way of recognizing posture from these scores would be to apply non-maximum suppression over the scores. However, the scores cannot be guaranteed to have the same scale and are, therefore, not comparable to each other. Moreover, the output of multiple descriptors may be informative of a posture, so that it makes sense to combine them.

Our solution is to treat the descriptor scores as a vector of features in a classification problem. We compute a classification result $P_k(I) = \Psi_k(S_1(I), ..., S_K(I))$, which could, in general, be a binary label or a probabilistic measure of the predicted label. For the purposes of this work, we used a standard SVM [17] with Gaussian kernel.

IV. EXPERIMENT AND RESULTS

In this section, we evaluate both the conventional approach and the proposed approach in the context of the top-view. We firstly describe the two datasets that are used for evaluation. We conducted three experiments, each of which gives answers to the one of the research questions introduced in Section I.

A. Data

1) TUM Kitchen Dataset: The first dataset that we use is the publicly available TUM Kitchen Dataset [18]. The dataset is recorded in a home-monitoring scenario where the actor performs daily activities in a kitchen. The dataset consists of 10 typical posture classes, including standing,

walking, reaching, taking objects, etc. The postures have been annotated for each of the frames. The dataset also provides the ground-truth body part locations in 3D, so that we can freely project these points to any camera view that we want.

The TUM Kitchen Dataset also contains image sequences that are captured with four cameras. However, like most benchmark datasets [19], [20], the TUM Kitchen Dataset contains only the side-view images. We therefore collect our own dataset to be able to evaluate in the top view scenarios.

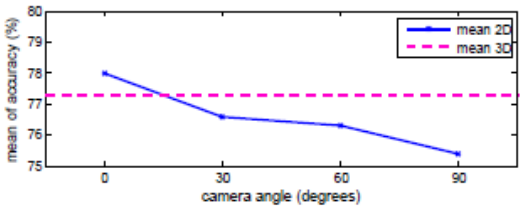
2) Our Dataset: The dataset is recorded with an omni-directional camera that is mounted on the ceiling. The persons in the frames are seen from the top-view and the body parts strongly occlude each other (see Fig. 5). To get the ground-truth body part locations, we mounted a Kinect sensor to capture the side view of the person, and we apply OpenNI skeleton tracking on the Kinect data. From the depth image, we use the skeleton tracker to generate a human skeleton that consists of 15 joint points, *i.e.* head, neck, torso, shoulders, ankles, hips, knees and feet. Since both the Kinect sensor and the omni-directional camera are calibrated within the same world coordinate system, we are able to project these joint points from the coordinate system of the Kinect sensor onto the omni-directional image plane. These projected points in 2D are manually corrected for errors, and they are used as the ground-truth body part locations for training.

The dataset contains 8 videos, and each of them has about 3000 frames. We annotated the posture labels every 10 frames (about 1 second),and the labels are as follows: standing, bending, sitting, pointing, stretching, and walking. Note that in our dataset one frame can be associated with multiple posture labels, *e.g.* a person may be standing and pointing at the same time.

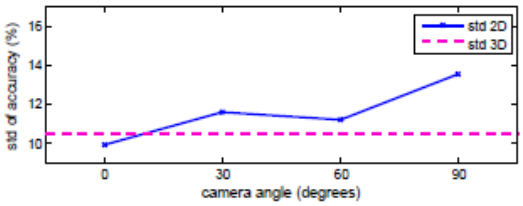
Next, we introduce the three experiments that we conducted. In the first experiment, we evaluate on the TUM Kitchen Dataset as their ground-truth pose are well annotated in 3D. In contrast, 3D poses in our dataset are less accurate as they are annotated in an automatic way using the Kinect. For the second and third experiments, we use our own dataset because the TUM Kitchen Dataset contains only persons with the side view.

B. Is 2D pose competitive with 3D pose for posture recognition?

Our first experiment is to evaluate the performance of posture recognition with respect to different camera angles. In this experiment, we use the TUM Kitchen dataset because it allows for easy comparison with the state-of-the-art 3D-based posture recognition approach, and also because the ground-truth locations in 3D are more accurate, compared with the points detected by Kinect in our own dataset. Following the work of [8], firstly we compute the geometric distance between the 3D body part locations. The geometric distances are computed within a certain temporal window, in such a way that the temporal changes of the body part



(a) mean performance



(b) standard deviation

Fig. 3. Performance of posture recognition with 3D locations and with different 2D projections based on the TUM Kitchen dataset. (a) shows the mean of the performance, and (b) shows the standard deviation. In the case of 2D, the performance drops gradually as the camera angles changes from the side-view (0°) to the top view (90°).

locations are also encoded. We apply these distance measurements as the features, and the postures are recognized by using the Random Forest [12] classifier. Furthermore, we manually define a set of mock cameras that captures different views of the humans. Following the positioning these cameras, we project the 3D body part locations onto the image plane, and then we evaluate the system in a 2D space.

We set the camera angles from 0° (side-view) to 90° (top-view) with the step-size of 30°. For this experiment, we use the posture recognition approach as described in [8]. Fig. 3 demonstrates the classification rate of postures over different camera angles. The results show that recognizing a posture becomes more difficult with the increasing camera angles. Notably, the mean drops by over 2% when the camera shifts from the side-view to the top-view. Also, we note that the side-view (2D) outperforms the 3D, which is rather surprising as projecting from 3D to 2D results in data loss. We infer that the data loss here contains mostly the noise in 3D. After projection, the 2D points in the side-view still hold the most discriminative information which can facilitate posture recognition. It is analogical to applying noise reduction using Principle Component Analysis (PCA), which reduces the dimension of the data from 3D to 2D.

This experiment shows that top-view is a more difficult task compared with the side-view. Again, the approaches are evaluated based on the ground-truth locations. In practice, however, getting the correct body part locations is already a very challenging task by itself. Next, we evaluate the state-of-the-art pose estimation approach on our top-view data to see how well the 2D pose can be estimated from the top-view.

C. How accurately can we obtain 2D pose from top-view images?

In this experiment, we evaluate on our own dataset to see how well the state-of-the-art approach can estimate body part locations from the top-view images. We randomly select 10% samples per posture category as the test set, and the rest are kept as the data for training. The positive training examples are the top-view images together with the associated body part locations. The negative training examples are taken from the INTRA dataset [21], which contains random background images with no person. To generate more positive training images, we mirrored and added slight rotation to the training examples.

We adopt the state-of-the-art approach [11] for estimating body articulations. We use the Histogram of Oriented Gradient (HOG) as our image features. We evaluate the system with the standard evaluation criteria of pose estimation, *i.e.* the probability of a correct pose (PCP). The PCP computes the percentage of correctly localized body parts. The results on the test set are shown in Fig. 4. The performance in general is rather bad, which is mainly caused by the self-occlusion. The single descriptor in the graph refers to the posture descriptor that is trained on all the data instead of a specific posture category. We compare the results of the posture descriptors with the single descriptor. We show that the posture descriptor always outperforms the single descriptor when evaluated on its specific posture class. This is because the single descriptor tries to model all the data which bare large variation over different posture classes. Note that the posture descriptor performs much better on its own posture category than on the others. It exhibits notable potential of distinguishing among posture categories using the posture descriptors, which can be very helpful for posture recognition.

D. How accurately can we recognize posture from imperfect 2D pose, and how does this performance compare to our proposed model?

This section compares the performance of posture recognition between our proposed system and the baseline approach on our top-view dataset.

In our approach, we adopt the method from [11] as our posture descriptor. The posture descriptor is learned from each of the posture categories. For classification, we train a Support Vector Machine (SVM) [17] with the RBF kernel per posture class. Using the matching scores from all posture descriptors, the SVM gives a binary decision on the posture label.

To compare with our proposed system, we form the baseline approach by combining two state-of-the-art approaches in pose estimation and posture recognition. Specifically, we follow the approach of [11] for pose estimation. We learn a single descriptor over all the data. Then we use the single descriptor to estimate body part locations. After that, we follow [8] to extract the geometric features from the 2D body part locations, and we infer the posture labels using random forest.

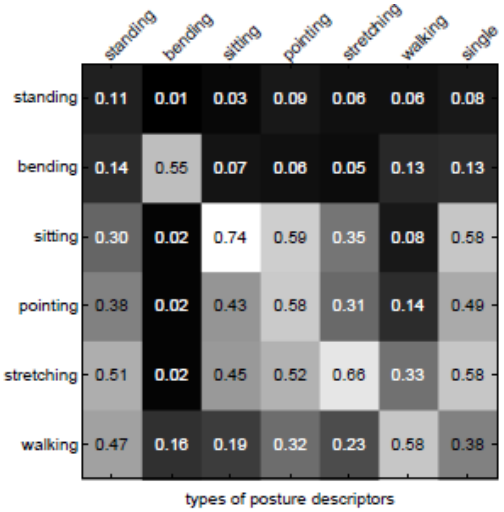


Fig. 4. The PCP performance of body part estimation over posture descriptors (columns) and posture classes of the test data (rows) on the top-view dataset. The first six posture descriptors are trained with the specific posture data. In contrast, the last "single" posture descriptor is trained on the mixing of all the training data, and therefore it is a general model that learned from all the data regardless of the posture categories. We show that when the posture descriptors are evaluated on its own posture category, the results (diagonal) always outperform the "single" model (last column).

Note that the geometric features in [8] are extracted within a short sequence of frames, therefore the temporal information are encoded in the baseline approach. In contrast, our proposed system is evaluated on still images, and we believe the performance can be further improved by adding temporal filtering to our current system. This is left as future work.

The performance of posture recognition is shown in Table I. The results show that our approach outperforms the conventional approach on all posture categories, and the average performance is better than the conventional approach by over 23%. In particular, the performance is improved by 69% on the bending data. This is because when people are bending, occlusion is more severe compared with the other postures, *e.g.* the limbs are most likely to be fully occluded by the torso when bending. Estimating the body part locations from these missing limbs becomes an extremely difficult task. Benefiting from the posture descriptors, our approach does not require the body part locations to be correctly localized and therefore our system still shows very high performance on the bending data. From our results, we believe our system is more robust to the self-occlusion as we do not rely on the body part locations which are rather unstable when estimated under the top-view. Moreover, we believe our system can be further improved after adding the temporal information. Finally, we show some sample postures recognized by our system in Fig. 5 which gives an illustration of our results.

TABLE I

RESULTS OF POSTURE RECOGNITION: F-SCORE

	standing	bending	sitting	pointing	stretching	walking	avg.
Baseline [8]+[11]	93.65	20.69	93.51	43.87	25.53	60.43	56.28
Our approach	**95.53**	**89.00**	**96.83**	**62.69**	**58.90**	**75.53**	**79.75**

Fig. 5. Results of the posture recognition based on our top-view dataset. The example images are randomly sampled from the testing results. The text on the left indicates the ground-truth posture label of the images in the row. Postures that are correctly recognized are in green rectangles, and postures are in red rectangles if wrong labels are predicted.

V. CONCLUSION

In this paper, we proposed a novel method to classify human postures from the top-view cameras. Using the posture descriptors, we get a vector of matching scores, and we use the scores for posture recognition instead of the conventional way which use the body part locations. The results show that leveraging the posture descriptors provides superior classification results in images with self-occlusion. We believe the posture descriptors can be further leveraged by enabling temporal filtering for activity recognition.

REFERENCES

[1] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, "People watching: Human actions as a cue for single-view geometry," in *ECCV*, 2012.
[2] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *CVPR*, 2010, pp. 17–24.
[3] A. Yao, J. Gall, and L. Van Gool, "A hough transform-based voting framework for action recognition," in *CVPR*, 2010, pp. 2061–2068.
[4] J. P. Wachs, D. Goshorn, and M. Kölsch, "Recognizing human postures and poses in monocular still images," in *IPCV*, 2009.
[5] E. Weng and L. Fu, "On-line human action recognition by combining joint tracking and key pose recognition," in *IROS*, 2012, pp. 4112–4117.
[6] N. Zouba, B. Boulay, F. Bremond, and M. Thonnat, "Monitoring activities of daily living of elderly based on 3d key human postures," *Cognitive Vision*, pp. 37–50, 2008.
[7] L. Sigal, "Human pose estimation," *Encyclopedia of Computer Vision*, 2011.
[8] A. Yao, J. Gall, G. Fanelli, and L. Van Gool, "Does human action recognition benefit from pose estimation?" in *BMVC*, 2011.
[9] Q. Lin, C. Zhou, S. Wang, and X. Xu, "Human behavior understanding via top-view vision," *International Conference on Control Automation and Systems*, vol. 3, pp. 184–190, 2012.
[10] S. Weerachai and M. Mizukawa, "Human behavior recognition via top-view vision for intelligent space," in *International Conference on Control Automation and Systems*, 2010, pp. 1687–1690.
[11] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR*, 2011, pp. 1385–1392.
[12] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *CVPR*, 2009, pp. 1022–1029.
[13] N. Ikizler and P. Duygulu, "Histogram of oriented rectangles: A new pose descriptor for human action recognition," *Image and Vision Computing*, vol. 27, no. 10, pp. 1515–1526, 2009.
[14] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, no. 1, pp. 55–79, 2005.
[15] T. Finley and T. Joachims, "Training structural svms when exact inference is intractable," in *Proc. of the 25th International Conference on Machine learning*. ACM, 2008, pp. 304–311.
[16] C. Hsu, C. Chang, C. Lin, *et al.*, "A practical guide to support vector classification," 2003.
[17] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[18] M. Tenorth, J. Bandouch, and M. Beetz, "The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition," in *ICCV workshop on Computer Vision*, 2009, pp. 1089–1096.
[19] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, vol. 3, 2004, pp. 32–36.
[20] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, vol. 2, 2005, pp. 1395–1402.
[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1, 2005, pp. 886–893.

Appendix B

Multi-User Identification and Efficient User Approaching by Fusing Robot and Ambient Sensors

Ninghang Hu*, Richard Bormann*, Thomas Zwölfer, and Ben Kröse

***Abstract*— We describe a novel framework that combines an overhead camera and a robot RGB-D sensor for real-time people finding. Finding people is one of the most fundamental tasks in robot home care scenarios and it consists of many components, *e.g.* people detection, people tracking, face recognition, robot navigation. Researchers have extensively worked on these components, but as isolated tasks. Surprisingly, little attention has been paid on bridging these components as an entire system. In this paper, we integrate the separated modules seamlessly, and evaluate the entire system in a robot-care scenario. The results show largely improved efficiency when the robot system is aided by the localization system of the overhead cameras.**

I. INTRODUCTION

Globally, aging of populations is becoming a potential problem. The growing group of elderly people requires efficient and accurate care-giving at an affordable level, and robots may offer a solution in future. In recent years, researchers have been extensively working on the tasks of people detection [1], people tracking [2], face recognition [3], robot navigation, and robot controls, however, mainly as isolated tasks instead of combining these systems for real life applications. In this paper, we study these tasks jointly, and we propose a unified system that integrates these components in a home care scenario, see Fig. 1. The system is very efficient and suitable for real-time applications. Moreover, the single components are complementary to help improving the robustness of the entire system.

Two fundamental tasks in robot home care scenarios are people localization and people identification. They are also the elemental components for more advanced tasks, *e.g.* activity recognition [4]. Commonly used sensors for these tasks include overhead cameras and RGB-D sensors on mobile robots. The overhead cameras are usually fixed at the ceiling, covering most of the areas in the room. The cameras only need to be calibrated once so that the coordinates of the detected person can be transformed easily from the image space to the ground-plane of the room. As the camera is mounted at the ceiling, people in the video are less likely to be occluded by each other. The overhead camera commonly has a wide field of view. Thereby one

The research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287624

*The two authors contribute equally to the paper.

Ninghang Hu and Ben Kröse are with Informatics Institute, Faculty of Science, University of Amsterdam, 1098 XH Amsterdam, The Netherlands. n.hu,b.j.a.krose@uva.nl

Richard Bormann and Thomas Zwölfer are with the Institute for Manufacturing Engineering and Automation, Fraunhofer IPA, 70569 Stuttgart, Germany. richard.bormann@ipa.fraunhofer.de, thomas.zwoelfer@ipa.fraunhofer.de

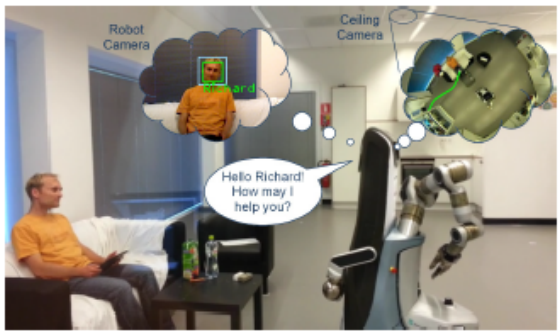


Fig. 1. Fusion of robot and environment cameras for direct user approach.

camera is often sufficient for detecting and tracking people in the whole room. Despite these benefits, it is very difficult for the overhead camera to tell people's identity. Faces can hardly be seen at many locations. The most prominent parts of people are the clothes, but they may be changed over time. Consequently, the overhead camera may be enough to find the person, but it is not sufficient for people identification.

The sensors on the robot, *e.g.* the Microsoft Kinect, provide a complementary view to the overhead camera. The on-board cameras are commonly mounted at a level that keeps the human face in sight. The RGB-D sensor provides both the color image from a color camera and the depth image from a range camera. By fusion of the depth image and color image, the face can be recognized robustly [3]. However, the RGB-D sensor is limited in both the range and the view angle. When people are too close, the face is outside the field of view; when they are far away, the accuracy and resolution of face data drops quickly. An advantage of the combination with ceiling cameras for tracking is that the robot does not need to keep monitoring the persons all the time. Hence, the robot may carry out other tasks, rather than allocating its resources to the task of tracking each person.

In this paper, we propose a system that combines the robot RGB-D sensor and the overhead cameras for real-world applications. The architecture of the proposed system is shown in Fig. 2. The system consists of three modules: a) people detection and tracking, b) people identification, and c) a joint tracker that combines both kinds of information. The first module finds multiple people that are present in the room using two overhead cameras. The second module identifies people using a Kinect sensor that is mounted on the robot. The third module collects information from the first two modules and associates tracks with human identities.

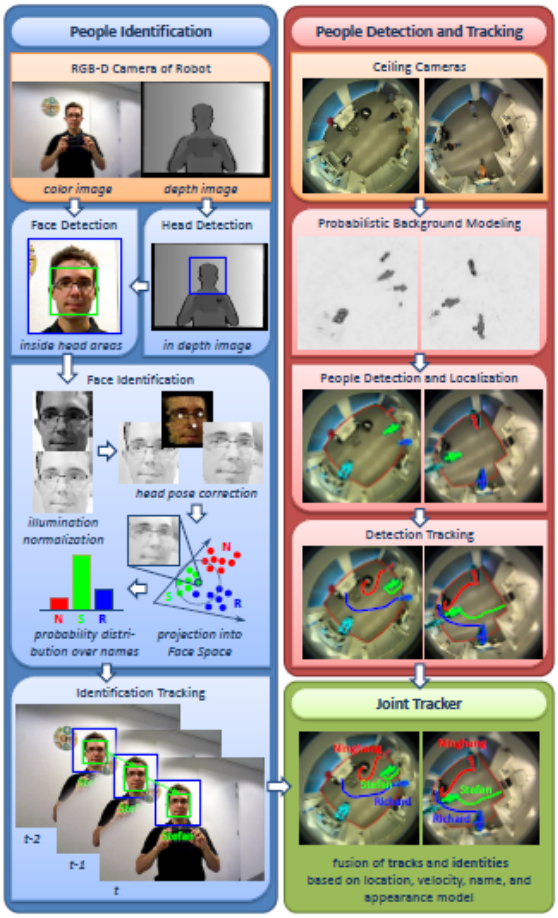


Fig. 2. An overview of the combined tracking and recognition system.

The contributions of the paper are:

1) We propose a system that integrates the overhead cameras with the RGB-D sensor for human localization and identification tasks. All components of our system follow a probabilistic approach so that the system is more robust to noise. The experiments show that the system is very efficient and can be applied in real-time.
2) Instead of evaluating the components as isolated tasks, we evaluate the effectiveness of the whole system together with robot path planning and navigation in a real life scenario, *i.e.* finding one or more persons.
3) We show a novel way of associating the faces with human tracks. We represent the face locations as weighted particles and the faces are associated with the tracks by evaluating the weighted particles in a set of Kalman Filters.

The remainder of the paper is structured as follows. We review the related work in Section II. Afterwards, we introduce three main components of the system, *i.e.* people localization and tracking in Section III, people identification in Section IV, and the joint tracker in Section V. We show the results of our experiments in Section VII.

II. Related Work

In the past few years, there has been considerable work on people detection and people tracking using computer vision techniques. Most of the work only adopts a single type of sensor, *e.g.* color cameras [5], [6], [7], depth cameras [8], [9], or laser range finders [10]. More recently, researchers worked on combining different sensors to make the tracking system more reliable. Cui et al. [11], Kristou et al. [12] and Kobilarov et al. [13] utilize a mobile robot platform to detect people with a laser range finder and video cameras. Luber et al. [14] fuse the data from a depth camera and a color camera. Nakazawa et al. [15] combine multiple cameras for multi-people tracking. For people detection, we employ a similar approach as in [16] and [17]. Instead of fusing the on-board laser range finder and the ambient camera, we combine the Kinect sensor with ambient cameras. We use the Kinect sensor for identifying people and ambient cameras for tracking. The results from the two components are fused in a probablistic way by a joint tracker.

Face recognition is usually split up into the detection of face regions in an image and the actual identification of the detected face image patches. The former task has been approached on color images using the Viola-Jones classifier [18]. Later, many extensions have been introduced [5], [19]. Face detection is tackled on point clouds using local curvature features [9]. An RGB-D fusion system for combined head detection in depth images and face detection in color images [20] is used for face detection in this paper.

There exists a large variety of methods for face identification that might be divided into projection methods [6], [21], [22], [23], local pattern-based methods [24], [25], generative models [26], [27], and sparse representations [28]. The latter represent the space of known faces with a set of carefully chosen gallery images whereas generative methods construct an illumination and pose model for each individual from training data recorded under specialized lighting conditions. Both kinds of methods suffer from long training and/or recognition times. Given the robustness and real-time demands of robotics applications, projection methods like Eigenfaces [21] or Fisherfaces [6], or local pattern-based methods are preferable. The first construct a handy representation of identities by projecting the high dimensional face image matrix into a low-dimensional subspace, which commonly reduces intra-class variance and amplifies inter-class differences. Local pattern-based methods compute dense local binary patterns [24] or local ternary patterns [25] which become accumulated in histograms over spatially constrained areas. The robustness of these methods can be improved by applying illumination normalization techniques like histogram equalization, logarithmic transform [29], gamma correction [30], discarding the low-frequency Discrete Cosine Transform coefficients [31], Difference of Gaussians filtering, or contrast equalization [25] to the face image in advance. Compensation measures for varying head

pose such as multiple orientation modeling on the training data [26], [32] or face plane estimation [9], [33], [34] also increase the robustness of the identification system. The face recognition module used for this work bases on our earlier work on robust real-time face recognition systems [3].

Most of the previous work focuses on solving only one type of the tasks and little work has been done on combining modules and evaluating the system as a whole. In this paper, we work on bridging the gaps between the different components and build an integrated tracking and recognition system that is suitable for real home care scenarios.

III. People Localization and Tracking

In this section, we introduce the sub-system for people localization and tracking. The pipeline of the sub-system is shown in Fig. 2 (red panel).

A. People Localization

The ground-plane area is discretized into small regions for localizing people. Our goal is to find in which regions the people are located. We define k as a random variable to indicate the index of a region, and $\boldsymbol{X} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, ..., \boldsymbol{X_C}\}$ as a set of images that are observed from the overhead cameras. We formulate the people detection problem in a Bayesian fusion framework [35].

$$P(k|\boldsymbol{X}) \propto P(k) \prod_{c=1}^{C} P(\boldsymbol{X_c}|k) \quad (1)$$

where $P(k)$ is the prior distribution of seeing a person located at region k, and $P(\boldsymbol{X_c}|k)$ is the likelihood of seeing the image $\boldsymbol{X_c}$ given there is a person at location k. Here we assume images captured from different cameras are conditionally independent given the location of the person. In this paper, we make a simple assumption that $P(k)$ follows uniform distribution, although it can also be learned from the training data.

The likelihood term $P(\boldsymbol{X_c}|k)$ in (1) computes the joint probability of all pixels from a single camera given that a person is located at k.

$$P(\boldsymbol{X_c}|k) = \sum_{x_c \in \Phi(k)} P_f(x_c) + \sum_{x_c \notin \Phi(k)} P_b(x_c) \quad (2)$$

where x_c is a pixel value in $\boldsymbol{X}_c$. $P_b(x_c)$ computes the background probability of the pixel x_c, and $P_f(x_c)$ computes the foreground probability. As we do not know any prior knowledge about how a person looks like, the foreground probability is a constant for all pixels. The background probability is estimated by the Adaptive Gaussian Mixture Model [36], see Fig. 2. $\Phi(k)$ is a binary mask where the ones indicate the foreground pixels and zeros the background pixels. The mask is generated by evaluating a 3D human-shape template at location k and then projecting the 3D template onto the image plane. Pixels within the area of the projection are considered as the foreground, and otherwise the pixels are labeled as the background. We sum up the foreground and background probabilities over all pixels in the image, and then we are able to evaluate the product of the likelihood terms with respect to all cameras as in (1).

We evaluate the posterior probability using (1) for all the regions on the ground-plane. People are detected at regions with local maximal posterior probabilities. The algorithm is implemented using logarithm likelihood probabilities to avoid numerical problems.

Next, we introduce the algorithm that generates tracks out of the frame-based detections.

B. Tracking

We use an online tracker that updates tracks every frame. The tracker associates the detected people with the existing tracks by evaluating the matching scores between the detection and the model of the tracks. After that, the new detection is appended to the corresponding track and tracks get updated separately using a Kalman Filter [37].

We compute the matching score using two cues, *i.e.* the appearance of a human and the location. Assume there are M detections and N tracks generated at time t. The matching score between the m^{th} detection and n^{th} track is computed as

$$S_t(m,n) = \Psi_1(h(o_m^t), h_n^{t-1}) \Psi_2(o_m^t, s_n^{t-1}, v_n^{t-1}) \quad (3)$$

where o_m^t is the location of the detected person, s_n^{t-1} is the previous tracked location, and v_n^{t-1} is the velocity of the track. Both s_n^{t-1} and v_n^{t-1} can be estimated by the Kalman Filter. Ψ_1 measures the similarity between the appearance template h_n^{t-1} and the histogram extracted at the detected location $h(o_m^t)$. Here we use the Bhattacharyya distance [38] for comparing two color histograms

$$\Psi_1(h(o_m^t), h_n^{t-1}) = \sqrt{1 - \sum_u \sqrt{h_u(o_m^t) h_{n,u}^{t-1}}} \quad (4)$$

Ψ_2 measures the consistency between the observed location o_m^t and the new location predicted by the Kalman Filter.

$$\Psi_2(o_m^t, s_n^{t-1}, v_n^{t-1}) = \mathcal{N}(v_n^{t-1} + s_n^{t-1} - o_m^t, \sigma^2) \quad (5)$$

We can evaluate the matching score using (3) for all track-detection pairs. Finding the optimal assignment from the table of matching scores can be efficiently solved using the Hungarian Algorithm [39].

IV. People Identification

The system for recognizing people with the robot's RGB-D camera consists of a head and face detection module, a face recognition module, and a face identification tracker, see Fig. 2 (blue panel).

A. Face Detection

The detection module searches for face images using the Kinect sensor. The detection has two stages: first, a Viola-Jones detector [18] is applied in the depth image for finding the objects that look like a human head. These candidate regions are then verified in the color image using another Viola-Jones classifier that is trained on color images. The whole detection procedure is detailed and evaluated in our previous work [3], [20].

B. Face Identification

Subsequently, those image patches which contain a face are processed by the recognition module. The recognition starts with pre-processing the face image with a gamma transform ($\gamma = 0.2$) and a downscaling of the first 5 coefficients of the Discrete Cosine Transform on the gray scale image by a factor of 50. This realizes an illumination normalization which renders the recognition algorithm more robust against lighting conditions that are different from the training data. Then, the algorithm tries to detect facial features like the eyes or the nose with a Viola-Jones classifier and generates a virtual frontal perspective on the recorded face if those face features can be identified successfully. This measure diminishes negative effects on recognition stemming from a badly aligned face image. Eventually, the recognition is conducted by projecting the pre-processed face image into a lower-dimensional space which minimizes the variance of training face images of the same person and maximizes the inter-class variance according to Fisher's Linear Discriminant [6]. The identity estimate is found as the nearest neighbor from training data in this space or as a probability distribution constructed from the labels of the neighborhood. Again, a thorough explanation and evaluation of those methods is given in [3].

C. Identity Tracking

The detector and recognizer work based on single image frames. However, due to noise or misalignment, only using the frame-based recognition may be problematic. Therefore, the recognized identities are filtered by a tracking module for stable identity estimation.

The tracker firstly matches the recognition results in consecutive frames. The matching score is computed between the previous detection i and current detection j:

$$C(i,j) = \|X_i - X_j\|_{L_2} + \alpha \|P_i - P_j\|_{\chi^2} + \beta \|H_i - H_j\|_{\chi^2} \quad (6)$$

where X denotes the 3d face coordinates in space, P is the probability distribution over all labels, and H is the histogram of local binary patterns of the head region. α and β are optional weighting factors for the single metrics. The first term measures the distance to the last detection, the second term computes the similarity in label predictions, and the third one establishes visual similarity of the tracked image regions. The global minimum cost assignment between previous and current recognition is found with the Hungarian method [39]. New detections in the current set are added afterwards and initialized with their estimated label probability distribution. To smooth sporadic false recognitions, the estimated probability distributions are filtered temporally with a Hidden Markov Model (HMM). The final identity assignment is estimated for each frame by considering the label probabilities for each detection as inverse costs and computing the globally optimal assignment with the Hungarian Method.

V. Joint Tracker

So far we have introduced two systems. One system detects, localizes, and tracks all persons in the room. The system, however, only gets the track ID and does not know who the person is. The other system identifies the person using the robot sensor, but the robot has to keep the person in sight all the time. The joint tracker solves the problem of the two separate systems by combining both of the sensors. The joint tracker assigns the tracks as unknown persons when they have not been recognized by the robot. Once people are identified by the robot, names of people are immediately associated with the tracks.

To increase the robustness of the system, we fuse data from the two sensors in a probabilistic way. In robotic scenarios, both the locations of the tracked persons and the locations of the robot can be very noisy. Our robot is localized using the SLAM approach [40], and the robot location is represented as a set of weighted particles. As the Kinect sensor is registered in the robot's transformation tree, we can always transform the face locations that are detected by the Kinect sensor into the particle representation in world coordinates. Let the set of the particles be $L = \{(w_1, \boldsymbol{l}_1), (w_2, \boldsymbol{l}_2) ... (w_N, \boldsymbol{l}_N)\}$, where $\boldsymbol{l}$ is the location of a particle in world coordinates and w is the weight of the particle.

The posterior distribution of the human location returned by the Kalman Filter is a Gaussian distribution, with its mean indicating the most likely position of people and variance indicating the uncertainty. The set of Kalman Filters from different tracks proposes multiple Gaussian density distributions at each time step. Our goal is to associate the detected faces with those Gaussian PDFs. We compute the score of associating the i^{th} track with the j^{th} face as

$$Q(i,j) = \sum_{n=1}^{N} \frac{w_n}{C\sqrt{|\boldsymbol{\Sigma}_i|}} \exp\left((\boldsymbol{l}_n^j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{l}_n^j - \boldsymbol{\mu}_i)\right) \quad (7)$$

where $\boldsymbol{\mu}_i$ is the mean location and $\boldsymbol{\Sigma}_i$ is the covariance matrix of the i^{th} track. C is a constant that does not affect the assignments. The best assignments between the tracks and the detected faces are calculated in the same way as in people tracking. Once the track has been associated with the face, the name of the person is attached to the track until a new face is detected for that track. In such a way, the correct name can be recovered if a person is wrongly recognized at the beginning.

VI. Strategies for User Recognition and Approaching

There are two fundamental tasks in Human-Robot Interaction (HRI). The first task is that the robot needs to identify unknown users that are present in the room. The other task is that the robot is asked to approach a specific person, *e.g.* for completing a delivery. This section presents different algorithms for tackling these two tasks. For each of the tasks, we compare two algorithms a) using the robot only, b) the robot assisted by an overhead camera.

1) Uninformed User Identification: Algorithm 1 describes the case when the robot has to identify all present people without help from an external system. The robot starts searching for users by successively moving to random locations and turning around by 360° each time. In a real situation, users can be supposed to move around so that a systematic search should not bear any advantages over a random strategy. Every found user is stored internally and announced via speech. The terminating condition may be application-driven, *e.g.* finding a certain set of people, searching for people at a given place, searching for 5 minutes, etc. The *moveTo* and *rotate360* functions interrupt if all people are found.

Algorithm 1 Uninformed recognition of present users.

```
function IDENTIFYUSERSUNINFORMED
  recognitions ← ∅
  while TerminatingCondition = False do
    goal ← computeAccessibleRandomPosition()
    recognition ← moveToAndRotate360(goal)
    recognitions ← {recognitions, recognition}
  return recognitions
```

2) Informed User Identification: The combined system allows approaching each human detected by the external tracking system directly and recognizing the face (see Algorithm 2). In contrast to Algorithm 1, this algorithm is aware of having labeled all present users.

Algorithm 2 Informed recognition of present users.

```
function IDENTIFYUSERSINFORMED
  detections ← getTrackedHumans()
  recognitions ← ∅
  for all detections do
    goal ← computePositionOnPerimeter(detection)
    recognition ← moveTo(goal)
    recognitions ← {recognitions, recognition}
  return recognitions
```

3) Uninformed User Approach: Algorithm 3 displays the method to approach a specific user using the sensors of the robot only. As with Algorithm 1, a random search strategy is employed to navigate the robot through the environment searching for the desired person.

Algorithm 3 Uninformed search for a specific user.

```
procedure APPROACHUSERUNINFORMED(targetName)
  targetLocation ← ∅
  while robotLocation ≠ targetLocation do
    goal ← computeAccessibleRandomPosition()
    recognitions ← moveToAndRotate360(goal)
    targetLoc. ← checkForUser(recognitions, targetName)
    if targetLocation ≠ ∅ then
      moveTo(targetLocation)
```

4) Informed User Approach: Algorithm 4 utilizes the additional sensory information from the external cameras to approach the specific person directly given that he or she has been recognized previously and tracked in the meantime.

Algorithm 4 Informed search for a specific user.

```
procedure APPROACHUSERINFORMED(targetName)
  detections ← getTrackedHumans()
  targetLocation ← checkForUser(detections, targetName)
  if targetLocation = ∅ then
    APPROACHUSERUNINFORMED(targetName)
  else
    moveTo(targetLocation)
```

VII. Experiment and Results

In our experiments, the robot is asked to complete two different tasks, both of which occur frequently in home care scenarios with robot assistance. The first task is to let the robot identify all the people that are present in the room. The second one is to let the robot find a specific person. We evaluate the efficiency of the integrated system by measuring the average time that the robot needs to complete the tasks.

A. Experiment Setup

The experiments are setup in a domestic environment, see Fig. 3 and 4. There are two GV-FE420 cameras mounted on the ceiling, one above the sofa area and the other one in the kitchen. Both cameras are calibrated with the highest resolution (2048 × 1944 pixels). For people detection and tracking, we use only 1/4 of the full resolution for efficient processing and we find that is sufficient to give stable tracks. The cameras have a very wide field of view (over 180 degrees), and provide a good overview of the entire room. We adopt a PC with an Intel Core i7-3770K (3.50GHz) processor for people detection, tracking and also fusing the incoming face recognition data provided by the robot. The processing rate of this system is around 9 Hz.

The mobile service robot is a Care-O-bot® 3 which features an omni-directional mobile base with a flexible torso, a 7 DOF manipulator and a movable sensor head that contains a pair of stereo cameras and a Kinect RGB-D camera. The control script, navigation and person recognition software are run on two build-in PCs. The people detection and recognition module processes the RGB-D data from the Kinect camera at a resolution of 640×480 pixels. The module uses an Intel Core i7-E610 (2.53GHz) PC and delivers recognition results at 6 Hz.

For each experiment, the robot was commanded by a control script whose functionality corresponds to the search strategies proposed in Section VI. Both experiments were conducted with 5 different subjects. To demonstrate the advantages of the combined recognition and tracking system, we carried out the experiments in two flavors, *i.e.* once with the robot sensor only and once combining the robot sensor with the ceiling-mounted cameras. The environment setup of the two experiments was always the same for comparison.

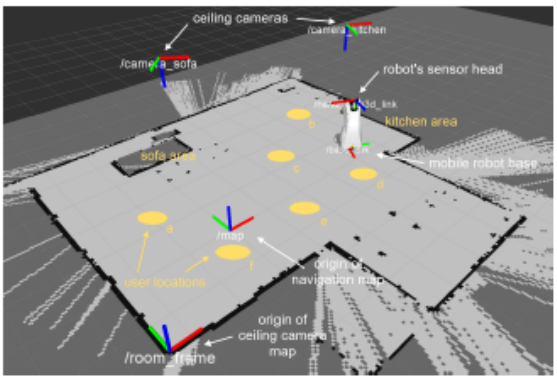


Fig. 3. Floor plan of the experimentation site with user locations a-f highlighted and the coordinate systems of the ceiling and robot cameras, the navigation and camera maps and the robot base drawn into.

Besides the sofa area, we further defined six locations in the room, see the labels a-f in Fig. 3. These points indicate the test locations where people were standing during our experiments.

B. Identification of All Present Users

The first experiment measures the performance of system on identifying and tracking all present users in the room. This is a prerequisite for tasks like activity recognition of multiple users or fast reacting to an individual call from one of the human users. The setup is that two or three subjects are distributed in the environment among positions a-f. We ask the users to look at the robot when it approaches so that the robot can recognize their faces. In a real scenario, the robot can either attract a user's attention by speech or move around the person until the face becomes visible. These options have been excluded from the experiments to avoid subjective factors in the results.

As a baseline, the robot is asked to find all users with its own sensors only. Algorithm 1 is applied to navigate the robot randomly until all people have been identified. This scenario is evaluated under 10 combinations of user locations, see Table I. We report both the time for finding all users and the number of wrongly identified persons. The baseline results vary largely in the time for finding all persons. It can be as fast as within one minute but it may also take more than 5 minutes. On average, all subjects are identified within 2 minutes and 10 seconds. All of the 26 persons within the 10 sessions have been labeled correctly, and only 3 people were wrongly identified initially but corrected after a couple of seconds. This yields an initial recognition rate of almost 90%. The two main drawbacks of the baseline approach are: a) the algorithm needs to be terminated manually as the robot never knows whether all present subjects have been found and b) the robot has to keep all users constantly in sight in order to keep track of them, which is not desirable for real-world applications.

The ambient camera system can detect and track humans. The information is shared with the robot via wireless connection so that the robot stays informed about the locations of all present persons along with their identities attached to the tracks, even when people are not visible to the robot. We use Algorithm 2 in our second experiment. After all persons have been visited and identified, this algorithm terminates automatically and announces that all users have been found. The times that are needed to identify all present users are listed in column "combined system" of Table I. The results show that the combined system is significantly better than the random search. The combined system requires only 1 minute and 5 seconds on average to complete the task. In contrast, the baseline approach takes twice the time than the combined system. The standard deviation of the combined system is around 19 seconds, which is much more stable than the time of 1 minute and 26 seconds in random search. In our experiments, the performance of the face recognition system is very stable as the guided approach can navigate the robot into an advantageous distance to the subjects so that the faces are captured with high quality. The ceiling-camera system fails at detecting people at position f for 4 times because people at that location are heavily occluded by the robot from both of the ambient cameras. Apart from these errors, the system performs outstanding, yielding an accuracy of correctly tracked and identified people of 85%.

TABLE I
RESULTS FOR FINDING ALL PRESENT USERS.

user	robot only		combined system		
locations	time	rec. errors	time	rec. errors	track. errors
b, c, d	5:48	0	1:10	0	0
b, c, e	2:18	0	1:05	0	0
b, c, f	1:14	1	1:27	0	0
c, d, e	1:28	0	0:40	0	0
c, d, f	2:56	0	1:20	0	1
d, e, f	0:58	0	1:35	0	1
b, e	1:30	2	0:53	0	0
b, f	2:00	0	1:04	0	1
c, f	1:00	0	0:40	0	1
b, d	2:30	0	0:51	0	0
average (3p)	2:27	1/18	1:13	0/18	2/18
average (2p)	1:45	2/8	0:52	0/8	2/8
average (all)	2:10	3/26	1:05	0/26	4/26
stddev (all)	1:26		0:19		

C. Approaching a Specific User

The second experiment evaluates the performance of our system on approaching a specific user, which is widely used in delivery tasks. In our scenario, the user is sitting on the couch, calling the robot, and ordering something. The robot then goes to the kitchen and in the meantime the user stands up and moves to another place. The task of the robot is to find the user for delivery. The experiment is conducted at three levels of difficulty: 1) with a cooperative user facing the robot all the time, 2) with a busy user facing a fixed direction, and 3) with two users facing the robot constantly.

The results of the first session are shown in the upper part of Table II. Column "robot only" corresponds to the baseline case when only the robot sensors are used, see Algorithm 3. We report the times that are spent by the robot on approaching the user among positions a-f as well as the

TABLE II

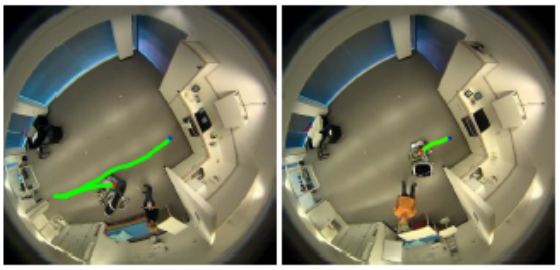
RESULTS FOR SEARCHING A SPECIFIC USER.

user	robot only		combined system		
location(s)	time	rec. errors	time	rec. errors	track. errors
with one user facing the robot constantly					
sofa	0:28	0	0:28	0	0
a	1:28	0	0:34	0	0
b	0:40	0	0:18	0	0
c	0:33	0	0:21	0	0
d	0:28	0	0:18	0	0
e	0:35	0	0:29	0	0
f	1:18	0	0:31	0	0
average	0:47	0/7	0:26	0/7	0/7
with one user facing a fixed direction					
a (+x)	1:12	0	0:33	0	0
a (-y)	7:37	0	0:43	0	0
b (-x)	0:26	0	0:21	0	0
b (-y)	9:59[1]	0	0:17	0	0
c (-x)	0:29	0	0:20	0	1
c (-y)	2:27	0	0:20	0	0
d (-x)	2:39	0	0:16	0	0
d (+y)	2:56	0	0:20	0	0
e (+x)	6:42	0	0:28	0	0
e (+y)	0:25	0	0:28	0	0
f (+x)	1:39	0	0:31	0	0
f (+y)	0:55	0	0:31	0	0
average	2:30	0/12	0:26	0/12	1/12
with two users facing the robot constantly					
b, a	0:38	0	0:15	0	0
b, c	2:17	0	0:20	0	0
b, d	1:12	0	0:21	0	0
c, d	0:22	0	0:17	0	0
d, a	0:43	0	0:16	0	0
d, f	0:37	0	0:14	0	0
e, a	1:03	0	0:26	0	0
e, b	1:21	1	0:22	0	0
f, b	1:43	0	0:52	0	0
f, e	1:10	0	0:27	1	0
average	1:07	1/20	0:23	1/20	0/20
average (all)	1:34	1/39	0:25	1/39	1/39
stddev (all)	1:44		0:09		

[1] aborted after 10 minutes of search, excluded from cumulative statistics

sofa area after leaving the kitchen area. All delivery times are quite fast with an average of 47 s. The delivery tasks are accomplished successfully as all of the users are correctly recognized. These results are compared with the performance of the "combined system" that keeps tracking the user after being identified. The times of delivery using Algorithm 4 are a little more than half of Algorithm 3, with only 26 s on average and deliveries are all successful.

The second session of this experiment requires the users to face a fixed direction. This makes the approaching a harder problem as the robot may not obtain a good perspective of the face. In Table II, signs + and - refer to the user's orientation, *e.g.* "a (+x)" means that the user stands at location a, facing the positive direction of the x axis. The results show that the combined system significantly outperforms the system that only uses the robot sensors. The average search time for the robot-only system is 2 minutes 30 seconds which is 5 times longer than the 26 seconds when using the combined system. The worst case of the robot-only system occurs at "b (-y)", which has to be manually terminated after 10 minutes of unsuccessful search. For the combined system, the results of the second session are similar to those of the first session.



(a) (b)

Fig. 4. Robot trajectories of finding a specific person when (a) using random search and (b) assisted with the external tracking system.

This is because the user's gaze direction is irrelevant when the position is known from the overhead cameras. For both systems, no recognition error is observed, but the tracking system loses the user once after delivery in case "c (-x)".

The third session is a more realistic situation where multiple people are present and the robot is required to approach the correct person. The lower part of Table II presents the results of 10 experiments with two users. The user who has ordered in the sofa area goes to the position that is underlined in Table II, the other user is located at the second position. Both users are told to face the robot constantly. Accordingly, the average time of approaching is 1:07 for the robot-only system, which is similar to the experiments with only one user. A recognition error occurs once when the order is delivered to a wrong person. Using the combined sensor system, the robot can be guided to the correct person in approximately the same time as in the tests before. Apart from one recognition error during delivery, the tasks are successfully completed.

In conclusion, the delivery tasks can be accomplished with the accuracy of 96.5% using both strategies of approaching. With the combined system, however, the delivery is three times faster than the unguided search, and the results have a smaller deviation. Hence, using the combined system is more efficient and more stable than just using the robot sensors for our tasks.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we use ambient cameras for detecting and tracking people, and we use a robot-mounted RGB-D sensor for identifying people. The information from the both sensors is combined in a joint tracking system for efficient and accurate people finding. The results show that by leveraging these two complementary types of sensors, the robot can be guided to approach people directly instead of searching through the room. This significantly reduces the approaching time and provides a more intuitive and comfortable way for the users.

Based on the people tracking and identification system proposed in this paper, we are currently working on activity recognition and prediction for multi-users. The working system allows the robot for deliberately offering assistance

on critical tasks and may serve for long-term medical check-ups by detecting deviations from the user's daily activity schemes. We furthermore plan to enhance the approaching algorithms with proxemics, *i.e.* having the robot to approach the user in a human-acceptable way.

REFERENCES

[1] G. Gate, A. Breheret, and F. Nashashibi, "Centralized fusion for fast people detection in dense environment," in *Proc. IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 76–81.

[2] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2002, pp. 695–701.

[3] R. Bormann, T. Zwölfer, J. Fischer, J. Hampp, and M. Hägele, "Person recognition for service robotics applications," in *accepted for publication at the 13th International IEEE-RAS International Conference on Humanoid Robots*, 2013.

[4] B. K. Ninghang Hu, Gwenn Englebienne, "Posture recognition with a top-view camera," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[5] J. Li, T. Wang, and Y. Zhang, "Face Detection using SURF Cascade," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 2183–2190.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[7] N. Hu, H. Bouma, and M. Worring, "Tracking individuals in surveillance video of a high-density crowd," in *Proc. of SPIE Vol*, vol. 8399, 2012, pp. 839 909–1.

[8] Y. Zhu and K. Fujimura, "A bayesian framework for human body pose tracking from depth image sequences," *Sensors*, vol. 10, no. 5, pp. 5280–5293, 2010.

[9] R. Hg, P. Jasek, C. Rofidal, K. Nasrollahi, T. Moeslund, and G. Tranchet, "An RGB-D Database Using Microsoft's Kinect for Windows for Face Detection," in *Proc. Intern. Conference on Signal Image Technology and Internet Based Systems*, 2012, pp. 42 –46.

[10] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proc. IEEE Internat. Conference on Robotics and Automation*. IEEE, 2008, pp. 1710–1715.

[11] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Multi-modal tracking of people using laser scanners and video camera," *Image and vision Computing*, vol. 26, no. 2, pp. 240–252, 2008.

[12] M. Kristou, A. Ohya, and S. Yuta, "Target person identification and following based on omnidirectional camera and LRF data fusion," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2011.

[13] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser," in *Proc. IEEE International Conference on Robotics and Automation*. IEEE, 2006, pp. 557–562.

[14] M. Luber, L. Spinello, and K. O. Arras, "People tracking in RGB-D Data with on-line boosted target models," in *Proc. International Conference on Intelligent RObots and Systems*, 2011, pp. 3844–3849.

[15] N. Atsushi, K. Hirokazu, H. Shinsaku, and I. Seiji, "Tracking multiple people using distributed vision systems," in *Proc. IEEE Internat. Conf on Robotics and Automation*, vol. 3. IEEE, 2002, pp. 2974–2981.

[16] N. Hu, G. Englebienne, and B. J. Kröse, "Bayesian fusion of ceiling mounted camera and laser range finder on a mobile robot for people detection and localization," in *Human Behavior Understanding*. Springer, 2012, pp. 41–51.

[17] A. A. Mekonnen, F. Lerasle, A. Herbulot, *et al.*, "External cameras and a mobile robot for enhanced multi-person tracking," in *Proc. Internat. Conference on Computer Vision Theory and Applications*, 2013.

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 511–518.

[19] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 577–584.

[20] J. Fischer, D. Seitz, and A. Verl, "Face detection using 3-d time-of-flight and colour cameras," in *Proc. ISR/ROBOTIK*, 2010, pp. 112–116.

[21] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[22] I. Naseem, R. Togneri, and M. Bennamoun, "Linear Regression for Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[23] C.-Y. Zhang and Q.-Q. Ruan, "Face Recognition Using L-Fisherfaces," *Journal of Information Science and Engineering*, vol. 26, no. 4, pp. 1525–1537, 2010.

[24] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[25] X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[26] A. Georghiades, P. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[27] K. Lee, J. Ho, and D. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[28] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Toward a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 372–386, 2012.

[29] H. Liu, W. Gao, J. Miao, and J. Li, "A Novel Method to Compensate Variety of Illumination in Face Detection," in *Joint Conference on Information Sciences*, 2002, pp. 692–695.

[30] T. Goel, V. Nehra, and V. P. Vishwakarma, "Comparative Analysis of various Illumination Normalization Techniques for Face Recognition," *International Journal of Computer Applications*, vol. 28, no. 9, 2011.

[31] W. L. Chen, M. J. Er, and S. Q. Wu, "Illumination Compensation and Normalization for Robust Face Recognition Using Discrete Cosine Transform in Logarithm Domain," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 36, no. 2, pp. 458–466, 2006.

[32] A. Pentland, B. Moghaddam, and T. Starner, "View-based and Modular Eigenspaces for Face Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 84–91.

[33] S. Gurbuz, E. Oztop, and N. Inoue, "Model free head pose estimation using stereovision," *Pattern Recognition*, vol. 45, no. 1, pp. 33–42, 2012.

[34] D. Beymer, "Face Recognition under Varying Pose," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 756–761.

[35] G. Englebienne and B. J. Kröse, "Fast bayesian people detection," in *Proceedings of the 22nd Benelux AI Conference, BNAIC*, 2010.

[36] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. International Conference on Pattern Recognition*, vol. 2. IEEE, 2004, pp. 28–31.

[37] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[38] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.

[39] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[40] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, pp. 34–46, 2007.

Appendix C

Learning Latent Structure for Activity Recognition*

Ninghang Hu[1], Gwenn Englebienne[1], Zhongyu Lou[1] and Ben Kröse[1,2]

***Abstract*— We present a novel latent discriminative model for human activity recognition. Unlike the approaches that require conditional independence assumptions, our model is very flexible in encoding the full connectivity among observations, latent states, and activity states. The model is able to capture richer class of contextual information in both state-state and observation-state pairs. Although loops are present in the model, we can consider the graphical model as a linear-chain structure, where the exact inference is tractable. Thereby the model is very efficient in both inference and learning. The parameters of the graphical model are learned with the Structured-Support Vector Machine (Structured-SVM). A data-driven approach is used to initialize the latent variables, thereby no hand labeling for the latent states is required. Experimental results on the CAD-120 benchmark dataset show that our model outperforms the state-of-the-art approach by over 5% in both precision and recall, while our model is more efficient in computation.**

I. INTRODUCTION

Robotic companions to help people in their daily life are currently a widely studied topic. In Human-Robot Interaction (HRI) it is very important that the human activities are recognized accurately and efficiently. In this paper, we present a novel graphical model for human activity recognition.

The task of activity recognition is to find the most likely underlying activity sequence based on the observations generated from the sensors. Typical sensors include ambient cameras, contact switches, thermometers, pressure sensors, and the sensors on the robot, *e.g.* RGB-D sensor and Laser Range Finder.

Probabilistic Graphical Models have been widely used for recognizing human activities in both robotics and smart home scenarios. The graphical models can be divided into two categories: generative models [1], [2] and discriminative models [3], [4], [5]. The generative models require making assumptions on both the correlation of data and on how the data is distributed given the activity state. The risk is that the assumptions may not reflect the true attributes of the data. The discriminative models, in contrast, only focus on modeling the posterior probability regardless of how the data are distributed. The robotic and smart environment scenarios are usually equipped with a combination of multiple sensors. Some of these sensors may be highly correlated, both in the temporal and spatial domain, *e.g.* a pressure sensor on the mattress and a motion sensor above the bed. In these scenarios, the discriminative models provide us a natural way of data fusion for human activity recognition.

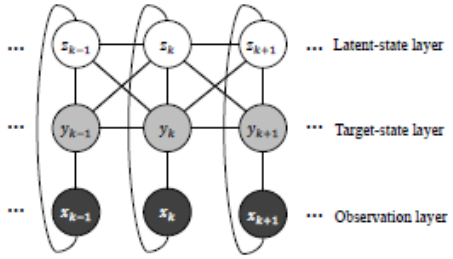


Fig. 1. The proposed graphical model. Nodes that represent the observations x are rendered in black, and they are observed both in training and testing. Grey nodes y are only observed during training but not testing, and they represent the target labels to be predicted, *e.g.* activity labels. White nodes z refer to the latent variables, which are unknown either in training or testing. Note that x_k, y_k, z_k are fully connected in our model, and also for nodes of transition.

The linear-chain Conditional Random Field (CRF) is one of the most popular discriminative models and has been used for many applications. Linear-chain CRFs are efficient models because the exact inference is tractable. However, they are limited in the way that they cannot capture the intermediate structures within the target states [6]. By adding an extra layer of latent variables, the model allows for more flexibility and therefore it can be used for modeling more complex data. The names of these models are interchangeable in the literature, such as Hidden-Unit CRF [7], Hidden-state CRF [6] or Hidden CRF [8].

In this paper, we present a latent CRF model for human activity recognition. For simplicity, we use "latent variables" to refer to the augmented hidden layer, as they are unknown either in training or testing. The "target variables", which is observed during training but not testing, represent the target states that we would like to predict, *e.g.* the activity labels. See Fig. 1 for the graphical model and the difference between latent variables and target variables. We evaluate the model using the RGB-D data from the benchmark dataset [3]. The results show that our model performs better than the state-of-the-art approach [3], while the model is more efficient in inference.

The contributions of this paper can be summarized as follows:

1) We propose a novel Hidden CRF model for predicting underlying labels based on the sequential data. For each temporal segment, we exploit the full connectivity among observations, latent variables, and the target

*The research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287624.

[1] N. Hu, G. Englebienne, Z. Lou and B. Kröse are with Intelligent System Lab Amsterdam, University of Amsterdam, 1098XH Amsterdam, The Netherlands {n.hu,g.englebienne,z.lou,b.j.a.krose}@uva.nl

[2] B. Kröse is also with the Amsterdam University of Applied Science

variables, from which we can avoid making inappropriate conditional independence assumptions.
2) We show an efficient way of applying exact inference in our graph. By collapsing the latent states and the target states, our graphical model can be considered as a linear-chain structure. Applying exact inference under such a structure is very efficient.
3) Our software is open source and will be fully available for comparison[1].

The rest of the paper is organized as follows. We describe the related work in section II. We formalize the model and present the objective function in section III. The inference and learning algorithms are introduced in section IV and section V. We show the implementation details and the comparison of the results with the state-of-the-art approach in section VI.

II. Related Work

Human activity recognition has been extensively studied in recent decades. Different types of graphical models have been applied to solve the problem, *e.g.* Hidden Markov Models (HMMs) [1], [2], Dynamic Bayesian Networks (DBNs) [9], linear-chain CRFs [10], loopy CRFs [3], Semi-Markov Models [4], and Hidden CRFs [11], [8].

As has been discussed in the introduction, the discriminative models are more suitable for data fusion tasks which are very common in HRI applications, where many different sensors are used. Here we focus on reviewing the most related work that uses discriminative models for activity recognition.

Recently Koppula et al. [3] presented a model for the temporal and spatial interactions between human and objects in loopy CRFs. More specifically, they built a model that has two types of nodes to represent sub-activity labels of the human and the object affordance labels of the objects. Human nodes and objects nodes within the same temporal segment are fully connected. Over time, the nodes are transited to the nodes with the same type. The results show that by modeling the human-object interaction, their model outperforms the earlier work in [2] and [12]. For inference in the loopy graph, they solve it as a quadratic optimization problem using the graph-cut method [13]. Their inference method, however, is less efficient compared with the exact inference in a linear-chain structure as the graph cut method takes multiple iterations before convergence, and usually more iterations are preferred to ensure of a good solution.

Other work [14] augments an additional layer of latent variables to the linear-chain CRFs. They explicitly model the new latent layer to represent the duration of activities. In contrast with [3], Tang et al. [14] solve the inference problem by reforming the graph into a set of cliques, so that the exact inference can be solved efficiently using dynamic programming. In their model, the latent variables and the observation are assumed to be conditionally independent given the target states.

[1]The source code will be fully available at https://github.com/ninghang/activity_recognition.git

Our work is different from the previous approaches in both the graphical model and the efficiency of inference. Firstly, similar to [14], our model also uses an extra latent layer. But instead of explicitly modeling what the latent variables are, we learn the latent variables directly from the data. Secondly, we do not make conditional independence assumptions between the latent variables and the observations. Instead, we add one extra edge between them to make the local graph fully connected. Thirdly, although our graph also presents a lot of loops as in [3], we are able to transform the cyclic graph into a linear-chain structure where the exact inference is tractable. The exact inference in our graph only needs two passes of messages across the linear chain structure which is much more efficient than [3]. Finally, we model the interaction between the human and the objects at the feature level, instead of modeling the object affordance as target states. In such a way, the parameters are learned to be directly optimized for activity recognition rather than making the joint estimation of both object affordance and the human activity. As we apply a data-driven approach to initialize the latent variables, hand labeling of the object affordance is not necessary in our model. Our results show that the model outperforms the state-of-the-art approaches on the CAD120 dataset [3].

III. Model

The graphical model of our proposed system is illustrated in Fig. 1. Let $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\}$ be the sequence of observations, where K is the total number of temporal segments in the video. Our goal is to predict the most likely underlying activity sequence $\boldsymbol{y} = \{y_1, y_2, \ldots, y_K\}$ based on the observations. We define $\boldsymbol{z} = \{z_1, z_2, \ldots, z_K\}$ to be the latent variables in the model. We assume there are N_y activities to be recognized and N_z latent states.

Each observation $\boldsymbol{x}_k$ itself is a feature vector within the segment k. The form of $\boldsymbol{x}_k$ is quite flexible. It can be collections of data from different sources, *e.g.* simple sensor readings, human locations, human pose, object locations. Some of these observations may be highly correlated with each other, *e.g.* the wearable accelerate meters and the motion sensors. Thanks to the discriminative nature of our model, we do not need to model such correlation among the observations.

A. Objective Function

Our model contains three types of potentials that in together form the objective function.

The first potential measures the score of seeing an observation $\boldsymbol{x}_k$ with a joint-state assignment (z_k, y_k). We define $\Phi(\boldsymbol{x}_k)$ to be the function that maps the input data into the feature space. $\boldsymbol{w}$ is the vector of parameters in our model.

$$\psi_1(y_k, z_k, \boldsymbol{x}_k; \boldsymbol{w}_1) = \boldsymbol{w}_1(y_k, z_k) \cdot \Phi(\boldsymbol{x}_k) \quad (1)$$

This potential models the full connectivity among y_k, z_k and $\boldsymbol{x}_k$, avoiding making any conditional independence assumptions. It is more accurate to have such a structure since z_k and $\boldsymbol{x}_k$ may not be conditionally independent over

a given y_k in many cases. To make it more intuitive, one could imagine that y_k refers to the activity drinking coffee and z_k defines the progress level of drinking. The activity drinking coffee starts with human grasping the coffee cup ($z_k = 1$), then drinking ($z_k = 2$), and then putting the cup back ($z_k = 3$). Knowing it is a drinking activity, the observation $\boldsymbol{x}_k$ varies largely over different progress level z_k.

The second potential measures the score of coupling y_k with z_k. It can be considered as either the bias entry of (1) or the prior of seeing the joint state (y_k, z_k).

$$\psi_2(y_k, z_k; \boldsymbol{w}_2) = \boldsymbol{w}_2(y_k, z_k) \tag{2}$$

The third potential characterizes the transition score from the joint state (y_{k-1}, z_{k-1}) to (y_k, z_k). Comparing with the normal transition potentials [8], our model leverages the latent variable z_k for modeling richer contextual information over consecutive temporal segments. Not only does our model contain the transition between states y_k, but it also captures the sub-level context using the latent variables. Intuitively, our model is able to capture the fact that the start of reading a newspaper is more likely to be preceded by the end of the drinking activity rather than the middle part of the drinking activity.

$$\psi_3(y_{k-1}, z_{k-1}, y_k, z_k; \boldsymbol{w}_3) = \boldsymbol{w}_3(y_{k-1}, z_{k-1}, y_k, z_k) \tag{3}$$

Summing all potentials over the whole sequence, we can write the objective function of our model as follows

$$F(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}; \boldsymbol{w}) = \sum_{k=1}^{K} \{\boldsymbol{w}_1(y_k, z_k) \cdot \Phi(\boldsymbol{x}_k) + \boldsymbol{w}_2(y_k, z_k)\} + \sum_{k=2}^{K} \boldsymbol{w}_3(y_{k-1}, z_{k-1}, y_k, z_k) \tag{4}$$

The objective function evaluates the matching score between the joint states $(\boldsymbol{y}, \boldsymbol{z})$ and the input $\boldsymbol{x}$. The score equals to the un-normalized joint probability in the log space. The objective function can be rewritten into a more general linear form $F(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w} \cdot \Psi(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x})$. Therefore the model is in the class of the log-linear model.

Note that it is not necessary to model the latent variables explicitly, but rather the latent variables can be learned automatically from the training data. Theoretically, the latent variables can represent any form of data, *e.g.* time duration, action primitives, as long as it can help with solving the task. Optimization of the latent model, however, may converge to a local minimum. The initialisation of the random variables is therefore of great importance. We compare three initialization strategies in this paper. Details of the latent variable initialization will be discussed in Section VI-D.

One may notice that our graphical model has many loops, which in general makes the exact inference intractable. Since our graph complies with the semi-Markov property, next, we will show that how we benefit from such a structure for efficient inference and learning.

IV. Inference

Given the graph and the parameters, the inference is to find the most likely joint states $\boldsymbol{y}$ and $\boldsymbol{z}$ that maximizes the objective function.

$$(\boldsymbol{y}^\star, \boldsymbol{z}^\star) = \underset{(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{Y} \times \mathcal{Z}}{\operatorname{argmax}} F(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}; \boldsymbol{w}) \tag{5}$$

Generally, solving (5) is an NP-hard problem that requires evaluating the objective function over an exponential number of state sequences. Exact inference is usually preferable as it is guaranteed to find the global optimum. However, the exact inference usually can only be applied efficiently when the graph is acyclic. In contrast, approximate inference is more suitable for loopy graphs, but may take longer to converge and is likely to find a local optimum. Although our graph contains loops, we show that we can transform the graph into a linear-chain structure, in which the exact inference becomes tractable. If we collapse the latent variable z_k with the activity state y_k into a single node, the edges between z_k and y_k become the internal factor of the new node and the transition edges collapse into a single transition edge. This results in a typical linear-chain CRF, where the cardinality of the new nodes is $N_y \times N_z$. In the linear-chain CRF, the exact inference can be performed efficiently using dynamic programming [15].

Using the chain property, we can write the following recursion for computing the maximal score over all possible assignments of $\boldsymbol{y}$ and $\boldsymbol{z}$.

$$V_k(y_k, z_k) = \boldsymbol{w}_1(y_k, z_k) \cdot \phi(\boldsymbol{x}_k) + \boldsymbol{w}_2(y_k, z_k) + \max_{(y_{k-1}, z_{k-1}) \in \mathcal{Y} \times \mathcal{Z}} \{\boldsymbol{w}_3(y_{k-1}, z_{k-1}, y_k, z_k) + V_{k-1}(y_{k-1}, z_{k-1})\} \tag{6}$$

The above function is evaluated iteratively across the whole sequence. For each iteration, we record the joint state (y_{k-1}, z_{k-1}) that contributes to the max. When the process has reached the last segment, the optimal assignment of segment K can be computed as

$$y_K^\star, z_K^\star = \underset{(y_K, z_K) \in \mathcal{Y} \times \mathcal{Z}}{\operatorname{argmax}} V_K(y_K, z_K) \tag{7}$$

Knowing the optimal assignment at K, we can track back the best assignment in the previous time step $K - 1$. The process keeps going until all $\boldsymbol{y}^\star$ and $\boldsymbol{z}^\star$ have been assigned, *i.e.* the inference problem in (5) is solved.

Computing (6) once involves $O(N_y N_z)$ computations. In total, (6) needs to be evaluated for all possible assignments of (y_k, z_k), so that it is computed $N_y N_z$ times. The total computational cost is, therefore, $O(N_y^2 N_z^2 K)$. Such computation is manageable when $N_y N_z$ is not very large, which is usually the case for the tasks of activity recognition.

Next, we show how we can learn the parameters using the max-margin approach.

V. Learning

We use the max-margin approach for learning the parameters in our graphical model. The observation sequences and ground-truth activity labels are given during training

Fig. 2. Activity composition of the CAD120 dataset

CAD120 dataset. More than half the instances of the dataset are "reaching" and "moving". Therefore we consider precision and recall to be relatively better evaluation criteria than accuracy, as they remain meaningful despite class imbalance.

C. Baseline

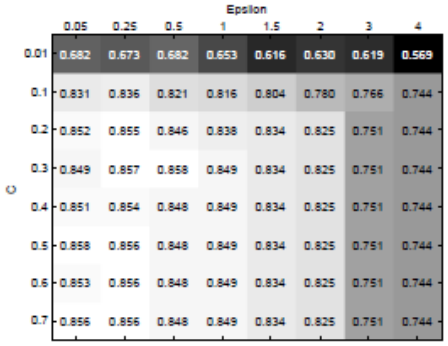
Our baseline approach uses only one latent state in our model ($N_z = 1$), which is equivalent to a linear-chain CRF. The parameters of the baseline model are learned with the standard Structured-SVM. We use the margin rescaling surrogate as the loss and L1-norm for the slacks. For optimization we use the 1-slack algorithm (primal) as being described in [22].

We apply a grid search for the best SVM parameters of C and ϵ. C is the normalization constant that is the trade-off between model complexity and classification loss. ϵ defines the stop threshold of optimization. When ϵ is small, the learning process takes longer time to converge and the trained model contains more support vectors. We show results of the grid search in Fig. 3. In Fig. 4 we show the curve of accuracy when keeping one of the parameters fixed.

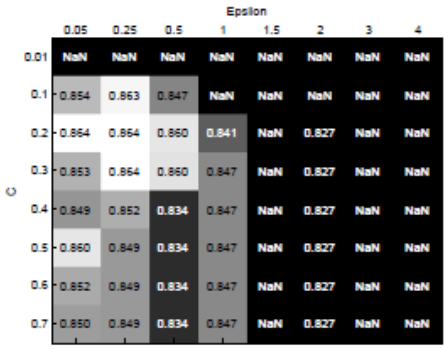
Based on these results, we choose $C = 0.3$ and $\epsilon = 0.25$ for our experiments.

D. Initialize Latent Variables

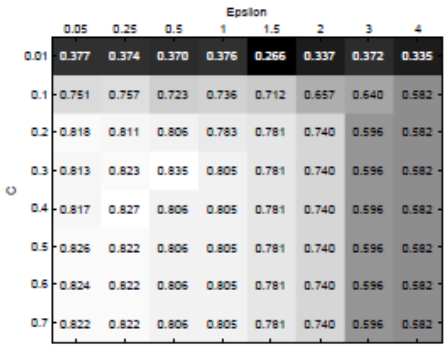
In the our latent model, we choose the same C and ϵ as in the linear-chain CRF. Parameters of the model are initialized as zeros. To initialize the latent states, we adopt three different initialization strategies. a) Random initialization. b) A data-driven approach. We apply clustering on the input data $\boldsymbol{x}$. The number of clusters is set to be the same as the number of latent states. We run K-means for 10 times. Then we choose the best clustering results that with the minimal within-cluster distances. The labels of the clusters are assigned as the initial latent states. c) Object affordance. The object affordance labels are provided by the CAD120 dataset, which are used for training in [3]. We apply the K-means clustering upon the affordance labels. As the affordance labels are categorical, we use 1-of-N encoding to transform the affordance labels into binary values for clustering.



(a) Average Accuracy

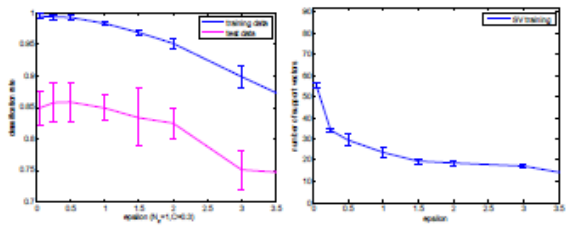


(b) Average Precision

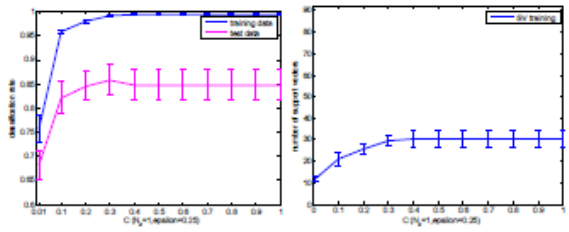


(c) Average Recall

Fig. 3. Performance of the baseline approach ($N_z = 1$). We apply a grid search to choose the best C and ϵ. The results are averaged on multiple runs of 4-fold cross-validation. The nan entry in (b) means that at least one of the classes gets no positive detection. Based on the grid search, we choose $C = 0.3$ and $\epsilon = 0.25$.



(a) $C = 0.30$, change ϵ



(b) $\epsilon = 0.25$, change C

Fig. 4. Another view of the grid search for the best C and ϵ. (a) shows the change of classification rate over ϵ when C is fixed to 0.3. When ϵ is small, a large number of support vectors is added and the model overfits. When ϵ is too large, the model is underfitting and the iterations stop too early, with too few support vectors. (b) shows the change of classification rate over C when ϵ is fixed to 0.25. When C is small, the learning algorithm tries to find a model as simple as possible, so that the performance is very low. When C is very large, the model overfits and the performance drops.

E. Results

Table I compares the activity recognition performance between our model and the state-of-the-art approach in [3]. We evaluate the model with different number of latent states, *i.e.* latent-2, latent-3 and latent-4, as well as the different initialization strategies, *i.e.* random, data-driven and affordance.

We show that with the optimal SVM parameters, the baseline performs better on the precision and recall compared with [3], but worse on the accuracy. This is because the baseline does not model the object affordance as target variables, and the parameters are optimized directly for minimizing the loss in activity recognition. The other reason is that the baseline model follows a linear-chain structure, and it is guaranteed to find the global optimal solution.

By adding the latent variables, our model can achieve better results than the baseline, but only when the latent variables are properly initialized. When the latent variables are randomly initialized, the average performance is much worse in most of the cases and shows a large variance as it most likely to have converged to a local minimum. We note that the data-driven initialization (clustering on x) performs as good as the initialization with the hand-labeled object affordances.

We also compare the model when different numbers of latent states are used. We obtain better performance when we use only 2 latent states instead 3 or 4. This is partly because there are more parameters to be tuned when the model contains more latent states. The other reason is that the

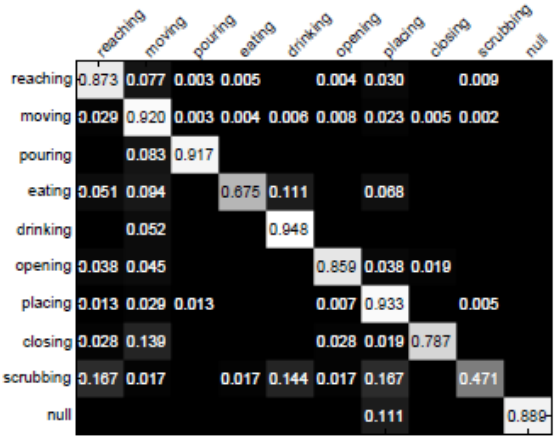


Fig. 5. Confusion matrix over different activity classes. Rows are ground-truth labels and columns are the detections. Each row is normalized to sum up to one, as one data object can only be associated with a single class label.

model may be too complex and overfits the data. Therefore choosing the number of latent states is also data related. If we use a more complex dataset, more latent states need to be used.

Fig. 5 shows the confusion matrix of activity classification. We can see that higher values present on the diagonal of the confusion matrix, and they represent the activities that are correctly classified. The most difficult classes are eating and scrubbing. Eating is sometimes confused with the drinking, and scrubbing is likely to be confused with reaching, drinking and placing.

Our best performance is obtained when we use 2 latent states and the model is initialized by clustering on the input data. We get 89.2% on the average precision and 83.1% on the average recall, which outperforms the state-of-the-art by over 5% on both precision and recall. We believe the performance can be further improved if we apply grid search for the optimal learning parameters of the latent-state model.

VII. Conclusion and Future Work

In this paper, we present a novel Hidden-state CRF model for human activity recognition. We use the latent variables to exploit the underlying structures of the target states. By making the observation and state nodes fully connected, the model do not require any conditional independence assumption between latent variables and the observations. The model is very efficient in that the inference algorithm is applied to a linear-chain structure. The results show that the proposed model outperforms the state-of-the-art approach. The model is very general that it can be easily extended for other prediction tasks on sequential data.

References

[1] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2154–2159.

TABLE I
RESULTS OF ACTIVITY RECOGNITION

	Accuracy	Precision	Recall	F-score
Koppula, et al. IJRR13' [3]	86.0 ± 0.9	84.2 ± 1.3	76.9 ± 2.6	80.4 ± 1.5
latent-1 linear-chain CRF	85.7 ± 2.9	86.4 ± 6.1	82.4 ± 4.0	82.6 ± 6.2
latent-2 random	84.0 ± 2.8	85.6 ± 4.6	79.5 ± 5.4	80.1 ± 6.5
latent-2 data-driven	$\mathbf{87.0 \pm 1.9}$	$\mathbf{89.2 \pm 4.6}$	$\mathbf{83.1 \pm 2.4}$	$\mathbf{84.3 \pm 4.7}$
latent-2 affordance	$\mathbf{87.0 \pm 2.1}$	$\mathbf{88.3 \pm 4.3}$	$\mathbf{84.0 \pm 3.2}$	$\mathbf{84.3 \pm 5.1}$
latent-3 random	83.1 ± 2.2	86.1 ± 4.5	76.3 ± 4.8	78.1 ± 6.1
latent-3 data-driven	86.0 ± 1.9	87.2 ± 2.9	82.3 ± 2.4	82.9 ± 4.2
latent-3 affordance	86.0 ± 2.0	88.0 ± 4.6	81.5 ± 3.4	82.1 ± 4.8
latent-4 random	82.8 ± 3.2	85.9 ± 5.0	76.3 ± 5.6	77.5 ± 6.9
latent-4 data-driven	85.9 ± 1.7	86.8 ± 2.7	82.4 ± 2.0	82.8 ± 3.7
latent-4 affordance	85.7 ± 1.6	86.4 ± 2.8	81.7 ± 2.9	82.0 ± 3.6

[2] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 842–849.

[3] H. Koppula and A. Saxena, "Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation," *International Journal of Robotics Research (IJRR)*, 2013.

[4] T. van Kasteren, G. Englebienne, and B. J. Kröse, "Activity recognition using semi-markov models on real world smart home datasets," *Journal of Ambient Intelligence and Smart Environments*, vol. 2, no. 3, pp. 311–325, 2010.

[5] B. K. Ninghang Hu, Gwenn Englebienne, "Posture recognition with a top-view camera," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[6] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 29, no. 10, pp. 1848–1852, 2007.

[7] L. Maaten, M. Welling, and L. K. Saul, "Hidden-unit conditional random fields," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 479–488.

[8] Y. Wang and G. Mori, "Max-margin hidden conditional random fields for human action recognition," in *Proc. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 872–879.

[9] Y.-c. Ho, C.-h. Lu, I.-h. Chen, S.-s. Huang, C.-y. Wang, L.-c. Fu, *et al.*, "Active-learning assisted self-reconfigurable activity recognition in a dynamic environment," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 1567–1572.

[10] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proc. International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2007, p. 235.

[11] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2006, pp. 1521–1527.

[12] B. Ni, P. Moulin, and S. Yan, "Order-preserving sparse coding for sequence classification," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 173–187.

[13] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *Proc. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–8.

[14] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1250–1257.

[15] R. Bellman, "Dynamic programming and lagrange multipliers," *The Bellman Continuum: A Collection of the Works of Richard E. Bellman*, p. 49, 1986.

[16] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," pp. 1453–1484, 2005.

[17] C.-N. Yu and T. Joachims, "Learning structural svms with latent variables," in *Proc. of International Conference on Machine Learning (ICML)*. ACM, 2009, pp. 1169–1176.

[18] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (cccp)," *Advances in Neural Information Processing Systems (NIPS)*, vol. 2, pp. 1033–1040, 2002.

[19] J. E. Kelley, Jr, "The cutting-plane method for solving convex programs," *Journal of the Society for Industrial & Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[20] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, 2007, vol. 382.

[21] J. M. Mooij, "libDAI: A free and open source C++ library for discrete approximate inference in graphical models," *Journal of Machine Learning Research*, vol. 11, pp. 2169–2173, Aug. 2010.

[22] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.